ENS 491-492 – Graduation Project

Final Report

Project Title: Interactive Visualization of Human Flows and Movement

Group Members: Utku Akgüner, Ali Erencan Gerçek, Şevval Tufan, Fatih Öztank

Supervisor(s): Selim Balcısoy

Date: 29/05/2021

Sabancı Üniversitesi

1. **EXECUTIVE SUMMARY**

Project aims to visualize human flows and movement in Sabancı University using a map. Project mainly focuses on two problems: Visualization and classification of the student trajectories. In our project, we used router connection-based movement. Our data is from Sabanci University campus which was captured between November 2015 and February 2016. Technologies such as React.js and JavaScript were used in the graphical user interface side of the project and Python, Node.js, was used to process data, generate data, generate heatmap and classify trajectories.

Visualizing trajectories allows us to analyse movement patterns within Sabancı University campus. Those movement pattern analyses provide us useful information such as the hourly population density of buildings and areas in the Sabancı University. This information may allow commerce or student clubs to maximize student engagement by doing their events in such areas. Besides the commercial benefit of this information the trajectory results can provide with the changes in behaviours of a student such as attendance ratios to the classes, library or sports center usage within students and cafe preferences among the residents.

Unsupervised learning is planned to be used in classification using machine learning. We are planning to use and combine many models such as k-means and decision trees. Individual trajectories are aimed to be classified as such: Dorm student, non-dorm students, professors, staff members, etc. Classification of the trajectories will be shown using different colors according to their trajectory class in the web interface.

2. **PROBLEM STATEMENT**

Visualizing Data has been a staple in data analysis as it provides an easier way to represent data to users with the help of specified filters. From the visualized data we can see patterns of movement and companies can choose their investments in the right projects as it provides them the patterns that show regularities in data. While the visualized data can be given with methods such as heat map, graphs, charts, etc. Most of them can have problems when it comes to detailed visualization of complex whole data. In this project, we are aiming to come up with a solution to this problem. Our purpose is to provide a web-based means of visualization such that it can represent that complex whole data in a comprehensible way. To be able to do this, we are analyzing the trajectory of the human movements in Sabancı University campus. The reason behind that we choose a campus environment is because it can simulate many types of movement patterns in a limited area.

It’s been planned to develop a project that will provide detailed trajectory visualization of the devices which are connected to routers located in Sabancı University. This visualization will include filters for devices such as FASS, FMAN or FENS faculty members. There are no specific solutions for this problem and a solution to this problem is highly appreciated by many data analysts. Optimal pathways must be constructed to deliver the best project. As an example for visualization there are no previously developed packages which provide the specific visualization required in this project therefore custom visualization is required to be planned and created using React.js. There is also no specific machine learning model for filtering specific types of human movements therefore additional research and implementation is required.

2.1. **Objectives/Tasks**

Visualization Method Selection: Data will be analyzed and the most suitable visualization for the data type will be found. One of the main concerns is being able to apply the decisions to be made on the data understandably.

Mapping out: Finding ways to determine the trajectory of people in the data which allow us to find the traveled path of the device in data. We had to choose an algorithm to generate road traffic between nodes. The Shortest Path Algorithm, Dijkstra, is one of the algorithms created with a simple logic and is widely used today. The starting point of the Shortest Path Algorithm aims to reach the target in the shortest way from different nodes. For this purpose, it is used today for directing internet traffic. Dijkstra's Algorithm is to determine the path between two target nodes with the least effort. Distance values between nodes were determined, a starting point, point 0, was determined. The distances of the other nodes are calculated from the starting point, it has the smallest distance. This process continues until it reaches the last node. Finally, the program shows us the nodes with the shortest path.

Data Processing for Visualization: Process is mainly for finding ways to development of filtering data. Data simplification is the process whereby large and complex data is rendered usable. Complex data must be simplified before it can be analyzed. Real-life data has too many dimensions (attributes). As the size grows, the time and resources we need to spend in all processes from data cleaning to model building increase. For this case, feature extraction was performed in order to reduce dataset size with minimal information loss. Data size was reduced by 50%. It is done by removing connect_at, kbps and device_id attributes. This leads to faster initialization of trajectories and allows more trajectories to be simulated.

Data Processing for Classification and Heatmaps: The original data we had consisted of the connection places and times of agents. Before working with this data we had to process it to something more readable since plain data was inconvenient for analysis. Since their connection time and disconnection time was given in a form that suits datetime format we used datetime library in python to get connection times based on their locations for each agent in terms of minutes. By using this data, we generate daily distributions for each agent according to their trajectories throughout the day.

	FMAN	REKTORLUK	IC	FASS	FENS	SUNUM	UC1	UC2	CP
3	0.000000	0.000000	0.000000	0.037700	0.009425	0.740811	0.018850	0.058435	0.018850
5	0.010494	0.000000	0.000000	0.000000	0.762695	0.000000	0.092417	0.072783	0.003385
19	0.000000	0.000000	0.000000	0.000000	0.793423	0.000000	0.096121	0.042159	0.051433
27	0.856113	0.009833	0.016388	0.006555	0.006555	0.000000	0.037693	0.000000	0.010488
39	0.091660	0.000000	0.015026	0.352367	0.294515	0.000000	0.039068	0.061608	0.037566
...	...	...	...	...	...	...	...	...	...
5432	0.839568	0.000000	0.000000	0.000000	0.000000	0.000000	0.033250	0.101413	0.000000
5440	0.820616	0.000000	0.041499	0.000000	0.028112	0.000000	0.013387	0.026774	0.026774
5450	0.006406	0.000000	0.000000	0.850096	0.039078	0.000000	0.000000	0.032031	0.039718
5452	0.878049	0.000000	0.000000	0.000000	0.000000	0.000000	0.030488	0.000000	0.000000
5453	0.740291	0.000000	0.110437	0.026699	0.025485	0.000000	0.000000	0.060680	0.000000

Table 1. A portion of a data frame after we generated daily distribution of agents according to their locations on campus.

<u>Machine Learning</u>: Using a ML model for classifying the trajectories of people. We did various research about machine learning algorithms in order to classify trajectories. Some of the machine learning approaches we investigated are geometric deep learning approaches, which is a learning procedure heavily used on graph based systems. Another machine learning discipline we investigated is unsupervised learning which is applied when the data has no labels. Still, data has to be classified in a way that similar entries shall be in the same group. We also investigated other methods such as k-means clustering and long short term memory implementations for deep neural networks.

Heatmap: While we planned to show movement of the trajectories on the map, we also wanted for users to have an easier way of seeing the overall population density on a daily basis according to places in the campus and time of the day.

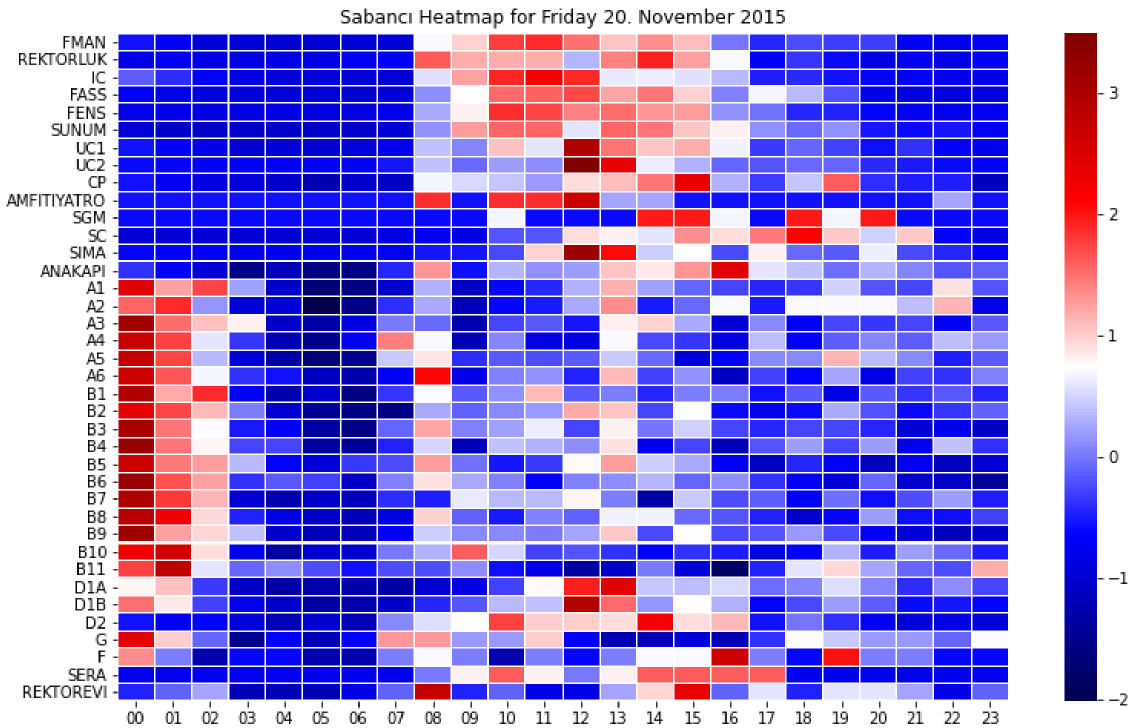

Figure 1. Heatmap of crowd densities according to time and place for 20 November 2015

API development: It is for setting up the connection between back-end and front-end.

Visualization: Making readable and detailed visualizations of the data within the final selected method.

Deployment: Finalizing the application and fixing the final in a comprehensible way.

2.2. **Realistic Constraints**

Economic: If the visualization is not correct, it can cost a lot of money for the company that bought it since it will lead to wrong investment hence, even though visualization is correct, if visualizations are not easy to understand, it will lead to wrong conclusions by stakeholders such that may cause dramatic losses for them. To prevent this issue we have manually checked the classifications to reaffirm the results.

Environmental: There shouldn't be any environmental problems as the data or visualization wouldn't harm the environment in any way.

Social: Confidentiality concerns regarding the use of personal data need to be carefully addressed, given the undeniability that individuals have no control over who, how and / or for what purposes, after the data is obtained.

Health and Safety: There can't be any problems with health and safety as there is nothing physical to interact with except the computer that the user is using.

Manufacturability: This Project may require high computing power if the given data set is too complex or too large to be worked on by commercial use computers

Sustainability: The information used in the decision-making process refers to the value that helps the success of the decisions made by transforming the data into a useful and meaningful form by subjecting the data to the information processing processes, and the informational raw materials that need to be processed before the data becomes meaningful and useful information. In this context, making strategic decisions with a data-based approach is of great importance for organizations to maintain their existence and to sustain their innovation-based competitive advantage.

3. METHODOLOGY

In order to use provided data on the web end, it was mandatory to turn data from xlsx format to json format. Json is short for javascript object notation which allows javascript to interact with data in a fast and efficient way. Javascript is the base programming language for the web end therefore we needed to convert trajectory data to json format. Process was done using node.js which is a framework allowing us to run javascript on a terminal to do any kind of tasks, in this case it was used to convert data format. It was used to split trajectory data into pieces day by day. All data was parsed according to separate days and each day data consisted of separate data. This process was mandatory due to hardship in loading all data to the web at once, therefore data was preferred to be loaded piece by piece. Also anomaly cases were removed which provides a better simulation. By this we could have simulated trajectories with a limit from 100 to max trajectory size. It has been aimed to allow any pc to simulate, for example: an old pc might have a hard time simulating 1000 trajectories at once therefore simulation consisting 100 or 250 trajectories can be simulated in an old pc by having a trajectory limit option.

Node.js was also used in order to group routers for the simulation. Simulation groups routers in terms of buildings, say Faculty of Natural Sciences has 50 routers but they are grouped as 1 group and simulated accordingly. This data was again turned into json format which is later used on the web app.

Web simulation runs on React.js which is another javascript framework which allows us to do asynchronous operations and dynamic rendering. Router data was imported to the app and rendered according to their locations which was accumulated using Google Maps. Leaflet.js which is a library was used to render the map which was the backbone of those router locations. A graph was created according to the roadways of the Sabancı University where campus residents were using on a day to day basis and again was plotted on the map.

In order to simulate trajectories it was necessary to implement a traveling algorithm, which in this case was dijkstra's algorithm. Dijkstra's algorithm is used in order to calculate the shortest path from point A to B. For every trajectory that was about to travel this algorithm was used and the result was cached in order to improve performance.

Simulation was carried out using time intervals. Every 10 minute in the simulation was equal to 10 seconds in real life allowing the client to see hours of trajectory movement under minutes. Every interval was set to be 10 seconds and every interval was splitted into another 18 period intervals. 18 period intervals were used in order to travel a trajectory from a node to another node just once. For example if A - B - C - D routes were to be traveled 18 periods would be splitted into 4 which would mean every 4.5 small intervals the trajectory would travel which is around 1 movement per 2.2 seconds. Usage of time intervals were really important in terms of performance since React.js uses a method called setState() which allows us to render elements on a web page dynamically. Every small interval program would call this method to update trajectory locations. But with no trajectory grouping formed under small intervals would allow us to update trajectory locations in a healthy way. Bad practise would be to call this method whenever a trajectory decides to change its place in its own function. For example for a 500 trajectory simulation we would only call setState() for 1 time where when trajectories are not grouped it would be needed to call for 500 times per 0.5 seconds.

Filtering options were provided to users such as trajectory limit, date, starting hour and trajectory class. Those filters would allow the user to create a simulation according to needs. Those filters can be used to analyze specific agents at a specific date and time.

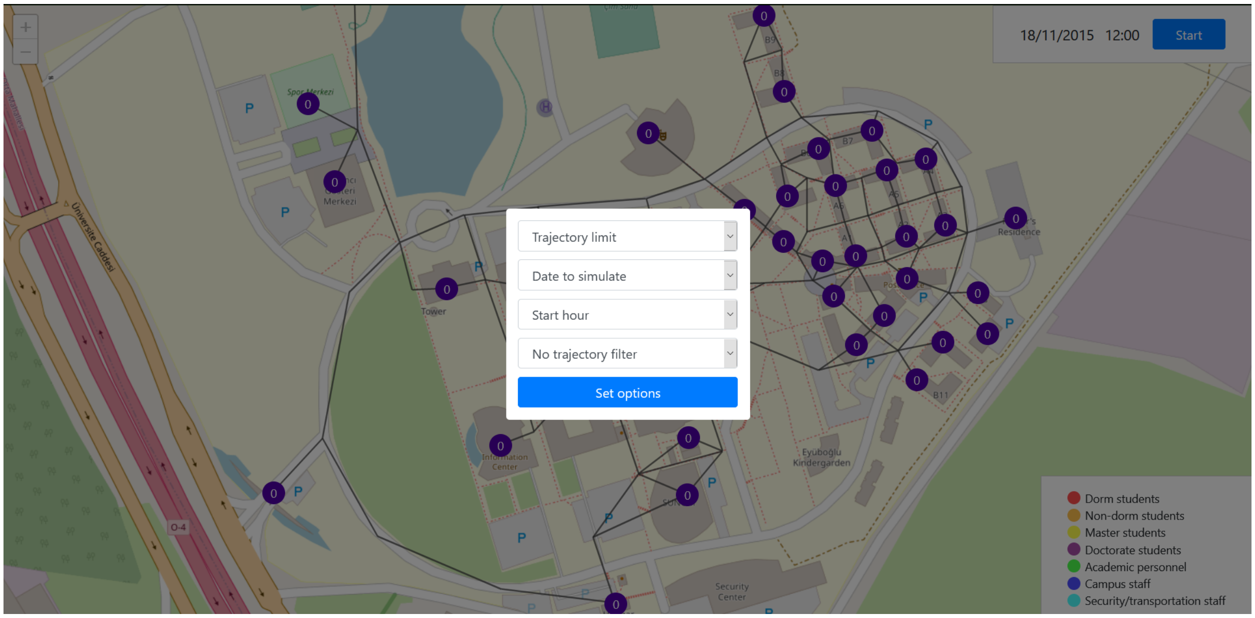

Figure 2. Option filtering for web app.

Creation of heatmaps was done by getting the trajectories of each agent and sampling each of those trajectories with one hour intervals. For generating this hourly trajectories, we created an algorithm to count the occurrences of each agent and map those occurrences to one of the nodes for each hour in a format similar to this: FMAN(12:00) -> UC(13:00) -> FENS(14:00)

After applying this algorithm to every agent in the dataframe we got a dictionary that contained hours as key and most visited place as value. By adding those dictionaries together, we acquired another dictionary for each hour of the day which holds unique places as keys and number of agents present at that location during a given time. With the use of the Seaborn library which is a library based on matplotlib such that it is used for visualizing data, in the end, we acquired daily heatmaps from 10 November 2015 to 9 February 2016.

For clustering this data we needed a dataframe that was parsable by the clustering algorithm. To achieve that we have worked with original data to create a dataframe that consisted of the times spent in each node in units of minutes. After that, a normalisation and elimination of outliers was done to the data for more precise clustering. During normalization, first we sum all the occurrences of specific agents during a specific day. After that we map that sum to 1 and according to this ratio, each occurrence in a day receives a normalised value between 0 and 1. For instance if an agent is present for 4 hours in our data on a specific day, and the distribution is like 2 hours for IC, 1 hour for UC and 1 hour for FMAN, then the normalised row will be in the format of 0.5 IC, 0.25 UC and 0.25 FMAN.

By doing this normalization, we overcome the possible inequality between trajectories which may occur due to their differences in their connection times. After the normalization, we also eliminated the outlier agents because they could manipulate the clustering results. By outlier, we mean agents whom show irregularities in their trajectories such as having very little or no change in their places or trajectories with patterns which are out of our interest. For detecting those outliers with stationary trajectories, we set a threshold of 90%. Which means any agent with more than normalized value of 0.9 will be labeled as stationary and dropped from the dataframe. For outliers which have trajectories other than our interest, we did a manual check after clustering and dropped the trajectories which most of their movements predominantly spread around locations such as gates, university center, service areas, doctorate dorms and rectorship building. Later, we mapped those trajectories into their own cluster and visualized those trajectories as well in our web interface.

At the end we had 39 columns for locations and this proved an issue as high dimensional data results in something called curse of dimensionality. This happens when the amount of dimensions is increased so much that it becomes infeasible for clustering algorithms to work with as increasing the dimension size results in the euclidean distance between agents to mean less and less and reducing the importance features therefore resulting in wrong clusters as it can’t put similar agents near each other. Therefore we tried reducing the dimension size by combining some columns such as time spent in dormitories to single column that had the total time spent. By doing this for every possible column we reduced 39 columns to 9 columns this showed promise as clusters started to appear more clear. Still 9 columns were affecting the clustering so we started searching for methods to overcome this issue. That's when we learned of the Principal Component Analysis(PCA) method.

Principal Component Analysis is a method to reduce the dimensionality of the data to an easily clusterable level such as 2 or 3 dimensions. It is often used for data visualization or clustering where the number of the features is too high for visualization or precise clustering. In this method, all features are analysed according to their impact on the variance in order to generate the requested number of features. For instance if we're reducing 10 dimensions to 2 dimensions, PCA will scan through the dimensions and combine those dimensions in 2 dimensions according to the impact on variance.

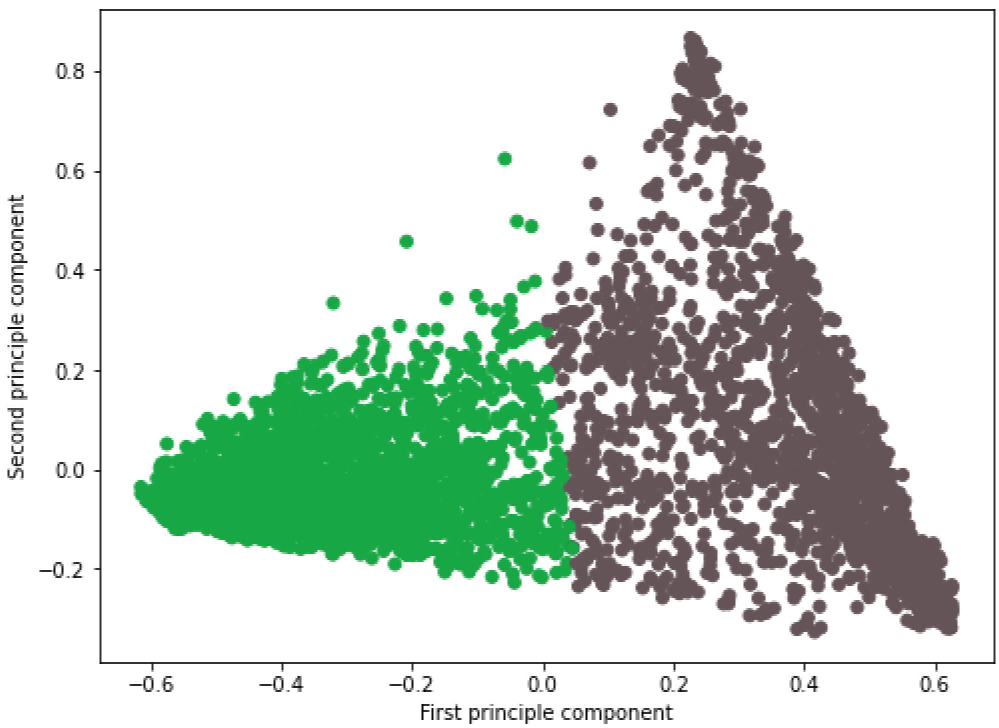

Figure 3. Our 9 dimensional data has reduced to 2 dimensions for visualizing and clustering.

After using PCA to reduce dimensions we used the elbow method to find optimal values for the amount of clusters. it gave us a range between 2-4 possible. After trying each one and checking the cluster values manually it's decided that two clusters were ideal as they separated almost perfectly compared to clustering with 3 and 4 clusters. With the clustering done we had 2 clusters. The first one was consistent with agents that almost never visited dorms and the second one was agents with consistent dorm entries. this allowed us to split the data in such way that we were able to identify students that were using the dorms and with the use of sub-clustering of non-dorm students we have identified these groups: Academic personal,non-dorm students, doctorate students, master students, rectorship building personnel, security and transport personnel, UC Staff. We have manually checked the correctness of these sub groups and verified its consistency.

4. RESULTS & DISCUSSION

Visualization demands were met at the end of the project which were demonstrating trajectory densities, trajectory flow, implementation of classification in a good visual manner, logical trajectory travel paths, demonstration by date and time. Initial objective was very clear and it was achieved by the hard working group individuals.

Trajectory densities were shown on the simulation by trajectory counts located on the routers. Every trajectory located on the routers were added up and shown accordingly. Also every router was colorized according to their densities compared to total trajectories that have been rendered.

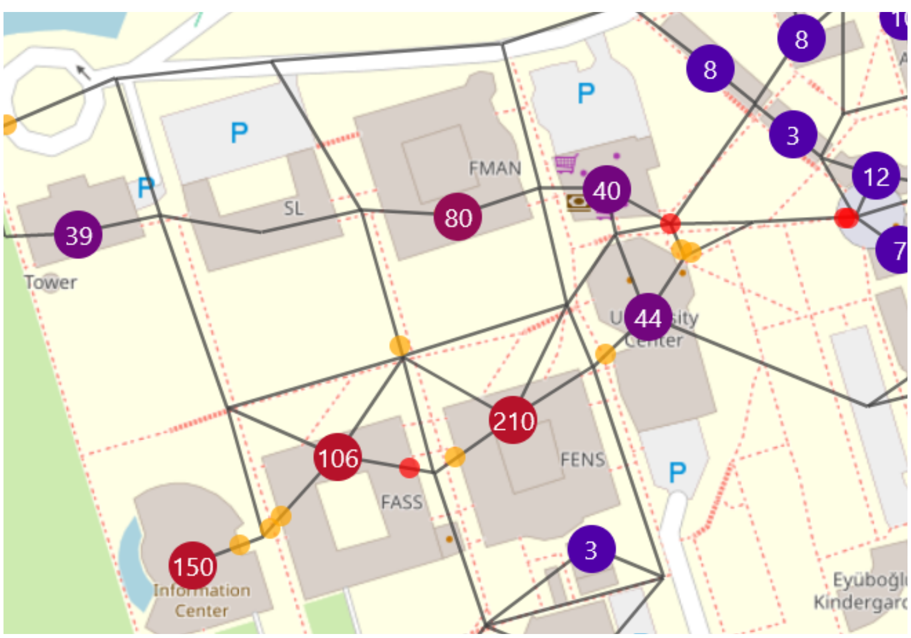

Figure 4. Density visualization of trajectories.

Trajectory flow was demonstrated using dijkstra's algorithm and colorizing trajectories according to their classes. Also travel time randomizing was added to travelling animation allowing separate trajectories to be seen even if the same path was being traveled.

Filtering options are satisfying in terms of the simulation considering less options were planned to be implemented at the beginning. Client was provided with 4 different filtering options which are: Date, starting time, classification and trajectory limit. Filters enable each client to run simulations with different specifications according to their interest.

The classification of the data was planned to perform by defining certain parameters where the agent has spent time. For that mean we have first tried to cluster the agents with k-means algorithm. Initial clustering was bad as we had many dimensions therefore causing mixed clusters where there were no difference between the clusters. We have handled the issue by reducing the dimensions with the use of the principal component analysis. With this method we clustered the data to 2 groups where one group has agents which have higher time spent on the dorms and the other group has agents which have higher time spent on the facilities. From this cluster, we labeled these groups as agents who use the dorms and possible agents who do not use the dorms.

Another clustering has been done to the agent group which we labeled as possible agents who do not use dorms. Reason behind this is that even though many agents in this group had minimal entries for dorms, some agents had dorm entries which could not be neglected. Those agents were labeled as possible masters students who use dorms because even though a master student uses dorms, still they spend most of their time at facilities in a stable trajectory. Rest of the agents in the non-dorm user group were divided into non-dorm user students and academic personnel based on the spread of their trajectories(A student would have a much more wide spread trajectory than an academic personnel because of their more versatile daily routine).

In the simulation it has been proved that data classification was correct. For example if trajectories were filtered as “Security/transport staff” nearly all trajectories would render near to the shuttle area and filtering other than “Dorm students” would render trajectories mostly out of dorms. “Academic personnel” was rendered around FENS, FASS and FMAN as intended, “Doctorate students” were located at F and G dorms at night time as expected.

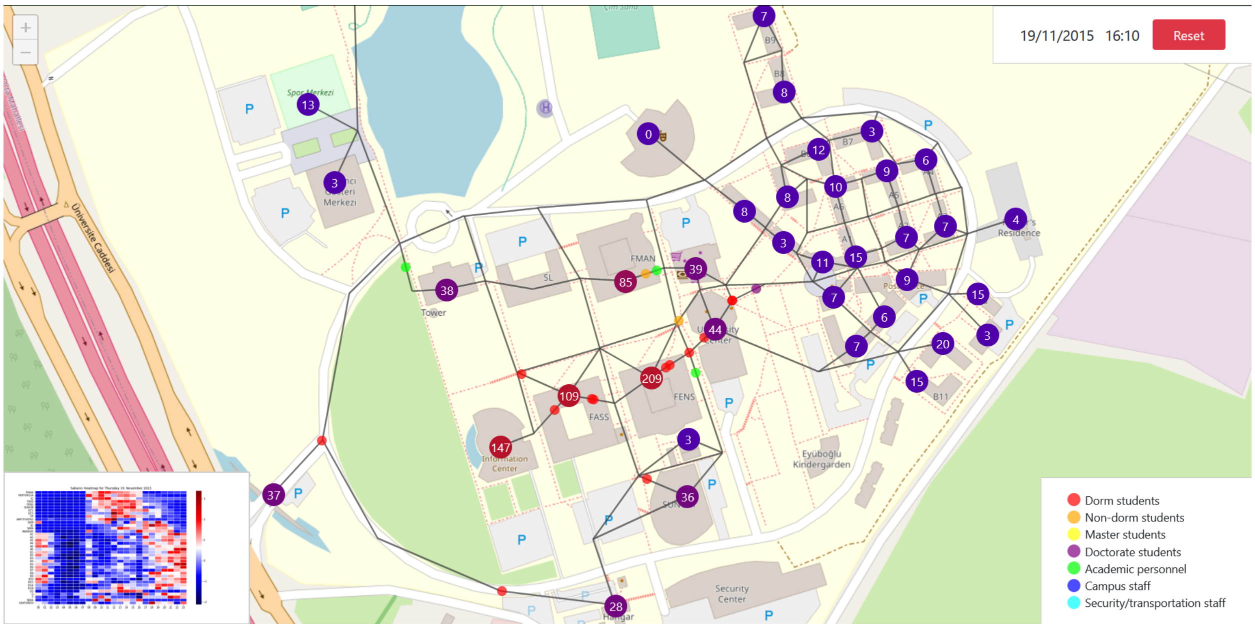

Figure 5. Web application final version.

Methods used in constructing this project might have different use cases in different projects. For example managers who want to increase buyer engagement with a brand in a mall that they own can use those methods in the same way. First of all data will be collected from the buyers who connect to shopping mall routers. Later a graph showing densities of the buyers will be plotted and the most buyer dense areas might be marked as an advertisement spot for the top sellers. Most likely this will increase engagement with the advertising brand, leading to more sales with that brand. Another method might be classifying buyer trajectories as low income buyer, middle income buyer and high income buyer. Those classifications might be used to select places to open top brand stores in areas where high income buyers are mostly located which will increase revenues of the top brand stores.

5. IMPACT

Data visualization is at the heart of analytics. When a data scientist is writing advanced predictive analytics or machine learning algorithms, it becomes important to monitor the results and visualize the outputs to make sure the models are performing as intended. This is because the visualizations of complex algorithms are often easier to interpret than digital outputs. Visuals make analysis easier and faster, while offering the ability to see important issues at a glance. What's more, most people respond much better to images than text. 90 percent of the information sent to the brain is visual, and the brain processes images 60,000 times faster than text. These considerations make it necessary to use data visualization to analyze and transport information.

Good data visualization is essential for analyzing data and making decisions based on that data. It enables people to quickly and easily see and understand relationships with structures, and to identify trends that cannot be noticed in tables consisting of only raw numbers. And in many cases no special training is required to interpret what is presented in the charts, universal understanding is provided.

Data Visualization in Brief; It is the process of converting the available raw data into easy and understandable image-photograph-graphics for fast, effective and correct decision making. It also has the power to see stories hidden between numbers and to trigger people to share and spread these stories.

Scope of this project does not include marketing, finance or sales. The data used specific to the project are those of Sabancı University that should not be shared. For this reason, with the data we work and use; we do studies and research on visualization of trajectories, machine learning on data, data processing, web development, API development and app deployment. However, the project may lead to social and technological developments within the boundaries of the school. By examining the structure of the school population, studies are carried out for the "classes" we have created. In particular, the opportunity to watch the density map of the students means activities and information that can be announced to more people. The study dynamics of the students can be analyzed with the map we have created, which can be enriched with additional data. In this way, a special system for students can be established. In addition, this point, where a software project is compatible with the subject of visual design, can be an example for preparing an interdisciplinary field of study.

6. ETHICAL ISSUES

Big data raises ethical issues. Big data analysis is to use large-scale data sets to define patterns in order to make economic, social, technical and legal claims. Big Data should not interfere with human will. Before making our decisions, who we are can be audited and determined with big data analytics. Companies or researchers should do a rigorous study of implications that should and should not be permitted.

Because the data being open to the public has some consequences such as researchers' use of user-derived content without permission. If asked, most users will not allow their content to be used elsewhere. Also, the vast majority of users are unaware of the algorithms that can store the content they produce. Therefore, in order for researchers to behave ethically, they must understand their accountability to research resources. Academicians receive approval from ethical committees while conducting human-based studies. However, many ethical committees do not understand that big data mining and anonymization processes can cause the problem of personally identifying data.

Data visualization ethics means things to be considered when data visualization is done. As a good designer, besides being able to visualize the data effectively, an ethical violation would have been created if the information was not presented or conveyed properly. Data visualization ethics draws attention to the need to do both at the same time.

7. PROJECT MANAGEMENT

Initial visualization technique for roadways was to plot two different roadways between the nodes in order to differentiate incoming and outgoing traffic but later it was scratched due to complexity it would cause to client, therefore only one roadway was plotted between nodes for incoming and outgoing traffic. It has been observed that simplifying the user interface is the best practise in visualizing complicated data.

Data was planned to be imported to be whole but due to performance related issues data was split into separate trajectories by separate days. Also importing techniques were changed from require() to fetch() function allowing data load to be pipelined which allows data to load faster and more efficiently in terms of dynamic memory. Pipelining data is found to be the better way to load data if there exist lots of separate data files in a project.

Heatmap was planned to be directly plotted on the map initially but later it was found out that it would be impossible to do due to dragging related issues on the map and image generation issues. Instead heatmap was shown as a minimap on the bottom left corner of the application. Also in order to visualize density, every router was colorized to indicate their population density compared to total population.

While processing the data for clustering and generating heat maps a problem related to dataset has become obvious. Some agents had anomalies in their trajectories such that those agents appeared at multiple locations at the same time. Such anomaly will result in wrong interpretation of data which will lead to wrong clusterizations and wrong visualizations. For solving this issue, disconnect time columns were rearranged in a manner that they will be consistent with the connection times.

	device_id	os	router_name	connect_at	disconnect_at	kbps
6189	23	iOS	Yurt_G_G03_0a:a4	2015-11-10 21:19:00	2015-11-10 21:29:00	0.00
6190	23	iOS	Yurt_G_G03_0a:a4	2015-11-10 21:29:00	2015-11-10 21:39:00	0.00
6191	23	iOS	Yurt_G_G03_0a:a4	2015-11-10 21:39:00	2015-11-10 22:10:00	0.41
6192	23	iOS	Yurt_G_G03_0a:a4	2015-11-10 22:10:00	2015-11-10 22:41:00	0.14
6193	23	iOS	Yurt_G_G03_0a:a4	2015-11-10 22:41:00	2015-11-11 12:13:00	199.50
6194	23	iOS	Yurt_G_G03_0a:a4	2015-11-11 12:13:00	2015-11-11 12:54:00	482.21
6195	23	iOS	Yurt_G_G03_0a:a4	2015-11-11 12:54:00	2015-11-12 00:25:00	10.22
6196	23	iOS	Yurt_Cafe2_d5:05	2015-11-11 13:26:00	2015-11-11 13:36:00	0.00
6197	23	iOS	FENS_L045_91:7d	2015-11-11 13:36:00	2015-11-11 21:07:00	0.40
6198	23	iOS	UC_akbank_eb:20	2015-11-11 21:07:00	2015-11-11 21:17:00	0.00

Table 2. An example of the disconnect inconsistency in the data. Between 12:54 and 00:25 agent with id 23 seem to be present at G dormitory but between 13:26 and 21:17 this agent seems to follow a regular trajectory pattern.

During the generation of heatmaps, another problem about the dataset itself has shown its effect. The problem was about a dramatic decrease in agent numbers during noon time. Especially around the facilities region, which is the region we expect to have highest agent density during noon time. To overcome this problem, a manual analysis applied to the data in order to find the source of the problem. After analyzing the data, we find that trajectories for midnight and noon were swapped. Upon considering this situation while creating heatmaps, a more precise heatmaps were generated.

Dynamic rendering was planned to be executed by every trajectory according to their own timeline, however synchronization and performance issues were encountered. As a result time intervals were brought up to the project, grouping all trajectory movements under time intervals. Synchronizing separate resources without causing performance issues was found out to be the most crucial component of a simulation, deciding whether a simulation is good or bad in terms of quality.

8. CONCLUSION AND FUTURE WORK

With this project, we had the opportunity to examine specific trajectory data. We have created various classes with classification with this data that follows the individuals at Sabancı University in motion with wi-fi connections. In this way, we have created a moving map with the agents belonging to certain classes with the colors selected according to their departments.

Inside our data, there are important insights that can help us move the project forward. The problem, however, is that it is difficult to connect the dots just by looking at the raw numbers. When we look at the data presented visually, we can see insights that will pave the way for models and connections.

Being able to see the story inside the numbers makes data visualization a powerful tool for information sharing and communication. Data visualization can be used to show performance, communicate trends, understand the impact of new strategies, show relationships, and more. These impressions add more value to all content where information sharing is required. It becomes a powerful tool for communication and collaboration. Data visualization is included in many business intelligence tools and is essential for advanced analytics. It helps people understand any information or data produced today.

As a limitation, it can be given that the data cannot be used outside the scope of the school. Only in this way, student or faculty member organizations to be established within Sabancı University can be optimized. In this way, we can create a more detailed solution project by narrowing our focus.

The moving map we have obtained can serve different purposes within various disciplines. In addition, with this experience, we can use data that is spread over large areas and test the results we have obtained with machine learning.

We have spoken with two other student groups which were going to continue from the same data set. One of the groups was planning to use this data in order to detect possible risky places during a pandemic. The other group was planning to use this data for grouping people based on their movements. Our project may help both of the groups in this endeavour as a starting or a guiding point since the data is already preprocessed.

9. **APPENDIX**

10. **REFERENCES**

Visualizing Highway Traffic as a Living Circulatory System. (2016, May 10). Retrieved January 16, 2021, from http://metrocosm.com/traffic-flow-maps/

Topology Density Map for Urban Data Visualization and Analysis (2020,July), from https://www.researchgate.net/publication/343390443_Topology_Density_Map_for_Urban_Data_Visualization_and_Analysis

Z. Zhou et al., "Visual Abstraction of Geographical Point Data with Spatial Autocorrelations," 2020 IEEE Conference on Visual Analytics Science and Technology (VAST), Salt Lake City, UT, USA, 2020, pp. 60-71, doi: 10.1109/VAST50239.2020.00011. , from https://ieeexplore.ieee.org/document/9308623

Geometric deep learning. (n.d.). Retrieved April 11, 2021, from http://geometricdeeplearning.com/

19 mobility network modeling. (n.d.). Retrieved April 11, 2021, from https://covid-mobility.stanford.edu//

Çelik, Sadullah & Akdamar, Emrah. (2018). BÜYÜK VERİ VE VERİ GÖRSELLEŞTİRME.

Veri Görselleştirme Nedir? Oracle Türkiye. (n.d.). https://www.oracle.com/tr/business-analytics/what-is-data-visualization/.

Bro, Rasmus & Smilde, Age. (2014). Principal component analysis. Analytical methods. 6. 2812. 10.1039/c3ay41907j.