Name-Vipul Anand Batch-DS2312

Project-Zomato Restaurant Machine Learning Project

1. Problem Statement-

In this dataset predict 2 things –

* 1. Average Cost for two
  2. Price range

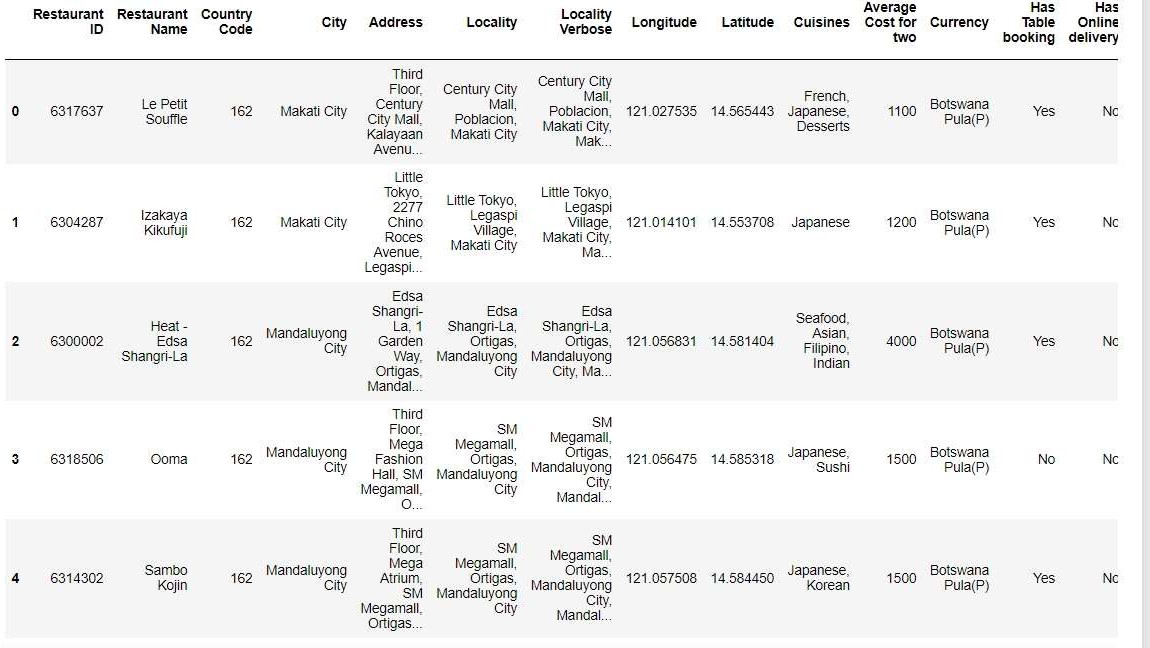
So in this project we have to predict the average cost for two which is a regression task and the other problem is predicting the price range which is a classiﬁcation task.

So let us proceed further . 2.Data Analysis:-

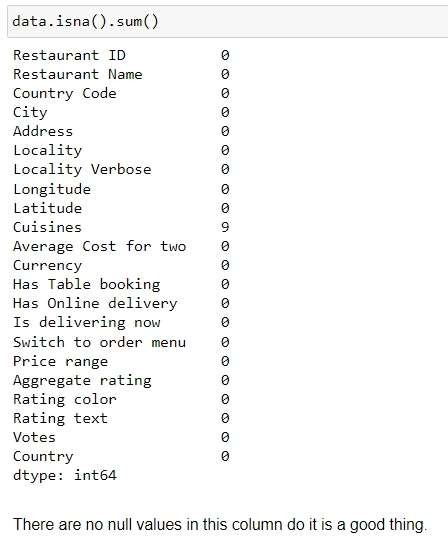
This dataset has 9551 rows and 21 columns. We have another dataset as country code.

So at ﬁrst we merged these two datasets on the common columns country code which is common in both of the datasets.

After merging the dataset looks like this-



This dataset has no duplicate values and has nulls in one column only. Let us see this in the picture.



Data analysis of various columns.

#Restaurant\_id- Altough the restaurant id has different Id for each row. Even then we did not deleted it and we will use this in our model building

#Restaurant\_name-

We replaced restaurant name with the value counts of each of the restaurant. Most of the restaurants were appearing only once. So in this way we represented each restaurant with a unique number without loosing too much of information.

#Country

Most of the restaurants were from india and many had 1,2 restaurants.

So we combined the restaurants with very less vaule counts together and later one-hot encoded them.

After the encoding the data looks like.



#City.

In this also we replaced the values with the value counts of the city. #Address

In the address also we replaced the address with value counts and later transformed them because the default dataset has very large number of outliers.

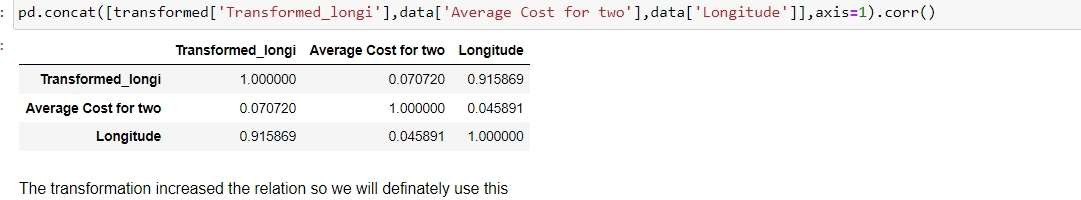
#Locality-

In the locality also we also replaced the values with the value counts of the locality columns and later transformed them with using the power transformer because this had also a large number of outilers.

#Locality Verbose- This was the same as the locality column so we deleted it because having two columns of the same type does not makes any sense for the model and it can result into an over ﬁtting.

#Longitude-

This is a numerical column but it had a large number of outliers and the skewness was also very high so we transformed the values using the power transformer and this also resulted into a incease in the correlation with the label also.



The transformation decreased the outliers and also increased the relation with the label so we will be using this transformed longitude for our model.

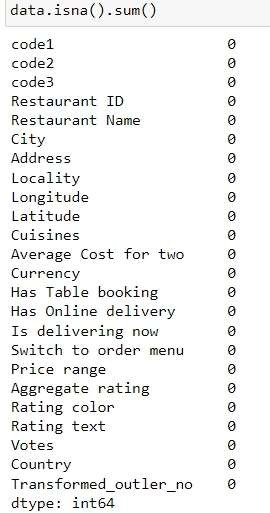
#Latitude-

For the latitude also the transformation increased the relation with the label and also reduced the outliers by a large number so in this case also we used the power transformed latitude column instead of the default latitude column.

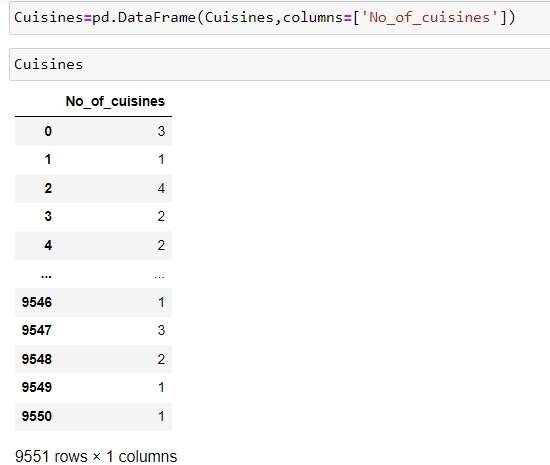
#Cuisines-

This is the only column in this dataset with nulls so we ﬁlled the nulls ﬁrst using the pandas ﬁllna method.

Now the nulls are ﬁlled.



For this column what we did is we used the number of cuisines a restaurant has because each restaurant has different types of cuisines.



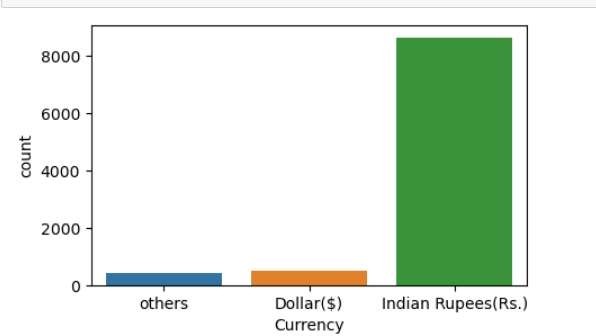
And later we one hot encoded the cuisines column on the basis of number of cuisinses a restautant has.

After encoding the dataset looks like this.

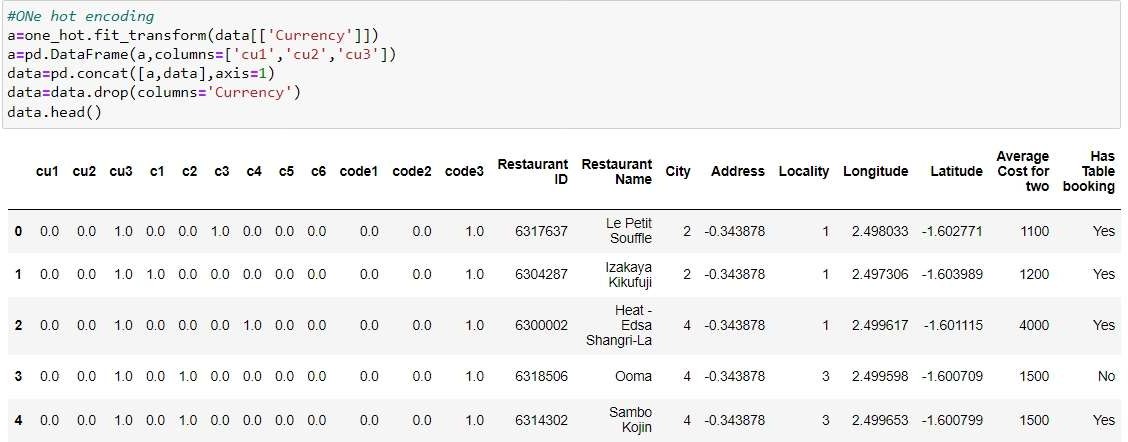


#Average cost for two- As this is the label for the dataset we will not be making any changes int his column at all.

#Currency- This column also has a very large imbalance so we grouped the currency with less value counts together .

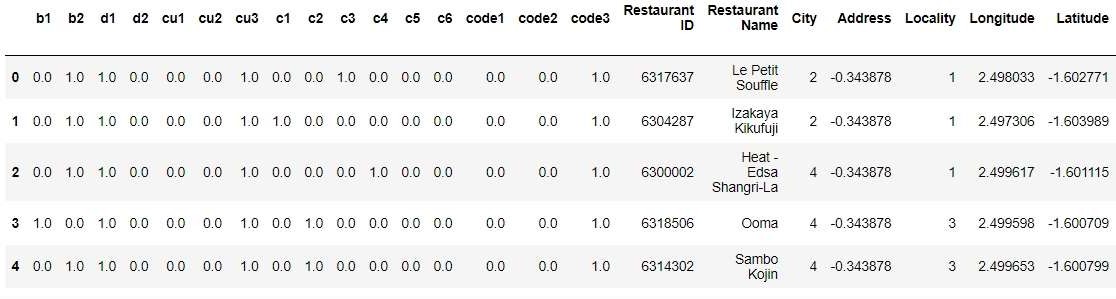


We one hot encoded this currency column also and this looks like this.



#Has online delivery and has online booking-

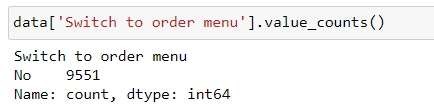
There are columns with Boolean values so simply we encoded them and this looks like this after the encoding



#Is delivering now-This is also a column with Boolean value and we simply encoded this using the one hot encoder.

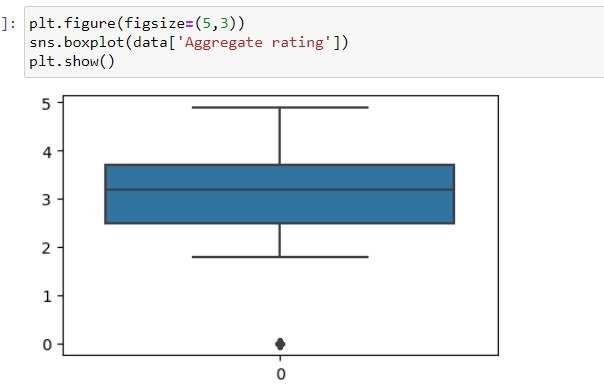
#Switch to order menu-

This columns has all rows same so we deleted this column.



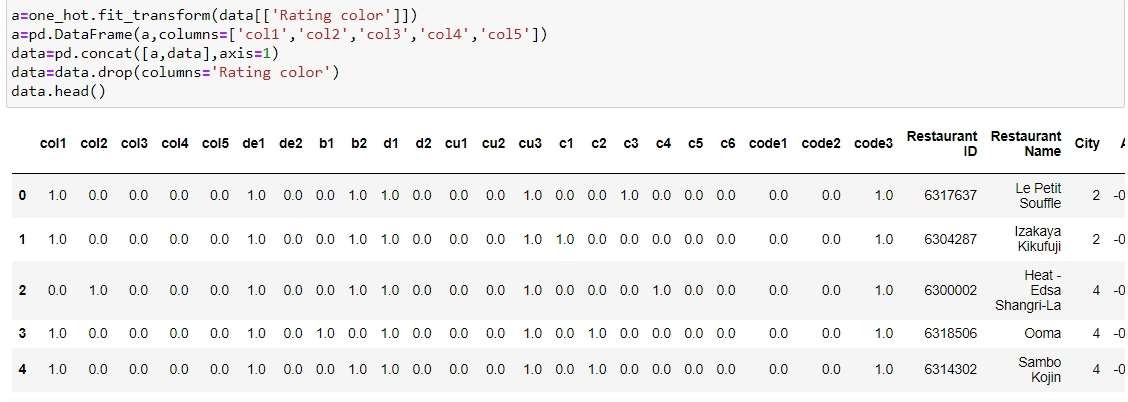
#Price Range-This is also one of our label so we made no changes here also. #Aggregate rating-

This is a continuous column with a little bit of outliers as per the box plot



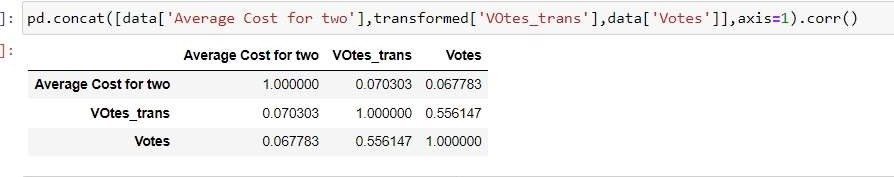
#Rating colour-

This is a categorical column with a bit of imbalance so what we did is we combined the classes with very little value counts together and later one hot encoded the column and after the encoding it looks like this

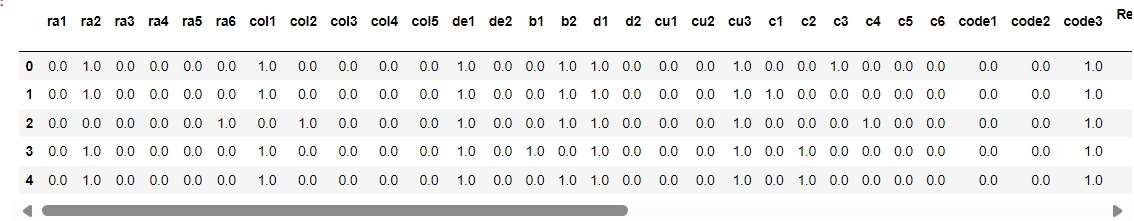


#Rating text- This is also a categorical column with some imbalance so we combined classes with less value counts together and later encoded them.

#Votes- This is continuous column but it has a very large number of outliers so we tried transforming it and this resulted in a less number of outliers along with a better relation with the label so we replaced the default column with the transformed column .

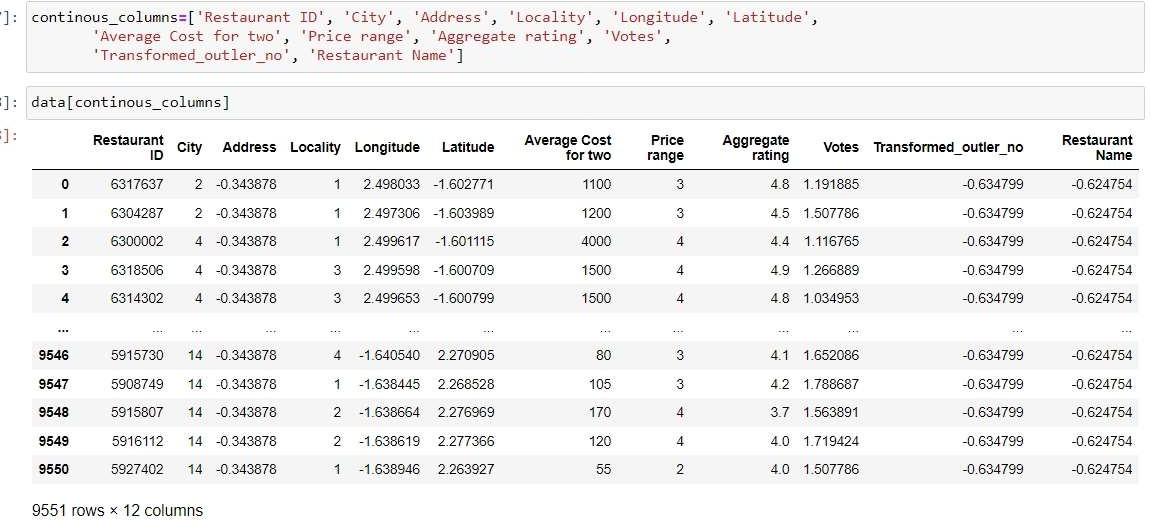


#So we have treated all the feature columns successfully and the ﬁnal dataset looks like this after all the transformation and the encodings.

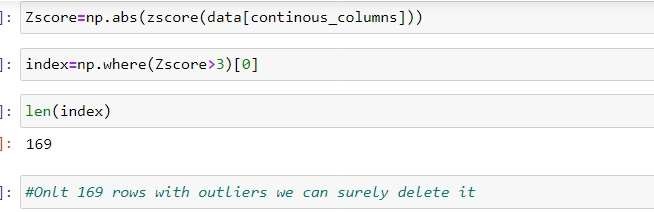


1. Preprocessing and outlier detection-

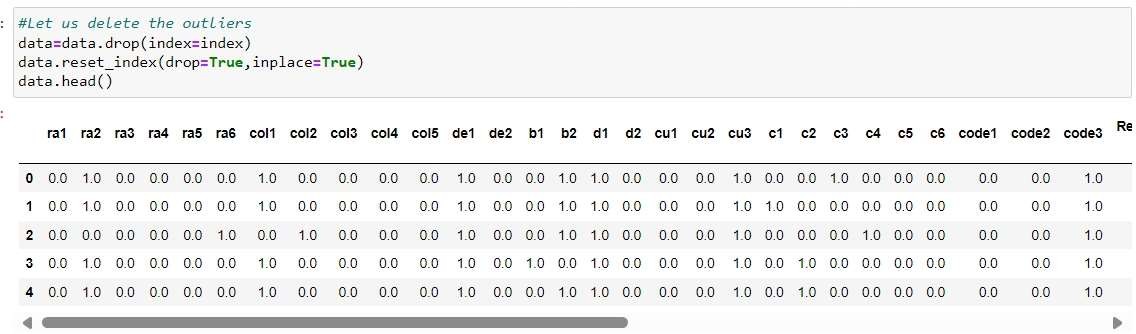
To delete the outliers we ﬁrst selected the continuous columns from our dataset because here we are only concerned about the continuous columns not categorical.



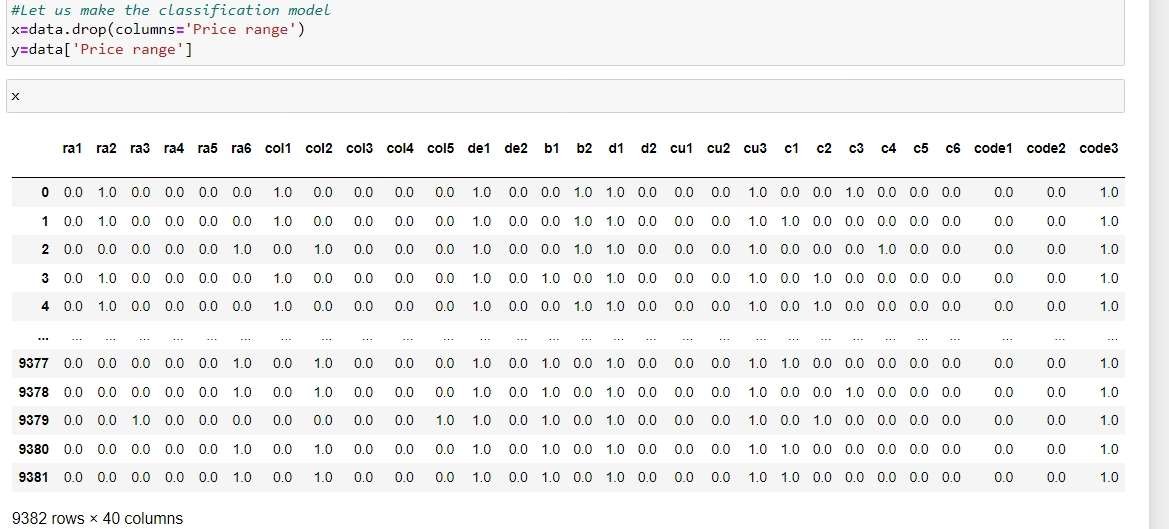
#Detecting and deleting the outliers.



Only 169 rows with the outliers have been detected so we will be deleting it. #After deleting the dataset looks like this

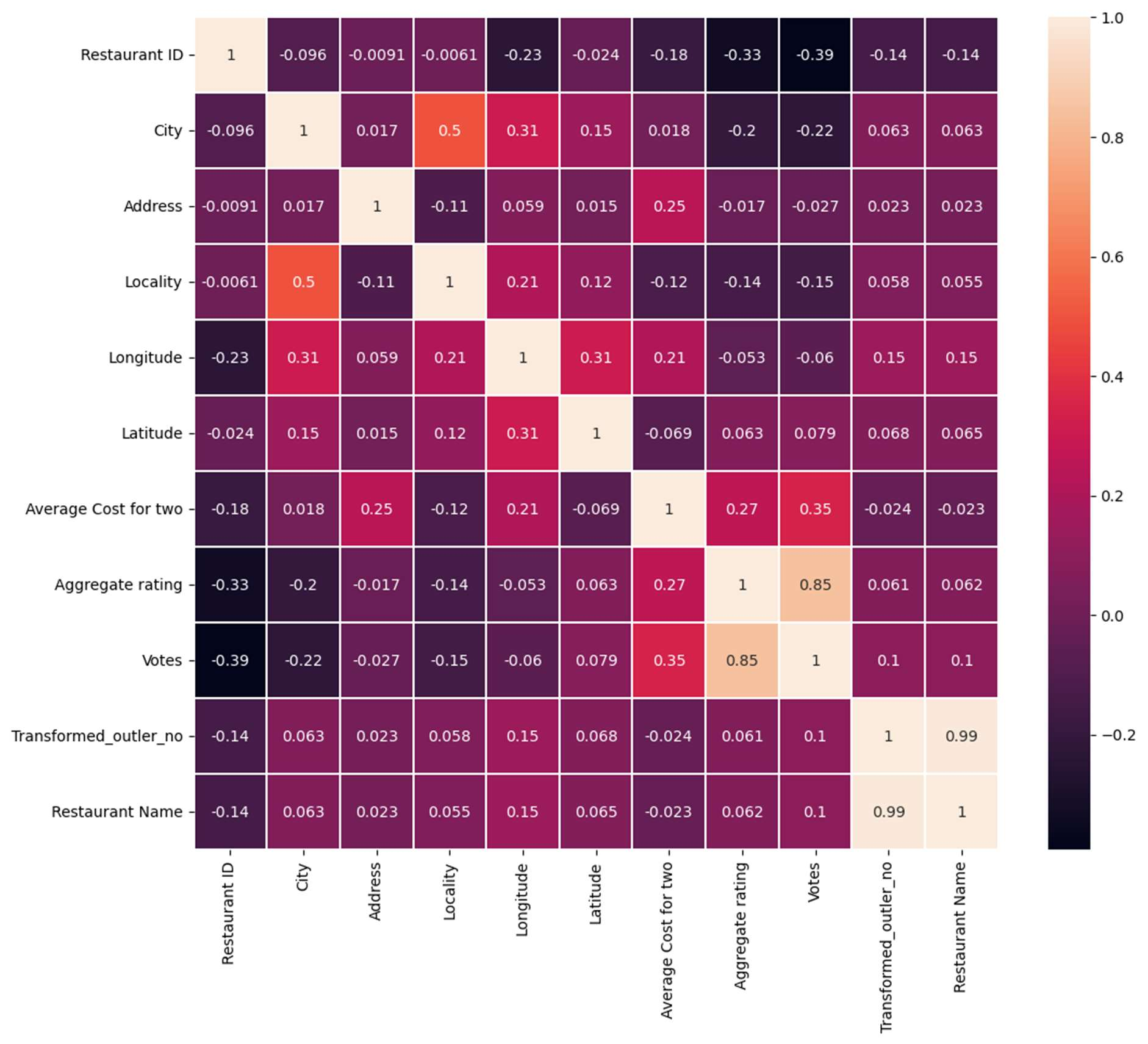


#At ﬁrst we will be making the classiﬁcation model we divide the features and labels according to this.



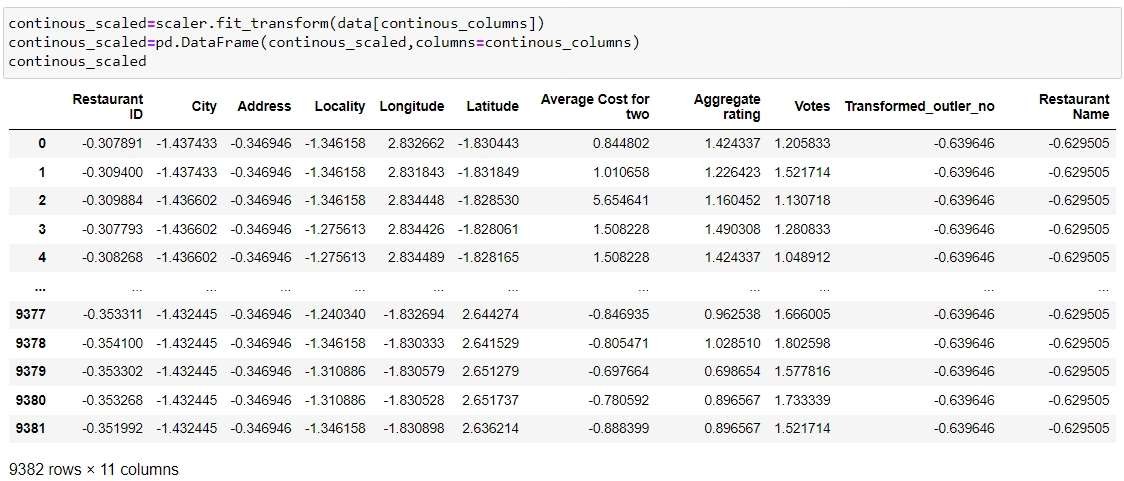
The average cost had very large number of outliers so what we did is we used the transformed columns for this also rather than the default one.

#Correlation of the continuous columns.

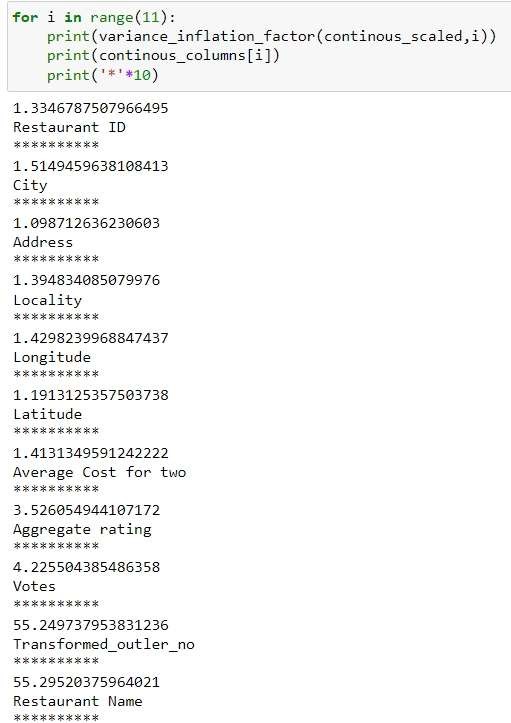


#After that we standardized the continuous columns and we did the analysis of the variance inﬂation factor score.

After the standardization the continuous columns looks like this.



#We saw the variance score of the continuous columns but some were having a high score.

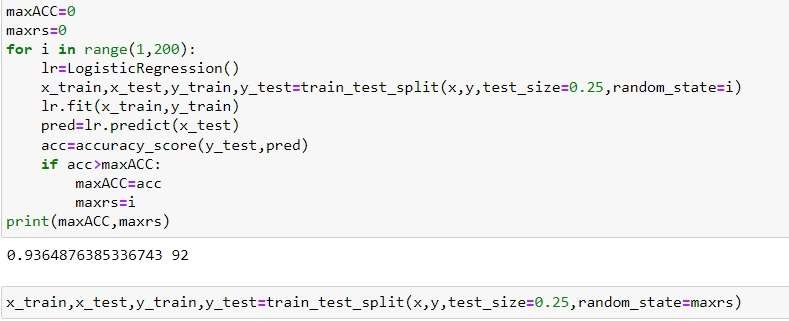


Altough some columns were having a high variance but we did not deleted them and made the ﬁrs model using all the columns.

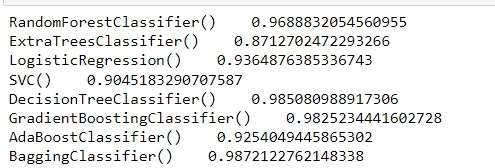
After this we joined the main continuous scaled columns to the main featue columns and the preprocessing is done for the classiﬁcation model we will make.

1. Model building for classiﬁcation task. #The best random state

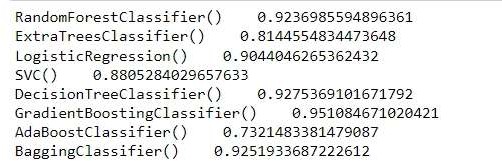
We used the logistic regression to know the best random state for the train test split.



We made the models using other algorithms also and some model had very high accuracy also. The accuracy of other model are below:-

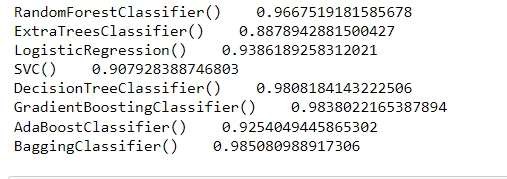


The accuracy after the cross validation is also below



The gradient boost is the best performing model till now.

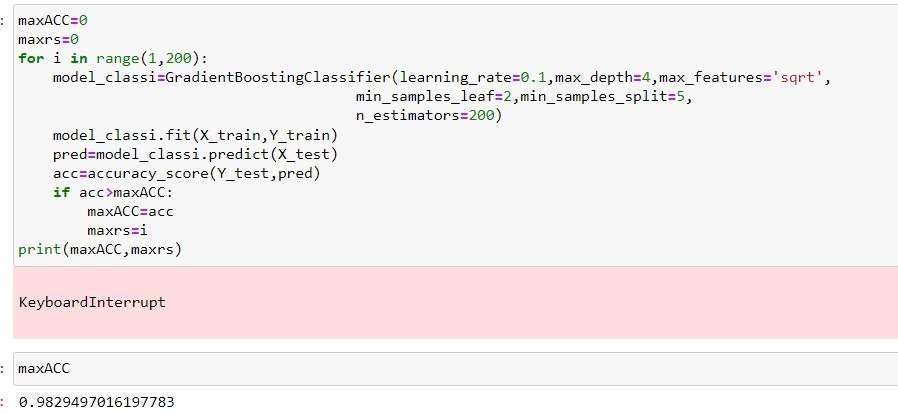
After that we deleted the restaurant id to se if the accuracy increases or decreases but it resulted In a decrease in the accuracy .



So we will be doing the tuning for the gradient boost and using all the feature columns because it had the best accuracy and the best accuracy after the cross validation also.

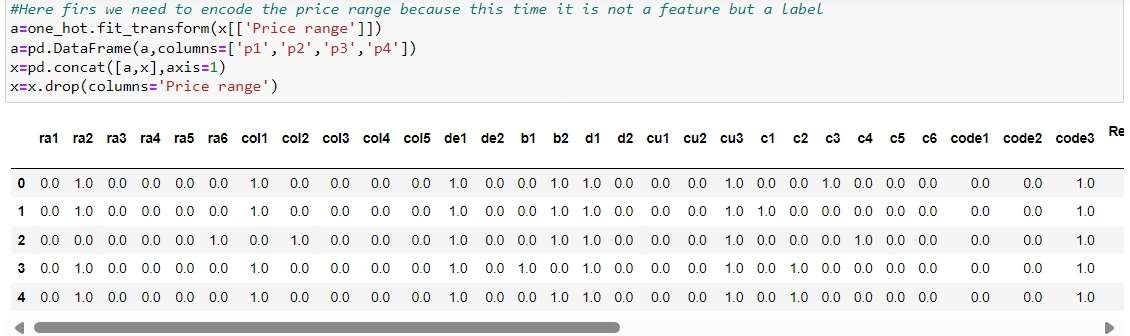
After the tuning the gradient boost had the accuracy of 98.

We also used different random state to know the best random state at which it has the best accuracy and after this the gradient boost had an accuracy of 98.2

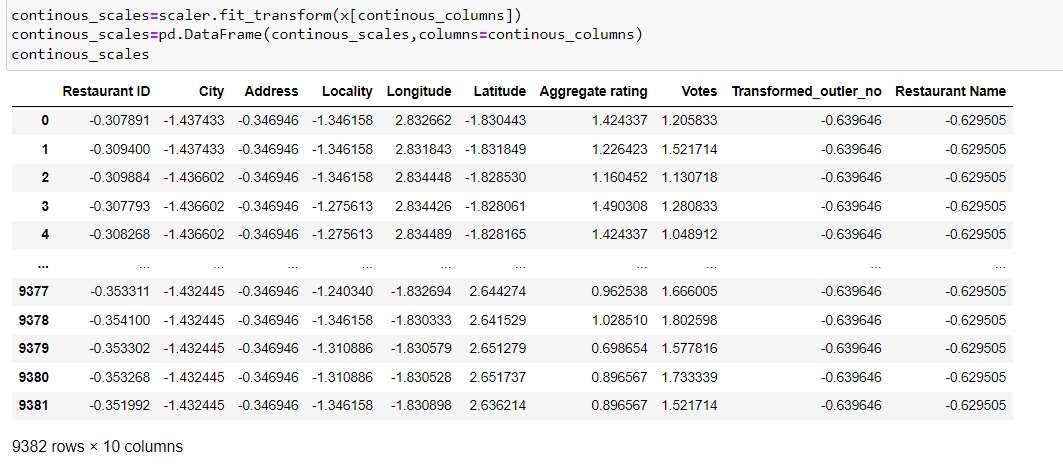


So this concludes the classiﬁcation task now let us proceed to the regression task.

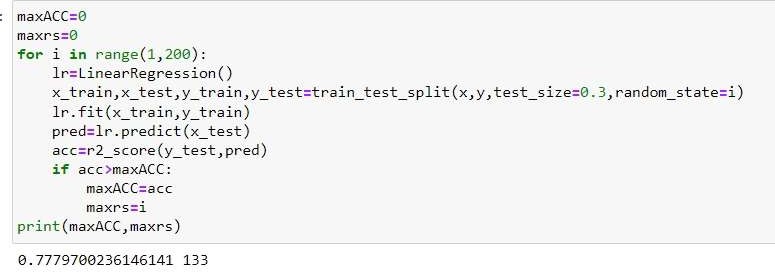
Now for the regression task the average price is the label the price range is the feature column so we have to encode the feature column respectively and after the encoding the feature column looks like this.



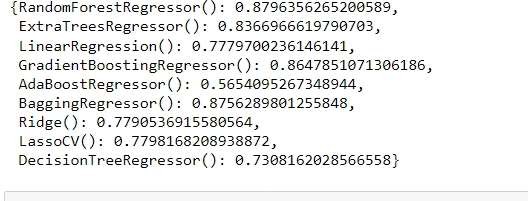
We dropped the continuous columns from the dataset and replaced them with the standardized continuous columns.



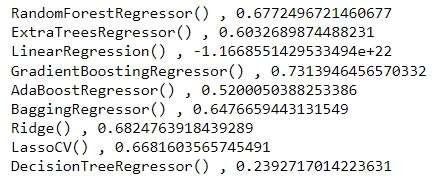
At ﬁrst we will be using all the feature columns and the default label also. #Finding the best random state for the train test split



So we divided the feature and the label on the basis of the best random state. Accuracy of different models on the basis of this split:-

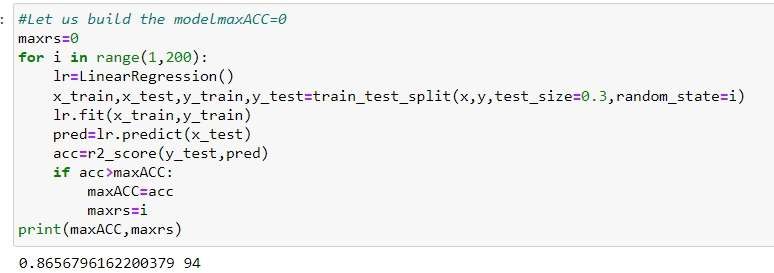


Cross validation score:-



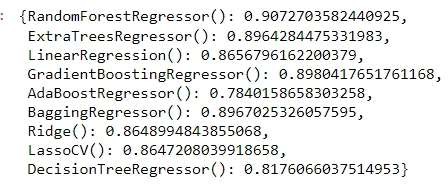
Using this way the models are performing very poorly. So we tried other way for this.

This time we will be using the power transformed label instead of the default label. #Finding the best random state for the split this time:-

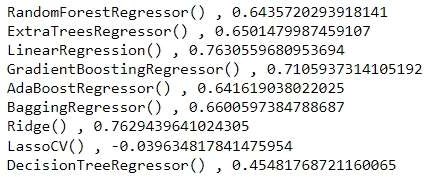


We can clearly see a increase In the accuracy of linear regression also

The accuracy of other models based on this split and the transformed label:-



Clearly a very big increase in all the models using the transformed label. The cross validation score:-



The gradient boost has been the best model we had done the tuning for this.

And after the tuning it had an r2\_score of 0.91 and a mean absolute error Of 0.2351706166989455. And the mean\_squared\_error was 0.10003684855627662.

So we also made the model using other techniques for example- by deleting t he restaurant id or by deleting the columns with high correlation but it re sulted in a decreased accuracy.



So this concluded the model building.

We used the transformed label . SO the actual label and predicted label for the best model where we used all the feature columns along with the Transformed label is below:-

We can clearly see that the prediction is close to the actual and there is not a lot of difference.

This concludes the model building.

1. Conclusion:-

Conclusions.(for classiﬁcation part where we predicted the price range of the restaurants):-

1. At ﬁrst we need to do a lot of preprocessing of the feature columns as there were a lot of categorical columns which needed to be encoded accordingly. So at ﬁrst we treated all the columns accordingly and made them suitable for the model buiding process.

After that we treated the outliers for the continous columns and deleted the outliers. In the last we separated the

label and the feature columns and started building the classiﬁcation model

1. For the classiﬁcation task we made the models in 3 ways:-
   1. In the ﬁrst technique we made the model without considering the multicollinearity and the variance inﬂation factor score

and mde the model using all the feature columns. In this way also some models like the random forest, gradient boost gave us

a great accuracy and the best accuracy was of the gradient boost which was 96%.

* 1. In the second time we deleted the columns with high vif scores and this resulted in a better accuracy. This time each model

performed better than the previous one and we got the best accuracy of 98% which was of the gradient boost and after the

tuning this gave us an accuracy of 98.2% which was the best model for us for the classiﬁcatin part

* 1. The best model was the gradient boost classiﬁer where we deleted the feature column with high variance inﬂation score and

also we used the transformed average cost column as it had a very high skewness so this is why we had to use the transformed column for this.

Conclusions.(for regression part where we predicted the average cost of the restaurants):-

1. The preprocessing was the same for this task also . They only difference was that this time the label was the average cost column and the price range column was the feature this time. So what we did is we deleted this time the average cost from the dataset and encoded the price range column accordingly and than we started building the model.
2. This time also we made the model using many techniques:-
   1. At ﬁrst we made the model using all the feature columns after treating and deleting the outliers. And model performed well

in this case also with randomforest regressor with the best accuracy of 87%.

* 1. After that we deleted the columns with high correlation and this resulted in a slight better accuracy.
  2. In the last we used the transformed label because the actual default label was having a very high skewness and it had a

very high no of outliers and this resulted in a very good accuracy. The best performing model was the gradientboost

regressor with an accuracy of 90.1% after the tuning and this ws 3% more than the previous best.

1. The best performing model was the gradient boost regressor with an r2\_Score of 90.1%. And in this model we deleted the

feature columns with very high correlation and we used the transformed label which resulted in a extra 3% accuracy.

this model had an mean abolute error of 0.2351706166989455 and mean squared error of 0.1000 respectively.

In the last we also converted the label and the prediction to the actual form to see the actual difference between our prediction and the actual values.

This concludes this project

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THANKS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*