NYC TAXI DATA

ANALYSIS AND PREDICTION

by Alexey Vlaskin

April 2013, Taxi trips and fare data analysis

The goal of this analysis is to use open data from New York Taxi and Limousine commission (http://www.nyc.gov/html/tlc/html/home/home.shtml) to make a number of suggestions for Taxi business owners and drivers and answer certain questions.

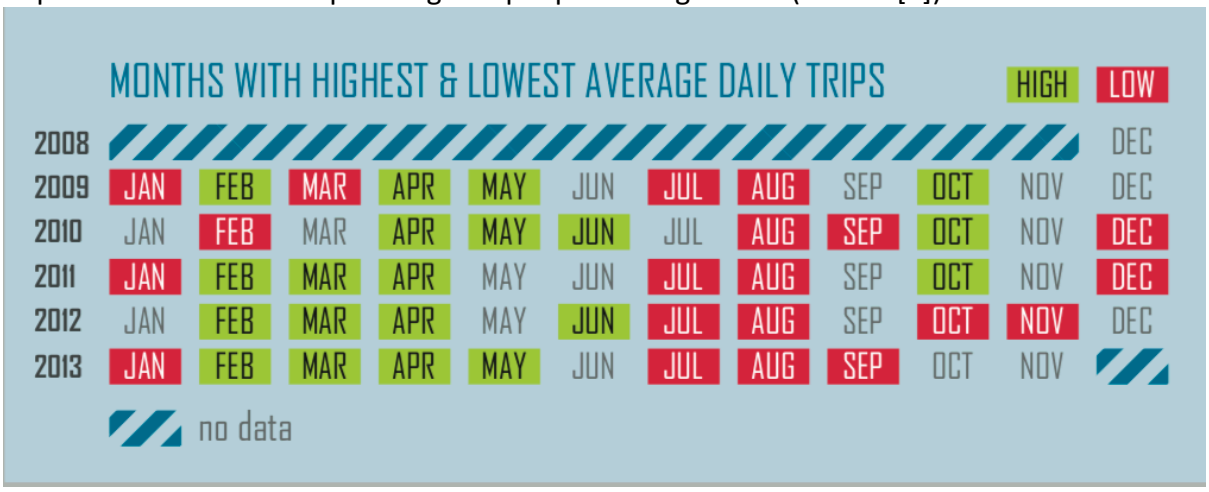
Basic research

Every research should not be conducted in isolation as there are many other people who might worked in this area before us. As a result of discovery we found a certain interesting facts about New York taxi which will be good to know for the Taxi business owners.

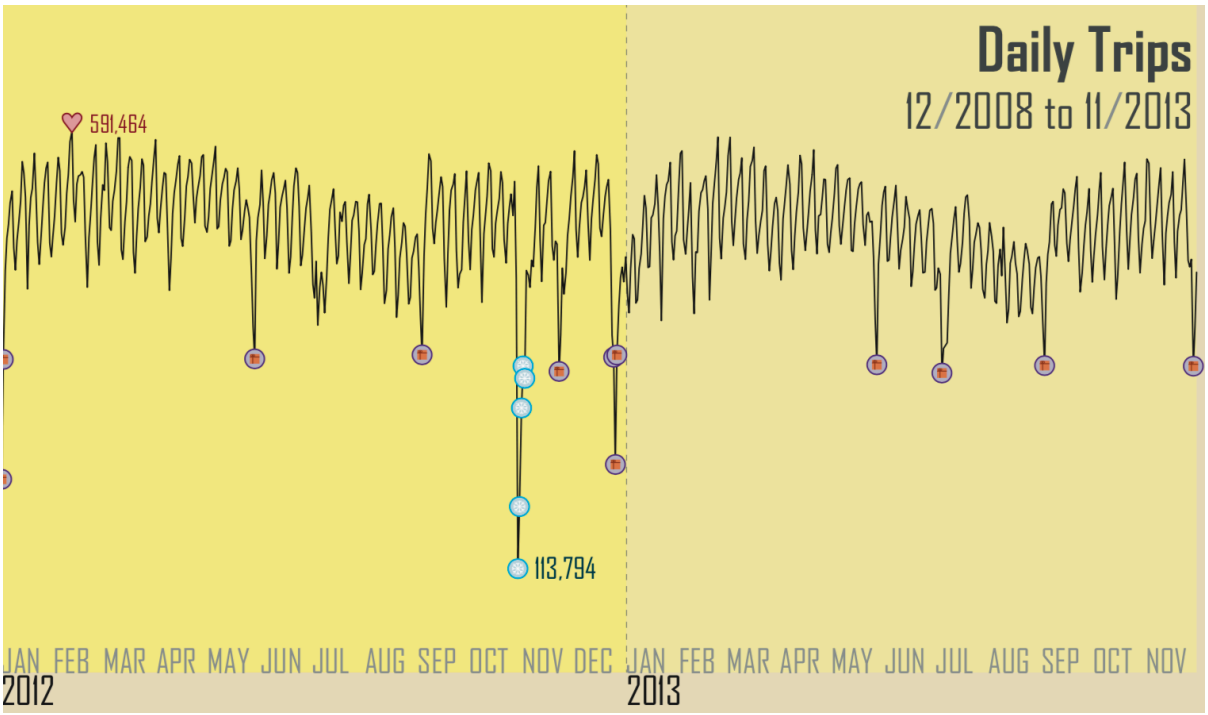
Found info

There are two types of taxis: Green and Yellow and Yellow Taxi now days works on prearranged basis and only Green Taxis are allowed to reply to street hail, however in 2013 there were no Green taxis.

Which months are high in load, which are low over the 2008-2013 years – can be an important data for shifts planning and people management. (Source [2]):

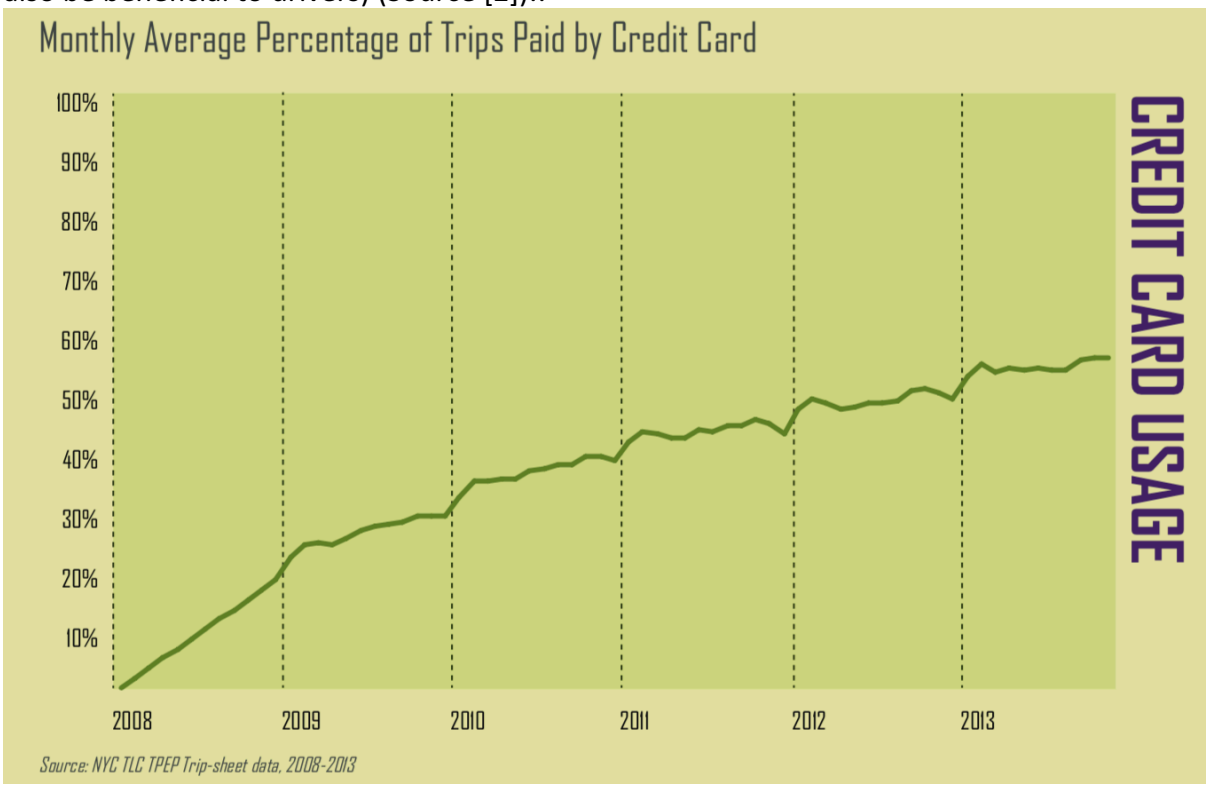


Also interesting to see how weather and public holidays and Valentin's day affect number of trips by taxi(Source [2]):

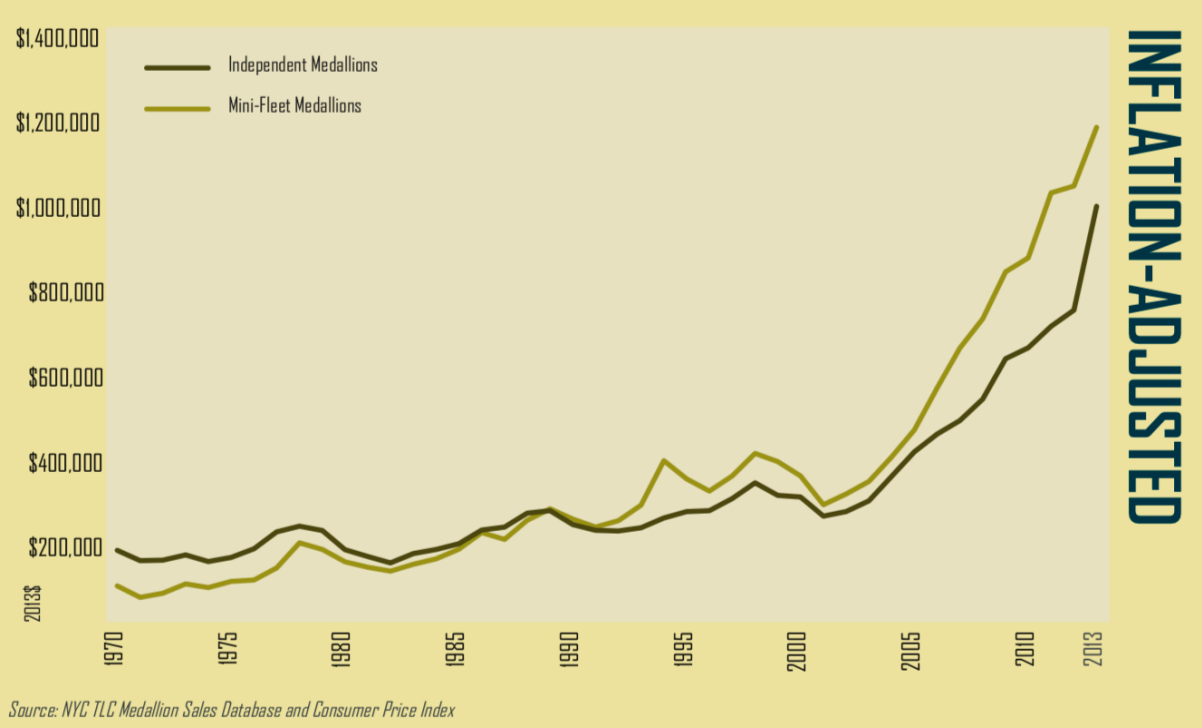


(red squares represent Public holydays, white/bleu – heavy snow, heart – Valentin's Day).

Credit card payments are increasing popularity over the years and ability to receive credit card payment in the car is a must have for new taxi owners(we will see later on how it might also be beneficial to drivers) (Source [2])::



One important bit for taxi owners is a license price(it sometimes can be considered as an investment based on value grow over the years, although Uber disrupted that business model and recently medallion prices dropped):



And a couple words about pricing model that is used by industry. Here is the quote from [3]:

> *"Fare Pricing Model. Every taxi company has their own pricing function they use to charge for fares, and this needs to be given as input. Let r_{fare} be the cost of one fare. Most taxi companies use a function of the following form:*

$$r_{\mathrm{FARE}}(C_{\mathrm{I}}, d, C_{\mathrm{D}}, p, C_{\mathrm{T}}, M) = C_{\mathrm{I}} + dC_{\mathrm{D}} + pC_{\mathrm{T}} + M$$

> *where C_I is an initial cost, d is the distance travelled during the fare, C_D is a cost per kilometer, p is the time parked at traffic lights, C_T is the cost per minute of waiting at lights, and M is miscellaneous fees."*

This information can be used for fare prediction problem and we will talk about it later.

Basic Data Overview

We will start by exploring the data that was given to us and try to find some insights for our later modelling. For more technical details – please refer file ***01_basic_data_overview***.

Typical data science workflow includes following steps:

1. Importing data
2. Overview looking for insights
3. Cleaning data.
4. Feature engineering.
5. Modelling.
6. Models quality assessment.
7. Reporting recommendations.

Note about our data bases

We have two data bases:

1. Trip data describes location, distance, trip time.
2. Fare data describes fare, tip, tax, payment type and payment related info.

Let's start with overview and it will give us enough information for data cleaning.

```
Trips all together: 15100468 Fare data: 15100468
```

Looks like we have two databases with 15 Million trips that happened in April 2013. We will merge these data bases based on medallion id, drivers id and pick up time. (Corresponding code is in cleanData.py).

It would be really good to first check if those records describe same set of trips. From the testing (notebook 02_Data_Merge) it turned out that they indeed describe the same data. We managed to merge them into one data base. This is important because it allows us to link back fares and trip data. For example, we now can predict fare amount based on location of pick up/drop off coordinates. It also allows us to explore people, how they work, what they earn, where the best drivers (in terms of collected fares) are driving, for example.

Let's check how many taxis and drivers were on a road:

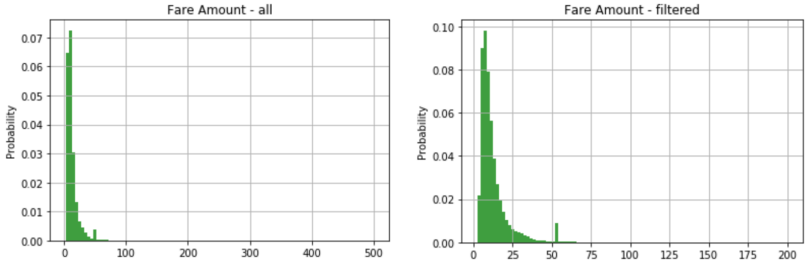
```
Taxis: 13464 Drivers 33111 Ratio drivers/cars = 2.46
```

Ratio between number of cars and number of drivers can give us a sense that people drive taxis in two-three shifts per day. That might be because license owners would like to utilize their cabs and licenses at maximum.

Let's look on the amount of money collected by the market:

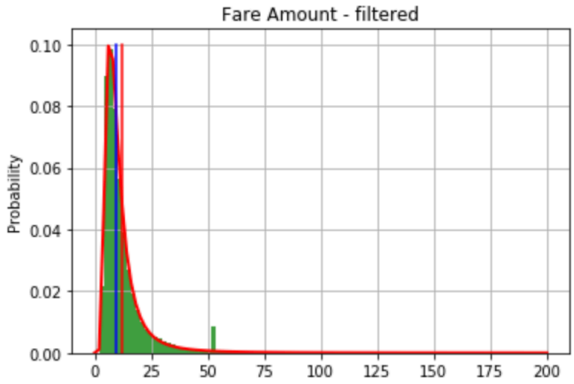
- $221 Million collected in total
- $185 Million collected by business owners in fares – 83%
- $20 Million collected in Tips – 9%

Let's look at fare and filter long tail of it's distribution. Most of the data lays in between 2 and 200 dollars, there is also interesting spike at 50 dollars (it can be certain destination default charges, like for airports):

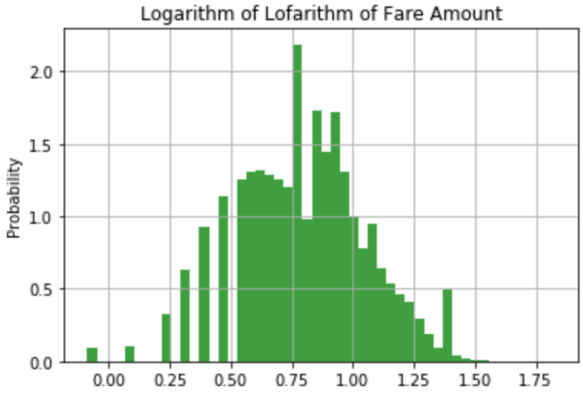


This does not look like a normal or Gaussian symmetrical distribution and looks more like log normal distribution as it is skewed towards 0. According to our numerical research (file fitDistribution.py) Alpha distribution fits it the best. Let's plot it, with a = 2.54.

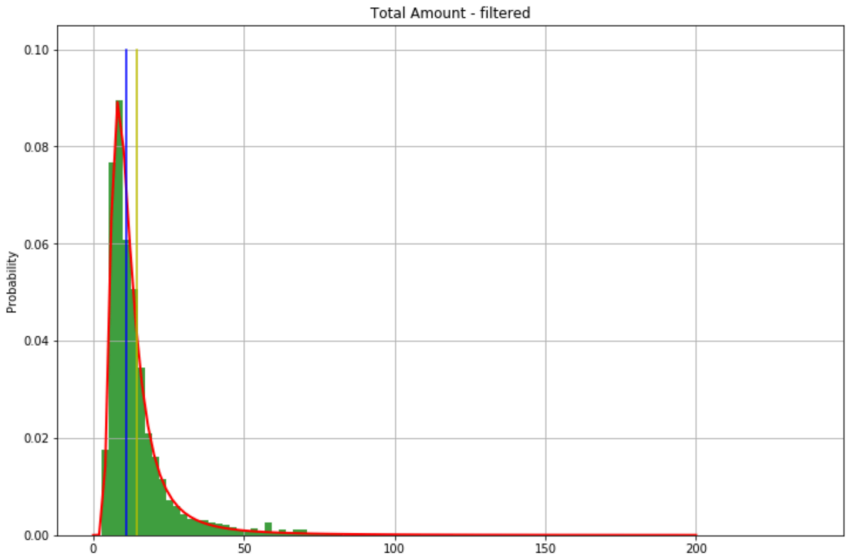
For asymmetrical distributions like log normal or alpha distribution **median gives a better central tendency than mean**. On this picture red vertical line is mean, blue – is a median.



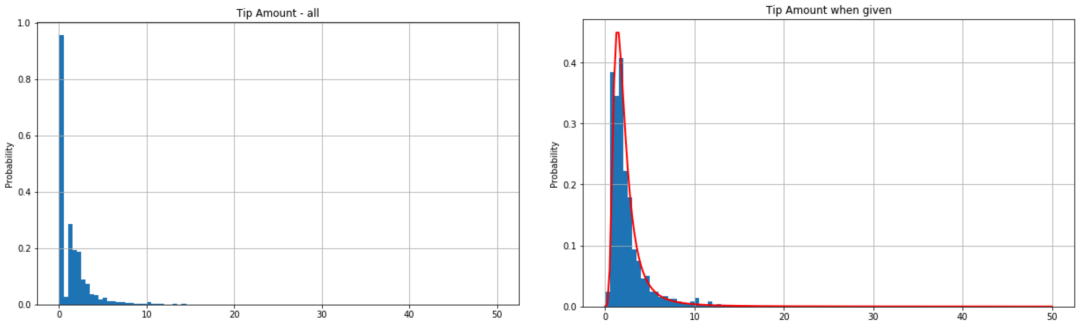
De-skewing the variable can be done by using logarithm function. This will make the errors of prediction follow normal distribution and will allow us to us R2 or variance explained scores to optimize models. This is a plot of logarithm of logarithm of Fare amount:



Similar situation is with total amount charged by taxi. It's distribution is well approximated by alpha distribution with alpha parameter 2.53, mean $14.66 and median of $11. Blue vertical line represents median, yellow – mean.

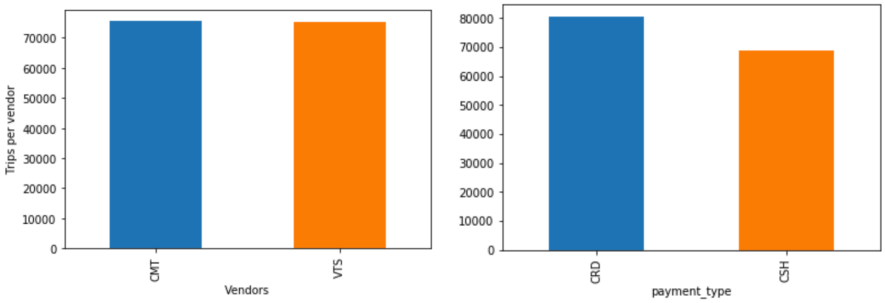


Tips are given in 52% of the trips. Probability distribution for Tip amount is looking different when we plot it with and without zero amount due to high number of trips with no tips given:

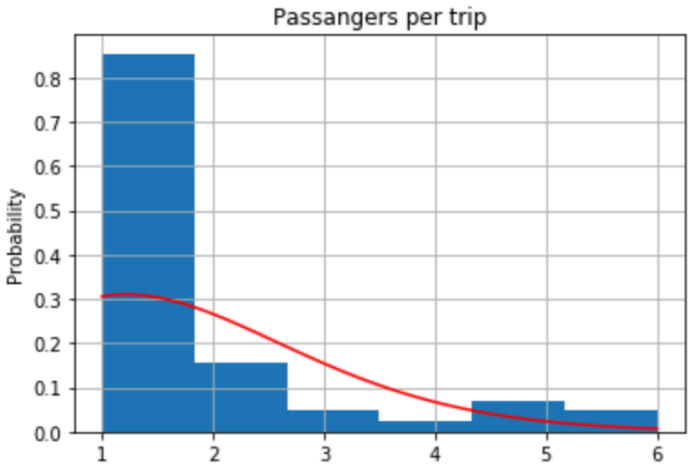


Alpha distribution fits it very well as well. Tip distribution is like sum of distribution functions, one is for 0.0 tips and one for >0 tips. That may suggest that classifying if tip will be given first is a good idea before actual tip amount prediction. Tip and Fare amount distributions are skewed so predicting them with linear regression may not give us good results. Instead we will try to use Random Forest and Neural Networks for predicting those variables.

Vendor and payment type are both can be described well by Bernoulli distribution(as number of payments other than by card or by cash can be neglected), however vendor type distribution is very close to uniform with two states as you can see on the picture:

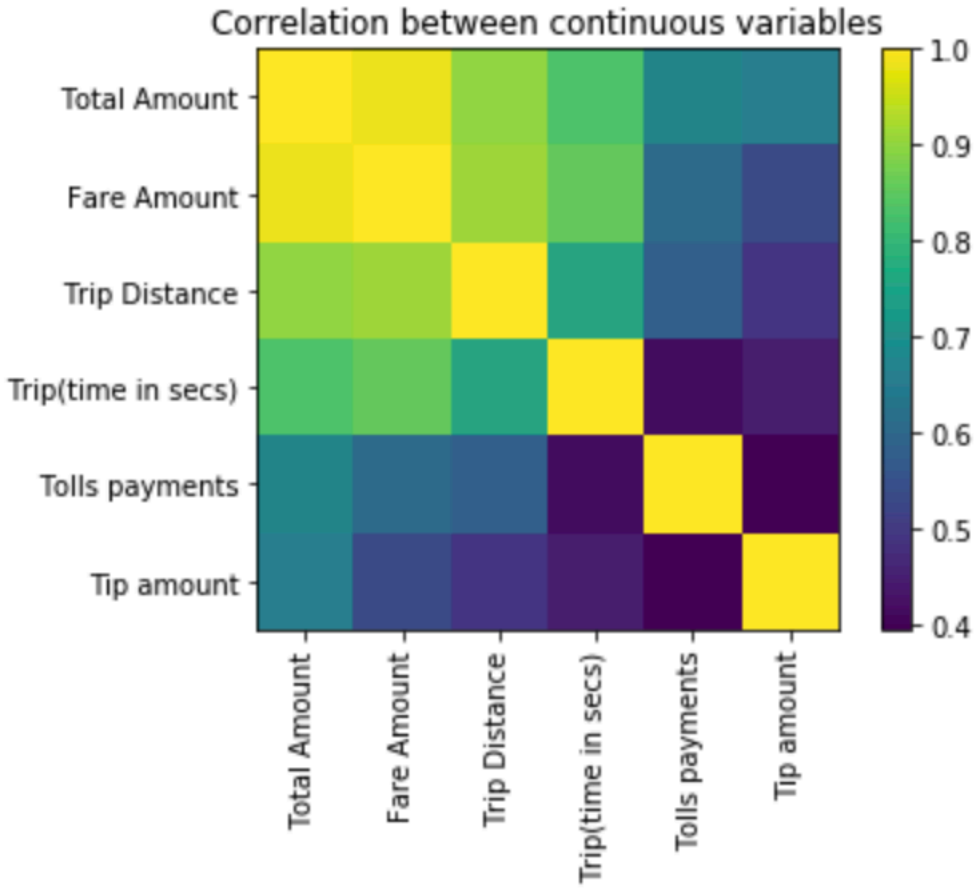


Passengers number per trip can be described by discrete Poisson distribution:



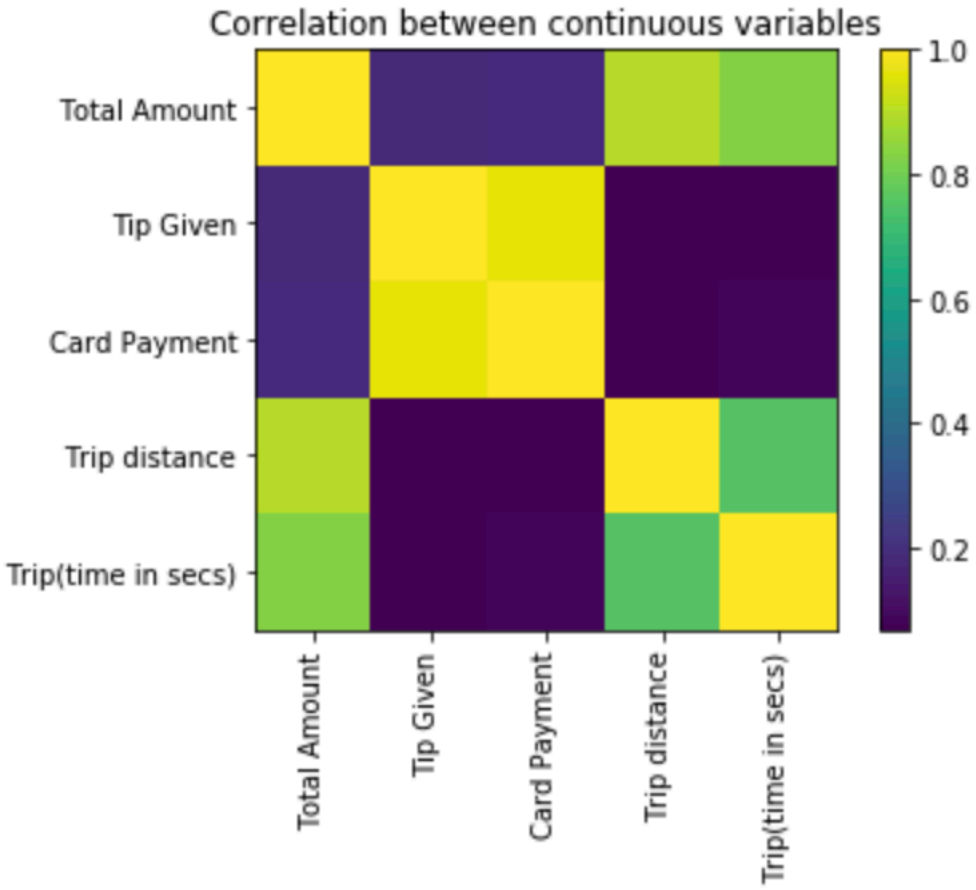
Most of the people using taxi by their own or with a friend(such a waist of natural resources of our planet :).

One of the tasks of this project is to predict fare and tip amount based on coordinates of pick up and drop off, so let's plot a graph that would show how those parameters are interdependent:



Looks like tip amount is very independent variable and it seems it will be hard to predict it. Fare amount correlates to trip distance which depending on time of a day and a week should correlate to trip distance. Total amount is of course to fare amount, that is probably why we have been asked to predict fare or total.

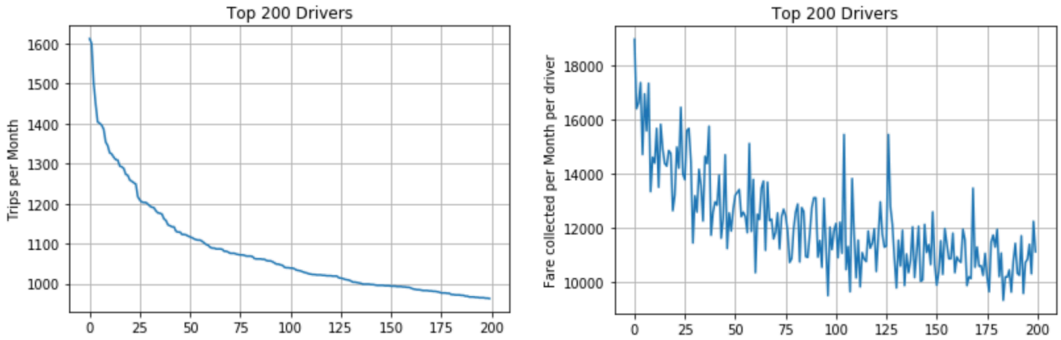
One more hypnotizes I would like to check with Tips which was mentioned in one of the sources. People found that most of tips were given while paying with credit card. Logical explanation would be the fact that money is less tangible when handled using credit cards. Let's see if there is correlation between payment type and tips given or not:



Here we can see that it is indeed correlates, very interesting and can be useful for our tip prediction model.

People

Corresponding notebook is in 09_People.



Source [4] investigated people data heavily and found some truly amazing results. For example, they separated pro drivers from new drivers and found that on average pro drivers earn 7% more that new ones, but it only takes about 100 trips for new driver to hit the same earnings, because humans learn.

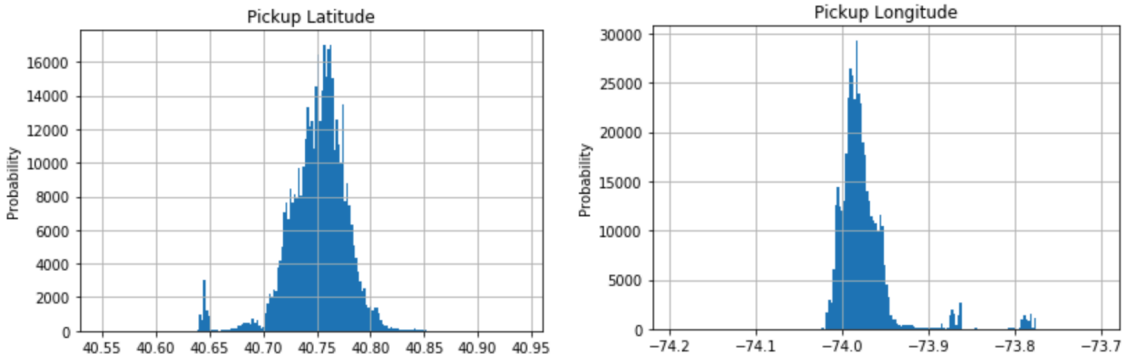
Unfortunately, we could conduct a decent people analysis due to time restrictions. However here are some ideas of what could we do. It seems that are a lot of interesting insights we can get about taxi business from people and car utilization data. Here there are a couple of ideas worth exploring:

1. We can look on median earnings per taxi and per driver.

2. Based on number of trips per months we can separate new drivers to professionals. In [4] they separated them base on earning and number of trips by certain day of the Month.
3. We can then explore where pro drivers their earnings per month and day. This can be a good business pulse check for our Taxi company.
4. We then can explore driving patterns of those people and see main sources of their earnings/trips.
5. We possibly can create a simulator and apply [deep Q] reinforcement learning technics for decision making regarding direction of driving after drop off.

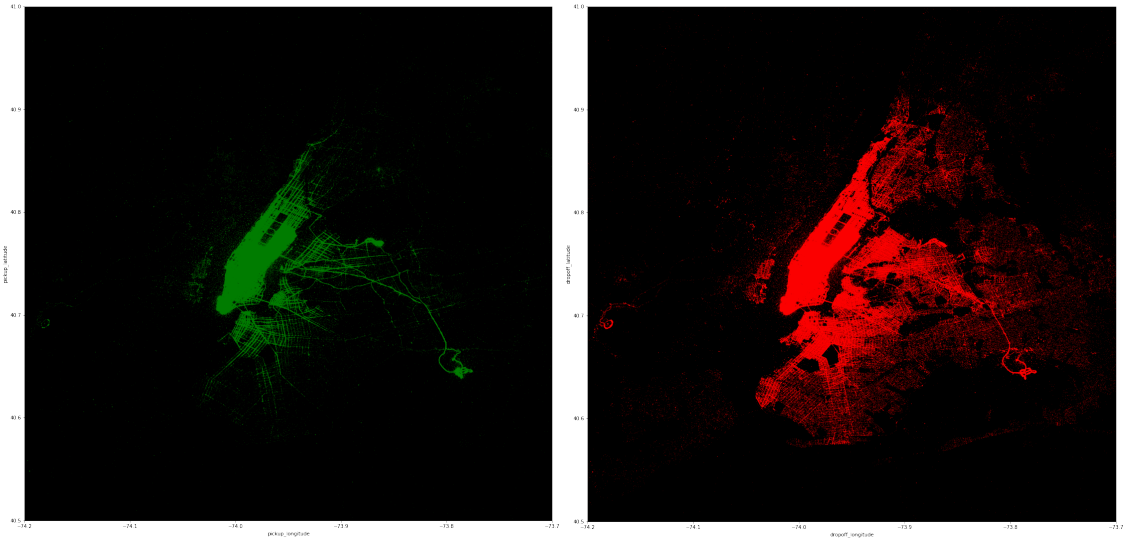
Space and Time

We will start by exploring coordinates of pickups and drop offs. Corresponding notebooks are 05_SpaceAnalysis and 04_Clustering.



Looking like a normal distribution which may mean that there is one busiest area of the city.

When we talk about geographical data we better show it on the map. Here is the overall geographical distribution of pickups (Green) and drop offs (Red):

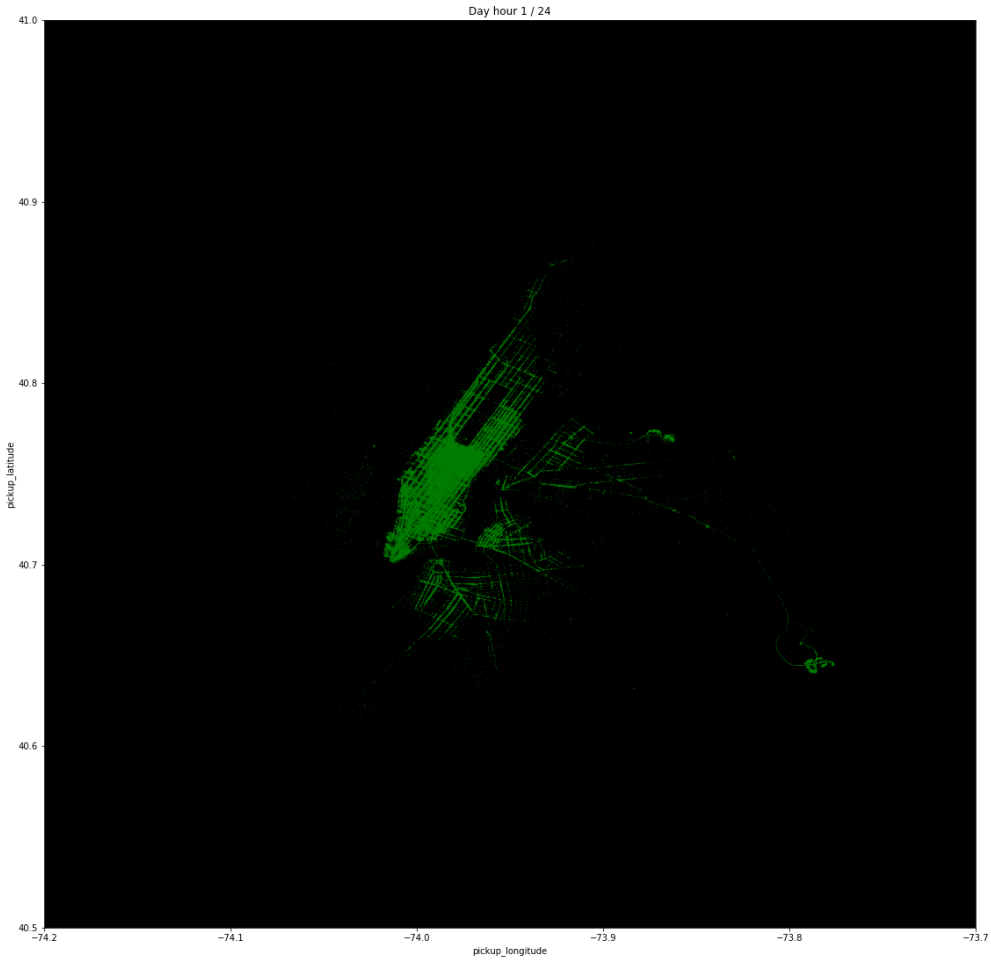


(better resolution files can be found in folder 'images').

This plot gives us a good sense of city business, and certainly the main areas of the city that participate in transportation. The first one is a business center – Manhattan. We also can

see highlighted three Airports. We can see that drop offs more sparced and cover ares of Brooklyn and Queens. About 5.7% of all trips are from or to airports.

Using code in 02_hourly_Visualisation we can plot hourly plot business for 24 hours(GIF does not seem like supported in Word, so please have look on files 'images/day_animation.gif' and 'images/week_animation.gif':



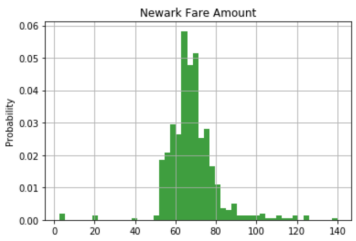
This will give us a great sense on which areas of the city the most busiest in terms of pick ups and when in terms of time. This visualization by itself can be used to instruct drivers on best pick up opportunities around the city based on time.

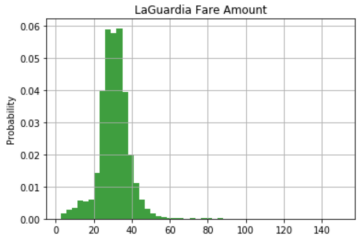
Let's apply cluster analysis (k-mean clustering, please see code in 03_Clustering notebook) to find the biggest 20 clusters(please check file cluster_map.html). Following locations marked with numbers:

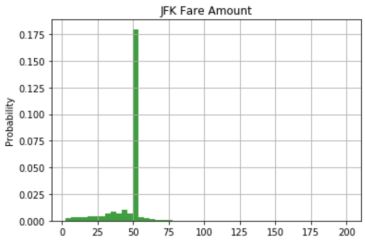
1. Newark airport. About 0.1% of all trips start or end in this airport.
2. LaGuardia airport. About 3.3% of all trips start or end in this airport.
3. John F. Kennedy airport. About 2.2% of all trips start or end in this airport.

4. Manhattan. About 96% of all trips start or end in this borough. 92% of all pickups is happening here. (code is in 04_Space_and_Time).

It would be interesting to check fares related to airport trips:

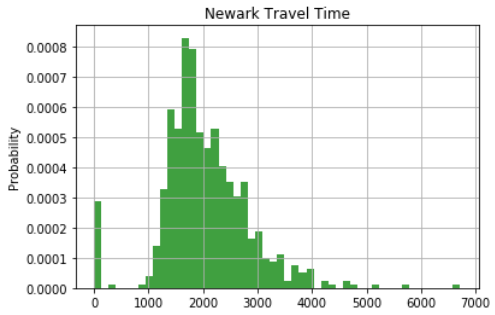




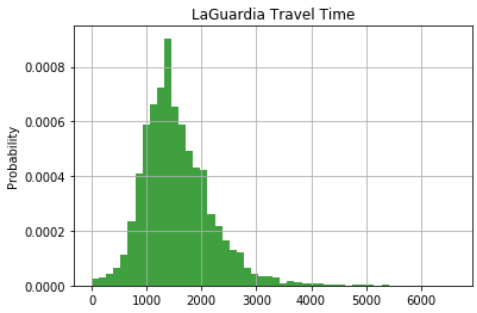


Looking like JFK airport has some regulation in place regarding fare price of travel to/from it and it is the most consistent fare of all of them. (Later on I found that it is indeed the case for JFK airport to have flat rate charge from and to Manhattan of $52: http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml)
Now let's see which is quickest airport to get to in terms of time:

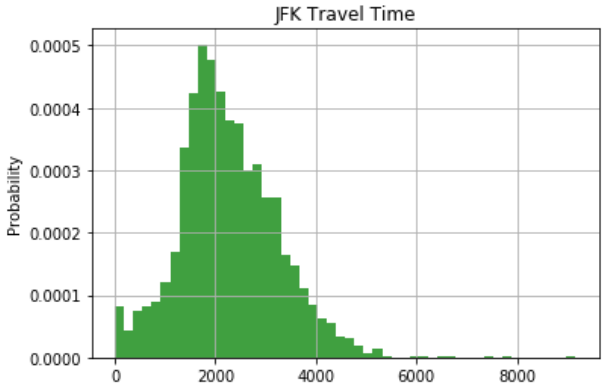
Standard Deviation of travel Time: 814.6858133766



Standard Deviation of travel Time: 627.1934272749



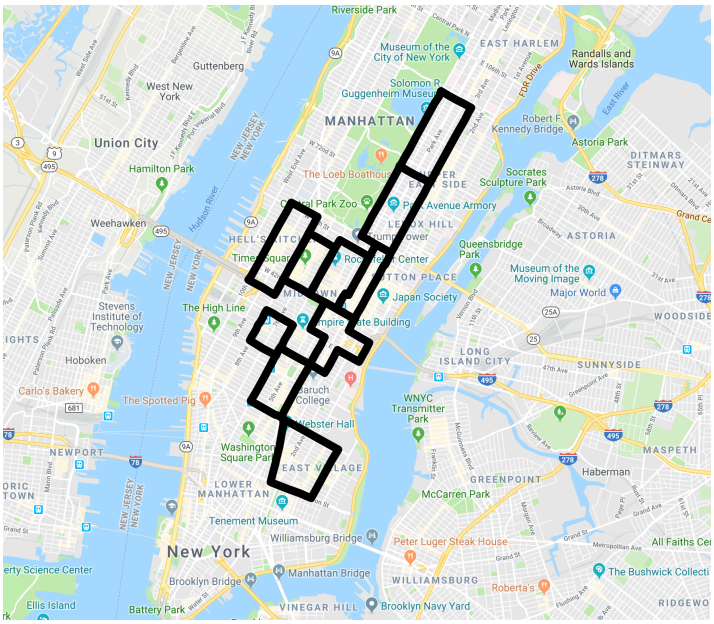
Standard Deviation of travel Time: 978.5380580296



JFK airport seems the worst airport to travel from/to as the travel time to it is very inconsistent and may vary a lot. May be that is why there is a flat rate fare for it.

In order to map trips to suburbs we acquired taxi zones data which is a name of borough and taxi zone with it's polygon GPS coordinates (https://s3.amazonaws.com/nyc-tlc/misc/taxi_zones.zip). Then we run an algorithm checking if trip starting or ending point belongs to the polygon. We mapped 3% of merged trip and fare data and will use for prediction problems. Low computation efficiency did allow to map more, we run several processes on a couple of computers for a night and got ~3% of all data mapped. That certainly can be improved and paralleled more. Code for this mapping can be found in file **identifySuburbs.py**.

And here are **busiest suburbs** (in terms of number of pick ups) according to 3% (441036 trips) sample of data that we managed to map against "Taxi zones" data:

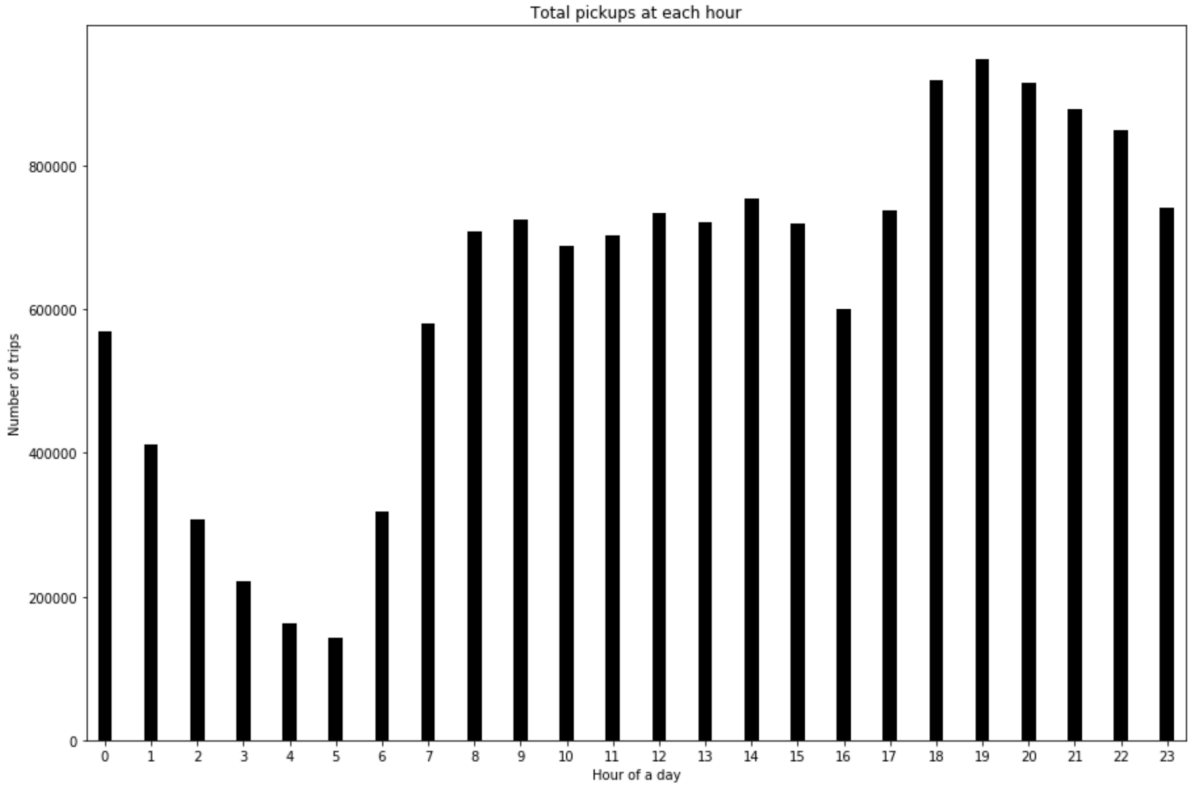


Suburb	Pick ups	%
Upper East Side South	16770	3.8
Midtown Center	15580	3.5
Midtown East	15085	3.4
Upper East Side North	15077	3.4
Murray Hill	15035	3.4
Union Sq	14871	3.3
Times Sq/Theatre District	14500	3.2
East Village	14313	3.2
Clinton East	13883	3.1
Penn Station/Madison Sq West	12798	2.9
…	…	
LaGuardia Airport	9517	2.1
JFK Airport	6994	1.5

Airport locations are given for the reference.
Essentially these are the most important parts of the city for picking up passengers and conducting Taxi business. This goes along well with our space/time visualization for pickups.

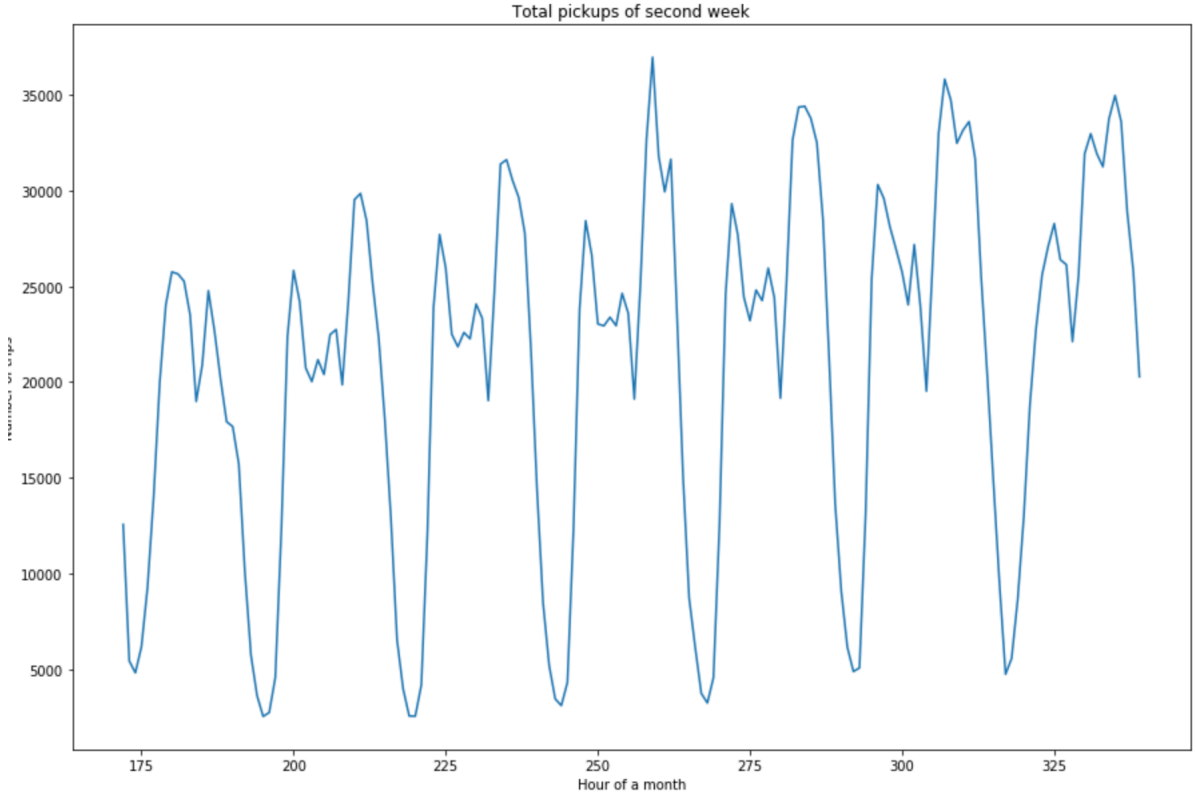
Time

Let’s plot a graph that can show us average trips per day hour:



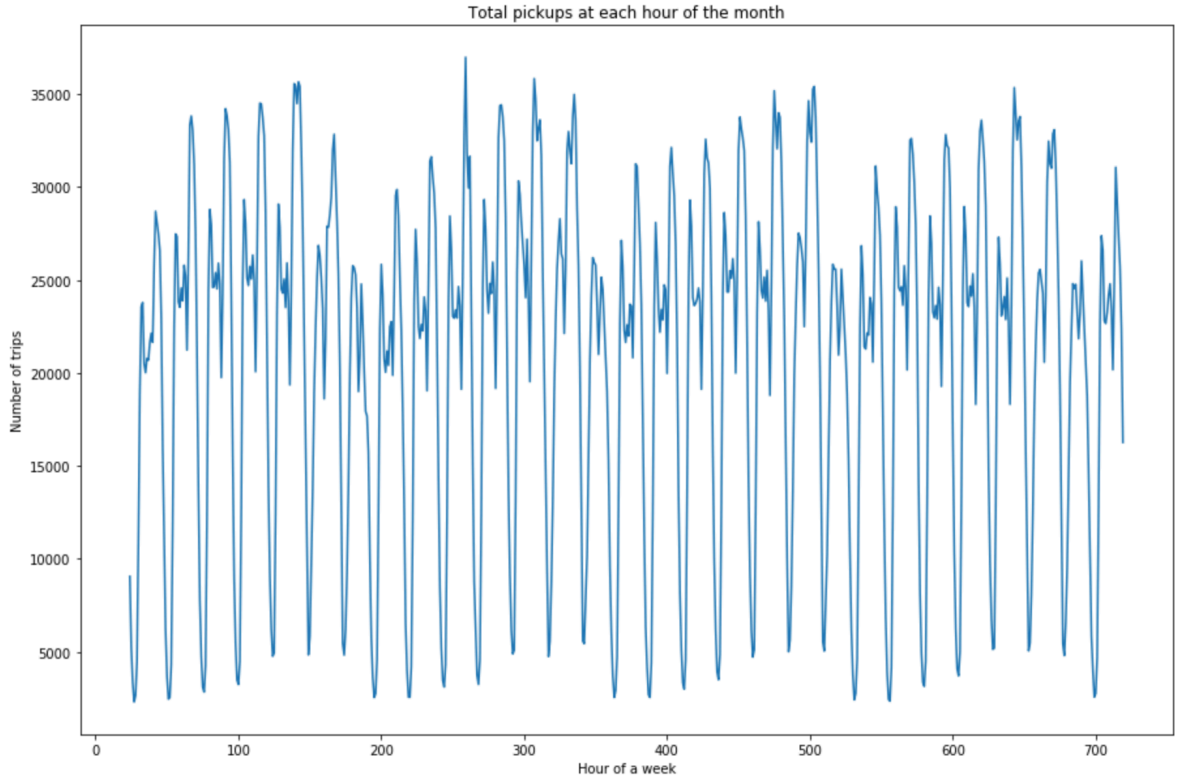
This is clearly non-linear seasonal data and it seems that at 4pm people are changing shifts. **Top 5 busiest hours are 18:00-22:00 clearly from this graph.** Seems that afternoon shift will collect more in fares which can be checked.

It would be interesting to plot weekly data pickups per hour (we take second week as our data starts on Tuesday of the first week):



We can see seasonal component (same variation during different days) and slight growing trend during the week from Monday to Sunday.

So lets have a look on month data graph(data starts on Tuesday 2nd April):



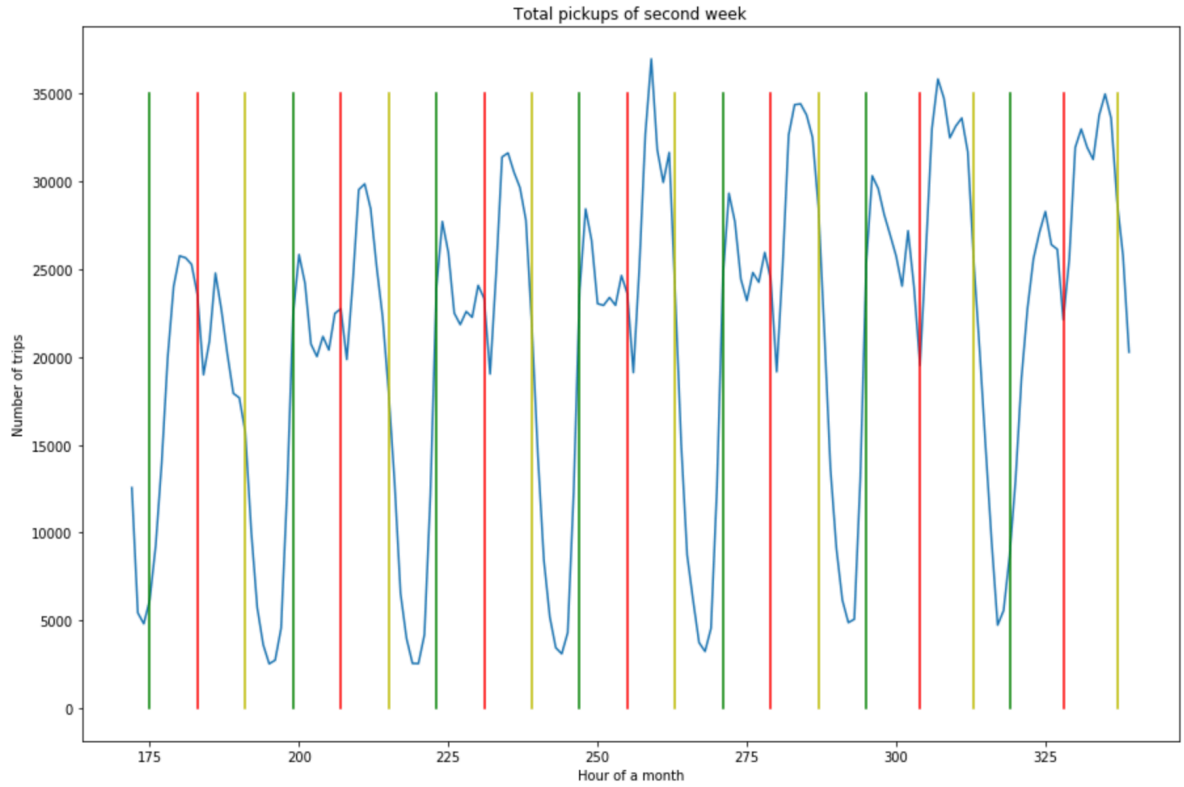
Very seasonal data, clearly number of pickups growing towards the end of the week. Mondays and Tuesdays have lowest pickups number. There is a big gap at midnight, so no point working during that time unless it is Friday or Saturday.

Maximizing earnings in a day

To maximize earnings in a day owner should organize three 8 hours long shifts and try to move to locations in Manhattan where pick ups are happening even at night. Owner ideally want to hire professional drivers that can bring back more in fares. Targeting people who travel to JFK Airport can be beneficial as it has flat rate fare which is above median of fares.

Reduce work time keeping earnings

Let's have a look on a graph:



To minimize work keeping the earning we need to work during the time when people need taxi. On a graph we will have the first shift between green and red lines, second between red and yellow lines. We will not work between yellow and green line.
as following:

1. We will organize 2 shifts during a week.
2. Our shift will be 8 hours.
3. First starts from 8am and finish it at 4pm.
4. We will start second shift at 4pm and finish it at midnight.

This strategy seems very logical, but have to be verified statistically.

Prediction modelling

Fare amount

Code is in the file 07_PredictionFare. We now can use our knowledge about Fare pricing model for prediction purposes (mentioned in basic research chapter). Ideally we would want to know the following:

1. Route and distance of the trip based on pick up drop off coordinates.
2. Traffic on the streets(as a function of time of the trip).
3. Number of traffic lights on our way.
4. Our cost of the trip based on distance and estimated duration of the trip.

Here we will try to predict fare amount based on following information:

1. Pick up drop off location.
2. Time of a day, week.

For this exercise, we used sample of 3% of merged data which also had pick up and drop off locations mapped to NYC suburbs using Taxi zones file.

Out of those features we can develop some more:

1. Pick up/drop off can be mapped into suburbs based on open Taxi zones data.
    a. Other papers used grid boxes or clusters information.
2. We can deliberately look for airports and Manhattan borough as it can be important.
    a. For JFK<->Manhattan trips we will have a perfect estimate $52.
    b. One hot encoding for trips from and to Manhattan.
3. Time of day/week – we will map to hour of week/day. Minute of the hour did not prove to be useful.
4. We will apply normalization for our features to make model to converge faster.
5. We will Principal Component Analysis to improve our features independence.
6. We will apply logarithm function to de-skew the target variable.

I fitted two models:

1. Random Forest. (Tried different number of estimators 50-1500, increasing number of estimators beyond certain point ~500 make model worse. Setting minimum number of leafs helped to improve the model.)
2. Neural Network.(Tried many different architectures 2,3,4 layers structures. Looks like the best structure has number of features+1 neurons in input layer and not hidden layers.).

Keras model parameters search with 30 epochs:

	Mean Squared Error on CV (smaller – better)	Variance Explained Score (bigger better)
NN 10x1	0.00262	0.859
NN 14x1	0.00260	0.861
NN 16x1	0.00251	0.865
NN 17x1	0.00254	0.864
NN 18x1	0.00252	0.865
NN 18x2x1	0.00255	0.863

Parameters/structure search got both models to similar performance on test/training/cross validation sets. Best models found:

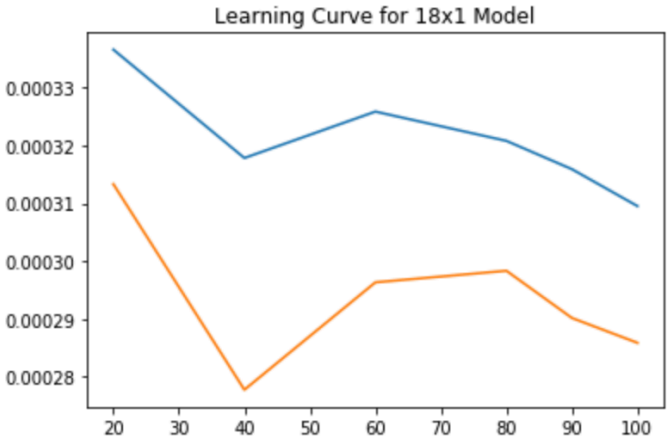
	Mean Squared Error on CV (smaller – better)	Variance Explained Score (bigger better)
Random Forest 300	0.00286	0.8749
NN 18x1	**0.00279**	**0.8782**

Formally speaking we can not apply these scores unless our predicting variable is normally distributed.

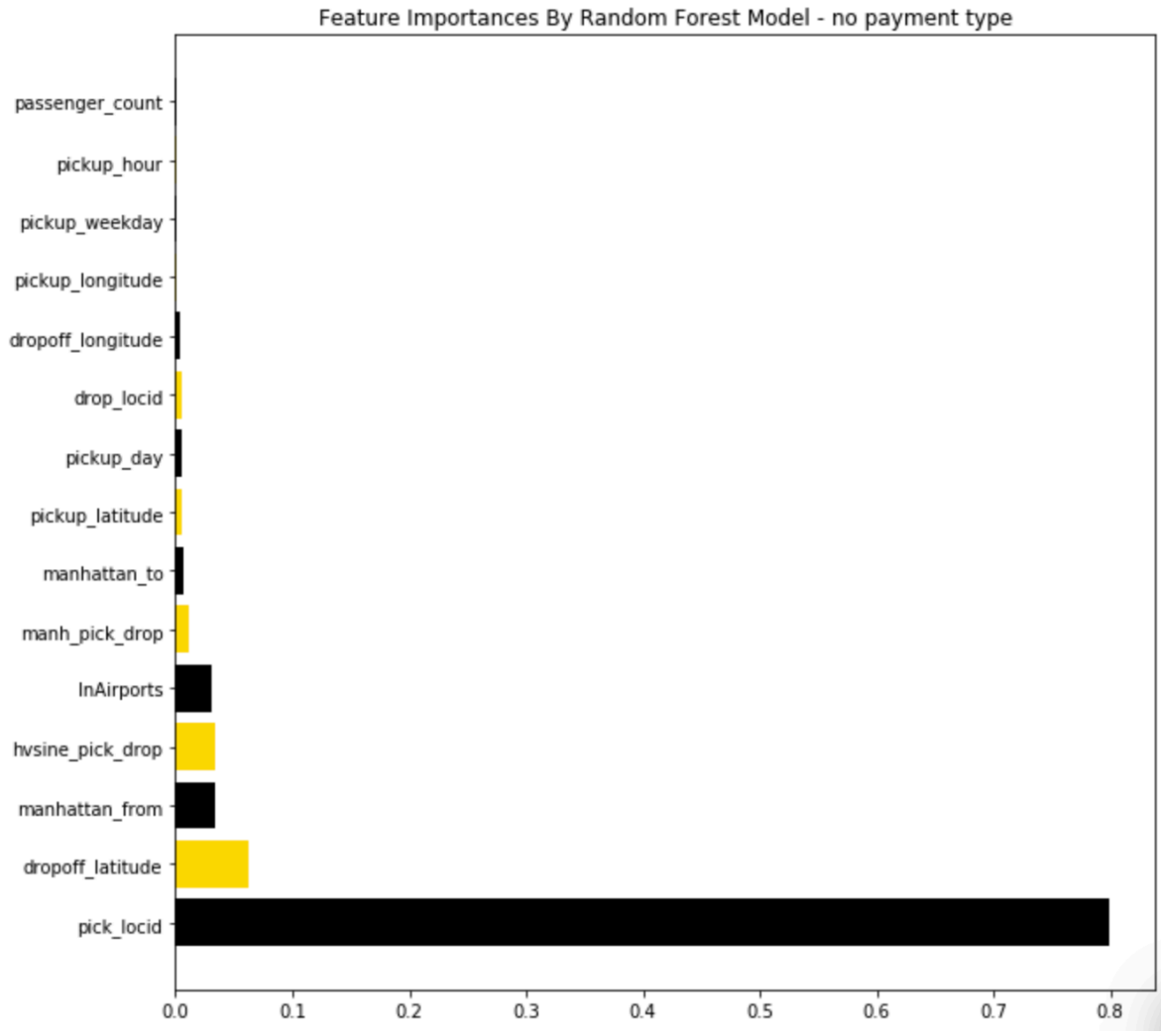
When we apply same algorithms for de-skewed target :

	Mean Squared Error on CV (smaller – better)	Variance Explained Score (bigger better)
Random Forest 300	**0.00204**	**0.8389**
NN 18x1	0.00205	0.8388

Learning curve created for the best model suggested that model can be improved with more data(so we would map more data to Taxi zones and repeat the training):



For random forest it is very interesting graph to explore feature importance for fare prediction:



Predicting Tip Amount

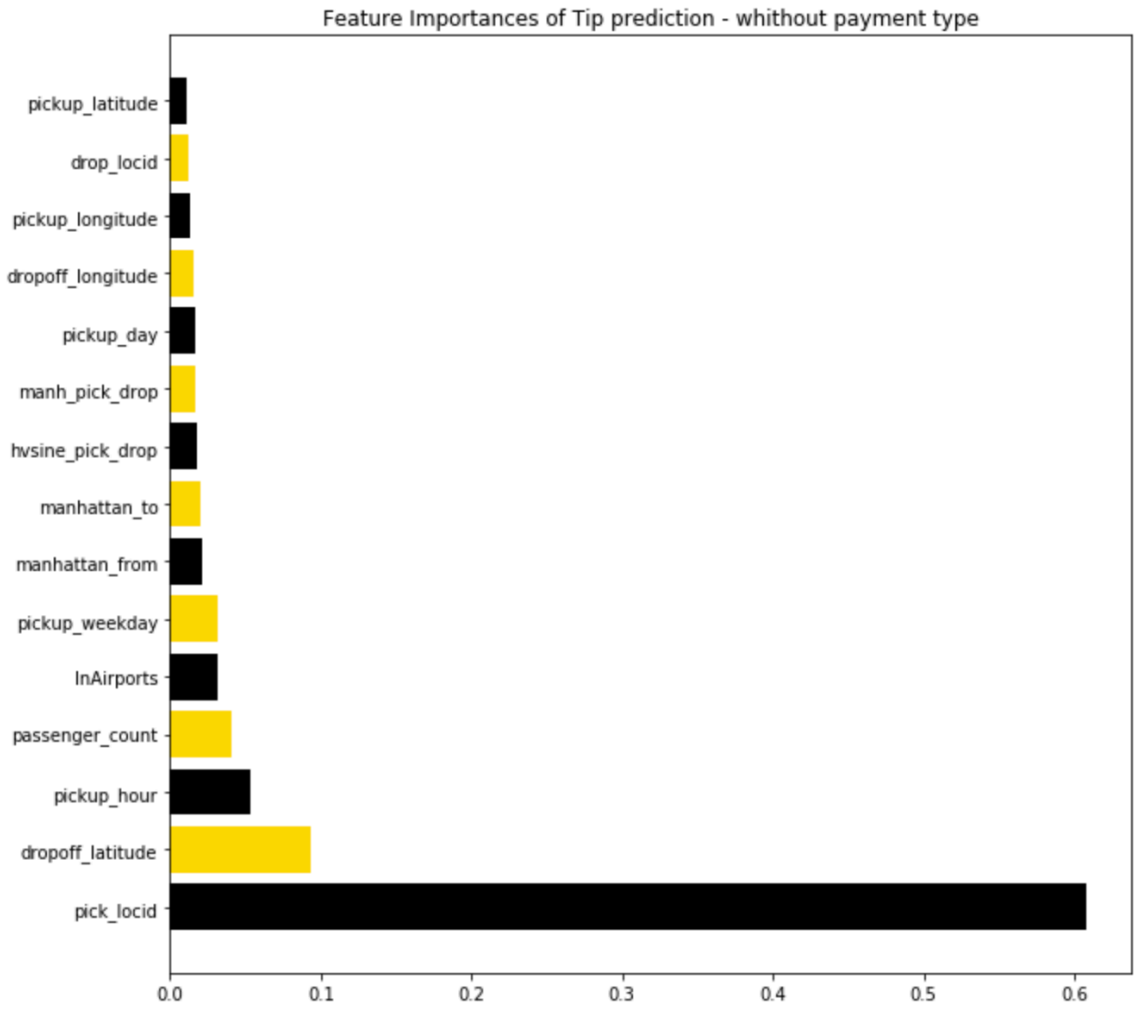
Predicting tip amount turned out to be more complex task than fare. Corresponding file 06_PredictingTip. We did not have time to explore full algorithm as follows:

1. Calculate direct distance and possible predict duration of the trip based on time.
2. Predict fare amount. Based on location, taxi zones attribution, hour of week.
3. Use estimated distance and time of travel.
4. Classify if tip will be given.
5. Predict tip for cases when we predicted it will be given.

Instead we just tried to predict tip amount in dollars and tip percent of the fare. Most of the literature about this problem predicts tip ratio and uses algorithms like random forest regression, random forest classifiers or gradient boosted decision trees[5] which are easy to try in Python.

Initial model performance was not really good:

	Mean Squared Error on CV (smaller – better)	Variance Explained Score (bigger better)
Random Forest 300	**0.00075**	**0.134**
NN 18x1	0.00087	0.122



Both NN and Random forest could not comprehend the data for tip prediction. The suggested improvement comes from fare prediction. We possibly should predict fare first and then use it for tip prediction or classify if tip will be given first and then make the prediction. We did not explore this way due to time restrains.

Correlation analysis found that tip correlates with usage of credit card, so we added that feature to the final model as one hot parameter (=1 when payment made by card). Although certainly it is an interesting task by itself to be able to predict if customer will be using credit card or not in the real-life scenario. Can be used as an insight for drivers to collect more tips.

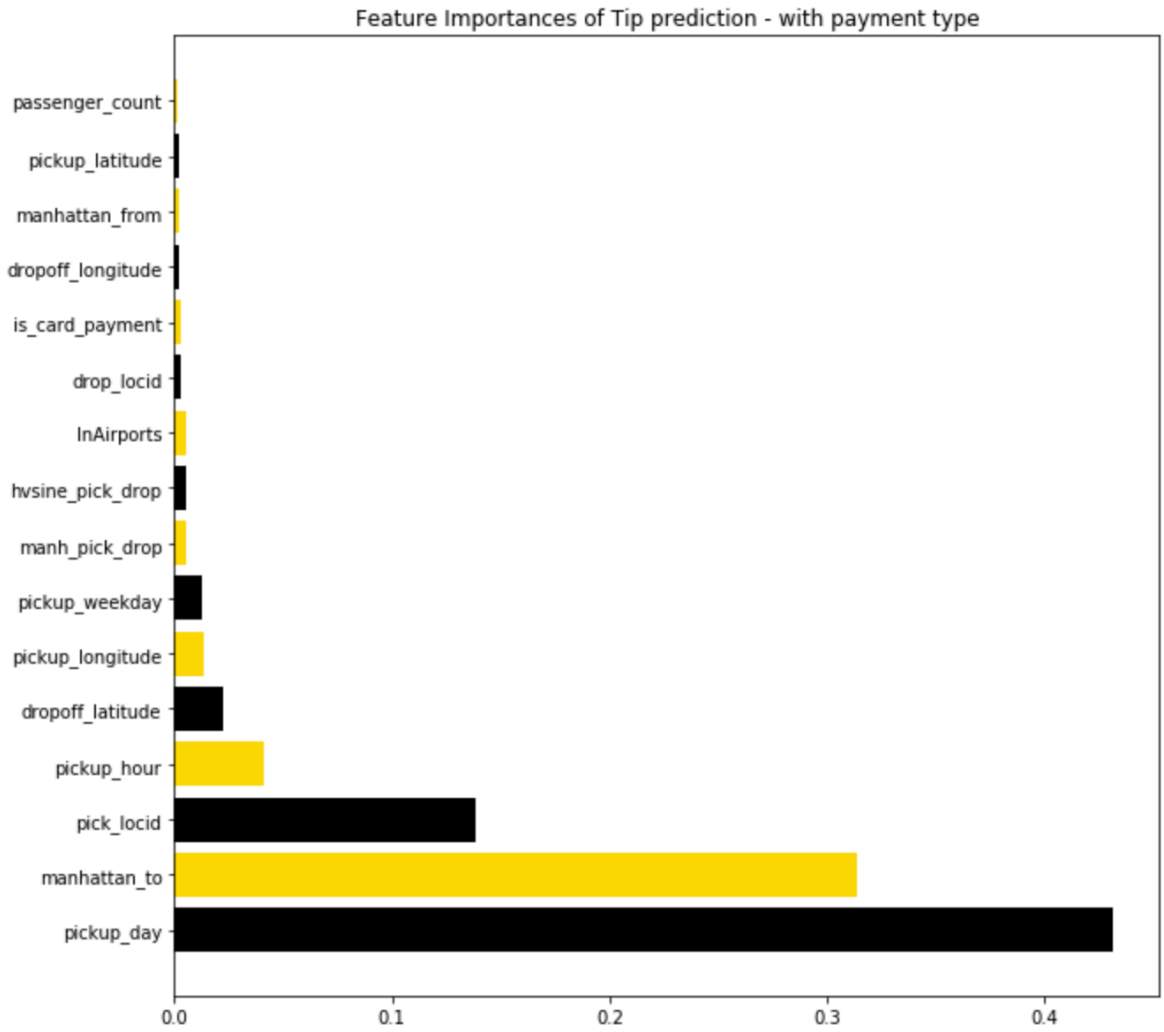
Tip amount can take zero values so we used this formula for de skewing the target variable:

$$Target = \log(Tip\_amount + 1) / \max(\log(Tip\_amount + 1))$$

All of that helped to improve the prediction model:

	Mean Squared Error on CV (smaller – better)	Variance Explained Score (bigger better)
Random Forest 300	0.0059	0.74
NN 18x1	**0.0030**	**0.86**

Features importance:



Interestingly 'is_card_payment' feature is not at the top, but it definitely the one that sharply increased the quality of a prediction model.

Maximizing 10 taxi earnings

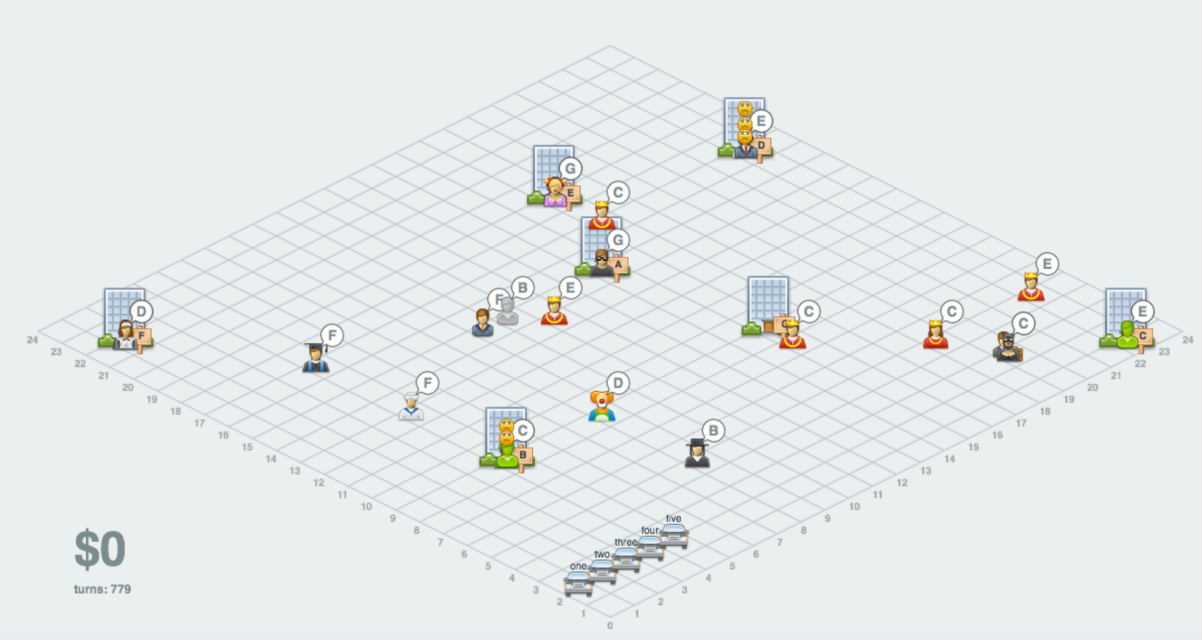
If you have a fleet of vehicles it is important to organize their work in order to optimize certain reward function. Reward can be just earnings a day or better be revenue (earnings-costs) per day. It sounds unrealistic that we are going to micromanage drivers around the city, so here is how we realistically can help drivers:

1. Optimal taxi allocations.
2. Choosing fastest/most economical routes between pick up and drop off points. Shortest driving time trips. [6]
3. Motivating drivers to make economical decision where it is possible. Salary based on fares collected for example.
4. Analyzing our cost model and optimizing it.

However for this exercise we just pretend that our fleet can be controlled fully as if like we have fleet of self-driving cars that follow all our orders.

This sounds like a dispatch taxi problem or dynamic vehicle routing problem which has some simulations and even online challenges: https://www.getlocalmotion.com/code-challenge

This is a bit more complex problem to implement it during those 10 evenings and we would suggest using Reinforcement learning for it.



We can use information about pick ups data probabilities and simulate random requests to our taxi service. People would appear and request a ride and our goal will be to send them the taxi. Old approach would use multi-objective Pareto function, modern can just use reinforcement learning. Taxi allocation can initially follow some simple epsilon greedy policy selecting closest available taxi with 1-Epsilon probability and choosing random available taxi with Epsilone probability.
We would need to define space(it can be a location grid space) and action spaces(for example: move to neighbor grid cell, finish shift, go to airport stand) for drivers strategies and also robust reward function(fare collected) for RL model. Then we can train reinforcement learning algorithms like approximation-SARSA in order to iteratively evaluate policy and found the optimal one.

Resources

N	Source	File
1	Description of data	data-dictionary-trip-records-yellow.pdf
2	NYC Taxi Fact Book	2014-taxicab-fact-book.pdf
3	Taxi business ROI from switching to hybrid vehicles.	Taxi-ROI-Paper.pdf
4	Learning by Driving: Productivity Improvements by New York City Taxi Drivers	Learning-by-Driving.pdf
5	NYC Taxi Tip-Rate Prediction	NYC-Taxi-Tip-Rate-Prediction.pdf
6	Simulation and dynamic optimization of taxi services	Sim-taxi-optimisation.pdf
