INTRODUCTION

Mobile now a days is one of the most selling and purchasing device. Every day new mobiles with new version and more features are launched. In this competitive mobile phone market, companies want to understand the sales data of the mobile phones and factors that drive the prices.

The objective is to find out some relation between features of a mobile phone (eg :- RAM, Internal Memory etc) and its selling price.
Further, based on different features of the mobile phones, develop a model that would classify each mobile into different categories of price range i.e. into categorical values of 0(low), 1(moderate), 2(high), and 3(very high).

DATA SUMMARY

- We have in total of 21 features (20 independent and one dependent) and 2000 observations.
- Our Dependent feature which we have to predict i.e. price range has multiclass categorical value 0,1,2 and 3.
- Overall the data is perfectly balanced (500 observations for each class) with minimal number of missing values.
- There is little to no correlation between our independent features, otherwise it was handled.

Columns Description

battery_power :- total energy a battery can store in one time measured in mAh

int_memory - internal memory in Gigabytes

blue:- has bluetooth or not (0,1)

clock_speed :- speed at which microprocessor executes instructions.

fc:- front camera Mega Pixel

four_g - has 4G or not(0,1)

dual_sim:- has dual sim support or not (0,1)

. . .

m_dep - mobile depth in cm

mobile_wt - weight of mobile phone

n_cores - number of cores of processor

pc - primary camera mega pixels

px_height - pixel resolution height

px_width - pixel resolution width

ram - random access memory in Mega Bytes

. . .

sc_h - screen height of mobile in cm

sc_w - screen width of mobile in cm

talk_time - longest time that a single battery charge will last when you are

three_g - has 3G or not

touch_screen - has touch screen or not

wifi - has wifi or not

price_range - this is the target variable with value of 0(low cost), 1(medium cost), 2(high cost) and 3(very high cost)

Preliminary EDA
And
Data Preprocessing

We have renamed columns to suitable & understandable names.

old_name	new_name
‘blue’	‘bluetooth’
‘fc’	‘front_cam’
‘four_g’	‘4G’
‘int_memory’	‘ROM’
‘m_dep’	‘depth’
‘mobile_wt’	‘weight’
‘n_cores’	‘cores’

old_name	new_name
‘pc’	‘rear_cam’
‘px_height’	‘resol_height’
‘px_width’	‘resol_width’
‘ram’	‘RAM’
‘sc_h’	‘height’
‘sc_w’	‘width’
‘three_g’	‘3G’

Divided the Features into categorical and continuous values for better understanding of the data.

Categorical Variables
dual_sim
wifi
3G
cores
4G
bluetooth
Touchscreen

Continuous variables
Resolution
Rear_cam, front_cam
Dimension
RAM, ROM
Clock_speed
Weight
Battery_power

The column ‘resol_Height’ had two null values, so both the observations were dropped.

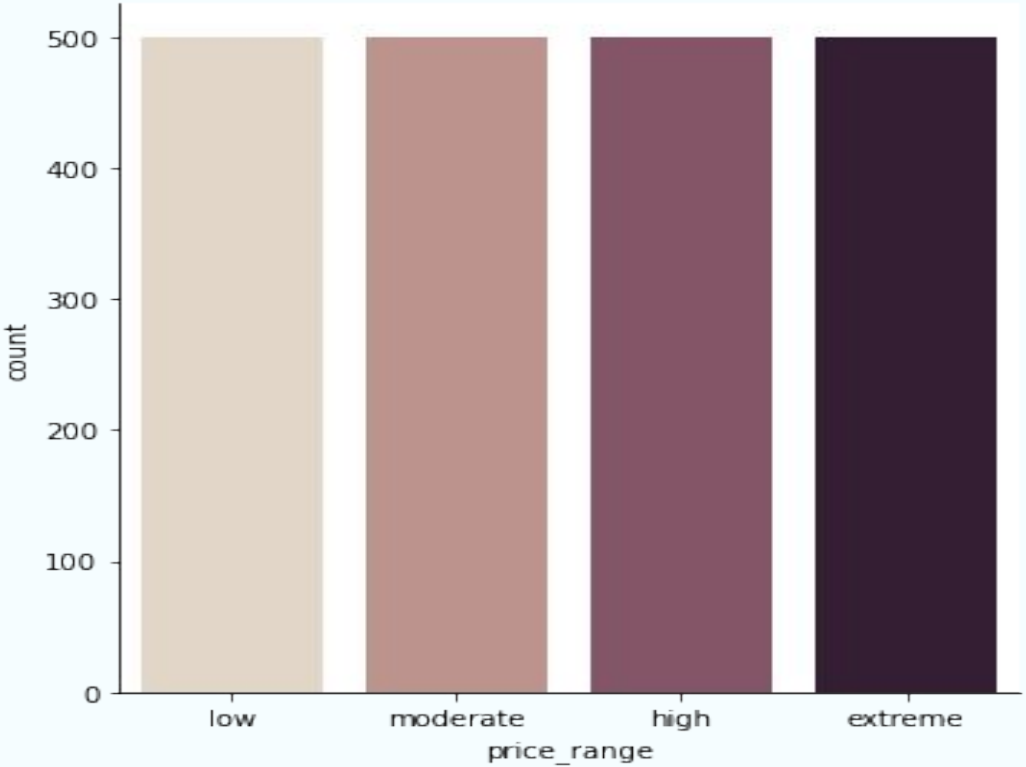
Also the column ‘width’ had 180 rows labelled as zero.
These 180 values were imputed through KNNImputer(n=1) method from the sklearn.impute module

Using statsmodel library, we calculated variance inflation factor for all features and dropped weight column because it had high value.

screen_height & screen_width columns was clubbed together as length of the phones.
Similarly, px_height & px_width columns were clubbed into one feature as resolution.

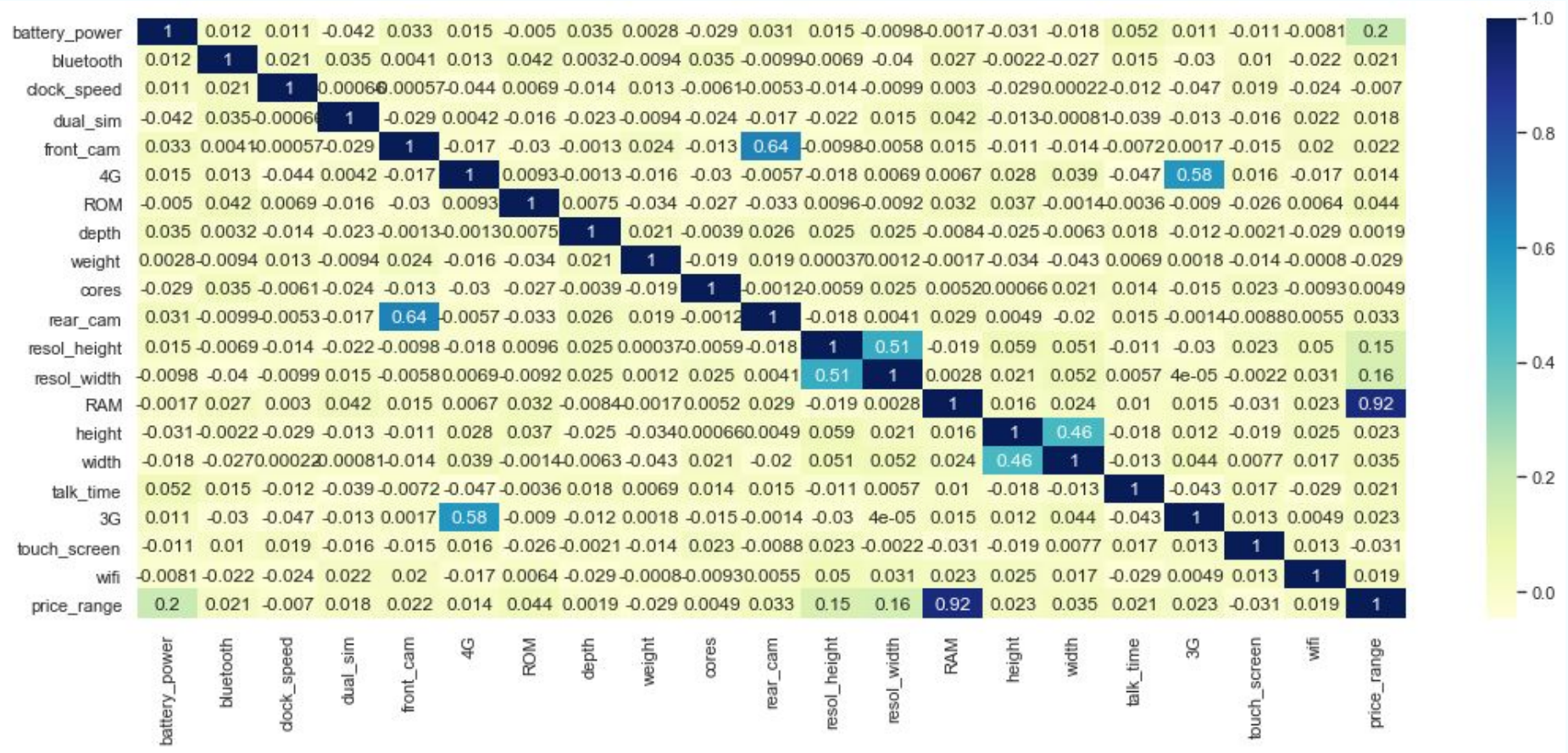
Final data prepared had 18 columns and 1998 number of rows.

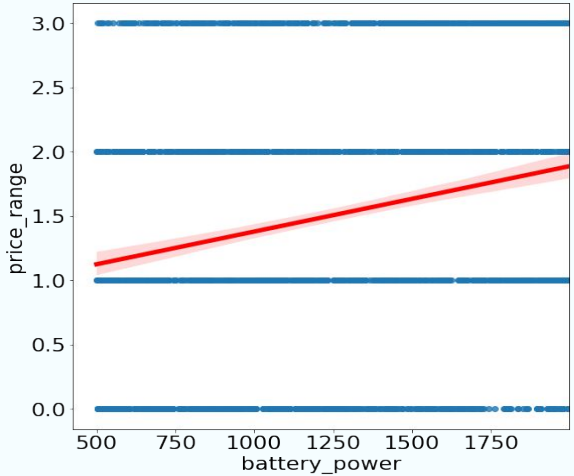
Graph of No. of Observations for each Price range.

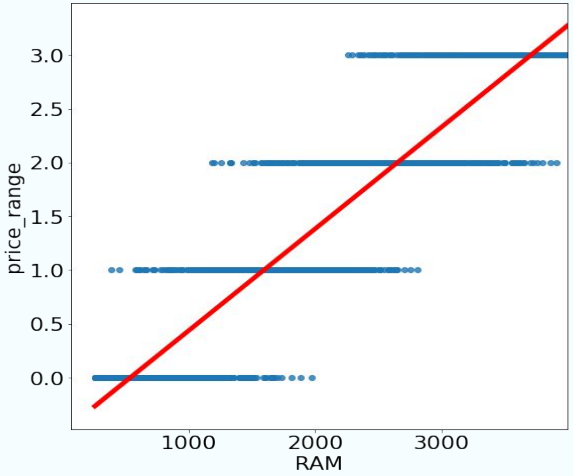


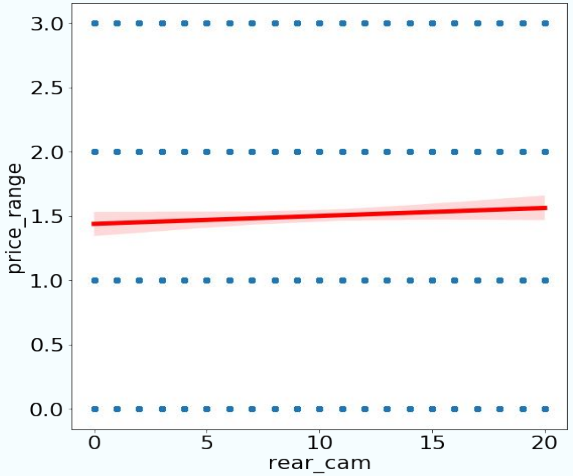
The data is equally divided into all the four class for price range i.e there is no class imbalance

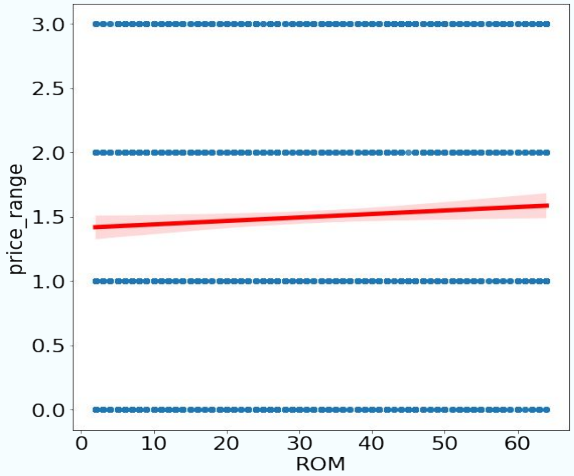
Correlation Heat Map.

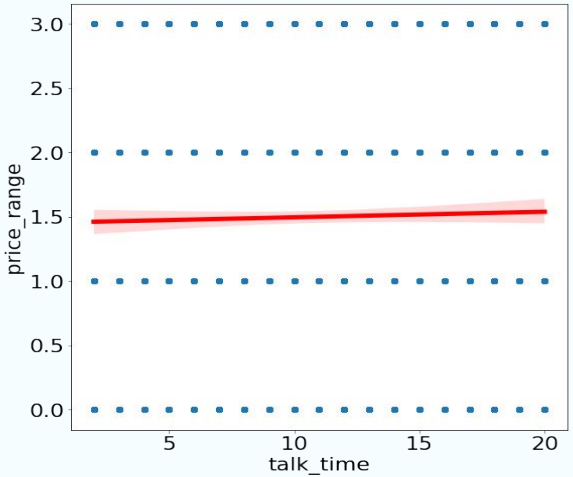


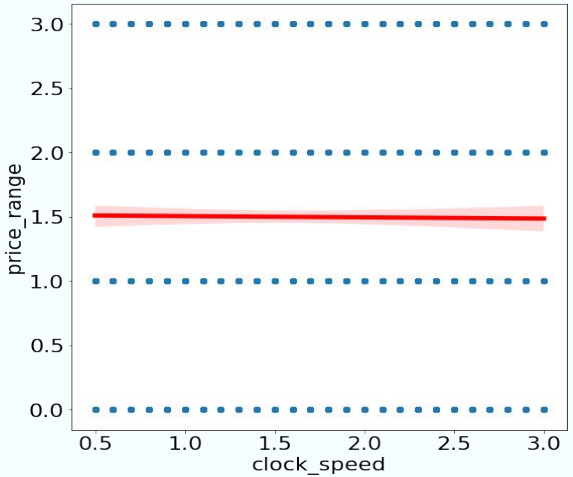

price_range
battery_power
500
750
1000
1250
1500
1750
0.0
0.5
1.0
1.5
2.0
2.5
3.0


price_range
RAM
1000
2000
3000
0.0
0.5
1.0
1.5
2.0
2.5
3.0


price_range
rear_cam
0
5
10
15
20
0.0
0.5
1.0
1.5
2.0
2.5
3.0


price_range
ROM
0
10
20
30
40
50
60
0.0
0.5
1.0
1.5
2.0
2.5
3.0


price_range
talk_time
5
10
15
20
0.0
0.5
1.0
1.5
2.0
2.5
3.0

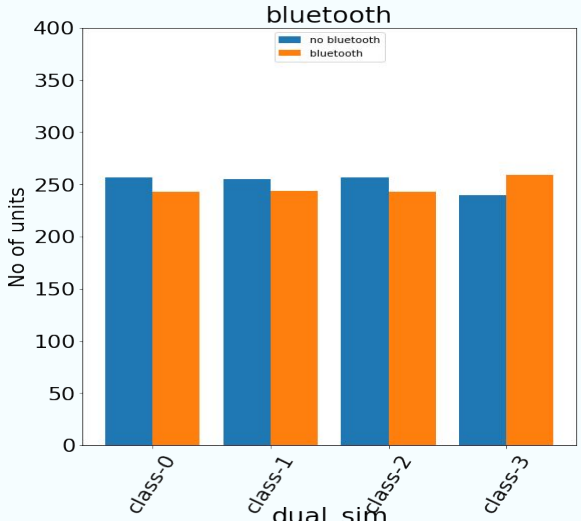

price_range
clock_speed
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

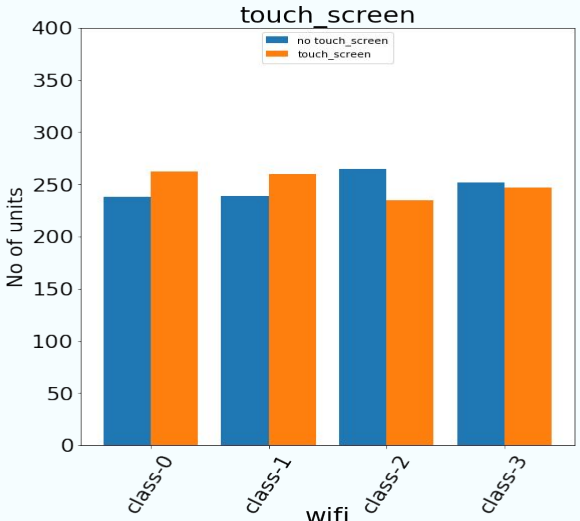
On plotting Regression Plots of all the continuous independent feature with our dependent feature (for class 0, 1, 2, 3),

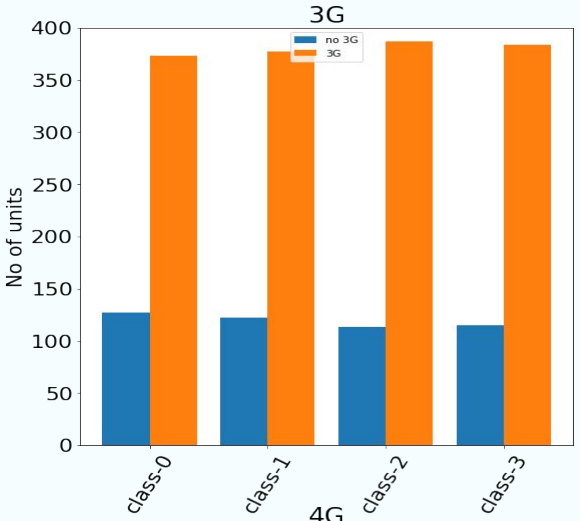
We could see that that RAM of mobile is very highly correlated with the price range associated with the mobile

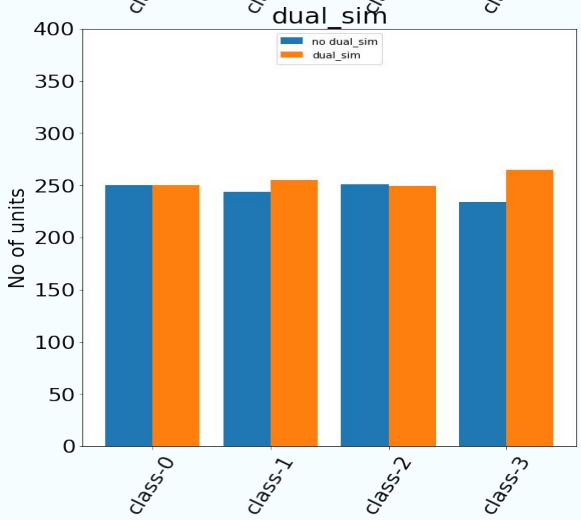
Also Battery_power shows similar trend but not to that extend

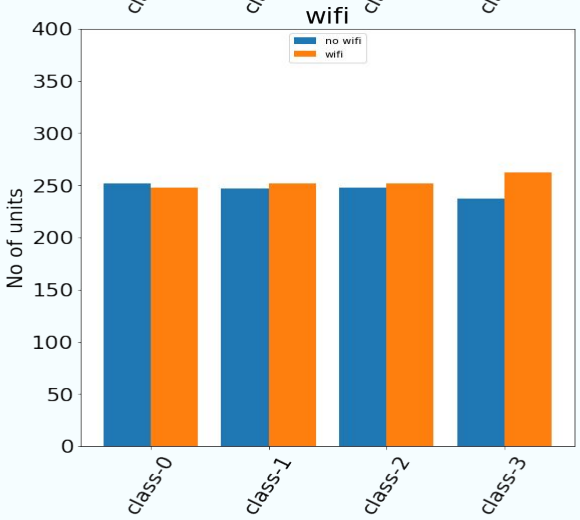
Others continuous features showed little to no collinearity with our dependent feature i.e Price range

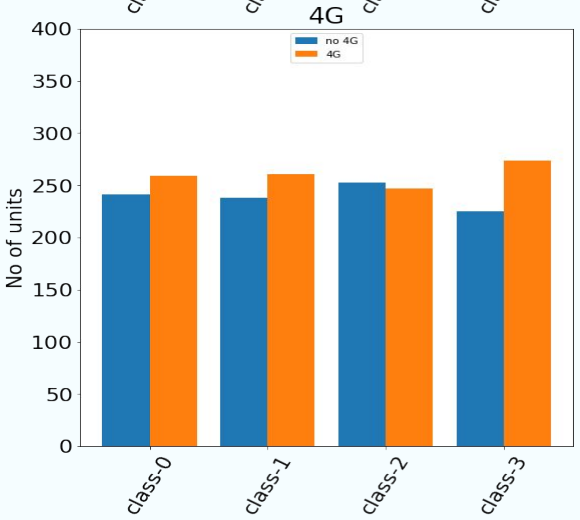

bluetooth
no bluetooth
bluetooth
No of units
0
50
100
150
200
250
300
350
400
class-0
class-1
class-2
class-3


touch_screen
no touch_screen
touch_screen
No of units
0
50
100
150
200
250
300
350
400
class-0
class-1
class-2
class-3


3G
no 3G
3G
No of units
0
50
100
150
200
250
300
350
400
class-0
class-1
class-2
class-3


dual_sim
no dual_sim
dual_sim
No of units
0
50
100
150
200
250
300
350
400
class-0
class-1
class-2
class-3


wifi
no wifi
wifi
No of units
0
50
100
150
200
250
300
350
400
class-0
class-1
class-2
class-3


4G
no 4G
4G
No of units
0
50
100
150
200
250
300
350
400
class-0
class-1
class-2
class-3

On plotting Bar Plots of all the Categorical independent feature with our dependent feature against all class

We can observe that number of the mobile phones with 3G is more than phones that don't have 3G.
Also, we can see that there is no imbalance of phones having or not having 3G with the price range associated with the mobile phones i.e the observations having 3G is equally distributed across all class and observations not having 3G is also distributed equally across all class.

For all other features the number of observation is equal across all class and also within the feature itself.

Model Implementation

Since our problem was of classifying mobile phones into four different class based on its features, we implemented classification algorithms using the sklearn library

- KNN
- Logistic Regression
- Decision Trees
- Random Forest
- AdaBoost
- GradientBoost
- XGBoost

Splitting :- Dividing the data set onto X(independent) and Y(dependent) variable and splitting it into train and test using sklearn's train_test_split function

Scaling :- Also, Feature scaling was done on the data using MinMaxScaler & Standard Scaler function in order to implement KNN and Logistic Regression

Hyperparameter Tuning :- Every algorithm was implemented using the default parameter settings from sklearn library.
Also using the BayesSearchCV func, for each algo, the best parameters settings was calculated based on the accuracy of the model.

EVALUATION

Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1 Score
- AUC RUC Curves

Since, this was a multiclass classification problem with very well balanced data, we used accuracy and F1 score which generally caters to better evaluation insights in classification problems.
Also, AUC-ROC curves for all class were plotted using one vs rest technique i.e class 0 vs class 1,2 & 3 combined and so on.

Before Tuning

	Train Data				Test Data			
	Accuracy	Precision	Recall	AUC_ROC	Accuracy	Precision	Recall	AUC_ROC
KNN	67.9	0.685	0.679	0.869	59.00%	0.614	0.590	0.837
Decision Tree	100.0%	1.000	1.000	1.000	85.5%	0.855	0.855	0.903
Logistic Regression	92.68%	0.927	0.927	0.992	89.25%	0.892	0.892	0.990
Random Forest	100.0%	1.000	1.000	1.000	87.5%	0.875	0.875	0.982
AdaBoost	100.0%	1.000	1.000	1.000	90.25%	0.902	0.902	0.987
Gradient Boost	99.69%	0.997	0.997	1.000	89.5%	0.895	0.895	0.988
XGBoost	100.0%	1.000	1.000	1.000	92.0%	0.920	0.920	0.992

After Tuning

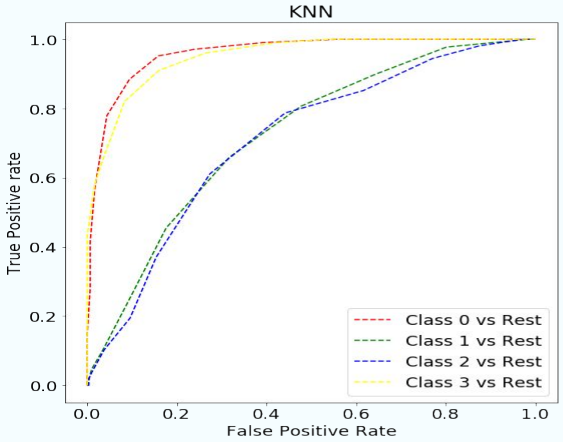
	Train Data				Test Data			
	Accuracy	Precision	Recall	AUC_ROC	Accuracy	Precision	Recall	AUC_ROC
KNN	66.5%	0.668	0.665	0.868	61.75%	0.637	0.618	0.838
Decision Tree	95.56%	0.956	0.956	0.998	87.25%	0.872	0.872	0.948
Logistic Regression	94.99%	0.950	0.950	0.997	93.0%	0.930	0.930	0.996
Random Forest	100.0%	1.000	1.000	1.000	88.75%	0.888	0.888	0.986
AdaBoost	100.0%	1.000	1.000	1.000	90.5%	0.905	0.905	0.991
Gradient Boost	100.0%	1.000	1.000	1.000	90.5%	0.905	0.905	0.991
XGBoost	94.43%	0.944	0.944	0.997	93.25%	0.932	0.932	0.996

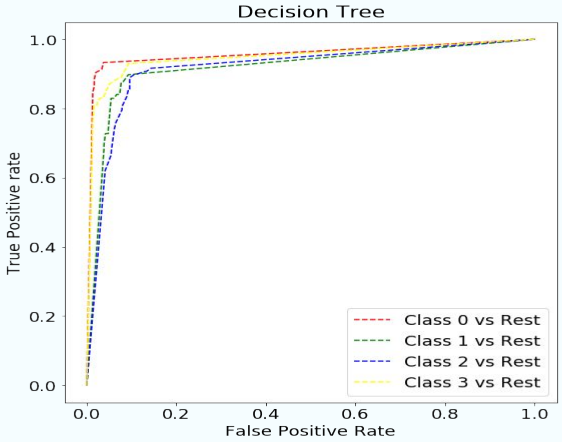
F1 score

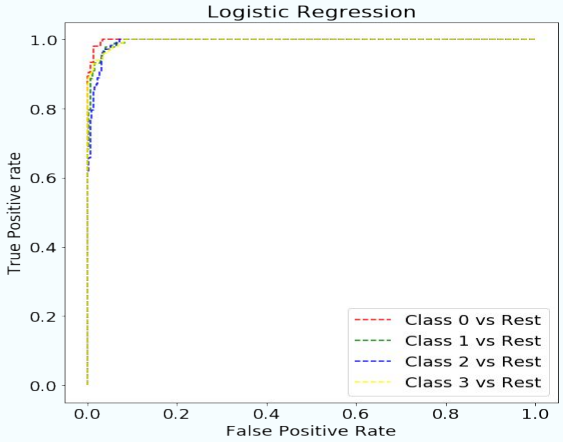
	Class-0	Class-1	Class-2	Class-3	Overall
KNN	0.74	0.49	0.51	0.73	0.622666
Decision Tree	0.92	0.82	0.84	0.91	0.872959
Logistic Regression	0.96	0.92	0.91	0.93	0.930094
Random Forest	0.94	0.83	0.82	0.90	0.874724
AdaBoost	0.96	0.92	0.89	0.92	0.919715
Gradient Boost	0.95	0.88	0.88	0.91	0.905377
XGBoost	0.96	0.93	0.92	0.94	0.940114

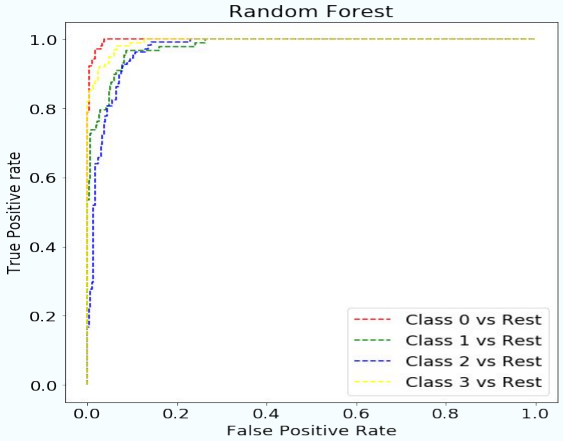
The F1 score for class 1 and class 2 is low as when compared to class 0 and class 3 across all algorithms. Overall F1 score for Logistic Regression and XGBoost algo faired better than others and KNN was the worst performing among all.

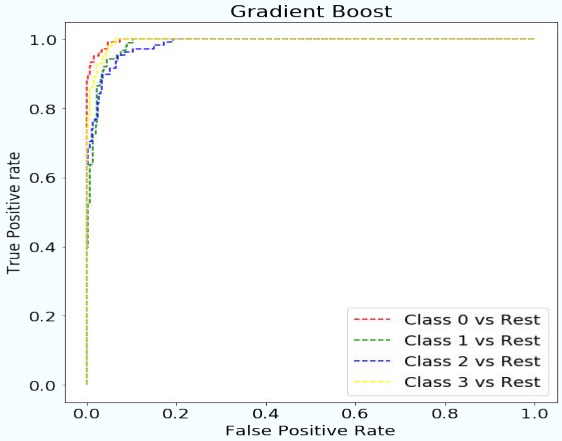
Comparision Of AUC_ROC plots

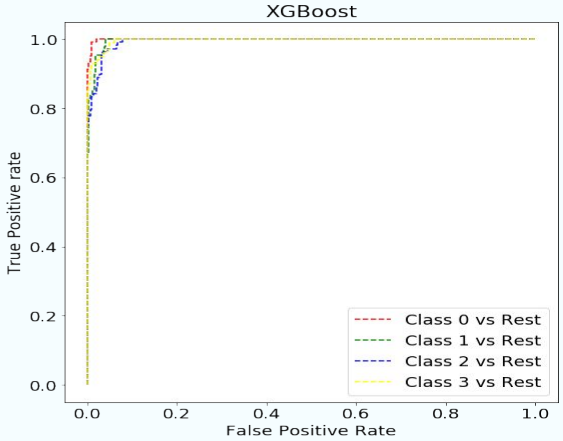












Conclusion

- As this was a multiclass classification problem and the data set was very well balanced across all output class, Accuracy was a good parameter to evaluate our models.
- In addition to Accuracy, we also calculated F1 scores across each output class to get overall picture.
- Based on the evaluation metrics(Accuracy, Precision, Recall, F1 score)of the train and test data, we can conclude that Logistic Regression and XGBoost performed well among all models whereas KNN performed worst.
- XGBoost is our goto model since it performed well and it is non parametric in nature for final model selection.

So we completed our exhaustive study to develop a model for mobile price ranges based on its features, and this model can help us in doing market research, competitive price estimates, and mobile phone's Price vs Features comparisons.