What Determines the Duration of a Cab Trip in New York?

2nd Capstone Project Report

1. Background, problems and questions to be answered

Cabs, buses, trains, any forms of public transportation are the integral and important parts of life of any city. This is especially true if we talk about large cities, such as New York, Chicago or Los Angeles. People drive or take public transport to go to work places and airports, visit friends and attend events. And everyone, before embarking on a trip, asks one and the same question: Do I depart early enough to be on time? And very often, not one factor is necessary to consider to answer this question. The duration of any trip, be it public or private, depends on the level of traffic congestion, day of the week (workday or weekend), hour of the day, weather conditions determined in part by the time of the year, state of the road, and even the unexpected accidents that slow down traffic. The influence of some factors is obvious and known well in advance allowing to make almost certain conclusions. For example, everyone knows that roads are less cluttered during weekends compared to workdays, meaning that a trip during weekend will be faster with very high probability (trips to special events such as Super Bowl are definitely exceptions). The role of other factors may not be that trivial and the majority of people overlook them. Suppose someone wants to take a cab to airport in August and tries to estimate the duration of the trip based on another business trip that was made in June. The time of the flight is approximately the same and the summer weather is excellent. But does this person take into account the fact that more people tend to go on vacation in August rather than in June, resulting in busier roads to airport and hence slower traffic? It is doubtful that many people go that far in their analysis. The abovementioned example strongly suggests that there are many factors, often hidden and intertwined, and thus difficult to analyze rigorously. That is why it is important to perform comprehensive statistical analysis of all available data that can affect the time it takes for a passenger to travel between two destinations.

In this project we will analyze just one large dataset that contains the continuous variable of duration of the trip, as well as several other variables such as pickup time, drop-off time, geo-coordinates of pick-up and drop-off, number of passengers (total 11 variables). The dataset was prepared by the NYC Taxi and Limousine Commission and can be downloaded from:

https://www.kaggle.com/c/nyc-taxi-trip-duration

By solving regression problem, we would like to answer one major question: Which features influence most the duration of the taxi trips in New York City, and what is their relative importance? Other questions we would like to explore are:

a) How correlated is the trip duration with the distance between pick-up and drop-off, as well as other variables?

b) How are the average speeds of the trips distributed?

c) Where do the fastest and slowest trips originate?

d) When do the fastest and slowest trips originate?

2. Potential clients

There are two groups of clients that could be interested in the findings from this project.

a) Taxi companies are interested in optimizing the trips within the city limits as well as between cities. In other words, they are interested in knowing which trips, originated at different locations and times, are likely to be faster given the same distances between origins and at destinations.

b) Citizens who use cabs frequently are themselves interested in knowing where and when it may be convenient to catch a cab in order to save time.

3. Data preparation and exploratory data analysis

3.1 Data preprocessing steps

a) We first read-off the dataset. Since, we are not participating in the Kaggle competition, we will be using just the training set that contains the target variable, namely the trip duration. One should download the file ‘NYCTripDuration_train.csv' and rename it to 'NYCTripDuration.csv'. We can see that our dataset is quite large containing 1458644 entries. In addition to the trip duration (the target variable), the dataset includes pickup datetime, drop-off datetime, geo-coordinates of pick-up and drop-off, number of passengers, and several other variables (total of 10 feature variables).

b) Albeit we don't expect any missing values in the Kaggle dataset, it is useful to ascertain that each column (out of 11) contains zero number of missing values. We then look at the 'id' and 'vendor_id' columns. The column 'vendor_id' contains just two values 1 and 2, and we will keep it. From the other side, all values in the 'id' column are different, meaning that this column can’t be anyhow useful and should be dropped. 'store_and_fwd_flag' column contains two categorical variables 'Y' and 'N', so we convert them to 1 and 0.

c) We then create the columns giving the year, month, day, hour and minute of pickup time. All data pertain to 2016 year, so we also drop this column. Also, because we have the duration of the trip in seconds, we drop then the 'dropoff_datetime' as well as 'pickup_datetime' columns.

d) Next, we create the special column containing the geographical distance in kilometers between the pick-up and drop-off places. We use the so-called Vincenti formula for the distance that is accessible through the "geopy" module. One should notice that it takes almost 8.5 minutes to do the calculation for 1458644 observations.

Strictly speaking, we need the driving distances instead of geographical ones. Importation of the driving distances, however, requires using the geolocation-python module which uses the Google maps API. Free usage of this application is limited to only 2500 requests per day and hence impractical for large datasets. We will see that engineering the geographical distance column will be already useful.

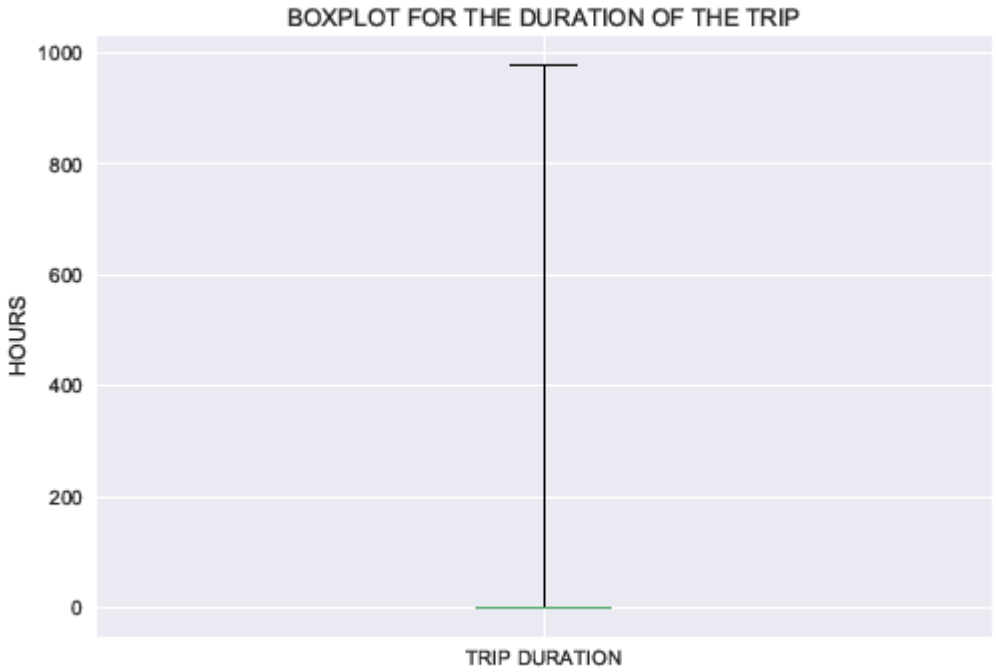
e) Finally, we convert the trip duration to being presented in hours rather than seconds. The corresponding column will be renamed to 'trip_duration(hrs)'. We also create another feature column 'effcetive_speed(kmph)', given by the ratio of the value in 'geographical_dist(km)' to that in 'trip_duration(hrs)'. This is an important column in exploratory data analysis, but we can not use it in predictive modelling because it was itself obtained using the target variable.

The new working dataframe contains now 15 features, one of which 'trip_duration(hrs)' will be our target variable. The feature columns will be the 'vendor_id', 'passenger_count', 'pickup_longitude', 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag', 'pickup_month', 'pickup_day', 'pickup_hour', 'pickup_minute', 'pickup_day_of_week', geographical_dist(km)' and 'effective_speed(kmph)' columns.

3.2 Exploratory Data Analysis

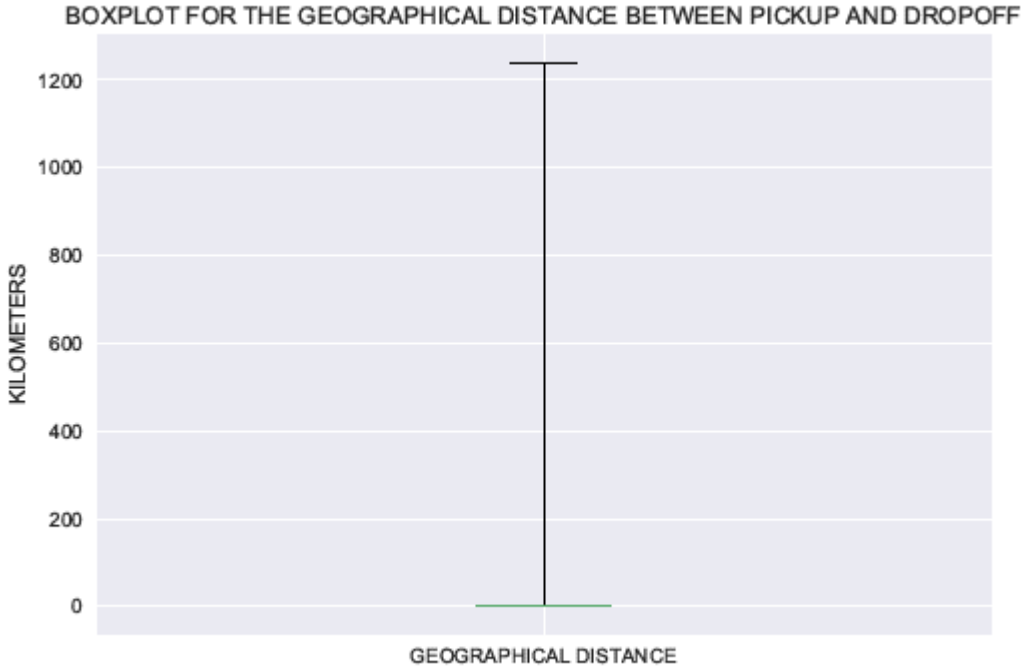
Looking at our columns to start exploratory data analysis (EDA), we assume that 'vendor_id', 'passenger_count' and 'store_and_fwd_flag' columns are rather trivial and of little interest to visualize, so we focus on the data in other columns. We first make the boxplots for 'trip_duration(hrs)', 'geographical_dist(km)' and 'effective_speed(kmph)'.

a) Let's first make the boxplot of and apply the describe methods to the 'trip_duration(hrs)' column. We see that the maximum value in the dataset is as high as 979 hours, while the minimum values is around 1 second.



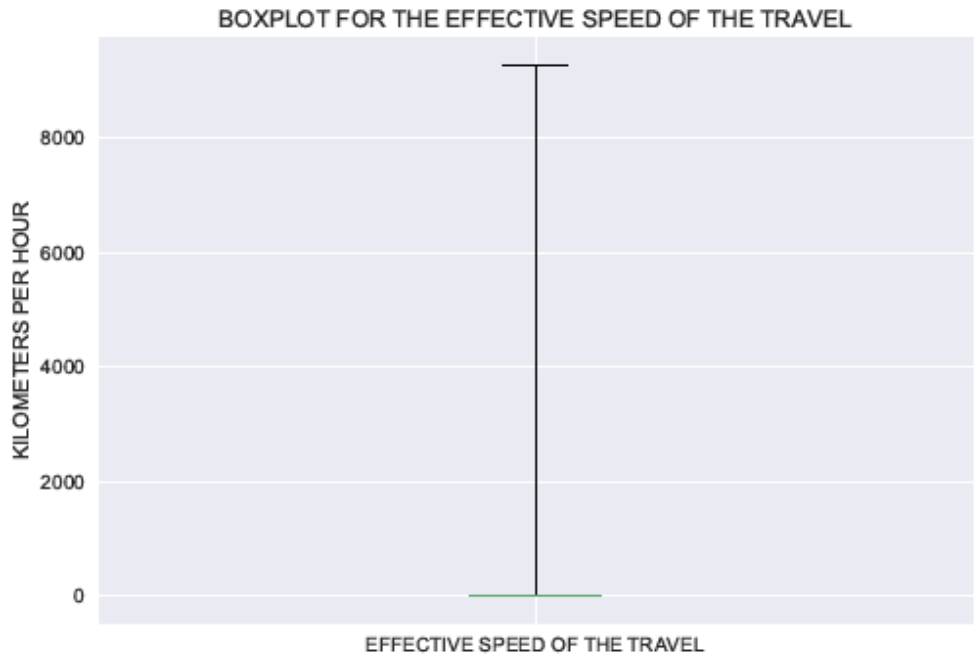
From our point of view, the durations that last more than 16 hours do not reflect a real single trip and could be either the result of data errors, or some combinations of actions or events that simply can’t be interpreted as a normal trip. There are 1968 such trips.

b) Then we make the boxplot of and apply the describe methods to the 'geographical_dist(km)' column. We see that the maximum value in the dataset is as high as 1240 km, while the minimum values is just 0 km.



It is clear that the entries having 'geographical_dist(km)' equal to zero mean the trip did not occur at all. There are 5897 such entries. There are also 13 trips between locations separated 150 km apart. It is difficult to say whether these trips were real, but these data are anomalous from the point of view of cab drivers and passengers.

c) The data for the 'effective_speed(kmph)' column (the artificially engineered effective speed of the travel feature) should be handled in a similar way. The minimum effective speed is 0 kmph, while the maximum is 9297 kmph.



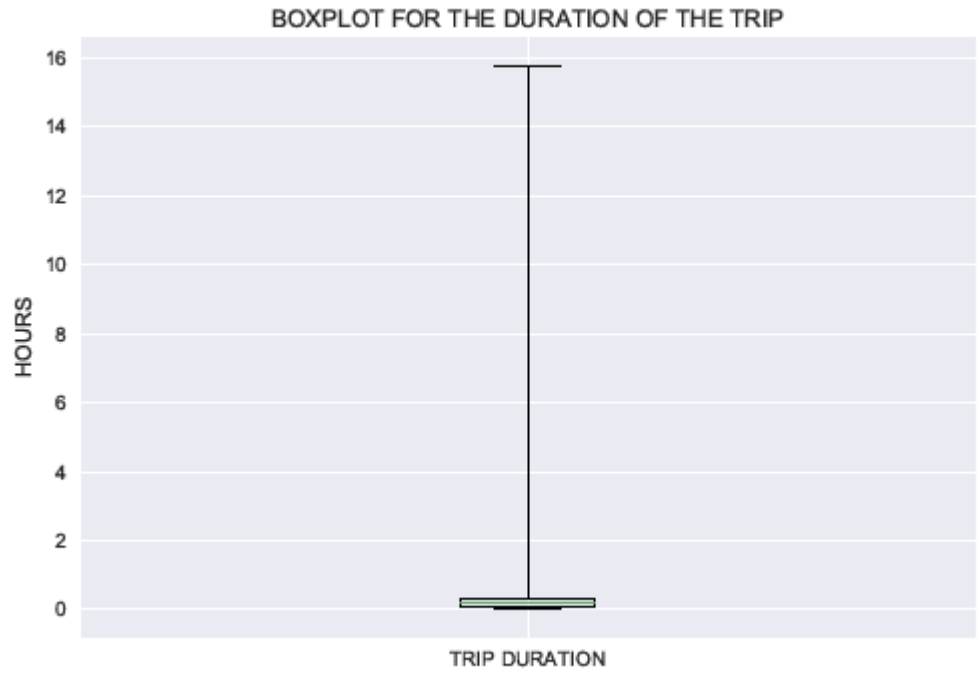
The speed limit in New York is 55mph, or 88.5 kmph. There are 205 trips that resulted in effective speed greater than the speed limit.

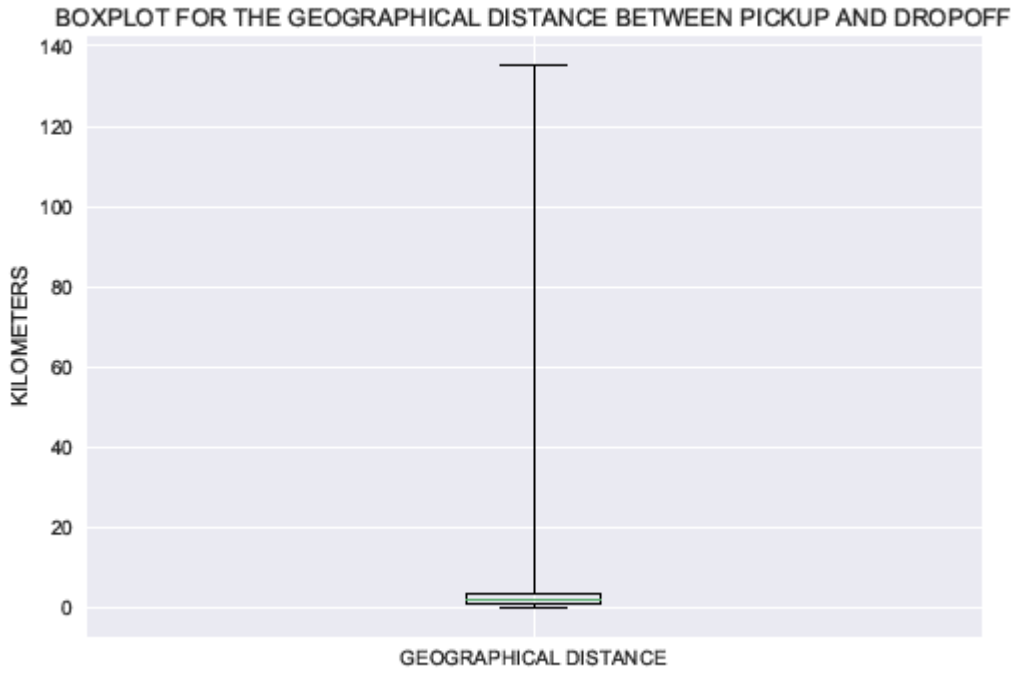
e) We remove from the data: (i) trips that lasted more than 16 hours, (ii) trips with the geographical distances of either 0 km or more than 150 km, (iii) trips where the effective speed was greater than the speed limit.

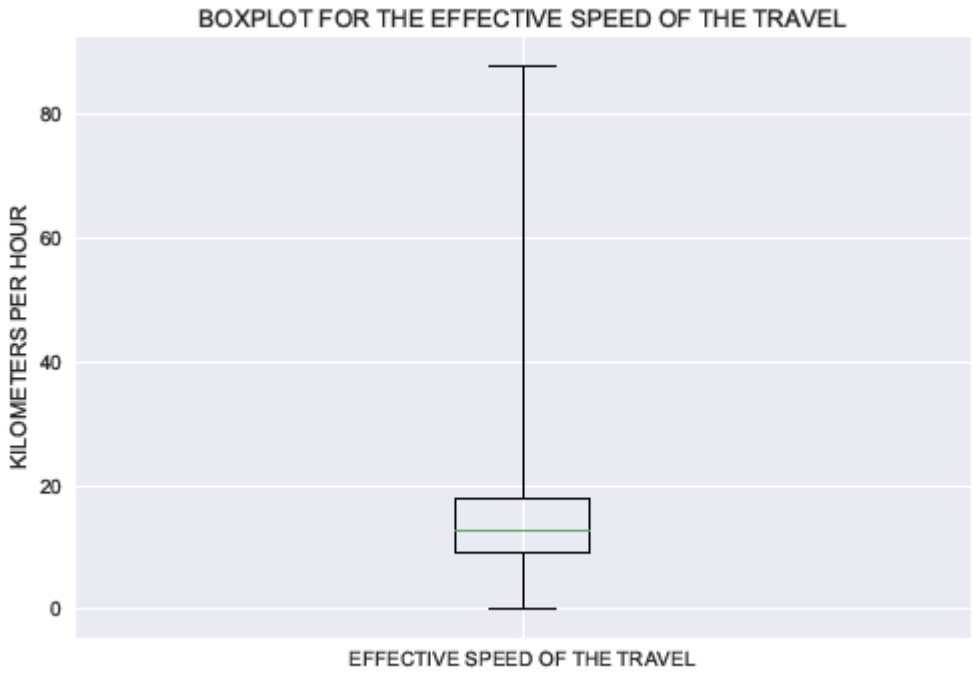
f) We also remove two entries with suspicious longitudes from the data-frame (entries 1177854, and 1062842), since it is unlikely that those trips occurred in New York, and these points may negatively affect predictive data analysis. Our column will then contain 1450573 instances.

g) The mean value of the trip duration is 0.23 hour, while the standard deviation is 0.2 hour. The mean value of the geographical distance between pickup and drop-off is only 3.45 km with the standard deviation being 3.95 km. We see that the trip durations and geographical distances are quite broadly distributed. We see also that the average effective speed for the taxi trips in New York is rather small (14.4 km per hour which is smaller than 20 km per hour); this is quite reasonable for a busy megapolis. Larger speeds are likely to occur for the intercity trips in which the high-speed highways were used.

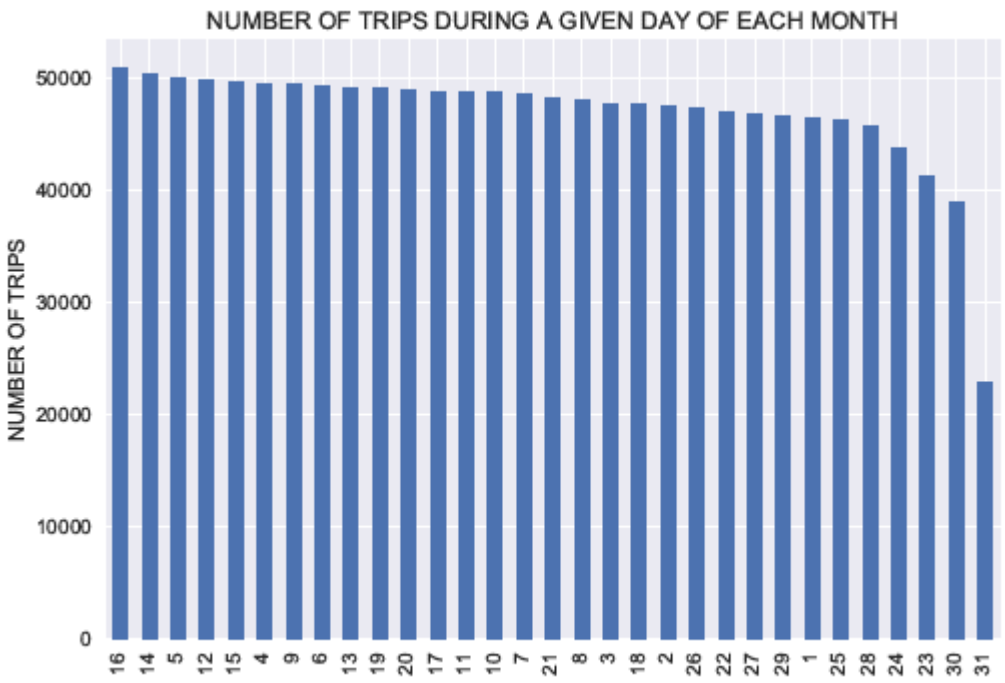
h) We then make the boxplots of the the 'trip_duration(hrs)', 'geographical_dist(km)', 'effective_speed(kmph)' once, what we believe the anomalous data has been removed. The plots reveal that there is still a small number of trips that lasted either very long, or resulted in large travel distances. This means that the data for the trip duration as well as for geographical distances have very long tails in their distributions.

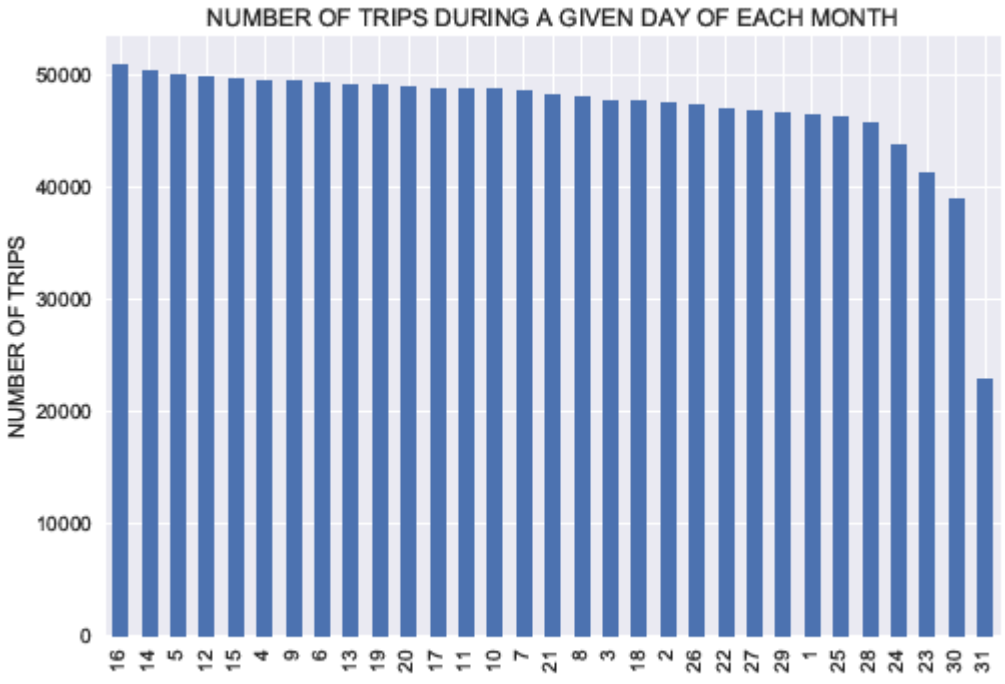

BOXPLOT FOR THE DURATION OF THE TRIP
HOURS
16
14
12
10
8
6
4
2
0
TRIP DURATION

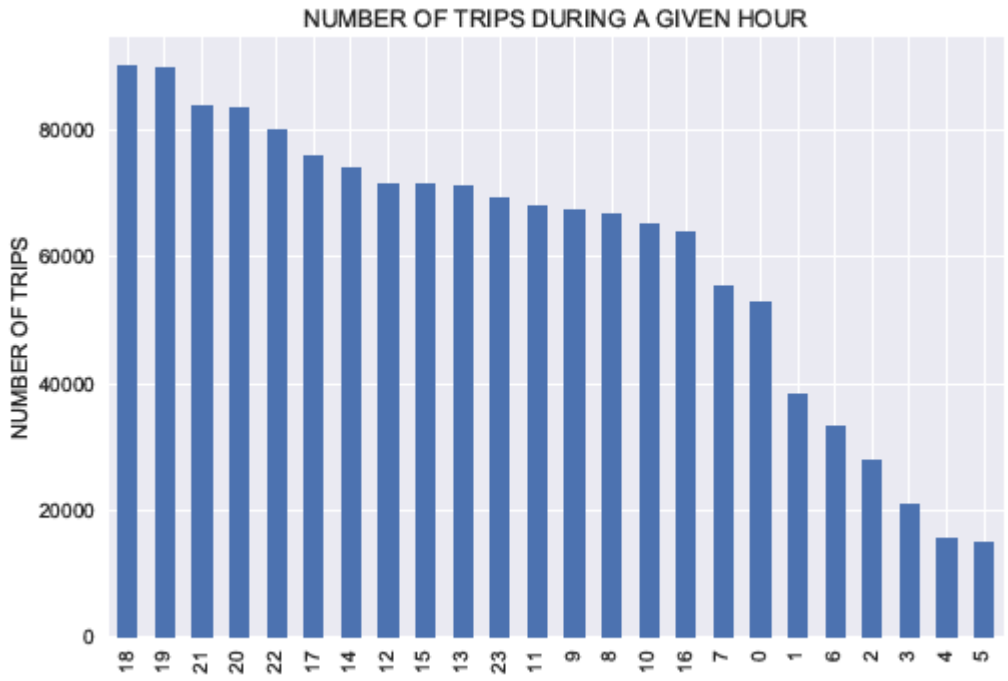

BOXPLOT FOR THE GEOGRAPHICAL DISTANCE BETWEEN PICKUP AND DROPOFF
KILOMETERS
140
120
100
80
60
40
20
0
GEOGRAPHICAL DISTANCE

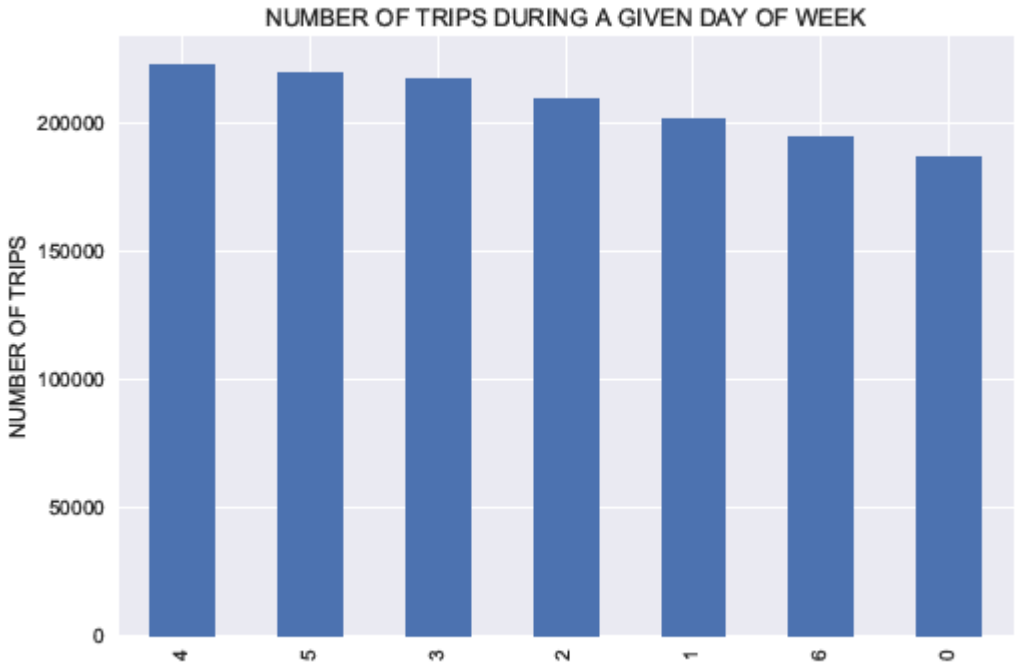


i) Let's make four bar plots for the trip numbers for given values in the 'pickup_month', 'pickup_day', 'pickup_hour' and 'pickup_day_of_week' columns. The first bar plot suggests that the number of trips does not vary significantly with month, while the plot displaying the number of trips per given day of each month suggests that people are slightly more reluctant to take cabs during the third decade of each month. The number of trips made on the 31st day is roughly half the similar number for any other day, which is expected. The third plot, that shows the hourly distribution of the number of trips, reveals that 6,7,8,9 and 10 pm are the busiest hours, while late night and early morning are the least busiest ones. Finally, the last plot suggests that people are least likely to take cabs on Sundays, Mondays and Tuesdays.


NUMBER OF TRIPS DURING A GIVEN DAY OF EACH MONTH
NUMBER OF TRIPS
50000
40000
30000
20000
10000
0
16 14 5 12 15 4 9 6 13 19 20 17 11 10 7 21 8 3 18 2 26 22 27 29 1 25 28 24 23 30 31

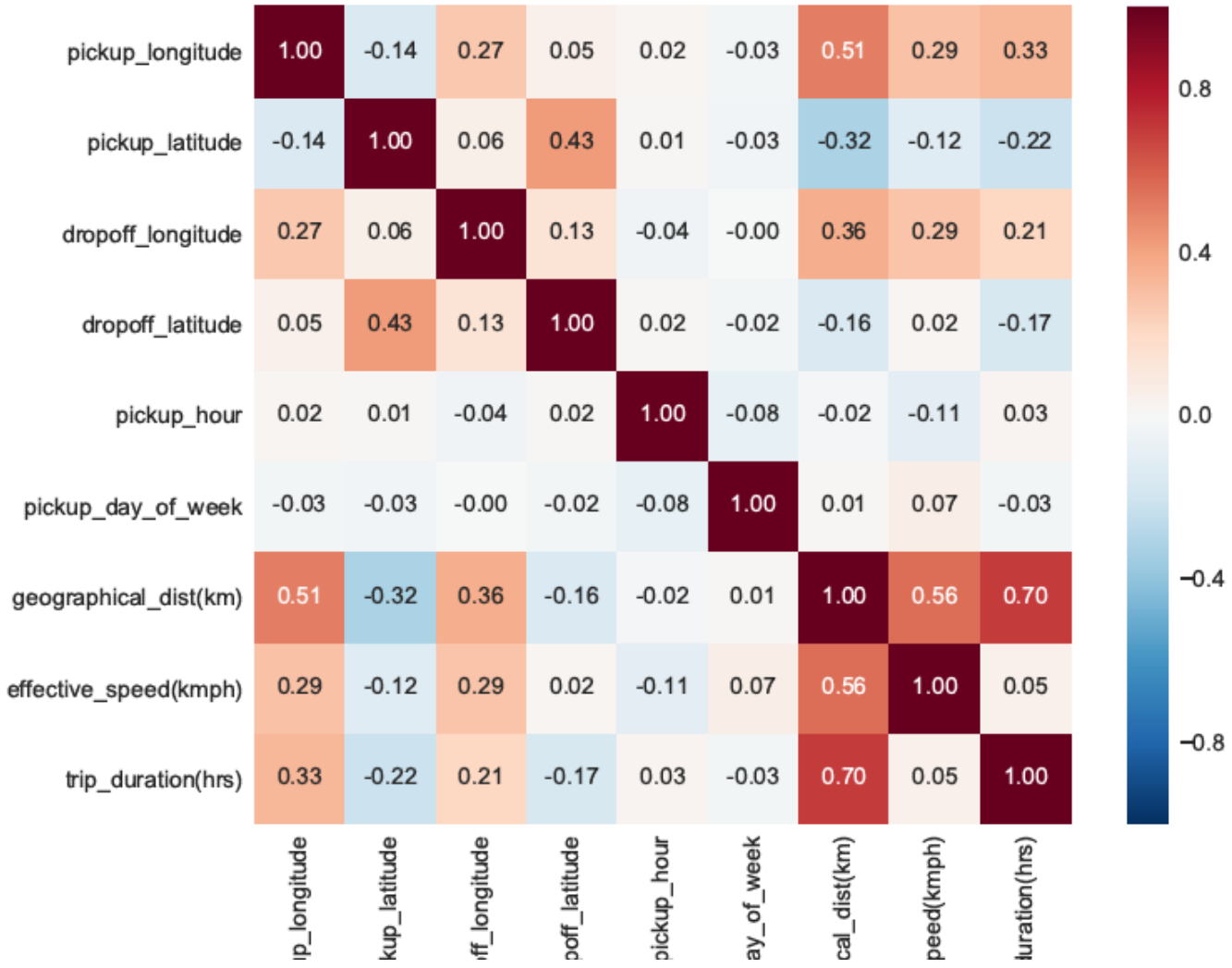

NUMBER OF TRIPS DURING A GIVEN DAY OF EACH MONTH
NUMBER OF TRIPS
50000
40000
30000
20000
10000
0
16 14 5 12 15 4 9 6 13 19 20 17 11 10 7 21 8 3 18 2 26 22 27 29 1 25 28 24 23 30 31





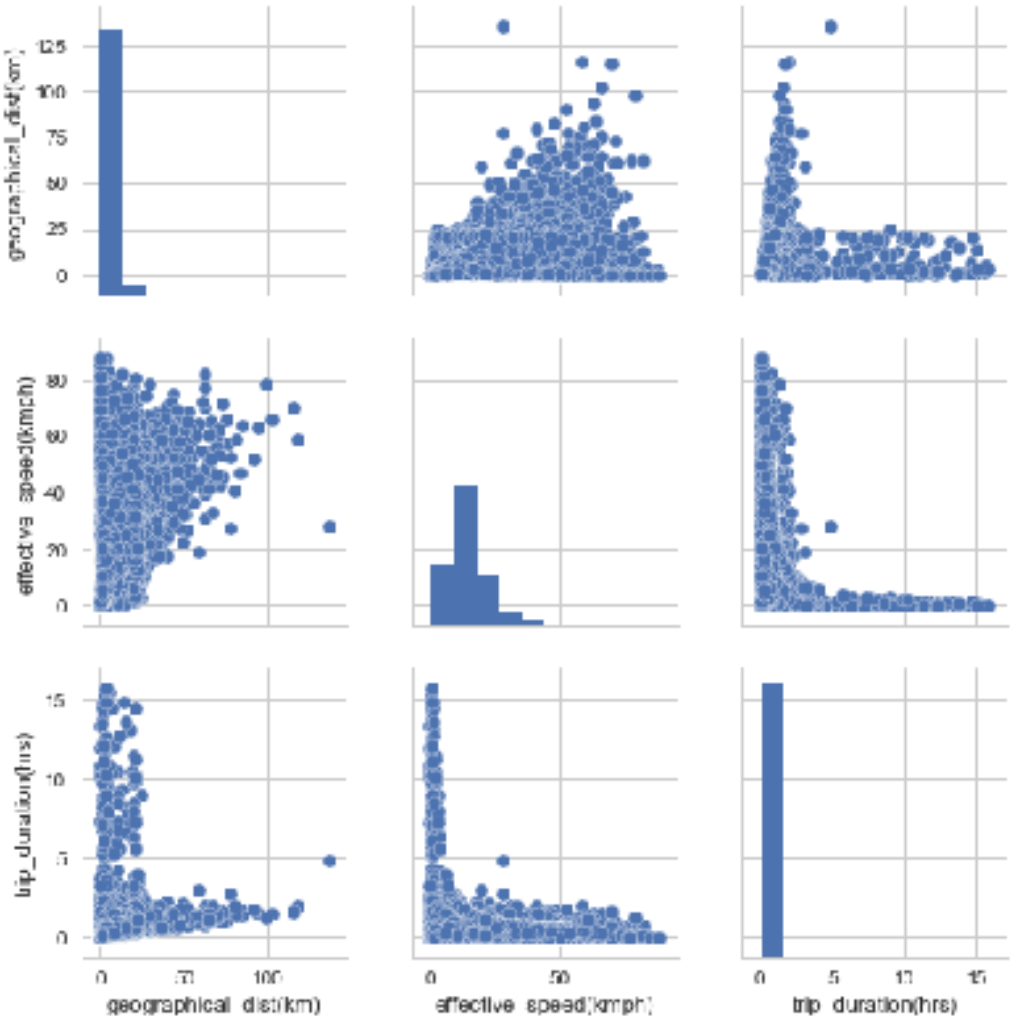
j) Now we plot the heat map of the correlation matrix array choosing the following columns: 'pickup_longitude', 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude', 'pickup_hour', 'pickup_day_of_week', 'geographical_dist(km)', 'effective_speed(kmph)' and the target column 'trip_duration(hrs)'. Values in the 'pickup_month'and 'pickup_day' columns, having almost uniform distributions, are almost not correlated with anything else, and we are not plotting these columns. We see that the drop-off and pickup longitudes are rather strongly correlated with each other; the same can be said about the drop-off and pickup latitudes. There is also a notable

correlation between the values in the 'geographical_dist(km)' and 'effective_speed(kmph)' columns, which is not unexpected. There is also weak but far from zero correlation between the values in 'geographical_dist(km)' column and those in the 'pickup_longitude', 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude' columns respectively; this latter fact can be explained by the presence of the sea coast near New York city. There is a strong correlation between the 'trip_duration(hrs)' values and the values in the 'geographical_dist(km)' column. The 'trip_duration(hrs)' values are only a little bit correlated with the values in the 'pickup_longitude', 'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude' columns.



k) Finally, we make the seaborn pair plot involving the columns 'geographical_dist(km)', 'effective_speed(kmph)', and 'trip_duration(hrs)'. It follows from the plot that geographical distances between the pickup and dropoff locations are narrowly distributed with a small number of large distances. The same is true for the trip durations -- there are few instances with very large time intervals. Large trip durations probably mean substantial waiting periods during the trips as follows from the scatter plots 'trip_duration(hrs)' vs 'geographical_dist(km)' and 'effective_speed(kmph)'. One should pay attention to the broad distributions of effective speeds. It follows from the data that low effective speeds correspond to the presence of notable waiting

periods during which there was no motion of the cab, but taximeter kept counting. One can also conclude that the larger geographical distance the higher the effective speed is. Also, the larger trip durations correspond, on average, to smaller effective speeds, as expected.



We have now the modified data-frame that is ready for predictive modelling using certain methods. We convert it back to the csv file ('NYCTripDuration_modified.csv') to use it in another program.

3. Predictive modelling

In this section we will try to solve the following problem. Given the features: latitude of pickup, longitude of pickup, latitude of drop-off, longitude of drop-off, vendor identification number, store and forward flag variable, number of passengers on a trip, month of pickup, day of pickup, day of the week of pickup, hour of pickup, minute of pickup and geographical distance between the pickup and drop-off locations, we would like to predict the duration of the trip with the best possible accuracy. This is the classical regression problem, and to evaluate the goodness of the fit of the models, we will be calculating the coefficient of determination R^2. We will be also looking at how accurately we are predicting the average value and the standard deviation of the trip duration. Let's remind ourselves that we can’t use the effective speed of the trip as a feature variable, since it was engineered using the trip duration itself.

Our data set is very large and contains 1450573 instances. Compared to the original dataset (1458644 instances), it is a bit shorter since we removed a very small number of instances with the data believed strange and thus reduced the size of the original dataset by 0.5533 percent. The implementation of the standard machine learning algorithms on this dataset is complicated by two things: i) the size of the dataset resulting in procedures being vulnerable to computer memory, and ii) the broad distribution of the target variable 'trip_duration(hrs)' as well as one of the feature variables 'geographical_dist(km)' that is strongly correlated with the target one. Point ii) means that we have many points that can be considered outliers and affect the accuracy of prediction, but nevertheless constitute reasonable data corresponding to instances that most likely actually happened in reality.

We will use five regressors: Elastic Net Regression, Stochastic Gradient Descent Regression, Decision Tree Regression, Random Forest Regression and Multilayer Perceptron Regression. The details will be presented in three separate notebooks: NYTaxiTrips_linear.ipynb (Elastic Net and Stochastic Gradient Descent), NYTaxiTrips_tree.ipynb (Decision Tree and Random Forest) and NYTaxiTrips_neural.ipynb (Multilayer Perceptron). This is done because of the complexity of the problem.

3.1 Elastic Net Regression

We use this type of regression because of its ability to combine Lasso and Ridge regressions. After creating the data-frame from the 'NYCTripDuration_modified.csv' file prepared earlier, we drop the 'Unnamed: 0' and 'effective_speed(kmph)' columns, and shuffle the data-frame arbitrarily. We then write the function that calculates the accuracy of the predicted quantities. This function will be used to compute the accuracies of the predicted mean and standard deviation of the continuous target variable. After that, we split the dataset into arrays of feature and target variables, and create the list of column names. There are 1450573 instances at this point.

After running ElasticNet Regression on the original dataset (with the values alpha = 0.01 and l1_ratio = 0.5), we obtain miserable results. Not only the coefficient of determination is low (0.527), the accuracy of prediction of the standard deviation for the target variable is also low (72.03 percent) (although mean was predicted well enough). This is due to the fact that this method is very susceptible to outliers. Indeed, the target variable 'trip_duration(hrs)', as well as on of the feature variables 'geographical_dist(km)' are very broadly distributed.

We can try to improve the accuracy of predictions by truncating our dataset. Namely, let's remove the instances where 'geographical_dist(km)' is greater than 60 km and those that last more than 6 hours.

Once the dataset is truncated, we run Elastic Net once again using 3-fold cross-validation to determine the optimal values of parameters alpha (possible values are 0.0001, 0.001, 0.01 and 0.1) and l1_ratio (possible values are 0.1, 0.3 and 0.8). We see that the accuracy of the model has increased considerably; the test score is 0.597, while the mean and standard deviation of the target variable are predicted with accuracy 99.93 and 77.31 percent respectively.

Let's examine the weights after running the regression. The most important feature (highest weight in absolute value) is 'geographical_dist(km)', as expected. The second and third most important features are 'dropoff_latitude' and 'dropoff_longitude' respectively.

	COLUMN_NAME	ELNETREG_COEFF
0	geographical_dist(km)	0,803694204
1	pickup_month	0,046158133
2	pickup_latitude	0,043822995
3	pickup_hour	0,039277098
4	pickup_day	0,007015792
5	passenger_count	0,006895233
6	store_and_fwd_flag	0,005765822
7	vendor_id	-0,000805516
8	pickup_minute	-0,00420062
9	pickup_longitude	-0,031822179
10	pickup_day_of_week	-0,038225646
11	dropoff_longitude	-0,052426336
12	dropoff_latitude	-0,065096524

Trying to improve the accuracy, let's drop the least important features: 'pickup_day', 'passenger_count', 'store_and_fwd_flag', 'vendor_id' and 'pickup_minute', and use the PolynomialFeatures() to generate the squared number of the remaining features. We use the parameters alpha = 0.0001 and l1_ratio = 0.1. We see that the accuracy has increased although not as much as we hoped; the test score is 0.649, while the mean and standard deviation of the target variable are predicted with accuracy 99.91 and 80.72 percent respectively. This level of accuracy can hardly be regarded acceptable. It is possible that the accuracy can be improved further by generating cubic interactions between the features. However, this approach would require large computational resources, and we think it is better to try other methods.

3.2 Stochastic Gradient Descent

Following the same preparatory steps, we run the Stochastic Gradient Descent regression with the optimal (not constant) learning rate, and use 3-fold cross-validation to determine the best values of parameters alpha (possible values are 0.0001, 0.001, 0.01, 0.1 and 1.0) and l1_ratio (possible values are 0.1, 0.3 and 0.8). We see that the accuracy of the model is not very high; the test score is 0.596, while the mean and standard deviation of the target variable are predicted with accuracy 99.82 and 75.47 percent respectively.

As in the case of elastic net regression, most important feature is 'geographical_dist(km)', as expected. The second and third most important features are 'dropoff_latitude' and 'dropoff_longitude' respectively.

	COLUMN_NAME	SGDREG_COEFF
0	geographical_dist(km)	0,780250681
1	pickup_month	0,042058403
2	pickup_hour	0,039743043
3	pickup_latitude	0,036967211
4	passenger_count	0,005884071
5	pickup_day	0,005792425
6	store_and_fwd_flag	0,003663024
7	vendor_id	0
8	pickup_minute	-0,005858839
9	pickup_longitude	-0,024413926
10	pickup_day_of_week	-0,038673723
11	dropoff_longitude	-0,050903746
12	dropoff_latitude	-0,061414987

Trying to improve the accuracy, we attempted to drop the least important features: 'pickup_day', 'passenger_count', 'store_and_fwd_flag', 'vendor_id' and 'pickup_minute', and use the PolynomialFeatures() to generate the squared number of the remaining features. However, doing this resulted in the learning procedure picking up unstable solution. We also even tried to go to cubic interactions and run the SGD regressor on smaller dataset due to memory constraint; the result was the same -- convergence to wrong solution resulting in negative coefficient of determination. Thus, we abandon attempts to improve the accuracy further using SGD.

The results of Elastic Net and Stochastic Gradient Descent methods are summarized in the table below.

	Characteristics	Elastic Net	Stochastic Grad. Descent
0	Training score	0,651995467	0,598166445
1	Test score	0,649106913	0,596248681
2	Actual mean of the test set	0,232793846	0,232793846
3	Predicted mean of the test set	0,232984344	0,232380235
4	Accuracy of prediction of the mean	0,999181685	0,998223273
5	Actual std of the test set	0,184052192	0,184052192
6	Predicted std of the test set	0,148579306	0,138914835
7	Accuracy of prediction of the std	0,807267245	0,754757837

The conclusion one can make looking at the results is that it is better to use the nonlinear methods in an attempt to improve the accuracy of predictions. So, let's consider the Decision Tree and Random Forest regressions.

3.3 Decision Tree Regression

Following the same steps, we use Decision Tree Regression to predict the trip duration given the dataset prepared earlier. From the very beginning we will be using the truncated dataset in which the geographical distance between pickup and drop-off locations is smaller than 60 km, and the trip durations are limited to 6 hours. We use the Decision Tree Regressor with default values of parameters. We see that the training score is 0.999, while the test score is 0.547. This confirms that Decision Tree regressor is prone to overfitting. The accuracy of predictions for mean and standard deviation is very high, however, 98.56 and 97.49 percent respectively.

Examining the relative importance of features (their sum is equal to one in this method), we see that the most important feature is 'geographical_dist(km)', followed by 'pickup_hour' and 'dropoff_latitude'. One should pay attention to the importance of 'pickup_hour' feature; this feature had relatively low importance in the linear methods.

	COLUMN_NAME	DCTREG_FEATURES
0	geographical_dist(km)	0,651356045
1	pickup_hour	0,0733079
2	dropoff_latitude	0,051920579
3	pickup_longitude	0,046203952
4	dropoff_longitude	0,045169856
5	pickup_latitude	0,035910931
6	pickup_day_of_week	0,028917022
7	pickup_minute	0,022424885
8	pickup_day	0,021193526
9	pickup_month	0,014274311
10	passenger_count	0,005799948
11	vendor_id	0,002874098
12	store_and_fwd_flag	0,000646946

3.4 Random Forest Regression

Then we use Decision Tree Regression to predict the trip duration. We again, from the very beginning we will be using the truncated dataset in which the geographical distance between pickup and drop-off locations is smaller than 60 km, and the trip durations are limited to 6 hours.

The Random Forest Regressor is run with default values of parameters. We see that the training score is 0.958, while the test score is 0.764, which is much better than the result obtained using Decision Tree regressor. The accuracy of predictions for mean and standard deviation are also acceptable, 98.51 and 90.47 percent respectively. We did not use the scaling of features for both Decision Tree and Random Forest methods, because the accuracy is almost not affected by such a scaling. The relative importance of features is the same as in the Decision Tree method, although numerical values for feature importance are slightly different.

	COLUMN_NAME	RFREG_FEATURES
0	geographical_dist(km)	0,652704503
1	pickup_hour	0,073318869
2	dropoff_latitude	0,05169099
3	dropoff_longitude	0,046449902
4	pickup_longitude	0,04545165
5	pickup_latitude	0,035373783
6	pickup_day_of_week	0,029227161
7	pickup_minute	0,02164567
8	pickup_day	0,020096605
9	pickup_month	0,014454147
10	passenger_count	0,006173695
11	vendor_id	0,002814296
12	store_and_fwd_flag	0,000598728

The results of Decision Tree and Random Forest methods are summarized in the table below.

	Characteristics	Decision Tree	Random Forest
0	Training score	0,999999917	0,958195279
1	Test score	0,547308295	0,764729328
2	Actual mean of the test set	0,232793846	0,232793846
3	Predicted mean of the test set	0,236139811	0,236244041
4	Accuracy of prediction of the mean	0,985626919	0,985179182
5	Actual std of the test set	0,184052192	0,184052192
6	Predicted std of the test set	0,188671551	0,166518983
7	Accuracy of prediction of the std	0,974901909	0,904737841

3.5 Multilayer Perceptron Regression

Finally, we use the Multilayer Perceptron (MLP) Regression to predict the trip duration given the dataset prepared earlier. From the very beginning we will be using the truncated dataset in which the geographical distance between pickup and drop-off locations is smaller than 60 km, and the trip durations are limited to 6 hours.

Let us start with the regressor in which there is one hidden layer with 100 elements (default value) and each element has the rectified linear unit ('relu') activation function. The regularization parameter alpha is set to 0.001 which is also the default value. We use the default 'adam' solver, a kind of stochastic gradient descent-based optimizer. The results show that despite the relatively large running time, MLP regressor gives the coefficient of determination equal to 0.769, while the mean and standard deviations are predicted with the accuracies of 99.46 and 85.08 percent respectively.

We already see that the MLP regressor shows reasonably good results. However, can we improve the results further, for example, by adding one more hidden layer and changing the activation function for the elements in layers? Let's add one more layer with 100 elements, and employ the 'tanh' activation function. The motivation for the latter step is that the 'tanh' activation function, contrary to the 'relu' one, does not nullify the negative inputs to layers that are possible given the substantial number of negative weights seen in the Elastic Net regression study. As a result of changing the activation function and adding the second layer, we are able to reach the coefficient of determination equal to 0.798 and the accuracy of predicting the mean and standard deviation equal to 99.80 and 87.94 percent respectively. We see also that the amount of time necessary to run the computations nearly quadrupled.

The final results of using Multilayer Perceptron Regression are summarized below. It is possible that they can be further improved by increasing the sizes of layers, as well as the number of elements in them. It is possible that one needs to simply find the appropriate relation between these two parameters as a result of fine tuning. This will require, however, more powerful computational resources, and the necessity to parallelize the process of learning which is beyond the scope of this project.

	Characteristics	Multilayer Perceptron
0	Training score	0,805849008
1	Test score	0,798172035
2	Actual mean of the test set	0,232793846
3	Predicted mean of the test set	0,23546014
4	Accuracy of prediction of the mean	0,988546544
5	Actual std of the test set	0,184052192
6	Predicted std of the test set	0,164005113
7	Accuracy of prediction of the std	0,891079378

We now present the summary table containing the results obtained by all methods.

Characteristics	Elastic Net	SGD	Dec. Tree	Rand. Forest	MLP
Training score	0,652	0,5982	0,9999	0,9582	0,8058
Test score	0,6491	0,5962	0,5473	0,7647	0,7982
Actual mean of the test set	0,2328	0,2328	0,2328	0,2328	0,2328
Predicted mean of the test set	0,233	0,2324	0,2361	0,2362	0,2355
Accuracy of prediction of the mean	0,999	0,9982	0,9856	0,9852	0,9885
Actual std of the test set	0,184	0,184	0,184	0,184	0,184
Predicted std of the test set	0,1486	0,1389	0,1887	0,1665	0,164
Accuracy of prediction of the std	0,8073	0,7547	0,9749	0,9047	0,8911

Looking at the summary table presented above, the reader can see that the following quantities used to evaluate the quality of each regressor were used: training score, test score, actual mean value of the test set, predicted mean value of the test set, accuracy of prediction of the mean value, actual standard deviation of the test set, predicted standard deviation of the test set and the accuracy of prediction of the standard deviation. The accuracy of prediction of the mean value is defined as one minus the absolute value of the difference between the actual and predicted means divided by the actual mean itself; the same is used for the accuracy of prediction of the standard deviation. One can see that the actual means and standard deviations of the test set are the same in all columns. This is because we used the same train-test splitting while running each regression method.

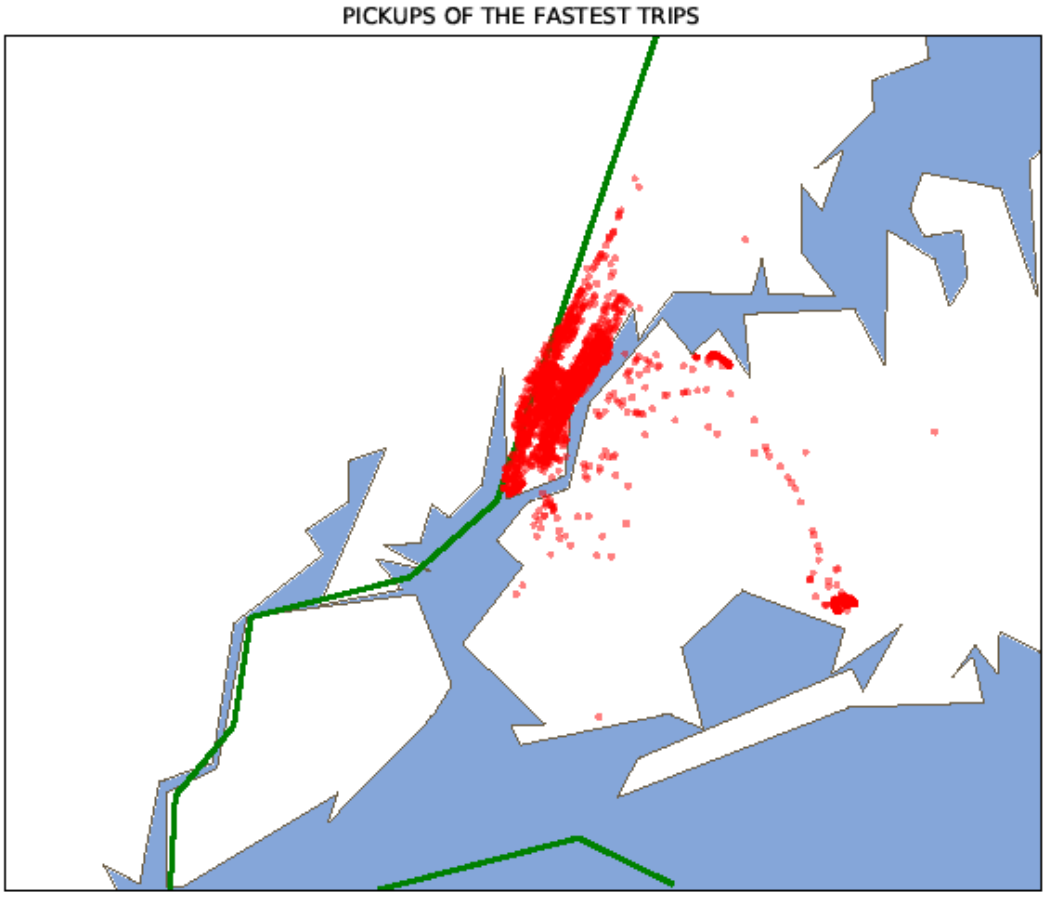
The data in the tables indicate that we managed to achieve the acceptable values for the coefficient of determination only for Random Forest and Multilayer Perceptron regressors. Running Random Forest on our, albeit slightly truncated but still large, dataset does not take prohibitively large amount of time. At the same time, employing the Multilayer Perceptron (MLP) regressor is much more time consuming. Tuning numerous parameters in this method requires a lot of patience. We see, however, that MLP regressor leads to the best results given the available computer memory and power. The small difference between training and test scores is also a good sign. The higher quality of prediction using MLP compared to other methods testifies of the remarkable power of deep networks.

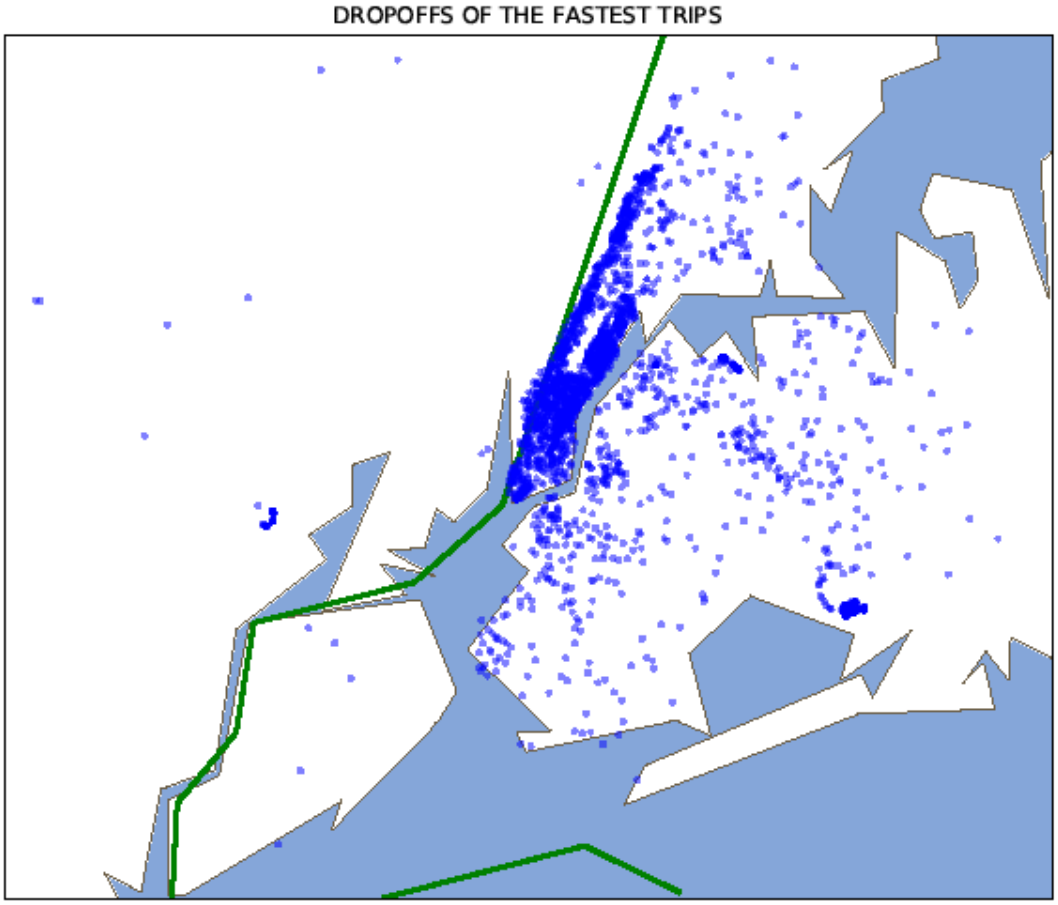
4. Future Research and Recommendations

1) A natural question to ask is how one can improve the accuracy and reach higher values of the coefficient of determination. This can constitute the subject of future research. One way to proceed is to generate larger number of features and try the regressors we used above. However, for the dataset that large this would require higher amounts of memory leading to the necessity to parallelize the learning process. To do this, one can use the methods provided in PySpark machine learning library run externally. The other way to try to increase the accuracy is to use XGBoost regressor, which is currently a very popular method. Finally, one can think of how and which deep learning methods can be employed to achieve highest possible accuracy in this problem.

It is difficult to see that many recommendations can be made to potential clients solely from running various regression models and calculating the coefficient of determination (apart from rather obvious fact that geographical distance is the most important feature influencing the trip duration). That is why below we will do some additional exploratory data analysis trying to give the specific recommendations to potential clients based on the dataset studied.

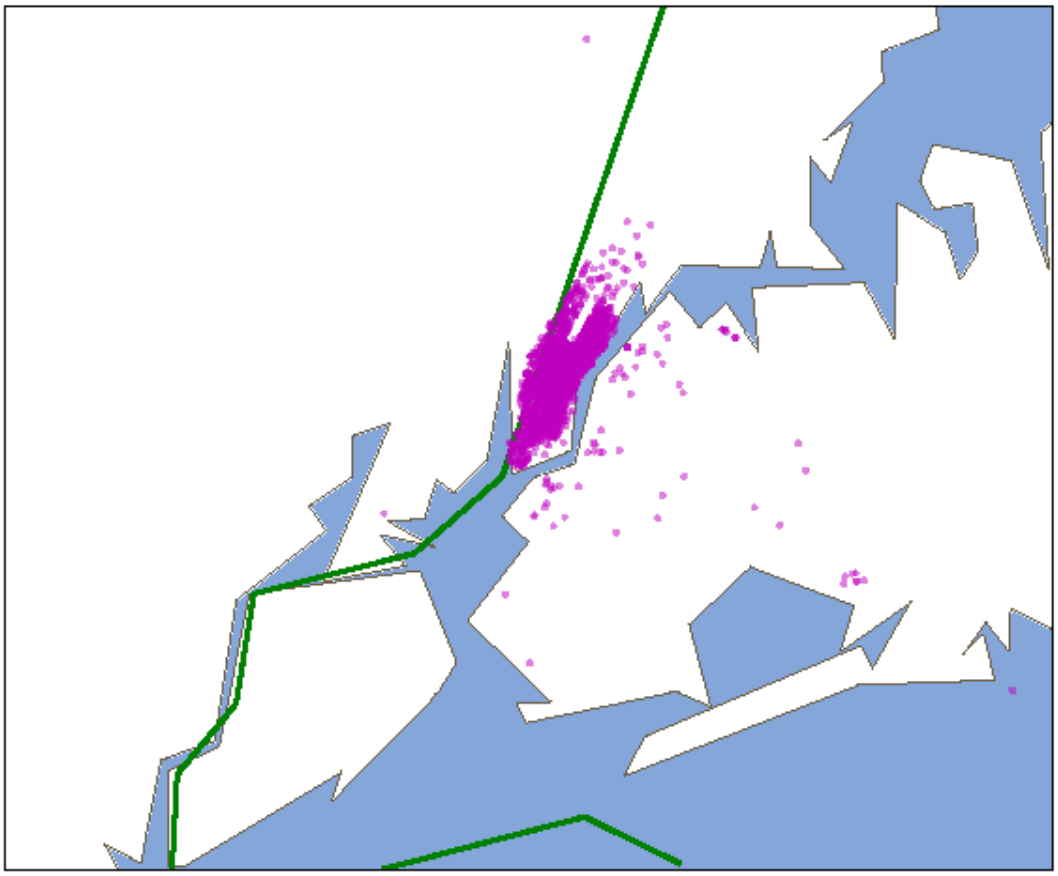
2a) Let's try to answer the question: Where do the fastest trips originate? To do this we use the Basemap module from mpl_toolkits.basemap. We plot the map of New-York and, on top of it, the pickup latitudes and longitudes of the arbitrary chosen 3000 trips whose effective speed is as least twice the average effective speed. It is very interesting that the fastest trips originate mainly in Manhattan, but their final destinations have much broader geography.


PICKUPS OF THE FASTEST TRIPS



2b) The same investigation can be done for the slowest trips, namely those whose effective speed is smaller than half of the average effective speed. We also limit ourselves to plotting only 3000 random samples. The results show that geographically the pickup and drop-off places are distributed almost identically. This confirms the conjecture that, in contrast to the fastest trips, almost all of the slowest trips did not involve travelling large distances. The overwhelming majority of the slowest trips also start or end in Manhattan.

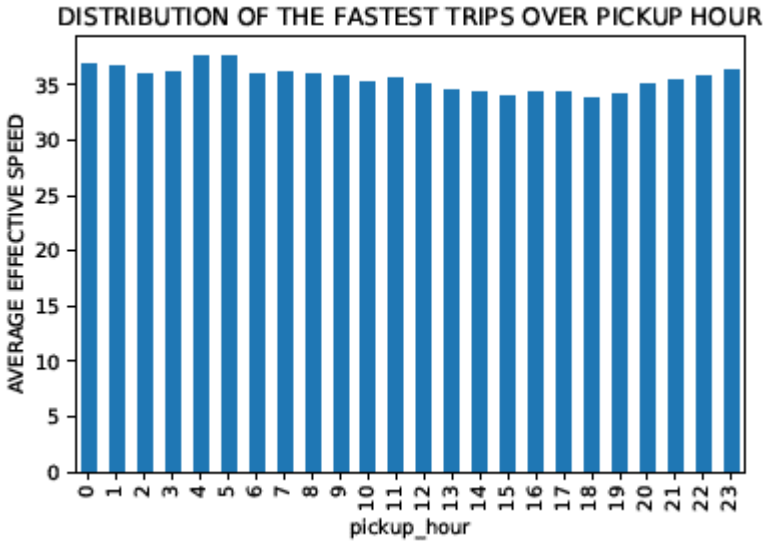
PICKUPS OF THE SLOWEST TRIPS

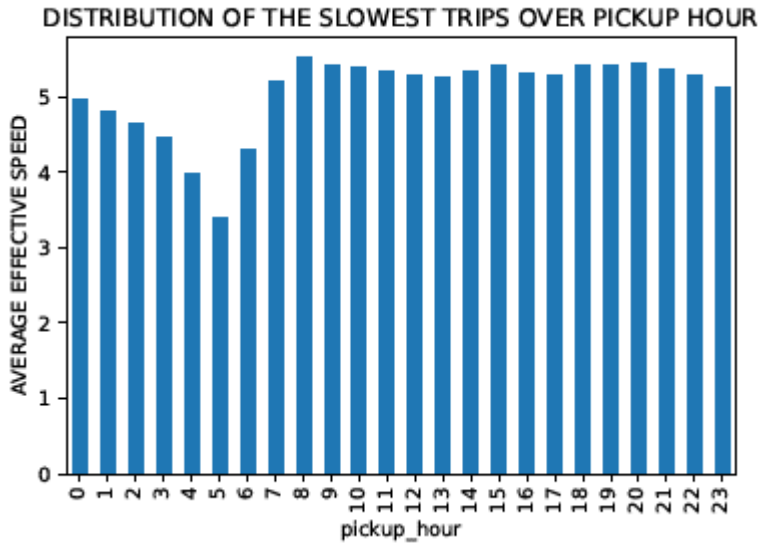


One can conclude that the majority of people who take cabs travel from/to Manhattan, and that the effective speed of the trip is determined by how busy the traffic in Manhattan is. Given this, one can recommend the estimation of the time of the trip by simply looking at the traffic congestion near Manhattan.

3) To conclude, we answer the question of how mean values of the fastest, as well the slowest trips, distributed over hours of pickup. To do this, we present the corresponding bar plots. Let us remind ourselves that the trips belong to the label of the fastest ones, if their effective speed is at least twice the average effective speed. Similarly, a trip is considered the slowest if its effective speed is smaller than the half of the average one. We see that the fastest trips occur, on average, either during the late night or early morning hours, which is not surprising. From the other side, the slowest (on average) trips occurred between 0 and 6 am, which is a bit surprising. This can be explained by the fact that during those hours passengers tend to take only the short distance

trips, and the drivers tend to drive slower to be extra safe during the night and early morning hours.





As a recommendation, if one wants to take a cab and expect the fastest trip, it is better to take a cab during the late-night hours.

5. Acknowledgements

I would like to thank the teams working on building and improving Scikit-Learn and Pandas, opensource communities, as well as the Springboard team.

Special thanks to my mentor Hassan Waqar who provided me with many invaluable advices and guidance.