Data Analytics Project Report

New York Taxicab - Tip Prediction

A picture containing text, sign

Description automatically generated

CONSTRUCTOR UNIVERSITY, BREMEN

January,2023

Student: Professor:

Arben Rexhepi Dr. Wilhelm Adalbert F.X.

## 

**Table of Content**

1. Executive Summary
2. Problem Scoping
3. Dataset Attributes Information
4. Data Preprocessing
5. Feature Engineering
6. Data Exploration/Data Analysis
7. Modeling
8. Model Evaluation/Comparision
9. Conclusion
10. References

Executive Summary

Task :

To perform descriptive and engineering analysis of the dataset by understanding of machine learning for using the models to predict the tip amount of passengers that are given by taxi service users.

Dataset Background :

The dataset is taken from the TCL New York City Taxi, that represents data collection during the period of 2016 Feb. It includes features: pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. These features are important to estimate the target variable tip amount.

Approach:

As the target variable “tip amount” consists of real numbers that means is a numeric variable, we must do a regression task. Before training the ML models, we need to make some changes, like add new features to the dataset which help the ML models to predict the tip amount. After feature engineering and data exploration, we are going to use three models. The selection of the model is done by their time and space complexity. We choose Linear Regression as baseline model and then train more complex models like Decision Trees and XGBoost.

|  |  |  |
| --- | --- | --- |
| **Model Name** | **RMSE** | **R ^2** |
| Linear Regression | 0.42 | 0.97 |
| Decision Tree | 0.568629 | 0.95 |
| XGBoost | 0.51857 | 0.96 |

Results:

Problem Understanding

A group of yellow cars on a street with buildings in the background

Description automatically generated with low confidence

Taxicabs are, and always will be, an iconic part of New York. The history of their characteristic yellow colour goes as follows: owners of cab companies painted their fleets a distinct signature colour, resulting in cabs ranging from brown, white, red, and even checker ones. And some were yellow. After a few years, two big cab companies decided that yellow was the way to go, with both ultimately contributing to the tradition of yellow cabs in New York City. These companies were the Yellow Cab Company, started by John Hertz in Chicago in 1910, and the Yellow Taxicab Company, which was incorporated in New York by Albert Rockwell in 1912.

Trying to predict incomes has become a pillar of any transportation company namely taxi ones which are looking always to stand out against the online travel business high demand (Uber and other ways of traveling). In this article, I am going to predict the tip amount for a taxi ride in New York City, according to the suitable features which will be discovered thanks to the beforehand several data operations.

All along this project, I will dive into not only data ingestion and cleansing but also feature engineering and selection and in the end models building.

Why are so important taxi services?

The taxi services have completely changed the urban transportation industry. Although, it has created many job opportunities for drivers and helped them to feed their families by earning a pleasant amount of money. On the other hand, it created chances for riders to travel long distances without owning a car. However, there are still some problems that we should address like:

1. Shorter travel acceptance time
2. Difficulty of finding a taxi at remote locations
3. Difficulty of finding taxis during odd hours
4. Cost Pricing

In my country, Republic of Kosovo the average waiting time to find a cab is around 10-15 minutes, which is alarming, as it would make customers to choose another company. However, in some cities the lack of competitors is another reason that contributes to the waiting time and making it so enormous.

Taken Into the consideration the statistics of Taxi cabs In my own country as well the statistics that are about New York Taxi cabs we can see some analogies.

Based on statistics of my country, taxi cabs are more opted to receive tip amounts when they are picking-up or dropping-off passengers In rich neighbourhoods, ridding people in long distances and taking or sending people In the airport

Another feature that It does not have to do with distance or condition but that Is related with the driver itself is: If he has an easy going character and sometimes even pays attention to passengers requirements like play passengers favourite music or chats with him. The taxi driver will get a good tip.

Let’s, take few examples to understand in which situations we can have chances to accept tips:

1. Traveling taken from airport which are outside of the city. (National Airport- Adem Jashari, Republic of Kosovo)
2. Traveling at the peak rush hour after work (usually between 4:30pm-5pm)
3. Traveling from a FIFA world cup stadium
4. Traveling from a distant Hotel (Resort Arberia, Republic of Kosovo)

# 

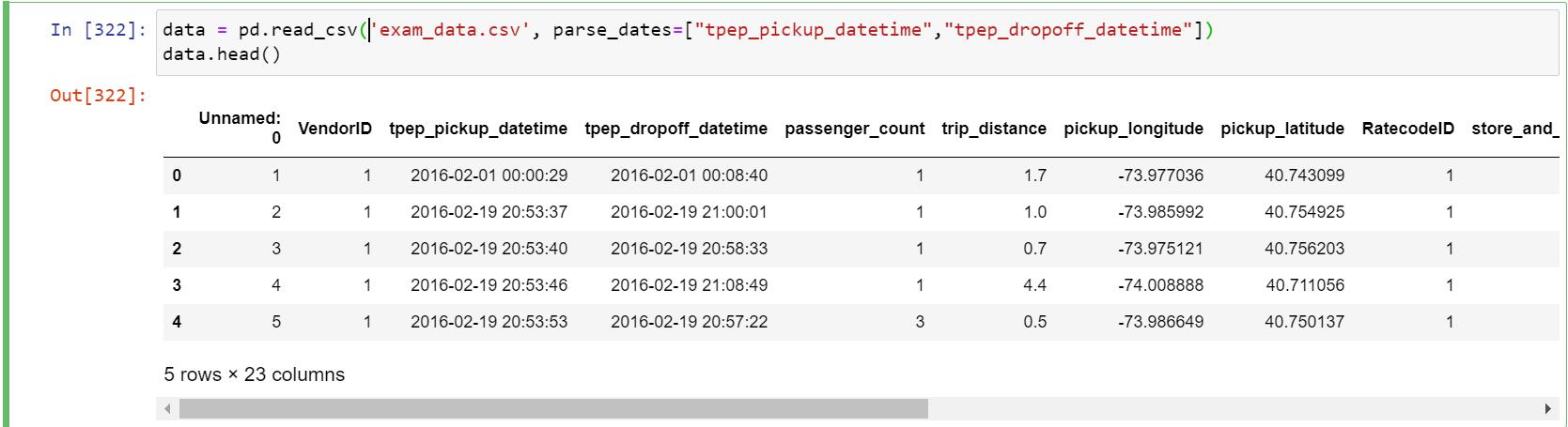
# **Dataset Attributes Information**

**Description:** The aim is to predict the tip amount for a taxi ride in New York City taken into consideration some of the features. The Data set is taken from the TCL New York City Taxi data collection, we are looking at the time of 2016 Feb.

|  |  |  |
| --- | --- | --- |
| S.No | Variables Name | Description |
| 1 | VendorID | * A code indicating the TPEP provider that provided the record.   - Creative Mobile Technologies  - VeriFone Inc. |
| 2 | tpep\_pickup\_datetime | * The date and time when the meter was engaged. |
| 3 | tpep\_dropoff\_datetime | * The date and time when the meter was disengaged |
| 4 | Passenger\_count | * The number of passengers in the vehicle. This is a driver-entered value. |
| 5 | Trip\_distance | * The elapsed trip distance in miles reported by the taximeter. |
| 6 | Pickup\_longitude | * Longitude where the meter was engaged. |
| 7 | Pickup\_latitude | * Latitude where the meter was engaged |
| 8 | RateCodeID | - The final rate code in effect at the end of the trip.   * + Standard rate   + JFK   + Newark   + Nassau or Westchester   + Negotiated fare   + Group ride |
| 9 | Store\_and\_fwd\_flag | * Store\_and\_fwd\_flag: - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka “store and forward,” because the vehicle did not have a connection to the server.   + Y= store and forward trip   + N= not a store and forward trip |
| 10 | Dropoff\_longitude | * Longitude where the meter was disengaged. |
| 11 | Dropoff\_ latitude | * Latitude where the meter was disengaged. |
| 12 | Payment\_type | * A numeric code signifying how the passenger paid for the trip.   + Credit card   + Cash   + No charge   + Dispute   + Unknown   + Voided trip |
| 13 | Fare\_amount | * The time-and-distance fare calculated by the meter. |
| 14 | Extra | * Miscellaneous extras and surcharges Currently, this only includes the $0.50 and $1 rush hour and overnight charges. |
| 15 | MTA\_tax | * 0.50 MTA tax that is automatically triggered based on the metered rate in use. |
| 16 | Improvement\_surcharge | * 0.30 improvement surcharge assessed trips at the flag drop. the improvement surcharge began being levied in 2015. |
| 17 | Tip\_amount | * Tip amount – This field is automatically populated for credit card tips. Cash tips are not included. |
| 18 | Tolls\_amount | * Total amount of all tolls paid in trip. |
| 19 | Total\_amount | * The total amount charged to passengers. Does not include cash tips. |
| 20 | GoodTip | * Categorical variable indicating an above average tip |
| 21 | Extra | * An indicator for additional charges included. |
| 22 | Cash | * An indicator whether payment was made by cash or not |

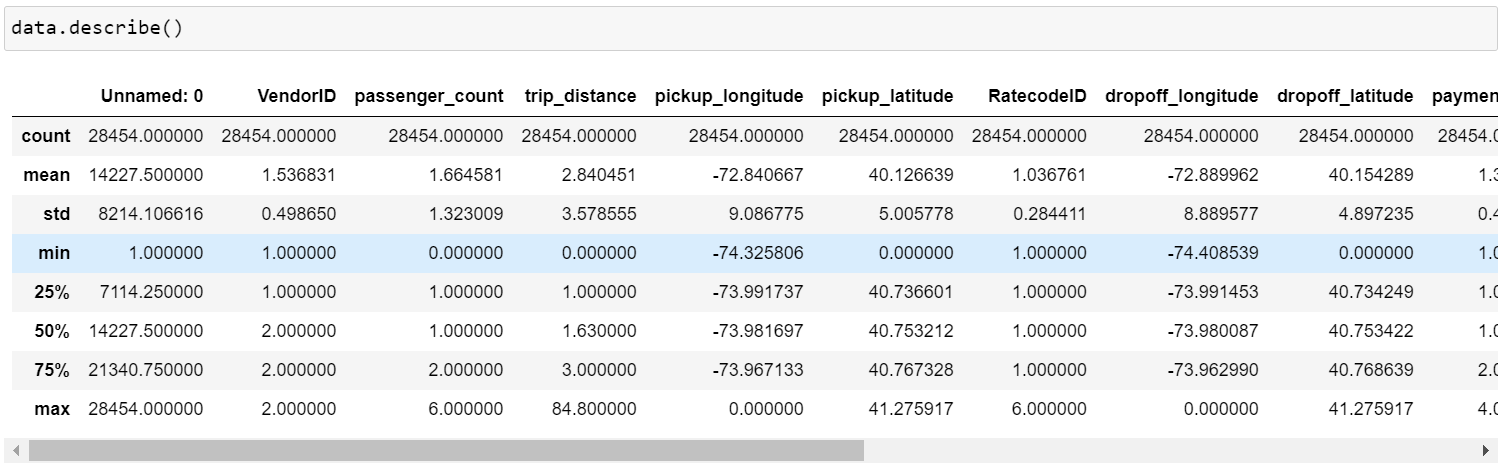
## **Data Preprocessing**

## **Step 1: Loading the dataset**



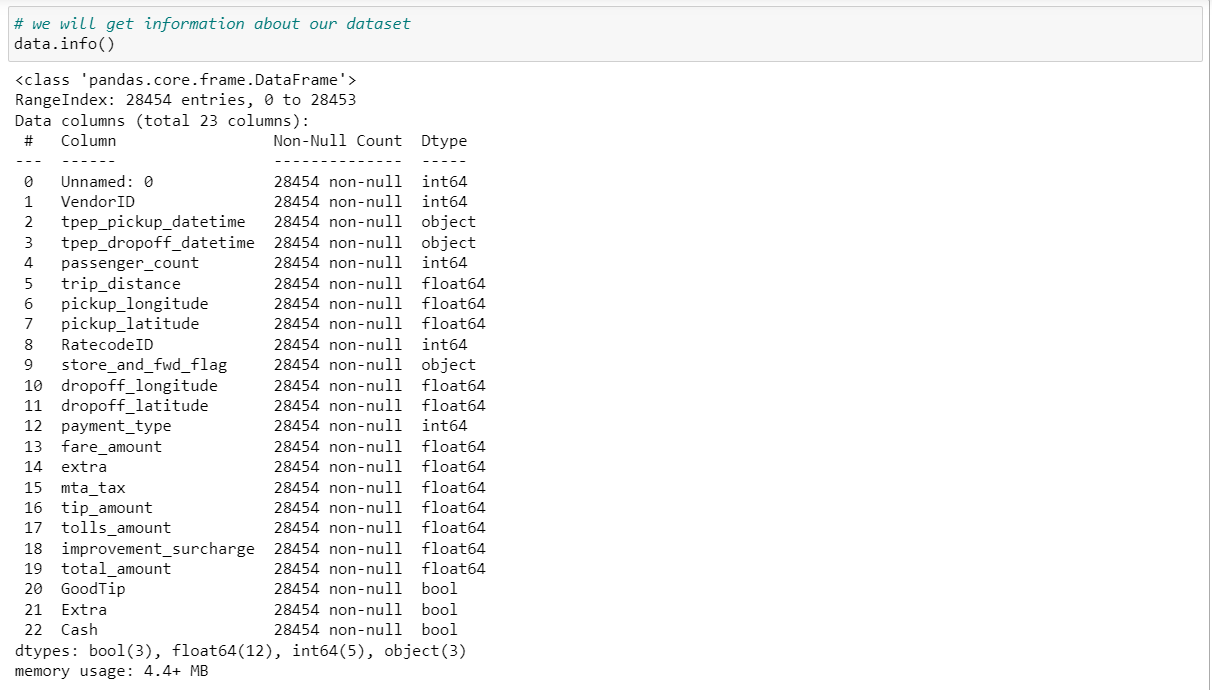
* We load the dataset using the read\_csv method from Panda’s library.
* We used the parse attribute of read\_csv method to handle the pickup/drop off date and time.
* We observed that there are few variables which are object data type that means they are not int or float type. We should convert them before training our ML model.

## **Step 2: Descriptive Statistical Analysis**



* We can see that the minimum tip amount is negative (-1.29) this does not make any comprehension.
* We can observe that the minimum fare is negative (-52.00). This is clearly an error, as in real world this is not possible.
* The maximum pickup longitude is 0.00 which is not possible for New York City.
* We have some outliers since some traveling points are outside of the New York city's geographic coordinates.
* The minimum passenger count is 0, which again is an anomaly.
* We can see also that Extra has negative minimum value that does not have meaning.

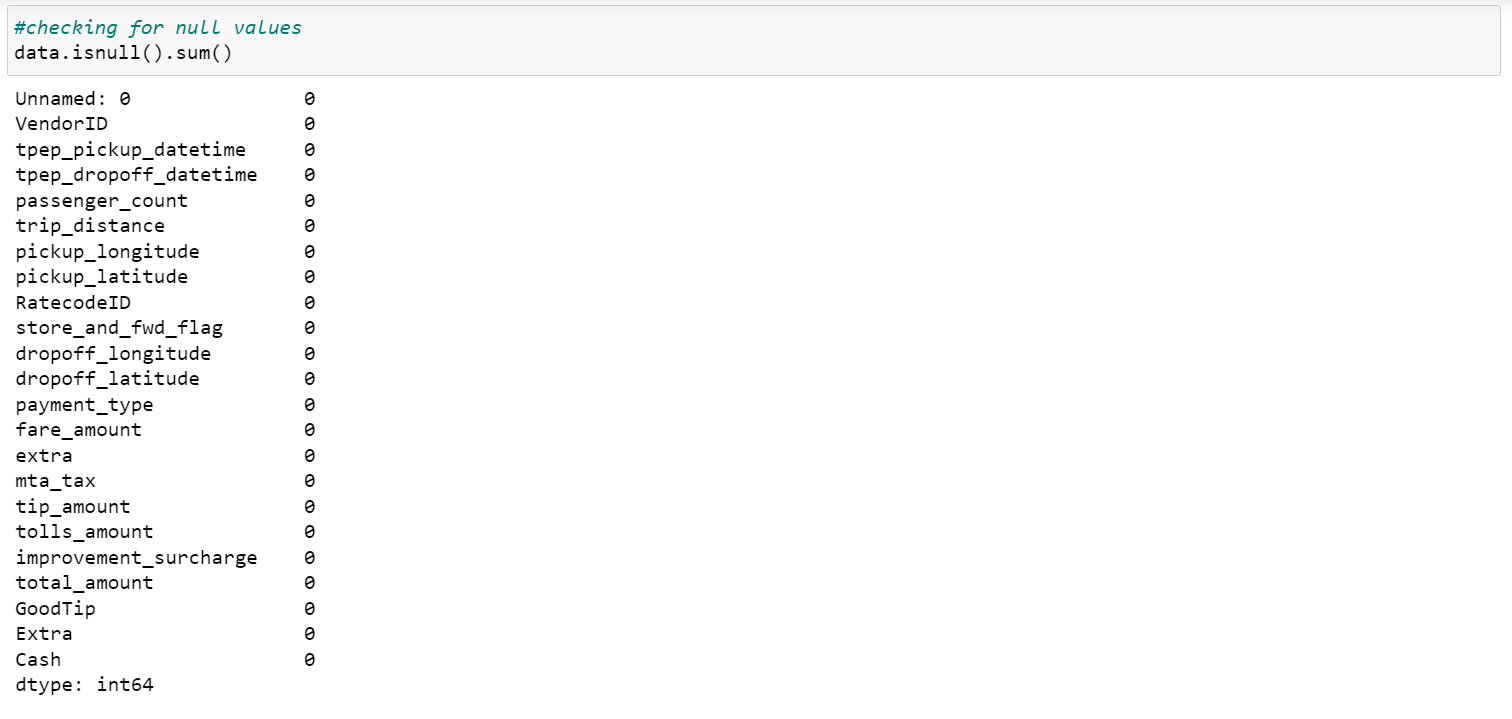
**Step 3: Getting Information about dataset**



Using this function, we will get Information about our dataset:

* Range Index that shows the number of the observations.
* Names of the variables and their respective data types
* The exact number of data type for each data type
* The space that uses in memory.

## **Step 4: Handling Null Values**



* As we can see there aren’t any Null values present for any of the variables of our dataset, so we don’t have to worry about data imputation/removing.

## **Step 5: Handling Duplicates**



* We observe that there is not any duplicate value in our dataset.

Now that we have better understanding of our dataset, we can proceed with further analysis.

## **Feature Engineering**

## **We are going to address the issues discovered during the descriptive statistical analysis, by changing the features.**

## **Step 1: Removing negative values– Tip Amount**

## Graphical user interface, text, application, Word Description automatically generated

* Tip amount values that are negative do not have any comprehend meaning so we removed them.
* Tip amount values that are zero indicate cases when the driver did not receive any tip.

## **Step 2: Setting limits – Fare Amount**

Graphical user interface, text, application

Description automatically generated

## **There should not be any negative fare amount values, we keep the datapoints which correspond to the range of fare amount.**

* As the New York Taxicab association in 2016 had a base fare of $ 2.5 according to their rules, we are going to set the lower limit of the fare as $2.5, and we are going to assume that any fare which is more than $250 is an outlier.

**Step 3: Setting limits for feature – Passenger Count**

Graphical user interface, text, application

Description automatically generated

* As, the minimum passenger count cannot be zero, we are going to drop all those datapoints where passenger count is zero.
* The maximum is six passengers, however that makes sense for New York taxi cabs, taken into consideration the New York rules, taxi policy and the taxis size allows to fit 6 people physically inside the taxi vehicle, so we do not have to remove it or set an upper limit.

**Step 4: Setting limits for feature – Trip Distance**

Graphical user interface, text, application

Description automatically generated

* **Trip Distance zero does not have meaning since the fare amount is not zero and the payment indicates that is done, so we removed zero values on this feature.**

**Step 5: Scaling the values for feature – Payment type**

Graphical user interface, text, application

Description automatically generated

* **As you can see the values are right- skewed**
* **We reduced the number of values and the scale on that column by using the map function.**

**Step 6: Converting feature – Store\_and\_fwd\_flag**

Graphical user interface, text, application

Description automatically generated

* **As you can see the values are textual Boolean, so we decided to map text values with numerical ones.**

**Step 7: Creating a new feature – Trip duration**

* **In this task, we will create a new feature called trip duration.**
* **It will be calculated from the difference between the drop off time and the pickup time and I express the value in minutes unit as shown below:**

## 

* **The next step we have decided to do is to create trip durations classes and intervals based on the freshly created column.**

## 

* We will use the get dummies function to convert trip duration categorical variable into dummy/indicator variables as part of the One Hot Encoding. The aim of this transformation is to convert categorical values into a 1-dimensional numerical vector.

Graphical user interface, text, application, email

Description automatically generated

* We have converted the trip duration into its 5 elements one-hot encoded vector now represented as 5 columns each with the categorical values and passed to the get dummies function and see if they have a connection with tip amount variable later.

**Step 8: New York City – Geographic coordinates**

## Text Description automatically generated

## **We know that New York City lies within the set of latitudes and longitudes, thus we can use that information to reveal outliers in our dataset.**

* We build a function which detects all those points which are outside boundaries of the city. New York City lies between the coordinates [-75, -72, 40, 42]

**Step 9: Removing outliers**



* Here we find the outliers - geolocation that are lied outside the New York geographic coordinates and show them below.

Table

Description automatically generated

## **As it can be observed we have 321 datapoints which are lying outside the New York City area. Hence, they can be considered as outliers and dropped.**

**Step 10: Calculating Distance**

Using the start point and end point we can understand displacement distance**:**

1. **Haversine Method**
2. **Chebyshev Method**

**These features can support the existing distance feature in the dataset, while training the ML models.**

Table

Description automatically generated

**Step 11: Converting the Datetime Feature**

Graphical user interface, text, application

Description automatically generated

* **We are going to extract the hour, minutes, seconds, day of week and month, as this might give us insights on variation of time with the target variable tip amount.**

**Step 12: Label Encoding features**



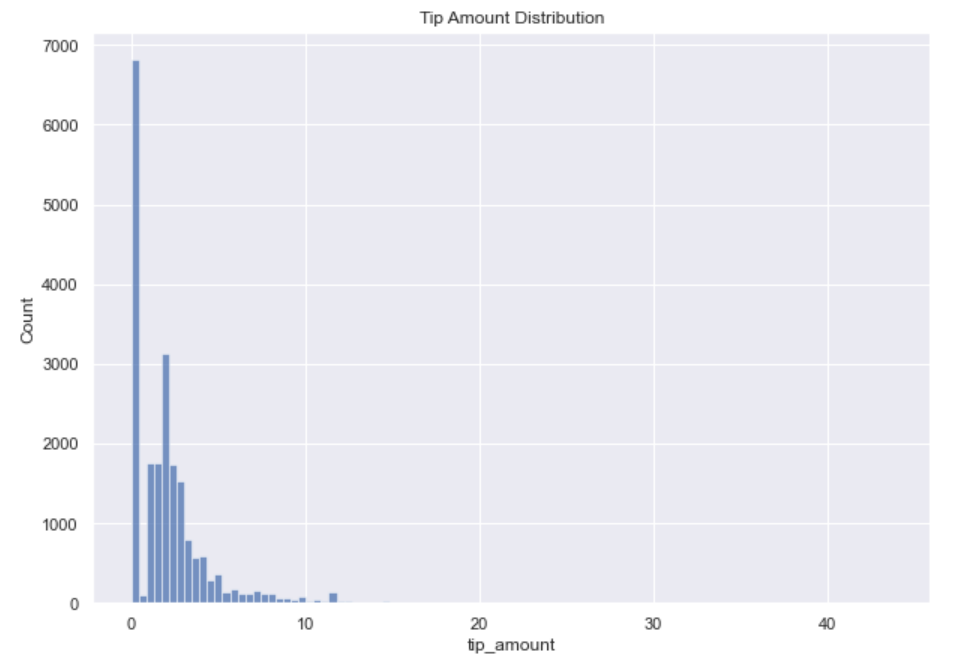
* **As we observed earlier that some of the features in our dataset have data type object that means non-int/float data type, we need to convert these categorical variables to numerical variables using label encoding. Each class will be assigned a unique integer.**

## **Data Exploration**

Through Data Exploration we will be trying to answer the following questions:

1. What does the distribution of tip amount look like? What is the range of the distribution? Is the distribution skewed?
   * If the distribution is Normal/Poisson/Power Law- then we can use their properties to perform further analysis of the target variable.
2. What is the passenger count for each trip?
   * As this might have a direct relationship with the tip amount and can be very useful for training the ML models.
3. What is the correlation between different features of the dataset?
   * As we might experiment with ML models like Linear Regression, which primarily assumes that all features are independent from each other, we need to ensure that we don’t keep redundant features in our dataset.
   * Correlation Matrix also helps us analyze the positive and negative (1, -1) correlation of the independent features with the dependent features. Hence, we can analyze with correlation of features with our target feature – tip amount.
4. Is there a variation of tip amount with time?
   * We will be able to establish if the target variable - tip amount is time dependent.
5. Does the tip amount depend on pickup and drop off location?
   * By analyzing the drop off and pick up location, we will be able to identify outliers. (Any pickup/drop off location which exists outside the bound of New York city can be removed)
   * We can also discover the areas where there is a high demand for taxis.

**Step 1: Plotting the Distribution of Target Variable**



* Taken into the consideration cases when we received tips we have a Right-skewed distribution.
* As we can see the number of the cases when we did not receive tip is high almost 7000.
* This distribution also resembles the power law distribution, which is very common in real world.

**Step 2: Tip amount, mean & median**

Graphical user interface, text, application

Description automatically generated

* **A** Most of the tip amounts are on the band between 1 and 3.
* We can see that some tips have maximum value around 44.

**Step 3: Plotting the Distribution of feature - Fare Amount**

Chart, histogram

Description automatically generated

* We can observe a right skewed distribution and determine that most of the taxi fares range from $ 2.5 to $ 20.
* The average taxi fee varies between (8-10) dollars.
* There are some peaks observed at 52, that are representing fixed fees.

**Step 4: Plotting distribution of Distance travelled (Haversine)**

Chart, histogram

Description automatically generated

* We can observe that most of the passengers travel on average 1-5 km.
* The average travelling distance is around 2km.
* This distribution looks very similar to the distribution of the fare amount. Thus, we can assume that, both fare amount and distance have overlapping distributions.

**Step 5: Payment type distribution**

Chart, histogram

Description automatically generated

* As we can see, most of the payment has been done through the Credit card as represented by first bar chart.

**Step 6: Frequency of Passenger in rides**

A picture containing chart

Description automatically generated

* Most of the passengers are traveling alone.
* We have more than 2500 cases that 2-passengers are travelling together.
* The reset is pretty small compared with these two cases.

**Step 7: Tip Amount vs. Passenger Count**

Chart

Description automatically generated

* The tip amount in 1 passenger rides is higher compared to more than 1 passenger ride.

**Step 8: Number of Rides vs. Hour of day**

Chart, histogram

Description automatically generated

* The number of rides at 5 am in the morning are the least. The number of rides remains constant for the remaining hours of the day. This signifies the rush hour timings when people are going to and returning from work.

**Step 9: Tip Amount vs. Hour of day**

Chart, scatter chart

Description automatically generated

* We can observe that the tip amount for different hours of the day remains almost uniform, hence we don’t fetch any meaningful insight.

**Step 10: Tip Amount vs. Day, Week and Month**

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

* We can observe that the smallest tip amount is on Saturday and the highest on Thursday.
* We can also observe that the fare amount is the highest between the 3rd and 4th week of the month.
* The smaller the demand, the higher the tip amount and vice versa.
* The peak of the tip amount is on the 25th day of the month.

**Step 11: Trip distance vs Pickup hours of the day**

Chart, line chart

Description automatically generated

* **A** The relatively long-distance-trip of taxi takes place during 3am — 7 am.
* The relatively short-distance trip time range is 9 am-11am and 6pm-7pm which makes sense because that is the usual work and daily operation time.

**Step 12: Trip duration distribution**

Chart, bar chart

Description automatically generated

* There are considerable cases when trip lasts longer than 5 minutes.
* The trip that lasts between 5-10 minutes has the highest frequency.
* The trips that last less than 5 minutes happen less compared with others.

**Step 13: Tip amount vs Trip duration**

Graphical user interface

Description automatically generated

* As we can see the tip amount is the highest when we have long duration of the trip.
* The Tip is proportionally depended on the trip duration.

**Step 14: Passenger count vs Day, Week and Month**

Chart, bar chart, histogram

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

Chart, bar chart

Description automatically generated

* **We can see that during the 6-7 am and 2-3 pm we have less passengers compared with other hours.**
* **On the 7th, 13th, 14th, 18th and 27th days of month we have more passengers.**
* **During the weekends we have more passengers than other days of week.**

**Step 15: Correlation Matrix (Pearson’s Correlation)**

Timeline

Description automatically generated with medium confidenceA picture containing graphical user interface

Description automatically generated

* We can observe that our target variable “tip\_amount” has high positive correlation with features like total\_amount, GoodTip, Have\_Distance, fare\_amount.
* We can also observe a considerable negative correlation with payment\_type.

**Step 16: Bivariate analysis - Tip vs Distance (Haversine)**

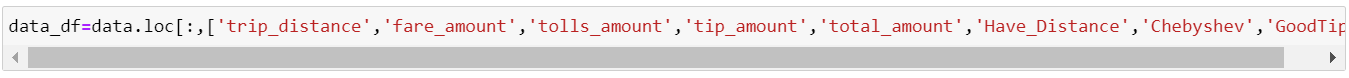
Chart, scatter chart

Description automatically generated

* We can observe a linear relationship between the tip amount and the distance travelled. The distance feature is going to be very helpful for ML models like Linear Regression as we can visualize the line which can intersect all the points when these two dimensions are chosen.

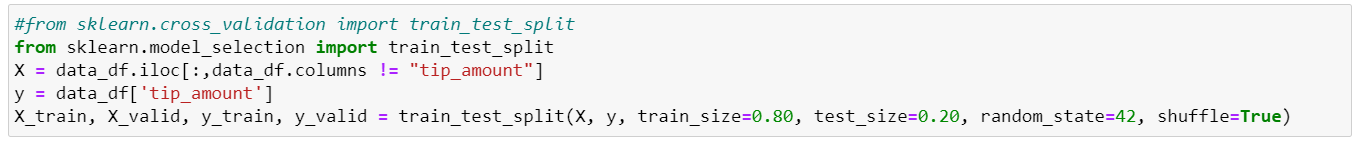
## **Modeling**

**Step 1: Sub-dataset**



* **A** creating a sub-dataset with the elected features and containing the target as well.

**Step 2: Train-Test Split**



* We are splitting the data into two parts – Train and Test
* The split ratio is 80% train data and 20 % test data. This is because, the dataset is inherently showing the behavior of being clean. Therefore, we can test the ML models on smaller sample size.

**Model 1: Linear Regression – Baseline Model**

Graphical user interface, text, application, email

Description automatically generated

* We can observe that the root means square error for both train and validation datasets is similar.
* Also, there isn’t a significant difference deviation between the RMSE of train and validation dataset, therefore we conclude there isn’t an overfitting problem.

**Model 2: Decision Tree Regressor (Tree based Model)**

Graphical user interface, text, application, email

Description automatically generated

* We can observe that the root mean square error for decision tree model is 0.568629, which is higher than the RMSE of Linear Regression.

**Model 3: Extra Gradient Boosting Tree (Boosting Model)**

Text

Description automatically generated with medium confidence

* We can observe that the final RMSE for XGboost model is 0.51857 at the 550th iteration. This model is significantly performing better than Decision Tree but not as good as Linear Regression.

Feature Importance - Linear Regression

Text

Description automatically generated

* We can observe that linear regression is assigning high negative coefficients to features like fare\_amount, tolls\_amount.

Feature Importance – XGBoost

Chart, histogram

Description automatically generated

* We can observe that XGBoost is finding features like trip\_distance, fare\_amount and total\_amount more useful to find the target variable.

**Model Evaluation - Performance Comparison**

|  |  |  |
| --- | --- | --- |
| **Model Name** | **RMSE** | **R ^2** |
| Linear Regression | 0.42 | 0.97 |
| Decision Tree | 0.568629 | 0.95 |
| XGBoost | 0.51857 | 0.96 |

# Conclusion

We can clearly observe that a relatively simple model like Linear Regression is outperforming more complex models like XGBoost and Decision Tree. This means, that the features in the dataset have linear relationship with the target variable. We can use this predictive model to set a baseline for building more dynamic tip amount models, which depends on even more number of factors like holidays, music events, neighborhoods

etc.

References

1. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
2. https://towardsdatascience.com/how-to-clean-your-data-in-python-8f178638b98d
3. https://www.tutorialspoint.com/python\_data\_science/python\_data\_cleansing.htm
4. https://realpython.com/python-data-cleaning-numpy-pandas/
5. https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html
6. https://www.kaggle.com/datasets/microize/newyork-yellow-taxi-trip-data-2020-2019
7. https://en.wikipedia.org/wiki/Haversine\_formula
8. https://www.w3schools.com/datascience/ds\_linear\_regression.asp