Ana Flávia Vital - s344046
Haakon Bernhard Movig - s306458
Theo Alexander Sperre Olsen - s350233

DAVE3625 - Introduction to Artificial Intelligence

Potential uses of AI for Oslo Bysykkel

Final Project: Report on AI

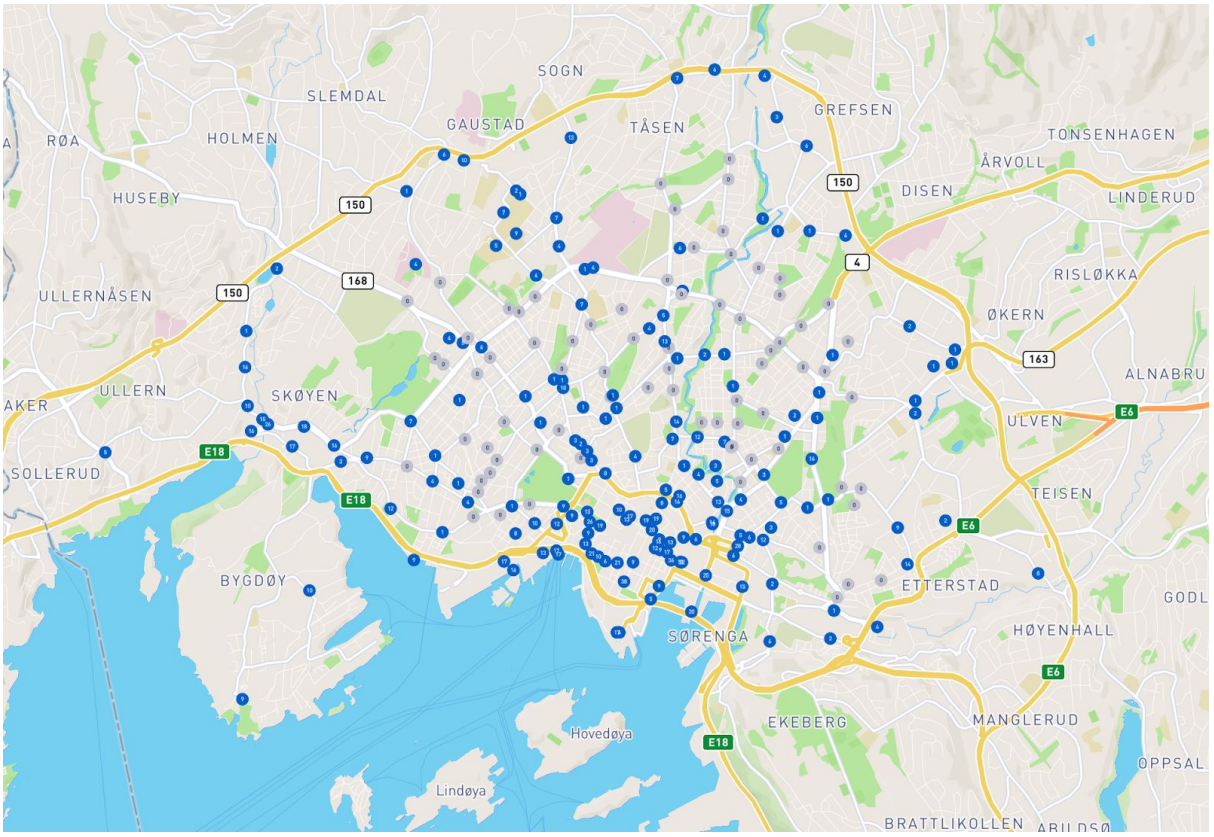
Implementation use case

Wordcount: 2924
(except sources, cover page and footnotes)

Introduction: the problem

Oslo Bysykkel is a shared bicycle service implemented within the Oslo municipality. The business model relies on making bicycles available between April and November of each year on stations spread throughout the city. Customers may pick up a bicycle at any one station and return it to either that same location or another preferred station, provided that parking spots are available.

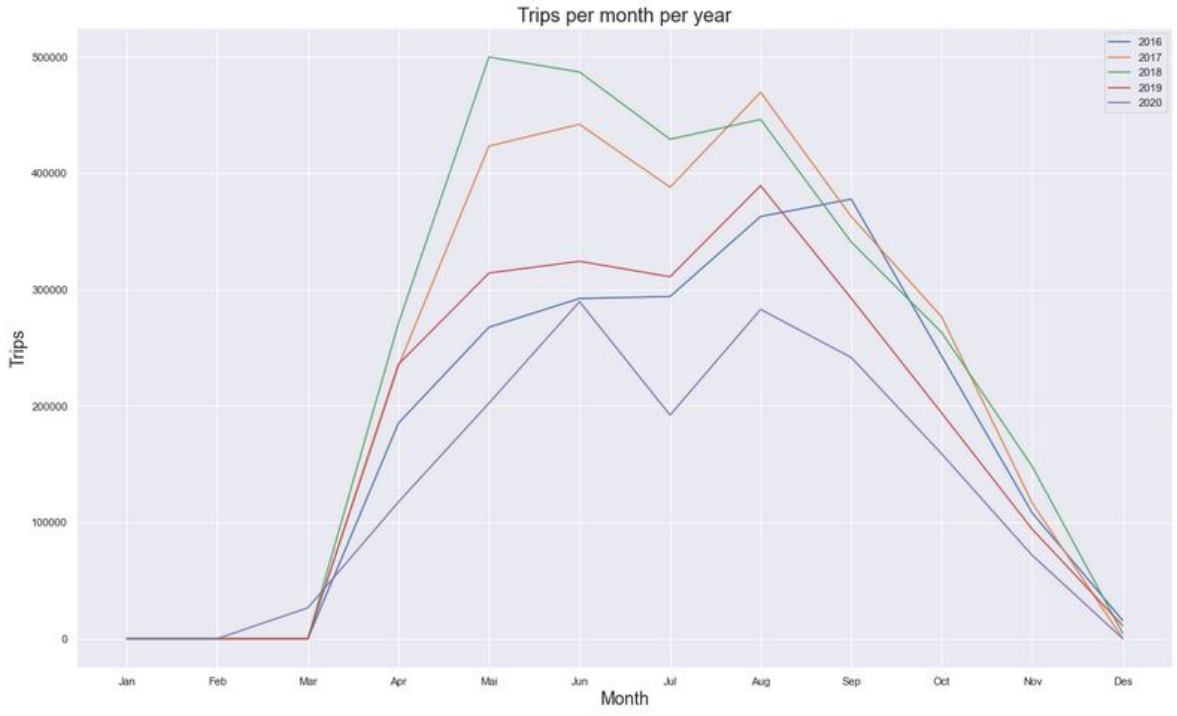
As of November 2020, the company has installed 247[1] stations with varying capacity throughout the Oslo municipality. Users of the bicycles may rent for 0 to 7 hours either under the subscription modality or by purchasing a day pass. (Oslo Bysykkel, n/a)



Bicycle stations as of November 2020[2]

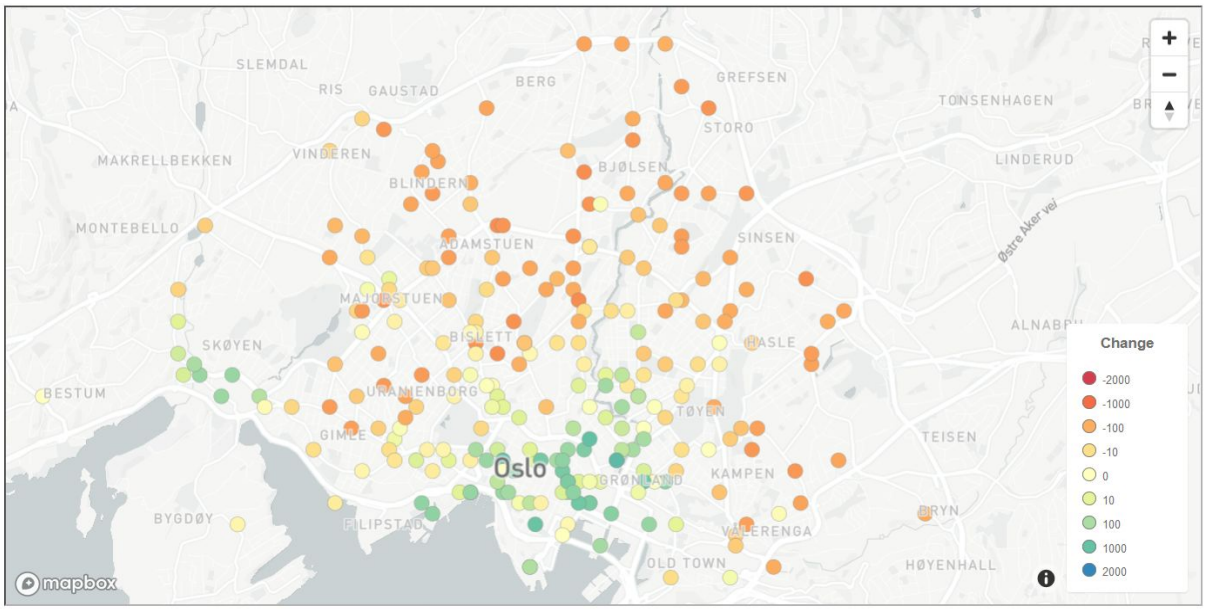
In Oslo, almost 100.000 users rely on Oslo Bysykkel for either daily commutes or sporadic trips. (Bergskaug, 2018) In addition, the company offers business-targeted services.[3] Due to the elevated number of users, and their usage pattern, Oslo Bysykkel faces a challenge: during peak time (07.00-09.00, and 15.00-17.00), entire areas of the city have their racks either brimming with bicycles or completely empty. This impacts commuters who are either unable to find a bicycle or a place where to park.

[1] https://gbfs.urbansharing.com/oslobysykkel.no/station_status.json
[2] Image retrieved from https://oslobysykkel.no/stasjoner on 24 November 2020, at 10:30
[3] https://oslobysykkel.no/bedrift



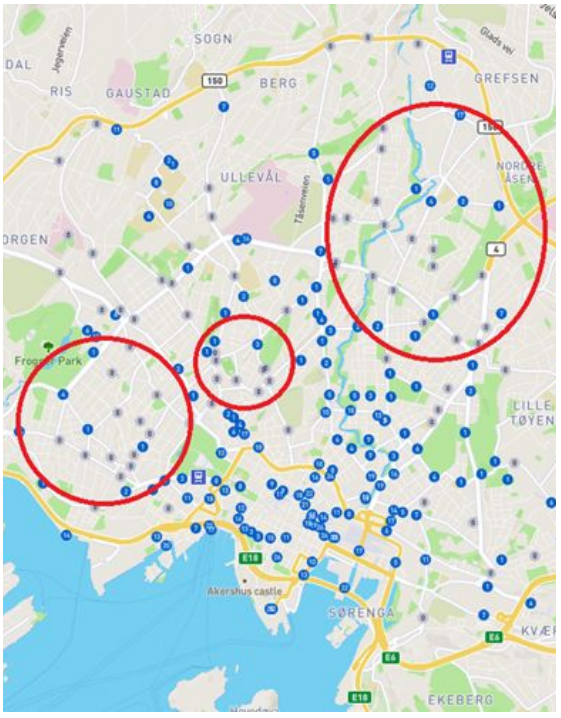
Where Oslo Bysykkel seems to fail is in handling several users at a specific time. Whereas daily commute to work causes the outskirts of Oslo to be drained from bikes, an exceptional public event will fill every rack nearby and a sunny day will congest the racks at the beaches.

Oslo's geography worsens the problem. Since the city center has lower elevation, most users are happy to cruise downwards, but refrain from cycling back up. With the considerable risk of finding racks empty when one most needs a bicycle, or finding no parking spots when in a rush to return a bike, Oslo Bysykkel may be viewed as an unreliable means of transportation to potential users.

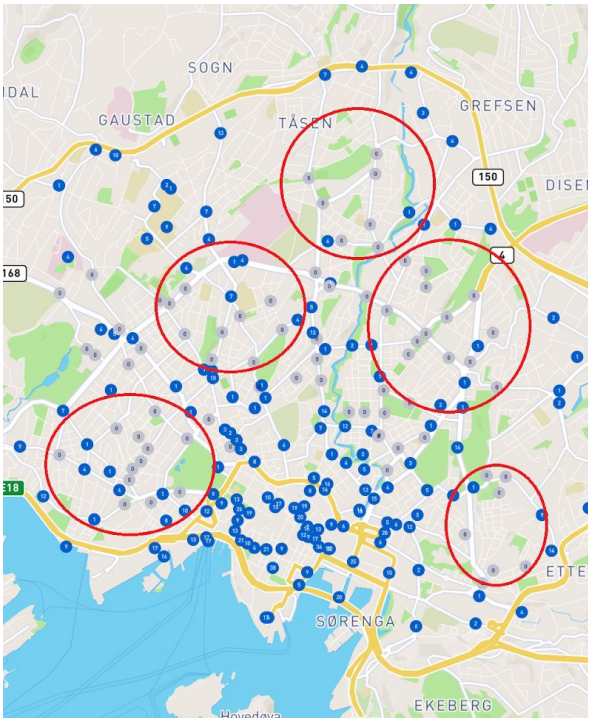


Cumulative movement of bicycles in correlation with user trips - September 2020

Oslo Bysykkel has acknowledged the issue on their own website[4] and has been making attempts to mitigate the problem with its team of bike redistribution drivers. According to their website, drivers "get notified as racks are about to get empty or being filled." (Oslo Bysykkel, n/a) Nonetheless, these actions have proven to be insufficient, as shown, first and foremost, by the fact that the company openly states it "cannot promise that there will be an available lock or bike near [the user] at all times"[5]. Secondly, through systematic consultation throughout the period of development of this project, we have consistently identified empty racks, very often throughout large areas, at different times of day and on different dates. This, we emphasise, in the context of COVID-19, where authorities have established that home-office should be the rule for all feasible situations (OSLO KOMMUNE, 2020), effectively reducing the number of individuals who need to commute daily. Finally, we drew conclusions from the group's personal experiences with the service, one of our members is a subscriber that is often affected by empty racks.



Empty racks on 6 November 2020, at 14:30[6]



Empty racks on 24 November 2020, at 10:30[7]

Proposal: a solution

The identified issues create an opportunity for improvement. Through the use of a machine learning model, Oslo Bysykkel could better predict bike usage, anticipate issues and make bike allocation decisions more precisely.

In order to build an adequate prediction tool, the machine learning model could be trained using past user data - some of which is already collected daily by Oslo Bysykkel. This would

[4] http://hjelp.oslobysykkel.no/en/articles/803739-full-and-empty-stations
[5] *ibid*
[6] Image retrieved from https://oslobysykkel.no/stasjoner on 6 November 2020, at 14:30
[7] Image retrieved from https://oslobysykkel.no/stasjoner on 24 November 2020, at 10:30

need to be combined with weather data, allowing for predictions that are congruent with human behaviour. Other data, such as events, local traffic, public transportation alerts and user-specific behaviour, could be incorporated in more advanced versions, as such factors may also greatly impact the flux of Oslo Bysykkel users.

Once trained, the machine learning model should be capable of making correct decisions on where to transport bicycles to. This information would be sent to the redistribution drivers, who would know in advance where to move bikes from and to. For further implementations of this solution, drivers could have a dedicated algorithm (perhaps Dijkstra's algorithm) tracing the optimal route.

By ensuring all its stations have bicycles and parking spots available, Oslo Bysykkel would have better effective coverage of the areas of the city where its stations are already present. Furthermore, a solution reliant on a machine learning model could help drivers know with more precision which stations need attention, as well as allow the company to optimise its resource use, as a better distribution of bicycles in stations could prevent:

a. Buying more bicycles to supply for customer need;
b. Building more stations in areas where there is high demand in specific times;
c. Expanding already existing stations to fit more bicycles.

Additionally, customers would benefit from a more reliable service, which could generate more customer satisfaction and loyalty, as well as result in an increase in the number of rides. Finally, the company would provide a better structure for accessible, friendly transportation, benefiting the community and the environment.

If left untreated, this problem might lead to an increase in frustration for customers, leading them to use alternatives such as electric scooters or Ruter services. What is more, Oslo Bysykkel might incur in significant expenses in an attempt to mitigate the suboptimal distribution of bicycles, such as buying more bikes or building and expanding stations.

We argue, therefore, that this is a valuable solution to Oslo Bysykkel, especially in the context of new and growing competitors in the market. Furthermore, as of this project, the presented solution is a business improvement opportunity rather than an business opportunity, once the algorithm has been tailored to Oslo Bysykkel's model of data collection. It could, however, be generified and expanded, as several companies around the world provide similar services.

Resources: the data

Oslo Bysykkel has made their datasets publicly available[8], having published information about all trips of over 1 minute made by users in both CSV and JSON formats. In 2019 the company changed what data they collected and, as a result, the pre-2019 datasets lack some of the features that the newer sets have. Basic data is still present in the legacy datasets: start of trip, end of trip, start station id and end station id. Therefore, features such

[8] https://oslobysykkel.no/apne-data/historisk

as trip length can be calculated with certainty, if the station IDs have not changed. It is also possible to retrieve all other station-related data by fetching it through the station ID, from the newer datasets. (Oslo Bysykkel, n/a)

The data provided in the datasets is structured, clean and complete, but unlabeled (no definition *e.g.* of what is a "station that needs to be filled") although it doesn't contain any information on bikes that might eventually not have been returned. This has most likely been cleaned from the original dataset before it was made public, but should be available in a complete dataset retrieved directly from Oslo Bysykkel.

Collection of data is done by the parking racks, which log the movement of any given bicycle. Every bike has a unique ID which allows it to be tracked from start to end point, as well as the time for which it was used. Furthermore, every station has a unique ID, which is connected to longitude and latitude coordinates.

The dataset does not contain information about conditions of each station (*i.e.* how many bikes are available at the start of the day or when a certain bike was picked up). This data is ultimately what will be used for the model. Therefore, it is necessary to obtain station-specific data - by either directly collecting it, *e.g.* through Oslo Bysykkel's API[9], or retrieving it from existing data about trips.

Furthermore, as is known that the weather has a direct influence on human behaviour(Hem & Iversen, 2019), this is the first set of external data that will be inserted into the model. The dataset will be enriched with historical weather data and weather forecast obtained from the Meteorologisk Institutt's API. (Meteorologisk Institutt, n/a)

Tools: the features

Once established that we need a station-based dataset, the first step is to fetch the available data from Oslo Bysykkel's API, which contains:

1. station_id
2. is_installed
3. is_renting
4. is_returning
5. last_reported
6. num_bikes_available
7. num_docks_available

The station ID (1) is the key feature, as all data will be linked to one station through it. Items 2 to 4 can be condensed into a single feature that determines whether the station is fully functional (and excludes it otherwise from the process). Finally, items 6 and 7 can be directly turned into features indicating bikes left and parking spots respectively, as well as be combined to create a new feature with the station's capacity.

[9] Oslo Bysykkel provides an open API at https://oslobysykkel.no/apne-data/sanntid, which allows tracking of absolute and current capacity of racks.

From here, station data and trip data come together: for each day, the initial number of bikes at the station is set according to the value collected from the API and, every time a bike is removed or parked, a new line - updating the station status after the transaction - is added to the dataset that will be used by the prediction model. Other information like weather forecast, actual weather, etc. can also be added to the dataframe.

	station_id	capacity	bikes_at_station	timestamp	status	Delivered	data1	data..n
0	1	10	5	2020-08-01 08:51:09.122000+00:00	0.50	0	sample	sample
1	1	10	6	2020-08-01 08:54:13.898000+00:00	0.60	1	sample	sample
2	3	16	4	2020-08-01 08:54:14.878000+00:00	0.25	0	sample	sample

Example of features listed in dataframe

Features from the trip datasets[10] will be used to create the updates. These datasets contain:

1. started_at (Timestamp)
2. ended_at (Timestamp)
3. duration (Integer)
4. start_station_id (String)
5. start_station_name (String)
6. start_station_description (String)
7. start_station_latitude (Desimalgrader i WGS84)
8. start_station_longitude (Desimalgrader i WGS84)
9. end_station_id (String)
10. end_station_name (String)
11. end_station_description (String)
12. end_station_latitude (Desimalgrader i WGS84)
13. end_station_longitude (Desimalgrader i WGS84)

Items 1 and 2 are useful for the registry of events (delivery/rental), as well as generating the features of timestamp itself, day of the week, time of day (morning/afternoon/evening/night) and time of year (roughly, season).

As of the initial version of the algorithm, duration (3) should not be turned into a feature. However, for further versions, it may be used to calculate the probability of a bicycle, having departed from station A, to arrive at station B.[11]

For the station status update, it is, of course, essential to take into account the station where a given trip starts or ends (4 and 9). Information such as the name of a station (5 and 10) and its description (6 and 11), however, would be of little use as a feature, once they are directly connected to and dependent on the station's ID number.

[10] https://oslobysykkel.no/apne-data/historisk

[11] By considering the average duration of a trip between A and B, and comparing it with the current duration of the ongoing trip, the probability of that bicycle - as well as any other bicycle in active use - going to an emptying or filling rack B could be used to determine on short notice whether station B should be prioritised in refilling/emptying, for example. This calculation could be especially useful for frequent users' journeys. This, however, would introduce a new level of complexity to the algorithm, as well as require user-related features, which may be undesidered and/or unnecessary.

On the other hand, the latitude (7 and 12) and longitude of a station (8 and 13), although also directly connected to the station's ID number, may be crucial features when taking into account the weather - and, in further versions, events in the area, traffic, public transportation issues and other data. What is more, if clustering is used - as we argue would be advisable - these two features become indispensable. For the older datasets, where latitude and longitude are missing, these missing features can be recovered through the station ID.

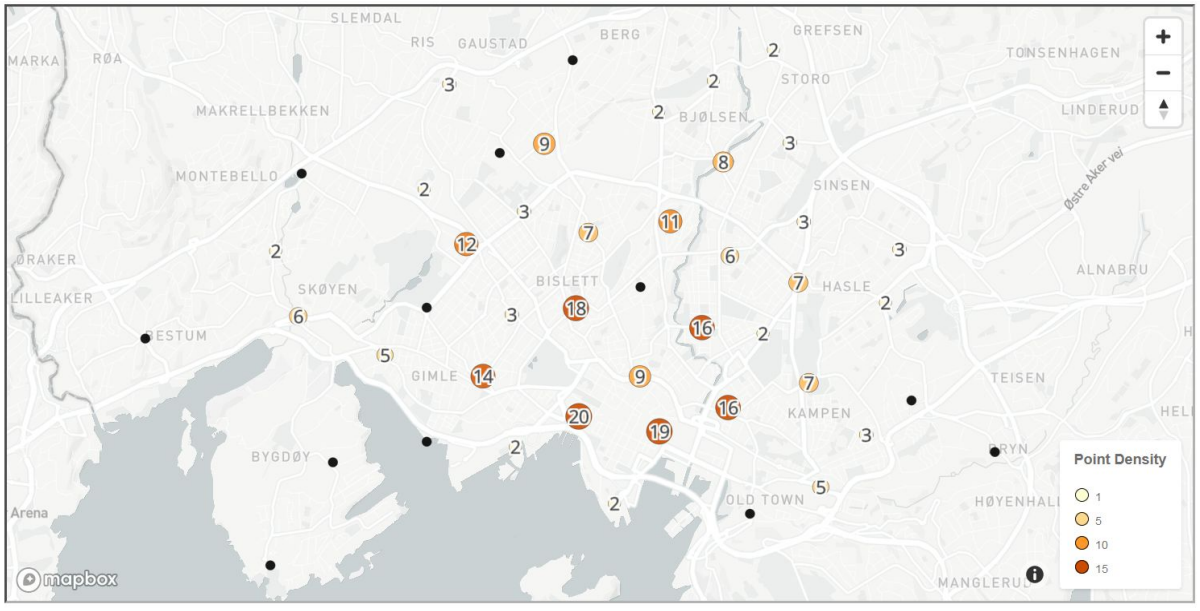
For the weather, there are two features that would need extracting: temperature and "type" of weather (sunny/cloudy/rain/snow) - in relation, of course, to date, time and location. These would be retrieved from the Meteorologisk Institutt's historical weather data and forecasts via their api[12] solution.

Features regarding the traffic, Ruter issues, events, etc. should be included in further versions of the product.

Method: the algorithm(s)

The primary goal of this tool is to identify whether a station needs refilling, emptying or is fine. However, in some areas, Oslo Bysykkel has multiple stations in close proximity. Before proceeding to the main algorithm, there is thus another question to be answered: should each station have its own prediction or should the algorithm look at whether *areas* are in need of refilling/emptying?

The image below shows the density of stations in different areas of the city, illustrating this reality.



Stations in the Oslo area (grouped)

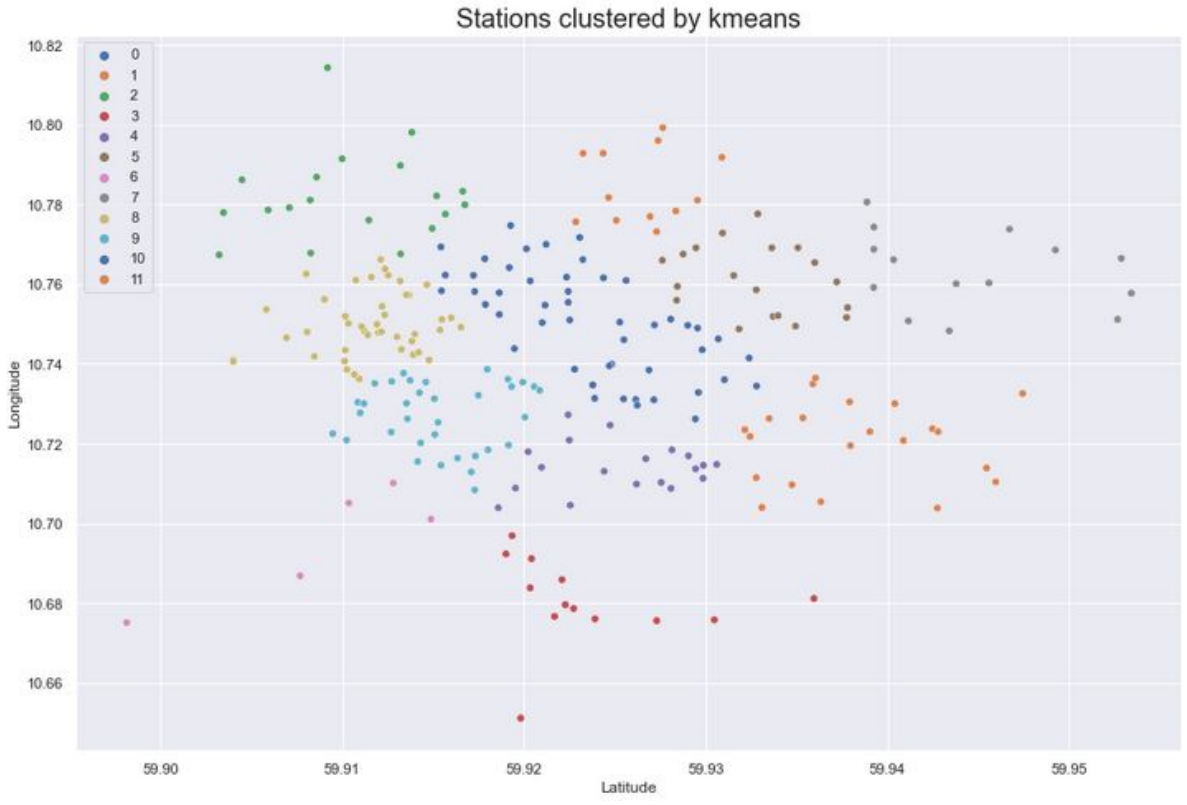
[12] https://api.met.no/

In case Oslo Bysykkel believes zones are a priority, it is possible to cluster the stations into meaningful areas. Selecting the most adequate algorithm might imply testing several different solutions. We have, however, identified KMeans2 as a potential candidate, as shown below.[13]

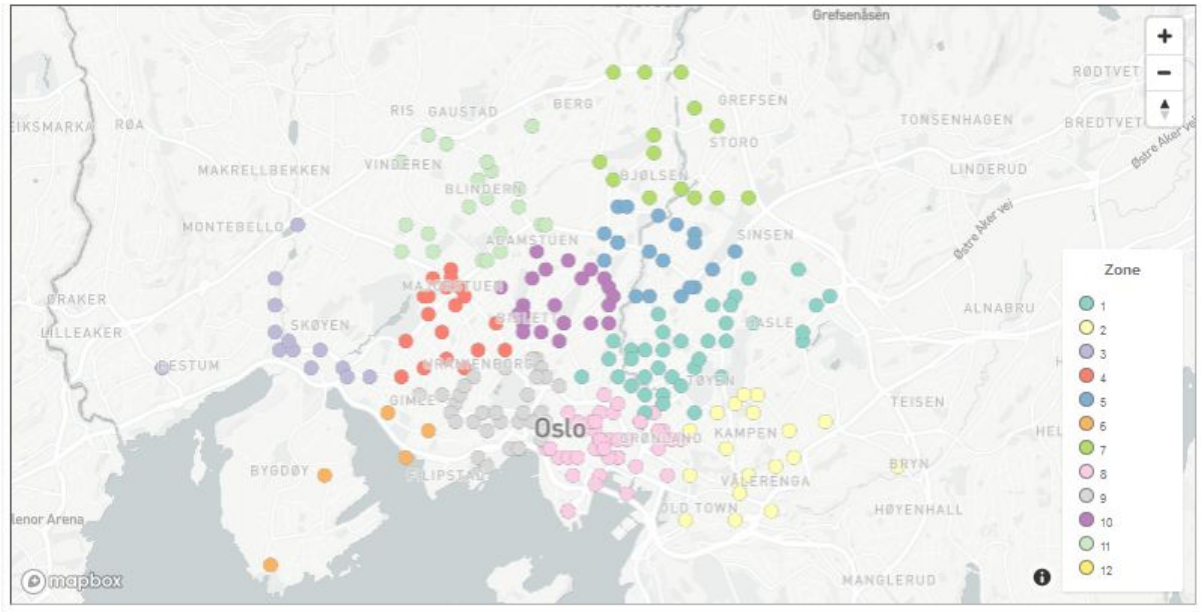
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.cluster.vq import kmeans2, whiten

cor = []
for station in listOfStations:
       cor.append(station.getLongLat())
coordinates= np.array(cor)
x, y = kmeans2(whiten(coordinates), 12, iter = 50)
#y is the var with zones

sns.set(rc={'figure.figsize':(15,10)})
sns.scatterplot(data=coordinates, x=coordinates[:,0], y=coordinates[:,1], hue=y,
palette="deep")
plt.title("Stations clustered by kmeans", fontsize=20)
plt.xlabel('Latitude', fontsize=12)
plt.ylabel('Longitude', fontsize=12)
```



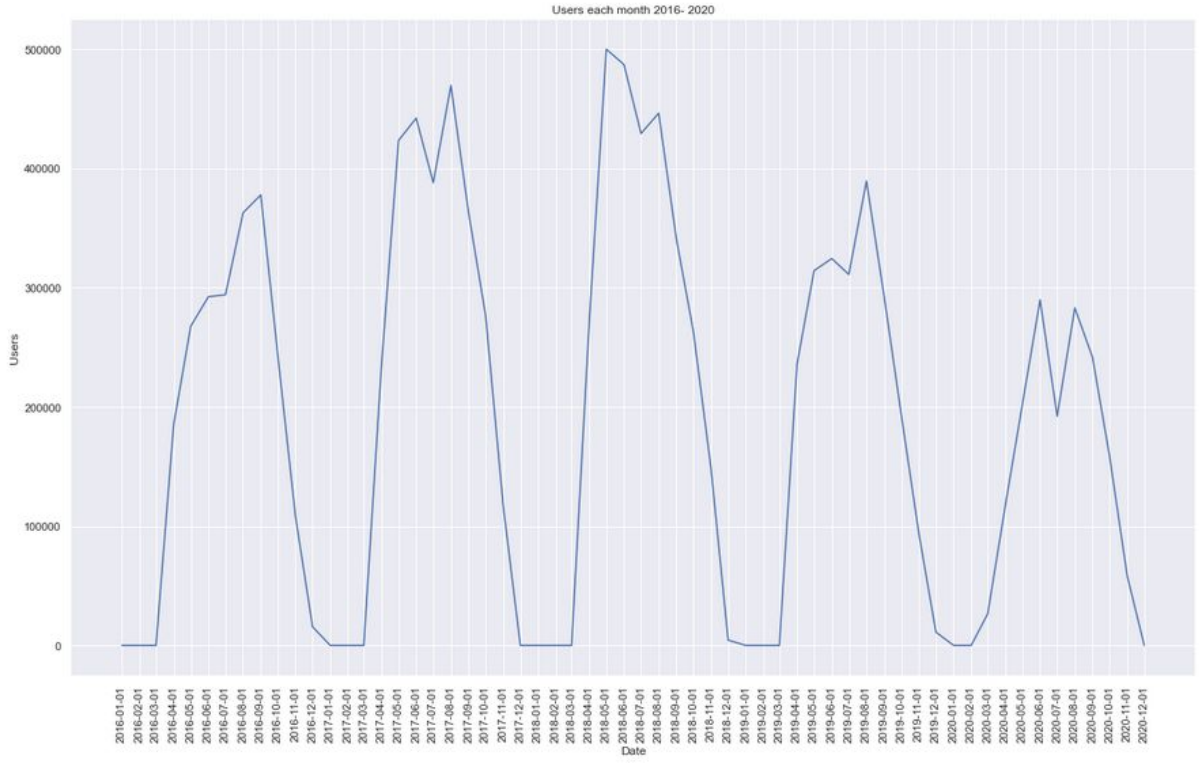
[13] For the calculation, the absolute distance between stations was used. Adaptations such as changing to walking distance or altering the number of zones may need to be made.



Stations (clustered)

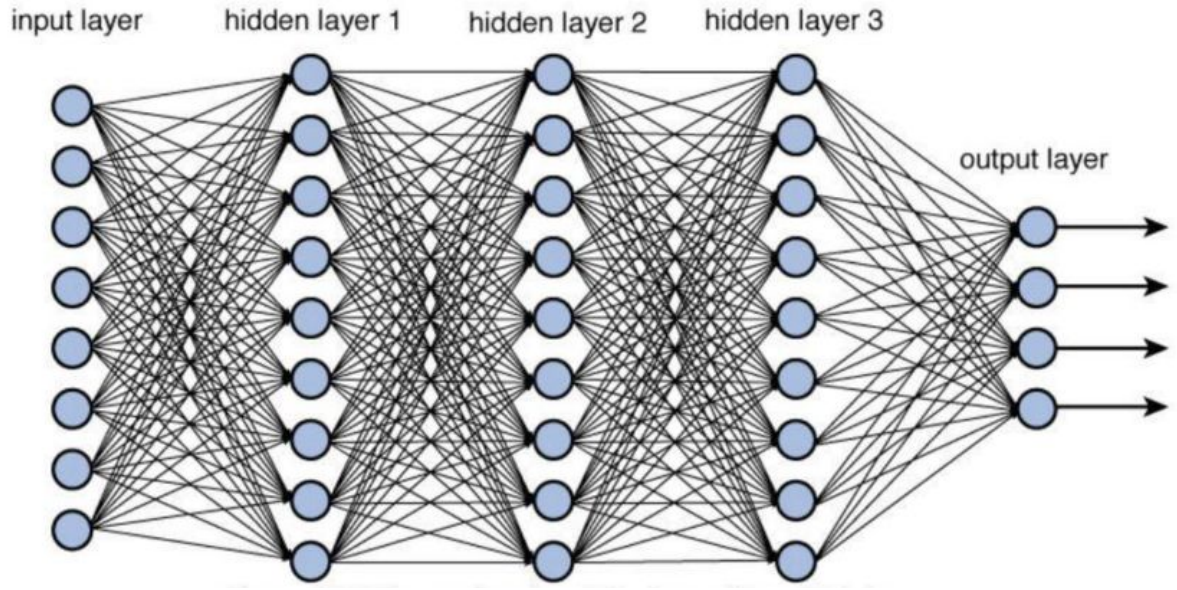
Whether it is proven to be more convenient to cluster or not, the goal is, ultimately, to use a classification machine learning algorithm to determine whether a station is, at any given time: a) Perfectly fine; b) In need of more bicycles; or c) In need of more free parking spots.

In order to do so, the input will be constituted of the many station status updates, including information about the station itself (ID, capacity), available bicycles, date and time (of day, of the year) and weather (temperature and type of weather). Our model should also consider past data for its prediction. Since, as seen in the graph below, the number of bike trips varies in time according to a certain pattern.



We believe it would be ideal for the data considered to include information from the 30 days prior to the date in analysis, as well as measurements from the same period in the previous year.

Considering the volume of input data, it is necessary to find the adequate algorithm to output accurate predictions. However, as research shows, it is often necessary to try different algorithms before settling for the most effective one. A study regarding a similar prediction problem (traffic on sports events days),used ARIMA, neural networks, support vector regression and k-nearest neighbour (*k*-NN), in their attempt to obtain adequate results. (Essien, Petrounias, Sampaio *et al*., 2020) For our specific case, we estimate that neural networks may be the most appropriate choice.



Many-to-many neural network[14]

The neural network algorithm is well fitted to work with multiple inputs and multiple outputs. In our case, testing would be necessary to determine whether the optimal output model is many to many (algorithm returns prediction for all stations at once) or many to one (algorithm returns prediction for only one station, despite having inputs from all stations). Nevertheless neural networks may be used for both options and have a high performance with complex datasets, if configured correctly.

One of its disadvantages is that the model can be viewed as a black box (“hidden layers”), and understanding the prediction may be time consuming. Furthermore, neural networks need vast amounts of training data, although, in this specific case, data from the last four years is available, which is estimated to be enough. Finally, another challenge with this algorithm is to define the model's parameters: it requires time and repetitive testing of values until the optimal one is found.

Considering the magnitude of the task, the complexity and volume of the data and the potential benefits from a correction to Oslo Bysykkel’s issues, we sustain that the advantages of neural networks far outweigh its disadvantages.

[14] Source: https://niessner.github.io/I2DL/

Quality checking: tests and updates

Once the algorithm is trained, it should be tested using past data. As common for such tests, it should consist of requesting a prediction of the status (fine/empty/refill) of given stations at given times; the predicted results should be then measured against the real status of the stations at those times in order to identify the algorithm's accuracy.

Furthermore, we propose that this solution should be developed under continuous integration - as illustrated by the aforementioned possibilities of further, more complex versions. This would also allow for testing of the efficiency of the newest version of the algorithm against older versions. A test for the initial version could be conducted through a comparison with Oslo Bysykkel's current refilling method[15]. It would be expected that the new version is more accurate than the previous one(s). Since a new distribution of bikes will lead to new user behaviour it is important we continuously test, manage and observe so we can be sure we are fixing the problems, not just changing where the empty zones emerge.

In the long term, the accuracy of the algorithm may also be measured through an analysis of customer satisfaction (complaints or lack thereof) and customer loyalty, which could be done by analysing Oslo Bysykkel's customer support data[16]. Although customer satisfaction is not solely dependent on the station classification algorithm, a noticeable improvement in the service quality and reliability could be identified through these measurements.

Finally, it is paramount to track the company's ability to follow the algorithm's recommendation. If the drivers and trucks are insufficient to perform all the suggested bicycle moves or if the company deliberately disregards the algorithm's results, customer feedback is no longer a possible source of feedback.

In accordance to insights derived from long term feedback and other observations, as well as due to the need of regular maintenance, the algorithm should be retrained in the middle of the bike season, as well as once it is over. For the second case, there is a three month gap; however, for the first, issues such as the availability of workers in a busy period might arise. Additionally, retraining an algorithm consumes time and requires computational resources - such as computer clouding - that might be costly.

[15]Although Oslo Bysykkel reports identifying when bicycle racks are filling and emptying, and consequently informing drivers about the need of moving bikes, there is no further information on https://oslobysykkel.no/en/about specifying what makes up that decisional process.

[16] Messages sent by customers, for example, may be run through a NLP algorithm. This data may, furthermore, give insights about the areas in which areas the model performs more poorly.

Sources

BERGSKAUG, E. (2018, 18 September) Oslo bysykkel har mer enn tredoblet antallet unike brukere på tre år. *abc Nyheter*. Retrieved from https://www.abcnyheter.no/nyheter/norge/2018/09/18/195434010/oslo-bysykkel-har-mer-enn-tredoblet-antallet-unike-brukere-pa-tre-ar

ESSIAN, A., PETROUNIAS, I., SAMPAIO, P. *et al.* (2020) A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web*. Retrieved from https://doi.org/10.1007/s11280-020-00800-3

HEM, L. E, IVERSEN, N. M. (2019) Hvordan påvirker været vårt forbruk? *Econa.* Retrieved from https://www.magma.no/hvordan-pavirker-varet-vart-forbruk

METEOROLOGISK INSTITUTT (n/a). Weather API. Retrieved from https://api.met.no/

OSLO BYSYKKEL. (n/a) Retrieved from https://oslobysykkel.no/

OSLO KOMMUNE. (2020) Hjemmekontor. Retrieved from https://www.oslo.kommune.no/koronavirus/rad-og-regler-i-oslo/hjemmekontor/

Original graphs and images produced by the group available at https://github.com/anafvana/DAVE3625_FinalProject