Collegium Da Vinci

MICHAŁ LEWANDOWSKI

Prediction of the bike sharing demand in Poznań. Evaluation of pre- and mid pandemic data

MA thesis

written under the supervision of

dr. Jacek Nożewski

Poznań 2021

Table of contents

Introduction

November 2019 marked an end to another season of the local bike sharing system in Poznań, Poland named "Poznański Rower Miejski", also interconnected with the systems in neighbouring cities of Luboń and Komorniki (respectively "Luboński Rower Miejski" and "Komornicki System Rowerowy"). That year revolutionised the public transport and commuting in more than one way – firstly, a new range of bikes has been introduced in addition to the ones the users knew since the they first launched in 2012. As opposed to the latter (referred to as the "3G" bike type), the new "4G" bikes did not need to be checked in and out at a predefined docking station – they could be now found standing freely in the vicinity of the stations or anywhere around the city. Returns came at no extra cost in pre-defined drop-off zones, such as the stations, or the city centre, and came with a fee if the bike has been checked in outside of them. Users riding them back into their rightful parking places could even earn them a commission (Zarząd Transportu Miejskiego w Poznaniu, 2021).

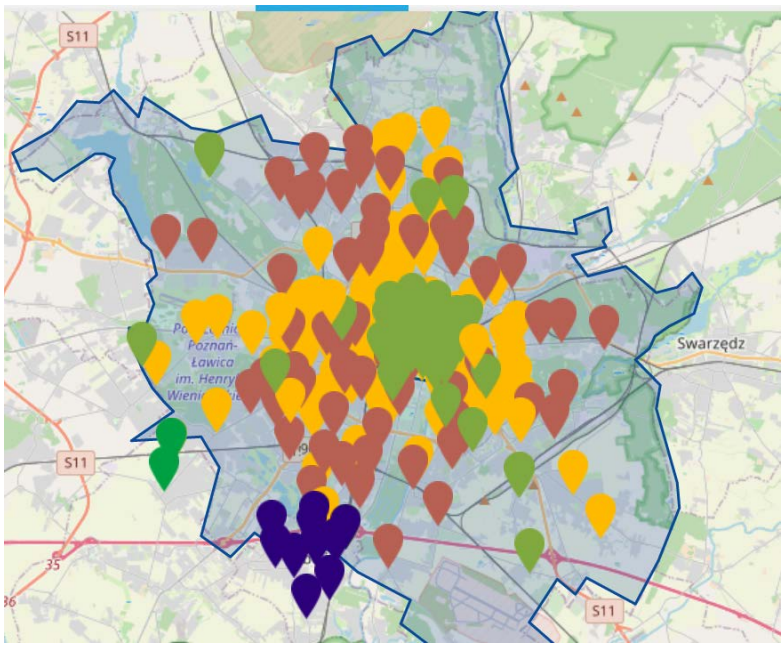

Figure 1. Poznań city boundaries (blue area), 3G stations (yellow marks), 4G drop-off zones (brown marks), freely standing bikes (light green marks), and stations belonging to neighbouring cities (blue and dark green marks) as of August 2021.

No longer than a month later, the unknown at that time change in everyone's daily lives first manifested itself to the wider public on 31 Dec 2019, when World Health Organisation's Country Office in Wuhan, People's Republic of China (PRC) picked up an official statement from the local health authority on cases of "viral pneumonia"

(World Health Organization, 2020), later named COVID-19 in February 2020, by which point it had already claimed 1017 lives in PRC and one abroad (World Health Organization, 2020). By March 2020 the disease has reached Poland with the first case being diagnosed on the 4th of March, and first restrictions and lockdown follow shortly on the 10th of March. By the end of the month bans are placed on social gatherings, pubs, restaurants, beauty parlours; shops and public transportation are required to meet strict capacities on the number of people per square metre. These continue to be lifted and placed back, as throughout the rest of 2020 the country struggles through several spikes in cases, and lockdowns come and go (Stróżyk, 2020), only to be put on a steady track towards returning back to normal with the conditional approval of the first vaccine against the virus in the European Union on 21 December 2020 (European Medicines Agency, 2021), with other similar medicines soon following suit (European Medicines Agency, 2021), and a nation-wide vaccination plan being rolled out and deployed in Poland (Ministerstwo Zdrowia, 2021).

The Polish government response as well as the social fear of being infected had a substantial negative impact on the demand in the common transport market among the passengers country-wide, however as compared to some other regions (i.e. Kuyavian-Pomeranian, Lower Silesian, Masovian, Pomeranian and West Pomeranian), the Greater Poland voivodeship where Poznań is located has not suffered a significant decrease in mobility in the public transport (Wielechowski, Czech, & Grzęda, 2020). The bike sharing system in the aforementioned city has not yet been subject to any works focused on building a regressive machine learning model predicting the demand yet, which will be the primary focus of this thesis, taking into account the pre- and mid-pandemic circumstances. Other research on building predictive demand models in different cities around the world and assessment of COVID-19 on bike-sharing usage has already been conducted, which will be reviewed in the subsequent literature review.

Given the above, this work will focus on building a machine learning model to predict the demand for the bikes at different stations. We will evaluate both the rentals and returns in order to build groundwork for a flexible system, that would allow the operator to determine where bikes would be returned and rented the most. As features we will consider using various weather measurements, like air and ground temperature, count of days above a certain temperature, days with storms/fog, amount of precipitation, air pressure, etc., as well as cultural factors – the amount of holidays, observances, and trade Sundays each month in Poland.

Literature review

The aforementioned negative tendency against public transport indicated by Wielechowski et al., has not only been observed in Poznań alone. As noted in (Nikiforiadis, Ayfantopopulou, & Stamelou, 2020), a study based on a street survey in Thessaloniki, Greece has been conducted by the authors to determine COVID-19 impact on the perception of bike-sharing of its citizens. Nikiforiadis et al. point out that in the years prior to the pandemic, the main transportation node in the city was private cars, while there were little efforts to promote bike rentals from the local service. The infrastructure of both docking stations and bike lanes are also deemed to be sparse, nonetheless the study has shown that bike-sharing popularity will not be significantly affected by the disease, and it could have become more attractive for "a proportion of people". Additionally, it has been determined that the population deems this type of transportation safer than other public transport means, and people who previously commuted as private car passengers (not drivers) are now inclined towards bike-sharing.

A number of studies has already been conducted on utilising machine learning techniques in order to predict their demand as noted by Sathishkumar & Yongyun (2020) as "there is a need to manage the bike rental demand and manage the continuous and convenient service for the users". Balancing the count of bikes between station is also seen as a logistic challenge, to which the latter work proposes a data mining approach to obtain the data – which includes the count of bikes at every docking point in hourly intervals, and an assortment of weather information at those points in time, such as the temperature, humidity, precipitation, solar radiation, etc. Five statistical models are then picked and trained using this data in order to compare their efficiency in this scenario. The models are: CUBIST, regularised random forest, CART (Classification and Regression Trees), KNN (K-nearest Neighbours) and CIT (Conditional Interference Tree). They are then measured against each other using indices such as: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Variance (R^2) and Coefficient of Variation (CV), based on which the rule-based CUBIST model is deemed to be most efficient (about 95% R^2). The most influential features of the dataset are determined as temperature and hour of the day.

A different approach to this problem is seen in the article by Thirumalai & Koppuravuri (2017), where a deep neural network is applied as the sole solution. The

dataset is more granular than the one used by Sathishkumar et al., as it includes not only the hourly rentals and weather data (temperature, humidity, wind speed), but also the split of users into two categories – "casual" and "registered" ones, although the difference between those is not explained. The deep neural networks achieved an accuracy score of ~80%.

A similar issue has been approached in the research paper conducted by Ashqar, et al. (2017) where authors modelled the availability of bikes at San Francisco Bay Area stations. An anonymised dataset containing both the state of every station (number of bikes, docks, timestamp) as well as all the records of rentals/returns that happened between August 2013 and August 2015. Given the large area the system covers, each station was identified by not only its ID, but also the ZIP code where it was located, which was then matched with 22 variables describing the weather that day for a particular code. The authors have applied Random Forests and Last-Squares Boosting algorithms to predict how many bikes would be available at each station in the network at a given time, and to investigate how the variables influence their count. Mean Absolute Error was used to determine the prediction error at each station – which decreased as the authors experimented with incrementing the number of trees generated by the models from 20 until reaching sufficient accuracy at 140 trees. In addition to these univariate models, a multivariate Partial Least-Squares Regression model has been implemented to reduce the number of models and the tracking needed for those (in the univariate setting each station had its own model). The study results have shown that the univariate models produced predictions with a lower MAE than the multivariate one, although the authors have found it within acceptable levels when taken into account with the relative simplification it offers over a large number of stations.

1. Methods

1.1. Theoretical framework

In this section we will give a brief overview of theory behind the methods applied in this work, such as data retrieval, and the chosen machine learning model.

1.1.1. Data retrieval techniques

All pieces of the final dataset described one by one in Dataset have been obtained and/or downloaded manually. No data mining techniques have been utilised, although the entities from which data has been obtained offer either highly restricted Application Programming Interfaces (APIs) or paid ones. Due to the low manual effort related to the data retrieval, as well as relatively high cost of automation, the authors have decided not to automate the data retrieval.

1.1.2. Support Vector Machines

In this thesis we will utilise the Support Vector Machines (SVM) model. It is characterised with the ability to obtain satisfying generalisations on a limited training data set size, which we believe may be the best fit given the dataset aggregation level. SVM utilises a learning algorithm that can recognise subtle patterns in complex data sets (Basak, Pal, & Patranabis, 2007).

In SVM there are two main categories: Support Vector Classification and Support Vector Regression. The latter, which we will implement, is the most common application of this model. While the purpose majority of linear regression models is to minimise the sum of squared errors, SVM actually allows an acceptable error range to be defined as a hyperparameter and finds the plane that best fits the data given the error margin. Maximum error denoted with an ϵ (epsilon) can be then tuned to achieve the desired accuracy within the model, however there are further error-centred hyperparameters such as slack (C parameter) used in SVM. Potentially there always could be deviations from the error margin, so slack allows us to define how significant it can be before the model stops considering those marginal values. Larger C values result in a higher desire to

predict more closely, while smaller ones allow more outliers (in the case of classification) or less accurate output variables to the point of over- or underfitting (Sharp, 2020).

The model also utilises kernel functions to modify the definition of the dot product in the linear formulation, therefore transforming a non-linear training data set into a higher dimension so a linear equation for the hyperplane can be obtained.

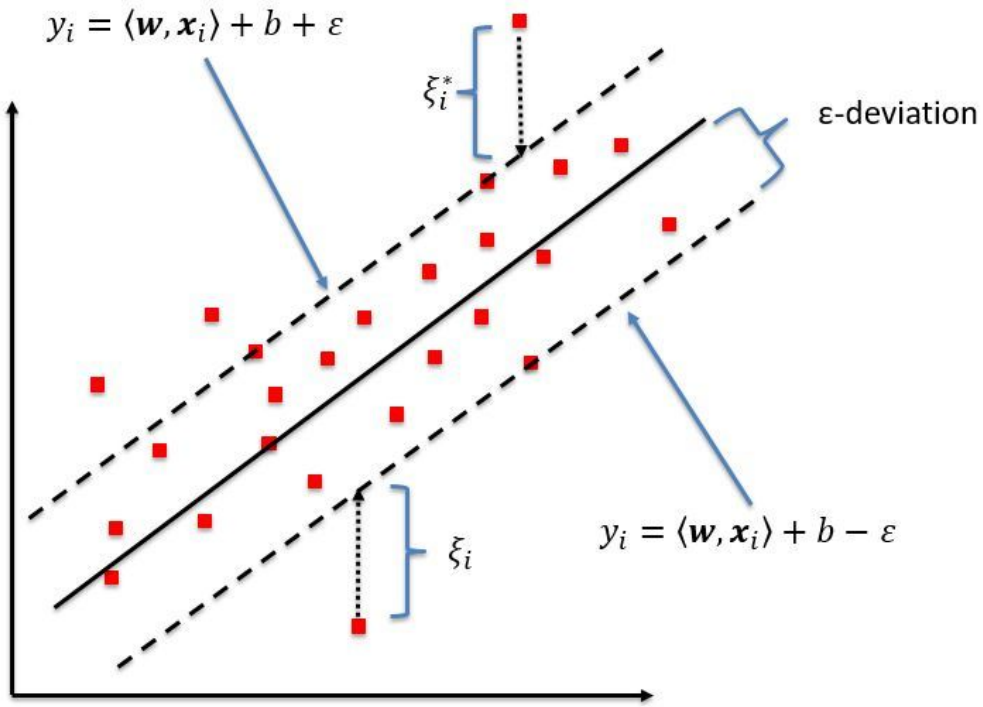

Figure 2. Schematic of an one-dimensional support vector regression (SVR) model. Only the points outside of the 'tube' are used for making predictions. Source: (Kleynhans, Montanaro, Gerace, & Kanan, 2017)

Since the implementation of the actual model will be done in Python and the *scikit-learn* library, the following kernels can be utilised: linear, polynomial, Radial Basis Function (RBF), sigmoid, and a precomputed one. The linear kernel should be used when the data is linearly separable and when there is a large number of features. It is generally considered the fastest one to train and only requires the C parameter to be optimised – it does not modify the dot product. In the RBF kernel, the induced space is made of Gaussian distributions, where each point is transformed into a probability density function of a normal distribution. The polynomial kernel induces space of polynomial combinations of the used featured up to a certain degree, which can be then tuned as an additional hyperparameter in addition to C and the maximum error. (Pedregosa, et al., 2011).

1.2. Dataset

The primary building block of the dataset used for our model is the historical data for the bike sharing system in Poznan. It is not publicly available and it has been provided to the authors upon their request to Nextbike – the operator of the system in Poznań, who then redirected us to the local Zarząd Transportu Miejskiego (ZTM; *Office of Public Transportation*) as they are the entity that hold direct control of the data. The request for data has been submitted in the end of April 2020, when we also committed to share the final model and the research findings with the office should the data be shared, and was approved soon after in the beginning of May 2020. It should be noted that the e-mail reply from ZTM did not relate in any way to our commitment – the dataset could be simply found as an attachment with several .xlsx spreadsheets within it.

The data contained there spans three years: 2019, 2020, and 2021. The 2019 data starts in March, when the bike sharing season usually starts, up until its regular closure in November. The year when the pandemic has reached its peak and the country was subject to several lockdowns – 2020 – also started with the bike season launching in March (with the first lockdown hitting only on March 10), however a month later the Polish Health Authority introduced a range of restrictions around public transport, including a temporary ban on the use of any bike sharing systems (Zarząd Transportu Miejskiego, 2020), which resulted in the Poznań service shutting down all the way until May 6, when it restarted (Zarząd Transportu Miejskiego, 2020). This can be seen in the dataset as a sharp drop all the way to 0, and possibly a small decrease in the results on May. Starting from June 1 up until September 15, the operator decided to stop charging any medical staff using the system in all 40 systems they run in Poland, including Poznań, which could have provided direct stimulus to increase the utility of the system in that period, and possibly after it because of a more positive image of the company (Zarząd Transportu Miejskiego, 2020). It should be noted that while the season is usually concluded in November, however due to the decrease in other forms of public transportation and a spike in the amount of new users, ZTM has extended it to December 23 (Zarząd Transportu Miejskiego, 2020). During the last month the available bike count also fluctuated, as since December 8, 40% of the bikes were already withdrawn from use. These two years of activity can be seen plotted out in the figure below.

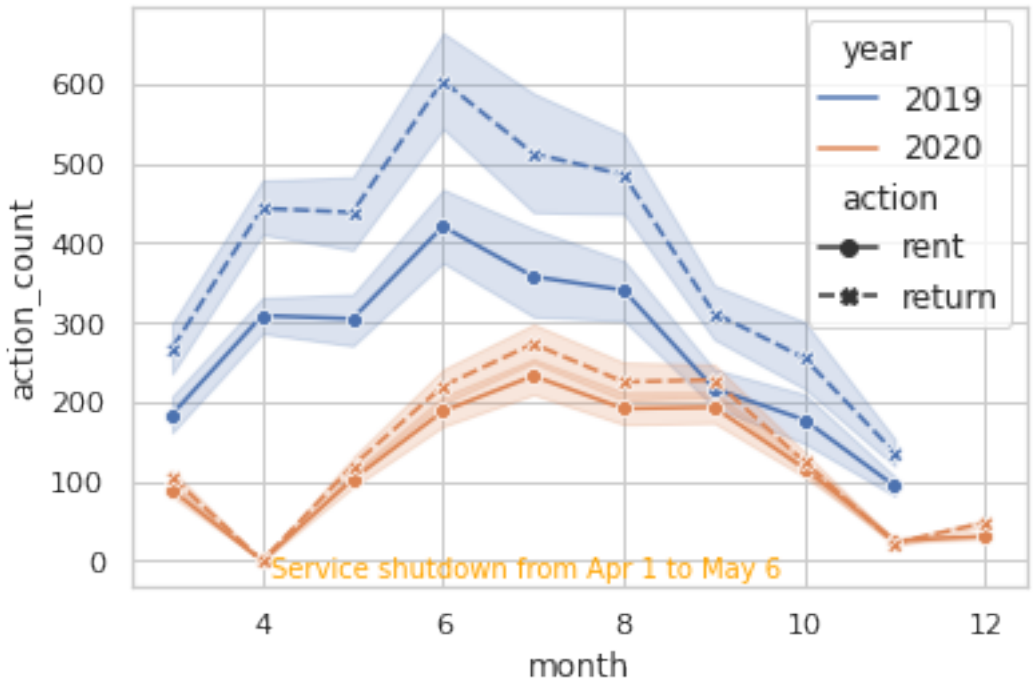

Figure 3. Bike rentals and returns in Poznań in 2020 as compared to 2021.

Year 2021 in the dataset is of small relevance, as only the data for March has been provided. In order not to minimise the presence of outliers in the training dataset, this year has been excluded from it, however it can be used to validate the model. It should also be noted that the count of rentals subtracted from the returns in a given month almost never equals 0, as bikes could be lost, stolen, broken and withdrawn from service until repaired, or simply overdue for a return for longer than 30 days.

In addition to the number of rents and returns each month, which will be used as target variables later, this part of the dataset also includes a set of categorical variables, such as: *location* (the name of the station; note that 4G bike returns outside of the stations and the city centre are not included in this dataset; city centre is locations starting with "strefa_centrum[…]"), *drive_type* (some of the 4G bike models can be equipped with an electric motor), *bike_type* (or the generation of the bike – either 3G, which needs to be returned at a docking station, or 4G that can be left outside of it), *child_seat* (used to indicate 3G bikes that feature a child seat in the back), and *bike_size* (used to indicate whether a bike is adult- or children-sized; in the given timeframe there were only 3G child bikes, that could be rented and returned only at 3 special stations located around the Malta Lake and its recreational areas).

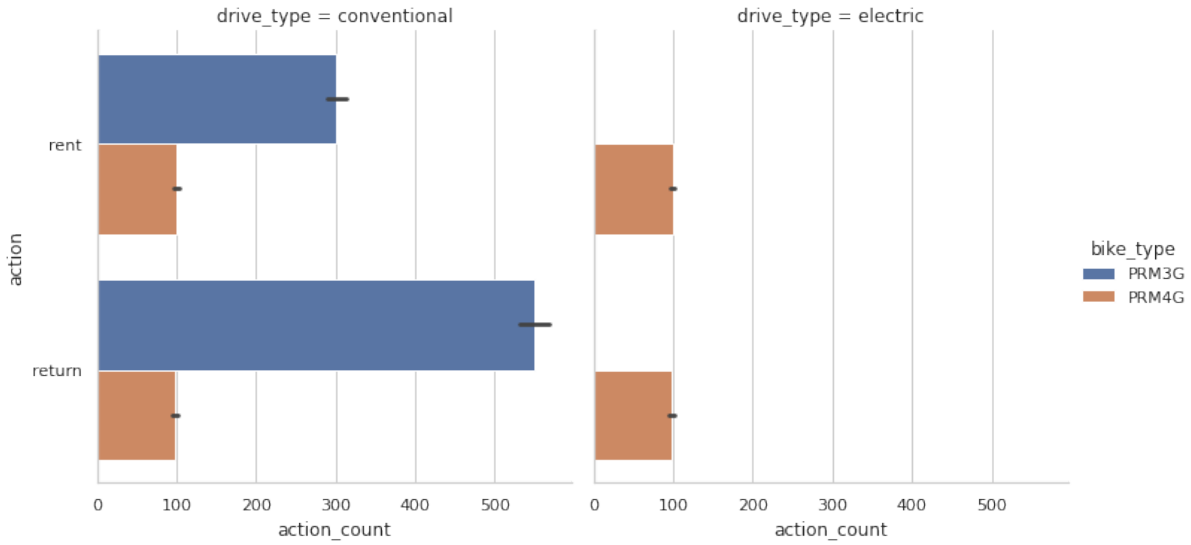

Figure 4. Distribution of bike usage across all years per bike type and its drive type

As mentioned previously, the raw data the authors received were spread across several spreadsheets – each for a given year (2019-21) and bike type (adult 3G without a child seat; adult 3G with a child seat and children bikes; electric or conventional adult 4G). Additionally, the data was spread across different tabs based on the bike type – a baseline of minimum 2 tabs was always present, as each contained either the rentals or the returns, which could then be further increased to cover the additional categories (i.e. drive types, presence of a child seat, etc.). In each file the rows consisted of the station names, while the months of a given year made up the columns. Due to the above, the data required to be transposed and transformed into a longer and more suitable format. The *date* column at this point has been artificially extended to also include a dummy day (1), so it could be parsed as the *datetime* data type in Python and improve human readability, however before feeding into the model it was split into separate columns of *year* and *month* respectively. An example of the dataset pre and post the modifications is given below using the output of the *head()* function of Pandas in Python.

Table 1. Raw data for 2019 rentals for adult 3G bikes.

index	*location*	*total*	*MAR*	*APR*	*MAY*	*JUN*	*JUL*	*AUG*	*SEP*	*OCT*	*NOV*
0	Aleje Marcinkowskiego	6747	584	868	931	1079	880	854	643	588	320
1	Aleje Solidarności	10558	845	1479	1316	1747	1583	1467	924	822	375
2	AWF	13235	1042	1801	1655	2585	1794	1762	1136	968	492

3	Bałtycka/Gdyńska	73	2	5	5	11	23	16	5	4	2
4	Bohaterów Westerplatte	151	1	13	34	23	24	22	14	18	2

Table 2. Fully transformed bike sharing data.

index	*location*	*drive_type*	*action*	*bike_type*	*child_seat*	*bike_size*	*action_count*	*date*
0	Aleje Marcinkowskiego	conventional	rent	PRM3G	False	adult	584	2019-03-01
1	Aleje Solidarności	conventional	rent	PRM3G	False	adult	845	2019-03-01
2	AWF	conventional	rent	PRM3G	False	adult	1042	2019-03-01
3	Bałtycka/Gdyńska	conventional	rent	PRM3G	False	adult	2	2019-03-01
4	Bohaterów Westerplatte	conventional	rent	PRM3G	False	adult	1	2019-03-01

Following the example of other research on this subject, the above dataset was then enhanced with a range of continuous weather features manually extracted from the annual State Research Institute publications (Szokalska, Rocznik meteorologiczny 2019, 2019) and (Szokalska, Rocznik meteorologiczny 2020, 2020). These tables required a moderate amount of transformations, after which they have been left-joined to the bike sharing dataset, so each data point now featured the weather conditions at that point in time in Poznań. The variables describing the weather at a monthly level are:

- Pśr – average air pressure
- Pmax – maximum air pressure
- Pmin – minimum air pressure
- Tśr – average air temperaturę (sensor located 2m above ground unless stated otherwise)
- Tmaxśr – maximum average air temperature
- Tminśr – minimum average air temperature

- ABS Tmax – maximum absolute air temperature
- ABS Tmin – minimum absolute air temperature
- Uśr – average relative humidity
- Umin – minimum relative humidity
- ffśr – average wind speed
- ffmax – maximum wind speed
- Nśr – average cloudiness
- Rdsuma – total precipitation
- Rdmax – largest precipation total recorded in one day that month
- Tgminśr – minimum average air temperature at ground level (sensor 5cm above ground)
- ABS Tgmin – average absolute air temperature at ground level (sensor 5cm above ground)
- Ssuma – total insolation

Finally the dataset has been enhanced with a set of features not yet used for the purpose of predicting the bike sharing demand - the amount of days in a month that were:

1. bank holidays
2. observances
3. equinoxes
4. trade Sundays

Their distribution can be seen on the below graph.

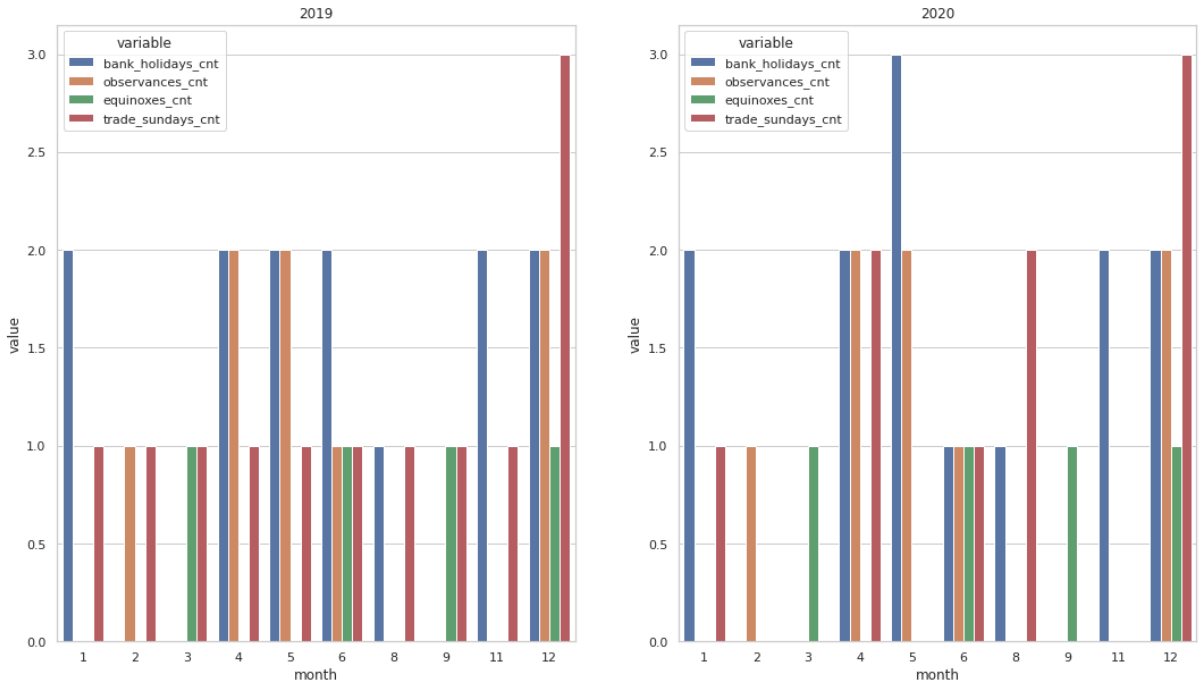

Figure 5. Distribution of bank holidays, observances, equinoxes and trade Sundays in the dataset.

It should be noted that since January 2018, the state has banned trading on most Sundays in Poland. The amount of those days when trade is actually allowed decreases as time goes by, as in 2018 there were 29 trade Sundays out of total 52 Sundays; in 2019 it was 15 out of 52; in 2020 it was 8 out of 52, and so on. This data has been collected manually based on Ustawa o ograniczeniu handlu w niedziele i święta (2018). The holidays, observances and equinoxes have been sourced from the work of Thorsen (2021).

In total, the entire dataset used for the modelling spans 17,001 rows and 57 columns, out of which 5 are categorical, and the rest is continuous (either *floats* or *integers*), and there are no missing data points, therefore it wasn't necessary to filter them out or impute them. The entire code base for the data pipeline, analyses and model build & testing can be found in the *graduation* branch of a GitHub repository.

1.3. Research assumptions

This work aims to check several hypotheses revolving around the issue of predicting the bike sharing demand in Poznań based on external factors:

1. Bike sharing demand (rentals and returns) on a monthly level can be predicted using SVR while achieving a R^2 score of at least 75%. While multiple works have explored modelling the demand on a more granular level, this has never been done on more aggregated data, and predictions are certainly able to be made.
2. The SARS-CoV-2 pandemic has positively impacted the demand for bike sharing in Poznań. As indicated in the work of Wielechowski, Czech and Grzęda (2020), the Greater Poland voivodeship has not suffered a decline in popularity of public transport, and while we are not aware of data from the same period that we could compare against bike sharing, the demand for the service and its bikes might have grown as it's a form of public transit not involving others.
3. Weather variables (temperature, precipitation, pressure) are correlated to the bike sharing demand, achieving at least 0.5 correlation in comparison to the target variable. Previous studies on this subject using more granular data have shown that there is indeed correlation between those two variables, therefore moving to an aggregated dataset requires additional validation of this claim.
4. Occurrences of holidays, observances, and equinoxes are correlated to the bike sharing demand, achieving at least 0.5 correlation in comparison to the target variable. No previous studies that we are aware of explored the effects of combinations of working and non-working events (going even as far as astronomical events), and we assume that the demand increases at least during holidays, when used for recreational purposes.
5. Occurrences of trade Sundays are correlated to the bike sharing demand, achieving at least 0.5 correlation in comparison to the target variable. Similar to the previous hypothesis, the effect of an increased trade activity represented by the amount of trade Sundays in a month on the bike sharing demand has not been investigated before. Our assumption is that the demand increases as the number of their occurrences increases, as people bike to and/or from shopping centres and other such venues.

   The above assumptions will be evaluated in the subsequent sections.

1.3.1.1. Research process

In order to develop any machine learning model, the data first need to be cleaned and transformed. Since our dataset consists of several different data sources, it was necessary to ensure that all of them are in the same format, so they can be joined together. It has been achieved using several built-in functions available in Python, and some from the *Pandas* package; they include, most notably: *map*, *melt*, *transpose*, *append*, *merge* and the anonymous functions, also called lambda functions. Selected examples of their use can be seen in the following code snippet, which has been used to transpose two of the datasets with month-oriented column names (using either their Polish names, or using Roman numerals) into a longer data format with a singular date column.

```
month_mapping = {'I': 1,
                 'II': 2,
                 'III': 3,
                 'IV': 4,
                 'V': 5,
                 'VI': 6,
                 'VII': 7,
                 'VIII': 8,
                 'IX': 9,
                 'X': 10,
                 'XI': 11,
                 'XII': 12}
df['month'] = df['month'].map(month_mapping)
df['date'] = df['month'].apply(lambda x: datetime.datetime(file_year, i
nt(x), 1))
df = df.drop('month', axis=1)
```

The assumption is made that *dataframe* (*df*) is loaded in from a file that spans only one year (hence the variable *file_year*). A mapping from the string describing the month to its number is then applied, so the result can be then joined together into a full date with a dummy day of 1 and cast into the *datetime* data type inside a lambda function. The result of these transformations can be seen in the *date* column in Table 2.

The data then needed to be cleaned – especially the location names. While no values describing the same station with a different amount of whitespaces or typos that would lead into treating them as two different ones have been found, their punctuation is highly inconsistent – most notably around slashes, which indicate that a place is situated on the corner of two different streets. Also, most of the names refer to streets where they are situated, however only some feature *ul.* (abbreviation for *ulica* – *street*) in the

beginning, while others do not. Some names start with *ul.* in the 2019 dataset, but without it in the 2020 one. They would also be treated as two different locations. Similarly, names of stations situated next to avenues (*al.* or *aleja*/*aleje*), squares (*pl.* or *plac*) or housing estates (*os.* or *osiedle*) interchangeably use the full and shortened word variants in Polish. Examples of the above inconsistencies are: *Głogowska / ul. Krauthofera* and *Bałtycka/Gdyńska*; *Pl. Wolności* and *Plac Wiosny Ludów*; *Aleje Marcinkowskiego* and *Al. Jana Pawła II*. All the letters were also made lowercase and normalised into Unicode. The following code was then used to transform them as mentioned previously.

```
df['location'] = df['location'].str.lower()
df['location'] = df['location'].str.unicodedata.normalize('NFD')

chars_to_drop = ['.', ',', 'ul']
slashes_combo = [' / ', ' /', '/ ']
normalisation_dict = {'os ': 'osiedle ',
                      'pl ': 'plac ',
                      'al ': 'aleja',
                      '  ': ' ',
                      ' ': '_'}

for char in chars_to_drop:
  df['location'] = df['location'].str.replace(char, '')
for char in slashes_combo:
  df['location'] = df['location'].str.replace(char, '/')

for i in normalisation_dict:
  df['location'] = df['location'].str.replace(i, normalisation_dict[i])
```

After transforming and cleaning, the data needed to be reviewed to select the most relevant features for the target variable – this has been done by building a correlation matrix against column *action_count*, with the correlation coefficient being made an absolute value in the code in order to more easily determine the features that are more strongly correlated. The below correlation matrix has been constructed from coefficients before turned into absolute values to more suitably display the interdependencies to the reader, which is then followed by a code snippet of the top absolute coefficients.

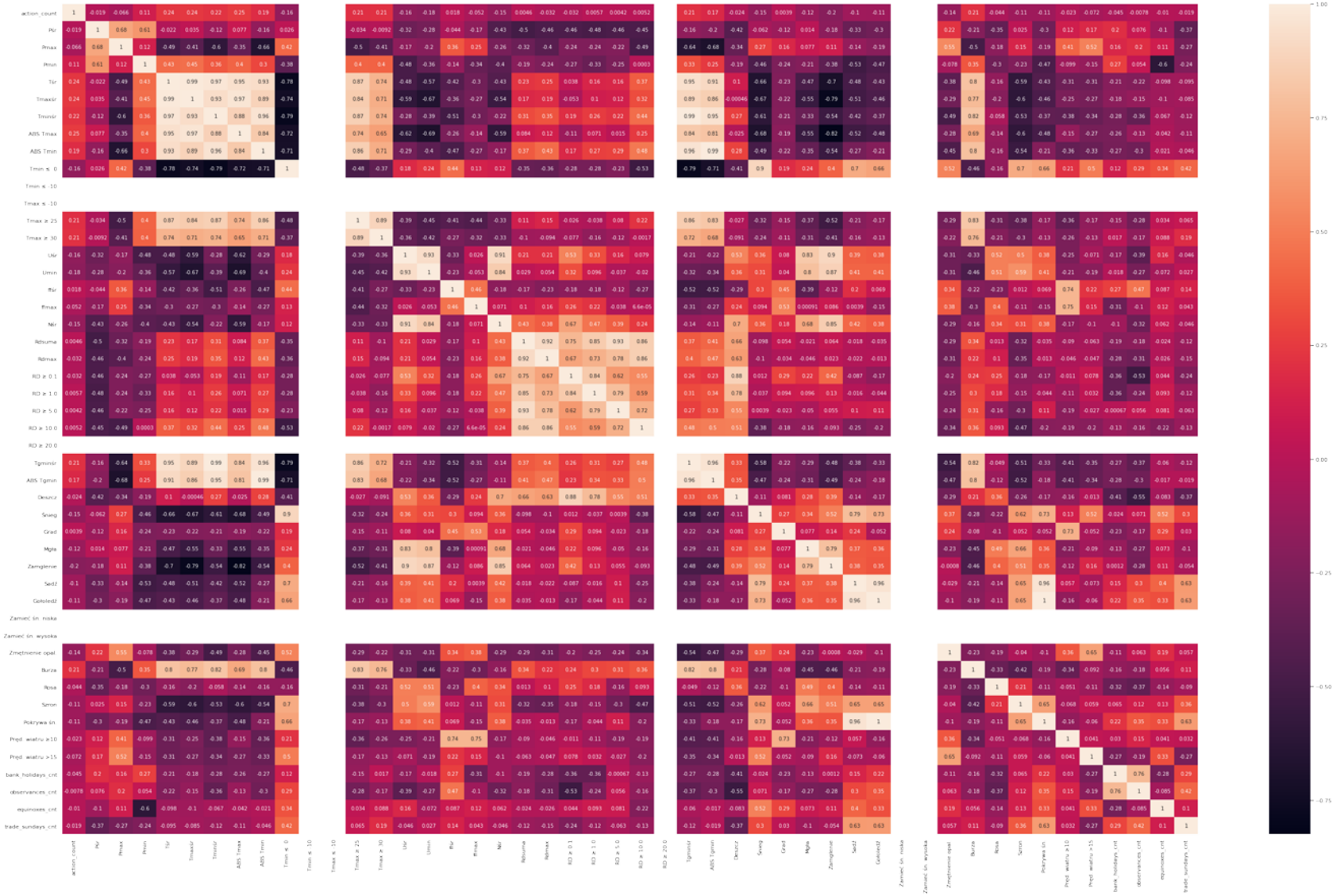

Figure 6. Correlation matrix of all features and the target variable.

```
# set the dependant variable correlation results to a var
corr_target = abs(corr_matrix['action_count'])

# select highly correlated features
relevant_features = corr_target[(corr_target > 0.15) & (corr_target < 1
)].sort_values()
print(relevant_features)

# Śnieg         0.150948
# Tmin ≤  0     0.158801
# Uśr           0.163050
# ABS Tgmin     0.172713
# Umin          0.176469
# ABS Tmin      0.188384
# Zamglenie     0.198589
# Tmax ≥ 25     0.207963
# Tgminśr       0.210081
# Burza         0.213232
# Tmax ≥ 30     0.214112
# Tminśr        0.218982
# Tśr           0.236673
# Tmaxśr        0.240265
# ABS Tmax      0.253286
# Name: action_count, dtype: float64
```

As it can be seen above, the coefficients are not high at all, with the highest one for *ABS Tmax* being 0.253286, and others as low as 0.0052 (the code snippet only includes features where 0.15 > *corr* < 1.0), which instantly overturns the last 3 hypotheses we have made: the impact of holidays, observances, equinoxes and trade Sundays is minuscule, with equinoxes being surprisingly the mostly correlated one (0.01). The atmospheric variables also do not reach the presumed correlation goal of at least 0.5, which alone might decide the low accuracy of the model, however the study continues in order to determine its scores.

Subsequently the top performing features (where 0.15 > *corr* < 1.0) are checked for multicollinearity in order to provide the model with the most independent features. A correlation threshold of less than 0.1 is established for this purpose and features not meeting this criterion was removed from the dataset:

```
relevant_uncorrelated_features = relevant_features_corr[relevant_featur
es_corr < 0.1].unstack().sort_values(ascending = False).dropna().drop_d
uplicates().reset_index(level=[0,1]).level_0.drop_duplicates().values

print(relevant_uncorrelated_features)
#['Tgminśr' 'ABS Tgmin' 'Śnieg' 'Burza' 'Uśr' 'ABS Tmin' 'Tmax ≥ 25'
'Umin'
 'Tmin ≤  0' 'Zamglenie' 'Tśr' 'Tmaxśr' 'ABS Tmax' 'Tminśr']
```

The features obtained from the above code snippet are then extracted from the initial dataset, and put in a separate *dataframe*. An early version of the model could then be built, having split it into the subset for the rents (*action = rent*) and returns (*action = return*), as the bike sharing demand is defined by both of those variables – the operator needs to know where most bikes will be returned, and then move them overnight to where they will be most rented in order to satisfy the consumers. Due to the obvious drop in both metrics in 2020 as seen in Figure 3 we have also made the decision to further break down those two datasets by the year. Before the 4 datasets could be fed into the model, they were first split into training and test data (75:25 split), which only then was normalised in order to avoid cross-contamination and spoiling the test data. The SVR models were then trained using RBF, linear, sigmoid and polynomial kernels, to see under which one the data performs the best, which will be presented in the following section.

2. Data analysis and results

As mentioned in the previous section, the data needed to be split between both years, therefore the resulting models could then be interpreted as the pre-pandemic, and mid-pandemic models, that can be then applied to make predictions based on how the circumstances move forward - when (and if) the world returns to a normal state of affairs, the 2019 model can be used to make predictions, while the 2020 one could be used if further lockdowns and irregularities in the service operation keep occurring. This fact on its own proved the second hypothesis to be false – the bike sharing service in 2020 did not come close to 2019 during all but one month – September – where both metrics came close meeting last year's demand, and throughout the rest of it, they were halfway there.

The data has also proven to be a miss in terms of the correlation – all additional datasets that we have appended (i.e. trade Sundays, holidays, etc.) have proven their correlation to the target variable to be lower than 0.15 (the bare minimum threshold we set out given the overall weak correlation). The atmospherical factors have also shown a weak overall correlation, as seen on Figure 6, with the highest coefficient being at ~0.25. Those features included: the maximum absolute temperature recorded in a single day in a month, temperature averages, count of days where a storm has occurred or when fog has risen, precipitation minimum and average, and count of days with snowfall occurring.

At this point we have decided to continue research and still train the models to see how they would perform. First, they have been trained with the default kernel available in *scikit-learn* for SVR – RBF. This did not yield good results, as this kernel is not suitable for this data, as it can be seen in the below code snippet, where a custom function was used to print out various metrics:

```
[print_model_scores(x) for x in [rents_2019_model, returns_2019_model,
rents_2020_model, returns_2020_model]]

#R2 score: -0.15742536324316436
#Mean squared error: 281555.997780484
#Explained variance score: 0.018916846164180767
#----------
#R2 score: -0.18782860779922417
#Mean squared error: 398027.2820036541
#Explained variance score: 0.02897247992038532
#----------
#R2 score: 0.0028862267863679625
#Mean squared error: 28336.136637604282
#Explained variance score: 0.1152039933242337
#----------
#R2 score: 0.037492873866914245
#Mean squared error: 33423.345822261275
#Explained variance score: 0.1276423500950019
#----------
```

As it can be seen above, the provided set of features together with the default kernel produces R^2 scores that fall even below 0 for the 2019 models, and slightly over 0 for the 2020. The models have then been re-trained with using the polynomial and linear kernels, where the former performed only slightly better, however significantly improved results were reached with the latter:

```
#R2 score: 0.250593479879628
Mean squared error: 182301.08585529806
#Explained variance score: 0.31592148687439325
#----------
#R2 score: 0.23764249967694617
#Mean squared error: 255456.95883759568
#Explained variance score: 0.2875392613246269
#----------
#R2 score: 0.30680482319636926
#Mean squared error: 19699.32998029858
#Explained variance score: 0.35020209175896533
#----------
#R2 score: 0.3016796232520397
#Mean squared error: 24249.382485664435
#Explained variance score: 0.3381533762015452
```

The linear kernel allows the model to reach relatively good scores (as compared to the other kernels), with the R^2 falling between ~0.24 and ~0.30, and MSE being as low as ~19600 in the case of the model for 2020 rentals. Additionally, we have also attempted to improve the score by eliminating the child bikes from the dataset, as they represent a small subcluster within the data (being accessible only from 3 closely located stations), which only slightly improved the scores as seen below:

```
#R2 score: 0.2658162446602612
#Mean squared error: 175791.74141244878
#Explained variance score: 0.32695485573066874
#----------
#R2 score: 0.25877934098812116
#Mean squared error: 248849.68153132976
#Explained variance score: 0.3042545077789357
#----------
#R2 score: 0.32434740994813305
#Mean squared error: 19235.503637000264
#Explained variance score: 0.3645297276815227
#----------
#R2 score: 0.3004200934903948
#Mean squared error: 24915.527117129524
#Explained variance score: 0.3396841992300146
```

Nevertheless, none of these models has reached the threshold of at least 0.75 of R^2, which leads to the rejection of last of the hypotheses we have set out previously. In the subsequent section we will draw conclusions from this study and lay out proposals for how future studies on the bike sharing demand modelling in Poznań could be improved.

Conclusion

The goal of this thesis was to develop a machine learning model to predict the demand for the bike sharing service in Poznań at different stations using weather data aggregated on a monthly level together with the count of various days in each given month – holidays, observances, equinoxes, and trade Sundays. As previous studies have shown, this aim is achievable with more granular datasets (down to the hour, or even minute), however no studies have been conducted on whether this level of aggregation could be used to reliably train a model and receive accurate predictions. Additionally, no research has been conducted on modelling the bike sharing demand in Poznań. In order to build groundwork for a complete system to predict it, we have evaluated both the rentals and returns as the output variables.

This, however, has been proven not viable with the gathered data, as in the final SVR models we trained the R^2 metric varied between ~26% to ~34%, meaning that the datasets could only explain as little of the target variable, although it should be noted that the score has been increased from -0.18% all the way to the indicated levels thanks to fitting the right kernel (linear) and discarding a small subset of child bikes. It should be noted that the elimination of the aforementioned group within the data brought more value into the model and little to none to the demand prediction, as all of the stations are situated around a recreational lake, and nowhere else. Surprisingly enough, the models trained on the 2020 rentals dataset performed slightly better than the returns in the same year, and significantly better than both 2019 ones. This suggests that there were either less factors making up the demand during the pandemic which we were missing, or that the target variable was more dependant of the atmospheric factors (i.e. due to prolonged social isolation during lockdowns and restrictions people were more prone to get out and bike if the weather was good). The following section will further explore the limitations to the study and how further studies can improve on it.

Limitations to the study

The primary limitation to the study has been the overall low correlation of features to the target variable, although as previous studies on this subject have shown, atmospheric variables have a direct effect on the amount of bikes rented in a bike-sharing system, therefore the unsuccessful nature of this study could be attributed to the weather and historical bike rental datasets being aggregated at a too high level; as no studies have been conducted on the influence of holidays, observances, equinoxes and trade Sundays on the bike-sharing demand in Poland, it is not possible to determine the same for those features. While a machine learning model is always a generalised reflection of reality, in this case it has proven to be significantly underfitting and unable to produce reliable results on monthly data.

As the data granularity within the investigated dataset has proven to be too general, an obvious limitation to the study appears – an insufficient number of samples, which a more granular dataset set in the 2019-2020 timeframe would have solved.

Another factor which inhibited the study was limited time to gather further data and experiment with it. Given enough time, we could have tried tuning the model hyperparameters or engineer more features in order to push the scores higher. Additional datasets which could help explain the target variable could also be sought after. The following subsection will explore those possible datasets and improvements for further studies.

Possible improvements and missed opportunities

The first and foremost improvement to future studies on this subject in general, would be to continue using data with greater granularity (i.e. down to a hourly reading of bikes' state/location, or down to a transaction – between the check-out and check-in), but more specifically (for Poznań or any city where the service is operated by NextBike) to use the API, which has a limited availability and has previously been utilised in a study (Dzięcielski, Radzimski i Woźniak, 2020). A weather API (i.e. OpenWeather) would also need to be utilised, especially if the model would be productionised for use by either the operator, or the local authorities. Choosing the right one with not only the fitting level of granularity, but also a forecast that goes well into the future would be crucial to obtain accurate predictions, should an accurate model be developed.

Another opportunity for improving the model accuracy that we have missed, but could be explored in the future would be to tap into datasets (preferably over an API) on various happenings throughout the city, which could direct the traffic in one direction at a time, or disrupt it. An example of the former would be shows at cultural venues and festivals, while the latter would incorporate sporting events that take place on the city streets, like the annual marathon, triathlons, etc., where main roadways are closed off completely, or partially (i.e. when traffic is allowed to go across the path of the event if no participants are around). This would require careful examination of the data sources, as an incorrect signal about an event could potentially trigger the bikes to be shifted away from areas with a high demand for them, and incur additional costs for the operator at no benefit.

Yet another dataset that could be tapped into for more accurate real-time predictions would be the real-time tram and bus arrival times (and their location throughout the city) dubbed "Wirtualny Monitor" (Polish for "the virtual monitor"), as well as the public notices published by MPK (short for Miejskie Przedsiębiorstwo Komunikacyjne – the Municipal Transportation Enterprise) informing of tram/bus crashes and disruptions to the traffic. This could be used to drive short-term demand prediction caused by trams derailing, buses crashing, etc., when the passengers disembark their mode of transportation and wait for the operator to dispatch a temporary bus to get them to their destination, which is lengthy at times.

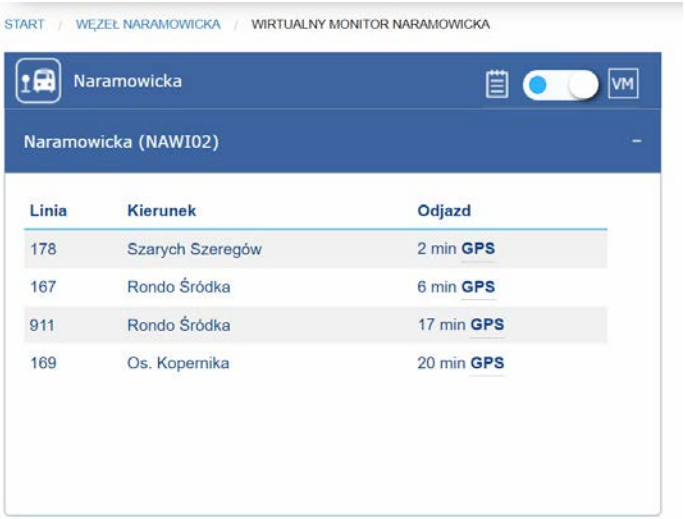

Figure 7. Screenshot of "Wirtualny Monitor" showing real-time information about arrivals based on GPS.

In order to accurately measure the demand for an element in the public transit system, it would be beneficial to also account for other modes of transport and their usage by the passengers. This does not cover just the public transit in the form of buses, trams and taxis/Ubers since the bike sharing service was launched in the city in 2012, many vehicle sharing companies have introduced their products, most, if not all, available through a smartphone app, including passenger cars, cargo trucks (i.e. Panek, Traficar) and scooters (i.e. Blinkee, Lime). Special attention should be paid to the shared scooters, as according to a recent market study the demand for this type of personal transportation devices reached demand (37700 units) twice the total number of shared bikes available in Poland in 2021 (Mobilne Miasto, SmartRide.pl, 2021). The below chart shows the percentage of the 2021 market share by each operator, the number of devices available to rent ("hulajnogi"/"hulajnóg") and count of cities where they are present ("miasta"/"miast").

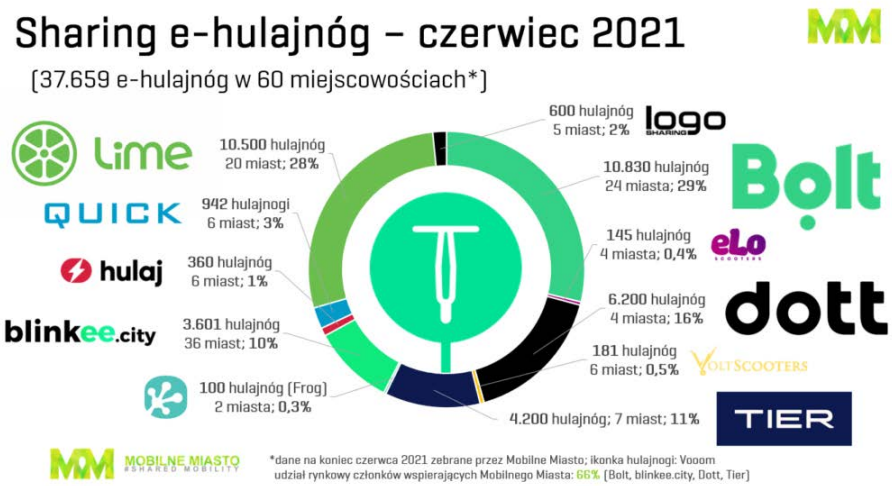

Figure 8. 2021 e-scooter market shares by operating company in Poland. Source: Mobilne Miasto / SmartRide.pl.

After gathering the best possible data, the choice of the model itself should be taken into consideration as well – several possible regression models should be built as it has been in the case of works of Ashqar, et al. (2017) and Sathishkumar & Yongyun (2020), and then compared against each other. For a better understanding of the ride sharing market, investigating a Decision Tree model could prove of much value to deepen the insights from the data, especially if the above mentioned datasets on other modes of shared transport can be tapped into.

Finally, a model of sufficient accuracy should be obtained, possible ways to productionise it should be investigate, as only after deployment the model can serve the operators, and accessing the predictions in a real working environment is key. In order to help scale the code used in the research for further use, the following points should be considered:

- Following good code practices for which Python scripts should entail, e.g.:
  - the principles of OOP (object-oriented programming) and utilising built-in functions (Zhang, 2019);
  - designing test cases and the code itself around them (Mece, Binjaku, & Paci, 2020);
- code containerisation in order to streamline its deployment on production servers or cloud machines (e.g. using Docker) (Zhang, 2019);
- engineering an ETL (Extract-Transform-Load) process and developing machine learning pipelines (i.e. using the built-in *scikit-learn* Pipeline class) (Buitinck, et al., 2013);
- carefully choosing how the model will be accessed in an environment where software is built in different languages. It can be either re-written in the target language, however "majority of languages like Java, a popular software development programming among engineers and dev-ops, compared to Python which is data scientist's toolkit, do not have great libraries to perform data manipulation and ML training, and that might cause the model outcomes to change" (Zhang, 2019). The author of the aforementioned quote suggests another approach instead, which is to build the model into a REST API, that could be then accessed by any piece of software;

- avoiding using IDEs (Integrated Development Environments) such as Jupyter Notebook in favour of plain Python scripts post the prototyping phase (Zhang, 2019).

In conclusion the subject of modelling the bike sharing demand prediction in Poznań, in either pre, mid, and post pandemic circumstances is still widely unexplored and has a lot of potential for future studies that would aim to explore it deeper.

Bibliography

Ashqar, H. I., Elhenawy, M., Almannaa, M. H., Ghanem, A., Rakha, H. A., & House, L. (2017). Modeling Bike Availability in a Bike-Sharing System. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. Naples. doi:10.1109/MTITS.2017.8005700

Basak, D., Pal, S., & Patranabis, D. (2007, November). Support Vector Regression. *Statistics and Computing (Stat Comput)*(11(10)). Retrieved from https://www.researchgate.net/publication/228537532_Support_Vector_Regression

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., . . . Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. *European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases*, 108--122.

Dzięcielski, M., Radzimski, A., & Woźniak, M. (2020). Bike-sharing system in Poznan – what will Web API data tell us? *Transport Geography Papers of Polish Geographical Society*(23(3)), 29-40. doi:10.4467/2543859XPKG.20.018.12786

European Medicines Agency. (2021, January 12). *Comirnaty - COVID-19 mRNA vaccine (nucleoside-modified)*. Retrieved from An official website of the European Medicines Agency: https://www.ema.europa.eu/en/medicines/human/EPAR/comirnaty

European Medicines Agency. (2021, June 24). *COVID-19 vaccines*. Retrieved from An official website of the European Medicines Agency: https://www.ema.europa.eu/en/human-regulatory/overview/public-health-threats/coronavirus-disease-covid-19/treatments-vaccines/covid-19-vaccines

Kleynhans, T., Montanaro, M., Gerace, A., & Kanan, C. (2017). Predicting Top-of-Atmosphere Thermal Radiance Using MERRA-2 Atmospheric Data with Deep Learning. *Remote Sensing*, *9*, 1133.

Mece, E. K., Binjaku, K., & Paci, H. (2020). The Application Of Machine Learning In Test Case Prioritization - A Review. *EJECE, European Journal of Electrical and Computer Engineering*, *4*. doi:10.24018/ejece.2020.4.1.128

Ministerstwo Zdrowia. (2021, June 24). *Narodowy Program Szczepień przeciw COVID-19*. Retrieved from Oficjalna strona Programu Szczepień przeciw COVID-19:

https://www.gov.pl/web/szczepimysie/narodowy-program-szczepien-przeciw-covid-19

Mobilne Miasto, SmartRide.pl. (2021, July 15). *E-hulajnogi. Sharing. Polska. Drugi kwartał 2021 roku*. Retrieved from Smartride.pl: https://smartride.pl/Strefa_Danych/e-hulajnogi-sharing-polska-drugi-kwartal-2021-roku/

Nikiforiadis, A., Ayfantopopulou, G., & Stamelou, A. (2020). Assessing the Impact of COVID-19 on Bike-Sharing Usage: The Case of Thessaloniki, Greece. *Sustainability*, 8215. doi:10.3390/su12198215

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 2825-2830*, 2825--2830. Retrieved from http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Sathishkumar , V. E., & Yongyun, C. (2020). A rule-based model for Seoul Bike sharing demand prediction using weather data. *European Journal of Remote Sensing*(53:sup1), 166-183. doi:10.1080/22797254.2020.1725789

Sharp, T. (2020, March 3). *Towards Data Science Blog*. Retrieved from An Introduction to Support Vector Regression (SVR): https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2

Stróżyk, A. (2020, December 16). *Pandemia koronawirusa na świecie i w Polsce - kalendarium*. Retrieved from Medicover: https://www.medicover.pl/o-zdrowiu/pandemia-koronawirusa-na-swiecie-i-w-polsce-kalendarium,7252,n,192

Szokalska, A. (2019). *Rocznik meteorologiczny 2019*. Warszawa: Instytut Meteorologii i Gospodarki Wodnej - Państwowy Instytut Badawczy. Retrieved from https://danepubliczne.imgw.pl/data/dane_pomiarowo_obserwacyjne/Roczniki/Rocznik%20meteorologiczny/Rocznik%20Meteorologiczny%202019.pdf

Szokalska, A. (2020). *Rocznik meteorologiczny 2020*. Warszawa: Instytut Meteorologii i Gospodarki Wodnej - Państwowy Instytut Badawczy. Retrieved from https://danepubliczne.imgw.pl/data/dane_pomiarowo_obserwacyjne/Roczniki/Rocznik%20meteorologiczny/Rocznik%20Meteorologiczny%202020.pdf

Thirumalai, C., & Koppuravuri, R. (2017). Bike Sharing Prediction using Deep Neural Networks. *International Journal on Informatics Visualization*(1(3)). doi:10.30630/joiv.1.3.30

Thorsen, S. (2021, May 10). *Holidays and Observances in Poland*. Retrieved from timeanddate.com: https://www.timeanddate.com/holidays/poland/

Ustawa o ograniczeniu handlu w niedziele i święta (Sejm Rzeczypospolitej Polskiej January 10, 2018).

Wielechowski, M., Czech, K., & Grzęda, Ł. (2020). Decline in Mobility: Public Transport in Poland in the time of the COVID-19 Pandemic. *Economies*(8(4)), 78. doi:10.3390/economies8040078

World Health Organization. (2020, February 11). *@WHO*. Retrieved from Twitter: https://twitter.com/WHO/status/1227248333871173632

World Health Organization. (2020, June 29). *Listings of WHO's response to COVID-19*. Retrieved from World Health Organization Web site: https://www.who.int/news/item/29-06-2020-covidtimeline

Zarząd Transportu Miejskiego. (2020, April 9). *Poznański Rower Miejski*. Retrieved from Przedłużenie zakazu korzystania z rowerów miejskich do 19 kwietnia: https://poznanskirower.pl/przedluzenie-zakazu-korzystania-z-rowerow-miejskich-do-19-kwietnia/

Zarząd Transportu Miejskiego. (2020, May 04). *Poznański Rower Miejski*. Retrieved from Poznański Rower Miejski zostanie ponownie uruchomiony o północy z wtorku na środę (z 5 na 6 maja): https://poznanskirower.pl/poznanski-rower-miejski-zostanie-ponownie-uruchomiony-o-polnocy-z-wtorku-na-srode-z-5-na-6-maja/

Zarząd Transportu Miejskiego. (2020, July 15). *Poznański Rower Miejski*. Retrieved from Darmowe przejazdy rowerami Nextbike dla medyków do połowy września!: https://poznanskirower.pl/darmowe-przejazdy-rowerami-nextbike-dla-medykow-do-polowy-wrzesnia/

Zarząd Transportu Miejskiego. (2020, November 20). *Poznański Rower Miejski – wydłużenie sezonu do 23 grudnia*. Retrieved from ZTM's official website: https://www.ztm.poznan.pl/pl/aktualnosci/poznanski-rower-miejski-wydluzenie-sezonu-do-23-grudnia

Zarząd Transportu Miejskiego w Poznaniu. (2021, June 23). *Poznański Rower Miejski*. Retrieved from Zarząd Transportu Miejskiego w Poznaniu Web site: https://www.ztm.poznan.pl/pl/komunikacja/rowery/

Zhang, D. (2019, October). *Case study: Productionizing Machine Learning Pipelines*. Retrieved from Toyota Connected:

https://www.toyotaconnected.net/insights/productionizing-machine-learning-pipelines

Table of figures

Table of tables