# Capstone Project - 3

## Mobile Price Range Prediction

By:- Milan Ajudiya

# Content:

- Problem Statement
- EDA and Feature engineering
- Feature selection
- Apply model
- Model validation and selection
- Hyperparameter Tuning
- Conclusion

# Problem Statement:

Predict the price range of the mobile based on the features ,description of the target variable class :
This is the target variable with value of:

0 ➔ low cost mobile
1 ➔ medium cost mobile
2 ➔ high cost mobile
3 ➔ very high cost mobile

mobile phone come in all sorts of price, features, specification etc. Price estimation and prediction is an important part of customer strategy. Deciding on the correct price of a product is very important for the market success of product.

# Data summary:

**Independent Features:**

**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

# Data summary(continue...)

**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
**Three_g**- Has 3G or not
**Touch_screen** - Has touch screen or not
**Wifi** - Has Wi-Fi or not

# Data summary(continue...)

**Target Variable:**

**Price_range** - This is the target variable with value of:

0 ➔ low cost mobile

1 ➔ medium cost mobile

2 ➔ high cost mobile

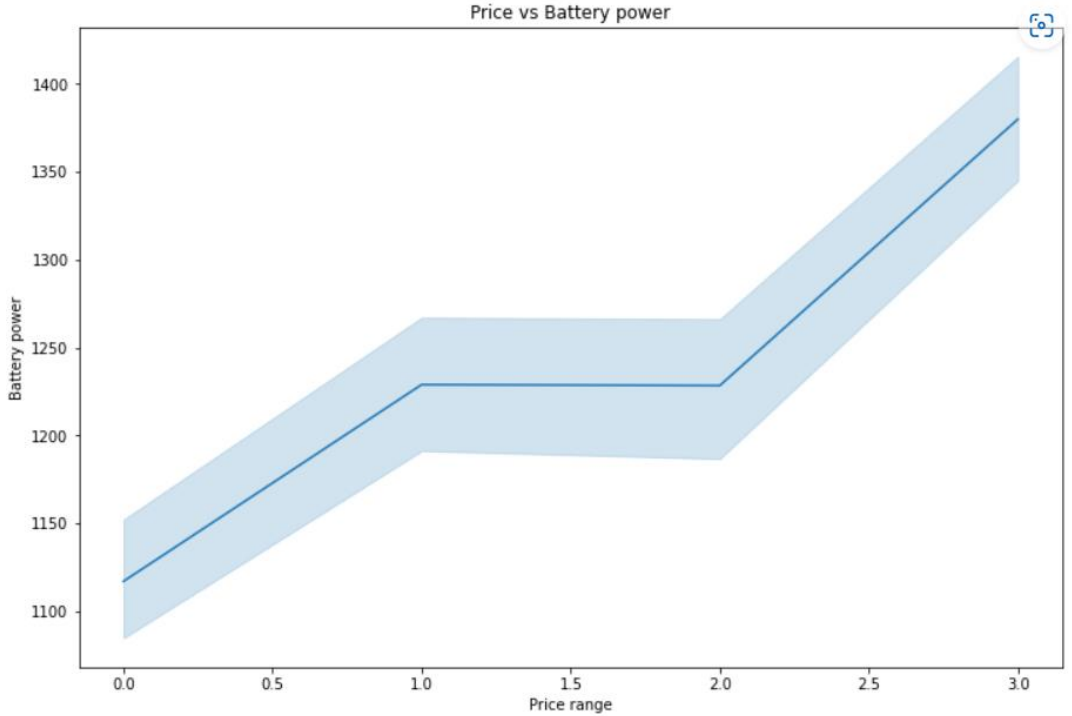
3 ➔ very high cost mobile

# Exploratory Data Analysis:

## Relation Between Price Range and Battery power:

From line graph we can see that increase in battery power capacity price is also increase.

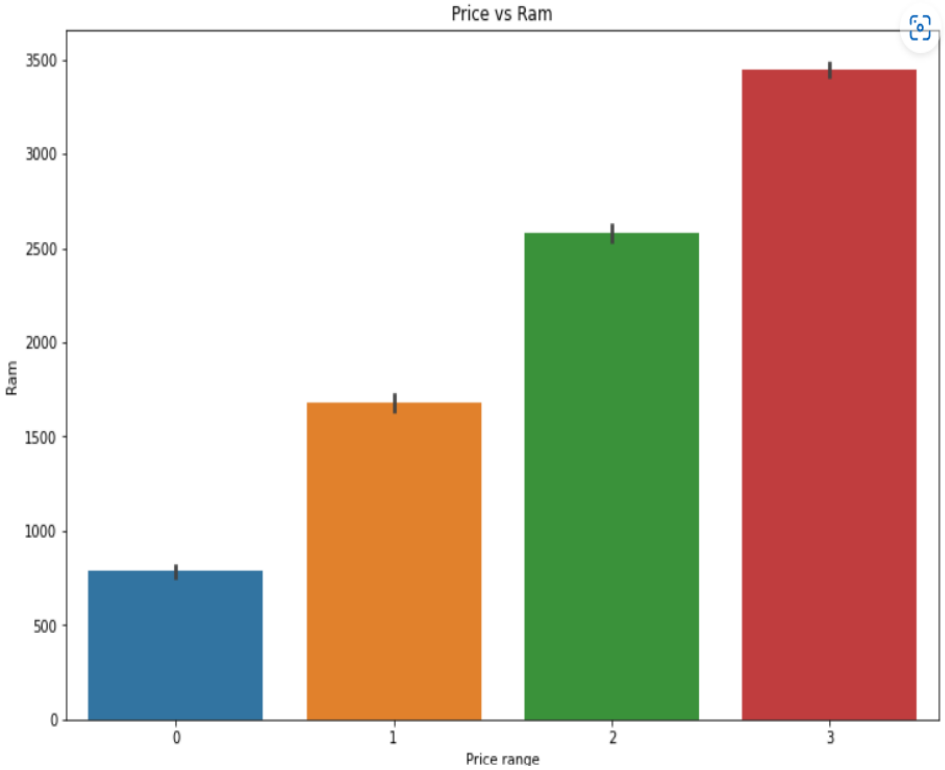
There are 4 types of price range
0 means low cost , 1 means medium cost , 2 means high cost and , 3 means very high cost of mobile.



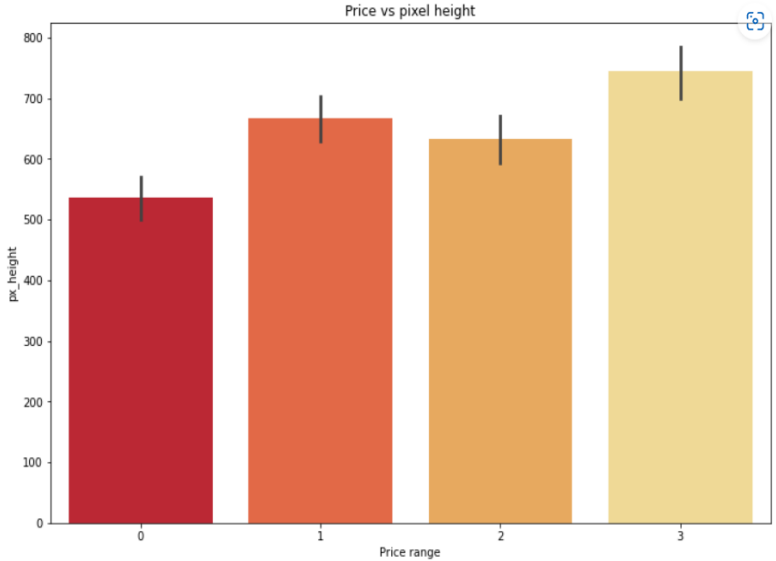
# Relation between Price Range and ram:

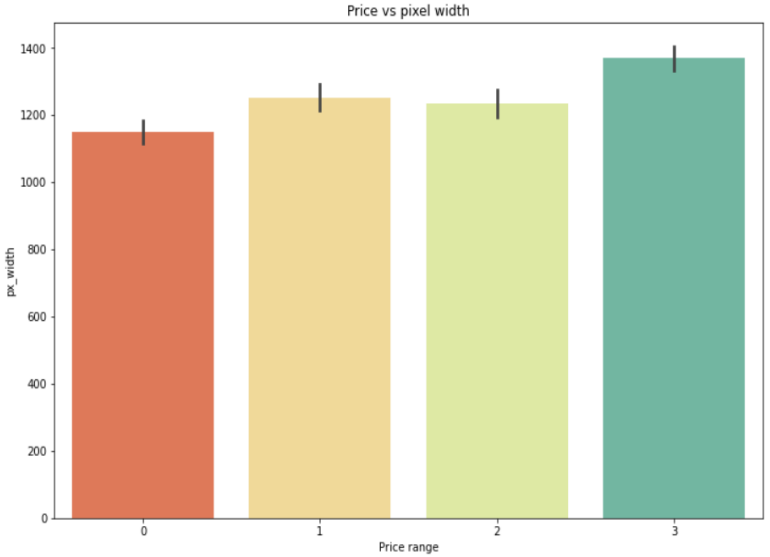
From bar graph we can see that increase in size of ram price is also increase.

Ram is very important feature in increase in price because the cost for building ram is high so it is obvious that as size of ram increase the price also increase because these both are directly proportional.



# Relation between Price Range with Pixel height / width:

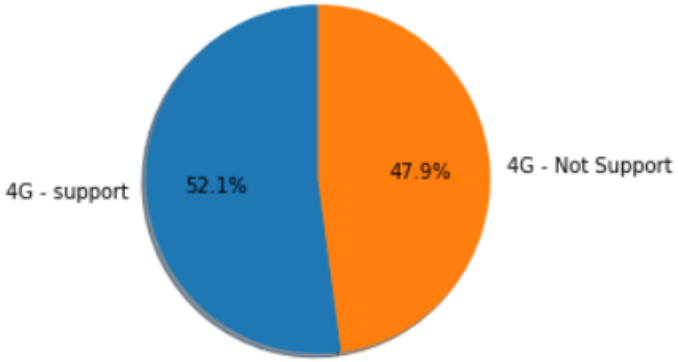


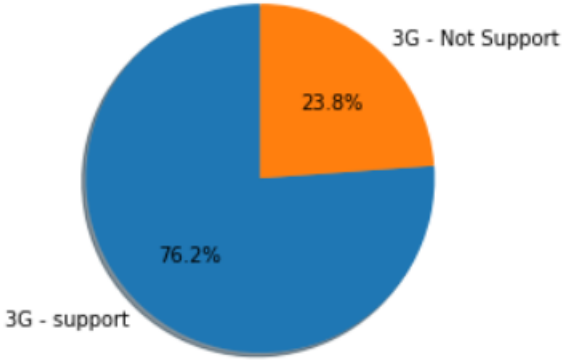




As see in bar graph, we can see that average pixel width and height are high for price range 3 (very high cost).

Low cost mobile having smaller pixel width and height.

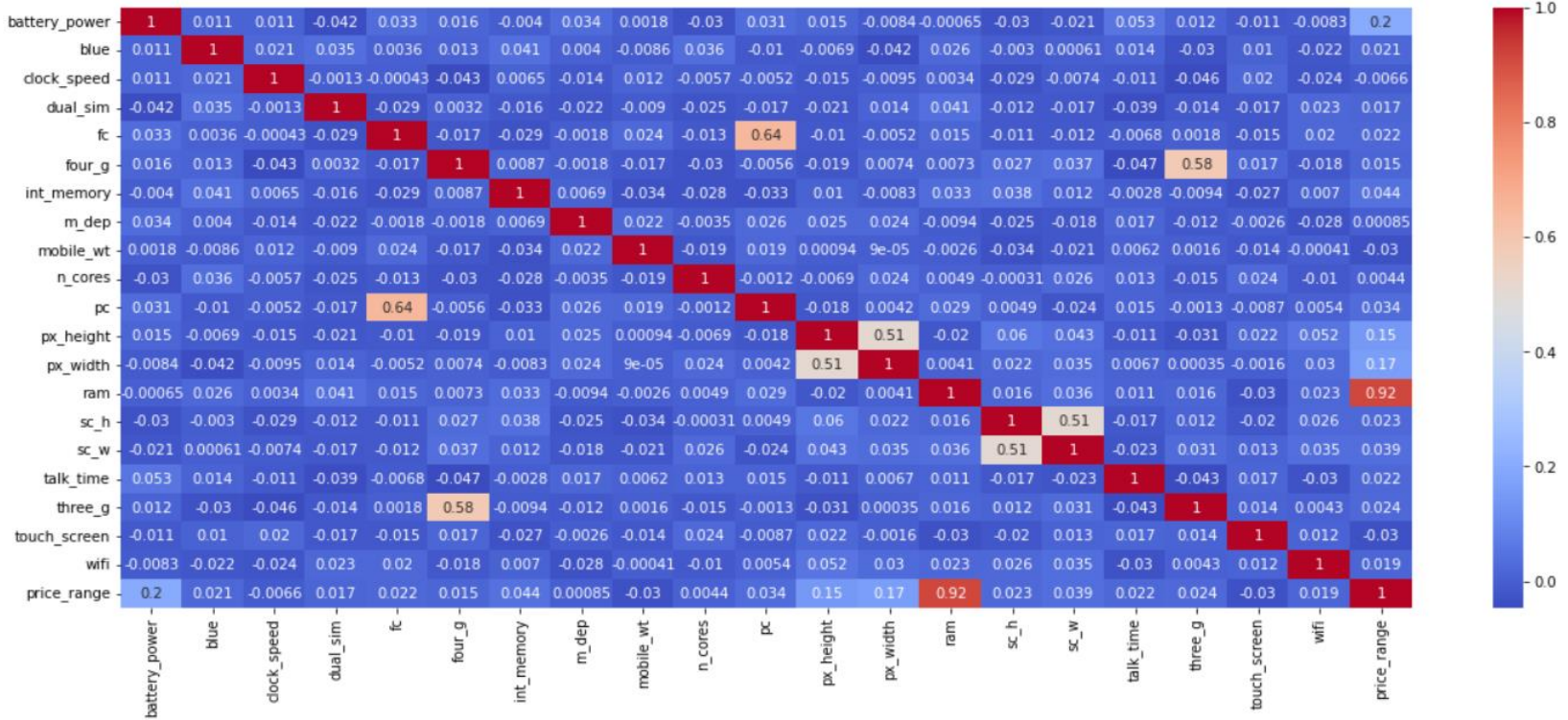
# Mobile is 3G or 4G Supported:




From pie chart 52.1% mobile are 4G supported, and 47.9% mobile are not supported 4G.
While 76.2% of mobiles are 3G supported and only 23.8% mobiles are not supported 3G.

# Correlation Heatmap:



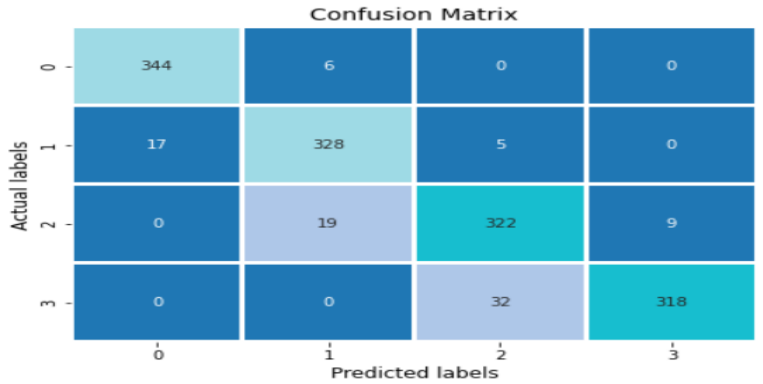
# Correlation Heatmap(continue...)

- Ram is positive and highly correlated with target variable price it is the most important feature.
- Battery power is also has positive correlation with price range , it is the second most important feature for price range.
- three_g is positive and high correlated with four_g variable.
- Pc(Primary camera) positive and has high correlation with fc(Front camera).
- px_width and px_hight has high correlation.it is also has positive correlation with target variable price range.

# Implanting K Neighbors Classifier:

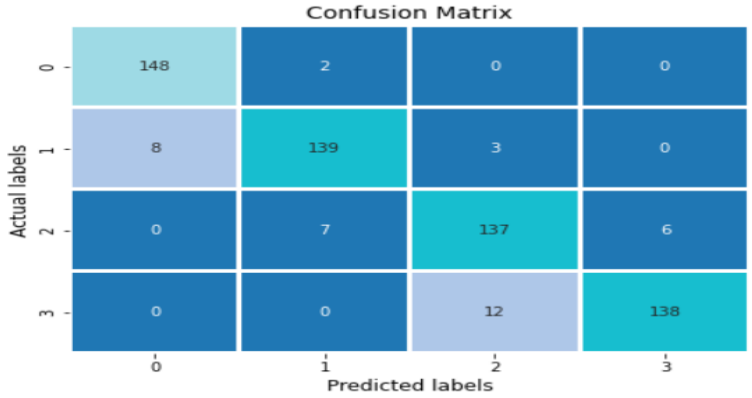
## Train Metrics:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.98 | 0.97 | 350 |
| 1 | 0.93 | 0.94 | 0.93 | 350 |
| 2 | 0.90 | 0.92 | 0.91 | 350 |
| 3 | 0.97 | 0.91 | 0.94 | 350 |
| accuracy | | | 0.94 | 1400 |
| macro avg | 0.94 | 0.94 | 0.94 | 1400 |
| weighted avg | 0.94 | 0.94 | 0.94 | 1400 |




## Test Metrics:

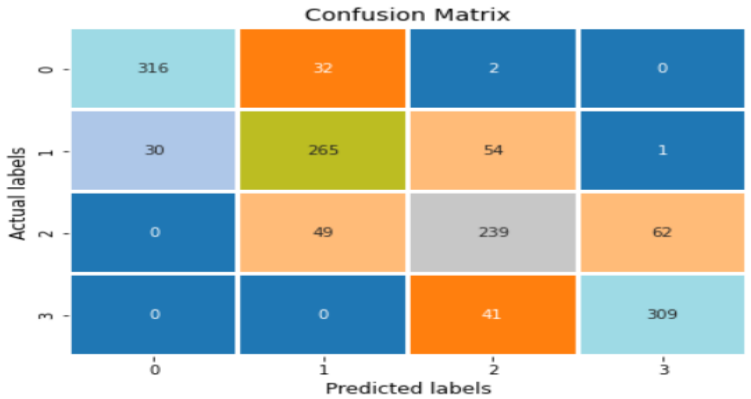
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.99 | 0.97 | 150 |
| 1 | 0.94 | 0.93 | 0.93 | 150 |
| 2 | 0.90 | 0.91 | 0.91 | 150 |
| 3 | 0.96 | 0.92 | 0.94 | 150 |
| accuracy | | | 0.94 | 600 |
| macro avg | 0.94 | 0.94 | 0.94 | 600 |
| weighted avg | 0.94 | 0.94 | 0.94 | 600 |



# Implanting Logistic Regressor:

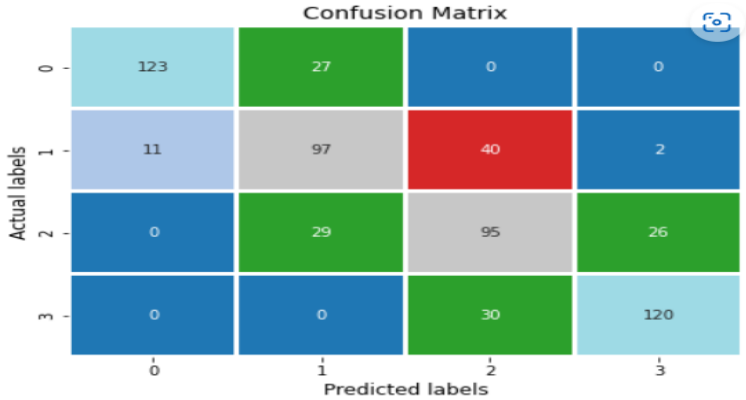
## Train Metrics:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.91 | 0.90 | 0.91 | 350 |
| 1 | 0.77 | 0.76 | 0.76 | 350 |
| 2 | 0.71 | 0.68 | 0.70 | 350 |
| 3 | 0.83 | 0.88 | 0.86 | 350 |
| accuracy | | | 0.81 | 1400 |
| macro avg | 0.81 | 0.81 | 0.81 | 1400 |
| weighted avg | 0.81 | 0.81 | 0.81 | 1400 |




## Test Metrics:

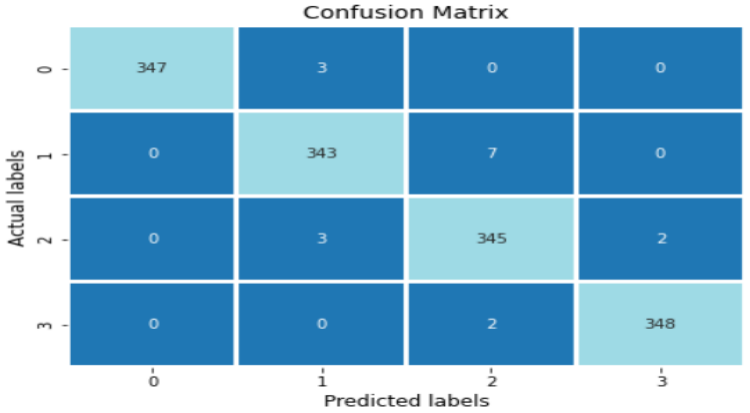
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.82 | 0.87 | 150 |
| 1 | 0.63 | 0.65 | 0.64 | 150 |
| 2 | 0.58 | 0.63 | 0.60 | 150 |
| 3 | 0.81 | 0.80 | 0.81 | 150 |
| accuracy | | | 0.73 | 600 |
| macro avg | 0.73 | 0.72 | 0.73 | 600 |
| weighted avg | 0.73 | 0.72 | 0.73 | 600 |



# Implanting Gradient Boosting Classifier:

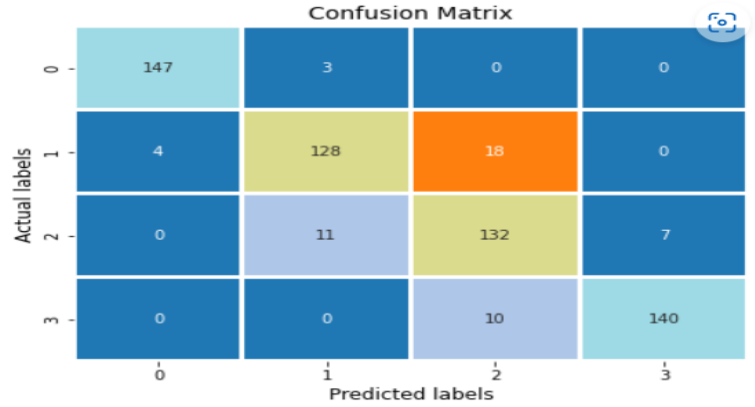
## Train Metrics:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.99 | 1.00 | 350 |
| 1 | 0.98 | 0.98 | 0.98 | 350 |
| 2 | 0.97 | 0.99 | 0.98 | 350 |
| 3 | 0.99 | 0.99 | 0.99 | 350 |
| accuracy | | | 0.99 | 1400 |
| macro avg | 0.99 | 0.99 | 0.99 | 1400 |
| weighted avg | 0.99 | 0.99 | 0.99 | 1400 |




## Test Metrics:

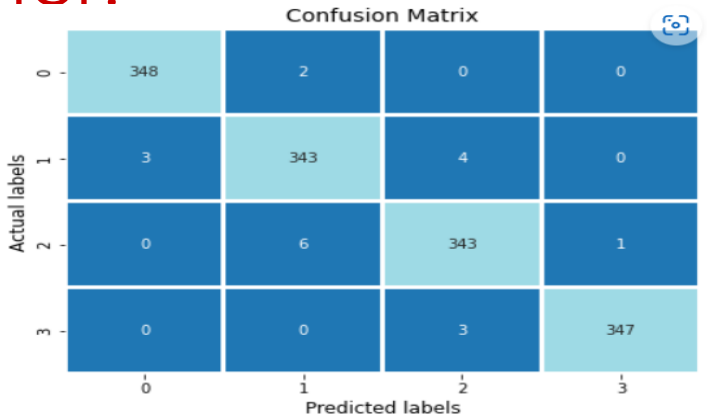
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.98 | 0.98 | 150 |
| 1 | 0.90 | 0.85 | 0.88 | 150 |
| 2 | 0.82 | 0.88 | 0.85 | 150 |
| 3 | 0.95 | 0.93 | 0.94 | 150 |
| accuracy | | | 0.91 | 600 |
| macro avg | 0.91 | 0.91 | 0.91 | 600 |
| weighted avg | 0.91 | 0.91 | 0.91 | 600 |



# Implanting XGBoost Classifier:

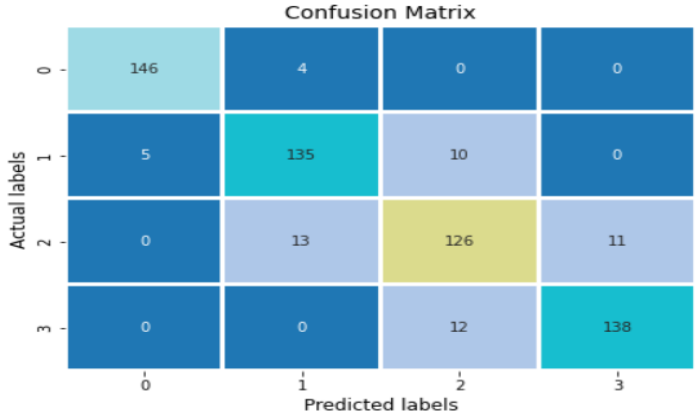
## Train Metrics:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.99 | 0.99 | 350 |
| 1 | 0.98 | 0.98 | 0.98 | 350 |
| 2 | 0.98 | 0.98 | 0.98 | 350 |
| 3 | 1.00 | 0.99 | 0.99 | 350 |
| accuracy | | | 0.99 | 1400 |
| macro avg | 0.99 | 0.99 | 0.99 | 1400 |
| weighted avg | 0.99 | 0.99 | 0.99 | 1400 |




## Test Metrics:

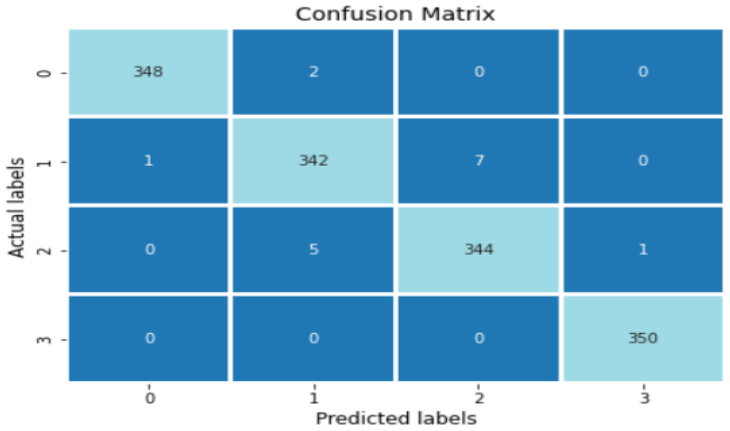
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.97 | 0.97 | 150 |
| 1 | 0.89 | 0.90 | 0.89 | 150 |
| 2 | 0.85 | 0.84 | 0.85 | 150 |
| 3 | 0.93 | 0.92 | 0.92 | 150 |
| accuracy | | | 0.91 | 600 |
| macro avg | 0.91 | 0.91 | 0.91 | 600 |
| weighted avg | 0.91 | 0.91 | 0.91 | 600 |



# Implenting Random Forest Classifier:

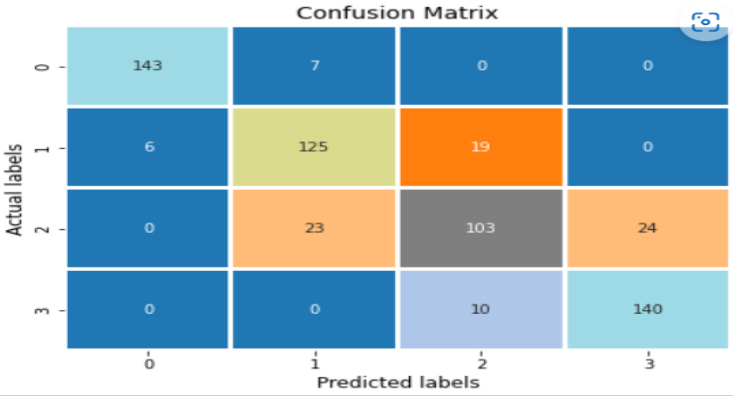
## Train Metrics:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.99 | 1.00 | 350 |
| 1 | 0.98 | 0.98 | 0.98 | 350 |
| 2 | 0.98 | 0.98 | 0.98 | 350 |
| 3 | 1.00 | 1.00 | 1.00 | 350 |
| accuracy | | | 0.99 | 1400 |
| macro avg | 0.99 | 0.99 | 0.99 | 1400 |
| weighted avg | 0.99 | 0.99 | 0.99 | 1400 |




## Test Metrics:

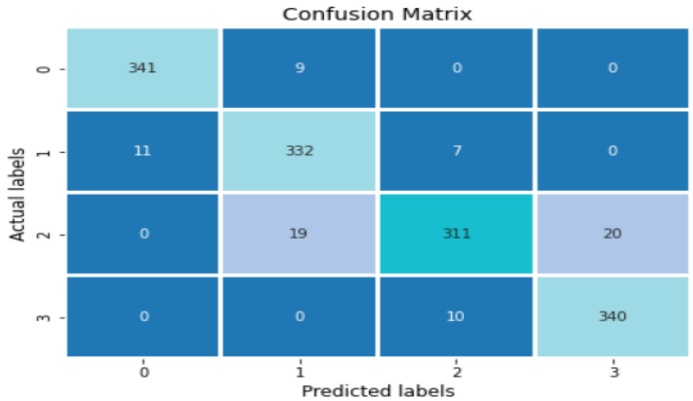
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.95 | 0.96 | 150 |
| 1 | 0.81 | 0.83 | 0.82 | 150 |
| 2 | 0.78 | 0.69 | 0.73 | 150 |
| 3 | 0.85 | 0.93 | 0.89 | 150 |
| accuracy | | | 0.85 | 600 |
| macro avg | 0.85 | 0.85 | 0.85 | 600 |
| weighted avg | 0.85 | 0.85 | 0.85 | 600 |



# Implenting Support Vector Classifier:

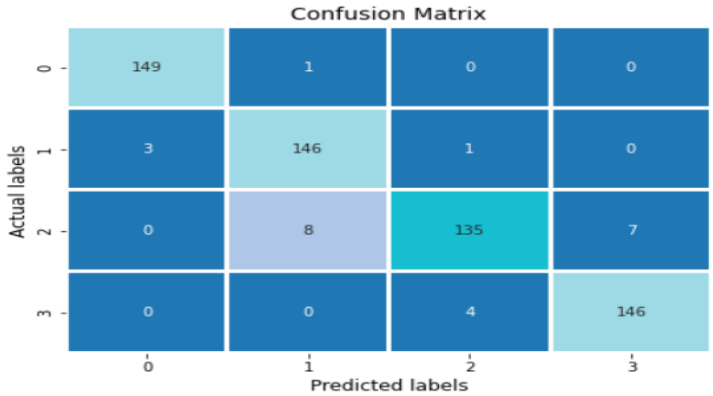
## Train Metrics:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.97 | 0.97 | 350 |
| 1 | 0.92 | 0.95 | 0.94 | 350 |
| 2 | 0.95 | 0.89 | 0.92 | 350 |
| 3 | 0.94 | 0.97 | 0.96 | 350 |
| accuracy | | | 0.95 | 1400 |
| macro avg | 0.95 | 0.95 | 0.95 | 1400 |
| weighted avg | 0.95 | 0.95 | 0.95 | 1400 |




## Test Metrics:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.99 | 0.99 | 150 |
| 1 | 0.94 | 0.97 | 0.96 | 150 |
| 2 | 0.96 | 0.90 | 0.93 | 150 |
| 3 | 0.95 | 0.97 | 0.96 | 150 |
| accuracy | | | 0.96 | 600 |
| macro avg | 0.96 | 0.96 | 0.96 | 600 |
| weighted avg | 0.96 | 0.96 | 0.96 | 600 |



# Model validation and selection:

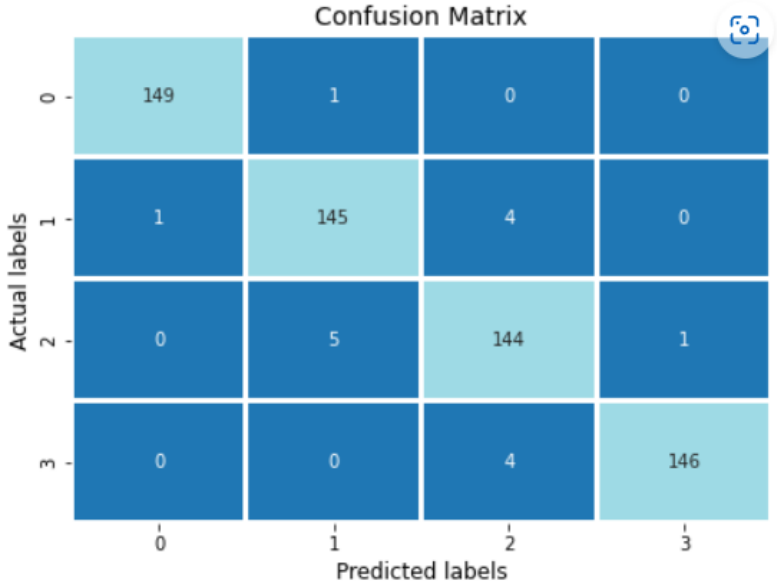
- From the above observation Random forest classifier is not giving better result, Gradient boosting classifier is giving better result in recall and precision than Random forest Classifier.
- Gradient boosting and XGBoost are giving the almost same result.
- Logistic Regression is giving low result among the all algorithm.
- K Neighbors is giving the good result than Random forest ,Gradient Boosting, and XGBoost.
- Support Vector Machine Classifier is giving best performance for this dataset.

# Hyperparameter tuning:

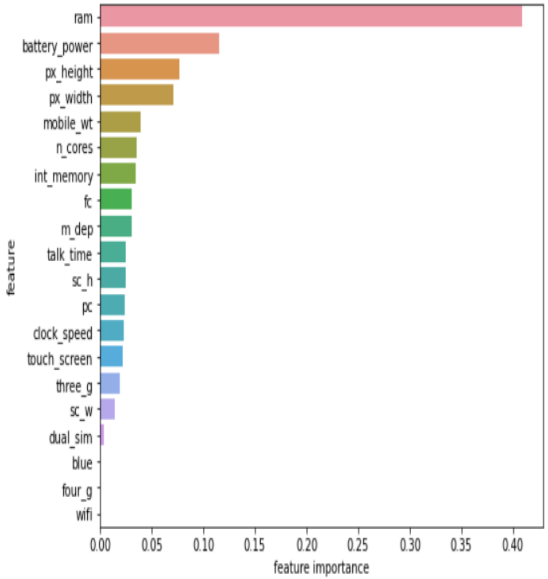
From above Observation I choose the Support Vector Machine Classifier for Hyperparameter tuning and the result is :

Classification Report:

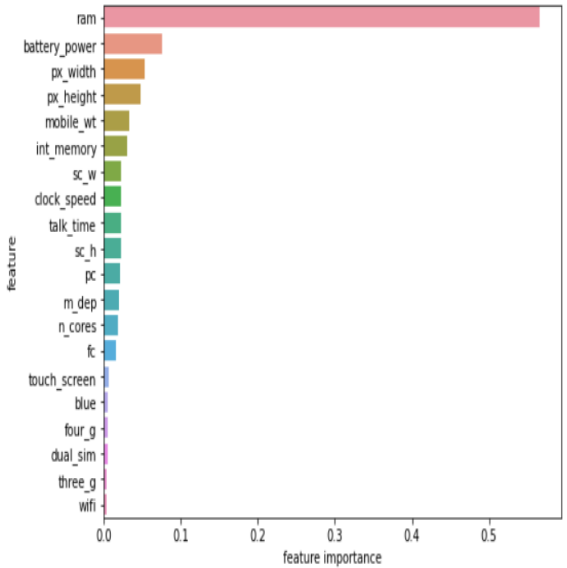
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.99 | 0.99 | 150 |
| 1 | 0.96 | 0.97 | 0.96 | 150 |
| 2 | 0.95 | 0.96 | 0.95 | 150 |
| 3 | 0.99 | 0.97 | 0.98 | 150 |
| accuracy | | | 0.97 | 600 |
| macro avg | 0.97 | 0.97 | 0.97 | 600 |
| weighted avg | 0.97 | 0.97 | 0.97 | 600 |



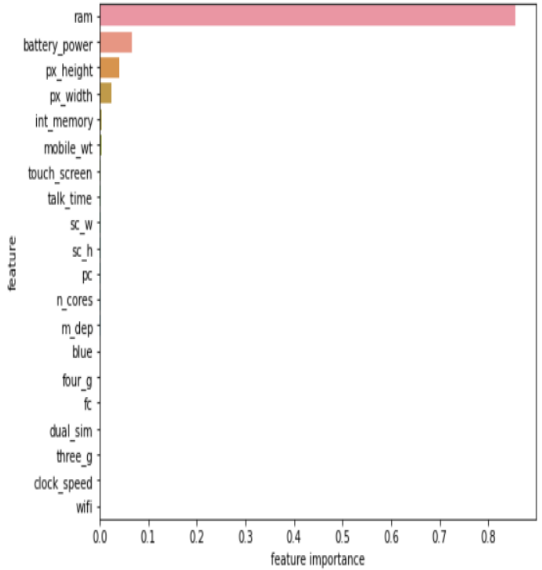
# Feature Importance:




XGBoost Classifier




Random Forest Classifier




Gradient Boosting Classifier

# Conclusion:

- Ram is the most relevant feature for predicting the price, from feature importance we can see than ram, battery_poer, px_width, px_height are the more relevant to predicting price of mobile.
- Kneighbours gives the accuracy score of 93% for train and test, and logistic regression gives the accuracy score 80% for train and 70% for test data.
- Logistic regression gives the less accuracy among the all algorithms.
- Support vector machine classifier gives the accuracy score of 95% for train and 96% for test data this algorithm gives the best result, so I tune the hyperparameter in svm classifier and it gives the accuracy score of 97% for test data.
- So, I conclude that the svm classifier gives the best result for the dataset.
- so, we can say that if the ram size and battery capacity is increase the price of mobile is also increases.

# Thank you