Illinois Institute of Technology

Divvy Bike Analysis

Manish Tandel (A20442343)
Kartik Kini (A20449710)
Mohit Roy (A20434544)
Harshit Paliwal (A20449708)
Radhika Barbole (A20450289)

CSP 571: Data Preparation and Analysis
Professor Adam McElhinney

INTRODUCTION

Divvy is a bicycle sharing system in the city of Chicago and two adjacent suburbs operated by Lyft for the Chicago Department of Transportation. This concept has been gaining quick popularity around the world. The system consists of a fleet of bikes placed in a network of docking stations. Each of these docking stations consists of multiple docks that are used to pick up or drop bikes. These bikes can be rented by people and then returned to any docking station. These trips are charged on the basis of the time used and not the distance traveled.

Divvy is an ever-growing business and hence, it is important to keep the customers intact and attract new customers with the best possible services. In order to achieve this goal, it is necessary to analyze the trends of users and make useful inferences. Moreover, a deep analysis of the customer trip history can provide very useful insights on how they can efficiently improve.

Through this project, we have provided solutions to four problems that would provide a business gain for Divvy. Following are the four problem statements:

· **Predicting which station needs more docks on the basis of the traffic at the station. This is also by keeping in mind the revenue generated v/s the cost required to build more docks**. It is often observed that at the prime stations, there is no availability of an empty parking dock at the station. This issue causes the customer to either wait for the dock to be empty or find a new nearby station with an empty dock. In both scenarios, the customer may tend to rethink his/her choices to rent a Divvy bike the next time to save time.

· **Identifying the need and feasibility (revenue) for new stations between 2 stations for providing parking spaces at heavy traffic stations. This is to ensure that Divvy does not lose customers, based on the parking docks' unavailability**. As explained above in the previous business issues, the most occurring issue could be the lack of parking space at the prime locations. But building a new station must be done by weighing the cost required to build a new station and the profit it might generate.

· **At what time/season can the bikes be put for maintenance so that the Divvy company does not lose any revenue by putting the bikes into maintenance.** On a given day, which bikes can be put to maintenance based on the bike usage (trip duration), the weather, and the need for the bike on that day. Considering the need of the bike ensures that Divvy doesn't lose any customers or the profits.

· **Predicting the need for bikes according to the season in Chicago in order to ensure that the customers do not have to wait for more bikes, by ensuring the increase or decrease in the number of bikes at the docking station to maintain the profit.** As per the season, the need for bike availability varies. Keeping the bikes available during the high demand seasons and accordingly can help Divvy optimize its bike availability strategies.

DATA ACQUISITION

Data Acquisition is the most primitive step in a project development cycle. Often challenges are faced in acquiring the right data as required to you may or may not be available. This can be because of confidentiality and legal reasons, access to data is restricted. We were fortunate to have found the primary dataset on the Divvy website with public access. Though it came along with few restrictions like missing revenue, profit attributes.

Hence primary datasets often come with some shortcomings and need the support of secondary datasets to provide more in-depth information. The datasets used for our analysis and their description is as follows:

1. **Primary Dataset-** The primary dataset that we have used is a "Divvy trips quarterly dataset". It is a CSV file and contains data from the year 2013 to the year 2019. It contains attributes like TripID (unique id for each trip), BikeID (unique for each bike), StationID (unique id for each station) start date, start time, end date, end time, gender, birth year. This dataset was obtained from the City of Chicago data portal.

2. **Secondary Dataset-** We used a couple of secondary datasets to add to the primary dataset.

1. Docks dataset: The primary dataset did not contain any information about the number of docks present on each station and was necessary to predict the increase/decrease in the number of docks based on the high traffic and need of the station. This dataset is a CSV file and contains attributes like StationID, number of docks, latitude, and longitude of the stations.
2. Weather Dataset: The primary dataset does not talk about the weather conditions and was necessary to predict when to put bikes on maintenance and predicting the need for bikes on an hourly basis. Weather is an important factor in predicting both these things. Weather data is acquired from the "Dark Sky" API. The dark sky API is the easiest, most advanced weather API on the planet. It provides a minute update of the weather for any place of your need. We fetched the attributes like date and weather icon attributes from the API. The weather icon is an attribute that gives a weather summary of the day in terms of icons like cloudy, clear, partly cloudy, and so on.
3. Bikes availability dataset: In order to get available bikes at the station docks, we made use of RSocrata API to fetch the attributes.

Once the data is acquired, it paves the path for getting the data ready for the model development. To get the data ready, the data needs to be prepared by cleaning the data and manipulating it according to our needs.

DATA PREPARATION

Data preparation is the step followed by data acquisition. Just like every dataset, our dataset had various aspects that needed attention. For example missing values, column name alignment, and more. To handle this the data preparation was carried out in four steps: Primary Dataset Manipulations, Secondary Dataset Manipulations, New calculations from an existing dataset, and Transformations on the attributes of the dataset. The goal of preparing the dataset is to ensure that there is no discrepancy in the data that would negatively affect the model in any way.

1. Primary Dataset Manipulations:

· The data was present in the quarterly format. For example- for the year 2018 the data was present in the form of 4 quarters. This was the case with the data for all the years. To deal with this, all the data of the required years were merged into a single CSV file.
· Since, the data was not available in a single CSV file. It was discovered that the column names in these quarterly formats of data were not aligned. For example- the column names in the 2nd quarter of 2018 did not align with the column names of the rest of the data.
· Missing values is a very common and prominent issue while dealing with large data. Missing values were found in attributes like gender, birth year, trip duration. While dealing with trip duration was necessary as it is an important attribute and an independent variable in a couple of models. The method to deal with missing trip duration is making use of the start time and the end time to calculate the trip duration. Moreover, the missing values of gender and birth year were handled as they did not play any role in the model we built. Additionally, a station was found with StationID as 1 and it meant special events. Since the number of trips with such StationID was less than 100 in a dataset of around a million rows. Such entries were ignored.
· Extracting only the required columns: Not all attributes are required to formulate a model to answer a specific problem. Hence, extracting the required attributes from the whole dataset not only makes it easier to analyze but makes it computationally efficient. For example: For the problem statement dealing with the maintenance of bikes, attributes like gender, the birth year seemed to be of no significance and hence were dropped out.

2. Secondary Dataset Manipulations:

· Merging the dock dataset: The dock dataset contained some additional dock information that was not present in the primary dataset like the number of docks present in each station and the latitude and longitude coordinates of stations. These attributes were mapped to the primary dataset by using the StationIDs.
· Figuring out the empty docks and the bikes by the minute was achieved by making use of the RSocrata API. Using this API, only the details of the required stations were fetched. A process like this is a good practice when it comes to dealing with large-scale real-time data. This ensures that we only fetch the required attributes from the API that declutters the data to a great extent.

· Fetching the required attributes of the weather data from Dark Sky API was an important step. Dark sky API is a very rich weather API that includes innumerable attributes of almost all major cities around the globe. Only the attributes like date and weather icon were extracted and stored in the CSV file as per the requirements. Weather icon gives the weather summary of the day in a word like snow, cloudy, rainy, and so on. This data was mapped to the primary dataset by start date attribute to the Date attribute of the weather dataset.

3. New calculations from the existing attributes:

· Attribute like revenue isn't present in the primary dataset or the secondary dataset. And due to confidentiality and legal reasons, such data is not available for public use. We calculated the revenue from attributes like trip duration and customer type. This attribute was necessary to provide analysis for predicting the feasibility of building new docks and stations by keeping revenue in mind.
· Attribute like distance traveled is not available in the primary dataset or on any data sources. The traveled distance was calculated using the attributes like latitude, longitude of the origin, and destination points.
· To be able to better predict the need for new docks on the stations a new attribute named Dock_difference was calculated in order to visualize the difference between arrivals and departures at each station.
· Trip duration was calculated from the start time and the end time of the trips.

4. Transformations on the existing attributes:

· Date attribute is a very important attribute for the kind of analysis we aimed to provide. Hence, bringing varied timestamp formats to one format is a transformation performed on the date attribute. This was achieved by making use of an R library named 'lubridate' that served the purpose with ease.
· The start time and the end time of the trips were present in the columns of start date and end date respectively. For analysis and extracting more information from the start time and the end time of the trip, the Data and time were split into two different columns using 'str_split'.
· Missing rows is the most common type of data discrepancy and only those missing values should be handled that are relevant to the analysis. We handled the missing rows by eliminating the rows with such values because the number of missing values to the relevant attributes was around 50 rows in millions of rows.
· A lot of data type conversions were made. For example, converting the trip duration from string to numeric and changing it to minutes.

EXPLORATORY DATA ANALYSIS

After the data is cleaned and before the models are built, one of the most important steps in performing exploratory data analysis. Exploratory data analysis is an approach to analyzing datasets to be able to provide some insights on the data. These insights are usually provided with visuals and graphical representations. The purpose of such analysis is to achieve one or all of the following things:

1. Maximize insights into data sets
2. Extract important variables
3. Uncover underlying structure
4. Determine optimal factor settings
5. Test underlying assumptions
6. Detect outliers and anomalies
7. Develop parsimonious models

We will now have a look at all the insights drawn from our datasets with the visual representation.

1. Bar graphs and Histograms: The Following are a few bar graphs and histograms and the insights drawn from them.

(Figure 4.1)

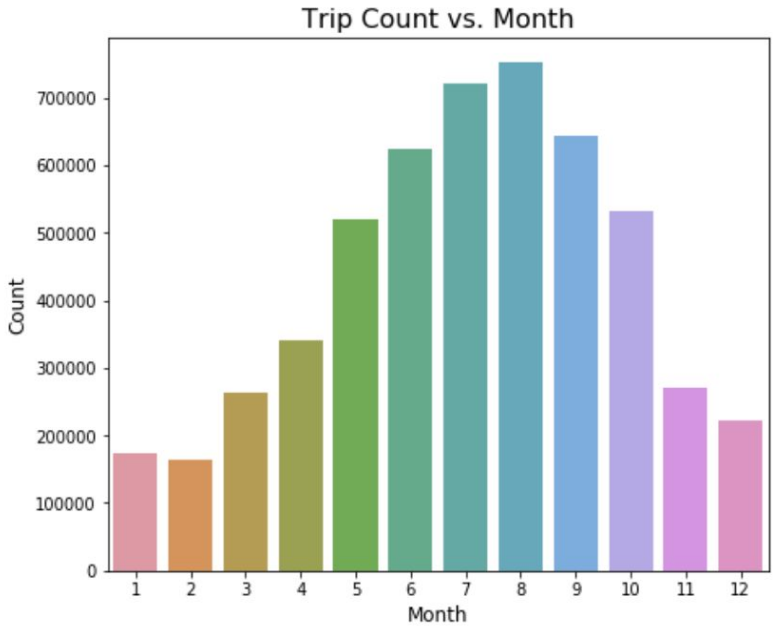


Figure 4.1 shows a histogram that represents the count of trips v/s month.

Analysis: From 4.1 we can see that months are represented in numbers from 1 to 12 and the count is given on a scale of 100000. It is quite evident that the usage of Divvy rental bikes is maximum in month number 7 and 9, which means July and August. Which makes perfect sense as these months fall between the period where there is no extreme heat of the May month or extreme cold of the November. And this aligns with the fact that Chicago is a windy city known for its chilly climate and so people prefer to use bikes when the temperature is pleasant.

(Figure 4.2)

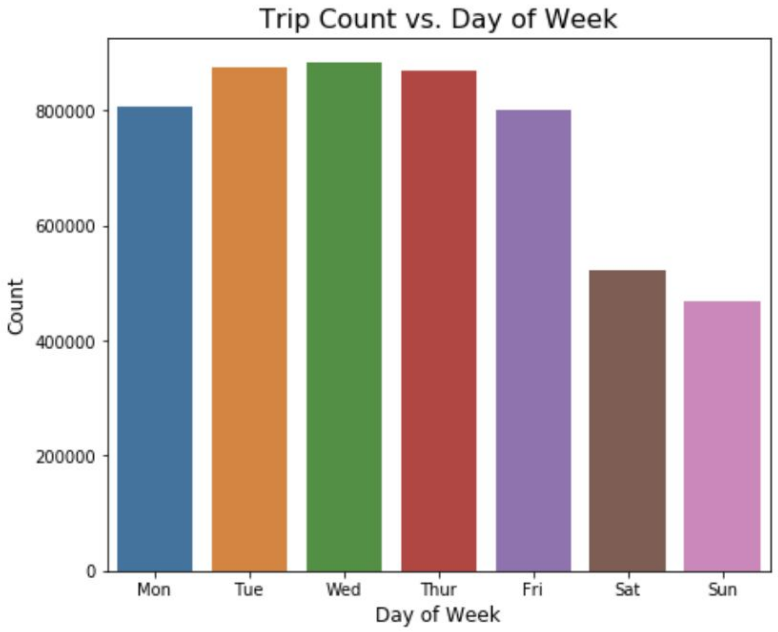


Figure 4.2 is a histogram that represents the number of trips according to the day of the week.

Analysis: From figure 4.2 we can see that the days of the week with the highest number of trips are on Tuesday, Wednesday, Thursday. Whereas, Monday and Friday are not too far behind. On the contrary, the number of trips on the weekend is comparatively less. This explains the trend of users and gives us a little insight into the type of customers Divvy has. For example, this explains that the majority of Divvy users use bike services to go to work or come back from home. Only a few use the services on weekends for recreational activity.

2. Heat Maps: Following are a few heat maps and the insights drawn from them.

(Figure 4.3)

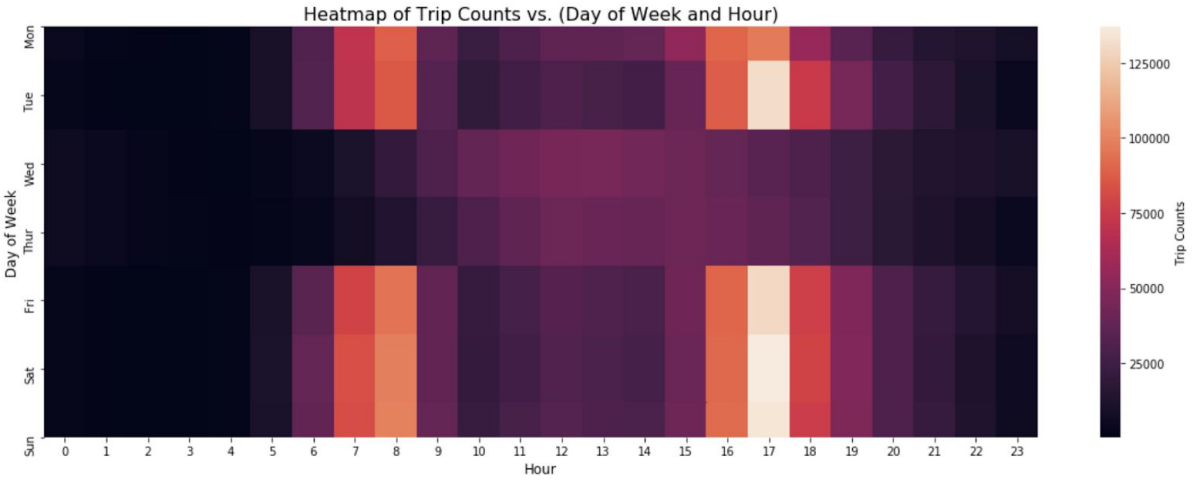


Figure 4.3 shows a heat map that represents the number of trip counts on the basis of the day of the week and the time.

Analysis: The scale of the heatmap shows that the darkest shade of brown/purple represents the least number of trips and then the gradual increase from the dark to lighter shades represents the increase in the number of the trips. After observing the heat map above we can see that the most number of trips have been taken first, during the time span of 7-8 AM and then during the time span of 4-6 PM. This explains that the Divvy bikes are rented by the people while going to or coming from the office, schools, university, or work and hence it can be associated with the working hours of people.

(Figure 4.4)

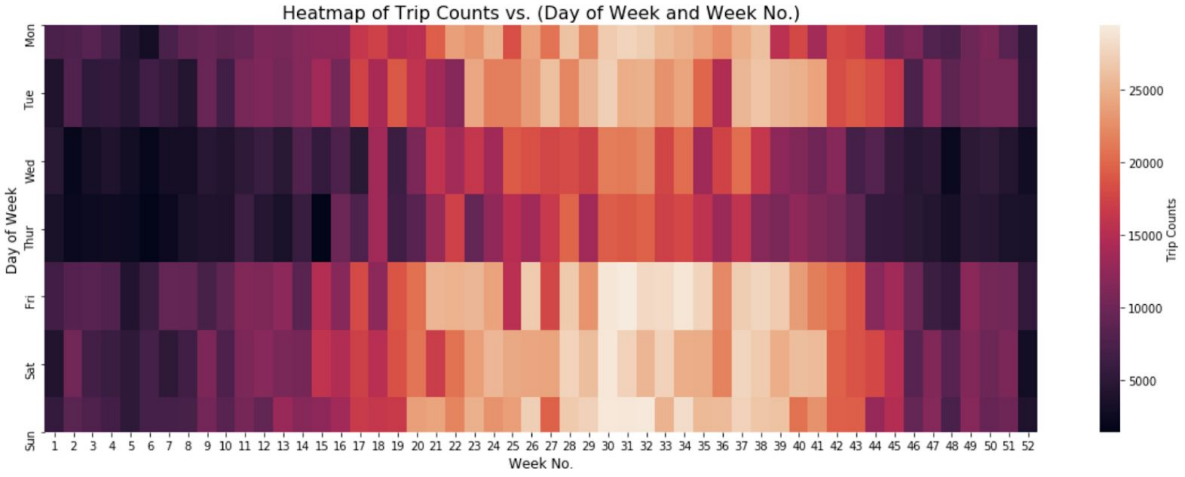
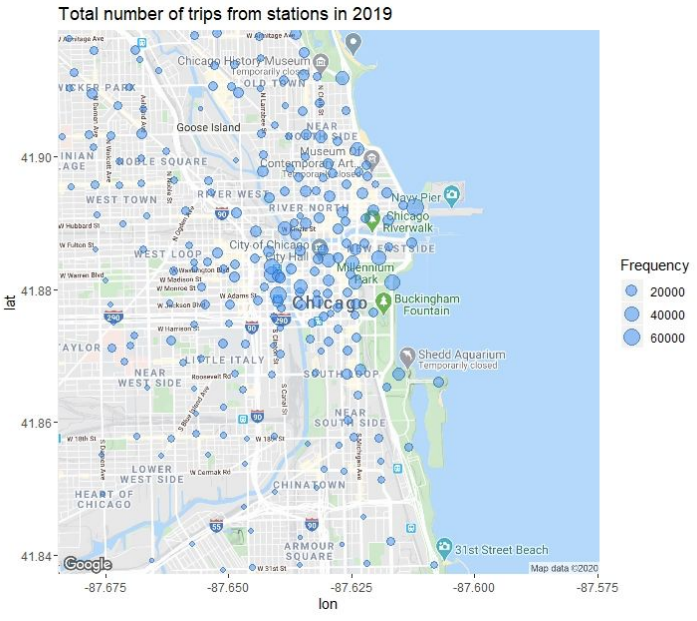


Figure 4.4 shows a heat map that represents the number of trip counts on the basis of the week of the year and the day of the week.

Analysis: The scale of the heatmap shows that the darkest shade of brown/purple represents the least number of trips and then the gradual increase from the dark to lighter shades represents the increase in the number of the trips. After observing the heat map above we can see that the most number of trips have been taken during the span of week 20-40 and on Wednesday and Thursday it's the highest. This aligns with the analysis obtained from the histograms above.

3. Spatial Distribution plots: The following are a few spatial distribution plots and the insights drawn from them.

(Figure 4.5)



(Figure 4.6)

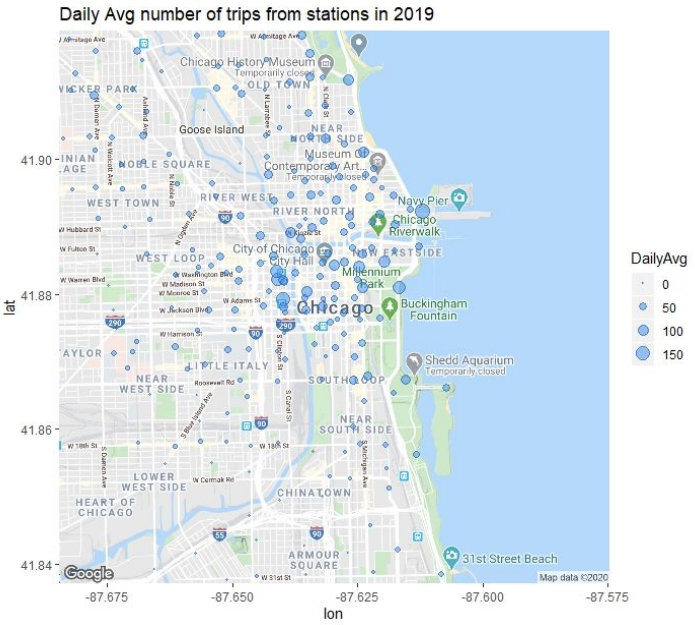
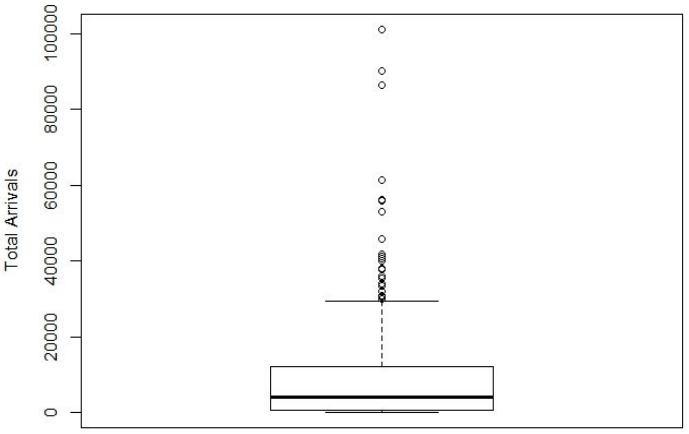


Figure 4.5 and Figure 4.6 are spatial distributions
plots that represent the number of trips and the daily average number of trips on the stations respectively.

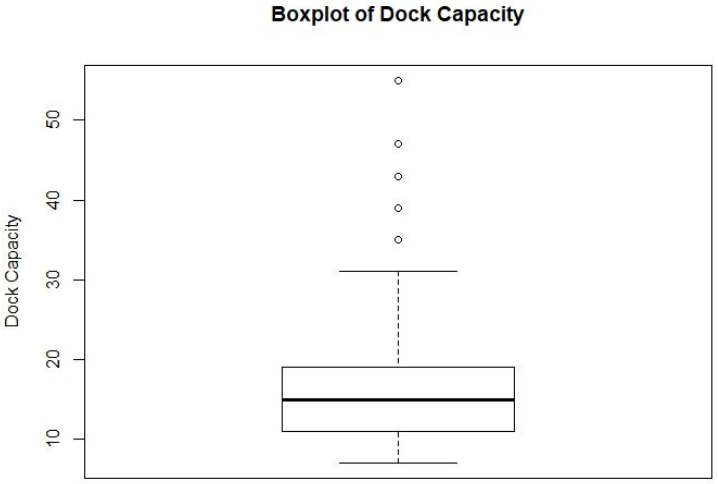
These graphs plot the frequency of trips on the map of Chicago city and the Divvy stations present in Chicago.
Analysis: The spatial distribution plot shows the total number of trips from all divvy stations and it can be observed that central Chicago is the most used location and very few customers take Divvy bikes in the Southside of Chicago which makes it a potential area to advertise Divvy service in that region. It can be observed that most of these stations are in Central Chicago so ultimately new dock stations if needed to be added should be placed there.

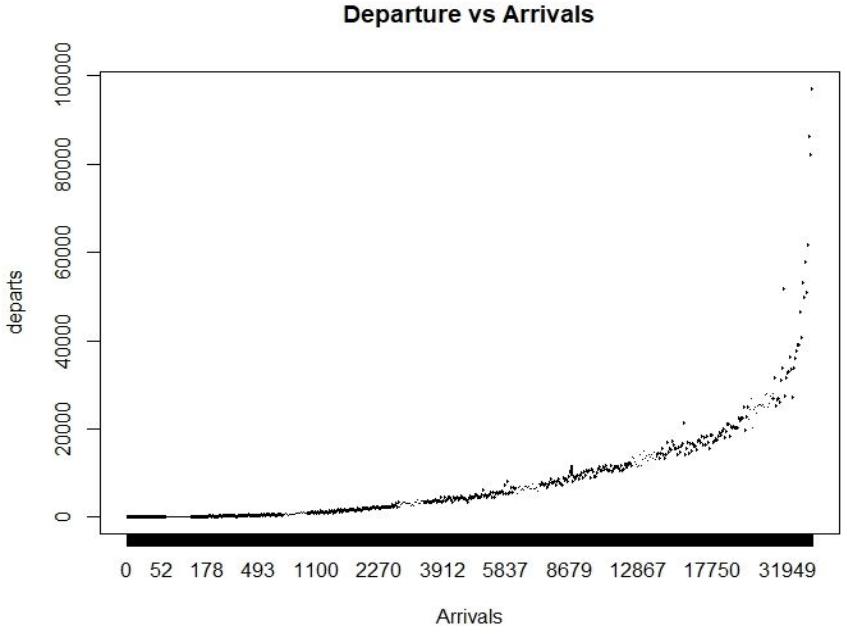
3. Box plots: The following are a few box plots and the insights drawn from them.

(Figure 4.7)



(Figure 4.8)





(Figure 4.9)

Figure 4.7 represents boxplots for analyzing the number of arrivals. Figure 4.8 represents the boxplot for the number of docks at stations. Figure 4.9 represents the relation between the arrivals and the departures.
Analysis: Figure 4.7, 4.8, and 4.9 togetherly shows that there remains a positive correlation among the arrivals and the departure of the trips where the dock capacity maintains a bandwidth with a couple of outliers. This helps Divvy to be able to know better about the need for building new docks or removing the existing ones.

DATA MODELS

After thoroughly cleaning the data and performing concentrated analysis on the data and deriving insights from them. The next step in the development of the Data Science cycle is building the models. As explained in the introduction we have provided solutions to four different business problems, we have built three models per problem statement.

1. Problem Statement: **Predicting the need for building a new station**
Firstly the models were implemented for the need and feasibility for a new dock station. Three models are implemented namely Multi Regression, KNN classification, and logistic regression. For all these three models we assumed 400$ for each new dock and median docks in each station to be around 23.

Multi Regression and Correlation: This model was applied to find out the relationship between variables to determine our target variable and predictors with the help of a correlation plot. The departure and arrival for all 621 unique stations and its difference were calculated to predict the excess or deficit in docks. With our assumed model Divvy needed to add 22 extra stations which would require about 200k $ which is well in the budget of Divvy.

K- nearest neighbor classification: Using optimal k value we could categorize Dock_diff in the manner of 'Yes' or 'No' (1 or 0) depending on if they needed a new dock station or not. Conclusion: We found out that 25 extra dock stations were required which would cost Divvy around 230k $ if dock stations are set up as per this prediction model.

Logistic Regression: With this model obtained probabilities were calculated only when greater than 0.8 and to predict the number of dock stations required based on predictor variables duration.mins, to_station_id, Longitude, Latitude, Total.docks identified using a correlation matrix. This model gave 29 new dock stations that require additional dock stations. The total estimated cost of building these 29 dock stations is around 260k $ if dock stations are set up as per this prediction model.

Evaluation Parameter: Accuracy in measurement of a set, refers to the closeness of the measurements to a specific value.

Station.Name	departs	arrivals	ID	Total.Docks	Dock_diff
Columbus Dr & Randolph St	51693	30631	195	55	-21062
Franklin St & Monroe St	53128	45856	287	31	-7272
Stetson Ave & South Water St	21488	15917	264	19	-5571
Orleans St & Merchandise Mart Plaza	46414	41363	100	35	-5051
Desplaines St & Kinzie St	33736	29964	56	27	-3772
Wacker Dr & Washington St	31689	28248	18	19	-3441
Clark St & Lake St	26996	23623	38	27	-3373
Wells St & Walton St	11810	8750	46	19	-3060
Theater on the Lake	24985	22312	177	31	-2673
State St & Randolph St	27809	25412	44	27	-2397
Artesian Ave & Hubbard St	8253	5867	376	35	-2386
Kingsbury St & Erie St	36348	34053	74	23	-2295
Fairbanks Ct & Grand Ave	27951	25783	24	15	-2168
Franklin St & Lake St	25005	22889	164	27	-2116

Result: This shows the output result for the KNN model.
Conclusion: Classification gives the best result because if we remove outliers the remaining dock stations that require new dock stations between them are 24 which is closest to the prediction of KNN classifier. Divvy can either add in more docks at particular stations that will attract new customers. So it is better to add in new stations between the predicted station names.
2. Problem Statement: **Predicting the need for building new docks on stations.**

The entire purpose of this problem statement is to address the convenience of the end-user by identifying those stations where there is a need to add or remove docks on the basis of the amount of traffic and number of trips taking place from a station. The algorithms used to obtain the required results are K-nearest neighbor classification, Logistic Regression, Naive Bayes algorithm. The data from the year 2017 and 2018 formed the training dataset whereas the data from the year 2019 formed the testing dataset. End result: List of stations with Dock Difference stating if that station needs to add docks or remove docks. Along with revenue needed in the case, docks are to be added and revenue saved in case docks are to be reduced. The assumption made: Revenue from each trip is calculated based on a predetermined cost of $3 / 30 minutes. The cost of setting up a dock in the station is $400.

K- nearest neighbor: Assuming $400 for each dock using KNN model classification it is estimated that Divvy can save $ 3M per annum if they set up docks as per prediction made by this model.

Logistic Regression: Assuming $400 for each dock using logistic regression, it is estimated that Divvy can save$ 4,10,000 per annum if they set up dock as per prediction made by this model.

Naive Bayes Model: Assuming $400 for each dock using logistic regression, it is estimated that Divvy can save $ 15,02,800 per annum if they set up dock as per prediction made by this model.

Evaluation Parameter: Accuracy in measurement of a set, refers to the closeness of the measurements to a specific value.

ID	Station Name	dock_diff	cost
418	Carpenter St & 63rd St	-20.0	$-8.0k
249	Claremont Ave & Hirsch St	12.0	$4.8k
322	Clark St & Elm St	-20.0	$-8.0k
168	Pulaski Rd & Lake St	-12.0	$-4.8k
316	Racine Ave & Congress Pkwy	4.0	$1.6k
300	Clark St & 9th St (AMLI)	4.0	$1.6k
75	Austin Blvd & Chicago Ave	32.0	$12.8k
480	Clark St & Touhy Ave	-4.0	$-1.6k
334	Morgan St & Polk St	4.0	$1.6k
220	Cityfront Plaza Dr & Pioneer Ct	8.0	$3.2k

Result: This shows the output result for the KNN model.

Conclusion: Out of three ML algorithms KNN Classification gives the best result. The majority of stations that needed upgrade in dock numbers are in Central Chicago where most probably daily office-goers must be using divvy. Whereas stations that need to cut short on the number of docks are also identified which can be used to save money and use them for stations where there is a need to increase docks.

3. Problem Statement: **Predicting the maintenance for bikes considering the weather and revenue.**

To implement this problem statement, we have predicted the number of bikes and the bikeIds of the bikes that can be put for maintenance on a given day. The algorithms used to obtain the required results are Decision Tree Algorithm, the Random Forest Algorithm, and the Support Vector Machine algorithm. The dependent variable for these models is Maintenance(0 = no & 1= yes) and the independent variables required to build the model are weatherIcon, Date, BikeId, Bike demand on that date, total duration traveled by that particular bike and number of trips completed by the bike. The data from the year 2017 and 2018 formed the training dataset whereas the data from the year 2019 formed the testing dataset. Assumption: We have assumed 40 hours completed by the bike or 100 trips completed as a threshold that qualifies or disqualifies a bike for maintenance. It is also taken into consideration that if the weather of the day is snowy or rainy, all the bikes selected for maintenance will be taken care of the next day (by considering the weather of that day too). If bad weather persists, the bikes will be put on maintenance on the next optimum weather day.

End Result: After providing the Date to the model, it returns the bike ids that need to be put on maintenance based on the weather of that day.

Decision Tree: On 2019-01-17, using Decision Tree it is estimated by the model that Divvy can put 7 bikes on maintenance to ensure apt availability. Accuracy for decision tree was observed to be 82.15%

Random Forest: On 2019-01-17, using Random Forest it is estimated by the model that Divvy can put 3 bikes on maintenance to ensure apt availability. Accuracy for decision tree was observed to be 76%

Support Vector Machine: On 2019-01-17, using SVM it is estimated by the model that Divvy can put 13 bikes on maintenance to ensure apt availability. Accuracy for decision tree was observed to be 69.27%

(Figure 5.1)

```
> Visualize("2019-01-17")
      bikeid Maintenance   MainDate Weather_Icon
44295   5577           1 2019-01-17       cloudy
44296   6041           1 2019-01-17       cloudy
44297    755           1 2019-01-17       cloudy
44300   6150           1 2019-01-17       cloudy
44301   5299           1 2019-01-17       cloudy
44302   1119           1 2019-01-17       cloudy
44304   4066           1 2019-01-17       cloudy
> Visualize("2019-02-03")
      bikeid Maintenance   MainDate      Weather_Icon
44371   4800           1 2019-02-03 partly-cloudy-day
44372   1021           1 2019-02-03 partly-cloudy-day
44373    369           1 2019-02-03 partly-cloudy-day
44376    866           1 2019-02-03 partly-cloudy-day
44377   6355           1 2019-02-03 partly-cloudy-day
44378     87           1 2019-02-03 partly-cloudy-day
44380   1581           1 2019-02-03 partly-cloudy-day
44381   1331           1 2019-02-03 partly-cloudy-day
44382   4601           1 2019-02-03 partly-cloudy-day
> Visualize1("2019-02-18")
       MainDate Weather_Icon
44708 2019-02-18        snow
> Visualize("2019-02-08")
[1] bikeid      Maintenance  MainDate     Weather_Icon
<0 rows> (or 0-length row.names)
```

Evaluation Parameter: Accuracy in measurement of a set, refers to the closeness of the measurements to a specific value.

Result: This shows the output result values for the Decision Tree model.

Conclusion: Out of three algorithms, the Decision Tree model gives the best result with an accuracy of 82.15%. It then rightly predicts the bikeIds of the number of bikes that can be put on maintenance on a given day considering the weather.

4. Problem Statement: **Predicting the need for bikes according to the season in Chicago**
To implement this problem statement, we have predicted the number of trips on an hourly basis for the most frequent station. The algorithms used to obtain the required results are Lasso Regression, Ridge Regression, and elastic net regression. The dependent variable for these models is the number of trips and the independent variables required to build the model is the weather, hour, date, and available bikes. The data from the year 2017 and 2018 formed the training dataset whereas the data from the year 2019 formed the testing dataset. Assumption: The analysis has been carried out on the most frequent stations, a station that contains the highest number of trip starts. In our case, it's a station with ID 35.

Lasso Regression: Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). The need to use this in this particular problem statement was because the output is in a continuous format (hourly number of trips). The RMSE value obtained for this model is 0.6183471.
Hyperparameter for Lasso : Lambda: 0.001995262. Tested for lambda between [0.001,100]

Ridge Regression: Ridge regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity (correlations between predictor variables). This was the case with our independent variables like the weather and date. The RMSE value obtained for this model is 0.6189368.
Hyperparameter for Ridge : Lambda: 0.001584893. Tested for lambda between [0.001,100]

ElasticNet Regression: Elastic net is a regularized regression method that linearly combines the L_1 and L_2 penalties of the lasso and ridge methods. It handles the penalties of lasso and ridge and hence is considered to be most optimum out of the Lasso, Ridge, and ElasticNet. The RMSE value obtained for this model is 0.6180832.
Hyperparameters for ElasticNet: alpha = 0.48, lambda = 0.00298, method = "repeatedcv", number = 10, repeats = 5, search = "random" **.**

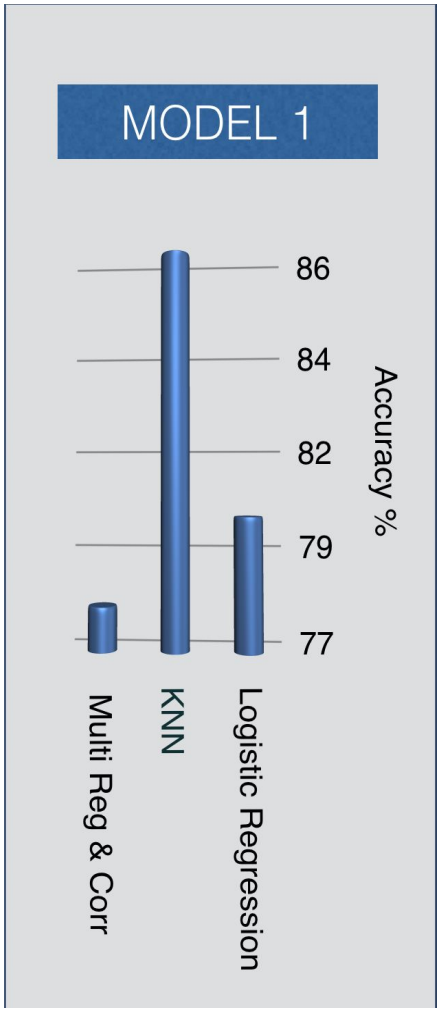
Hyperparameter: The amount of the penalty can be fine-tuned using a constant called lambda (λ). Selecting a good value for λ is critical. When λ=0, the penalty term has no effect, and ridge regression will produce the classical least square coefficients.
Evaluation Parameter: Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; it tells you how concentrated the data is around the line of best fit.
Result: This shows the output result RMSE value for the elastic net model.

```
> predictions_test <- predict(elastic_reg, Test)
> eval_results(Test$trips, predictions_test, Test)
[1] 0.6180832
```

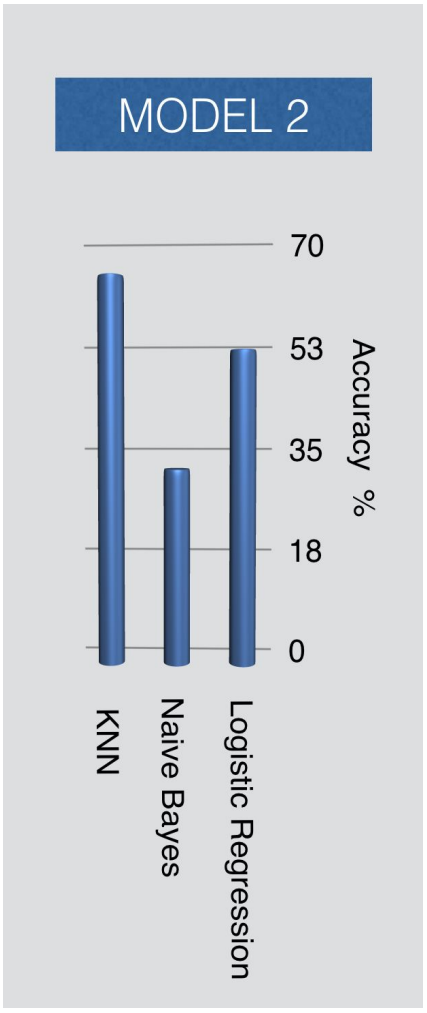
Conclusion: Out of three algorithms ElasticNet Regression gives the best result. This can be inferred in comparison by the Root Mean Square Error. The least value of RMSE is for ElasticNet.

MODEL COMPARISON

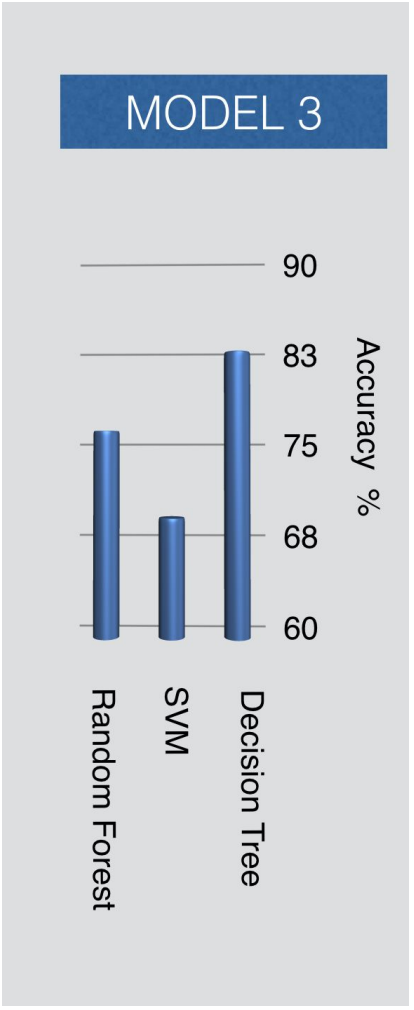
Following are the graphs that compares the various models based on accuracy as a metric measure:



(Figure 6.1)



(Figure 6.2)



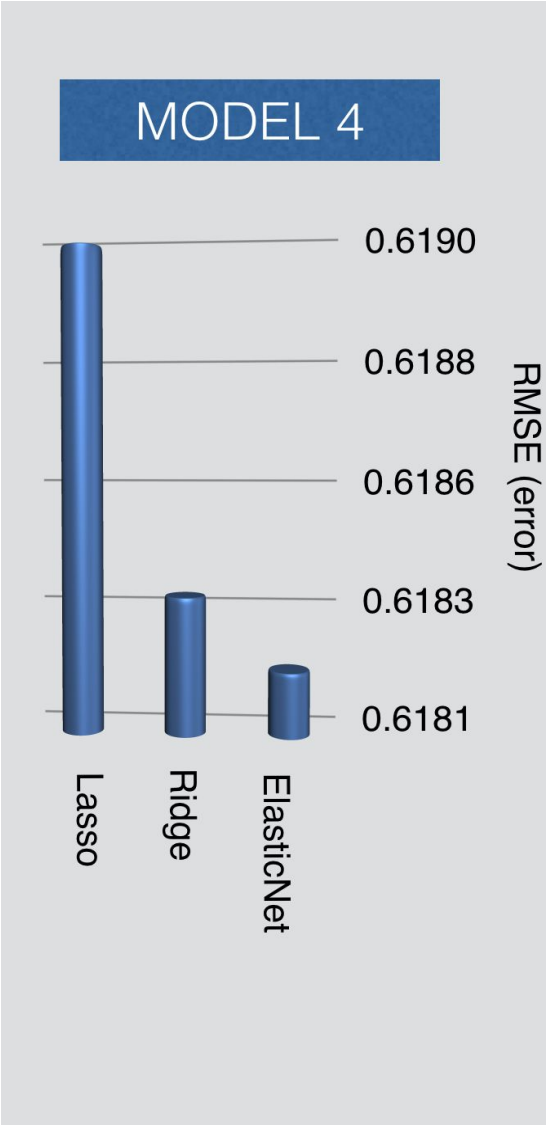
(Figure 6.3)

Models set for problem 1: The models used for the analysis and prediction of the need and feasibility of building new stations are Multi Regression and Correlation, K-nearest neighbor classification, Logistic Regression. As we can see in Figure 6.1, the three models are being compared on the basis of the accuracy measure. The highest accuracy is observed for K-Nearest Neighbors with an accuracy of 86%. Followed by Logistic regression with an accuracy of 80% and the least optimum model of the three is multi regression and correlation with an accuracy of 78%.

Models set for problem 2: The models used for the analysis and prediction of the need and feasibility of building new docks are Naive Bayes, K-nearest neighbor classification, Logistic Regression. As we can see in Figure 6.2, the three models are being compared on the basis of the accuracy measure. The highest accuracy is observed for K-Nearest Neighbors with an accuracy of 63.28%. Followed by Logistic

regression with an accuracy of 53% and the least optimum model of the three is Naive Bayes with an accuracy of 22.85%.

Models set for problem 3: The models used for the analysis and prediction of the bikes that need to be put on maintenance are Decision Tree, Random Forest, and Support Vector Machine. As we can see in Figure 6.3, the three models are being compared on the basis of the accuracy measure. The highest accuracy is observed for the Decision Tree model with an accuracy of 82.51%. Followed by Random Forest with an accuracy of 76% and least optimum model of the three is the Support Vector Machine model with an accuracy of 69.27%.



Following are graphs that compare the three models built for problem 4, based on RMSE as a metric measure:

Models set for problem 4: These models used for the analysis and prediction of the number of trips on an hourly basis, considering the weather are Lasso Regression, Ridge Regression, and ElasticNet Regression. As we can see in Figure 6.4, the three models are being compared on the basis of the Root Mean Square Error(RMSE) measure. Least error is observed for ELasticNet models as expected with an error value of 0.6180832. Followed by Ridge regression with an RMSE value of 0.6189368 and the least optimum model of the three is Lasso regression with an RMSE value of 0.6183471.

Conclusion: It can be observed that various problem statements need different models for optimum results. And different models are evaluated on different parameters. For example, the best use of metrics for evaluating the Lasso, Ridge, and ElasticNet is RMSE rather than accuracy or recall.
(Figure 6. 4)

CONCLUSION

Hence, from this process of Data Science project development, we can come to the conclusion that:

1. While building models remains a challenge, preparing the data and getting it ready for the models proves to be a bigger challenge.
2. Data acquisition proves to be a task as confidentiality sometimes obstructs obtaining the parameters required for analysis.
3. For different requirements, different algorithms prove to be optimum. This was observed from the various algorithms used for various problem statements. And different algorithms prove to be optimum for different problem statements.
4. In the first two problems, KNN classification seems to be optimized to provide a prediction for the number of new docks at the station as well as a prediction for new stations.
5. The last two problems were optimally solved with a different approach as, decision tree rightly predicted which bikes should be put to maintenance and when.
6. Whereas, ElasticNet optimally predicted the availability of bikes on an hourly basis.

FUTURE SCOPE AND LESSONS LEARNED

During the time span of the project we have learned a couple of lessons:

1. Filtering and fetching out the required attributes from API is a more efficient and easier way to deal with larger data than dealing with CSV data files.
2. Date manipulations can be very tricky and sensitive depending on the problem statement needs. (For example the inculcation of day-light-savings in hourly data analysis).
3. Data acquisition can be a difficult task considering the confidentiality.
4. The future scope of our project involves including taxi, subway, high schools, universities, office timings, restaurant, retail shop datasets as we assume there could be a strong correlation between demand for cycles with these datasets.

REFERENCES

[1]“Divvy Trips: City of Chicago: Data Portal.” Chicago. https://data.cityofchicago.org/Transportation/Divvy-Trips/fg6s-gzvg/data.

[2]JifuZhao. (2018, February 26). Chicago Divvy Bicycle Sharing Data. Retrieved from https://www.kaggle.com/yingwurenjian/chicago-divvy-bicycle-sharing-data

[3]”National Weather Service: Data Portal.” Chicago. https://w2.weather.gov/climate/index.php?wfo=lot

[4]Eren, Ezgi, and Volkan Emre Uz. “A Review on Bike-Sharing: The Factors Affecting Bike-Sharing Demand.” *Sustainable Cities and Society*, vol. 54, 2020, p. 101882., doi:10.1016/j.scs.2019.101882.

[5]J. Zhang, X. Pan, M. Li and P. S. Yu, "Bicycle-Sharing System Analysis and Trip Prediction," 2016 17th IEEE International Conference on Mobile Data Management (MDM), Porto, 2016, pp. 174-179.

[6] Guo Y, Zhou J, Wu Y, Li Z (2017) Identifying the factors affecting bike-sharing usage and degree of satisfaction in Ningbo, China. PLoS ONE 12(9): e0185100. https://doi.org/10.1371/ journal.pone.0185100

[7]Si, Hongyun & Shi, Jian-gang & Guangdong, Wu & Chen, Jindao & Zhao, Xianbo. (2018). Mapping the bike-sharing research published from 2010 to 2018: A scientometric review. Journal of Cleaner Production. 213. 10.1016/j.jclepro.2018.12.157.

[8]Faghih-Imani, A., Hampshire, R.C., Marla, L., & Eluru, N. (2017). An empirical analysis of bike-sharing usage and rebalancing: Evidence from Barcelona and Seville. *Transportation Research Part A-policy and Practice, 97*, 177-191.

[9]Jonathan Corcoran, Tiebei Li, David Rohde, Elin Charles-Edwards, Derlie Mateo-Babiano, Spatio-temporal patterns of a Public Bicycle Sharing Program: the effect of weather and calendar events, Journal of Transport Geography,Volume 41,2014,Pages 292-305,ISSN 0966-6923, https://doi.org/10.1016/j.jtrangeo.2014.09.003.

[10]Bicycle sharing system. (2015, July 18). Retrieved from http://en.wikipedia.org/wiki/Bicycle_sharing_system

[11]Guilherme N. Oliveira, Jose L. Sotomayor, Rafael P. Torchelsen, Cláudio T. Silva, João L.D. Comba, Visual analysis of bike-sharing systems,Computers & Graphics,Volume 60,2016,Pages 119-129,ISSN 0097-8493,
https://doi.org/10.1016/j.cag.2016.08.005.