

Master's Thesis – Human Computer Interaction

**End-user Development for Machine Learning  
with Shared Mobility Data**

*Submitted by:* Ahmad Maher Taha AlHadidi

[ahmad.alhadidi@student.uni-siegen.de](mailto:ahmad.alhadidi@student.uni-siegen.de)

*Matriculation Number:* [REDACTED]

*First Examiner: Prof. Dr. Gunnar Stevens*

*Second Examiner: Prof. Dr. Volker Wulf*

*Supervised by: Mr. Lukas Böhm*

*Human Computer Interaction*

*University of Siegen*

*Siegen -* [REDACTED]

## **Acknowledgements**

I would like to thank my examiners, Professor Doctor Gunnar Stevens and Professor Doctor Volker Wulf for their constant support on this thesis and for sharing their invaluable knowledge during the course of my studies. Additionally, my sincere gratitude and deepest appreciation go to my supervisor, Mr. Lukas Böhm, for without his unwavering trust and unconditional support, this thesis and its Master's project counterpart could not have been possible.

## **Dedication**

I dedicate this thesis to my father, Maher AlHadidi, my brother, Anas AlHadidi, and to my cousin, Fadel AlHadidi.

## **Abstract**

As transportation systems and shared mobility programs continue to grow in cities, concerns related to increasing traffic congestions and pollutants continue to grow as well. With recent practices of datasets being published by shared mobility programs to the public, people have taken the opportunity to use them to perform studies and to develop applications. In this thesis, a tool for “End-user Development for Machine Learning with Shared Mobility Data” has been designed. It aims to provide non-expert users with the ability to employ novel machine learning algorithms to perform complex computations that can help in the decision-making process of various challenges posed by the transportation system. It derives its foundations from multiple fields that were reviewed, such as, Interactive Machine Learning, End-User Development, and Shared Mobility Services. Due to its unconventionality, the design process of its interface was governed by a set of design guidelines and principles that were specifically defined for Interactive Machine Learning applications. The tool and the design of its user interface employs novel techniques and technologies to visualize and to analyze shared mobility datasets, such as, Kepler.GL, Opendash, and Highcharts. Design Science Research was used as a research methodology, and FEDS was used to guide the evaluation process of the produced research artifact. Several Streamlined Cognitive Walkthroughs were used by the author to evaluate the developed medium-fidelity prototypes after their low-fidelity variants were initially created.

# Table of Contents

1.	Introduction.....	1
2.	Related Works.....	3
2.1.	End User Development .....	4
2.1.1.	Many Eyes: A Site for Visualization at Internet Scale .....	4
2.1.2.	End-user Development of Information Visualization.....	5
2.1.3.	Cuscus: An End User Programming Tool for Data Visualisation .....	6
2.2.	Interactive Machine Learning .....	6
2.2.1.	A Review of User Interface Design for Interactive Machine Learning.....	7
2.2.2.	Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators.....	12
2.3.	Shared Mobility Services .....	13
2.3.1.	Bike Sharing Atlas .....	13
2.4.	Public Domain.....	15
2.4.1.	Visualization.bike .....	15
2.4.2.	Shared Mobility Flows – Berlin.....	20
3.	Methods and Material .....	21
3.1.	Methods.....	21
3.1.1.	Design Science Research .....	22
3.1.2.	FEDS: A Framework for Evaluation in Design Science Research.....	22
3.1.3.	Streamlined Cognitive Walkthrough .....	24
3.2.	Material .....	25
3.2.1.	Opendash.....	25
3.2.2.	Kepler.GL .....	26
3.2.3.	Highcharts .....	32
3.2.4.	Dataset.....	32
4.	Tool Design.....	33
4.1.	Low-Fidelity Prototype .....	33

4.1.1.	Feature Selection.....	33
4.1.2.	Model Selection .....	35
4.1.3.	Model Steering.....	37
4.1.4.	Model Steering Part 2 .....	39
4.1.5.	Quality Assessment.....	40
4.1.6.	Termination Assessment .....	41
4.1.7.	Transfer .....	41
4.1.8.	Model Browser.....	42
4.2.	Medium-fidelity Prototype .....	43
4.2.1.	Feature Selection.....	44
4.2.2.	Model Selection .....	45
4.2.3.	Model Steering.....	45
4.2.4.	Quality Assessment.....	46
4.2.5.	Termination Assessment .....	47
4.2.6.	Transfer .....	47
4.2.7.	Model Browser.....	48
5.	Evaluation .....	49
5.1.	Happy Scenario: New Model .....	49
5.2.	Happy Scenario: New Model – Revisited .....	51
5.3.	Happy Scenario – Complex.....	52
5.4.	Happy Scenario – Complex – Revisited.....	53
5.5.	Loaded Model.....	54
5.6.	User Wants to Define a New Use of ML .....	54
5.7.	User Defines a New Use of ML .....	54
5.8.	Conclusion.....	55
6.	Discussion .....	56
7.	Future Work and Limitations .....	57
8.	Conclusion .....	58

9.	References.....	59
10.	Appendix.....	65
10.1.	SCW 1 – Happy Scenario: New Model.....	65
10.2.	SCW 2 – Happy Scenario: New Model – Revisited.....	69
10.3.	SCW 3 - Happy Scenario – Complex .....	74
10.4.	SCW 4 – Happy Scenario – Complex – Revisited .....	79
10.5.	SCW 5 – Loaded Model .....	84
10.6.	SCW 6 – User Wishes to Define A New Use of ML .....	87
10.7.	SCW 7 – User Defines A New Use of ML.....	91

## List of Figures

Figure 1: Many Eyes Visualization.....	5
Figure 2: IML Workflow by Dudley & Kristensson [26] .....	9
Figure 3: Bike Demand Prediction Tool [53] .....	12
Figure 4: Bike Sharing Atlas - Stations Page.....	14
Figure 5: Bike Sharing Atlas - Bike Sharing Network Utilization .....	15
Figure 6: System page of Visualization.bike (1/2) .....	16
Figure 7: System page of Visualization.bike (2/2) .....	17
Figure 8: Heatmap Page from Visualizaion.bike .....	18
Figure 9: Station Analysis - Visualization.Bike.....	19
Figure 10: Shared Mobility Flows Prototype.....	20
Figure 11: Design Science Research Cycles diagram by Hevner [25] .....	22
Figure 12: Master's Project Finalized Work .....	28
Figure 13: Points Datalayer.....	30
Figure 14: The Trips and The Clusters Datalayers .....	31
Figure 15: Heatmap Datalayer .....	31
Figure 16: Feature Selection - Variant 1 .....	33
Figure 17: Feature Selection - Variant 2 .....	34
Figure 18: Feature Selection - Variant 3 .....	34
Figure 19: Model Selection - Variant 1 .....	36
Figure 20: Model Selection - Variant 2 .....	36
Figure 21: Model Selection - Variant 3 .....	37
Figure 22: Model Selection - Variant 4 .....	37
Figure 23: Model Steering - Variant 1 .....	38
Figure 24: Model Steering - Variant 2 .....	39
Figure 25: Model Steering Part 2 - Variant 1.....	39
Figure 26: Quality Assessment - Variant 1 .....	40
Figure 27: Quality Assessment - Variant 2 .....	40
Figure 28: Termination Assessment – Variant 1 - Below Inflection Point.....	41
Figure 29: Termination Assessment - Variant 1 - Inflection Point Imminent .....	41
Figure 30: Transfer - Variant 1 .....	41
Figure 31: Transfer - Variant 2 .....	42
Figure 32: Model Browser .....	43
Figure 33: Feature Selection - Collapsed and Expanded .....	44



Figure 34: Model Selection.....	45
Figure 35: Model Steering - Collapsed and Expanded .....	45
Figure 36: Quality Assessment - Collapsed and Expanded .....	46
Figure 37: Termination Assessment Interface .....	47
Figure 38: Transfer.....	47
Figure 39: Model Browser .....	48
Figure 40: SCW - 2 .....	51
Figure 41: SCW - 4 .....	53
Figure 42: SCW - 5 .....	54
Figure 43: SCW - 7 .....	54
Figure 44: Complete UI - Dark Mode.....	55
Figure 45: Complete UI - Light Mode .....	55

## List of Abbreviations

<b>BSO</b>	Bike Sharing Operator
<b>BSS</b>	Bike Sharing Systems
<b>CSCW</b>	Computer Supported Cooperative Work
<b>CSV</b>	Comma-Separated Value
<b>CW</b>	Cognitive Walkthrough
<b>DSR</b>	Design Science Research
<b>EUD</b>	End User (Development / Developer)
<b>GEOJSON</b>	Geographic Java Script Object Notation
<b>GHG</b>	Green House Gasses
<b>IML</b>	Interactive Machine Learning
<b>InfoVis</b>	Information Visualization
<b>JSON</b>	Java Script Object Notation
<b>LC</b>	Learned Concept
<b>MDS</b>	Mobility Data Specification
<b>ML</b>	Machine Learning
<b>SciViz</b>	Scientific Visualization
<b>SCW</b>	Streamlined Cognitive Walkthrough
<b>SMS</b>	Shared Mobility Services
<b>SOV</b>	Single Occupancy Vehicle
<b>UI</b>	User Interface

## 1. Introduction

As the transportation sector has been identified to be one of the culprits in the production of air pollutants, green-house-gasses (GHG) [1], and noise pollution [2]. Countries around the globe have been urging their citizens to lower both their consumption and their dependency on motor vehicles [3], mainly when they are used by a single occupant (Single-Occupancy Vehicle-SOV) [2]. With 95% of the vehicles using fossil fuel [4] and causing traffic congestion in dense cities [5], the governments have also been setting targets to reduce the amount of fuel being used [6]. For example, Germany has set a target that aims to reduce the amount of CO<sub>2</sub> emissions up to 60% by 2050 [4], and it is carrying out a process known as “Mobilitätswende” (Mobility Transformation), which aims to convert the use of fossil-fuels to sustainable sources of energy and to better connect local public transport with individual forms of transport [7, 8].

Meanwhile, increasing public awareness, gaining a positive image [4], being offered to the masses through bike-sharing services (BSS), and the governments promoting their use [3, 9, 10], cycling has faced significant success during the recent years with metropolitans using shared-bicycles as an alternative mode of transport in large cities. Not only using BSS was found to be a faster alternative than the conventional modes of transport [11], it has been recorded to produce fewer emissions [12], and has been claimed to require less additional maintenance [13].

As a result, shared mobility services (SMS) in general and BSS in particular have seen an “explosive” growth in recent times [14, 15].

While BSS are widely available in major metropolitan areas, that does not seem to be the case for smaller cities and provincial areas. However, there may be other forms of shared mobility services. Nevertheless, available SMS are not only popular with metropolitans and their municipalities but they have also been reported to be popular among urban planners and researchers as well. With the challenges of urban planning and the growing urbanization of areas around the cities, the recent integration of IoT devices onboard the fleet of these services have provided valuable insight for the planners to be able to better plan ahead [16], to enforce laws that ensure a better state of well-being for the metropolitans [17], and as a learned lesson, to better manage the introduction of a new and superior shared mobility service in the city to avoid negative responses from the public [18]. For example, acts of vandalism occurred to shared bikes supplied by the Singaporean company “Obike” in Munich: These acts - stemming from the protest of the citizens - has caused the company to file for bankruptcy and has led to the abandonment of the bikes around the city [19, 20].

The recent practice of publishing anonymized trip data generated from these IoT devices, and having them “often openly accessible”[21] to the public through Application Programming Interfaces (API) has enabled researchers from multiple disciplines to conduct various studies on the matter. Such studies can be understood as an attempt to achieve certain goals, which with the availability of this data, it has allowed the researchers to provide results that are based on real trips as opposed to simulated ones.

To mention a few examples: Caggiano & Ottomanelli used such data to investigate a problem that has been plaguing BSS known as “distribution imbalance” by supplying it to a neural network [22], while Jiao & Bai [23] used Shared E-Scooter data to identify hotspots and to explain the behavior of metropolitans during certain times of the year in Austin, Texas. While both of these works tackle two different research questions, this data has more potential left to offer. The creation of an interactive interface that combines this data with the fields of End-user development (EUD), Interactive Machine Learning (IML), Spatio-temporal analysis, and Scientific Visualization (SciViz) has been little researched. Therefore, it is a research gap that should be explored as it is believed that it can become a tool that enhances the decision-making process of policymakers, and potentially, the aforementioned stakeholders of SMS.

With users usually perceiving machine learning (ML) as a “black box” [24], it is therefore the primary goal of this research to demystify this perception by designing an interactive interface in a manner that enables non-expert users to perform complex analysis on shared mobility datasets through the use of machine learning methods.

The underlying foundation that produced the system (Miaas - Mobility Intelligence As A Service), and its details that facilitated this work is the result of a joint effort between Universität Siegen and Open Inc. However, an explanation of this work is beyond the scope of this thesis and will not be further discussed.

In this thesis, we will deal with the following research question:

*“How can a user interface be designed that enables non-experts to apply mobility-specific machine learning methods?”*

The overall approach that will be engaged to answer this research question will begin by the application of the design science research by Alan R. Hevner, this will be used as the approach to

perform the literature review and to create the research artifact [25], the “Rigor Cycle” used in the paper will be used for each discipline to obtain the requirements necessary for the interface. The design of the interface, in other words, the research artifact, will be guided using the set of the proposed guidelines by Jon Dudley and Per Kristensson in their recent study “A Review of User Interface Design for Interactive Machine Learning” in 2018 [26]. Resultingly, the interface will be evaluated using the “Streamlined Cognitive Walkthrough” method proposed by Spencer in 2000 [27].

Ideally, this work will yield the following outcomes: 1) to provide an interface that is influenced by the above-mentioned fields. 2) To satisfy a reasonable number of the requirements that were obtained from the body of literature. 3) To design and implement the interface in an extendible manner. 4) To contribute to the body of literature by creating a new field that has the above-mentioned fields as its foundation.

An initial implementation of the extensible geospatial analysis tool (Kepler.GL [28]) was previously performed in the form of a widget within the dashboard system that was provided by Open Inc (Opendash [29]) in the scope of a Master’s project. As a result, the BSS trip data that was supplied by the supervisor has been connected to Kepler.GL. The result of this work facilitated the possibility to visualize, analyze, and to interact with the trip data through Kepler.GL interactively. Furthermore, a collection of charts in an extended tab through Kepler.GL using the dashboards’ built-in charting framework (Highcharts [30]) was provided. Thus, this thesis will emphasize the design aspect of an extended interface within Kepler.GL.

The remainder of this thesis proceeds as follows: In chapter 2, a presentation of the literature review on the related fields are discussed. Chapter 3 covers the methods and materials that were implemented. Low-fidelity and medium-fidelity prototypes are presented and discussed in chapter 4. The evaluation of the interface is performed in chapter 5. A discussion on the research question and the resulting works’ implications in chapter 6, future work and limitations in chapter 7, and the conclusion in chapter 8.

## **2. Related Works**

Due to the limited research that exists for the previously discussed context, an opportunity is presented to provide a literature review of the foundational fields that are believed to comprise an AI-enabled shared-mobility tool for non-experts. Therefore, each subsection will be covered in a manner that briefly describes it, provides a definition of its field and a summary of the work with an emphasis on the extracted findings that relate to this thesis.

## 2.1. End User Development

EUD is a nascent discipline that has risen from several fields, such as, Human-Computer Interaction, Artificial Intelligence, CSCW, Software Engineering, Cognitive Science, Information Systems and Psychology of Programming [31]. According to Lieberman et al., EUD can be defined as the “set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point create, modify and extend a software artifact” [32]. This discipline mainly strives to enable its users to “create, modify and extend software artifacts, and as a result gain more control over their applications by engaging them in development” [33]. Researchers in this field conduct their research to present several aspects related to the field, such as: models, applications, design guidelines, and frameworks [34].

In the following sub-sections, a summary of the relevant articles that discuss tools created for this field and their relevant findings intended for the tool will be presented.

### 2.1.1. Many Eyes: A Site for Visualization at Internet Scale

Published in 2007, Viegas, Wattenberg, and Van Ham et al. published a paper that discusses “the design and the deployment” of a website (Many Eyes) which enables non-expert users to discuss and interactively create visualizations on the internet [35].

The tool was developed with the main target of enabling collaboration between the users and providing a space in which they could discuss these visualizations and the datasets they are comprised of. The tool allowed the users to upload their datasets using an HTML form that accepts free text or tabs-delimited records. The upload functionality was created in this manner as it was believed to be an easy action to perform a copy-and-paste function from spreadsheet editors to the form. The data set columns that it accepts can be either textual or numerical. Moreover, upon importing the dataset into Many Eyes, it guesses the columns that are in the data set to reduce the data preparation effort that is required from the user.

Another use case of the tool is for users to browse through the available data sets by visiting its corresponding page. Each page contains identifying information about the data set, a partial preview of its contents, and a ‘discussion forum’ for the users.

Many Eyes supplied several visualizations that the users could use to present their datasets. While some of these visualizations were common, such as bar charts, there were also “experimental” visualizations, such as bubble charts, that the users could use. The interaction with the visualizations was facilitated by clicking on the available elements for each visualization. For example, Figure 1 presents the possibility of modifying the visualization by selecting different items, such as

binding the x-axis of the visualization to a different feature in the dataset from the drop-down menu available under each visualization.

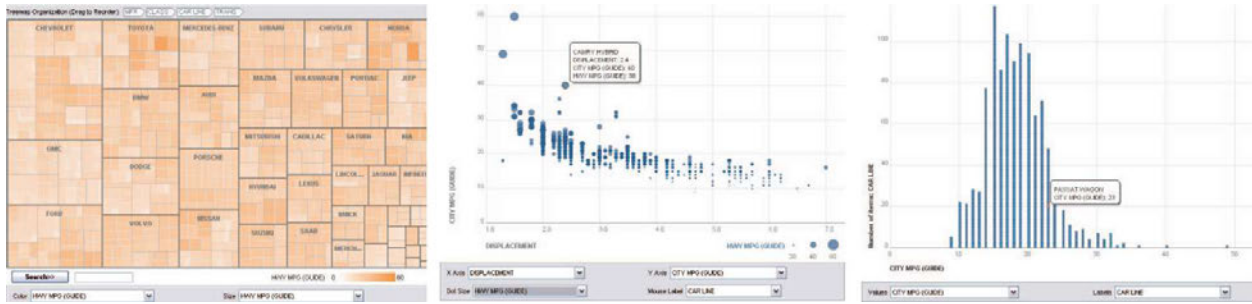


Figure 1: Many Eyes Visualization

The tool provides the possibility to leave “visualization bookmarks” which allow the users to place a bookmark on visualizations that could be discussed asynchronously. Moreover, this user can leave a bookmark after altering the visualization, for example, swapping the axis of a visualization to have a discussion about it. This enables the community to not only have a discussion on one visualization state of the data but other states made by other users as well. Finally, the authors found that the users were not always expecting bar charts to present an average of certain numerical columns, they were also expecting summation.

These highlighted features can be beneficial to the intended tool. Hence, a tool – or a framework - that can offer these features natively can be useful to the users. While this is partly true for Kepler.GL from the features related to the data preparation step, the same cannot be said to the coordinative features between users. However, these can be implemented at a different stage.

### 2.1.2. End-user Development of Information Visualization

Pantazos, Lauesen, and Vatrapu have published an article that surveys 18 information visualization (InfoVis) tools in the context of end-user development in 2013 [33]. The authors argue that the tools surveyed “are representative of the InfoVis field and that have contributed significantly to it”, the survey was carried out with questions that relate to the accessibility of the tools from the perspective of end-user developers, such as, the ability to customize visualizations, to what extent can it be done, and if present, how can it be performed.

The authors noted that the surveyed tools ranged in the programming proficiency required to create visualizations. Some of the tools offered the creation of visualizations using a programming language, others did so with wizards, drag-and-drop interactions, or both.

The paper concludes with the following insights: 1) A tool that provides its users the ability to modify the visualizations through a programming language becomes less accessible to end-user developers.

2) The usage of drag-and-drop actions and wizards to create visualizations enables users to easily connect the data and its resulting visualization cognitively, where the programming language falls short.

### 2.1.3. Cuscus: An End User Programming Tool for Data Visualisation

Published in 2019, the work of Marasoiu, Nauck, and Blackwell discusses a tool (Cuscus) that was developed for end-users to create visualizations that are difficult or even impossible to make using popular data analysis tools [36]. The datasets can be imported through Comma-Separated Value (CSV) files and the workspace can be saved and loaded through the use of JavaScript Object Notation (JSON) files. Resembling a combination of a spreadsheet editor and a graphics editor, the tool was used to import weather data and to recreate famous visualizations, such as the visualization of the progression of Napoleon's Grand Army on its march to Russia in 1915 [36].

The authors, having performed a user study on the tool with data journalists and business analysts, mentioned several insights: 1) The users found it pleasant that they were able to interact with the visualized shapes. 2) The ability of presenting changed visualizations after editing the dataset in the spreadsheet editor is a feature that has led to positive comments. 3) Some of the participants noted that they prefer to spend the least amount of time possible while creating visualizations as they are interested in performing a preliminary data exploration first. 4) A participant suggested that in order to use the tool, it would require a larger monitor to be able to completely see the information. 5) The authors noted that the users implemented an unintentional color mapping on some of the functionalities. 6) There was not a great demand of mental effort by the users to use the tool. 7) As suggested by one of the participants, the visualization should automatically resize itself according to the window that it is being displayed in - this can be perceived as a further improvement to the 4<sup>th</sup> insight mentioned. 8) The visualization presented by the tool evolves with each step the user performs. 9) The users noted that an "undo" button is also needed.

The authors concluded that the tool was perceived to be flexible and that it offers the creation of several visualizations. They received detailed feedback on usability that they plan to include in the next releases of the tool.

## 2.2. Interactive Machine Learning

Ever since it was coined by Fails and Olsen in 2003 [37], the field of Interactive Machine Learning (IML) has been receiving increasing interest by the researchers worldwide, it is widely used as a means to assist end-user developers to employ machine learning models on problems that require a vast amount of computations. Such problems are tackled through an IML software interface to reach a solution to the problem. Dudley and Kristensson define IML as "an interaction paradigm in which



a user or user group iteratively builds and refines a mathematical model to describe a concept through iterative cycles of input and review” [26]. The uses of IML are several, to name a few, it can be used to classify images [38], to recognize speech [39], and to assist in writing user reviews [40]. However, as noted by [26], the creation of IML interfaces is not an easy task for the designer in comparison to designing conventional interfaces. In this section, relevant articles on the design of user interfaces in the context of IML will be presented, as well as an article that implements IML practices in BSS.

### 2.2.1. A Review of User Interface Design for Interactive Machine Learning

This article explores the workflows, principles, and challenges that an IML interface designer might face by performing a detailed revision of the body of literature related to the field.

The authors argue that there are four challenges that are encountered when an IML application is designed: 1) As the user is human, they can be prone to make mistakes, to be biased, and to be generally inconsistent while training the model. In a setting where the feedback from the end-user is key to reach the desired result from the model being trained, such mistakes or “inconsistencies” can lead to an undesired state of the model. 2) The users face a certain degree of doubtfulness while they are interacting with the IML application, this challenge entails that: as the user selects (or does not select) a feature from a dataset, it is not a given that the model will consider this input (or lack thereof) as a clue to be considered. It is therefore necessary that the designer should not excessively limit the user in terms of the available functionalities. 3) Due to the fact that ML models are probabilistic, the output of the model can be wrong, a single edge case in the model that produces an incorrect answer can confuse the user [26, 41]. 4) Compared to non-IML software - where deeming a task to be complete can be answered in a binary manner - in IML, the completeness of the task cannot be binarily ascertained. Furthermore, if a model could be trained to reach 100% accuracy, it can be argued that that the task “maybe both undesired and impossible” [26].

Due to these challenges, the authors performed an extensive study on the past IML applications available to propose solution principles for IML interface designers, the elements of an IML interface, and a workflow that comprises a “Comprehensive IML Process” to serve as an anchor point for the designer. These are used to distinguish the progression of the tasks while the UI is being designed.

Quoting the six solution principles that were listed by the authors to serve as a means to assist the IML interface designer are the following:

1. *Make task goals and constraints explicit.*
2. *Support user understanding of model uncertainty and confidence.*

3. *Capture intent rather than input.*
4. *Provide effective data representations.*
5. *Exploit interactivity and promote rich interactions.*
6. *Engage the user.*

**Principle 1**

Entails that the designer is required to design the interface in a manner that communicates to the user what the IML system can do and what it cannot do. When the interface is clear, the user can formulate a suitable strategy, it decreases their inconsistency, it helps the user identify a point where the model is suitable for the task at hand, and it aids the user in a manner that helps them to avoid overfitting the model.

**Principle 2**

Reiterates the fact that ML comes with a degree of uncertainty. As the user “steers” the IML system towards the desired result, it is important for the interface to communicate to the user that the model can produce an undesired result. Managing the expectations of the user from the IML system and its performance due to this “inevitable feature” is a challenge that must be addressed by the designer.

**Principle 3**

Remarks on the notion that the designer must capture the intent from the users rather than their input. It can be perceived that as the user selects features and executes training cycles on the IML system, there exists a possibility where the user intentionally included/excluded a feature from the dataset in the training phase to steer the model to output results correctly. The intent of the user to leave a feature unselected (or selecting it) does not yield significance to an algorithm or a machine learning model, regardless of how significant this action (or lack thereof) is to the user. For example, a proposed solution to this principle was addressed by [42]; in which they designed the interface in a manner that requires the user to select “a good example” and “a bad example” that the model should consider as it trains. The authors argue that in this manner, the intent is captured, and the interface does not limit the user from training the model with excessive precise examples.

**Principle 4**

Proposes to the designer to find the balance between avoiding the oversimplification of presenting the data and overcomplicating it. However, it should be done in a manner that raises the cognitive abilities of the users while they are engaged in reviewing the sample data. Also, the interface should be designed in a manner that makes this task be done in the fastest way possible.

### Principle 5

Instructs the designer to strive for making interfaces that the user can use in a manner that captures their intents and grants them (and gains from them) insight into the process. The interface should present to the user the widest range of interactions possible to aid in the creation of a correct mental model [26, 43]. Furthermore, the authors highlight an article by both [44, 45] which states that, as the users interact, they understand the task at hand more. However, the interface should be designed in a manner that acknowledges that users are prone to err, therefore, the ability to navigate to previous states of the model is recommended to be available as well. Moreover, [37] and [43] find that rapid interactivity without prolonged waiting periods yields positive results, [37] also believe that the use of quick algorithms, as opposed to slower ones with higher induction capabilities, yield higher significance to the users.

### Principle 6

Recommends that including motivating aspects into the design of the interface encourages the user to reach the required result of the IML system. The authors find that this was a topic of discussion by several studies, for example, [46] finds that engagement from the user can be maintained by reducing the amount of boolean (Yes / No) questions. Furthermore, the frequency of the feedback that is provided by the IML system should be done in a manner that respects the field that it is implemented in, a study finds that this frequency can be higher when the IML system is aimed at users that are operating in STEM fields [47].

The proposed workflow (Figure 2) of an IML system is illustrated by the authors as follows:

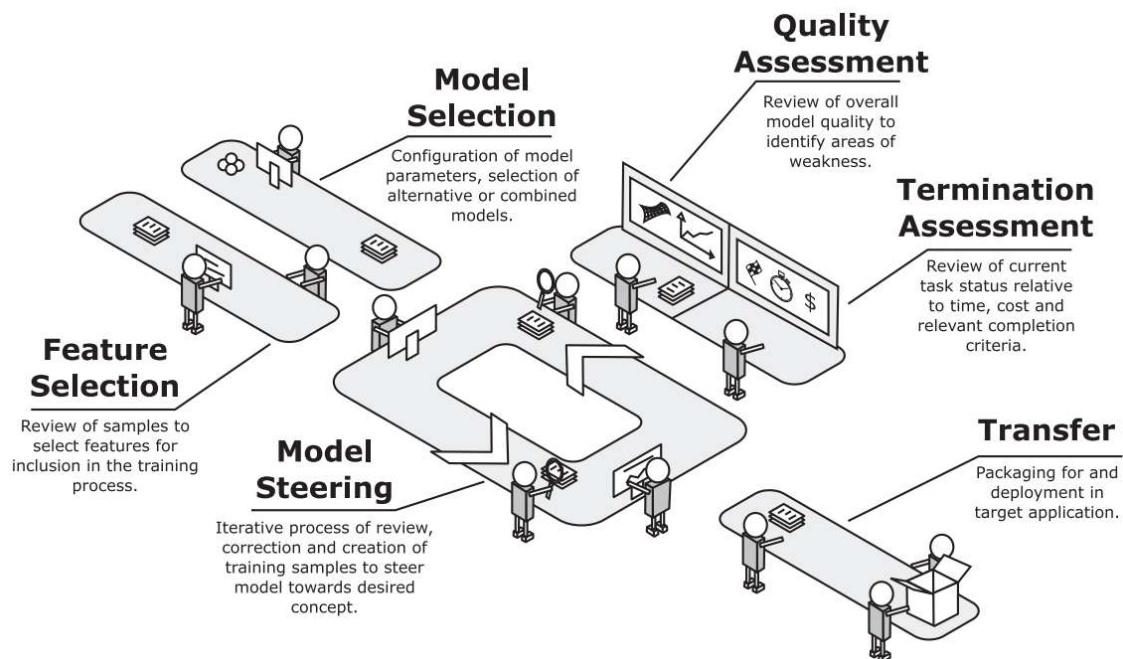


Figure 2: IML Workflow by Dudley & Kristensson [26]

As the authors explain, this workflow resembles a “behavioural breakdown into distinct user activities”, it describes the process that the users encounter while they interact with an IML system.

### **Feature Selection**

The users during this activity select the features from the dataset (columns) that are desired to be included during the training process. The authors found that facilitating the means for the user to select features are both efficient and yield higher quality [26, 48, 49]. It was also highlighted that the users found the feature selection activity is more relevant than the outcome of the classification from the IML, let alone it being an enjoyable task as well [26]. Furthermore, as the users select the features, not only does it provide more interpretability, it might also increase the understanding of the user in upcoming activities.

### **Model Selection**

In this step, the user is able to select a ML algorithm and to experiment with them. Not only does allowing the user to select an algorithm can help to reduce the rate of interventions needed by ML practitioners, but it also allows users to form an understanding of the model, regardless of how correct it is. Users can compare the models from many aspects, such as the time needed to finish executing and the estimated accuracy of the results.

### **Model Steering**

It can be argued that in this task, the user spends the most amount of work. Here, the user directs the model towards the desired direction and adjusts the training data in a series of interactions accordingly. The authors highlight here several issues: 1) As the user is a non-expert, it is not a given that they will follow an iterative fashion [50]. 2) It is possible that the user might experience boredom from this iterative interaction, thus negatively affecting the outcome of the model (learned concept<sup>1</sup>). 3) To maintain the engagement between the user and the IML system, as mentioned before, [37] believes that the use of quick algorithms, as opposed to slower ones with higher induction capabilities, yield higher significance to the users.

### **Quality Assessment**

During this task, the user performs an iterative cycle approach to minimize the number of errors produced and improves the accuracy of the learned concept. In order for the user to perceive the accuracy of the model, the interface has to provide to the user the ability to view the accuracy metric that the LC is at (c.f. **Principle 4**) while assisting the user to avoid overfitting the model.

---

<sup>1</sup> The authors in [26] refer to the model as it is being trained as “the learned concept”

### **Termination Assessment**

The authors and other research remarks upon this activity of when the user stops training the model. The importance of this step is high because at a certain point during the training process, the user concludes that the model has been through sufficient training and that it is producing the desired results [26, 51]. The authors identify this certain point as the “inflection point”, where crossing it would expose the model to a higher chance of overfitting it and a reduced amount of gained accuracy of training past it. Therefore, a feedback mechanism embedded in the interface can be implemented to inform the user that the model has achieved this point and warns them about the undesired overfitting phenomena.

### **Transfer**

The act of presenting the model in its target environment is coined as the transfer activity. In this task, the model undergoes a process in which it ends in visualizing the result of executing the previous workflow.

---

## **IML Interface**

According to the reviewed literature by the authors, they highlighted that an IML interface consists of 4 elements: 1) Sample Review 2) Feedback Assignment 3) Model Inspection 4) Task Overview. They suggest that the interface and the targeted user experience can vary from application to application based on the use case and the context.

### **Sample Review**

This element presents to the user a sample output of the model forming its current state, this provides insight to the users on what should be done in the next iteration.

### **Feedback Assignment**

This element enables the user to assign their feedback on the model by setting labels, choosing features, and/or producing new samples. It is highlighted that this element requires careful design, as a balance must be found by not overwhelming the user with requesting excessive feedback against the amount of control that is offered to the user.

### **Model Inspection**

An IML interface contains an element that provides the user the ability to inspect the model in a manner that allows them to view the model’s quality, accuracy, confidence, and coverage. Furthermore, this interface element provides to the user significant visibility regarding the model’s

performance, therefore, preventive measures while designing this element can be necessary to avoid the user from overfitting the model.

### Task Overview

An IML interface can provide to the user additional information regarding the system, in addition to the status of the task and its termination conditions. Similar to Jakob Nielson’s “Visibility of System Status” usability heuristic [52], a dedicated element in such a system can provide further assistance to the user, such as communicating the operational status of the IML system, the required duration of a task, and the “availability of training data” [26].

#### 2.2.2. Development of a station-level demand prediction and visualization tool to support bike-sharing systems’ operators

Boufidis, Nikiforiadis, and Chrysostomou et al. [53] have created a tool that predicts the demand of bikes by the customers of a BSO that operates in Thessaloniki, Greece. The tool utilizes ML regression algorithms that use both BSO data and weather data from the past and the present to predict the utilization of the BSO’s network in the future. The tool predicts the demand for bikes up to 3 hours into the future from a certain point in time that the user selects.

The ML algorithms that the tool uses are: XGBoosting, Gradient Boosting, Random Forest, and Neural Network. It is worthy to note here that the tool divides the utilization of the BSO network in bike rentals and bike returns. With XGBoosting performing the best out of the aforementioned algorithms. The authors remarked on the remainder of the algorithms that were used (See Section 3.3 in [53]).

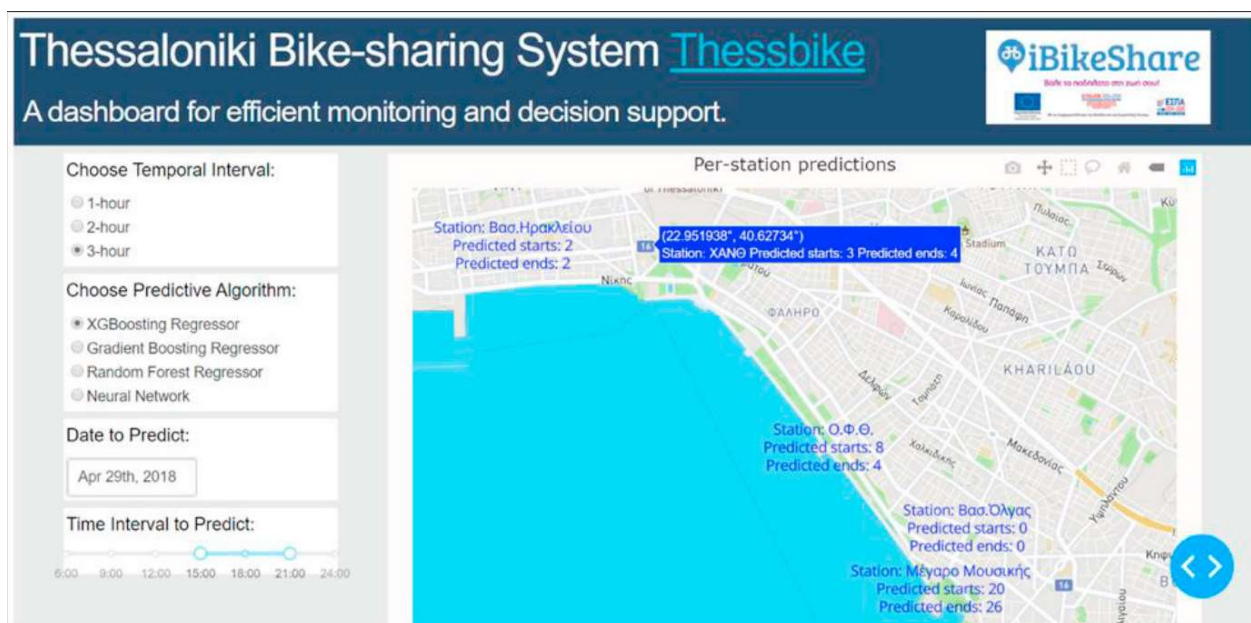


Figure 3: Bike Demand Prediction Tool [53]

Figure 3 presents the interface of the prediction tool after the prediction process concludes. On the left, the user adjusts the temporal interval parameter for the training process, below it, the user chooses which algorithm to be used, and finally, the user selects the date and time for the prediction.

On the right, the user is presented with the prediction results on an interactive map that allows the user to drag, zoom and hover over bike stations - these functions go inline with Shneiderman's mantra (See Section 2.3.1- Bike Sharing Atlas).

As it will be discussed in detail in the next sections, it must be noted that presenting to the user the possibility of choosing an algorithm without additional information can lead to an unpleasant user experience [26], especially to non-expert users. While the authors do not claim that the prediction tool was made for non-expert users, it is a requirement of this thesis to design the interface with such criteria.

Reflecting on this article and its accompanying prediction tool, several points were highlighted relating to aspects of learnability and usability when applied to this thesis' work and its target audience. Arguably, it seems that the tool is aimed at users who are already knowledgeable in ML, the options listed under "Choose Predictive Algorithm" in Figure 3 can be challenging to EUDs without prior ML knowledge, as it assumes that they know *why* they have selected one of the four available algorithms. Also, at a first glance, it is unclear what is required from the user to do in order to produce a result. Furthermore, the tool does not seem to offer the possibility of combining datasets with the available BSS dataset which - potentially – can provide more insight. Addressing these points in the UI designed for this thesis is believed to be vital to the target users and will be addressed in the upcoming chapters.

### **2.3. Shared Mobility Services**

According to Machado, shared mobility services can be defined as "trip alternatives that aim to maximize the utilization of the mobility resources that a society can pragmatically afford, disconnecting their usage from the ownership" [2]. Due to the scarcity of literature that combines SMS with desktop user interfaces, this section will focus on a subset of SMS: BSS.

#### **2.3.1. Bike Sharing Atlas**

In 2018, Oppermann, Möller, and Sedlmair [54] published an article that introduces a system capable of visualizing and analyzing bike-sharing data originating from multiple Bike Sharing Operators (BSO) called "Bike Sharing Atlas". As the tool targets and was also validated by a collection of stakeholders, namely: BSO, urban researchers, urban city planners, policymakers from



municipalities, and the public, it was designed with the challenge of simultaneously including all of them in one application, thus making it a vital article to be reviewed for the research question at hand.

The tool provides a set of functionalities that uses bike-sharing, meteorological, and geographical datasets, which enable the following functions: providing an overview of the stations on the map with occupancy rate details, route planning, presenting bike-sharing activity affected by weather data, and allowing the public to rent bicycles. For new users, the tool presents a set of tooltips and informational elements that guide the user on how to use the application, along with instructional videos.

The authors note that the UI follows “Shneiderman’s venerable information-seeking mantra (“overview first, zoom and filter, details on demand”)” [55], which was also the mantra for the tool that was chosen for this thesis.

As the aforementioned analytical functions are presented, the tool utilizes a set of interactive visualizations that support the cognitive abilities of the user through aspects of “direct manipulation” proposed by Shneiderman [56]. For example, on the “Stations” page, hovering over a station presents a tooltip that communicates the name of the station, the current total number of the available bicycle slots, the current number of available bikes, and a histogram that presents the average utilization rate of the station. This can be seen in Figure 4.

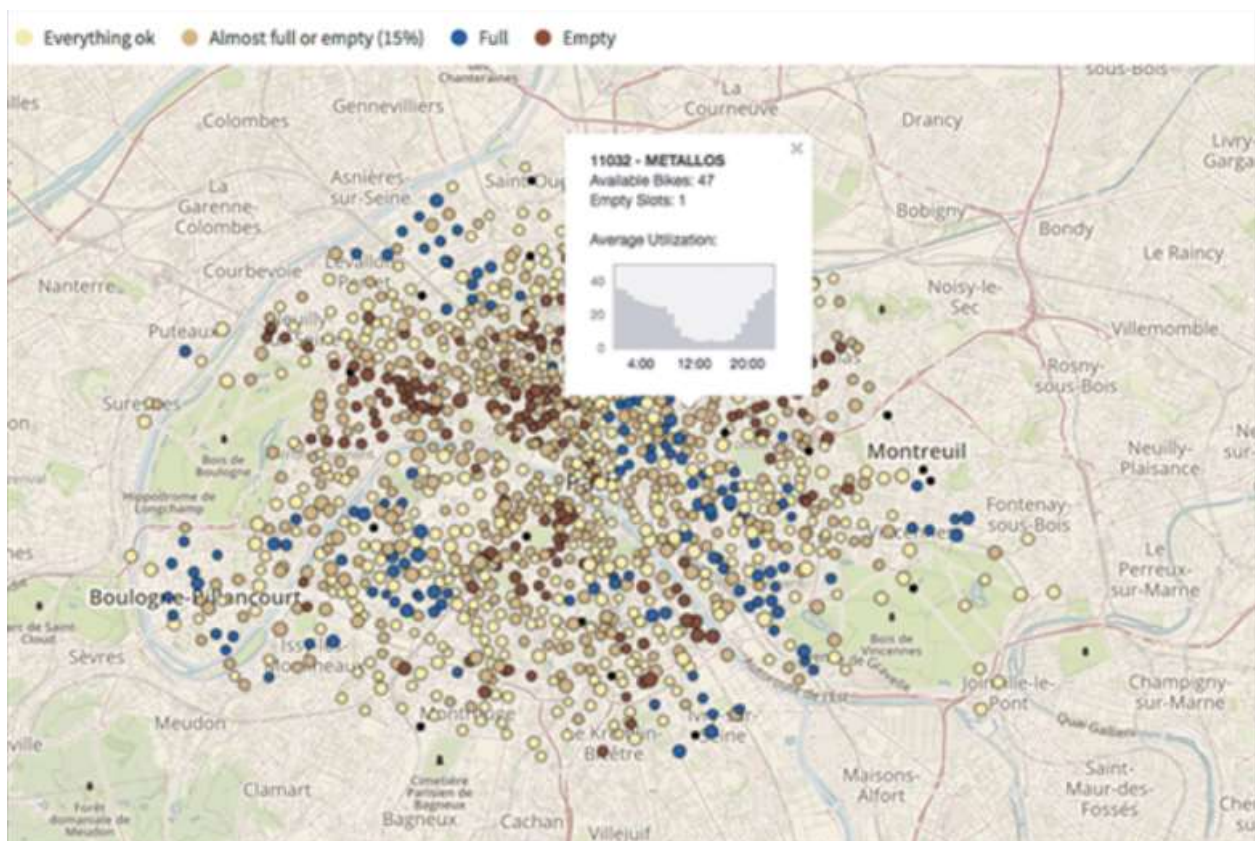


Figure 4: Bike Sharing Atlas - Stations Page



Furthermore, the visualization communicates the number of available bikes in their respective stations through color-coded nodes which are explained in the map legend located on the top-left corner.

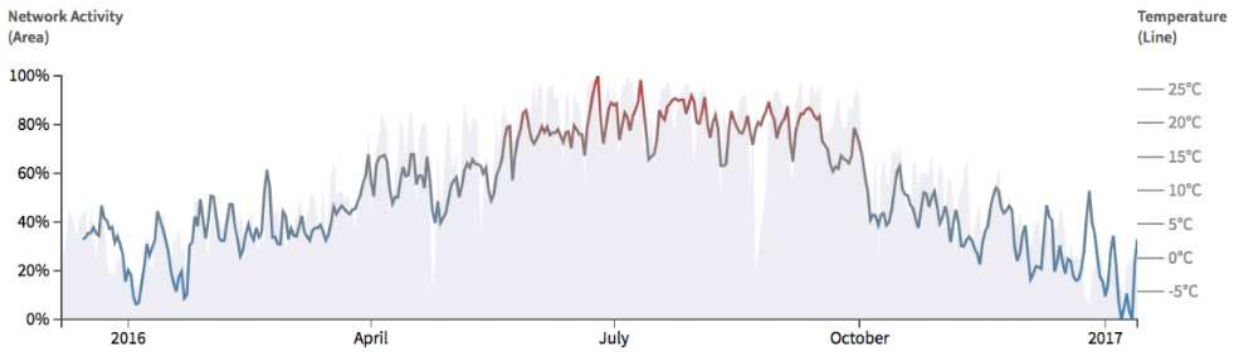


Figure 5: Bike Sharing Atlas - Bike Sharing Network Utilization

Figure 5, presents an interactive chart that shows the utilization of the bike-sharing network of Vienna in 2016, the chart combines both the bike-sharing dataset and Vienna’s weather dataset of the selected period. As noted by the authors, this provides insight into the relation between the utilization rate and the temperature.

As Bike Sharing Atlas provides these functionalities, the system lacks the ability to import additional datasets by the user from the available interface in an easy manner. Furthermore, it does not provide the possibility of employing ML methods to the available datasets through its interface. These capabilities were found to be significant in importance for the research question at hand.

## 2.4. Public Domain

Contrasting scholarly articles with the public domain, a collection of shared mobility web applications employing IML elements for non-expert users were collected. Furthermore, they were found to be providing relevance to the research question at hand. As it will be covered in this section, having data sets available to the public through “Open data” shows that this data can be of value to them through performing studies and analyzing large datasets [57, 58].

### 2.4.1. Visualization.bike

Visualization.bike [59] is a website that provides analysis to its users through a collection of interactive visualizations of bike-sharing datasets. These datasets are obtained from BSOs from several cities, such as New York, Oslo, and Guadalajara. Subsequent to selecting one of the BSOs, the user is redirected to its dedicated page. In this instance, a dock-based BSO called “New York Citibike” was selected.

Starting from the top of the page, the user can select a certain year or a specific month within a year. This causes the page to load the relevant dataset, populate, and visualize the charts with it. Figure 6 presents the charts containing the analysis of the BSO data. Having set the year to 2020 and selected “All months” as the period within the year are displayed in this instance, five charts and two tables containing an analysis of the dataset are presented.



Figure 6: System page of Visualization.bike (1/2)

The first chart provides a visualization of the daily count of trips that occurred from January 2020 until December 2020, toggling from “daily” to “monthly”, presents a line chart that shows the total sum of trip counts that took place each month. The second chart, “Weekday activity”, presents a weekday distribution of bike trips that occurred during the selected timeframe, selecting (average) presents the average count of trips for the selected time frame. Similar to the “Weekday activity” chart, the third chart, “Hourly activity”, presents the data in a similar fashion, however with the x-axis resembling a 24-hour distribution of when the trips have occurred.

It is worthy to note that the “hourly activity” chart does not clearly communicate what classifies as a trip within one of the 24 hours. While it can be assumed that this chart presents the number of trips that started at a certain hour during the day, a different user might interpret the trips as non-unique trips. To clarify, considering a trip that started at 12:55 and concluded at 13:15 on the same day,

would this trip be added to the count of trips that occurred at 12? 13? Or both? On one hand, it can be argued that a more accurate albeit potentially longer chart title can resolve this confusion, on the other, implementing an icon - usually resembling an informational artifact, e.g., an (i) or a (?) - positioned next to the title, can provide an accurate description of what precisely this chart is representing.

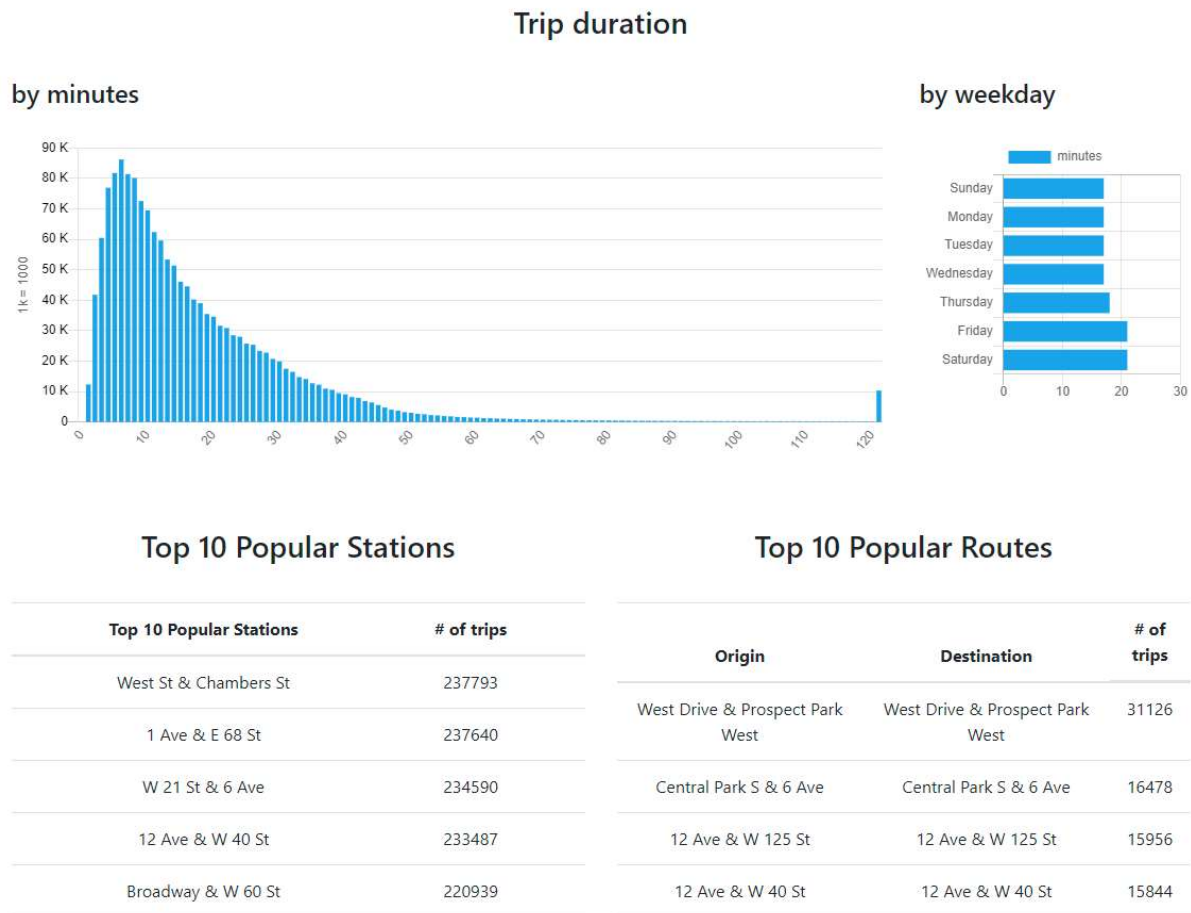


Figure 7: System page of *Visualization.bike* (2/2)

Figure 7 presents a continuation of the page with the remaining 2 charts and 2 tables: “Trip duration by minutes”, “Trip duration by weekday”, “Top 10 Popular Stations”, and “Top 10 Popular Routes”.

The “Trip duration by minutes” chart presents a distribution of the duration of filtered trips with each step on the x-axis resembling a minute, and each step on the y-axis resembling 10,000 trips.

Similar to the previous chart, “Trip duration by weekday” presents a distribution of the filtered trips on a per weekday basis. Hovering over “Sunday” the chart displays “17 minutes”, however it is unclear what this value means. While it can be the average duration of trips, it can also mean the maximum trip duration that has taken place on a Sunday. I argue that this further increases the

necessity of evaluating the charts with multiple end-users formatively and summatively as proposed by [60] along with providing informational tooltips.

The table “Top 10 Popular Stations” presents the count of bike trips that form a connection with a particular bike station, the stations are sorted in relation to the number of trips in a descending manner. Finally, the “Top 10 Popular Routes” table, presents the number of bike trips that originated from a station and concluded in another. Similar to the previous table, this table is sorted in a descending manner as well.

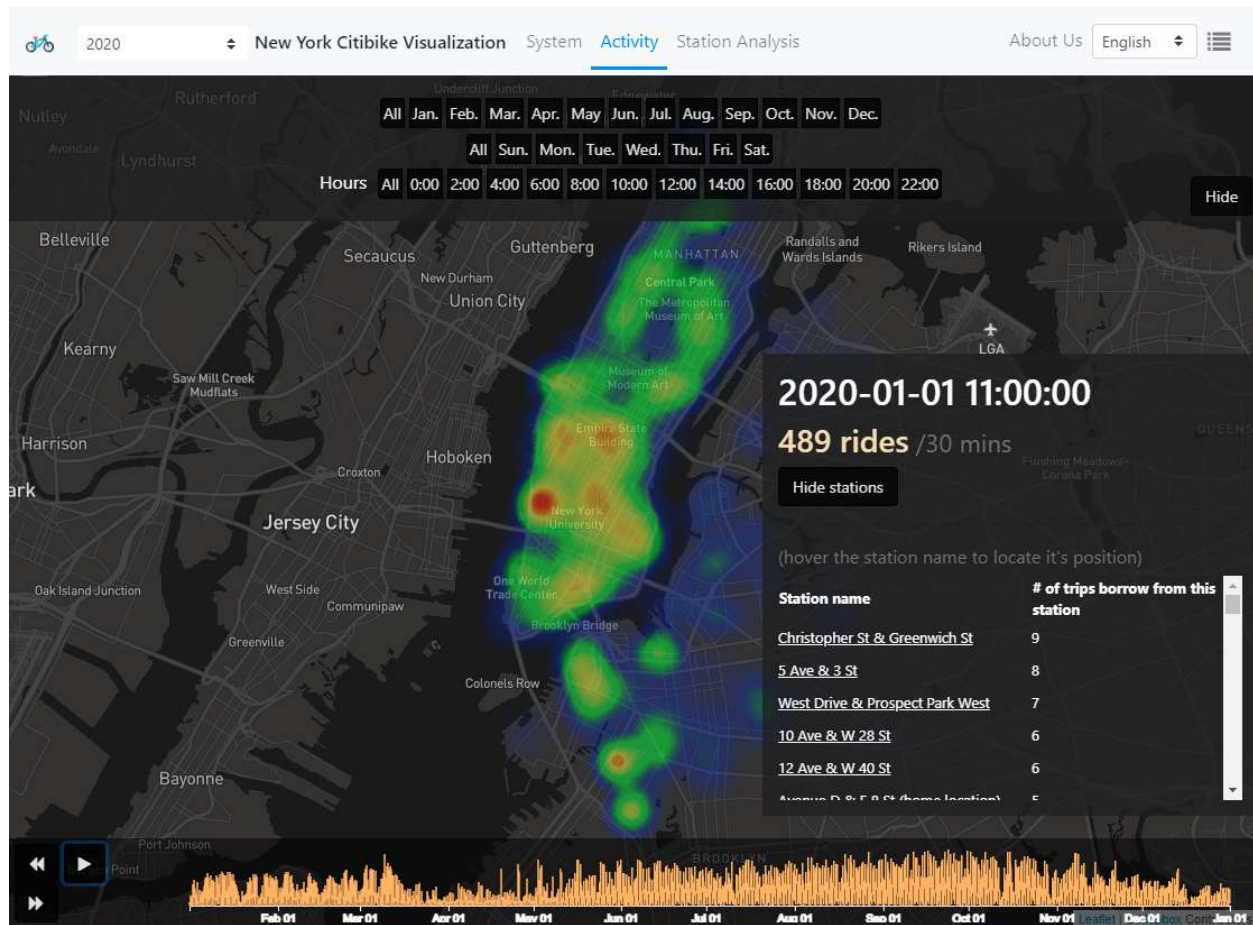


Figure 8: Heatmap Page from Visualizaion.bike

Navigating to the “Activity” page, an interactive spatiotemporal visualization of the number of trips from the included bike stations is presented. The number of trips is visualized in the form of a color-coded heatmap. As shown in Figure 8, the user can temporally constrain the dataset from the top banner, and would receive the response from that action reasonably quickly – (c.f. **Principle 5**). Clicking on the play button, the heatmap animates the usage of bikes from the stations. The table on the right corner updates its values with the heatmap animation, while also providing the functionality of displaying an icon of the station on the map as the user hovers with their pointing device over it.



The timeline on the bottom of the page presents the number of trips across the selected period with a red line that appears when the user wishes to manually seek to a certain point in time.

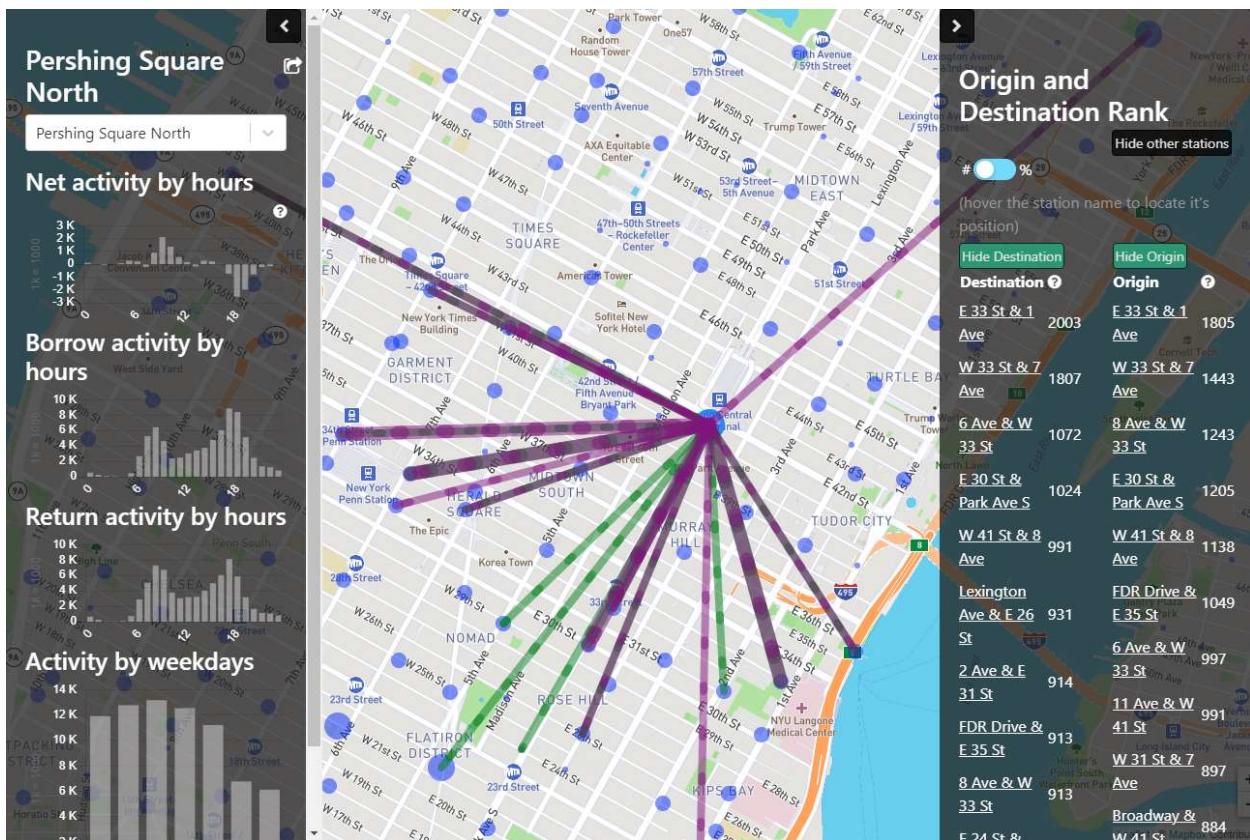


Figure 9: Station Analysis - Visualization.Bike

Figure 9 presents the “Station Analysis” page. Consequent to clicking on a station, the charts on the left display the relevant historical data of the station. Along with the possibility of viewing the exact number of the columns by hovering over them with a pointing device. The panel itself can be minimized by clicking on the less-than (chevron) icon, and the user can manually select the station’s name from the drop-down menu. On the right side, a ranking of bike rentals from the selected station and bike returns from other stations is denoted as the “Destination”, and a ranking of the rented bikes from a station that are returned to the selected station are denoted as the “Origin”. Similar to the left panel, the user can hide the panel by clicking on the greater-than (chevron) icon, users can view the value of the ranking in percentages, and by hovering on any of the stations, the visualization hides all the lines that are connected to the chosen station and only keeps the hovered station.

Reflecting on this website, the inclusion of informational tooltips closely attached to charts and functions that require intricate explanation is paramount. Furthermore, what each chart represents can add towards the overall value of this research. Moreover, the placement of the charts as presented in Figure 9 can be argued to reduce the cognitive effort needed to make a visual relation with the charts and their respective station(s). Also, as presented in Figure 8 temporally manipulating the heatmap

visualization and representing the frequency of the number/amount of an event occurring in a timeline can increase the value of the tool. Fortunately, Kepler.GL is capable of performing this natively with additional degrees of customizability through its interface.

### 2.4.2. Shared Mobility Flows – Berlin

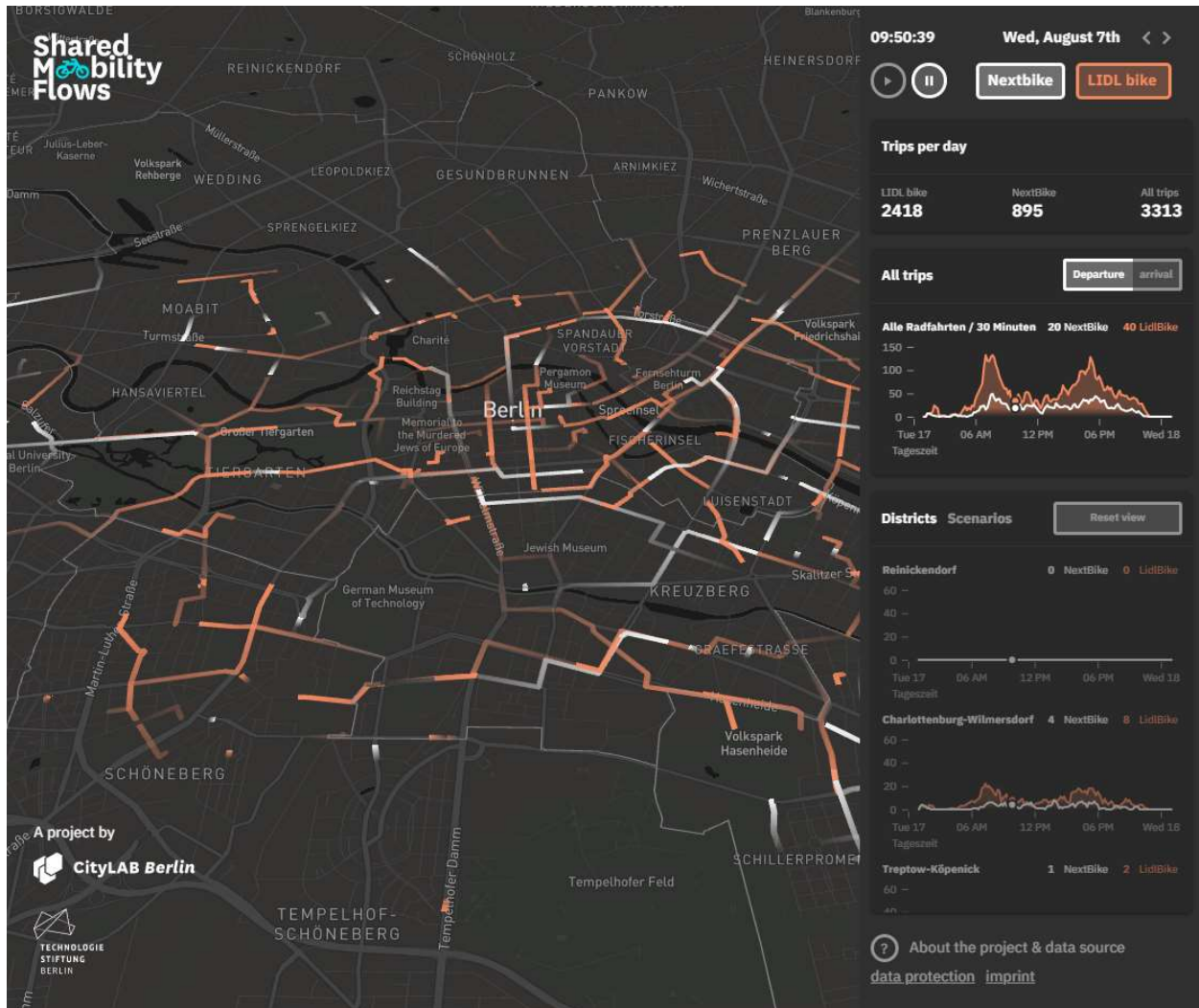


Figure 10: Shared Mobility Flows Prototype

As a response to utilize the Mobility Data Specification (MDS) standard, CityLAB Berlin developed a tool called “Shared Mobility Flows” that provides an analysis and a visualization of the BSOs in Berlin [61]. Upon navigating to the prototype [62], users are met with a dialog box that provides information about the project, along with an animated demonstration of the available features of the tool. Figure 10 presents the tool subsequent to closing the dialog box, users are presented with the map of Berlin, along with color-coded bike trips resembling two BSOs. In addition, they are presented with a panel on the right that allows them to control the visualization along with an interactive representation of the analyzed BSO dataset.

Breaking down the right panel into five sections, the first section of the panel provides controls to start/stop the animation of the trips, along with the possibility of hiding or showing the trips from the available BSOs.

The second section presents to the user the corresponding number of trips that have taken place by both BSOs. Section 3 provides an interactive chart that displays the count of trips that occurred during the selected day, the x-axis represents a 24-hour interval of the day, while the y-axis presents the count of trips, clicking on either line on the chart allows the user to seek to the selected time of the day in relation to the trip animation.

Section 4 represents a chart for each district in Berlin, with properties similar to section 3 with the difference of highlighting the district upon hovering over one of the corresponding charts. Finally, Section 5 contains legal and administrative information about the project, while clicking on the “About the project & data source” displays the aforementioned initial dialog box.

This work perceivably resembles a fixed dataset of 2 BSOs in a city. The work in this thesis aims to provide the EUDs with the possibility of supplying datasets independent from the shared mobility program, its mode of transport, and the city it operates in. While Shared Mobility Flows offers interactive charts and a broad analysis of the BSOs, a detailed analysis of the dataset is not possible unless an EUD individually develops it. This, along with the lack of IML capabilities are features that will be explored in this thesis.

It is worthy to note that the technology that facilitated this work is called Deck.GL [63]. Similar to Kepler.GL, it is a project developed by Uber Technologies [64]. While Deck.GL was considered as a candidate tool, Kepler.GL was chosen due to the relevance of the additional features to the target audience (See 3.2.2).

### **3. Methods and Material**

This chapter explains the methods and materials that were used to guide this thesis. The methods will be explained in the upcoming section, and the materials will be explained in the second section.

#### **3.1. Methods**

As previously discussed, this thesis uses the design science research method to create the designed artifact. In this section, I will be briefly explaining the Design Science Research paradigm and the framework used to evaluate it.

### 3.1.1. Design Science Research

Proposed and commented upon by Hevner et al. [65] and Alan R. Hevner, respectively [25, 65]. Design science research (DSR) is a research paradigm that can be used to carry out a research project in the Information Systems (IS) field [25]. This paradigm uses three cycles: The rigor cycle, the design cycle, and the relevance cycle.

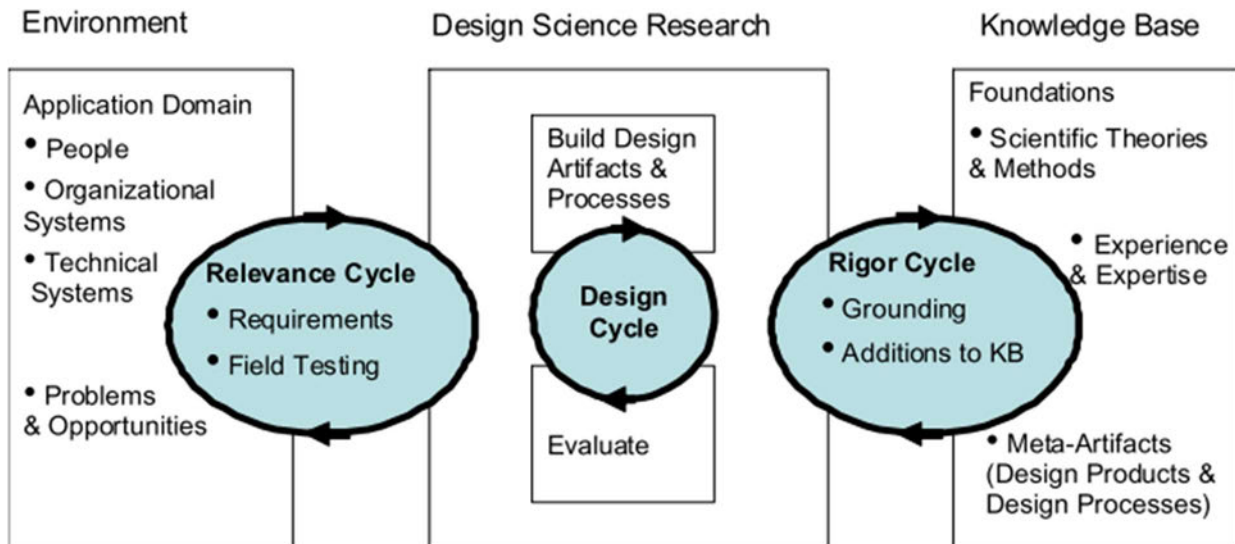


Figure 11: Design Science Research Cycles diagram by Hevner [25]

As illustrated in Figure 11, the relevance cycle is used to extract the information and the requirements from the relevant environment to the research question. The rigour cycle is used to link and to ground the current research question with the body of literature. And finally, the design cycle is used to iterate through the process of building a designed artifact and to subsequently evaluate it.

Both the relevance and the rigor cycle were iterated through in the requirements-oriented related works section. What remains therefore, is to iterate through the design cycle. By creating a designed artifact (the interface) that embodies the requirements in a manner that adheres to the previously discussed design guidelines for IML interfaces (C.f. Section 2.2.1 - A Review of User Interface Design for Interactive Machine Learning [26]), the paradigm necessitates us to evaluate it, for this, the next subsection will discuss the evaluation framework that was used to evaluate it.

### 3.1.2. FEDS: A Framework for Evaluation in Design Science Research

In order to provide a coherent evaluation process for the interface, I will be briefly elaborating on the evaluation framework used. Venable, Pries-Heje, and Baskerville [66] proposed a framework to evaluate artifacts made for Information Technology (IT) as an extension and as further elaboration on the design cycle step proposed in [25] (c.f. Section 3.1.1 - Design Science Research).



The authors propose that in order to perform a FEDS evaluation, 4 steps are to be executed: “1) explicate the goals of the evaluation, 2) choose the evaluation strategy or strategies, 3) determine the properties to evaluate, and 4) design the individual evaluation episode(s)” [66].

Furthermore, the article discusses the design cycle for multiple kinds of approaches, subsequent to analyzing the paper however, it was found that the interface will ultimately be implemented into what FEDS’ preceding paper [67] classifies the intended tool to be a socio-technical artefact (“Socio-technical artifacts are ones with which humans must interact to provide their utility”)[68].

Determining the goals of the evaluation for such a tool, the authors suggest to first ascertain the efficacy of the design (I.E. to determine that the enhancements of the artifact are indeed caused by it [68, 69]). This can be achieved by performing a formative ex-ante artificial evaluation to ensure that the design goes inline with the requirements derived from the literature (c.f. 2 - Related Works). Later on, the authors suggest performing a summative ex-post naturalistic evaluation to determine the effectiveness of the artefact (I.e., to determine the artifact's relevance in its contextualized environment [68, 69]). Thus, a low-fidelity and a medium-fidelity prototype will be developed for the formative ex-ante artificial evaluation and a summative naturalistic evaluation (E.g., a cooperative evaluation [70]). Choosing the strategy of performing an artificial formative evaluation in the early stages and following up with a summative naturalistic evaluation is denoted by the authors as a “Human Risk & Effectiveness Strategy”.

Being informed from the previously performed rigour cycle in the related works section, the extracted requirements (Step 3) can be presented in the form of a low-fidelity prototype and later evaluated in an artificial ex-post manner (discussed in Tool Design, and Evaluation). However, it must be noted - and as suggested by the authors - that performing a proper “Human Risk & Effectiveness” evaluation strategy is still necessary however it will not be covered in this thesis.

Corresponding the aforementioned four steps to this work would be:

- 1) Goals: Determine the efficacy and the effectiveness of the designed artifact.
- 2) Strategy: Human Risk & Effectiveness
- 3) Properties: The extracted requirements from the rigour cycle.
- 4) Design Episodes
  - a) Formative Ex-Ante Artificial Evaluation; Low-fidelity prototype, and a medium-fidelity prototype.
  - b) Summative Ex-Post Naturalistic Evaluation: Performed with end-users from the field which is out of scope for this thesis.

In conclusion, while this framework provides to the researcher a collection of evaluation strategies to use based on the desired artifact, their context of use, and their suggested evaluation methods, it does not cover how to perform these methods in detail. The decision of which methods to use is perceivably left for the researcher to decide.

Fortunately, the work of Sonnenberg and Vom Brocke identifies several patterns of the evaluation strategies used in DSR [71]. Among them, a pattern called “The prototyping pattern” has been used by several authors which evaluates the artefact by creating a prototype to demonstrate its suitability and feasibility in practice, thus making it “an adequate evaluation method for DSR artefacts” [71]. As the authors describe the pattern, they highlight that subsequent to creating the prototype, test cases or analytical examples could be used to determine whether the prototype allows its users to complete tasks. By conducting either of these evaluations - which can be perceived as an ex-post naturalistic evaluation. Its usefulness can be determined by “real users” [71] prior to its implementation in organizations. However, due to time constraints imposed on the author, the inclusion of “real users” in this thesis was not possible. Therefore, an evaluation of the resulting prototype [cf. Chapter 4 - Tool Design] could only be conducted independently by the author. The method was used will be discussed in the next subsection.

### **3.1.3. Streamlined Cognitive Walkthrough**

As noted above, the inclusion of real users to interact with the resulting prototype will not be possible. However, there exists a usability inspection method named Streamlined Cognitive Walkthrough (SCW) [27] which is a variation of the cognitive walkthrough (CW) usability inspection method that could be used within this context [72]. A CW can be defined as “a usability inspection method that links the interface walkthrough to a cognitive model. The evaluator uses the interface to perform tasks that a typical interface user will need to accomplish. The actions and responses of the interface are evaluated according to the user’s goals and knowledge through responses to questions related to the method’s cognitive model, the differences between the user’s expectations and the use reality (i.e., the steps really required by the interface).” [73]. As the definition states, an evaluator uses the interface, which can be the designer assuming the role of the user in this case [72], defines and analyses the tasks, i.e., the aforementioned task scenarios discussed by [71], and would later execute and evaluate them to determine the completability of the tasks. Next, the designer finds and amends usability issues before the prototype is presented to real users [71].

In order to perform a SCW, the author [27] presents the following overview:

- 1) Define inputs to the walkthrough
  - a. Identification of users
  - b. Sample tasks for evaluation
  - c. Action sequences for completing the tasks
  - d. Description or implementation of interface
- 2) Convene the walkthrough
  - a. Describe the goals of the walkthrough.
  - b. Describe what will be done during the CW
  - c. Describe what will not be done during the walkthrough
  - d. Explicitly defuse defensiveness
  - e. Post ground rules in a visible place
  - f. Assign roles
  - g. Appeal for submission to leadership
- 3) Walkthrough the action sequences for each task
  - a. Tell a credible story for these two questions:
    - i. Will the user know what do to at this step?
    - ii. If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?
  - b. Maintain control of the CW, enforce the ground rules
- 4) Record critical information
  - a. Possible learnability problems
  - b. Design ideas
  - c. Design gaps
  - d. Problems in the Task Analysis
- 5) Revise the interface to fix the problems

It shall be noted here that the items 2-d,2-e,2-f, 2-g, and 3-b will not be used as they involve participants. Furthermore, a detailed explanation of the items can be reviewed in [27, 72].

## 3.2. Material

This section covers the materials used to present the interface. First, I will be presenting the dashboard. Second, the geo-spatial analysis tool. Third, the charting framework, and finally, the dataset.

### 3.2.1. Opendash

According to its respective field, dashboards can be defined as “a visual and interactive performance management tool that displays on a single screen the most important information needed to achieve

one or several individual and/or organizational goals, allowing the user to identify, explore, and communicate problem areas that need corrective action” [74]. Indeed in our case, having such a technology to provide the functions in the aforementioned definition would be useful for such complex datasets and potential functions that are going to be executed on it. Therefore, among the various dashboard solutions that were available, it was decided that the dashboard solution Opendash from Open.inc [29], Opendash would be a suitable option as it provides the needed requirements for this research.

Opendash is an open-source dashboard system that provides the ability to visualize its stored data [29]. It allows its users to collect data from multiple sources, filter it, perform processing on it, and to visualize it in the form of widgets. Furthermore, it is designed to be customizable, extendable, responsive, device-agnostic, and deployable on the cloud or on local environments. Moreover, it offers to its users the functionality of including media from various sources and to display different data types. Having these features offered by Opendash, it can be perceived how it matches with the requirements highlighted in the previously discussed related works section (c.f. 2 - Related Works).

Also, Opendash grants the possibility of persisting the workspace settings performed by the user to setup the dashboard environment. This is done to eliminate the need of setting up the dashboard environment, its widgets, and the parameters of each widget whenever the user wishes to perform work on it (c.f. Section 2.1.3 - Cuscus: An End User Programming Tool for Data Visualisation).

### 3.2.2. Kepler.GL

Kepler.GL is an open-source geospatial analysis tool from Uber Technologies, Inc [64]. It was designed to be used on data sets of large-scale properties [28]. Furthermore, it was built using Uber Technologies’ framework Deck.gl [63]. Deck.gl is a visualization framework built for large-scale datasets [63].

It provides a significant number of functionalities that would be of sufficient adequacy for the aforementioned target audience of this research and the highlighted requirements.

Similar to Opendash and Cuscus. Kepler.GL provides the ability to save workspace settings of each visualized dataset (called Data Layers) by exporting the configuration to a JSON object that can be imported into Kepler.GL via the “Load Map Using URL” tab.

Further investigating the importability aspect of Kepler.gl, it is to the author’s best knowledge that datasets can be imported into it via 4 methods: 1) Drag-and-Drop, users can import datasets into Kepler.GL by dragging and dropping a dataset from the user’s machine onto the interface, as noted by [33], this reduces the effort that the user has to spend to visualize a dataset (c.f. Section 2.1.2 –

End-user Development of Information Visualization). 2) Providing a URL for the dataset from a web address. 3) Cloud storage, users can import data sets from cloud storage, such as, Dropbox. 4) On instantiation, Kepler.GL also allows for the possibility to include a dataset through code before it runs, this eliminates the need of users having to manually load the desired dataset every time they wish to interact with it. 5) Post instantiation, as the users are interacting with Kepler.GL, it also provides the possibility of importing other datasets either by code (if previously setup, of course) or through the drag-and-drop functionality. It is worthy to note here that Kepler.GL can accept datasets that are larger than 250MBs in size [75] using GEOJSON, CSV, and JSON formats, however, they must be imported via a URL. Having this range of methods not only increases the accessibility of end-user developers [33], it can also be argued that it is not necessary for end-user-developers to endure the development time needed to implement these datasets into this tool, instead, they are enabled to manage datasets without the need for external intervention, this can be perceived to further promote rapid interactions with the tool (c.f. **Principle 4**).

Similar to Many Eyes (c.f. Section 2.1.1 - Many Eyes: A Site for Visualization at Internet Scale), Kepler.GL has a functionality embedded in it that can guess the data types of the features in the datasets. This, as previously discussed, reduces the data preparation efforts needed from the end-user (c.f. 2.1.1 - Many Eyes: A Site for Visualization at Internet Scale).

Under the hood and as its website claims, Kepler.GL is designed to handle “large scale datasets” [28], it can perform visualizations of datasets containing hundreds of thousands of records relatively quickly as it utilizes the end-user’s graphics card [76], and it allows for relatively smoother animations as it supports high refresh rates on monitors.

It increases the directness of interaction, increases the visibility of information, and minimizes the data ‘wrangling’ step to format the data in the right form for visualizations (c.f. 2.1.3 - Cuscus: An End User Programming Tool for Data Visualisation). Furthermore, it natively provides support to the aforementioned mantra of Schneiderman [33]( c.f. Section 2.3.1- Bike Sharing Atlas).

In addition, it allows users to perform aggregation, display tooltips, provide native support for trips and to apply filters in real-time [28]. Furthermore, it is built on React [77] and Redux [78], this can allow for a relatively straightforward implementation and integration with the dataset provided from Opendash’s parsing server. Moreover, it provides the ability to customize its interface as a means to include functions that are not included in it [79]. This allows for fewer widgets to be developed and subjectively, it provides an aesthetically pleasing design.

It should be noted here that through the mentioned customizability feature offered by Kepler.GL, it has made both the previous work in the Master's project and the work on this thesis possible. By being able to customize its interface, IML features can be offered as extended components, and extra charting capabilities could also be added. In my previous Master's project work, Kepler.GL was customized to present graphs that chart the dataset that was already being visualized on the map. Figure 12 presents a screenshot showing the finalized work on the Master's Project.

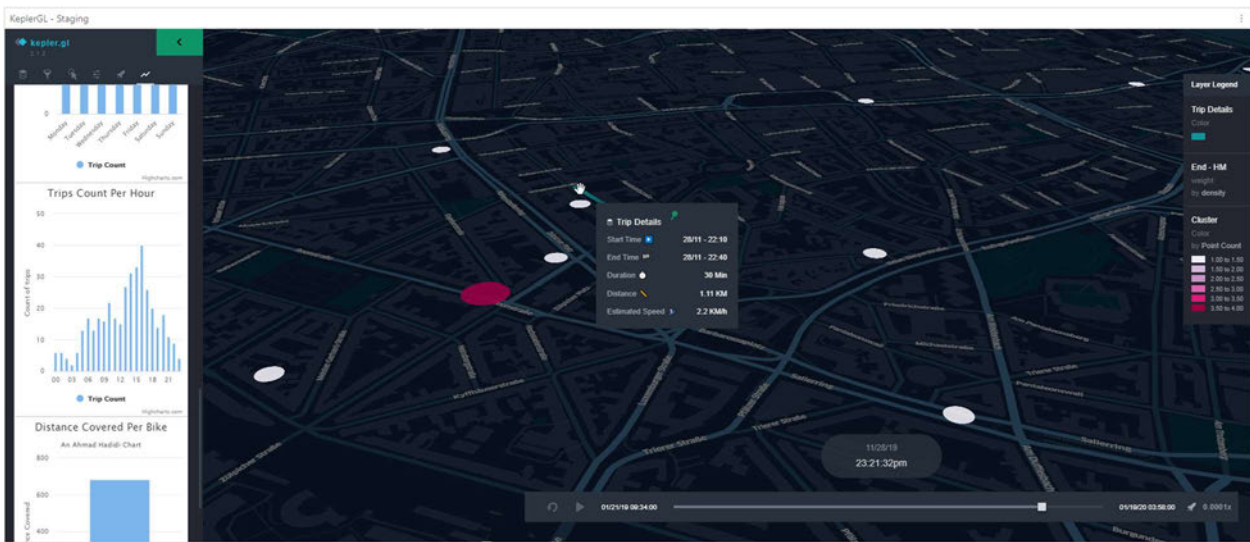


Figure 12: Master's Project Finalized Work

Explaining the figure, on the left, the customized sidebar is presented. It consists of six tabs (presented as icons): 1) Layers, it allows for the addition and configuration of various kinds of layers to the map using the loaded datasets. 2) Filters, it allows for the addition of filters and constraints on the loaded datasets. 3) Interactions, it provides a collection of functions to be performed on the datasets, such as, selecting which fields to display when a data point is hovered, a brushing tool to highlight areas of interest, and to display the coordinates underneath the mouse pointer. 4) Basemap, allows for the setting of custom map styling, selecting preset themes, and to display or hide map layers. The layers include: Road names, Borders, 2D Buildings, and 3D Buildings. 5) Rocket is a customized tab added by the author based on the example provided by the developers of Kepler.gl, it contains preliminary text that could be quickly traced in the source code of the widget [80]. 6) Charts, a custom tab that was used to extend Kepler.GL to present charts that will be discussed in the next subsection.

On the bottom, a rounded rectangle containing time-related data for the reached point in the animation is presented, along with a set of trip animation controls underneath it. It provides the ability to start and stop the animation, a slider to indicate the duration of the animation, and to seek through it, along with a setting that controls the playback speed of the animation.

On the top right corner, a legend for the metrics is presented, that contains relevant information for each layer that is currently being displayed on the map.

Finally, the result of the previously mentioned settings, configurations, and functions is reflected on the map presented in the middle, the hovered turquoise line represents a trip that is currently being animated along with a tooltip showing its related metrics. The white and red circles represent a parametrically controlled cluster layer that is - for demonstration purposes - being affected by the amount of the starting points of trips. The gray irregular shapes surrounded by the streets are buildings presented in 2D.

Contrasting the technical features that Kepler.GL offers with the visualization features, it offers to its users the following visualization methods: 1) Points 2) Arcs 3) Lines 4) Grids 5) Hexbin 6) Polygon 7) Clusters 8) Icons 9) Heatmaps 10) H3 11) 3D 12) Trips, and 13) S2. As there are several methods of visualizing datasets, I will briefly explain 4 visualization methods that are believed to yield benefit to the research question.

### **Points**

By supplying a data set that contains longitude and latitude, the points layer can be used to represent them on the map, for example, Figure 13 presents the taxi trips in New York city, this data layer can be used to visualize the number of trips starting on the map, this method also allows users to adjust the radius of the point by a factor to increase its visibility. Furthermore, if an additional feature in the dataset exists, the fill color scale of the point can represent a quantized/quantiled feature from the dataset as well. Moreover, it optionally accepts an altitude for the point, and the stroke of the point can also be dynamically affected to include other features.

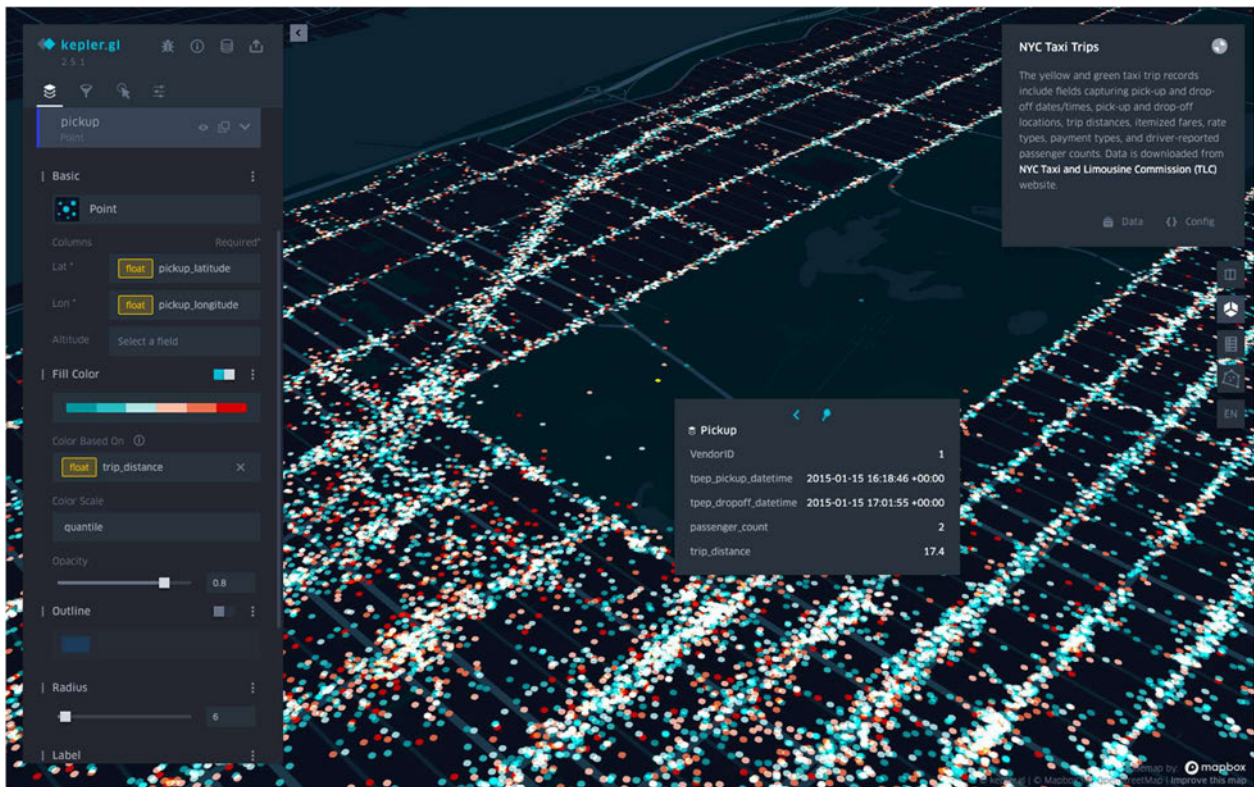


Figure 13: Points Datalayer

## Clusters

Interacting with the previous dataset, a cluster layer can be created to represent the number of trips finishing at a certain location in NYC. Similar to the configuration that can be performed on the previous points layer, it can be performed on this layer as well.

## Trips

By applying the trips layer, the user can visualize the trips from the imported dataset on the map. In Figure 12 a trip is represented by the hovered turquoise line that is currently being animated along with its accompanying tooltip showing its related metrics. The white and red circles represent a parametrically controlled cluster layer that is - for demonstration purposes - being affected by the amount of the starting points of trips. The irregular gray shapes surrounded by the streets are buildings presented in 2D, which can also be set and presented in 3D from the aforementioned base map tab.



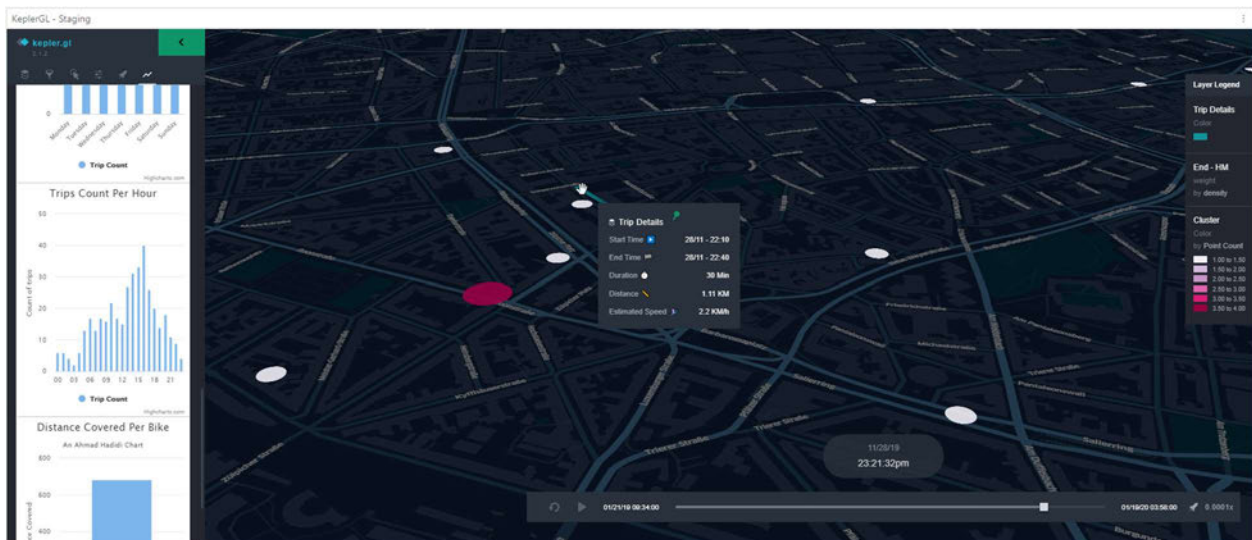


Figure 14: The Trips and The Clusters Datalayers

## Heatmaps

As presented in Figure 15, Kepler.GL offers heatmap data layers that can be added, similar to the previous visualization methods mentioned, the color of the heatmap can be configured, along with the heatmap's weight that can be affected by a feature in the dataset.



Figure 15: Heatmap Datalayer

By having these visualization methods available at hand, it can once more be argued that this reduces the amount of development time and reduces the potential time spent waiting for the implementation of such methods using other solutions, thus making this tool a further increase on the front of the accessibility aspect for end-user developers.

### 3.2.3. Highcharts

Highcharts is a charting framework provided by Highsoft [30]. It is the main charting framework that is included in Opendash and it allows for various charting methods, such as, bar charts, stacked columns, area charts, and pie charts [30]. Furthermore, it can be perceived that it allows for a faster population of charts as it relies on receiving the data for the chart in the form of JSON objects. For this project however, the basic bar chart was the only charting method used, as it was found to be sufficient for the needed information to be visualized.

A total of 5 charts were implemented as a result of individually performed brainstorming sessions, as it can be perceived they can provide insight for the parties of interest. The first chart presents the total number of trips that were performed on the selected bikes. The second chart presents the number of trips that were performed on a certain bike.

The third chart presents the number of trips that were performed on each of the selected bikes. The fourth chart presents a distribution of the number of trips the selected bikes have performed on a monthly basis. The fifth chart presents a distribution of the number of trips the selected bikes have performed at weekly intervals.

The sixth chart presents a distribution for the number of trips the selected bikes have performed at daily intervals. The seventh chart presents the total sum of the distance covered in Kilometers for each selected bike.

### 3.2.4. Dataset

In reference to the stored data that Opendash is capable of exposing for further processing, the dataset that was used represents bike trips that occurred in July 2019 from a BSO operating in Cologne, Germany. This dataset consists of 8 features:

1) ID, represents the id of the trip, 2) bike, represents the bike identifier from the BSO, 3) start\_datetime, the timestamp of when the trip had started, 4) start\_lat, the start latitude of the trip, 5) start\_lng, the start longitude of the trip, 6) end\_lat, the end latitude of the trip, 7) end\_lng, the end longitude of the trip, 8) end\_datetime, the timestamp of when the trip has ended.

Having these dataset features at hand, additional features could be calculated to unravel further information. For example, the duration can be calculated from the start and end time of the trip and an approximate distance of each trip can be calculated by plugging in the start and end points into the Euclidean distance formula [81]. Consequent to calculating both of these values, the approximate average speed could be calculated for each trip, thus making the tool generally more useful. These values can be seen upon hovering over a trip animating over the map as presented in Figure 12.

## 4. Tool Design

Creating a designed artifact from the rigour cycle led to the development of two prototypes. In this chapter, both the low-fidelity and the medium-fidelity prototypes along with a detailed discussion of low-fidelity prototype variants will be presented. As mentioned previously in Kepler.GL [c.f. Kepler.GL], components can be added as extra tabs.

### 4.1. Low-Fidelity Prototype

Referencing the steps discussed by [26] in the related works [c.f. 2.2.1 - A Review of User Interface Design for Interactive Machine Learning], this section will present a set of designed low-fidelity prototype variants of each step in the IML process.

#### 4.1.1. Feature Selection

As the user proceeds with the IML workflow to interact with the interface, [26] states that the first step would be to perform a selection of features. As previously discussed, the user in this step selects the features from the available dataset. In order for the user to do that, the following variants were designed.

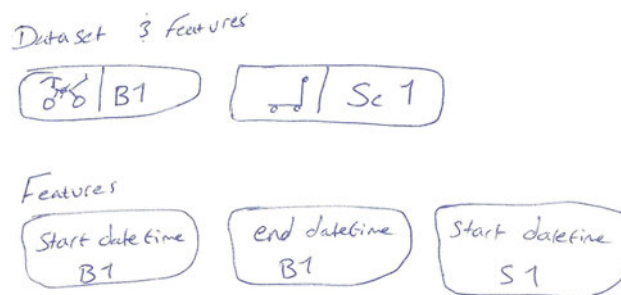


Figure 16: Feature Selection - Variant 1

Excerpted from the full interface, Figure 16 presents the first variant of selecting the dataset and its available features. The variant is split into two sections: First, the user is presented with the available dataset, denoted with the name, and an icon resembling which kind of dataset it is. As the user selects one of the datasets, the second section “Features” populates with the available features from that.

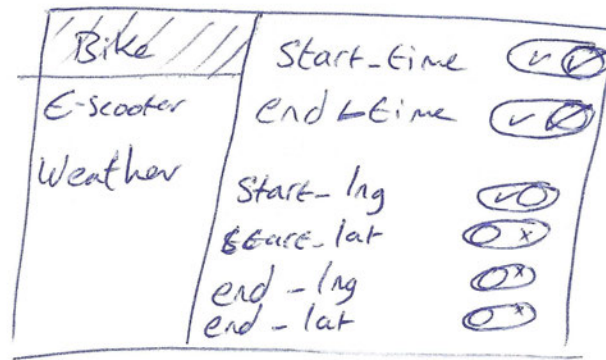


Figure 17: Feature Selection - Variant 2

Figure 17 presents a variant that provides a visual hierarchy similar to folder explorers in operating systems. On the left, the user selects the dataset, subsequent to that, the right panel gets populated with its selected features. The user here can select certain features by toggling features on or off. When the toggle resides to the right, a tick mark appears as a feedback mechanism to signify that the feature has been selected. While this variant provides a common method to select features, the visibility and readability of this element would produce undesired results when it gets implemented into the side panel of Kepler.GL. Furthermore, it can be perceived that toggling features from each dataset would require an extra step of browsing to the desired dataset. Moreover, the labeling of the features cannot exceed a certain number of characters which in this variant is the lowest number of characters.

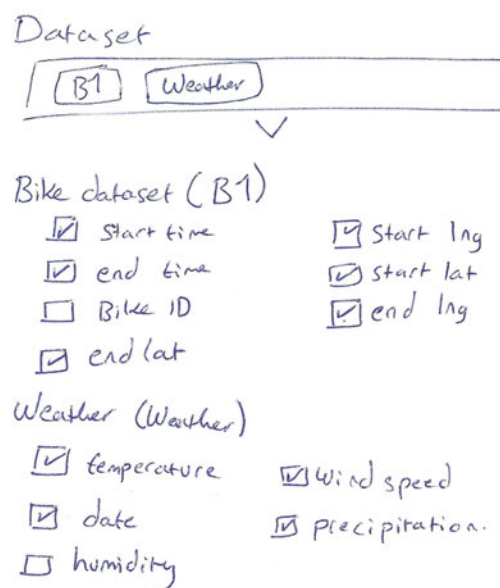


Figure 18: Feature Selection - Variant 3

In this variant, the datasets are presented into a rectangle which can be selected by clicking on the desired dataset. Consequently, the name of the dataset and its type would dynamically be displayed below the rectangle, along with a collection of checkboxes that has the feature names placed next to them. The user selects the features by checking the desired features. The selection rectangle for the datasets is required to be carefully designed to clearly indicate that the dataset has been selected by the user. Furthermore, as the features are displayed in 2 columns, the challenge of being limited with the number of characters for each feature still persists. Apart from these two potential challenges, this variant can be considered for the end design, with some minor modifications nonetheless.

#### **4.1.2. Model Selection**

As the users proceed to the second task in the workflow. They proceed to select a model from the ones available. Designing under the assumption that the target audience of this tool are not thoroughly informed to distinguish between the available ML models - after all, the scholars of the EUD field find that end-user developers can have a naïve understanding of the ML model [26], consider it a black box [24, 82], and are not familiar of the pros and cons of each ML model - this challenge needs to be taken into account. However, as Dudley and Kristensson find, the EUDs prefer to use fast and explorative models first and then to use other models with higher inductive capabilities later [26]. Therefore, the designed variants take into account these findings by providing hints to the EUDs regarding the available ML models by categorizing and sorting them from fastest to slowest. Although, a potential pitfall must be identified here: favoring a model against other models based on its speed does not necessarily mean that it is generally superior, it only means that it is the fastest, but not necessarily the best all-around. The design has to communicate this trade-off between the speed of the model and its inductive capabilities. This means that it should hold the designer accountable to fairly represent the available ML models in a manner that lets the EUDs to ultimately select a model based on a deliberately active weighing process that the aforementioned trade-off presents. Based on the findings, and on this challenge, the following variants are proposed.

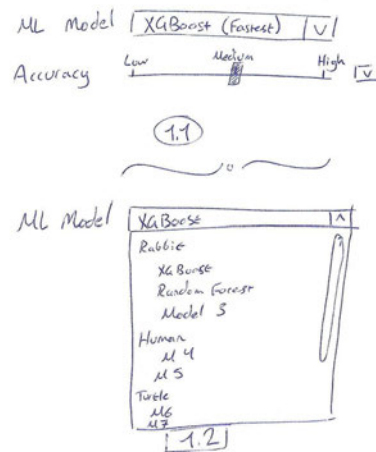


Figure 19: Model Selection - Variant 1

Variant 1 (Figure 19) presents the initial state of the subpart of the interface for this step, as noted under the variant with 1.1, the user is presented with a dropdown menu that allows them to select the model. The relative speed of the model is denoted next to it in the parentheses. Underneath the menu, the accuracy of the model is presented as a slider with three notches: low, medium, and high. Along with the possibility of clicking on the downwards facing chevron to expand the options of the accuracy, this can be generally perceived as the ‘advanced options’ of the presented controls, this functionality will be discussed in detail in the next task [c.f. 4.1.3 - Model Steering]. Above the (1.2) written in Figure 19, clicking on the drop-down menu presents a list categorized by the speed of the models in 3 main categories: a hare, a human, and a tortoise. Within each category, the models are sorted to indicate to the user that selecting a model from the hare category would yield faster results than a model from the tortoise category.

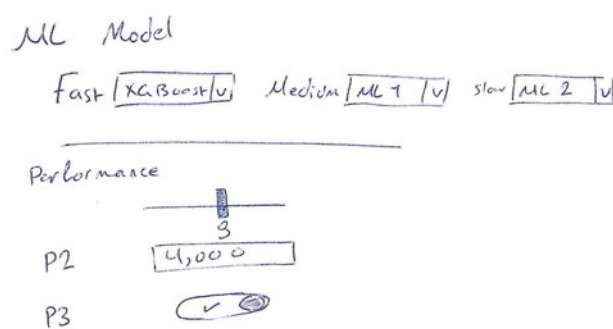


Figure 20: Model Selection - Variant 2

Variant 2 (Figure 20) presents three labeled drop-down menus denoted with the relative speed of the model, along with a control mechanism to configure the selected model. As the user selects one of the models, the other two menus are greyed as a feedback mechanism to indicate to the user that the



model from the chosen drop-down menu has indeed been selected. While this variant can be perceived to communicate relative speed, its visibility might be compromised when viewed on smaller displays.

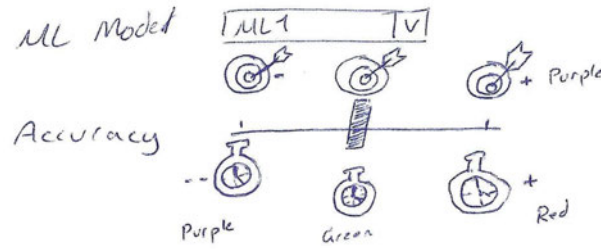


Figure 21: Model Selection - Variant 3

Figure 21 presents the third variant where the EUD can select the ML model from the drop-down menu and can consequently adjust the accuracy of the model via the accompanying slider. The effects occurring on the accuracy and the time required are communicated via colors and the plus and minus signs. In this variant, the red color means that the effect is negatively impacted, the green means that it is less impacted, and purple means that it has the least impact, or in other words, the best case from the time-required standpoint or the accuracies. Furthermore, it is believed that the meaning of the colors is better interpreted due to the inclusion of the plus and minus signs on the icons.

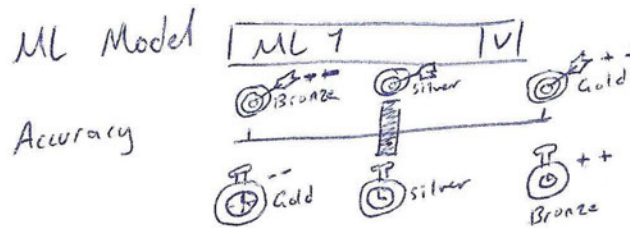


Figure 22: Model Selection - Variant 4

Figure 22 presents the fourth variant of this step, the drop-down menu and the accuracy slider are both similar to variant 3 with the difference of changing the colors of icons from red, green, and purple to bronze, silver, and gold.

#### 4.1.3. Model Steering

Revisiting the accuracy controls section from the previous variants to configure the model are denoted as the model steering task by [26]. As they discuss, users are expected to spend the majority of their time in this task as it is the core activity in the IML process [26]. Also, it is found here that they will select quick models rather than models with higher inductivity. Due to the complementary nature of these two steps in this interface, the drop-down menu presents the models sorted from fastest to slowest and the adjustment controls of the model itself are presented underneath it with preset values that support this preference.

Accuracy

Custom ^

P1 

X

2

P2 4,500 X

P3 32

Duration

From 1/1/2020-00:00 V

To 1/1/2021-23:59 V

Figure 23: Model Steering - Variant 1

Examining the extension of Figure 23, in Figure 19, clicking on the ‘advanced options’ (which is now an upwards facing chevron) presents parameters that were preset by the developers of the tool. Exposing these preset parameters provides to the EUD the possibility to experiment and to train the learned concept with their preferred parameter configurations, and as a feedback mechanism, the initial notches displayed in 1.1 in Figure 19 are replaced with the word ‘Custom’, this is to denote that the preset has been modified by the EUD, also, whenever the EUD changes a parameter, an icon appears next to revert the change [83]. Underneath the parameters, the duration (specified in date and time) is provided for the EUD to include the time span in which the selected features occurred in.

It shall be noted here that while adjusting parameters can further enhance the understanding of the EUD on the effect of each parameter to the learned concept, this variant does not provide the means to explain to the EUD the exact function of each parameter and their effect on the learned concept. Furthermore, it does not clearly indicate to the user that adjusting the accuracy of the model can affect the time required to finish the training process. Moreover, including the possibility in the duration section to select certain time spans during the day across the selected duration can be useful to the EUD. For example, an EUD can engage in the task of creating a learned concept that predicts a certain behavior during the year 2020 on select dataset features that have only occurred between 08:00 in the morning until 19:00 in the evening in the year 2019. It can be perceived that this case can happen, therefore, the medium-fidelity prototype will address it.



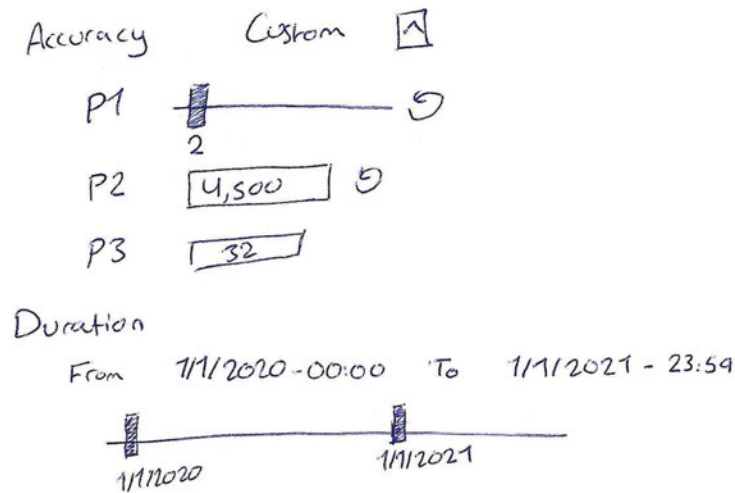


Figure 24: Model Steering - Variant 2

Similar to the previous variant, the accuracy controls are displayed after the EUD clicks on the chevron. In this variant, after the EUD edits a parameter, a circular arrow appears next to it to revert it to its previous value. The duration control has undergone the most change, where the EUD can click and drag on either of the two notches to specify the duration that the model has to be constrained in. Furthermore, the 'From' and 'To' elements provide functionalities for a perceivably more accurate method to set the duration.

#### 4.1.4. Model Steering Part 2

As a continuation to the model steering task, the user selects the goal of the task that is being carried out, as this thesis mainly focuses on bike-sharing systems, a variant was created to predict common challenges faced in their field [53, 84–87].

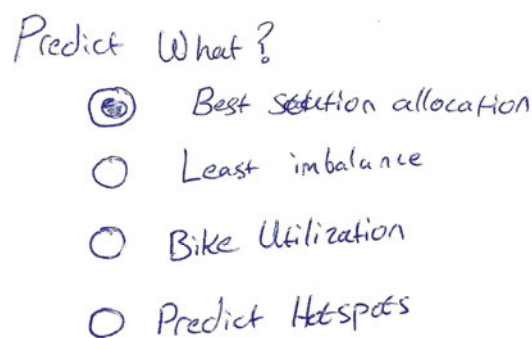


Figure 25: Model Steering Part 2 - Variant 1

In Figure 25 the user is presented with four radio buttons to choose what the learned concept shall predict. Based on the revised literature, the concept predicts 1) Best station allocation, where the model predicts the optimal placement of bikes or bike stations. 2) Least imbalance, where the model predicts the optimal location of bikes or bike stations where the least imbalance occurs. 3) Bike

utilization, the model provides a forecast on predicted bike utilization. 4) Predict hotspots, where the model provides a prediction on the locations where the most number of activities will occur in the city during certain times of the day, based on the historical placement of bikes across the selected timespan.

#### 4.1.5. Quality Assessment

As the user completes the previous step of training the model, a feedback mechanism is required to present to the user the current accuracy of the model [26, 51]. The authors find that this step can be helpful to the users as it provides the necessary information needed to stop or to continue the training process.

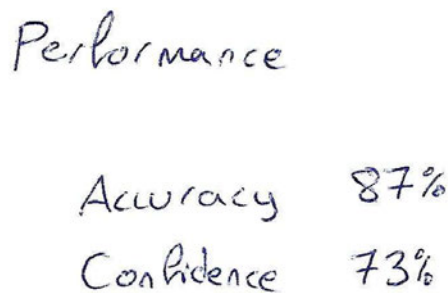


Figure 26: Quality Assessment - Variant 1

Figure 26 presents the first variant of where the user can be presented with the current accuracy of the model. This is communicated by showing the label 'Accuracy' and its value next to it.

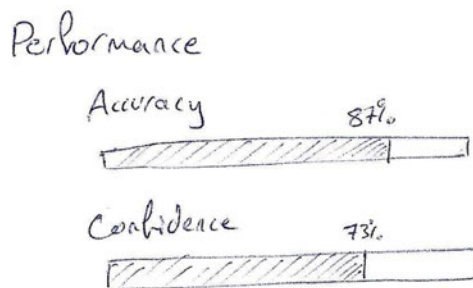


Figure 27: Quality Assessment - Variant 2

Similar to the previous figure, Figure 27 presents a progress bar element that is filled based on the accuracy value of the model. This variant can be used to warn the user about overfitting the model (to be discussed in the next step) by providing a line drawn on the accuracy progress bar that upon hovering over warns the user about that effect. This variant is found to be more beneficial as it can communicate the overfitting phenomena effectively.

#### 4.1.6. Termination Assessment

As users have been informed about the current accuracy of the learned concept in the previous step, the user is now presented with the choice of concluding the training process or to continue with steering the learned concept further. However, [26, 51] highlight that the model can overfit as the user keeps training the model. Therefore, the designed variant in this step needs to advise the user to abstain from training the learned concept once it reaches this “inflection point” [26].

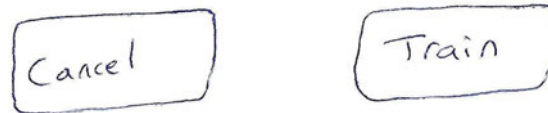


Figure 28: Termination Assessment – Variant 1 - Below Inflection Point

Figure 28 presents the termination assessment step by presenting two buttons: the ‘cancel’ button to provide control over the system [83] and to stop the training process, and a ‘Train’ button to steer the learned concept further.

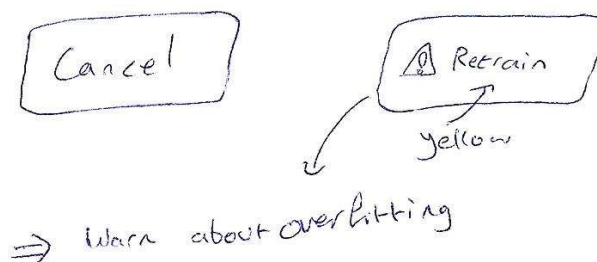


Figure 29: Termination Assessment - Variant 1 - Inflection Point Imminent

Figure 29 presents the termination assessment buttons with the previous ‘train’ button replaced with a ‘retrain’ button that has a warning sign next to it, which upon hovering, warns the user about the inflection point.

#### 4.1.7. Transfer

As a means to visualize the results of the prediction, in this step, the user is presented with the visualization layers that Kepler.GL has to offer to visualize them.

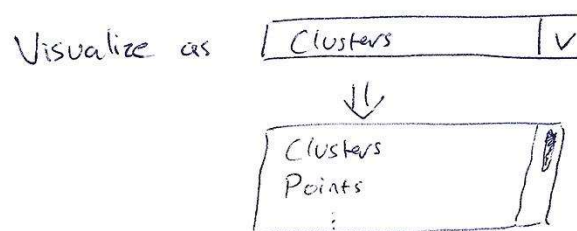


Figure 30: Transfer - Variant 1

Figure 30 presents the first variant of where the user can select the initial visualization layer offered from a drop-down menu. While this variant can be perceived to be relatively straightforward in presenting the options, the names of the visualization layers can have a learning curve, therefore, a different variant where the user can see an example shape of the layer with its name accompanied next to it can be a better solution.

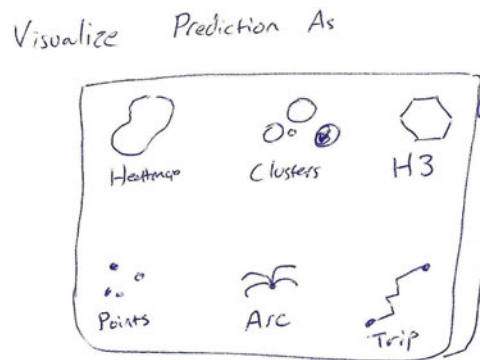


Figure 31: Transfer - Variant 2

Therefore, in Figure 31, the second variant shows the results of the predictions that can be visualized by selecting one of the available layers. As previously mentioned, an example shape of the layer and its name are believed to eliminate the learning curve for the user to memorize how each layer looks like. Furthermore, it matches the style that Kepler.GL follows in its interfaces.

---

#### 4.1.8. Model Browser

Due to the inevitable scenario of users wishing to continue their workflow from where they have left off, the need to design for loading-a-previously-created-model becomes necessary. However, to the best of the author's knowledge, the location of the button of where this functionality should be placed in respect to the workflow steps could not be ascertained. Consequently, the position of this button will be determined by the author in the final prototype available in Chapter 5 - Evaluation.

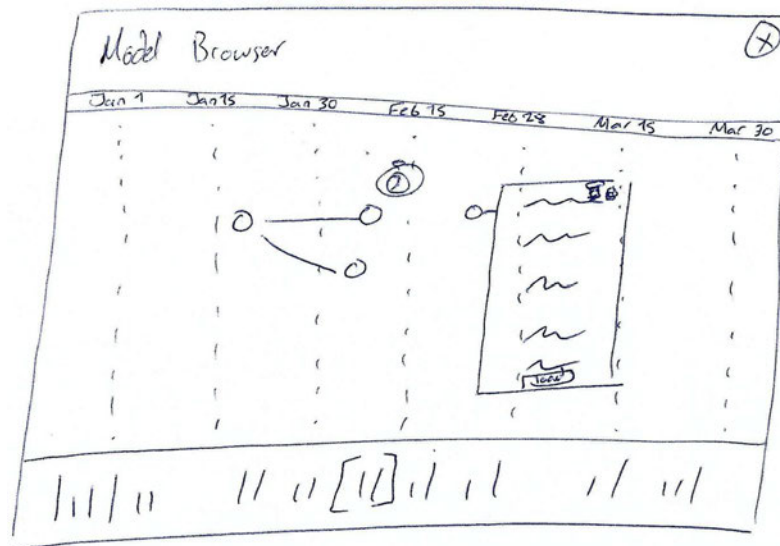


Figure 32: Model Browser

As the user clicks on the ‘Browse Previous Models’ button, the model browser window (Figure 32) appears. Starting from top to bottom, the title of the window is presented to the user along with a button to close the window opposite of it. Next, an element in the form of a responsive ribbon that indicates the date of when models were created is displayed. Below it, the models and their details are visually represented to the user as interactive nodes. The models are placed on the canvas in a manner that corresponds to their date of creation by users. Furthermore, a child-parent relationship is shown between the models by drawing a line between them, it is believed that the user will be able to discern the hierarchy between the nodes intuitively as the canvas’s x-axis represents time. Moreover, the status of each model is represented to the user with its color and - when needed - an accompanying icon. In the figure, a stopwatch icon is placed close to one of the nodes to indicate that the model is still training. Finally, as the user points over one of the nodes, an interactive tooltip appears containing details about the model, along with buttons to clone, delete, and to load the model onto the main interface. Below this canvas, an interactive timeline with a visual representation of the number of created models across a certain timespan is displayed, the user can find the models displayed on the canvas by dragging on either end of the available handles (denoted as squared brackets). It shall be noted here that although Figure 32 is a low-fidelity prototype, as the user drags either handle, the aforementioned ribbon and canvas should dynamically adjust and to display their values accordingly, this is believed to increase the communicativeness of the tool’s status [83] and its overall usability.

## 4.2. Medium-fidelity Prototype

Having discussed the variants and the functionalities that each step in the workflow contains in the previous section, a medium-fidelity prototype to illustrate these functionalities have been developed

in a manner that matches the theme of Kepler.GL and its elements to maintain the consistency of the tool used. The icon that was chosen to provide access to the IML interface resembles a brain fused with a circuit to signify that this tab contains some form of electronic intelligence. Each step of the workflow was designed discretely, along with the possibility of expanding most of the steps to grant the user further control over the general process [83]. To the preference of the author, the prototyping tool Figma [88] was used to create the subsequent interfaces.

#### 4.2.1. Feature Selection

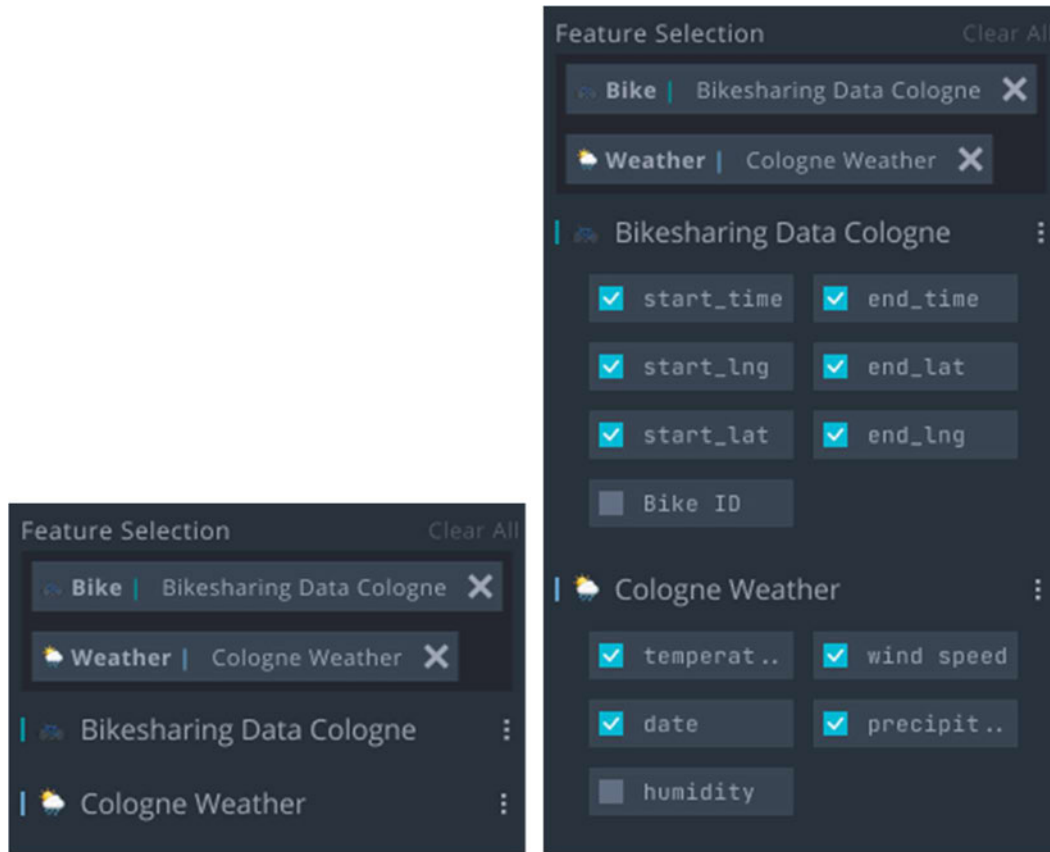


Figure 33: Feature Selection - Collapsed and Expanded

As presented in Figure 33, on the left, the users are presented with the feature selection interface, the datasets are selected by either typing in or by browsing for them from the multi-select field. The interface is capable of identifying which data each dataset represents, for example, the dataset named ‘Bikeshaing Data Cologne’ was correctly classified by the interface as a ‘Bike’ dataset that also associates a color to it, similarly, the dataset ‘Cologne Weather’ has been classified by the interface as a weather dataset with a light-blue color. It is worthy to note here that the association of colors with dataset categories was found to be beneficial, as it is generally known that colors help people identify categories without active conscious thought [89]. Since the user selected these two datasets, two entries of these datasets are populated beneath the input field. If the user clicks on the three

vertically aligned dots, the features of the dataset are displayed along with a checkbox to include, or exclude certain features from the dataset into the process as it is presented on the right.

### 4.2.2. Model Selection

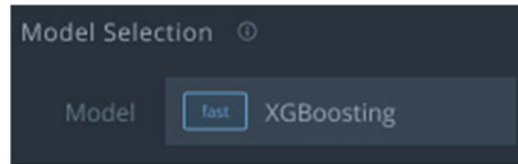


Figure 34: Model Selection

In the model selection step, users are able to select the desired model by interacting with the multi-select dropdown menu presented in Figure 34, this menu was particularly used due to its capability of displaying a category next to a model, which in this case, explicitly denotes the relative speed of the model.

### 4.2.3. Model Steering

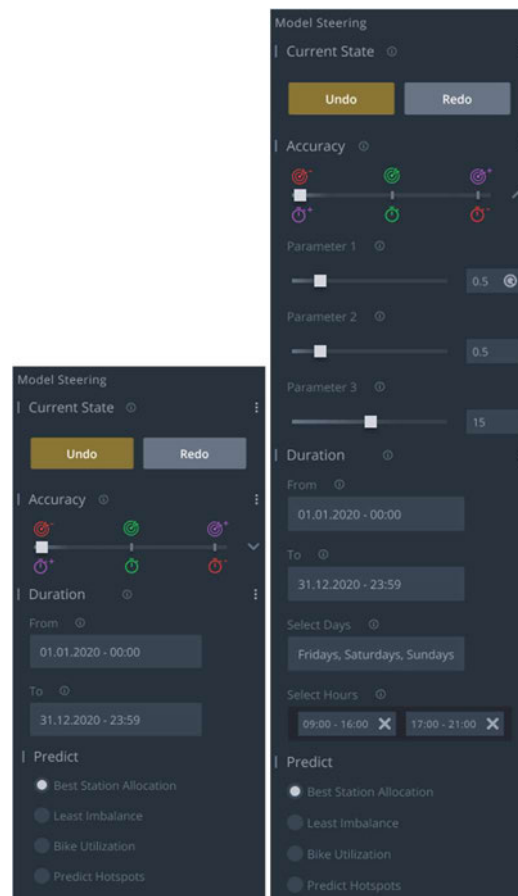


Figure 35: Model Steering - Collapsed and Expanded

Users are capable of steering the model by using the interface presented in Figure 35, on the left, the user is provided control over their progress by providing ‘Undo’ and ‘Redo’ buttons, not only does

this echo Jakob Nielsen’s “User Control and Freedom” usability heuristic [83], it was also highlighted by several papers finding that such a function yields significant importance in IML [26, 90–92]. Next, the trade-off between the speed required for a model to complete its work and its accuracy are provided to the user as a three-step slider [26]. Further, the user specifies the duration that they wish to include in the training process from the previously selected features from the ‘From’ and ‘To’ date-time-picker elements. Finally, the user chooses the desired prediction that the model should perform. On the right, the expanded interface presents further control over the accuracy and the duration sections, in accuracy, the user can adjust parameters related to the selected model via the available sliders, and in duration, the user can adjust the period to be included into the training process, as previously discussed in [c.f. 4.1.4 - Model Steering Part 2] the possibility of specifying certain days and certain spans of the day are presented to support and to provide discourse opportunities on the investigation of increased SMS during certain times of the day [84, 93], e.g., rush hours and public events.

#### 4.2.4. Quality Assessment

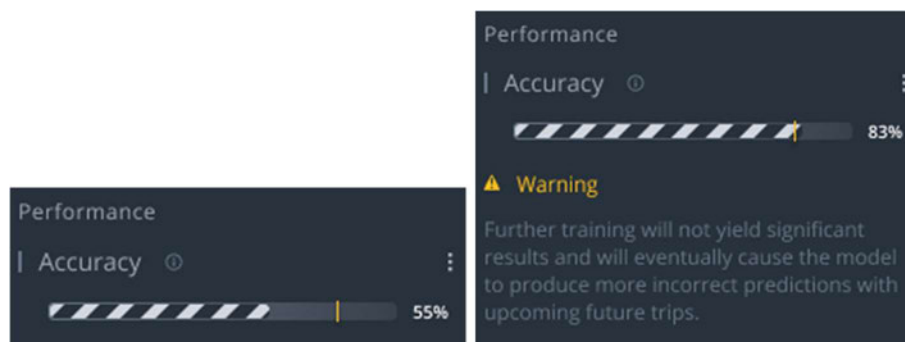


Figure 36: Quality Assessment - Collapsed and Expanded

While the user is engaged with the model steering process, the quality assessment interface constantly updates its values to reflect the quality state of the model. Additionally, the tool identifies and highlights to the user the inflection point of the learned concept in the form of a yellow vertical line [c.f. 2.2.1 - A Review of User Interface Design for Interactive Machine Learning], it warns the users when the quality of the model is near to/is past the point by showing a warning message which - in simple words - implores them to abstain from training the model further. The thought process that has influenced such a design decision was the findings of a recent paper published; it finds that using domain-specific terminologies can frustrate users while using IML applications, as well as, providing solutions to them as a measure to amend their errors [82].



Therefore, warning the users about the inflection point and the possibility of overfitting the model is undesired as opposed to showing a warning message that notifies the user that the model will produce more incorrect predictions on future new trips prior to them reaching this point.

#### 4.2.5. Termination Assessment

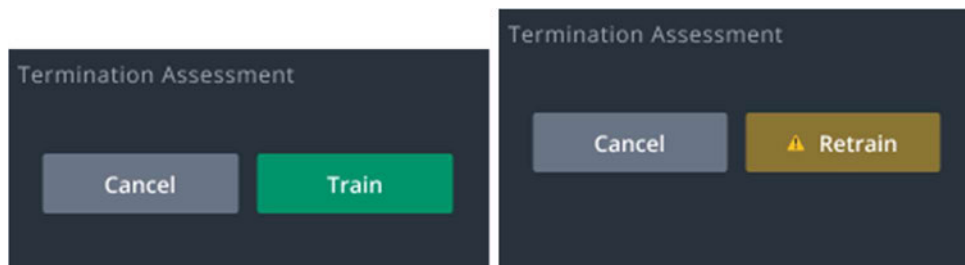


Figure 37: Termination Assessment Interface

As users steer the model in its respective interface, they are required to train the model after they have performed their adjustments by clicking on the ‘Train’ button as illustrated in Figure 37 on the left. However, as users near towards the inflection point of the model, the button gets replaced with a different button with a yellow background and a warning icon so that they are made aware of the potentially undesirable effect.

#### 4.2.6. Transfer

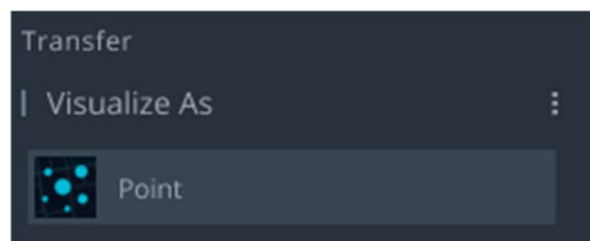


Figure 38: Transfer

In order for the users to visualize their results, the transfer interface presented in Figure 38 allows them to do so by selecting one of the included layers in Kepler.GL using the searchable dropdown menu.

### 4.2.7. Model Browser

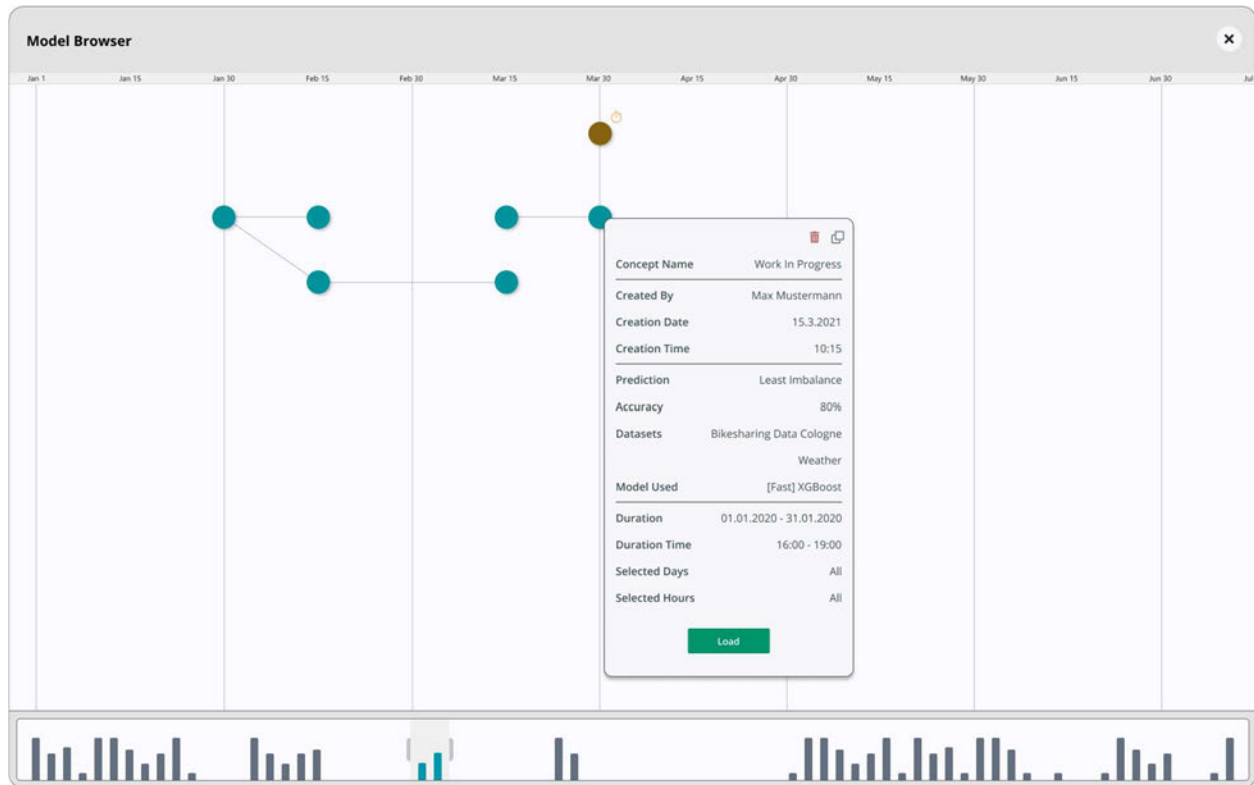


Figure 39: Model Browser

As a means to provide the users with the functionality of selecting previously made models for the aforementioned reasons [c.f. 4.1.8 - Model Browser], Figure 39 presents the model browser interface in a higher fidelity form.

In conclusion to this chapter, it can be perceived that while the variants that were presented for each step in the workflow in the low-fidelity prototype section, and the chosen ones in the medium-fidelity section are to be valid solutions for the research question at hand, a subtle challenge emerges: If we strictly adhere to the order of the process in the interface, we would find its order to be inconsistent. As a preliminary example for the next chapter: We notice that it is common practice to have call-to-action buttons - such as the ‘Train’ button in the termination assessment interface - to be placed on the bottom right of the interface. Consistent with this is the Gutenberg Diagram that was discussed by Hernandez and Resnick in which they denote “Content that is intended to be used at the end, such as a call to action button, should be placed in the lower right.” [94]. This finding reveals that this button should be placed as the final element in the interface. Indeed, these discrepancies are expected to occur, as Dudley and Kristensson note:

*“It is important to note, however, that these tasks may overlap and/or be performed iteratively. The structural and behavioural breakdowns are not intended to serve as blueprints but rather to support rich discussion of design considerations with clearly defined terminology. This formalisation of the design space will contribute to a better foundation for exchange of concepts in this emerging field.” - [26]*

It is therefore found to be necessary to perform an artificial formative evaluation to ensure that the interface follows a series of logical actions for the EUD.

## 5. Evaluation

As discussed in (Section 3.1.3), the SCW will be used to evaluate the interface by walking-through tasks that are necessary to ensure that the designed prototype lets the user to complete them. Most importantly, I will be evaluating the happy scenario first, afterwards, I will evaluate other tasks that are believed to yield benefit to the overall usability and learnability of the prototype. The actual CWs are available in the appendix and are ordered from 1 to 7. Each section number in this chapter corresponds to its respective CW number.

### 5.1. Happy Scenario: New Model

In order to tackle the aforementioned discrepancy, the recommendation of Wharton et. al. to evaluate “a single task [...] such as one that has proven problematic in a previous release” [72] can be used as a starting point in this chapter. In this task, a scenario of creating a model that predicts the best bike-station placement will be tested.

Following along with the CW (c.f. 10.1) – for a visual reference, refer to the figures in Section 4.2. The user clicks on the machine learning icon, selects the dataset, and its features. Next, the user selects the model from the drop-down menu and adjusts its accuracy if needed (in this simple CW, this will be ignored). Further, the user selects the duration of the historical data and constrains the duration to certain timespans. The user then selects the desired prediction (Best Station Allocation) and reviews the accuracy of the model (As it is a new model it will be at 0%). Finally, the user trains the model by clicking on the “Train” button and then chooses the method of visualization.

Analyzing the CW, several issues were found in the interface: 1) It can be argued that beginning the IML process with the selection of features can be challenging for beginners, even though the target users are expected to be trained to use this tool, future new users might be confused as to why datasets must be selected first. Therefore, the position of the feature selection UI can be demoted several positions down, along with the need to include an informational tooltip in the UI may be necessary.

2) The selection of the model can pose a significant challenge to the EUD, associating the speed of the model with its name may be indicative, however, the user needs to be further informed about the models. This indicates that the inclusion of a helping tooltip must be included close to the drop-down menu. 3) The section “Current State” which contains the “Redo” and “Undo” buttons can be either omitted when the user creates a new model, or the buttons can have less opacity to indicate that they are disabled at this stage. 4) Setting the accuracy of the model in a collapsed manner not only reduces the amount of space occupied, it also reduces the cognitive load of the user. For the time being, this part can be perceived as sufficient. 5) Setting the duration of the prediction can be a tedious task to be performed by the user, therefore, the date fields can be preset by default with the “From” field to be set 1 year before the user’s current date and the “To” field to be set to the current date of the user. 6) The labels “Select Days” and “Select Hours” can be found to be unclear with their purpose. Even though that each field has an informational tooltip associated with it, the labels can be renamed in a clearer manner. The author as a result sought to rename the labels to “Only include these weekdays” and “Only include these hours” respectively. 7) The position of the “Predict” menu can be perceived to not effectively communicate its criticality to the overall process. Resultingly, it was decided to promote its position higher in the UI. 8) The accuracy in the performance UI is expected to be set to 0%, however, a message encouraging the user to train the model can increase the usability of the UI as it suggests the desired action that the user can make while they are using the tool and to communicate the current status of the LC. Furthermore, the position of this UI can be promoted higher in the main UI. Moreover, the label “Performance” can be changed to “State” as this can be indicative of the LC’s current status. 9) The placement of the call-to-action button is problematic as additional setup still remains; therefore, it must be placed as the last element in the UI [94]. Furthermore, the UI does not provide feedback to the user when the button has been clicked and how long the training process will take. 10) Additionally, a button that allows the user to browse previously created models must be included in the interface, preferably to have it placed as the first element due to its expected high frequency of use.

As a result, this task can be deemed not possible to complete, mainly because the call-to-action button precedes the “transfer” UI. However, by performing the SCW and asking both of its questions (See 3.1.1 and 3.1.2 in Appendix 10.1), it also reveals that following a 1:1 naming scheme of each UI with the IML workflow can be confusing to the user. This indicates that further relabeling is required. Additionally, the general order of the elements can be improved by following a more logical and organized approach.

## 5.2. Happy Scenario: New Model – Revisited

Revisiting the previous task with the minor change of predicting hotspots instead of station-placement, the previous notes have been amended in the interface as it can be seen in Figure 40: SCW - 2.

The user clicks on the machine learning icon and is met with a Prediction UI that allows them to either browse previously created models or to select a prediction. In this case, the user clicks on “Predict Hotspots”. Moving on, the user is presented with the State UI which informs them about the current metrics of the LC and encourages them to train the model. On a side note, the user sees that LC history cannot be adjusted due to both buttons being disabled. Next, the Model Configuration UI prefills for the user the “From”, and “To” fields with the previously discussed values, and are presented with the two modified labels with their values set by default to include “all weekdays” and the entire hours of the day. Then, the user selects a suitable ML model from the drop-down menu and subsequently proceeds to set the accuracy (which is set to the fastest preset). Further, the user selects the dataset(s) they require from the Dataset Selection UI. It must be noted here that the interface automatically selects all of the features of the chosen dataset, if the user wishes to specify features, they are required to expand the dataset and to include/exclude features. Later on, the user selects Hotspot as the visualization method from the Finalize UI. Finally, the user clicks on the “Train” button, the interface disables the button and displays a success message to the user informing them that the model has started training, along with the possibility of viewing the current progress of the training process in the model browser window.

While the process of training the model in this CW can be perceived to be completable and relatively smoother than the previous model, some issues still remain: EUD new to the tool will still face difficulties determining a suitable ML model, and they might require proper orientation on the dataset feature selection and its features. These notes will be kept in mind and will be addressed once further complex scenarios have been evaluated.

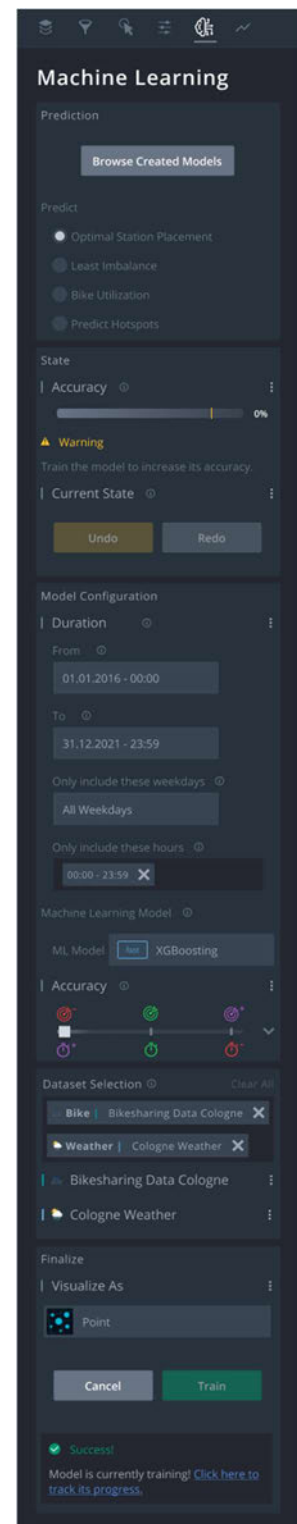


Figure 40: SCW - 2

### 5.3. Happy Scenario – Complex

As a more complicated task, this scenario tasks the user to create a model and to visualize it as a cluster - with set parameters to "1" – which predicts hotspots using the shared mobility and weather humidity data that was gathered during the weekend rush hours from the past 3 years.

Following along with the interface in Figure 40: SCW - 2, the user performs the actions that were mentioned before up until the Model Configuration UI. Here, the user sets the value of the fields “From” and “To” to 01/01/2018, and 31/12/2021, respectively. Next, the user removes the value “All Weekdays” from the “Only include these weekdays” and selects Saturdays and Sundays. Similarly, the user changes the value “00:00 – 23:59” in the field “Only include these hours” and sets it to “08:00 – 10:00” and “17:00 – 19:00”. Then, the user selects an ML model and expands the accuracy element. Upon expanding it, the user sets the parameters to 1 (as it was presented previously in Figure 35) and moves on to the dataset selection UI. Here, the user selects the shared mobility data dataset and the weather data. However, the user is required to be certain that the humidity feature is selected by expanding the weather dataset and to make sure that it is included in the LC. Finally, the user selects “Cluster” as the visualization method and ultimately proceeds to train the LC.

After analyzing this CW, it was found that the label “Accuracy” can be ambiguous to the user, especially since it is mentioned twice, thus making it necessary to rephrase it in a manner that implies that it contains further useful parameters. Also, having the user consciously expanding a dataset to make sure that certain features are going to be included can impact the user-friendliness of the UI, therefore, in the case of creating a new model, it was decided that the selected datasets would be expanded by default.

Additionally, the labels “Prediction” and “Predict” in the first UI can be further abstracted to adequately address the capability of ML to Predict *and* to Cluster. For this, the labels “Model Purpose” and “Use Machine Learning To” were respectively used. With these label changes combined with the previous UI ordering challenge, the author finds that the labels and the elements can be perceived to resemble a wizard following a series of logical steps.

Fortunately, a design idea surfaced involving the informational tooltips and the preset feature in a manner that can provide a reasonable solution to aid the user in selecting a suitable machine learning model and dataset features. This will be discussed in the upcoming CW and in the discussion chapter.

### 5.4. Happy Scenario – Complex – Revisited

Revisiting the previous scenario, the label changes and the expansion of the selected datasets by default yielded positive results when both SCW questions for their respective tasks were answered. Additionally, implementing the aforementioned design idea as presented in Figure 41 are positively answered in the SCW questions as well.

Briefly explaining the CW, the user selects “Predict Hotspots”, this causes the interface to automatically apply a pre-implemented preset that selects the ML model and informs the user about it with an explanation of the selected model itself. Skipping to the Model Configuration UI, a new icon in the form of a graduation cap is added which contains an onboarding process for the user to inform them about ML. Similar to this, the Dataset Selection UI has the same icon that informs the user about the necessity of including datasets in ML and the appropriate features to include into the selected purpose of the model.

Further analyzing the CW, it was found that when visualizing clusters, the ability to show/hide the clusters – and other appropriate visualization methods – on spatially constrained areas on the map can yield additional benefit. For example, a user can perform this CW and can later show heatmaps constrained to certain zones in the city or certain zip codes. This design idea will be addressed in the final form of the tool.

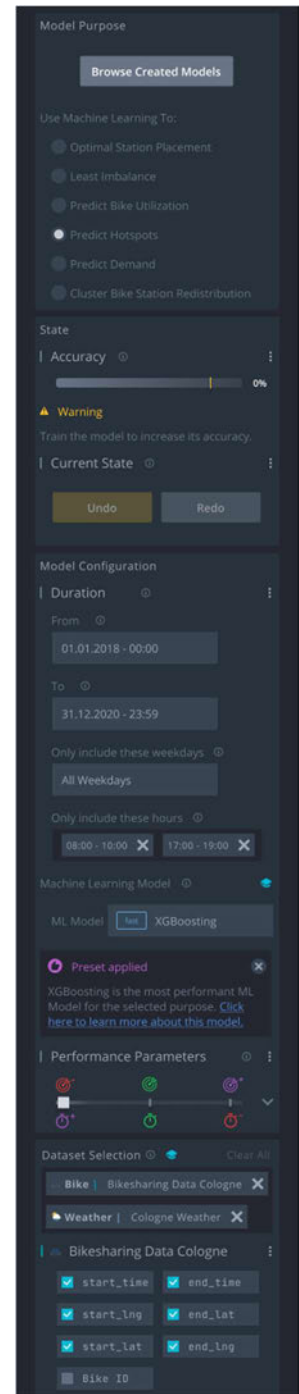


Figure 41: SCW - 4

### 5.5. Loaded Model

In this task, the user loads a previously created model, retrains it, and visualizes it as points on the map. First, the user clicks on the “Browse Created Models” button and chooses a random model from the model browser window. As a result, the main UI prefills its fields with parameters of the loaded model. Next, the user reviews the accuracy of the model, and the decision of retraining the model will be left to the user.

Examining this CW, it was found that if the user reviews the accuracy and realizes that the inflection point is going to be exceeded, retraining the model in order to visualize it becomes undesirable and causes a potential usability problem. Therefore, the interface must provide the ability to visualize the loaded model without retraining it. This was revised and can be seen in Figure 42.

### 5.6. User Wants to Define a New Use of ML

The user wishes to define a new use in ML. However, the interface does not offer this functionality. Therefore, this test is incompletable and the addition of a new button for the EUD to develop their own ML use is necessary.

### 5.7. User Defines a New Use of ML

Subsequent to revising the interface to address the previous functionality, a button has been added into the interface to allow the EUD to define a new use of ML. As presented in Figure 43, the user clicks on the machine learning icon, realizes that their desired use is not available and consequently clicks on the “Define New ML Use” button. This displays a new window that allows the EUD to perform the following: 1) Name the use of ML (Which will be present as a new option in the “Use machine learning to” element along with a distinguishing artefact to denote that this particular ML use has been made by an EUD). 2) Develop their own use of ML. 3) Input the preset values for the ML model and its preset performance parameters.

In reflection to this SCW, not only was the addition of this button was found to be beneficial for EUDs, it also provides an opportunity for discourse between the EUDs, ML experts, and the suppliers of this tool. Arguably, by

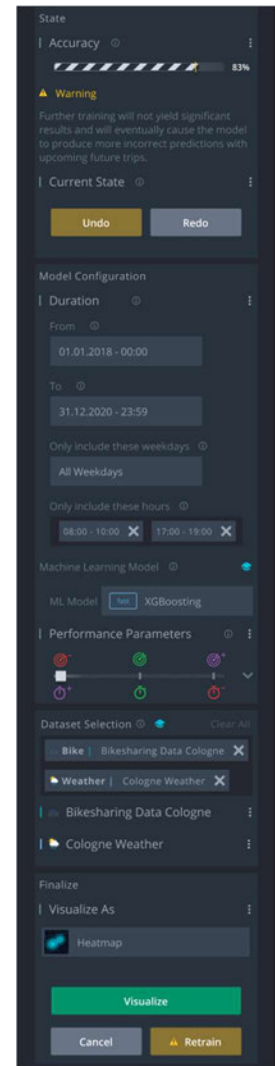


Figure 42: SCW - 5

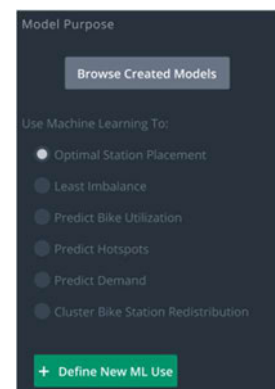


Figure 43: SCW - 7



having new uses of ML and their respective presets to be relatively easy to define in the UI, it achieves one of the expected outcomes of this tool (extendibility of the tool).

## 5.8. Conclusion

In conclusion to this chapter and to the entire design process. The consolidation of all of the previously highlighted requirements, features, proposed solutions to usability and learnability problems, design ideas, and design gaps has led to the creation of the following UI<sup>2</sup>:

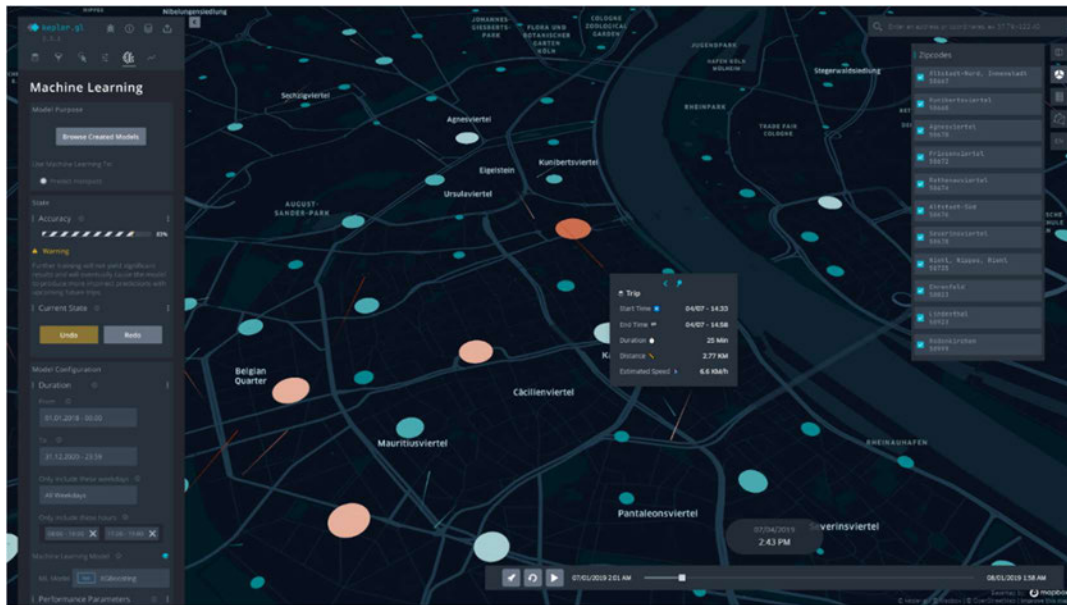


Figure 44: Complete UI - Dark Mode

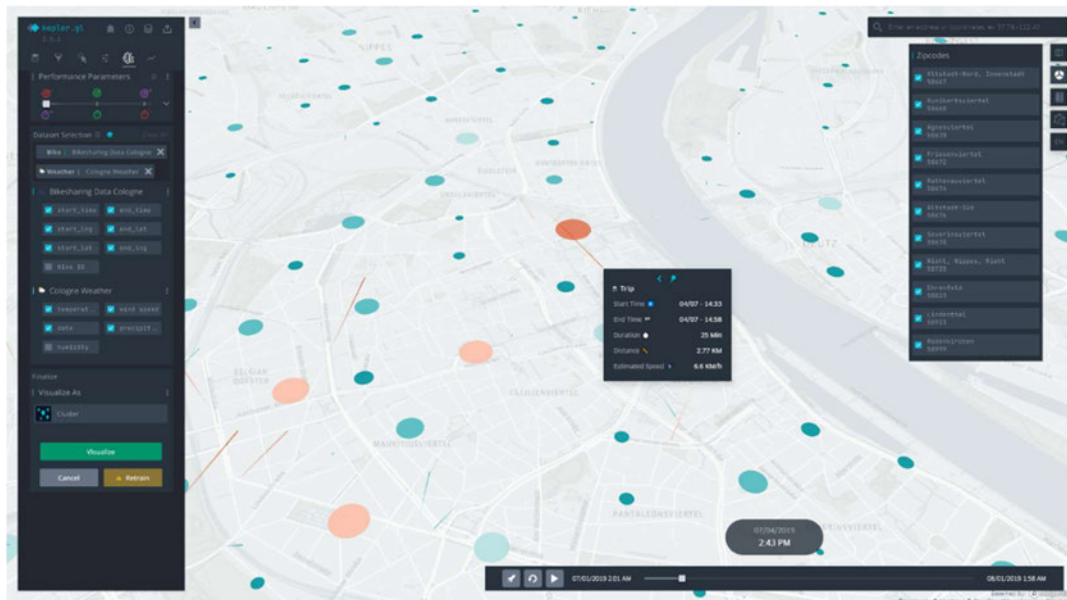


Figure 45: Complete UI - Light Mode

<sup>2</sup> The complete UI and its assets can be accessed through the following link:  
<https://www.figma.com/file/ZSSyVDHGT0CTuBqJnFV4go/Medium-Fidelity-Prototype?node-id=0%3A1>

## 6. Discussion

Creating a unifying platform that facilitates coordinated work on SMS and the transportation system can directly yield immense benefits to the target audience and indirectly to the metropolitans and their quality of life. Having the tool deployed in an environment that – upon authorization - can be interacted with through an internet browser removes the barriers that could hinder discourse. Furthermore, as such a sophisticated system with ML capabilities can be accessed through the internet, regional and/or international consultations could be arranged to find solutions to challenges that the target audience could face. These consultations can include data science experts, urban transportation researchers, city planners, and other shared mobility operators. Moreover, the tool can be beneficial in cases where the city municipality desires to coordinate the transportation system and SMS for upcoming city-wide events, such as festivals, elections, demonstrations, and sports events. Not only can this provide valuable insight for all the concerned parties, but it also can increase the cities' efficiency. Also, the capabilities of ML can help city municipalities to make better-informed decisions that are believed to be ultimately beneficial to the metropolitans, along with all parties involved.

As Kepler.GL was capable - to a wide degree – of satisfying the requirements of this project, the decision of selecting it as the main tool to visualize datasets can be even more useful. This is due to its ability to handle other datasets that were not mentioned in this thesis: combined with the possibility of adding datasets in a drag-and-drop manner to further investigate a certain issue, it can be a deciding factor to use this tool to gain quick insight by other fellow city municipalities for other use cases. For example, an EUD might upload datasets related to the economy of their city to investigate a relationship between the economical situation of neighborhoods and the most suitable mode of transportation to offer, this can provide help in the decision-making process and can be beneficial to the city's planning.

Reflecting on the use of the Streamlined Cognitive Walkthrough as an evaluation method in respect to the posed constraints was found to be effective to this work. The method helped in exposing problems in the design relatively quickly, it allowed the author to rapidly amend them, and to come up with design ideas that were testable in a short time. Furthermore, answering both questions at each step of the evaluation helped the author to criticize the design and to critically reflect on it, thus allowing to find more suitable approaches. Moreover, due to its repetitiveness, it has assisted the author to identify underlying patterns in the IML process that has helped in implementing new ideas into the design which solved some design issues that appeared during the evaluation process. This has helped the author to further explore the body of literature using less-abstract and more-constrained

search terms. Upon combining this with the feedback gained from the evaluating committee of this thesis, it has caused the UI to be positively impacted. As the author presented the medium-fidelity prototype and has immediately started to evaluate the prototype with the received feedback in mind, the formulation of the "credible stories" that [72] have instructed the evaluator to perform was of benefit to the interface. This has led to the design idea of providing the functionality to preset elements in the interface that were constantly failing as they were being evaluated (c.f. Section 5.1, 5.2, 5.3). To clarify, the recent systematic literature review [95] highlights the uses of ML and which algorithm was the most performant in particular bike-sharing contexts. With this, the radio buttons under the "Use machine learning to:" label, and their accompanying presets could be implemented in a relatively straightforward manner as each update to the ML aspect of the system can be added as a new button/drop-down menu option with a recommended preset that is grounded by the literature.

## 7. Future Work and Limitations

Due to the vast scope of this thesis, several aspects remain in need of investigation and evaluation. First, information tooltips need to be carefully formulated and validated with the target audience to clarify particular ML elements of the system.

Second, the content of the preset message that appears once a purpose has been selected can be a portal towards the field of Explainable-AI to adequately explain ML algorithms to EUDs.

Third, while the findings of [95] could be used to rank the ML models in their respective drop-down menu, further investigation is needed for more ML uses in other shared mobility modes of transport. Significant progress could not be achieved in regards to this within the allotted timeframe for the thesis. Resultingly, this can be considered as a limitation that requires further investigation and additional evaluation with real users.

Fourth, the content of the graduation cap icon remains to be populated with content that informs the users on how to use the tool. For this, several mediums could be considered, such as, video tutorials, images, text, etc.

Fifth, due to certain training sessions extended time periods, a notification mechanism can be implemented to inform the user once it finishes. As a suggestion, this option can be added in the Finalize UI, and the initial setup for the desired notification service can be added in Opendash.

Sixth, cooperative methods can be implemented when an EUD defines a new use of IML in the manner that they can cooperate with the ML experts and the creators of the tool to develop, troubleshoot and to possibly share these ideas.

Seventh, while the SCW has helped in this initial stage of the project, evaluating the UI with real users still remains paramount.

## **8. Conclusion**

To the benefit of this thesis, the literature, methodologies, and materials used resulted in the creation of a tool that is capable of satisfying a significant number of the features that were highlighted by research stemming from the related fields.

This, along with the creation of an interface that allows non-expert users in positions of high impact on the cities' transportation network to perform complex machine learning processes that ultimately aim to increase the well-being of their fellow metropolitans, to optimize the transportation system, and to reduce its impact on the environment seem to have satisfied the expected outcomes of this work.

Influenced by the literature and the methods used, the tool and its UIs were designed and are aimed to be implemented in an extendible manner that allows future versions to include new functions and new ML uses without the need to perform drastic changes to the core code or its UIs.

## 9. References

1. Andersen, P.H., Mathews, J.A., Rask, M.: Integrating private transport into renewable energy policy: The strategy of creating intelligent recharging grids for electric vehicles. *Energy Policy*. 37, 2481–2486 (2009). <https://doi.org/10.1016/j.enpol.2009.03.032>.
2. Machado, C.A.S., Hue, N.P.M. de S., Berssaneti, F.T., Quintanilha, J.A.: An overview of shared mobility. *Sustain.* 10, 1–21 (2018). <https://doi.org/10.3390/su10124342>.
3. Buehler, R., Pucher, J., Gerike, R., Götschi, T.: Reducing car dependence in the heart of Europe: lessons from Germany, Austria, and Switzerland. *Transp. Rev.* 37, 4–28 (2017). <https://doi.org/10.1080/01441647.2016.1177799>.
4. Canzler, W., Wittowsky, D.: The impact of Germany’s Energiewende on the transport sector – Unsolved problems and conflicts. *Util. Policy.* 41, 246–251 (2016). <https://doi.org/10.1016/j.jup.2016.02.011>.
5. Preisler, T., Dethlefs, T., Renz, W.: Self-organizing redistribution of bicycles in a bike-sharing system based on decentralized control. *Proc. 2016 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2016.* 8, 1471–1480 (2016). <https://doi.org/10.15439/2016F126>.
6. Beveridge, R., Kern, K.: The Energiewende in Germany: Background, Developments and Future Challenges Special Issue: Energy Grids and Infrastructure. *Renew. Energy Law Policy Rev.* 2013, 3–12 (2013).
7. Krautzberger, M., Umweltbundes-, P., Klimaschutzplan, D.: Elektromobilität volkswirtschaftlich klar im Vorteil | Umweltbundesamt. 49, (2016).
8. Johnsen, D., Vorholt, F., Gieschen, J.H., Müller, B., Randhahn, A.: Electric mobility and smart mobility concepts-restrained uptake in german cities. *World Electr. Veh. J.* 10, (2019). <https://doi.org/10.3390/wevj10040081>.
9. Pucher, J., Buehler, R.: Cycling for everyone: Lessons from Europe. *Transp. Res. Rec.* 58–65 (2008). <https://doi.org/10.3141/2074-08>.
10. Pucher, J., Buehler, R.: Making Cycling Irresistible: Lessons from The Netherlands, Denmark and Germany. *Transp. Rev.* 28, 495–528 (2008). <https://doi.org/10.1080/01441640701806612>.
11. McKenzie, G.: Urban mobility in the sharing economy: A spatiotemporal comparison of shared mobility services. *Comput. Environ. Urban Syst.* 79, 101418 (2020). <https://doi.org/10.1016/j.compenvurbsys.2019.101418>.
12. Shen, S., Wei, Z.Q., Sun, L.J., Su, Y.Q., Wang, R.C., Jiang, H.M.: The shared bicycle and its network—internet of shared bicycle (IoSB): A review and survey. *Sensors (Switzerland)*. 18, 1–24 (2018). <https://doi.org/10.3390/s18082581>.
13. Büttner, J., Mlasowsky, H., Birkholz, T.: Optimising Bike Sharing in European Cities - A Handbook. 90 (2011).
14. Department for City Planning New York: Bike-Share. Opportunities in New York City. New York. 142 (2009).
15. Chen, E., Ye, Z.: Identifying the nonlinear relationship between free-floating bike sharing usage and built environment. *J. Clean. Prod.* 280, 124281 (2021). <https://doi.org/10.1016/j.jclepro.2020.124281>.
16. Cohen, A., Shaheen, S.: Planning for shared mobility. (2016). <https://doi.org/10.7922/G2NV9GDD>.

17. Nikitas, A., Wallgren, P., Rexfelt, O.: The paradox of public acceptance of bike sharing in Gothenburg. *Proc. Inst. Civ. Eng. Eng. Sustain.* 169, 101–113 (2016). <https://doi.org/10.1680/jensu.14.00070>.
18. Zademach, H.M., Musch, A.K.: Bicycle-sharing systems in an alternative/diverse economy perspective: a sympathetic critique. *Local Environ.* 23, 734–746 (2018). <https://doi.org/10.1080/13549839.2018.1434494>.
19. Schönberg, T., Dyskin, A., Ewer, K.: Bike Sharing 5.0. 27 (2018).
20. R, A.: Bye Bye, Obikes!, <https://themunicheye.com/bye-bye-obikes-3604>, last accessed 2021/02/23.
21. Chen, L., Yang, D., Jakubowicz, J., Pan, G., Zhang, D., Li, S.: Sensing the pulse of urban activity centers leveraging bike sharing open data. *Proc. - 2015 IEEE 12th Int. Conf. Ubiquitous Intell. Comput. 2015 IEEE 12th Int. Conf. Adv. Trust. Comput. 2015 IEEE 15th Int. Conf. Scalable Comput. Commun.* 20, 135–142 (2016). <https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCCom-IoP.2015.43>.
22. Caggiani, L., Ottomanelli, M.: A Dynamic Simulation based Model for Optimal Fleet Repositioning in Bike-sharing Systems. *Procedia - Soc. Behav. Sci.* 87, 203–210 (2013). <https://doi.org/10.1016/j.sbspro.2013.10.604>.
23. Jiao, J., Bai, S.: Understanding the shared e-scooter travels in Austin, TX. *ISPRS Int. J. Geo-Information.* 9, (2020). <https://doi.org/10.3390/ijgi9020135>.
24. Lee, T., Johnson, J., Cheng, S.: An Interactive Machine Learning Framework. (2016).
25. Hevner Alan, R.: A Three Cycle View of Design Science Research. *Scand. J. Inf. Syst.* 19, 87–92 (2007).
26. Dudley, J.J., Kristensson, P.O.: A Review of User Interface Design for Interactive Machine Learning. *ACM Trans. Interact. Intell. Syst.* 8, 1–37 (2018). <https://doi.org/10.1145/3185517>.
27. Spencer, R.: The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company. *Conf. Hum. Factors Comput. Syst. - Proc.* 2, 353–359 (2000). <https://doi.org/10.1145/332040.332456>.
28. kepler.gl, <https://kepler.gl/>, last accessed 2020/09/05.
29. open.INC, <https://openinc.de/>, last accessed 2020/09/05.
30. Highcharts | Highcharts.com, <https://www.highcharts.com/demo>, last accessed 2020/09/05.
31. Lieberman, H., Paternò, F., Repenning, A., Wulf, V.: Perspective on end user development. *Conf. Hum. Factors Comput. Syst. - Proc.* 1048–1049 (2003). <https://doi.org/10.1145/765891.766142>.
32. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End-User Development: An Emerging Paradigm. *End User Dev.* 1–8 (2006). [https://doi.org/10.1007/1-4020-5386-x\\_1](https://doi.org/10.1007/1-4020-5386-x_1).
33. Pantazos, K., Lauesen, S., Vatrapu, R.: End-user development of information visualization. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*. 7897 LNCS, 104–119 (2013). [https://doi.org/10.1007/978-3-642-38706-7\\_9](https://doi.org/10.1007/978-3-642-38706-7_9).
34. Fischer, G.: End-user development: From creating technologies to transforming cultures. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*. 7897 LNCS, 217–222 (2013). [https://doi.org/10.1007/978-3-642-38706-7\\_16](https://doi.org/10.1007/978-3-642-38706-7_16).

35. Viegas, F.B., Wattenberg, M., Van Ham, F., Kriss, J., McKeon, M.: Many Eyes: A site for visualization at internet scale. *IEEE Trans. Vis. Comput. Graph.* 13, 1121–1128 (2007). <https://doi.org/10.1109/TVCG.2007.70577>.
36. Marasoiu, M., Nauck, D., Blackwell, A.F.: Cuscus: An End User Programming Tool for Data Visualisation. Springer International Publishing (2019). [https://doi.org/10.1007/978-3-030-24781-2\\_8](https://doi.org/10.1007/978-3-030-24781-2_8).
37. Fails, J.A., Olsen, D.R.: Interactive machine learning. *Int. Conf. Intell. User Interfaces, Proc. IUI.* 39–45 (2003). <https://doi.org/10.1145/604045.604056>.
38. Fogarty, J., Tan, D., Kapoor, A., Winder, S.: CueFlik. 29 (2008). <https://doi.org/10.1145/1357054.1357061>.
39. Sanchez-Cortina, I., Serrano, N., Sanchis, A., Juan, A.: A prototype for interactive speech transcription balancing error and supervision effort. *Int. Conf. Intell. User Interfaces, Proc. IUI.* 325–326 (2012). <https://doi.org/10.1145/2166966.2167035>.
40. Huang, S.W., Tu, P.F., Fu, W.T., Amanzadeh, M.: Leveraging the crowd to improve feature-sentiment analysis of user reviews. *Int. Conf. Intell. User Interfaces, Proc. IUI.* 3–13 (2013). <https://doi.org/10.1145/2449396.2449400>.
41. Kim, B., Glassman, E., Johnson, B., Shah, J.: massachusetts institute of technology iBCM: Interactive Bayesian Case Model Empowering Humans via Intuitive Interaction iBCM: Interactive Bayesian Case Model Empowering Humans via Intuitive Interaction. (2015).
42. Amershi, S., Fogarty, J., Weld, D.S.: ReGroup: Interactive machine learning for on-demand group creation in social networks. In: *Conference on Human Factors in Computing Systems - Proceedings.* pp. 21–30. ACM Press, New York, New York, USA (2012). <https://doi.org/10.1145/2207676.2207680>.
43. Shneiderman, B.: The future of interactive systems and the emergence of direct manipulation. *Behav. Inf. Technol.* 1, 237–256 (1982). <https://doi.org/10.1080/01449298208914450>.
44. Card, Stuart K. and Mackinlay, Jock D. and Shneiderman, B.: *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999).
45. Chi, E.H. Hsin, Riedl, J.T.: Operator interaction framework for visualization systems. In: *Proceedings of the IEEE Symposium on Information Visualization.* pp. 63–70. IEEE (1998). <https://doi.org/10.1109/infvis.1998.729560>.
46. Cakmak, M., Chao, C., Thomaz, A.L.: Designing Interactions for Robot Active Learners. *IEEE Trans. Auton. Ment. Dev.* 2, 108–118 (2010). <https://doi.org/10.1109/tamd.2010.2051030>.
47. Porter, R., Theiler, J., Hush, D.: Interactive machine learning in data exploitation. *Comput. Sci. Eng.* 15, 12–20 (2013). <https://doi.org/10.1109/MCSE.2013.74>.
48. Raghavan, H., Madani, O., Jones, R.: Active learning with feedback on both features and instances. *J. Mach. Learn. Res.* 7, 1655–1686 (2006).
49. Brooks, M., Amershi, S., Lee, B., Drucker, S.M., Kapoor, A., Simard, P.: FeatureInsight: Visual support for error-driven feature ideation in text classification. In: *2015 IEEE Conference on Visual Analytics Science and Technology, VAST 2015 - Proceedings.* pp. 105–112. Institute of Electrical and Electronics Engineers Inc. (2015). <https://doi.org/10.1109/VAST.2015.7347637>.
50. Kleinsmith, A., Gillies, M.: Customizing by doing for responsive video game characters. *Int.*

- J. Hum. Comput. Stud. 71, 775–784 (2013). <https://doi.org/10.1016/j.ijhcs.2013.03.005>.
51. Amershi, S.: Designing for effective end-user interaction with machine learning. In: *UIST’11 Adjunct - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. pp. 47–50 (2011). <https://doi.org/10.1145/2046396.2046416>.
  52. Nielsen, J.: Usability inspection methods. In: *Conference on Human Factors in Computing Systems - Proceedings*. pp. 413–414. Association for Computing Machinery (1994). <https://doi.org/10.1145/259963.260531>.
  53. Boufidis, N., Nikiforiadis, A., Chrysostomou, K., Aifadopoulou, G.: Development of a station-level demand prediction and visualization tool to support bike-sharing systems’ operators. *Transp. Res. Procedia*. 47, 51–58 (2020). <https://doi.org/10.1016/j.trpro.2020.03.072>.
  54. Oppermann, M., Möller, T., Sedlmair, M.: Bike Sharing Atlas: Visual Analysis of Bike-Sharing Networks. *Int. J. Transp.* 6, 1–14 (2018). <https://doi.org/10.14257/ijt.2018.6.1.01>.
  55. Shneiderman, B.: “The Eyes Have It: A Task by, (1996).
  56. Shneiderman, B.: Direct manipulation. In: *Proceedings of the joint conference on Easier and more productive use of computer systems. (Part - II) Human interface and the user interface -*. p. 143. ACM Press, New York, New York, USA (1981). <https://doi.org/10.1145/800276.810991>.
  57. Janssen, M., Charalabidis, Y., Zuiderwijk, A.: Benefits, Adoption Barriers and Myths of Open Data and Open Government. *Inf. Syst. Manag.* 29, 258–268 (2012). <https://doi.org/10.1080/10580530.2012.716740>.
  58. The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economics, Societies and Nations. *Pers. Psychol.* 59, 982–985 (2006). [https://doi.org/10.1111/j.1744-6570.2006.00060\\_10.x](https://doi.org/10.1111/j.1744-6570.2006.00060_10.x).
  59. Bike Visualization, <https://www.visualization.bike/>, last accessed 2021/03/18.
  60. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: A Framework for Evaluation in Design Science Research. *Eur. J. Inf. Syst.* 25, 77–89 (2016). <https://doi.org/10.1057/ejis.2014.36>.
  61. Seibel, B.: CityLAB, [https://www.citylab-berlin.org/projects\\_en/bikesharing](https://www.citylab-berlin.org/projects_en/bikesharing), last accessed 2021/03/20.
  62. Seibel, B.: Bike sharing flows, <https://bikesharing.citylab-berlin.org/>, last accessed 2021/03/20.
  63. deck.gl, <https://deck.gl/>, last accessed 2020/09/05.
  64. Earn Money by Driving or Get a Ride Now | Uber Germany, <https://www.uber.com/de/en/>, last accessed 2020/09/05.
  65. Hevner, A.R., March, S.T., Park, J., Ram, S.: Two Paradigms on Research Essay Design Science in Information Systems Research. *MIS Q.* 28, 75–79 (2004).
  66. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: A Framework for Evaluation in Design Science Research. *Eur. J. Inf. Syst.* 25, 77–89 (2016). <https://doi.org/10.1057/ejis.2014.36>.
  67. Venable, J., Pries-Heje, J., Baskerville, R.: A Comprehensive Framework for Evaluation in Design Science Research: Advances in Theory and Practice (DESRIST 2012). *Proc. 7th Int. Conf. Des. Sci. Res. Inf. Syst.* 423–438 (2012).
  68. Venable, J., Pries-Heje, J., Baskerville, R.: A Comprehensive Framework for Evaluation in



- Design Science Research. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 423–438 (2012). [https://doi.org/10.1007/978-3-642-29863-9\\_31](https://doi.org/10.1007/978-3-642-29863-9_31).
69. Mijač, M.: Evaluation of Design Science Instantiation Artifacts in Software Engineering Research. CECIIS 2019 Proc. 30th Cent. Eur. Conf. Inf. Intell. Syst. Oct. 2-4, 2019, Varaždin, Croat. 313–321 (2019).
  70. Monk, A., Wright, P., Haber, J., Davenport Appendix, L.: A volume in the BCS Practitioner Series ©Prentice-Hall Improving your human-computer interface: a practical technique Cooperative Evaluation: A run-time guide. (1993).
  71. Sonnenberg, C., Vom Brocke, J.: Evaluation patterns for design science research artefacts. Commun. Comput. Inf. Sci. 286 CCIS, 71–83 (2012). [https://doi.org/10.1007/978-3-642-33681-2\\_7](https://doi.org/10.1007/978-3-642-33681-2_7).
  72. Wharton, C., Rieman, J., Lewis, C., Polson, P.: The cognitive walkthrough method: A practitioner’s guide, (1994).
  73. Mahatody, T., Sagar, M., Kolski, C.: State of the art on the cognitive walkthrough method, its variants and evolutions. Int. J. Hum. Comput. Interact. 26, 741–785 (2010). <https://doi.org/10.1080/10447311003781409>.
  74. Yigitbasioglu, O.M., Velcu, O.: A review of dashboards in performance management: Implications for design and research. Int. J. Account. Inf. Syst. 13, 41–59 (2012). <https://doi.org/10.1016/j.accinf.2011.08.002>.
  75. Add Data to the Map - kepler.gl, <https://docs.kepler.gl/docs/user-guides/b-kepler-gl-workflow/a-add-data-to-the-map#ways-to-add-data>, last accessed 2021/04/09.
  76. Upgrade Guide - kepler.gl, <https://docs.kepler.gl/upgrade-guide#upgrade-from-v-1-1-12-to-v-2-0>, last accessed 2021/04/09.
  77. React – A JavaScript library for building user interfaces, <https://reactjs.org/>, last accessed 2020/09/05.
  78. Redux - A predictable state container for JavaScript apps. | Redux, <https://redux.js.org/>, last accessed 2020/09/05.
  79. keplergl/kepler.gl: Kepler.gl is a powerful open source geospatial analysis tool for large-scale data sets., <https://github.com/keplergl/kepler.gl#5-render-custom-ui-components>, last accessed 2020/09/05.
  80. miaas-dashboard/custom-panel.js at hadidi-keplergl · lukasboehm/miaas-dashboard, <https://github.com/lukasboehm/miaas-dashboard/blob/hadidi-keplergl/app/js/widgets/keplergl-staging/keplergl-components/custom-panel.js>, last accessed 2020/09/05.
  81. Euclidean distance - Wikipedia, [https://en.wikipedia.org/wiki/Euclidean\\_distance](https://en.wikipedia.org/wiki/Euclidean_distance), last accessed 2021/04/11.
  82. Nodalo, G., Santiago, J.M., Valenzuela, J., Deja, J.A.: On building design guidelines for an interactive machine learning sandbox application. ACM Int. Conf. Proceeding Ser. 70–77 (2019). <https://doi.org/10.1145/3328243.3328253>.
  83. Usability, T.: Heuristic Evaluation of Nielsen ’ s Ten Usability Heuristics, <https://pdfs.semanticscholar.org/5f03/b251093aee730ab9772db2e1a8a7eb8522cb.pdf>, last accessed 2021/05/21.

84. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: DNN-Based Prediction Model for Spatio-Temporal Data. (2016).
85. Chen, P., Hsieh, H., Su, K., Sigalingging, X.K., Chen, Y., Leu, J.: Predicting station level demand in a bike-sharing system using recurrent neural networks. *IET Intell. Transp. Syst.* 14, 554–561 (2020). <https://doi.org/10.1049/iet-its.2019.0007>.
86. Vogel, P., Greiser, T., Mattfeld, D.C.: Understanding bike-sharing systems using Data Mining: Exploring activity patterns. *Procedia - Soc. Behav. Sci.* 20, 514–523 (2011). <https://doi.org/10.1016/j.sbspro.2011.08.058>.
87. Schuijbroek, J., Hampshire, R.C., van Hoes, W.J.: Inventory rebalancing and vehicle routing in bike sharing systems. *Eur. J. Oper. Res.* 257, 992–1004 (2017). <https://doi.org/10.1016/j.ejor.2016.08.029>.
88. Figma: the collaborative interface design tool., <https://www.figma.com/>, last accessed 2021/05/24.
89. Lin, S., Fortuna, J., Kulkarni, C., Stone, M., Heer, J.: Selecting semantically-resonant colors for data visualization. *Comput. Graph. Forum.* 32, 401–410 (2013). <https://doi.org/10.1111/cgf.12127>.
90. Amershi, S., Fogarty, J., Kapoor, A., Tan, D.: Examining Multiple Potential Models in End-User Interactive Concept Learning. (2010).
91. Bernardo, F., Zbyszryski, M., Fiebrink, R., Grierson, M.: Interactive machine learning for end-user innovation. *AAAI Spring Symp. - Tech. Rep. SS-17-01-*, 369–375 (2017).
92. Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T.: Power to the people: The role of humans in interactive machine learning. *AI Mag.* 35, 105–120 (2014). <https://doi.org/10.1609/aimag.v35i4.2513>.
93. Zhou, X.: Understanding Spatiotemporal Patterns of Biking Behavior by Analyzing Massive Bike Sharing Data in Chicago. *PLoS One.* 10, e0137922 (2015). <https://doi.org/10.1371/journal.pone.0137922>.
94. Hernandez, A., Resnick, M.L.: Placement of call to action buttons for higher website conversion and acquisition: An eye tracking study. *Proc. Hum. Factors Ergon. Soc.* 1042–1046 (2013). <https://doi.org/10.1177/1541931213571232>.

## **10. Appendix**

### **10.1. SCW 1 – Happy Scenario: New Model**

#### **1 - Define inputs to the walkthrough**

##### **1.1 - Identification of users**

City Municipality Workers, Urban Planners, Transportation Researchers, Shared Mobility Researchers.

##### **1.2 - Sample tasks evaluation**

Happy Scenario: New Model, New Model - Revisited, New Model - Complex, New Model - Complex Revisited, Loaded Model. User Wishes to Define a New Use of ML. User Defines A New Use of ML.

##### **1.3 - Action sequences for completing the tasks**

Happy Scenario: Create a model that Predicts the Best Bike-Station Placement

- 1 - Click on the machine learning icon
- 2 - Select/Write down the datasets needed
- 3 - Select the features needed from each selected dataset
- 4- Select the model
- 5 - Adjust the accuracy if necessary
- 6 - Select the duration of the historical data
- 7 - Select the days and hours of the historical data
- 8 - Select the desired prediction
- 9 - Review the accuracy of the model
- 10 - Train or Retrain the model
- 11 - Select the visualization method

##### **1.4 - Description or implementation of interface**

## Medium Fidelity Prototype

### 2 - Convene the walkthrough

#### 2.1 - Describe the goals of the walkthrough.

- Ensure that the tasks can be finished.
- Reveal usability problems.

#### 2.2 - Describe what will be done during the CW

- To visualize the results of the model predicting the best bike-station placement.

#### 2.3 - Describe what will not be done during the walkthrough

—

### 3 - Walkthrough the action sequences for each task

#### 3.1 - Tell a credible story for these two questions:

##### 3.1.1 - Will the user know what to do at this step

—

##### 3.1.2 - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?

—

#### 1 - Click on the machine learning icon

Q1: Yes, the users are aware that the tool contains machine learning capabilities. Furthermore, the controls of the tool are grouped in the panel.

Q2: Yes, upon clicking on the icon the tab will replace its contents.

## 2 - Select/Write down the datasets needed

Q1: No, the user has to have prior understanding of how machine learning works in order to discern that it requires datasets to function.

Q2: Yes, the interface will show that the dataset has been selected as it will be displayed under the box.

## 3 - Select the features needed from each selected dataset

Q1: No, they need to inform themselves from the information icon during their first time.

Q2: Yes, they are expected to continue to the panel below it.

## 4 - Select the model

Q1: Yes, they can click on the drop-down menu and select one of the available models according to the desired preference.

Q2: Yes, they are expected to continue to the panel below it.

## 5 - Adjust the accuracy if necessary

Q1: Yes, as the model selector and the accuracy are in succession of one another, the user is expected to know that this slider configures the model.

Q2: Yes, they are expected to continue to the panel below it.

## 6 - Select the duration of the historical data

Q1: Yes, the interface pre-fills the time in the "from" field to be 365 days before the current day of the system, and the "to" field to be one day before the current day of the system. Furthermore, there is an (i) icon which the user can hover over to learn about the field.

Q2: Yes, the user will see the selected day in the box.

## 7 - Select the days and hours of the historical data

Q1: No, the label is not clear, it should be renamed.

*Suggestion: Only include these weekdays; only include these hours from the days.*

Q2: Yes, the selected days will be populated into the box and the selected hourly periods will be populated in their box as well.

#### 8 - Select the desired prediction

Q1: Yes, they will choose the prediction that they need by clicking on the corresponding radio button.

Q2: Yes, the radio button will be displayed next to the selected prediction.

#### 9 - Review the accuracy of the model

Q1: Yes, they can see accuracy of the model (which should be 0%).

Q2: No, an initial message encouraging the user to train the model should be displayed.

#### 10 - Train or Retrain the model

Q1: Yes, the train model button (which is a call-to-action element) is colored differently from the other elements on the panel.

Q2: No, a status message/feedback mechanism must be added to indicate to the user that the system has indeed started training.

#### 11 - Select the visualization method

Q1: Yes, however, there is a chance that it won't be used as the call-to-action button precedes it.

Q2: Yes, the selected visualization method will be displayed in the drop-down menu.

### 4 - Record critical information

#### 4.1 - Possible learnability problems

- New users might not know which suitable machine learning model to use.
- Users will not know that when they train the model that it needs time to display its results.

- The users may be overwhelmed and/or confused with the sequence of steps needed to perform the prediction.

#### **4.2 - Design ideas**

- Preview an example visualization on the map as the user selects which prediction they desire OR when they hover over a visualization method.
- Add a feedback mechanism to the model training button when it is clicked.
- Associate an encouragement message to train the model when the accuracy is at 0% or when the model has just been created.

#### **4.3 - Design gaps**

- Disable the element "Current State" from the interface if the model has just been created by the user.

#### **4.4 - Problems in the task analysis**

- The "Current State" element in the prototype was not mentioned as it cannot be used in this particular task.
- The happy scenario does not specify the task itself.
- The task "train or retrain the model" cannot have a choice in the happy scenario.

### **5 - Revise the interface to fix the problems**

## **10.2. SCW 2 – Happy Scenario: New Model – Revisited**

### **1 - Define inputs to the walkthrough**

#### **1.1 - Identification of users**

City Municipality Workers, Urban Planners, Transportation Researchers, Shared Mobility Researchers.

#### **1.2 - Sample tasks evaluation**

Happy Scenario: New Model, New Model - Revisited, New Model - Complex, New Model - Complex Revisited, Loaded Model. User Wishes to Define a New Use of ML. User Defines A New Use of ML.

### **1.3 - Action sequences for completing the tasks**

Happy Scenario: Create a model that Predicts Hotspots Using Data gathered from the last year.

- 1 - Click on the machine learning icon
- 2 - Select the desired prediction
- 3 - Review the accuracy of the model
- 4- Select the duration of the historical data
- 5 - Select the days and hours of the historical data
- 6 - Select the model
- 7 - Adjust the accuracy if necessary
- 8 - Select/Write down the datasets needed
- 9 - Select the features needed from each selected dataset
- 10 - Select the visualization method
- 11 - Train or Retrain the model

### **1.4 - Description or implementation of interface**

Medium Fidelity Prototype

## **2 - Convene the walkthrough**

### **2.1 - Describe the goals of the walkthrough.**

- Ensure that the tasks can be finished.
- Reveal usability problems.

### **2.2 - Describe what will be done during the CW**



To visualize the results of the model predicting the best bike-station placement.

### **2.3 - Describe what will not be done during the walkthrough**

—

## **3 - Walkthrough the action sequences for each task**

### **3.1 - Tell a credible story for these two questions:**

#### **3.1.1 - Will the user know what to do at this step**

—

#### **3.1.2 - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?**

—

#### **1 - Click on the machine learning icon**

Q1: Yes, the users are aware that the tool contains machine learning capabilities. Furthermore, the controls of the tool are grouped in the panel.

Q2: Yes, upon clicking on the icon the tab will replace its contents.

#### **2 - Select the desired prediction**

Q1: Yes, the predict label has several radio buttons beneath it which the user must one from, the predict hotspots is the fourth option.

Q2: Yes, they will see that there are other menus beneath the prediction which need to be filled.

#### **3 - Review the accuracy of the model**

Q1: Yes, the revised label of the panel is called "state", the accuracy is at 0% and it has an accompanying warning label telling the user that it hasn't been trained yet.

Q2: Yes, the user moves to the next panel.

4 - Select the duration of the historical data

Q1: Yes, they have an example in the form of a "place holder" in the field below the "From" field. Same applies to the "To" field.

Q2: Yes, the entered time for both fields will be set in the field.

5 - Select the days and hours of the historical data

Q1: Yes, the default value is set to "All Weekdays", therefore, for this task, the user doesn't have to input anything.

Q2: Yes, the field communicates the value and the user follows the below fields.

6 - Select the model

Q1: Yes, they can click on the drop down menu and select one of the available models according to the desired preference.

Q2: Yes, the selected model will appear in the field and they progress to the accuracy slider.

7 - Adjust the accuracy if necessary

Q1: Yes, there is a slider that can be moved to 2 other positions.

Q2: Yes, the slider will be set to the new notch.

8 - Select/Write down the datasets needed

Q1: No, the user has to have prior understanding of how machine learning works in order to discern that it requires datasets to function.

*Either the information tooltip should explain why a new user needs to select a dataset, or there should be an onboarding process.*

Q2: Yes, the interface will show that the dataset has been selected as it will be displayed under the box.

9 - Select the features needed from each selected dataset

Q1: No, they need to inform themselves from the information icon during their first time.

*Either the information tooltip should explain why a new user can include or exclude features, or there should be an onboarding process.*

Q2: Yes, they are expected to continue to the panel below it.

#### 10 - Select the visualization method

Q1: Yes, they can select the desired visualization method by selecting one of the available methods in the drop-down menu.

Q2: Yes, the desired method will be selected.

#### 11 - Train the model

Q1: Yes, the call-to-action button is placed at the bottom and it has a distinctive color enticing the user to click it.

Q2: Yes, first, they have been informed by the state panel that the model needs to be 'trained', also, the button will become disabled once the user clicks on the train button. Furthermore, a message will appear under the buttons stating that the model is being trained and its progress can be seen in the model browser window.

### 4 - Record critical information

#### 4.1 - Possible learnability problems

- New users might not know which suitable machine learning model to use.
- Either the information tooltip should explain why a new user needs to select a dataset, or there should be an onboarding process.

#### 4.2 - Design ideas

- Preview an example visualization on the map as the user selects which prediction they desire OR when they hover over a visualization method.
- Either the information tooltip should explain why a new user needs to select a dataset, or there should be an onboarding process.

#### 4.3 - Design gaps

---

#### **4.4 - Problems in the task analysis**

---

### **5 - Revise the interface to fix the problems**

## **10.3. SCW 3 - Happy Scenario – Complex**

### **1 - Define inputs to the walkthrough**

#### **1.1 - Identification of users**

City Municipality Workers, Urban Planners, Transportation Researchers, Shared Mobility Researchers.

#### **1.2 - Sample tasks evaluation**

Happy Scenario: New Model, New Model - Revisited, New Model - Complex, New Model - Complex Revisited, Loaded Model. User Wishes to Define a New Use of ML. User Defines A New Use of ML.

#### **1.3 - Action sequences for completing the tasks**

Happy Scenario: Create a model and visualize it as a cluster - with set parameters to "1" - that predicts hotspots using the shared mobility and weather humidity data that was gathered during the weekend rush hours from the past 3 years.

1 - Click on the machine learning icon

2 - Select the desired prediction

3 - Review the accuracy of the model

4- Select the duration of the historical data (01/01/2018 - 31/12/2021)

5 - Select the days and hours of the historical data ({Saturdays, Sundays} {8:00 - 10:00 | 17:00 - 19:00})

- 6 - Select the model
- 7 - Expand the accuracy (Click on Chevron)
- 8 - Set Parameters to 1.
- 9 - Select/Write down the datasets needed (Weather and SMD)
- 10 - Only choose the humidity feature from the weather dataset.
- 11 - Set the visualization method to cluster
- 12 - Train the model

#### **1.4 - Description or implementation of interface**

Medium Fidelity Prototype

### **2 - Convene the walkthrough**

#### **2.1 - Describe the goals of the walkthrough.**

- Ensure that the tasks can be finished.
- Reveal usability problems.

#### **2.2 - Describe what will be done during the CW**

- To visualize the results of the complex task on the map.

#### **2.3 - Describe what will not be done during the walkthrough**

—

### **3 - Walkthrough the action sequences for each task**

#### **3.1 - Tell a credible story for these two questions:**

##### **3.1.1 - Will the user know what to do at this step**

—

### **3.1.2 - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?**

—

#### **1 - Click on the machine learning icon**

Q1: Yes, the users are aware that the tool contains machine learning capabilities. Furthermore, the controls of the tool are grouped in the panel.

Q2: Yes, upon clicking on the icon the tab will replace its contents.

#### **2 - Select the desired prediction**

Q1: Yes, the predict label has several radio buttons beneath it which the user must one from, the predict hotspots is the fourth option .

Q2: Yes, they will see that there are other menus beneath the prediction which need to be filled.

#### **3 - Review the accuracy of the model**

Q1: Yes, the revised label of the panel is called "state", the accuracy is at 0% and it has an accompanying warning label telling the user that it hasn't been trained yet.

Q2: Yes, the user moves to the next panel.

#### **4 - Select the duration of the historical data (01/01/2018 - 31/12/2021)**

Q1: Yes, they have an example in the form of a "place holder" in the field below the "From" field. Same applies to the "To" field.

Q2: Yes, the entered time for both fields will be set in the field.

#### **5 - Select the days and hours of the historical data ( {Saturdays, Sundays} {8:00 - 10:00 | 17:00 - 19:00} )**

Q1: Yes, the default value is set to "All Weekdays", therefore, for this task, the user clicks on the field and inputs "Saturdays and Sundays".

Q2: Yes, the field communicates the value and the user follows the below fields.

#### 6 - Select the model

Q1: Yes, they can click on the drop down menu and select one of the available models according to the desired preference.

Q2: Yes, the selected model will appear in the field and they progress to the accuracy slider.

#### 7 - Expand the accuracy (Click on Chevron)

Q1: Yes, as the user specifically aims to expand a menu and the chevron icon is a universal icon that is known to expand menus.

*The label "Accuracy" can be changed to a more specific label.*

Q2: Yes, advanced contents of the accuracy of the model will be displayed.

#### 8 - Set Model Parameters to 1

Q1: Yes, the user recognizes that the fields are editable from the slider and the actual value shown next to it..

Q2: Yes, as the user adjusts the slider a revert value icon will appear, which informs the user that the value has indeed been changed.

#### 9 - Select/Write down the datasets needed (Weather and SMD)

Q1: No, the user has to have prior understanding of how machine learning works in order to discern that it requires datasets to function.

*Either the information tooltip should explain why a new user can include or exclude features, or there should be an onboarding process.*

Q2: Yes, the interface will show that the dataset has been selected as it will be displayed under the box.

#### 10 - Only choose the humidity feature from the weather dataset

Q1: Yes, the current interface follows the Kepler.GL design guidelines, the three vertically stacked squares indicate that the user can click to get more details about the selected dataset.

*When it is a new model, always expand the selected dataset.*

Q2: Yes, once the user ticks the checkbox next to the humidity feature it will be filled with a white checkmark on a blue background to indicate that it has been selected.

#### 11 - Set the visualization method to cluster

Q1: Yes, as the user sees that the drop-down menu is editable and it has a label called "Visualize As" before it, it should be clear. Furthermore, based on the previous Cognitive Walkthrough, as the user hovers over the cluster option, an example visualization appears on the map.

Q2: Yes, once the user clicks on the drop-down menu and selects the cluster method, the original visualization method "Point" will be changed to "Cluster".

#### 12 - Train the model

Q1: Yes, the call to action button is placed at the bottom and it has a distinctive color enticing the user to click it.

Q2: Yes, first, they have been informed by the state panel that the model needs to be 'trained', also, the button will become disabled once the user clicks on the train button. Furthermore, a message will appear under the buttons stating that the model is being trained and its progress can be seen in the model browser window.

### 4 - Record critical information

#### 4.1 - Possible learnability problems

- New users might not know which suitable machine learning model to use.
- Either the information tooltip should explain why a new user needs to select a dataset, or there should be an onboarding process.
- Either the information tooltip should explain why a new user can include or exclude features, or there should be an onboarding process.

#### 4.2 - Design ideas

- The label "Accuracy" can be changed to a more specific label
- When the user wishes to create a new model, always display the selected dataset's features.



- Abstractify the Prediction label and the panel containing it, as machine learning can also classify and cluster.
- Provide a new icon (graduation cap) that onboards the users on the dataset selection and their features.
- Use presets in the model configuration UI and inform the user about recommended models.

#### **4.3 - Design gaps**

- As a consequence of the use of ML to cluster data, an interface for city level (districts) is required.

#### **4.4 - Problems in the task analysis**

### **5 - Revise the interface to fix the problems**

## **10.4. SCW 4 – Happy Scenario – Complex – Revisited**

### **1 - Define inputs to the walkthrough**

#### **1.1 - Identification of users**

City Municipality Workers, Urban Planners, Transportation Researchers, Shared Mobility Researchers.

#### **1.2 - Sample tasks evaluation**

Happy Scenario: New Model, New Model - Revisited, New Model - Complex, New Model - Complex Revisited, Loaded Model. User Wishes to Define a New Use of ML. User Defines A New Use of ML.

#### **1.3 - Action sequences for completing the tasks**

Happy Scenario: Create a model and visualize it as a cluster - with set parameters to "1" - that predicts hotspots using the shared mobility and weather humidity data that was gathered during the weekend rush hours from the past 3 years.

1 - Click on the machine learning icon

- 2 - Select the desired prediction
- 3 - Review the accuracy of the model
- 4- Select the duration of the historical data (01/01/2018 - 31/12/2021)
- 5 - Select the days and hours of the historical data ( {Saturdays, Sundays} {8:00 - 10:00 | 17:00 - 19:00})
- 6 - Select the model
- 7 - Expand the performance parameters option (Click on Chevron)
- 8 - Set Parameters to 1.
- 9 - Select/Write down the datasets needed (Weather and SMD)
- 10 - Only choose the humidity feature from the weather dataset.
- 11 - Set the visualization method to cluster
- 12 - Train the model

#### **1.4 - Description or implementation of interface**

Medium Fidelity Prototype

## **2 - Convene the walkthrough**

### **2.1 - Describe the goals of the walkthrough.**

- Ensure that the tasks can be finished.
- Reveal usability problems.

### **2.2 - Describe what will be done during the CW**

To visualize the results of the complex task on the map.

### **2.3 - Describe what will not be done during the walkthrough**

—

### **3 - Walkthrough the action sequences for each task**

#### **3.1 - Tell a credible story for these two questions:**

##### **3.1.1 - Will the user know what to do at this step**

—

##### **3.1.2 - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?**

—

#### **1 - Click on the machine learning icon**

Q1: Yes, the users are aware that the tool contains machine learning capabilities. Furthermore, the controls of the tool are grouped in the panel.

Q2: Yes, upon clicking on the icon the tab will replace its contents.

#### **2 - Select the desired prediction**

Q1: Yes, the label "Use Machine Learning To" label has several radio buttons beneath it which the user must choose one from, the predict hotspots is the fourth option.

Q2: Yes, they will see that there are other menus beneath the prediction which need to be filled. Furthermore, the contents of the ML model will be changed to suggest a configuration of an ML model for the user.

#### **3 - Review the accuracy of the model**

Q1: Yes, the revised label of the panel is called "state", the accuracy is at 0% and it has an accompanying warning label telling the user that it hasn't been trained yet.

Q2: Yes, the user moves to the next panel.

#### **4 - Select the duration of the historical data (01/01/2018 - 31/12/2021)**

Q1: Yes, they have an example in the form of a "place holder" in the field below the "From" field. Same applies to the "To" field.

Q2: Yes, the entered time for both fields will be set in the field.

5 - Select the days and hours of the historical data ( {Saturdays, Sundays} {8:00 - 10:00 | 17:00 - 19:00})

Q1: Yes, the default value is set to "All Weekdays", therefore, for this task, the user clicks on the field and inputs "Saturdays and Sundays".

Q2: Yes, the field communicates the value and the user follows the below fields.

6 - Select the model

Q1: Yes, the revised UI as mentioned in task 2 explains that a message will appear informing the user that a preset has been set. In this task, they can click on the drop-down menu and select one of the available models according to the desired preference, if they don't know which model to use, they can click on the graduation cap icon to further inform themselves.

Q2: Yes, the selected model will appear in the field and they progress to the accuracy slider.

7 - Expand the performance parameters option (Click on Chevron)

Q1: Yes, as the user specifically aims to expand a menu and the chevron icon is a universal icon that is known to expand menus.

Q2: Yes, advanced contents of the accuracy of the model will be displayed.

8 - Set Model Parameters to 1

Q1: Yes, the user recognizes that the fields are editable from the slider and the actual value shown next to it.

Q2: Yes, as the user adjusts the slider a revert value icon will appear, which informs the user that the value has indeed been changed.

9 - Select/Write down the datasets needed (Weather and SMD)

Q1: Yes, the user has a graduation cap icon to inform themselves that datasets are needed for machine learning. This should inform the user on the necessity

of datasets for ML and the field matches the guidelines of Kepler.GL, therefore it should not be confusing for the user to know that the field is editable.

Q2: Yes, the interface will show that the dataset has been selected as it will be displayed under the box.

#### 10 - Only choose the humidity feature from the weather dataset

Q1: Yes, by performing the previous task, the dataset features are expanded by default.

Q2: Yes, once the user ticks the checkbox next to the humidity feature it will be filled with a white checkmark on a blue background to indicate that it has been selected.

#### 11 - Set the visualization method to cluster

Q1: Yes, as the user sees that the drop-down menu is editable and it has a label called "Visualize As" before it, it should be clear. Furthermore, based on the previous Cognitive Walkthrough, as the user hovers over the cluster option, an example visualization appears on the map.

Q2: Yes, once the user clicks on the drop-down menu and selects the cluster method, the original visualization method "Point" will be changed to "Cluster".

#### 12 - Train the model

Q1: Yes, the call to action button is placed at the bottom and it has a distinctive color enticing the user to click it.

Q2: Yes, first, they have been informed by the state panel that the model needs to be 'trained', also, the button will become disabled once the user clicks on the train button. Furthermore, a message will appear under the buttons stating that the model is being trained and its progress can be seen in the model browser window.

### 4 - Record critical information

#### 4.1 - Possible learnability problems

—

## **4.2 - Design ideas**

—

## **4.3 - Design gaps**

- As a consequence of the use of ML to cluster data, an interface for city level (districts) is required.

## **4.4 - Problems in the task analysis**

## **5 - Revise the interface to fix the problems**

## **10.5. SCW 5 – Loaded Model**

### **1 - Define inputs to the walkthrough**

#### **1.1 - Identification of users**

City Municipality Workers, Urban Planners, Transportation Researchers, Shared Mobility Researchers.

#### **1.2 - Sample tasks evaluation**

Happy Scenario: New Model, New Model - Revisited, New Model - Complex, New Model - Complex Revisited, Loaded Model. User Wishes to Define a New Use of ML. User Defines A New Use of ML.

#### **1.3 - Action sequences for completing the tasks**

Load a previously created model, retrain and visualize it as points.

1 - Click on the machine learning icon

2 - Click on Browse Created Models

3 - Select a model

4 - Review the accuracy of the model

5 - Set the visualization method to H3

6 - Retrain the model

#### **1.4 - Description or implementation of interface**

Medium Fidelity Prototype

### **2 - Convene the walkthrough**

#### **2.1 - Describe the goals of the walkthrough.**

- Ensure that the tasks can be finished.
- Reveal usability problems.

#### **2.2 - Describe what will be done during the CW**

To load a previously made model and to retrain it.

#### **2.3 - Describe what will not be done during the walkthrough**

—

### **3 - Walkthrough the action sequences for each task**

#### **3.1 - Tell a credible story for these two questions:**

##### **3.1.1 - Will the user know what to do at this step**

—

##### **3.1.2 - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?**

—

1 - Click on the machine learning icon

Q1: Yes, the users are aware that the tool contains machine learning capabilities. Furthermore, the controls of the tool are grouped in the panel.

Q2: Yes, upon clicking on the icon the tab will replace its contents.

## 2 - Click on Browse Created Models

Q1: Yes, the button text states that.

Q2: Yes, the model browser window will appear.

## 3 - Select a model

Q1: Yes, the canvas, the placement of the models, and the seeker on the bottom match similar functionalities that Kepler.GL already offers.

Q2: Yes, the model window will disappear and the machine learning tab will be populated with the selected model's parameters.

## 4 - Review the accuracy of the model

Q1: Yes, the revised label of the panel is called "state", the accuracy is at x% and it has an accompanying warning label to inform the user about overfitting the model if the retraining process will exceed the inflection point.

Q2: Yes, the user moves to the next panel.

## 5 - Set the visualization method to cluster

Q1: Yes, as the user sees that the drop-down menu is editable and it has a label called "Visualize As" before it, it should be clear. Furthermore, based on the previous Cognitive Walkthrough, as the user hovers over the H3 option, an example visualization appears on the map.

Q2: Yes, once the user clicks on the drop-down menu and selects the visualization method, the original visualization method "Point" will be changed to "H3".

## 6 - Retrain the model

Q1: Yes, the call-to-action button is placed at the bottom and it has a distinctive color enticing the user to click it.



Q2: Yes, as it is a previously made model, the previous "Train" button has been replaced with "Retrain" to further indicate to the user that this is not a new model. Furthermore, a message will appear under the buttons stating that the loaded model is being trained and its progress can be seen in the model browser window.

#### **4 - Record critical information**

##### **4.1 - Possible learnability problems**

—

##### **4.2 - Design ideas**

—

##### **4.3 - Design gaps**

- If the user does not wish to retrain, there should be a button that allows the user to only visualize the model.

##### **4.4 - Problems in the task analysis**

—

#### **5 - Revise the interface to fix the problems**

### **10.6. SCW 6 – User Wishes to Define A New Use of ML**

#### **1 - Define inputs to the walkthrough**

##### **1.1 - Identification of users**

City Municipality Workers, Urban Planners, Transportation Researchers, Shared Mobility Researchers.

##### **1.2 - Sample tasks evaluation**

Happy Scenario: New Model, New Model - Revisited, New Model - Complex, New Model - Complex Revisited, Loaded Model. User Wishes to Define a New Use of ML. User Defines A New Use of ML.

### **1.3 - Action sequences for completing the tasks**

Load a previously created model, retrain and visualize it as points

- 1 - Click on the machine learning icon
- 2 - Click on Browse Created Models
- 3 - Select a model
- 4 - Review the accuracy of the model
- 5 - Set the visualization method to H3
- 6 - Retrain the model

### **1.4 - Description or implementation of interface**

Medium Fidelity Prototype

## **2 - Convene the walkthrough**

### **2.1 - Describe the goals of the walkthrough.**

- Ensure that the tasks can be finished.
- Reveal usability problems.

### **2.2 - Describe what will be done during the CW**

To load a previously made model and to retrain it.

### **2.3 - Describe what will not be done during the walkthrough**

—

### 3 - Walkthrough the action sequences for each task

#### 3.1 - Tell a credible story for these two questions:

##### 3.1.1 - Will the user know what to do at this step

—

##### 3.1.2 - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?

—

#### 1 - Click on the machine learning icon

Q1: Yes, the users are aware that the tool contains machine learning capabilities. Furthermore, the controls of the tool are grouped in the panel.

Q2: Yes, upon clicking on the icon the tab will replace its contents.

#### 2 - Click on Browse Created Models

Q1: Yes, the button text states that.

Q2: Yes, the model browser window will appear.

#### 3 - Select a model

Q1: Yes, the canvas, the placement of the models, and the seeker on the bottom match similar functionalities that Kepler.GL already offers.

Q2: Yes, the model window will disappear and the machine learning tab will be populated with the selected model's parameters.

#### 4 - Review the accuracy of the model

Q1: Yes, the revised label of the panel is called "state", the accuracy is at x% and it has an accompanying warning label to inform the user about overfitting the model if the retraining process will exceed the inflection point.

Q2: Yes, the user moves to the next panel.

#### 5 - Set the visualization method to cluster

Q1: Yes, as the user sees that the drop-down menu is editable and it has a label called "Visualize As" before it, it should be clear. Furthermore, based on the previous Cognitive Walkthrough, as the user hovers over the H3 option, an example visualization appears on the map.

Q2: Yes, once the user clicks on the drop-down menu and selects the visualization method, the original visualization method "Point" will be changed to "H3".

#### 6 - Retrain the model

Q1: Yes, the call-to-action button is placed at the bottom and it has a distinctive color enticing the user to click it.

Q2: Yes, as it is a previously made model, the previous "Train" button has been replaced with "Retrain" to further indicate to the user that this is not a new model. Furthermore, a message will appear under the buttons stating that the loaded model is being trained and its progress can be seen in the model browser window.

### 4 - Record critical information

#### 4.1 - Possible learnability problems

—

#### 4.2 - Design ideas

—

#### 4.3 - Design gaps

- If the user does not wish to retrain, there should be a button that allows the user to only visualize the model.

#### 4.4 - Problems in the task analysis

—

## **5 - Revise the interface to fix the problems**

### **10.7. SCW 7 – User Defines A New Use of ML**

#### **1 - Define inputs to the walkthrough**

##### **1.1 - Identification of users**

City Municipality Workers, Urban Planners, Transportation Researchers, Shared Mobility Researchers.

##### **1.2 - Sample tasks evaluation**

Happy Scenario: New Model, New Model - Revisited, New Model - Complex, New Model - Complex Revisited, Loaded Model. User Wishes to Define a New Use of ML. User Defines A New Use of ML.

##### **1.3 - Action sequences for completing the tasks**

Load a previously created model, retrain and visualize it as points

- 1 - Click on the machine learning icon
- 2 - Click on Browse Created Models
- 3 - Select a model
- 4 - Review the accuracy of the model
- 5 - Set the visualization method to H3
- 6 - Retrain the model

##### **1.4 - Description or implementation of interface**

Medium Fidelity Prototype

#### **2 - Convene the walkthrough**

##### **2.1 - Describe the goals of the walkthrough.**

- Ensure that the tasks can be finished.
- Reveal usability problems.

**2.2 - Describe what will be done during the CW**

- To load a previously made model and to retrain it.

**2.3 - Describe what will not be done during the walkthrough**

—

**3 - Walkthrough the action sequences for each task****3.1 - Tell a credible story for these two questions:****3.1.1 - Will the user know what to do at this step.**

—

**3.1.2 - If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?**

—

**1 - Click on the machine learning icon**

Q1: Yes, the users are aware that the tool contains machine learning capabilities. Furthermore, the controls of the tool are grouped in the panel.

Q2: Yes, upon clicking on the icon the tab will replace its contents.

**2 - Click on Browse Created Models**

Q1: Yes, the button text states that.

Q2: Yes, the model browser window will appear.

**3 - Select a model**

Q1: Yes, the canvas, the placement of the models, and the seeker on the bottom match similar functionalities that Kepler.GL already offers.

Q2: Yes, the model window will disappear and the machine learning tab will be populated with the selected model's parameters.

#### 4 - Review the accuracy of the model

Q1: Yes, the revised label of the panel is called "state", the accuracy is at x% and it has an accompanying warning label to inform the user about overfitting the model if the retraining process will exceed the inflection point.

Q2: Yes, the user moves to the next panel.

#### 5 - Set the visualization method to cluster

Q1: Yes, as the user sees that the drop-down menu is editable and it has a label called "Visualize As" before it, it should be clear. Furthermore, based on the previous Cognitive Walkthrough, as the user hovers over the H3 option, an example visualization appears on the map.

Q2: Yes, once the user clicks on the drop-down menu and selects the visualization method, the original visualization method "Point" will be changed to "H3".

#### 6 - Retrain the model

Q1: Yes, the call to action button is placed at the bottom and it has a distinctive color enticing the user to click it.

Q2: Yes, as it is a previously made model, the previous "Train" button has been replaced with "Retrain" to further indicate to the user that this is not a new model. Furthermore, a message will appear under the buttons stating that the loaded model is being trained and its progress can be seen in the model browser window.

### 4 - Record critical information

#### 4.1 - Possible learnability problems

—

#### 4.2 - Design ideas

—

### **4.3 - Design gaps**

- If the user does not wish to retrain, there should be a button that allows the user to only visualize the model.

### **4.4 - Problems in the task analysis**

—

## **5 - Revise the interface to fix the problems**



**CONFIRMATION**

Hereby I confirm that I have composed the present thesis independently. I only have used the sources and means specified in this thesis. Especially from the internet, I only have used the denoted references. I have taken note of the section in the examination regulations concerning attempts to cheat.

I confirm that the electronic version of the thesis which I deliver is identical to the printed version with respect to the content. I agree that an electronic version of the thesis will be stored for purposes of inspection of plagiarism.

---

**(Date)**

---

**(Signature)**