Missing value and outlier handling

New Folder - Week 1

Introduction: missing value

There are different types of missing values. The type that we worked on this project alot was MCAR(missing completely at random). In this case, there is no relationship between the missing data and any other values observed or unobserved within the given dataset. That is, missing values are completely independent of other data. There is no pattern.

Deleting rows before 2023

Our goal is to predict demand for future months of 2023. But there are some rows from 2022 and even 2008 in our dataset. These are only noises and won't be useful for 2023 prediction. So we delete them and only keep rows that start or finish in 2023.

imputing passenger_count with median of passengers in each pickup zone and each 10m timestamp

Also passenger count is MCAR. Because it didn't have correlation with any other variable.

Testing after imputing each missing value

We calculate summary statistics such as mean, median, standard deviation, and variance for the original data and the imputed data, and compare them to ensure that they are similar. Large differences in these statistics could indicate bias in the imputed data.

Before imputing:

count 2.995023e+06
mean 1.362532e+00
std 8.961200e-01
min 0.000000e+00
25% 1.000000e+00
50% 1.000000e+00
75% 1.000000e+00
max 9.000000e+00

After imputing:

count 3.049679e+06
mean 1.330062e+00
std 9.088246e-01
min 0.000000e+00
25% 1.000000e+00
50% 1.000000e+00

75% 1.000000e+00
max 9.000000e+00

imputing less important variables by its mode

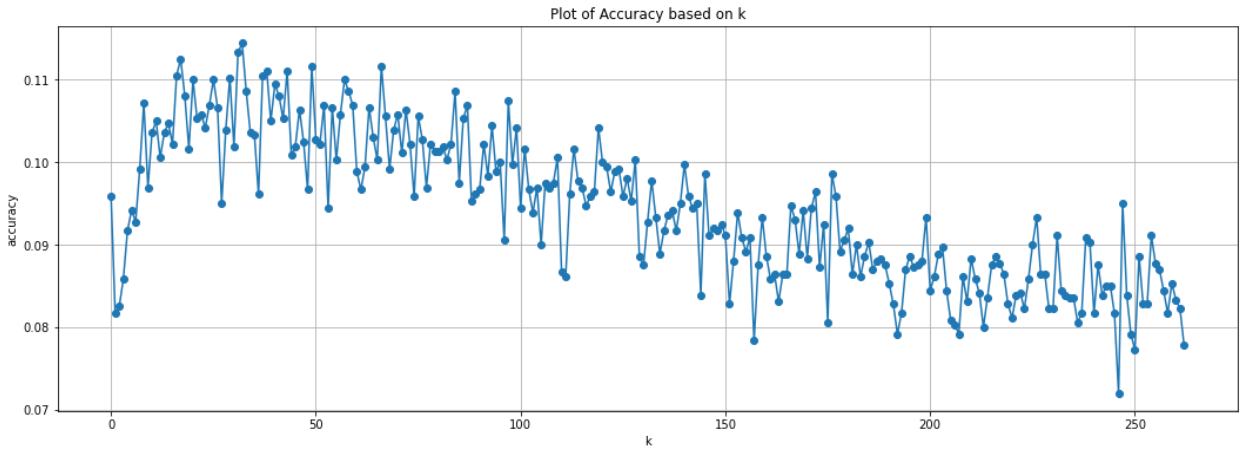
“Store_and_fwd_flag” and “RatecodeID” were not so effective to define demand. So we imputed them with their median and didn’t bother about them so much.

Imputing Location ID for Unknown Locations

There exists trips in our dataset, where either the pick-up or drop-off locations are unknown. Consequently, these instances lead to the loss of trip-related data. Acquiring knowledge of these locations would be beneficial, as it would facilitate the imputation of other variables, such as trip distance. By ascertaining the trip distance, we can precisely determine the fare amount, aiding in the computation of the total cost. Since we want to impute a categorical variable there are 3 ways that we could achieve it:

- ➔ Finding the nearest data based on the features and replacing the current data with the one that we are closest to. This can be done by using an KNN algorithm.For time series data, it is imperative to handle missing values separately to ensure that our model does not prematurely learn test set features. Since the specific dataset structure and the splitting amount have not been determined by the team, the code has been implemented as a function-based approach to accommodate future requirements. So the imputer function computes based on all three types (Median , Mode , Nearest data) and then chooses the one with the highest accuracy as our approach.

    - ◆ First we will only choose a specific features that seem to help us in imputing the unknown ones, and we choose the 'PULocationID' , 'DOLocationID' , 'trip_distance' , 'total_amount' , 'Duration_total_Seconds' columns and we trained it on rows which have no missing or unknown values.
    - ◆ For that we first need to encode our categorical features and scale the numerical ones. Once we are done with that we need to create a train and test dataset and then train the model on them.
    - ◆ Also we can try different values for K to find the best solution here (here the best one was 32). Since we wish to find a data with the nearest features to our current data point , classifying the whole data based on a K that is too big can cause overfitting, since we do not have much data from some of the zones and based on our experiments shown in figure below 32 seems to give the best accuracy with this method.
    - ◆ Once done we could use the trained KNN model to find the neighbors for each row of the data with unknown location id and then perform median or mode on them and replace the unknown values.



➔ If we wish to find location ids for drop off, unknown ids we could group our trips based on the pick up location ids and then perform mode or median on them so that we could replace the unknown ones and the same can be done for pick up location ids by grouping the drop of data. Although, In comparison to the previous method with the k having the value of 32 this method performs worse. The accuracy for mode is 7 percent and for median is 2 percent.

One thing that needs to be taken into consideration is that the missing variables at the PULocationID and DOLocationID are MCAR (Missing Completely At Random) MCAR means Missing Completely at Random(MCAR).

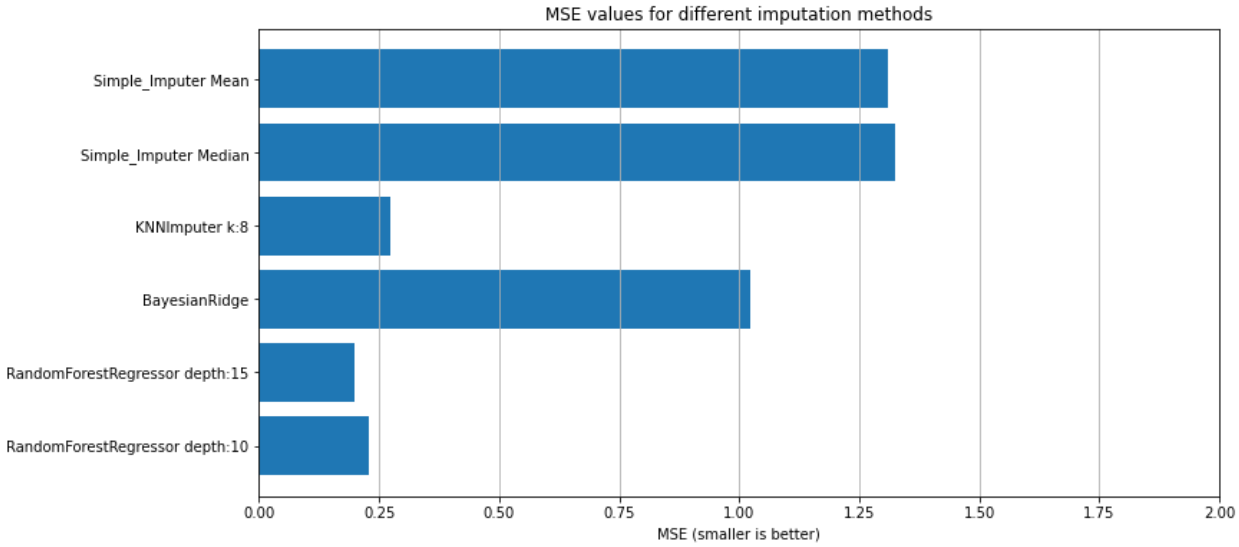
$$P(Missing \mid Complete\ data) = p(Missing).$$

Imputing The Numerical Columns

'Trip_distance' is a column within our dataset that may contain certain instances of zero values. We have access to data pertaining to the pick-up location ID and the drop-off location ID, and we can confirm that the trip duration exceeded two minutes. The trip distance is a critical parameter for computing the 'fare_amount', as the database description indicates that the fare amount is calculated based on both time and distance.

Our approach involves employing an Iterative Imputer to estimate the missing values for trip distance. In order to do that we have tested a variety of different imputers as shown in the figure below, by using a part of data and replacing it's values with nan values randomly and then compute mean squared error between the new imputed values and the old ones and as it is demonstrated in this figure the random forest regressor had the best result although it was the

most time and space consuming method .Subsequently, we will utilize the time and distance information to compute the 'fare_amount' for those rows that currently hold a value of zero.



Upon completion of the above steps, we will have the 'total_amount' column available in our dataset. This column represents the cumulative expenses incurred during each trip. We can then calculate the values for rows with a total amount of zero and proceed to remove the other detailed expense-related columns from the dataset. This elimination ensures independence among features and prevents any correlations between different variables.

Introduction: outlier

an Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.

Method 1 : Using Isolation Forest to identify passenger_count outliers and using Quantile based flooring and capping to impute outliers

In an Isolation Forest, randomly sub-sampled data is processed in a tree structure based on randomly selected features. The samples that travel deeper into the tree are less likely to be anomalies as they require more cuts to isolate them. Similarly, the samples which end up in shorter branches indicate anomalies as it was easier for the tree to separate them from other observations.
Isolation forest gives 0 and any number higher than 4 as outliers.

Outliers = [0, 5, 6, 7, 8, 9]
So we get [1,4] as our range for passenger_count and by using flooring and capping, we impute outliers with these values.

For testing outlier handling, we calculated CV(coefficient of variation) before outlier handling:

```
Coefficient of variation: 65.3917442531385
```

And after outlier handling:

```
Coefficient of variation: 54.944117507278946
```

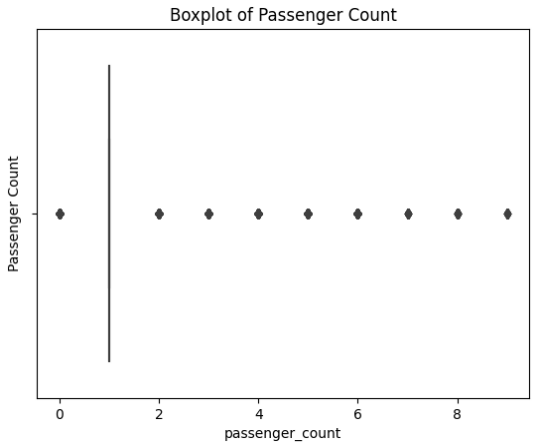
So variation of passenger_coount decreased after outlier handling.

Method 2 : Using Z-score to identify outliers and Quantile based flooring and capping to impute outliers

Z-score identifies any number higher than 4 as outliers.
Outliers = [5, 6, 7, 8, 9]

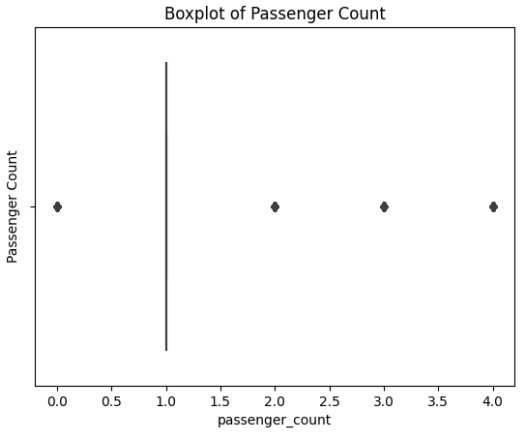
So we get [0,4] as our range for passenger_count and by using flooring and capping, we impute outliers with these values.
before outlier handling:

```
Coefficient of variation: 65.3917442531385
```



And after outlier handling:

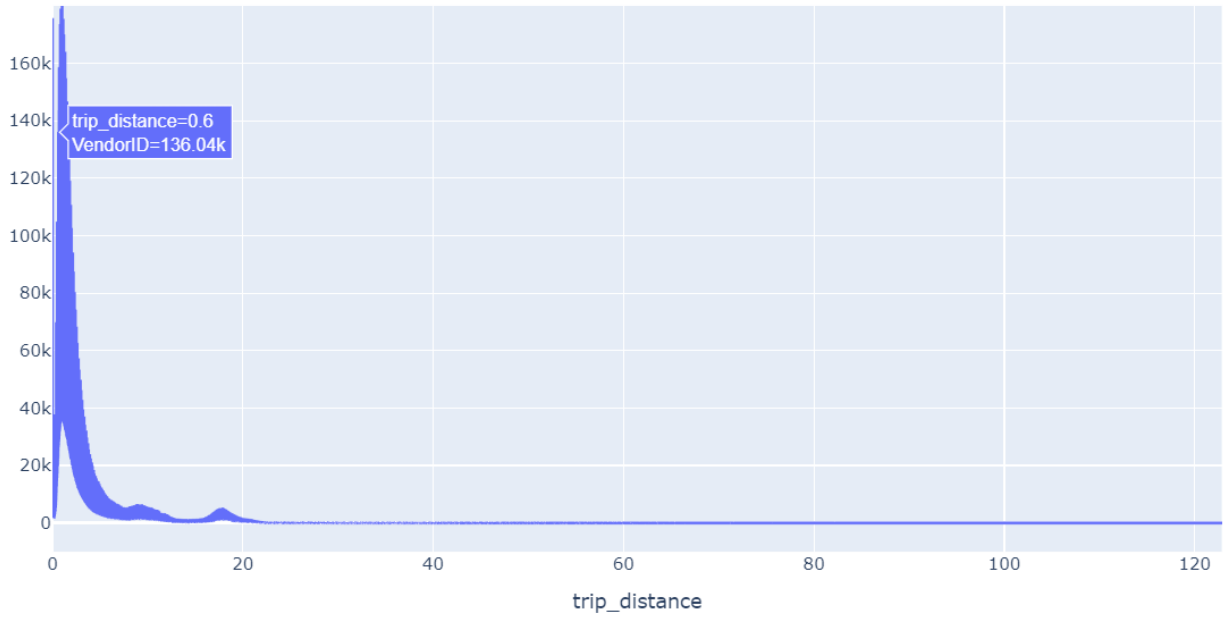
```
Coefficient of variation: 57.06712340977993
```



As we can see, isolation forest reduces variation more than z-score.

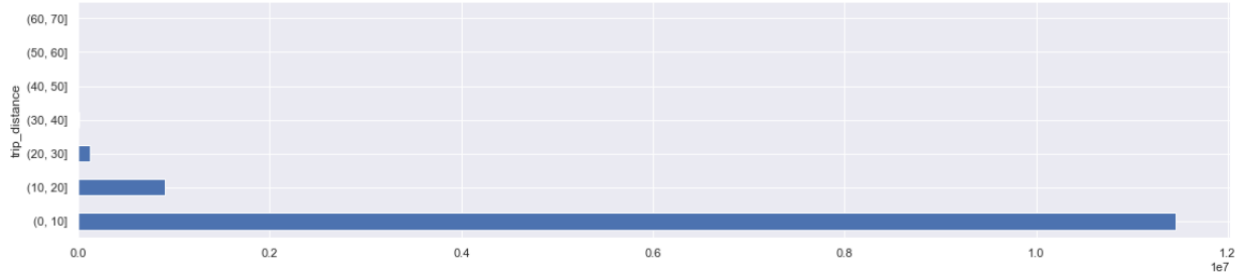
Imputing Too Long Trips

In our dataset, we have observed trips with extremely long moving distances. There have been many methods proposed on outlier detection. The simplest way to detect outliers is by drawing box plots. It is easy to observe the distribution of the data you're analyzing. The box demonstrates the central 50% of the data, with a middle line showing the median value. The lines extending from the box capture the range of the remaining data. Any data point that falls outside the lines indicates an outlier.
Let's now look at the distribution of the distance across the different types of rides:




```
q1 =  1.07
q2 =  1.8
q3 =  3.38
IQR =  2.3099999999999996
Upper bound =  6.844999999999999
```

There are some trips with over 60 miles distance while the mean distance traveled is approx 1.8 miles.



From the above observation, it is evident that most of the rides are completed between 1-10 miles with some of the rides with distances between 10-30 miles. Other buckets are not visible because the number of trips is very less as compared to these buckets.

The total area of the five boroughs that make up New York City is about **320 sq miles**; the longest distance between its boundaries, from the northeast to the southwest, is about **35 miles**.

According to the distribution of trip distances and the fact that it takes about 30 miles to drive across the whole of New York City, we decided to use 30 as the number to split the trips into short or long-distance trips.

```
Short Trips: 12661842 records in total.
Long Trips: 10895 records in total.
```

We considered short trips as possible trips and replaced the distance trip with long ones.
We calculated the mean of trip distances between each pick-up and drop-off zone and chose this value as the new trip_distance.

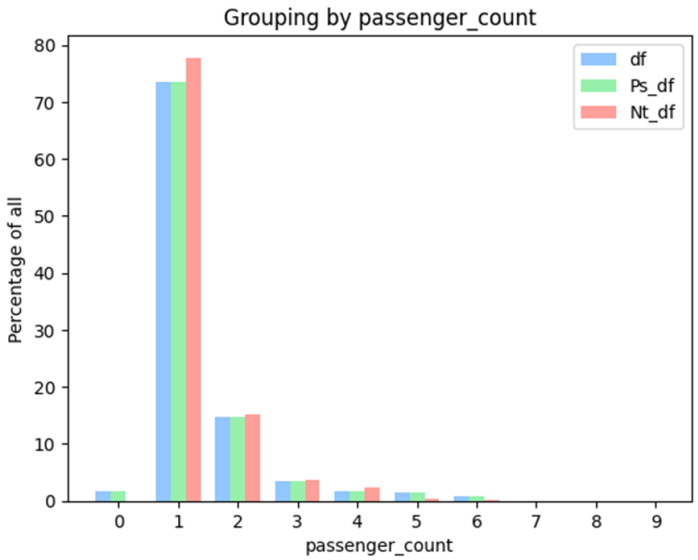
Imputing Negative total amount

By checking the total amount feature, you can see that some values from this column are negative. Considering that the cost of a trip cannot be negative, these values are identified as outliers and must be managed.

After examining the data, it can be found that 100,816 rows of data have such a feature. In the first step, we need to check whether we can find a certain pattern only among the negative data and find out the reason for the existence of negativity or not.

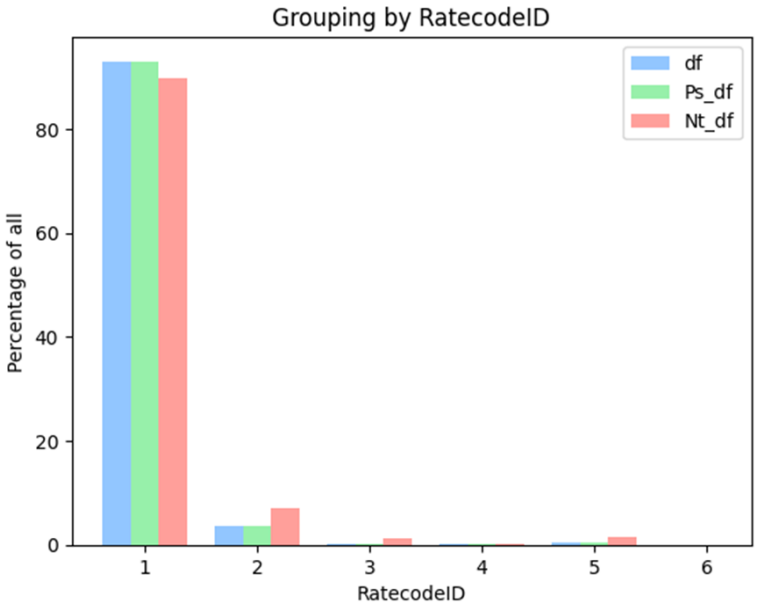
Therefore, in the first step, the amount of negative data is grouped based on each categorical feature, and the contribution of each of the possible values is checked in three cases of all values, the dataset that includes positive total amount values, and the total amount that includes negative values. It should be noted that this value is drawn in the graphs based on the percentage of the total data.

The first feature is passenger_count, which normally can have values between 0 and 9.

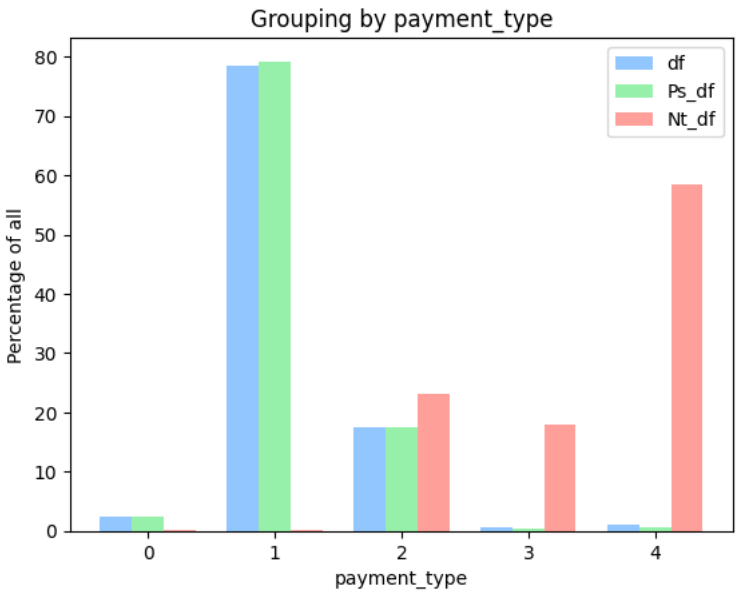


As it can be seen that Passenger_count does not significantly differ from each other in three different modes and no specific result can be obtained from this graph.

In the RatecodeID feature, the number of data with a negative total amount in the case where RatecodeID = 2 is twice that of other datasets. This is in case no such pattern is observed in other possible values for RatecodeID.



In the payment_type attribute, the significant issue is that 58% of trips encountered with payment_type = 4 have negative values. This is a significant amount.

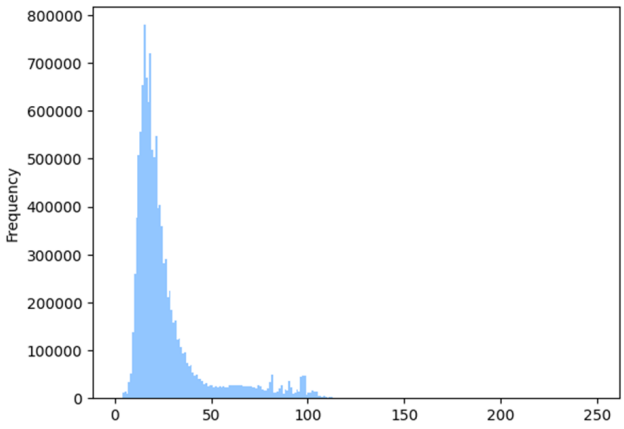


Also, this problem can be seen in the case of payment_type = 3. Therefore, according to the distribution of the data, it can be seen that the negative total_amount has a high correlation with the type of payment. so that nearly 60% of the negative data have payment_type = 4

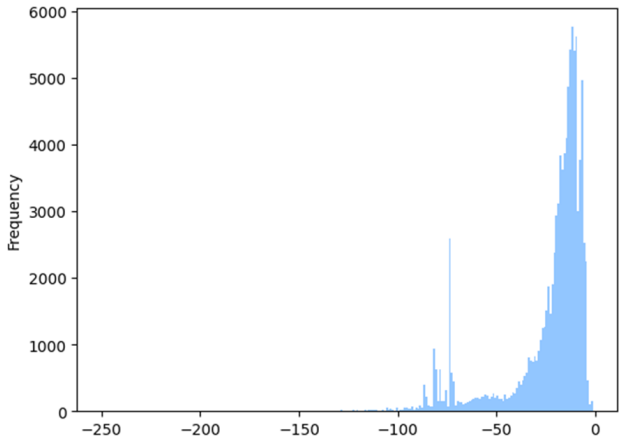
Also, another point is that sometimes, despite the negative value of total_amount, the value of tip_amount is positive and has been paid. This issue can indicate problems in the payment system or other factors. It is noteworthy that in such a situation the cost of the trip has been calculated correctly, but only the payment has faced problems. This problem may be due to the payment system or the passenger's lack of cooperation.

To test this hypothesis, the distribution of trips can be checked between positive and negative data. If the negative data has the same distribution as the positive data, it can be said with a good approximation that the cost of the trips were calculated correctly and the payment was not made for unknown reasons.

In the figure below, you can see the histogram related to the total amount for data with positive values:



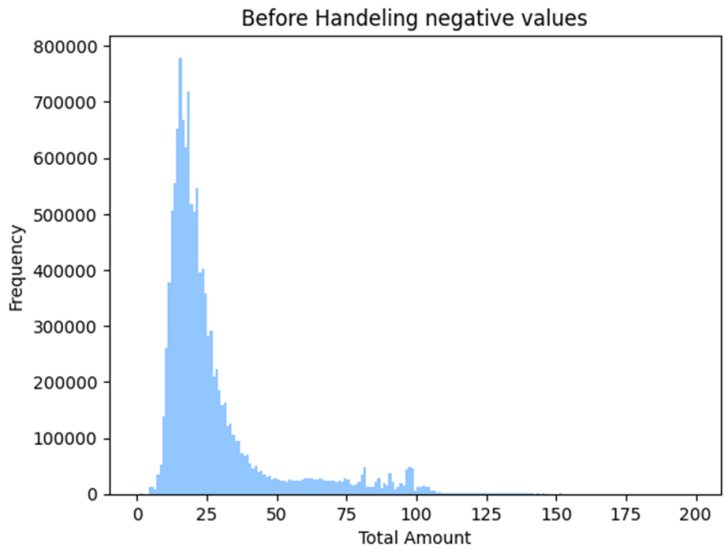
Also, in the figure below, the histogram of the total amount for data with negative values can be seen:

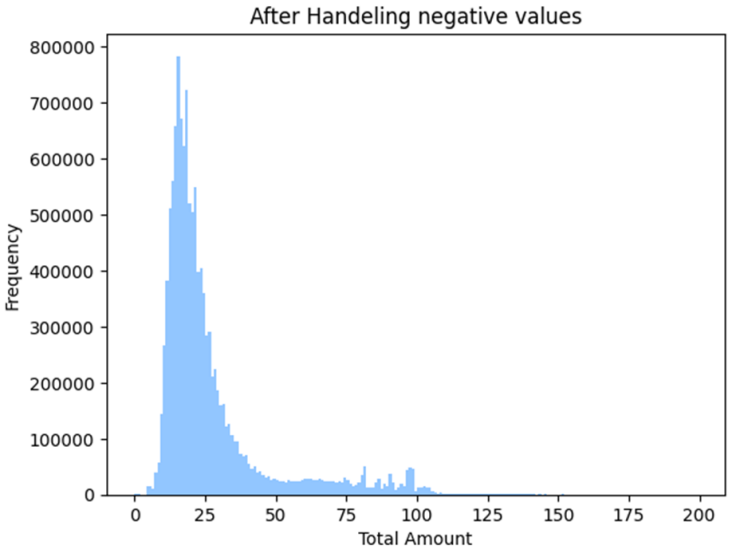


As it is intuitively clear, the data distribution is the same. Therefore, it can be considered with a high approximation that there is no reason for the assumed assumption to be incorrect.

For this reason and due to the small number of data, it is suggested to replace the outlier values that include negative data with their absolute values. Of course, in trips where the tip amount is positive, it should be kept in mind that this amount will be added to the absolute value of the negative amount.

After this review, a comparison can be made between the initial conditions of the dataset and the conditions after the change. According to the two graphs below, it can be concluded that the handling of outlier values here has not changed the overall distribution of the data and it corresponds to them.





Summary

One of the important stages of data pre-processing for machine learning models is the management of outliers and missing values. In these activities, we sought to handle outliers and check missing values to make the learning process for the machine learning model easier and prevent possible errors.

In these activities, we sought to identify outlier and missing data and handle them somehow by using exploratory data analysis, descriptive statistics, intuitive examination and more advanced methods such as Z-Score, Isolation Forest, etc.

Secondary investigations showed that the handling of Outlier and Missing data generally did not affect the general trend of the data and only provided more learning samples for the model.