Shahid Beheshti University

Anita Soroush

98222085

assignment 2 of Machine Learning

Dataset 1: [mobile-price-classification]

*The necessary cleaning and **preprocessing** steps were taken in the **previous assignment**.*

question 1.
Forward selection

Before running the selection algorithm, we are asked to merge the 2 lower price classes into one class and the two higher ones into another class so the problem will be converted into a **binary classification** one and the processing will be easier. So I simply relabeled the target column:

0, 1 ⇒ 0
2, 3 ⇒ 1

Now moving on to the forward feature selection, the algorithm that I implemented selected the 5 following features in the same order in which they are reported. the related AUC is also printed alongside the selected features:

1. ram 0.8941290191290191
2. battery_power 0.9204568579568579
3. px_height 0.966880341880342
4. px_width 0.9801841676841676
5. mobile_wt 0.9907407407407407

question 2.

Now let's make a logistic regression model using the 5 features chosen by the forward selector [ram, battery_power, px_height, px_width, mobile_wt].
The performance of this model can be measured using different metrics like:

- The **precision** is the ratio `tp / (tp + fp)` where `tp` is the number of true positives and `fp` the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0.
- The **recall** is the ratio `tp / (tp + fn)` where `tp` is the number of true positives and `fn` the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. The best value is 1 and the worst value is 0.
- The **F1 score** can be interpreted as a harmonic mean of precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:
  F1 = 2 * (precision * recall) / (precision + recall)

These values for the aforementioned model are as follows:
precision: 1.0
recall: 0.9814814814814815
F1_score: 0.9906542056074767

questions 3 & 4.

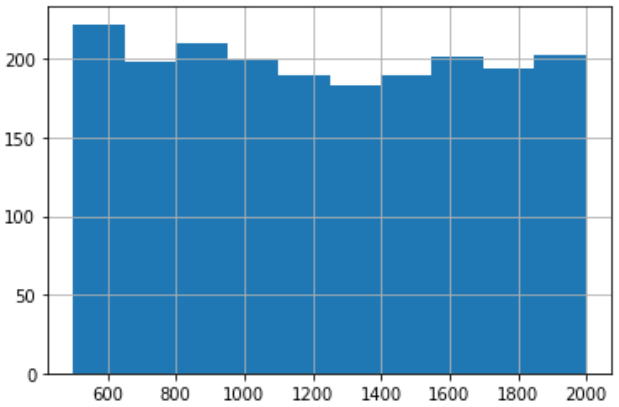
Now let's see the result of another logistic regression model in which the dataset is changed using a **PCA** algorithm with 5 components.

precision: 0.9844559585492227
recall: 1.0
F1_score: 0.9921671018276762

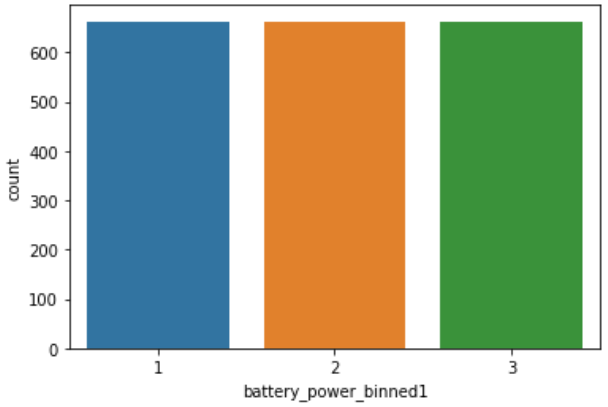
question **6 & 7**.

a. binning battery power:

We are asked to cut the battery power into 3 **unequal-sized** groups. The histogram below illustrates, this feature has a uniform distribution.

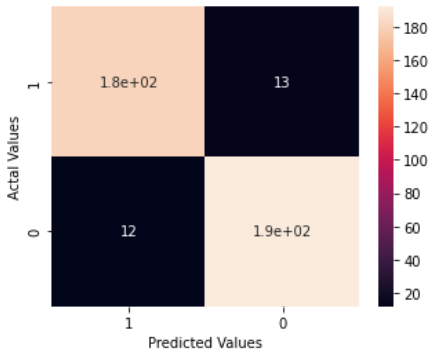


So if we want to have 3 **equal-sized** bins, we simply have to divide the whole interval into 3 equal-sized subintervals.

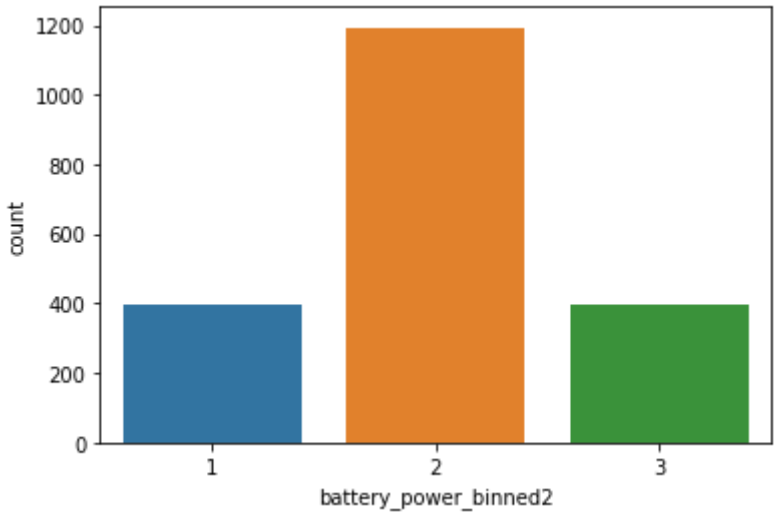


And how does SVM work on this?

Accuracy: 0.94

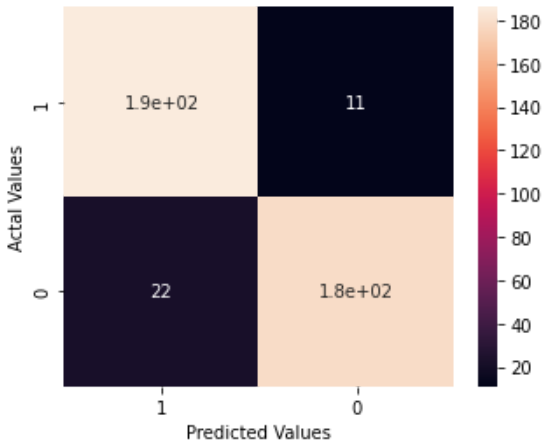


If we determine the sizes of the subintervals in an **unequal** way like:



Then the accuracy will decrease:

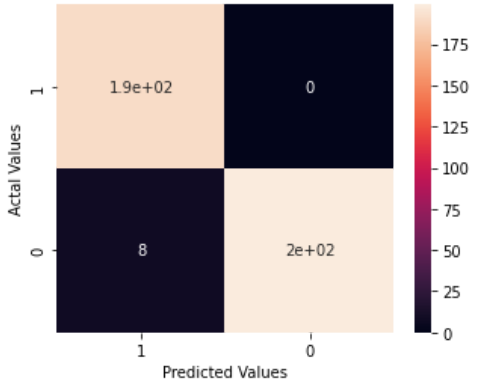
Accuracy: 0.92



b. One hot encoding:

While cleaning the dataset, I dropped some of the features from the raw data. In the current dataset, The Only categorical feature **which is not binary** is n_cores. After using one hot encoder, the SVM result is as follows:

Accuracy: 0.98



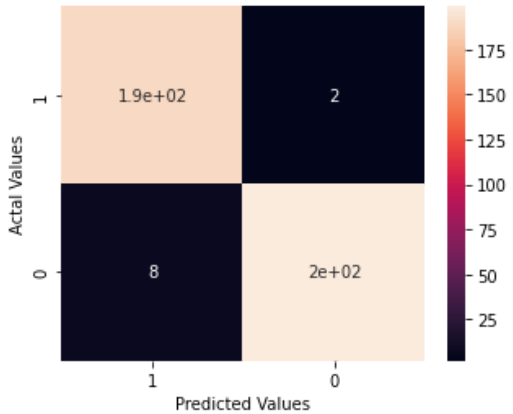
Some general reasons why we use one hot encoding are:

- Some classifiers can only use numerical values. So without one hot encoding we cannot use some categorical non-numeric features.
- One hot encoding makes our training data more useful and expressive, and it can be rescaled easily.
- By using numeric values, we more easily determine a probability for our values.

c. Area and Volume:

The SVM results:

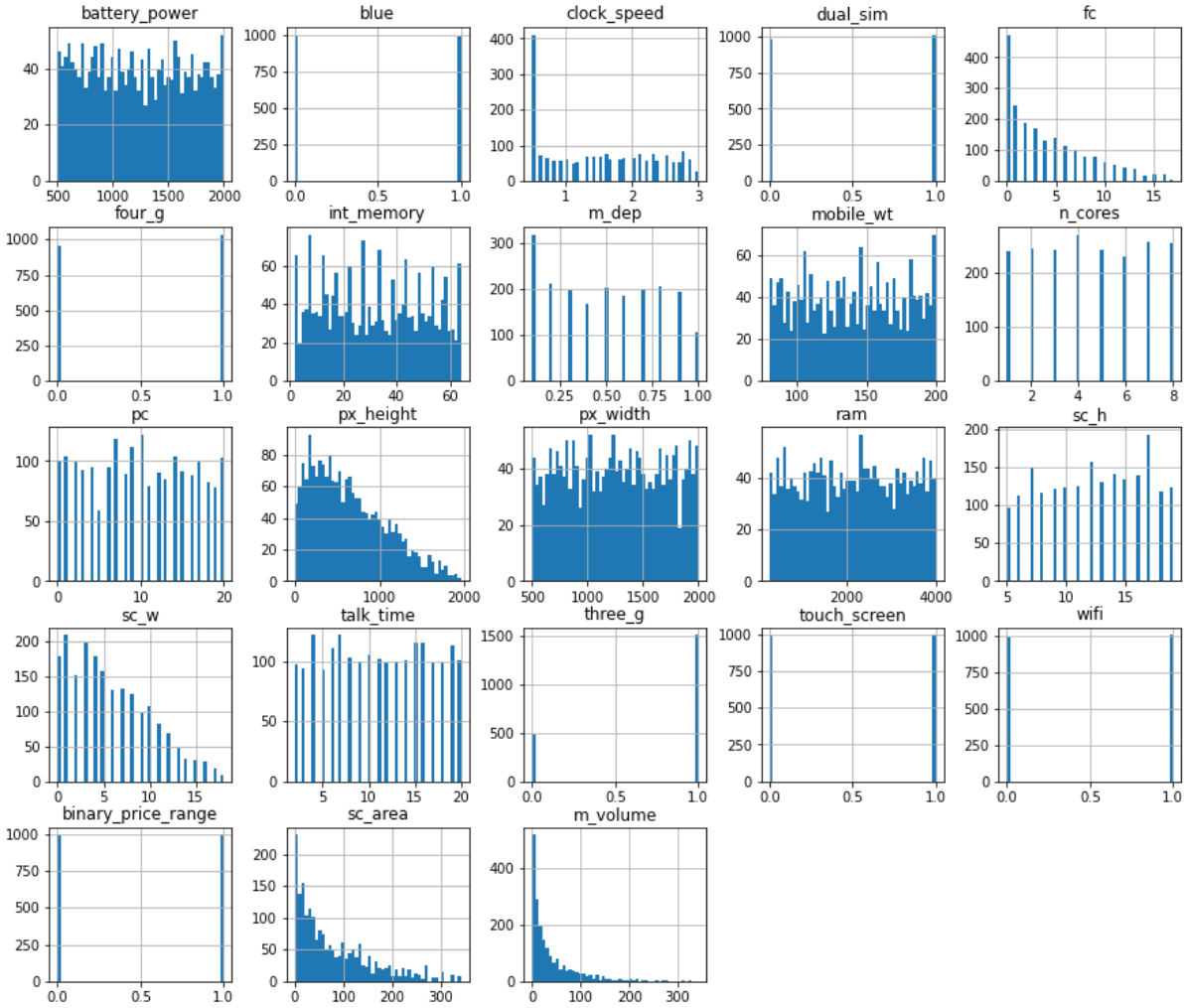
Accuracy: 0.97



d. Transformations:

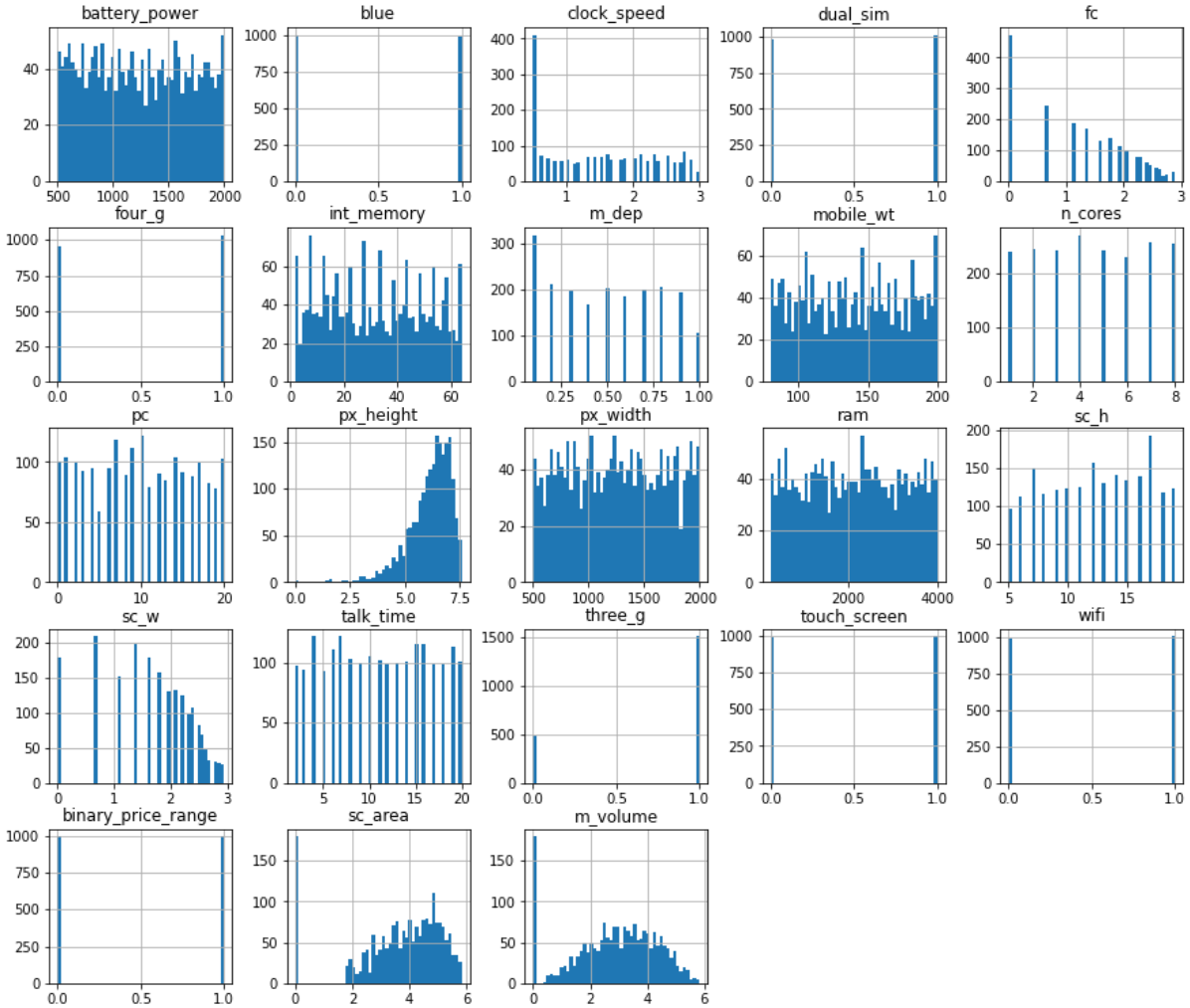
The Log Transform is one of the most popular Transformation techniques. It is primarily used to convert a skewed distribution to a normal distribution/less-skewed distribution. In this transform, we take the log of the values in a column and use these values as the column instead. But the point is that Log Transform can only be helpful when the data has a right-skewed distribution. A log transformation in a left-skewed distribution will tend to make it even more left skew, for the same reason it often makes a right skew one more symmetric. In a left-skewed distribution an exponential transformation will be helpful. Next to exponential and log transformation, there are also other techniques like square or cube roots transformations.

Moving on to our own dataset, the features histograms are as follows:



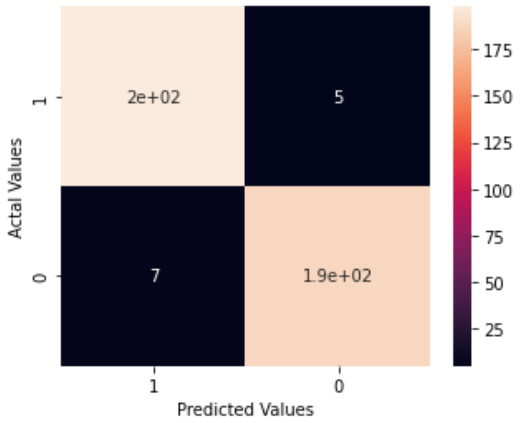
It is clearly seen that none of the features is left-skewed. so there is no need to use exponential transformation but log transformation can be helpful for **fc, px-height, sc_w, sc_area, m_volume.**

After applying log transformation for the aforementioned features the histograms have a more normal distributions:



And the SVC outcome is as follows:

Accuracy: 0.97



question **8**.

bootstrapping is a **resampling** technique that involves repeatedly drawing samples from our source data with replacement, often to **estimate a population parameter.** Cross validation splits the available dataset to create multiple datasets, and Bootstrapping method uses the original dataset to create multiple datasets after resampling with replacement. Bootstrapping Validation is a way to predict the fit of a model to a hypothetical testing set **when an explicit testing set is not available.**

question **9**.

As I mentioned in the previous question, Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure **has a single parameter called k** that refers to the **number of groups that a given data sample is to be split into.** For example In **2-fold cross-validation**, we randomly shuffle the dataset into two sets d_0 and d_1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). We then train on d_0 and validate on d_1, followed by training on d_1 and validating on d_0.
The expression **5*2 cross validation** refers to **5 iterations of 2-fold cross-validation.** Repeated k-fold cross-validation provides a way to **reduce the error** in the estimate of mean model performance.

question **10**.

In cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters, and **picking the elbow of the curve as the number of clusters to use.** The problem with clustering is that the more clusters we add, the more variance we explain because if we push the exercise to the extreme, we get as many clusters as points and the variance is fully explained and the bias is almost 0. We are then faced with a case of **overfitting.** To avoid overfitting, we generally use the so-called “elbow method”, which looks for a bend in the curve plotting the explained variance versus the number of clusters. However, **the elbow method doesn't always work well;** especially if the data is not very clustered and we see a fairly smooth curve, and it's unclear what is the best value of k to choose. In other words the elbow chart for the dataset **does not have a clear elbow.**

Dataset 2: apartment-rental-offers-in-germany

*The necessary cleaning and **preprocessing** steps were taken in the **previous assignment**.*

telekomUploadSpeed and serviceCharge are both numerical. But let's take a look at heatingType:

First of all, it is a categorical feature. So if we want to use Regression methods we should apply one hot encoder.

Not only this, the number of categories is 14 which is a relatively large number that can have a negative effect on the regressor performance.

The name of each category and its number of repetition is reported below:

central_heating	65368
NotAvailable	14599
district_heating	14562
gas_heating	10293
self_contained_central_heating	9042
floor_heating	7544
oil_heating	2507
heat_pump	1119
combined_heat_and_power_plant	967
night_storage_heater	689
wood_pellet_heating	459
electric_heating	371
stove_heating	147
solar_heating	95

I merged the gray categories into one category named "other":

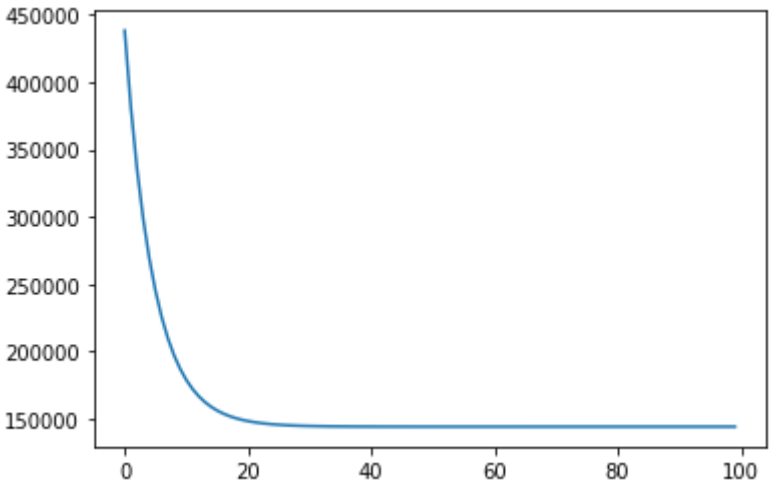
central_heating	65368
other	22940
NotAvailable	14599
district_heating	14562
gas_heating	10293

Now we can simply use one hot vector encoder and then use different regression methods to see the result.

question 1.
Implementation of Logistic Regression

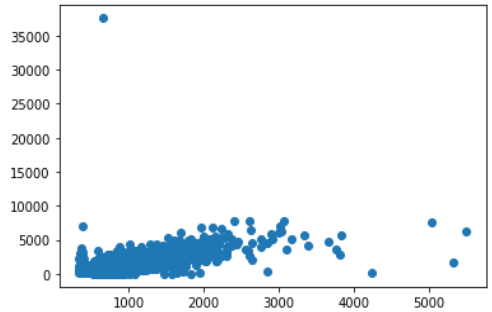
After implementation I initialized the parameters as follow:
w → initialized to zeros
b → initialized to zeros
learning rate → 0.1
epochs → 100

In the following plot, the x-axis is the epoch number and the y-axis is the cost value for the related epoch. It seems that after around 50 iterations the cost value doesn't change that much. So we better set the epoch parameter as 50 in order to prevent wasting of resources.



Now let's see the metrics:

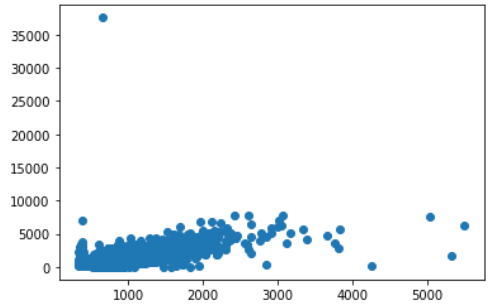
MAE: 260.7461396938489
MSE: 209840.88791046795
R2_score: 0.3847120494063474



question 2.

Using sklearn implementations:

MAE: 260.7036399979541
MSE: 209843.188996771
R2_score: 0.38470530224334387

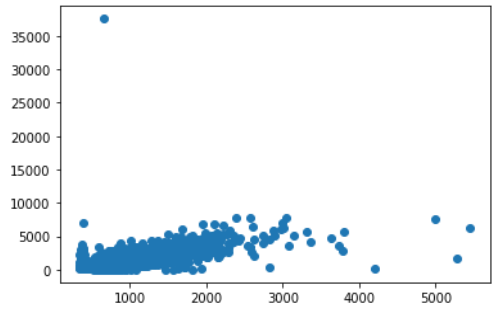


question 3.

I used the same features as the questions 1 and 2.

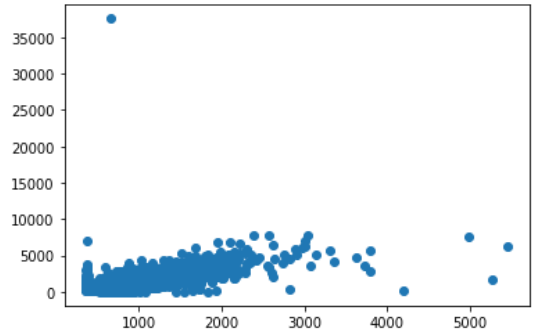
- ### Ridge Regression:

MAE: 261.39921076264426
MSE: 210514.85394568322
R2_score: 0.38273587028936684



- ### Lasso Regression:

MAE: 261.6697035216319
MSE: 210605.12698717648
R2_score: 0.3824711749039854



References:

- https://www.analyticsvidhya.com/blog/2021/04/forward-feature-selection-and-its-implementation/
- https://www.kdnuggets.com/2018/06/step-forward-feature-selection-python.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html
- https://discuss.analyticsvidhya.com/t/transformations-to-convert-left-and-right-skewed-distributions-into-normal/1463