Capstone Project - 3

MOBILE PRICE RANGE PREDICTION

SUPERVISED MACHINE LEARNING (CLASSIFICATION)

BY
RUSHIKESH ARJUN MANE
(Cohort Seattle)

Contain

1. Problem Statement
2. Work Flow
3. Data Collection and Understanding
4. Data Wrangling and Feature Engineering
5. EDA (Exploratory Data Analysis)
6. Preparation of Data for Model Building
7. Model Selection and Evaluation
8. Conclusion

Introduction

- In this Modern Era, Smartphones are an integral part of the lives of human beings.

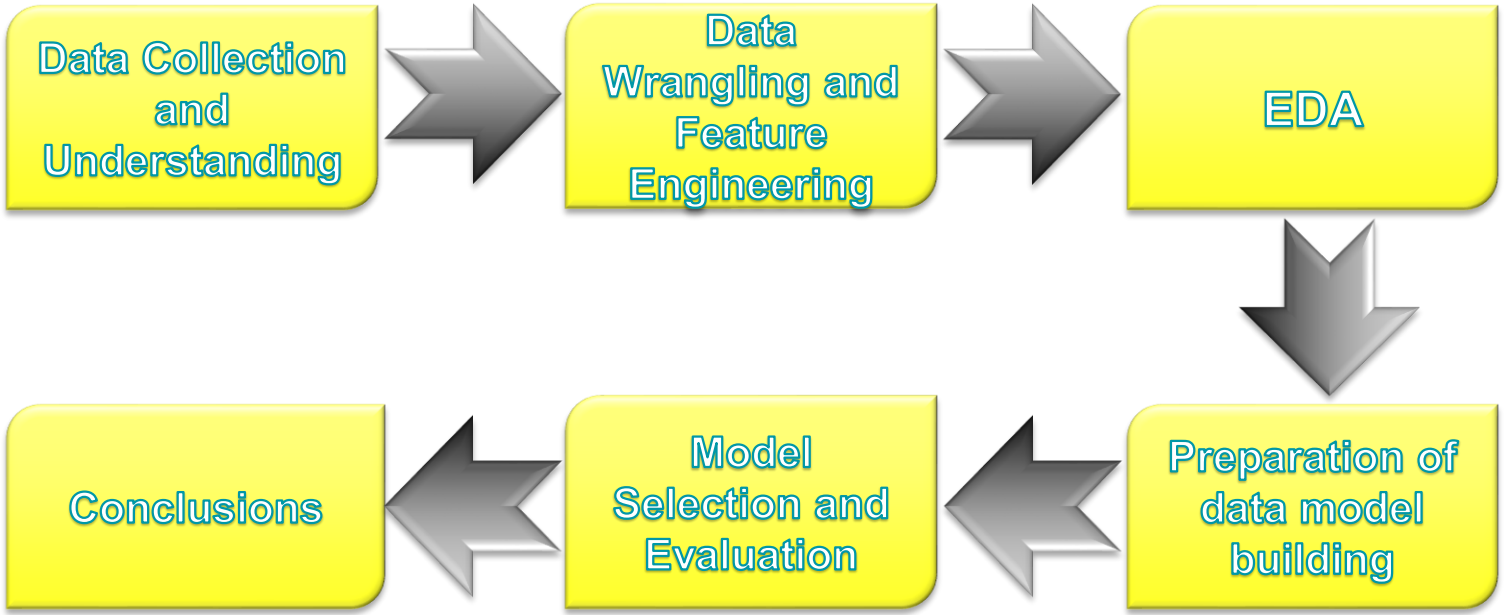
- When a smartphone is purchased ,many factors like the Display, Processor, Memory, Camera, Thickness, Battery, Connectivity and others are taken into account .

- Price estimation and prediction is an important part of consumer strategy. Deciding on the correct price of a product is very important for the market success of a product. A new product that has to be launched must have the correct price so that consumers find it appropriate to buy the product.

Problem Statement

- In the competitive mobile phone market companies want to understand sales data of mobile phones and factors which 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.
- In this problem, we do not have to predict the actual price but a price range indicating how high the price is.

Work Flow


Data Collection and Understanding
Data Wrangling and Feature Engineering
EDA
Preparation of data model building
Model Selection and Evaluation
Conclusions

Data Collection and Understanding

- For analysis and modeling we are having Mobile Price Range Data.
- The dataset contains 2000 rows and 21 columns.

- **DATA DESCRIPTION:**
  - **Battery_power** - Total energy a battery can store in one time measured in mAh
  - **Blue** - Has bluetooth or not
  - ***Clock_speed*** - speed at which microprocessor executes instructions
  - ***Dual_sim*** - Has dual sim support or not
  - ***Fc*** - Front Camera mega pixels
  - ***Four_g*** - Has 4G or not
  - ***Int_memory*** - Internal Memory in Gigabytes
  - ***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).
- Thus our target variable has 4 categories so basically it is a Multiclass classification problem.

Data Wrangling and Feature Engineering

- Dealing with mismatch values,

	count	mean	std	min	25%	50%	75%	max
px_height	2000.000000	645.108000	443.780811	0.000000	282.750000	564.000000	947.250000	1960.000000
sc_w	2000.000000	5.767000	4.356398	0.000000	2.000000	5.000000	9.000000	18.000000

```
# Checking How many observations having screen width value as 0.
print(df_main[df_main['sc_w']==0].shape[0])
```

```
180
```

```
# Checking How many observations having px_hieght value as 0.
print(df_main[df_main['px_height']==0].shape[0])
```

```
2
```

```
# As there are only 2 observations having px_height=0. so we will drop it.
df_main=df_main[df_main['px_height']!=0]
```

```
# Checking How many observations having sc_w value as 0.
df_main[df_main['sc_w']==0].shape[0]
```

```
0
```

- **Here we used K-Nearest Neighbours approach to impute the missing values where a Euclidean distance is used to find the nearest neighbours.**

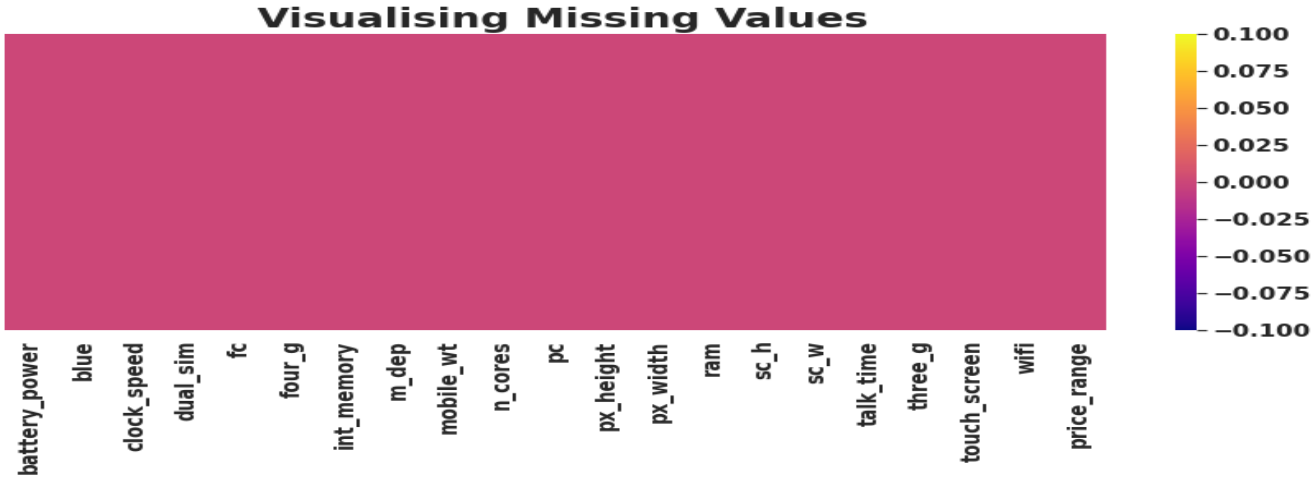
```
df_main.isnull().sum()
```

```
battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
dtype: int64
```

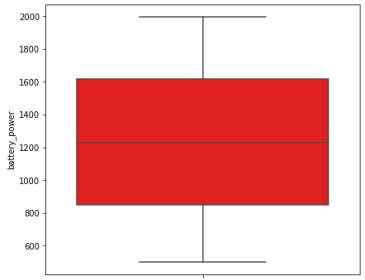
```
## Checking Duplicate rows in our Dataset.
duplicates=df_main.duplicated().sum()
print(f"We are haveing {duplicates} rows in our Dataframe.")
```

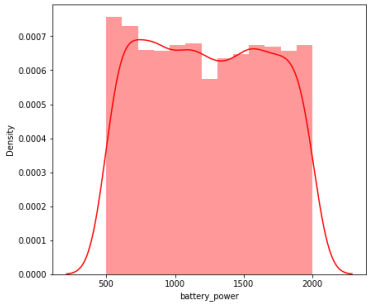
```
We are haveing 0 rows in our Dataframe.
```

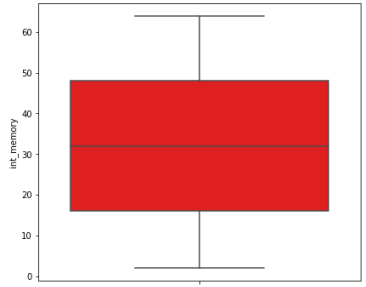
- **From the output we can see that we no having any missing value in our dataset after handling mismatching form our data.**
- **And also we can see that there in no duplicate value.**

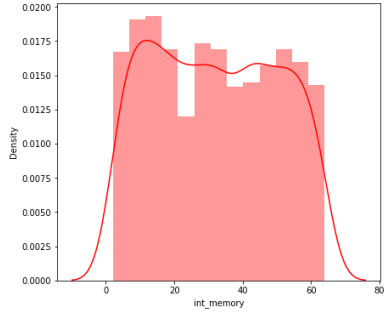


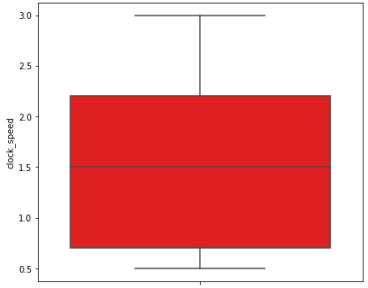
Checking Distribution of Numerical Variables and ouliers:

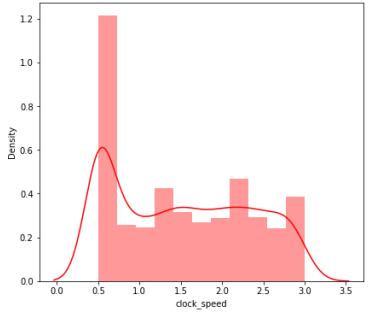


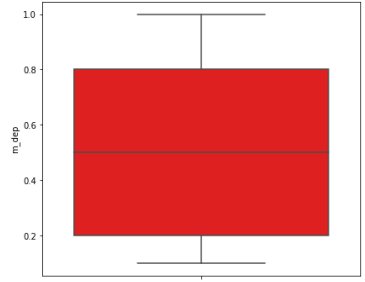


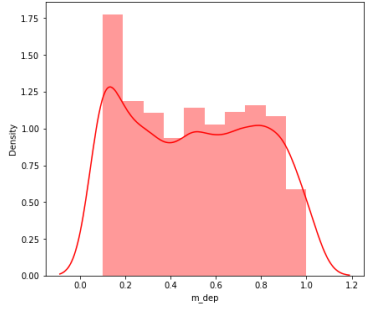


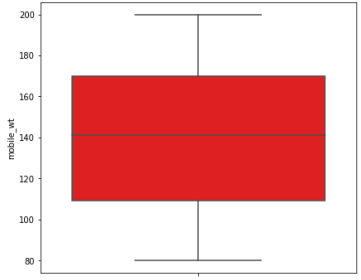


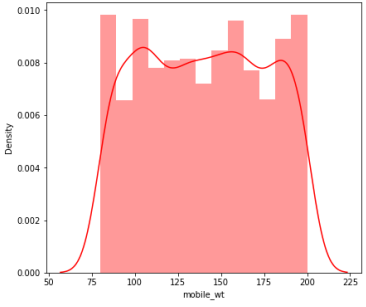


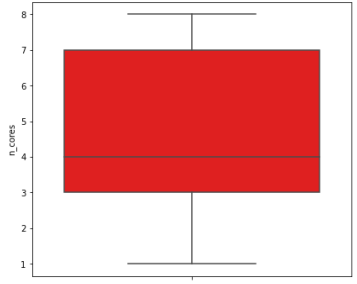


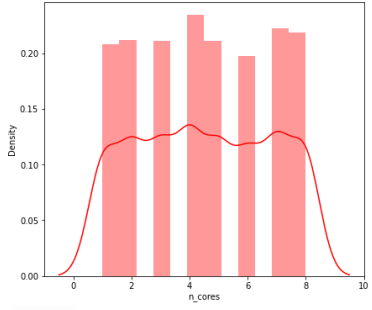


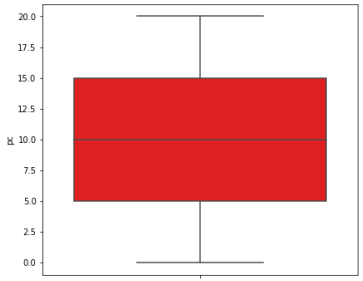


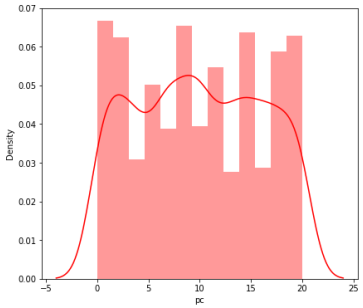


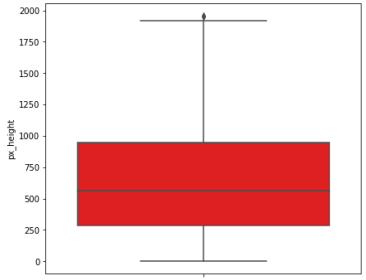


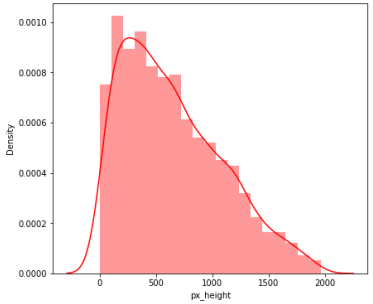


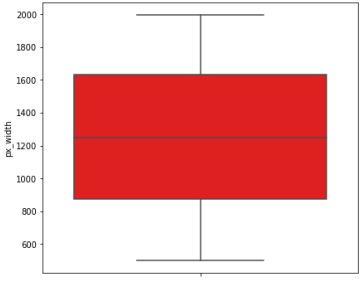


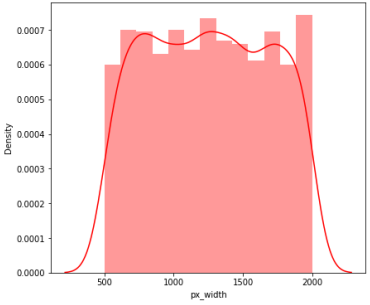

pc

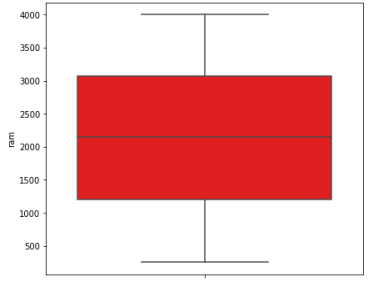

Density
pc

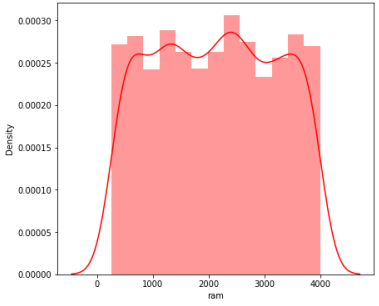

px_height

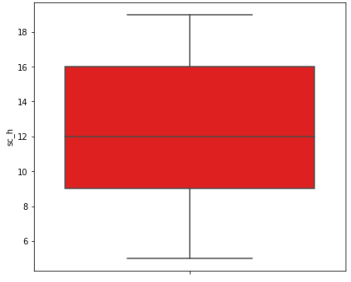

Density
px_height

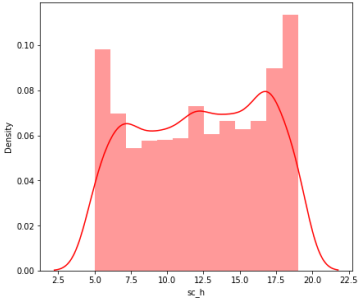

px_width

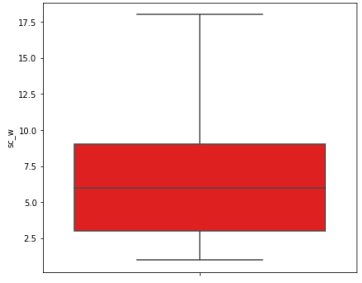

Density
px_width

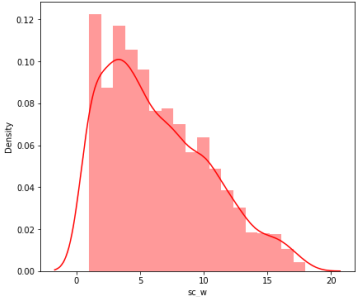

ram


Density
ram

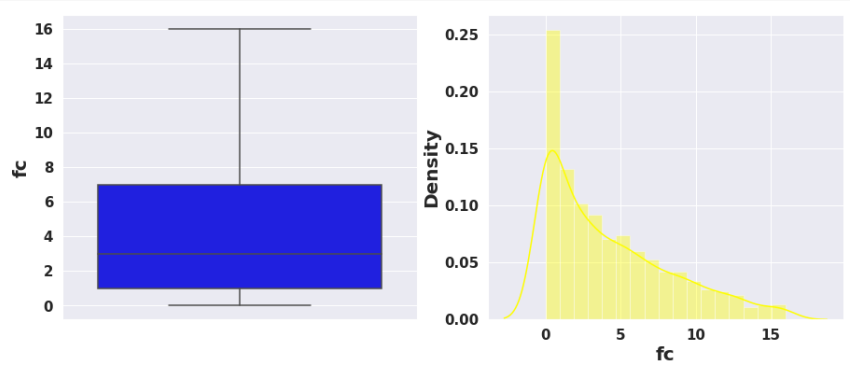

sc_h

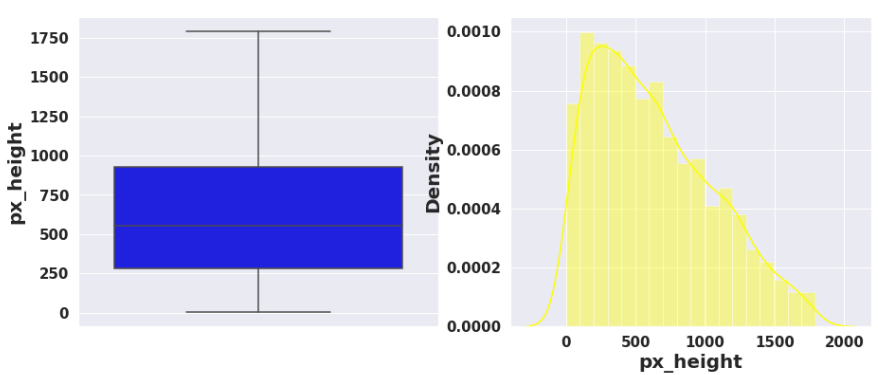

Density
sc_h


sc_w

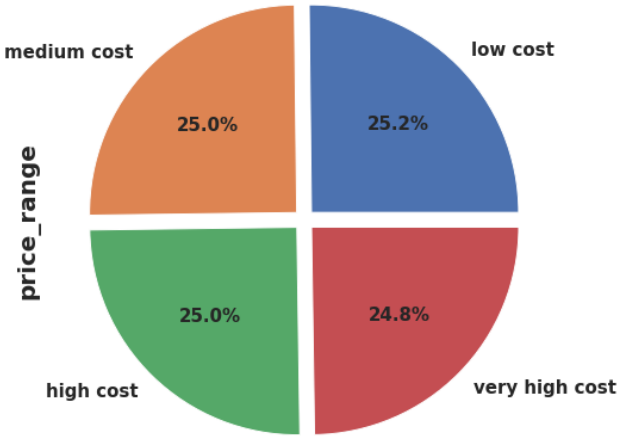

Density
sc_w

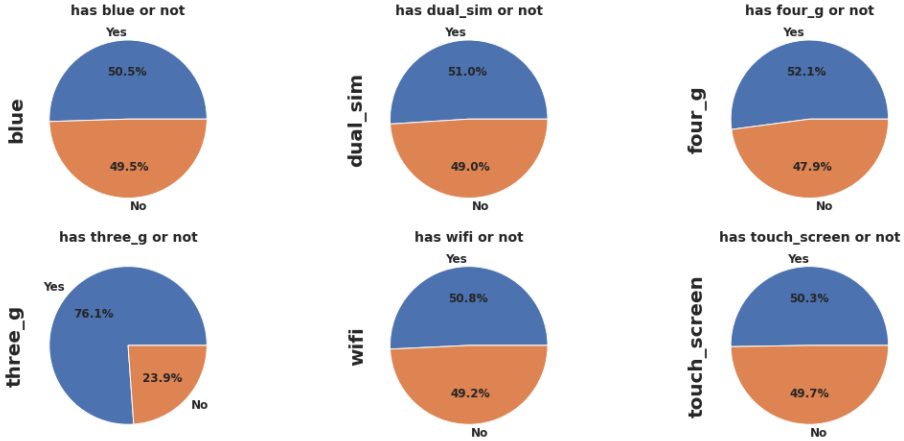
- Form the above chart we can see that our data is well distributed.
- And form boxplot we can see that fc and px_height has some outliers. So we removed those outlier and we can conform form below output that outlier are removed.



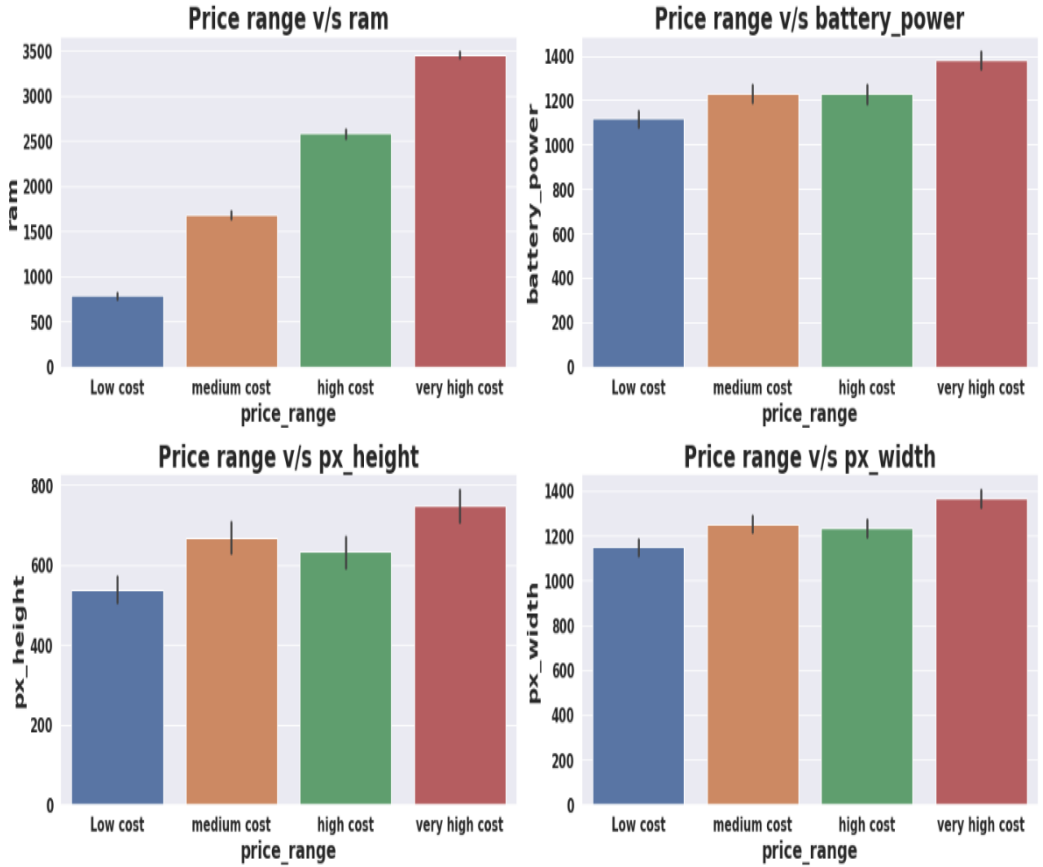


EDA :





- Target variable has equal number of observation in each category and target variable is nearly equally distributed.
- Percentage Distribution of Mobiles having bluetooth, dual sim, 4G,wifi and touch screen are almost 50 % and very few mobiles (23.8%) do not have 3G.

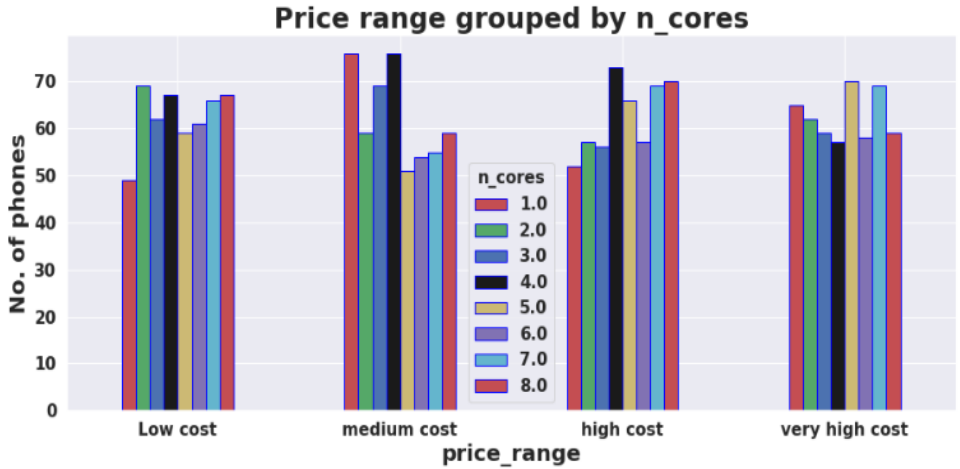


I. Mobiles having RAM less than 1GB are comes under low cost category.

II. Mobiles having more than 3GB RAM comes under very high cost category. So we can see that as RAM increases price of mobiles also increases.

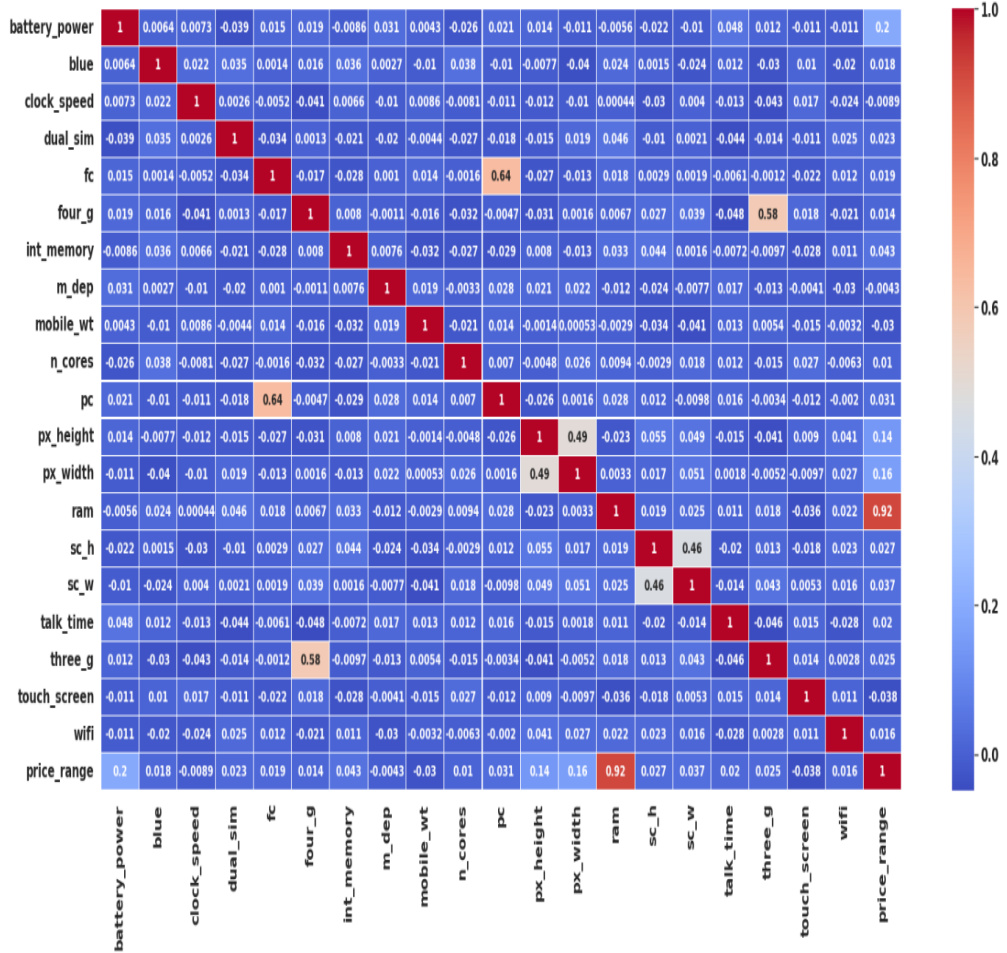
III. Mobiles more than 700 pixel hight and width more than 1300 has very high cost.

IV. Mobiles with battery power more than 1300 mAh has very high cost. And Mobiles with battery power between 1200 and 1300 mAh falls under medium and high cost category.



1. **There are very few mobiles in price range low cost and medium cost are having lesser number of cores.**

2. **Most of the mobiles in price range between high cost and very high cost are with more number of cores.**



Form the correlation matrix,

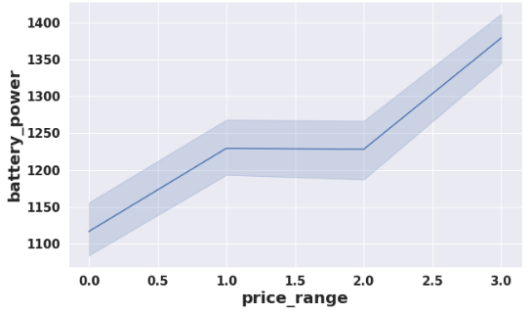
1. **Price range is having strong positive correlation with RAM. As we know mobiles are costly with high RAM.**

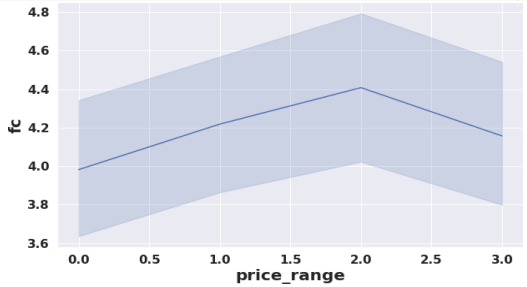
2. **Price range has positive correlation with Battery power. Mostly high price mobiles are having great battery backup.**

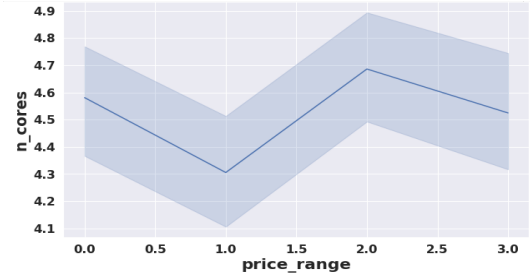
3. **Also px_height and px_width (Pixel Resolution Height and width) are positively correlated. Generally High price range mobiles have good resolutions.**

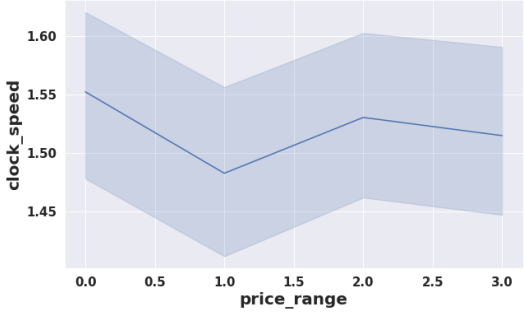
4. **3G and 4G are positively correlated with each other i.e. because nowadays all mobiles is rather 3G or 4G .**

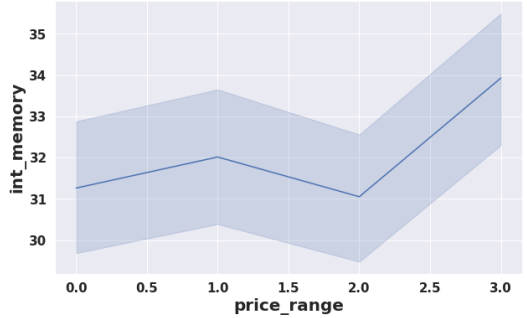
Different trends of price range (Target variable) with other attributes:

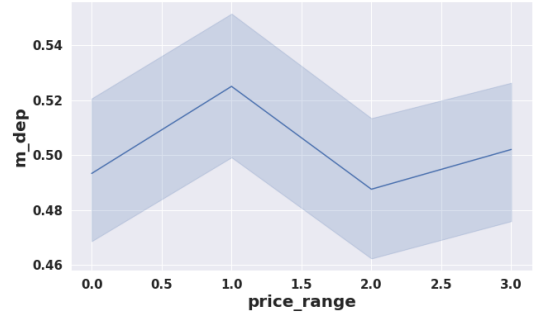


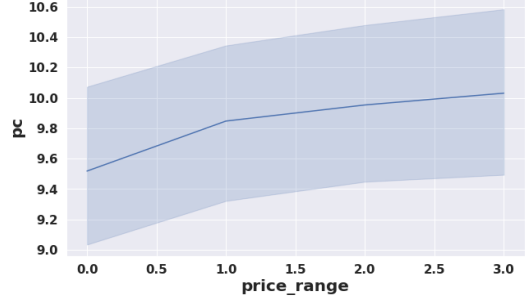


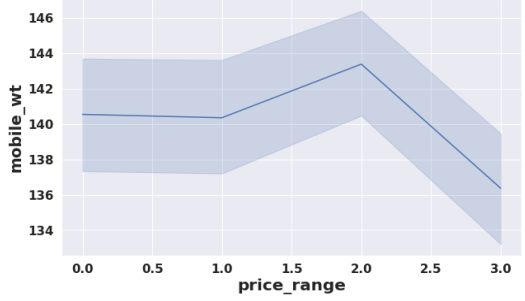


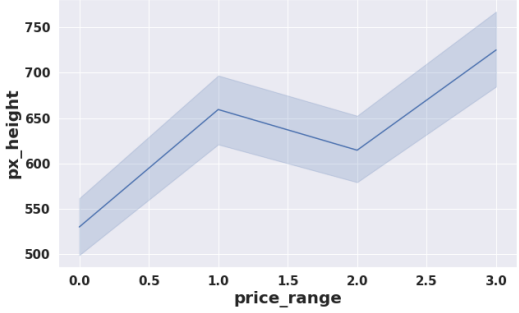


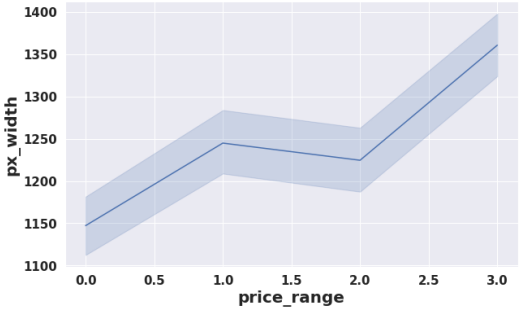


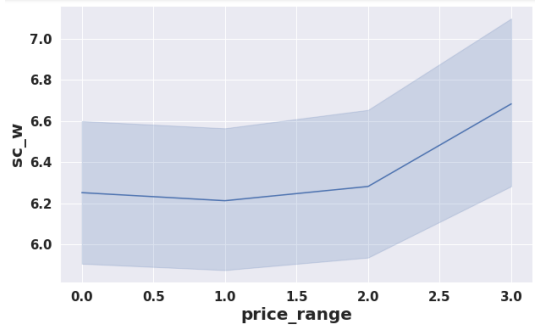


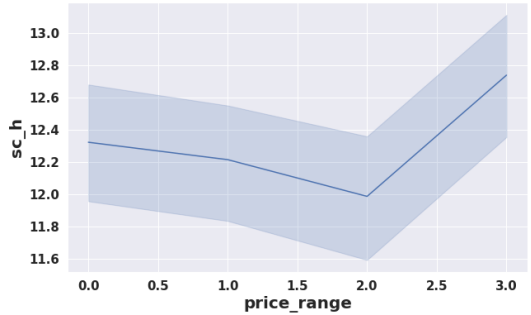


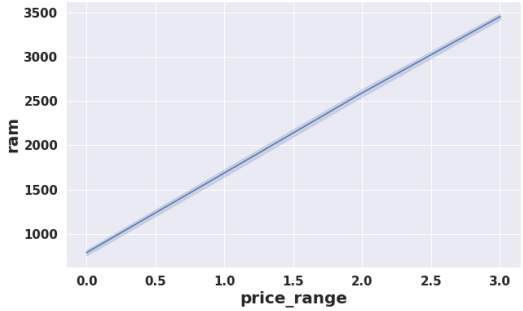


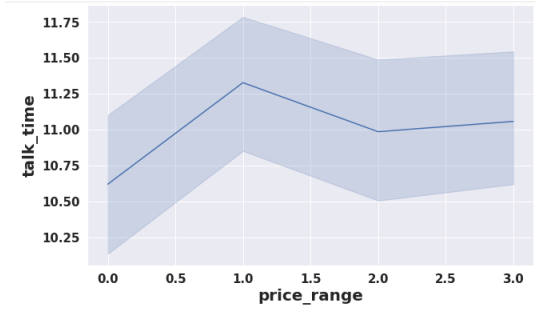



px_width
price_range


sc_w
price_range


sc_h
price_range


ram
price_range


talk_time
price_range

Model Selection and Evaluation:

Before doing analysis with models we performed the train test split. We kept 25% of the data for test and remaining 75% of the data for training the model.

We compared 6 Classification algorithms and evaluated them based on the overall accuracy score and the recall of the individual classes.

•Accuracy is the ratio of the total number of correct predictions and the total number of predictions.

•The recall is the measure of our model correctly identifying True Positives.

1. **Decision Tree**
2. **Random Forest Classifier**
3. **Gradient Boosting Classifier**
4. **K-Nearest Neighbor Classifier**
5. **XG Boost Classifier**
6. **Support Vector Machine (SMV)**

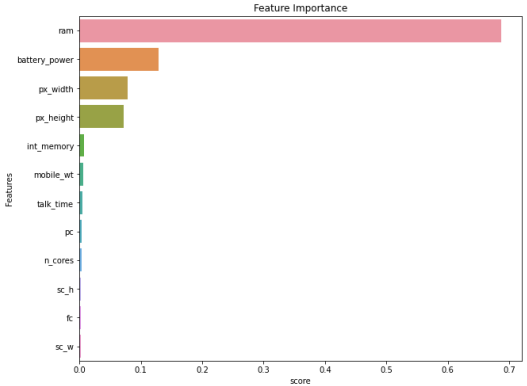
Decision Tree 2) Random Forest classifier 3) Gradient Boosting Classifier 4) K-nearest Neighbor classifier 5) XG Boost Classifier 6) Support Vector Machine(SVM)

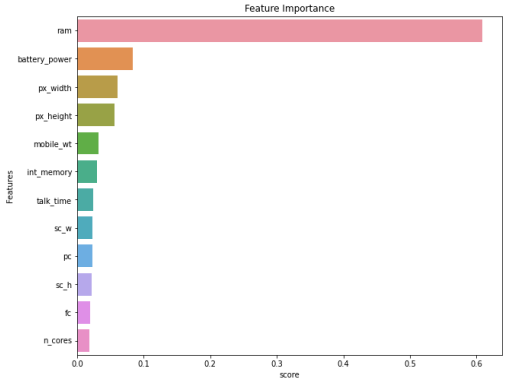
Evaluation of Models:

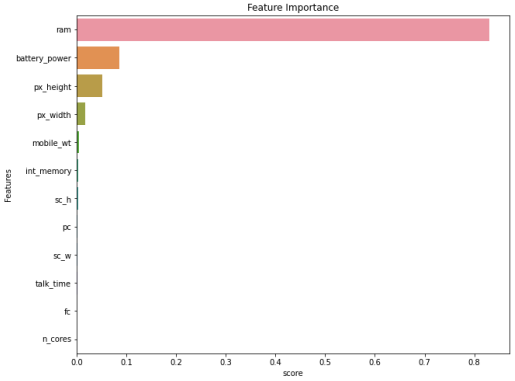
Algorithms	Training Set		Test Set	
	Accuracy score (%)	Recall (%)	Accuracy score (%)	Recall (Average of all 4 Classes)
Decision Tree	100	100	84	83.75
Decision Tree (Hyper-parameter Tuning)	97.62	97.5	85.13	84.75
Random Forest	100	100	88.6	88.5
Random Forest(Hyper-parameter Tuning)	100	100	89.81	89.5
Gradient Boosting	100	100	90.02	90
Gradient Boosting (Hyper-parameter Tuning)	100	100	90.42	90.5
KNN	75.86	76	59.47	59.25
KNN (Hyper-parameter Tuning)	76.61	76.75	70.26	69.75
XG-Boost	98.98	98.75	90.22	90
XG-Boost (Hyper-parameter Tuning)	100	100	92.46	92.25
SVM	98.57	98.5	89.81	89.75
SVM (Hyper-parameter Tuning)	98.3	98.5	97.96	98

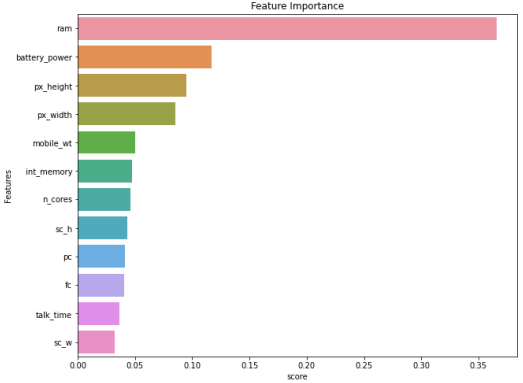
- **After hyper-parameter tuning the Best Fitted model came out to be SMV.**
- **XG boost (Hyper-parameter Tuned) can be considered as the second most good model.**
- **The worst performed model is KNN model.**

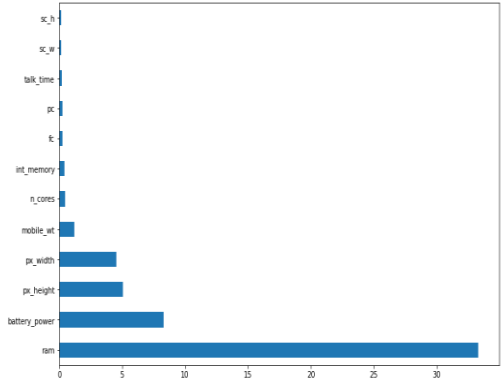
Feature Importance:





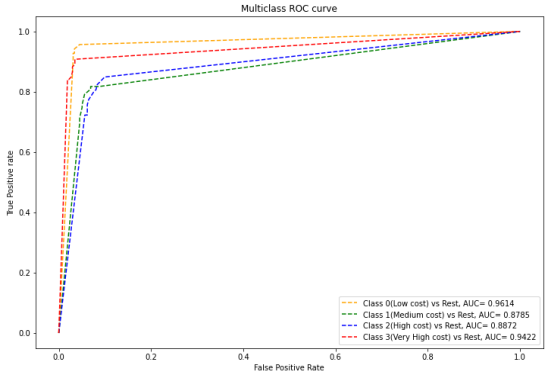


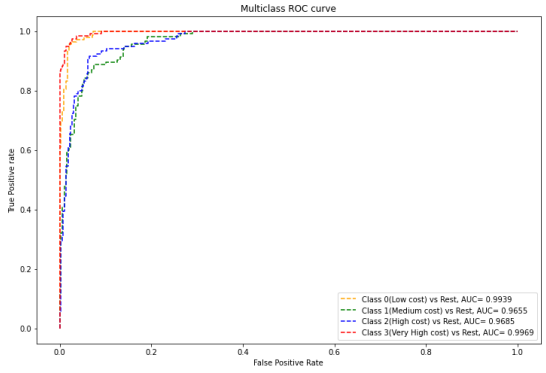


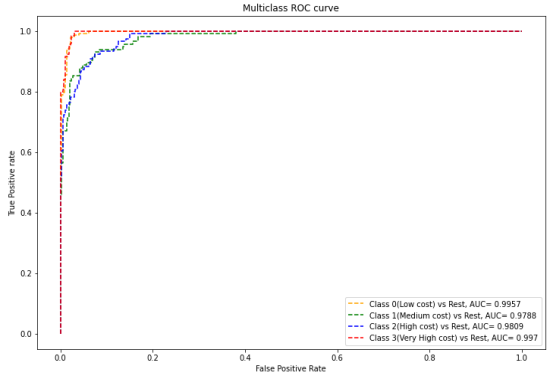


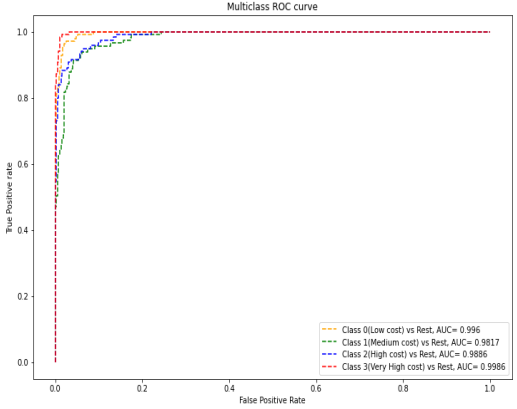
RAM, Battery Power, Pixel height and weight contributed the most in predicting the price range.

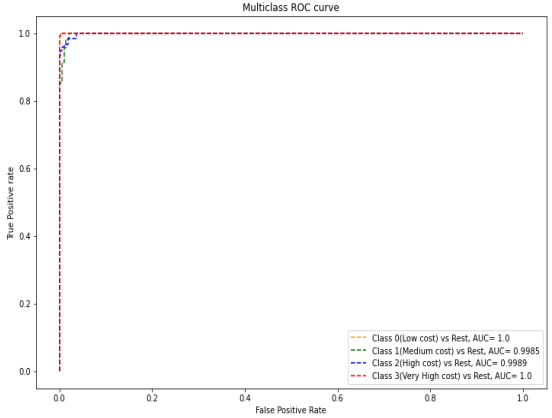
- AOC ROC Curve:

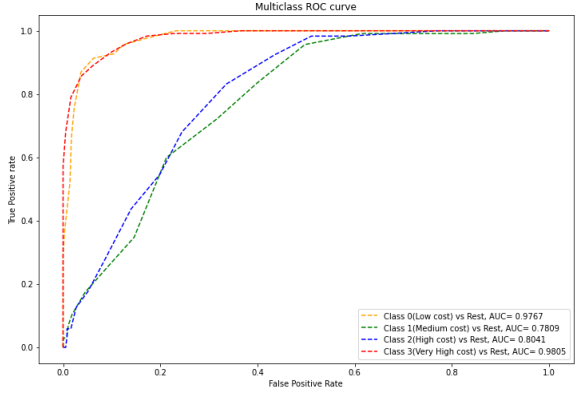












Conclusion:

- Target variable has equal number of observation in each category and target variable is nearly equally distributed.

- Percentage Distribution of Mobiles having bluetooth, dual sim, 4G,wifi and touch screen are almost 50 % and very few mobiles (23.8%) do not have 3G.

- Price range is having strong positive correlation with RAM. As we know mobiles are costly with high RAM.

- Price range has positive correlation with Battery power. Mostly high price mobiles are having great battery backup.

- **Support Vector Machine (SMV) algorithm gave best performance after hyper-parameter tunning with 98.3% train accuracy and 97% is test accuracy.**

- **XG boost is the second best good model which gave good performance after hyper-parameter tuning with 100% train accuracy and 92.25% test accuracy score.**

- **KNN gave very worst model performance.**

THANK YOU!