Mobile Price Range Prediction

Data Science Pro
AlmaBetter, Bangalore.

Tanay Tupe

MOBILE PRICE RANGE PREDICTION

For this Classification project we we are going to use " Mobile Price Range Prediction Data Set ". This data set has around "2000" observations/listings in it with 21 columns in it's raw form. This data set is a mix between categorical, numeric and Ordinal values. These observations have various features like battery capacity of mobile phones, weather the phone has Bluetooth or not, various information on RAM size, internal memory size also information on camera of a particular model it's connectivity features, it's pixel and screen size and more.

ABSTRACT

The mobile industry is a subset of the telecommunications industry focused on mobile phones, phone service, and peripheral devices. The number of smartphone subscriptions worldwide surpasses six billion and is expected to grow by several hundred million in the next few years further. China, India, and United States are the countries with the highest number of smartphone users.

This industry has been steadily developing and growing, both in terms of market size and models. This enormous number of sales creates enormous amount of data, data which can be studied and used to predict sales patterns, decide justifiable price brackets and make meaning full Business Decisions, improve yearly supply, apply robust marketing strategies, improve service for users and ultimately grow business.

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 and its selling price. In this problem, Our goal is **to build a Classification ML model to predict the rank of price range** for new models to launch.

- **Battery_power**
  Total energy a battery can store in one time measured in mAh.
  (Continuous)
- **Blue**
  Has Bluetooth or not. *(Categorical)*

- **Clock_speed**
  Speed at which microprocessor executes instructions.
  (Ordinal)

- **Dual_sim**
  Has dual sim support or not. *(Categorical)*

- **Fc**
  Front Camera mega pixels. *(Ordinal)*

- **Four_g**
  Has 4G or not.
  (Categorical)

- **Int_memory**
  Internal Memory in Gigabytes. *(Ordinal)*

- **M_dep**
  Mobile Depth in cm.
  (Ordinal)

- **Mobile_wt**
  Weight of mobile phone. *(Ordinal)*

- **Sc_h**
  Screen Height of mobile in cm. *(Ordinal)*

- **Sc_w**
  Screen Width of mobile in cm. *(Ordinal)*

- **Talk_time**
  Longest time that a single battery charge will last. *(Ordinal)*

- **Three_g**
  Has 3G or not.
  (Categorical)

- **Touch_screen**
  Has touch screen or not**.**
  (Categorical)

- **WiFi**
  Has WiFi or not.
  (Categorical)

- **Mobile_wt**
  Weight of mobile phone. *(Ordinal)*

- **Mobile_wt**
  Weight of mobile phone. *(Ordinal)*

- **Mobile_wt**
  Weight of mobile phone. *(Ordinal)*

- **N_cores**
  Number of cores of processor. *(Ordinal)*

- **Pc**
  Primary Camera mega pixels. *(Ordinal)*

- **Ram**
  Random Access Memory in Mega Bytes.
  (Ordinal)

- **Px_height**
  Pixel Resolution Height. *(Ordinal)*
- **Px_width**
  Pixel Resolution Width. *(Ordinal)*

Price_range -
This is the target variable with value of 0(low cost), 1(medium cost). 2(high cost) and 3(very high cost).
(Ordinal) ***(Target Variable)***

INTRODUCTION

Cellular phones with basic facilities such as text messaging, voice calling, audio and video visualization and camera are referred to as mobile phones. Cellular phones that offer advanced computing abilities such as Wi-Fi, web browsing, third-party applications and mobile payment, solutions for information management, such as documents, emails and contacts, inbuilt GPS applications, and provides features such as voice and video calls and web access are referred to as smart phones.

Our goal here is to predict degree of price range from "Mobile Price Range Prediction" data set using EDA techniques, implementing different classification models which will present a insight of our data and be able to predict price bracket for new models. Also, we will be answering some questions and visualize few trends using Numpy, pandas, matplotlib and seaborne.

STEPS INVOLVED/MODULES

1. Importing necessary libraries, Loading and understanding data.

- Importing required packages for data manipulation.(Numpy, Pandas, matplotlib and seaborne)

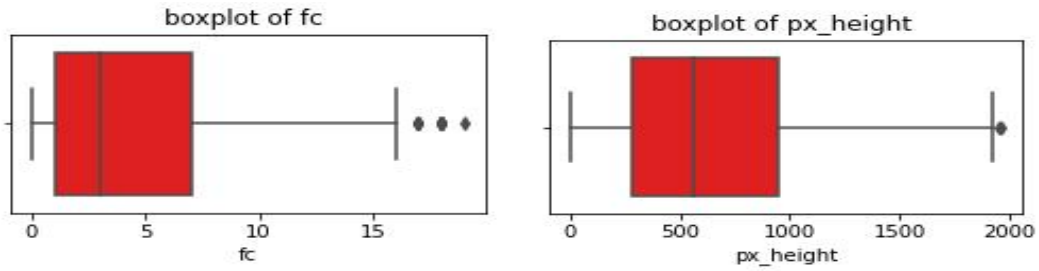
- Mounting our notebook to drive and Importing **" Mobile Price Range Prediction "** data set into notebook. For this analysis we are using google colaboratory.
- Finding shape of our data set to get number of rows and columns. In this data set we have "2000" columns and '21' rows.(From unmodified/raw data)
- Analyzing some basic information of data set (using .info function). Also, we have three different data types (int64, object, float64) occupying around '328.2 KB' of memory.
- We encounter "0" Null values In this set. Though it's highly unlikely to encounter such data sets. Here it works in our favour.
- Next, we Checked for duplicate records/observations for termination. But surprisingly, this data set had no duplicate observations.
- We also checked for number of unique values for each column in order to identify categorical features of this data set.
- Using describe function to check the min/max values, mean, standard deviation and spread of numeric columns.
- Now, we are done with cleaning, and our data is now ready for analysis.

2. Exploratory Data Analysis

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.

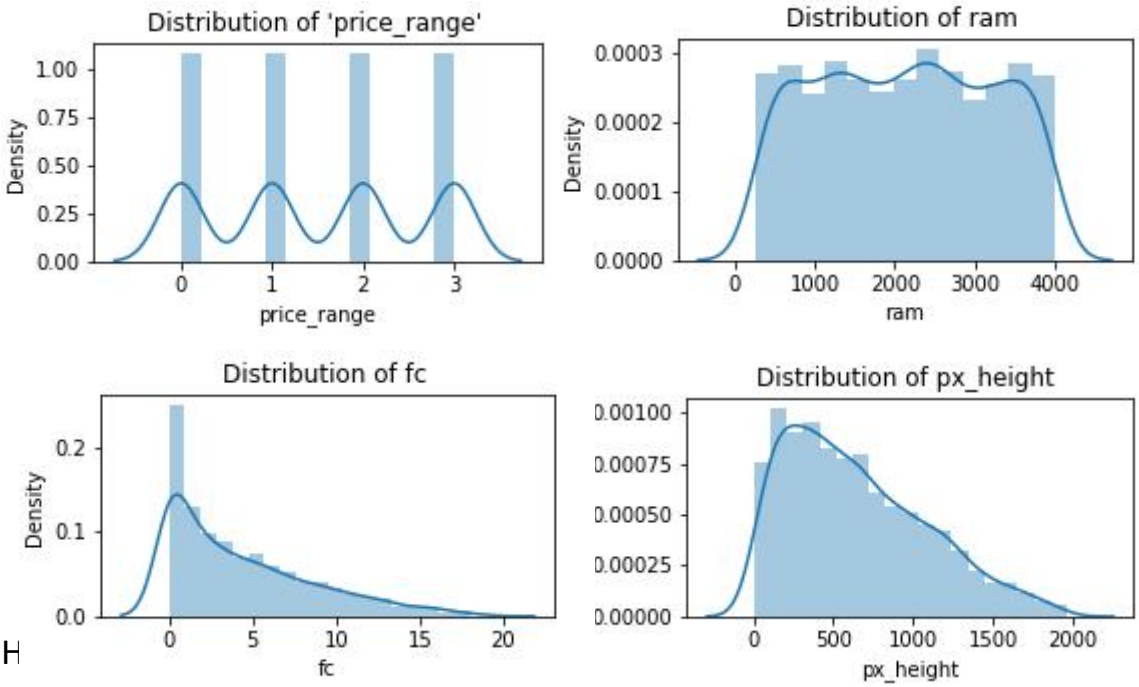
We began our EDA with univariate analysis. Where checks have been performed for type of distribution, skewness and outliers.

Considering outliers, we encountered outliers in few features like “Front Camera megapixels” and “pixel height”. Our dataset is fairly clean from outliers. We have very few of them which won't disturb our analysis.



We will be keeping them in our analysis and check how models fits. Reason we are considering outliers is because few important variables like "Front Camera megapixels" have a maximum value of about 20 Megapixels from dataset. But, mobiles in market today do have camera having 20+ Mp. It might be that those are legit observations.

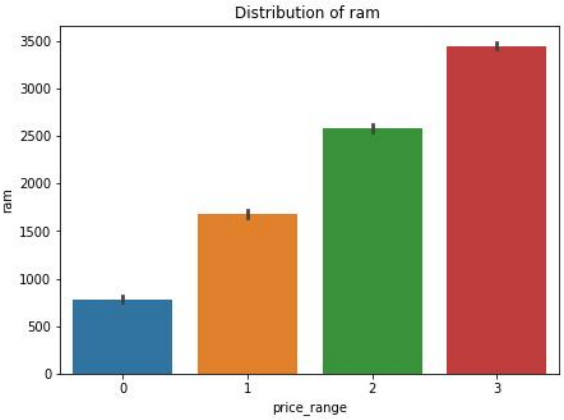
Analyzing distributions of variables.

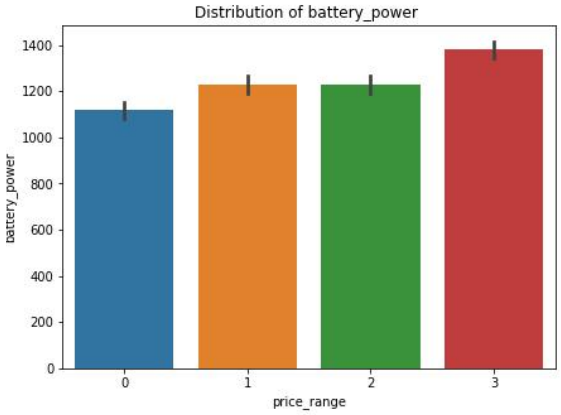


H

Most of the features have uniform distribution including our target variable. But, we have some skewed variables like “Front camera MP” and “Pixel Height”. It’s difficult to choose a algorithm in this case, but as it’s a multi-class classifier we’ll stick with KNN or Tree based models.

Bivariate analysis. Checking for relationship between dependent and independent variables.





Every feature was uniformly distributed considering our target variable. Only “RAM” and “Battery power” showed some amount of relation to our target variable. Here we can estimate that these two features will mostly affect our target variable.

3. Preparation for prediction model

Feature engineering

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In order to make machine learning work well on new tasks, it might be necessary to design and train

better features. Feature engineering facilitates the machine learning process and increases the predictive power of machine learning algorithms by creating features from raw data.

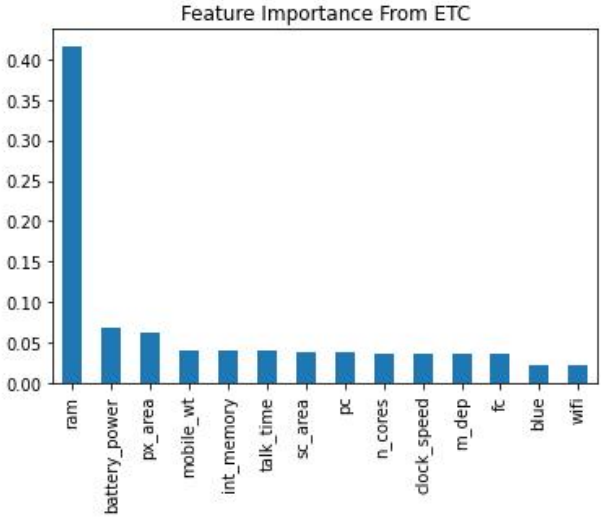
New Features created were,

* We introduced new feature **"px_area"** which will be combination of **"px_width"** and **"px_height"**. We multiplied these two to represent area of particular pixel.

* Also, **"sc_area"** was introduced, taking place of **"sc_h"** and **"sc_w"**. Here again we multiplied these two to get screen area.

Extra Tree Classifier for feature selection.

Extremely Randomized Trees Classifier(Extra Trees Classifier) is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.



As we can analyze from above, it’s difficult to choose features from this test. Therefore we will be using all features during classification.

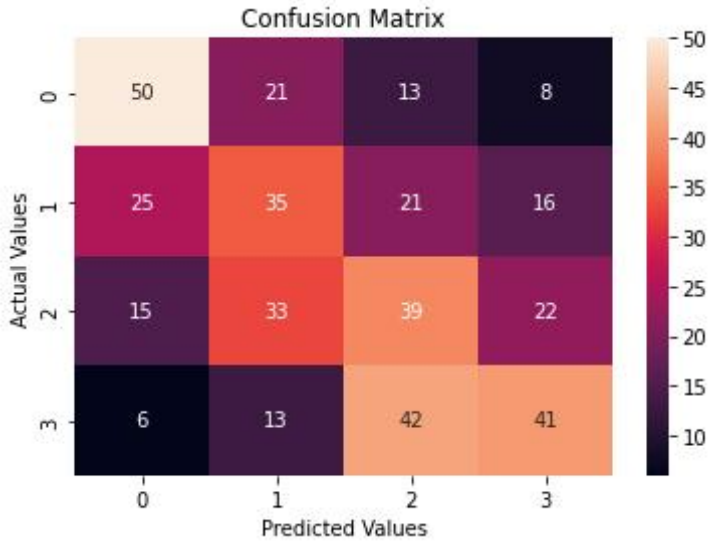
4. Implementing classification models

- **Implementing KNN Classifier.**

  K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It's a non-parametric algorithm, which means it does not make any assumption on underlying data. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

Evaluation Metrics for KNN Classifier

	precision	recall	f1-score	support
0	0.52	0.54	0.53	92
1	0.34	0.36	0.35	97
2	0.34	0.36	0.35	109
3	0.47	0.40	0.43	102
accuracy			0.41	400
macro avg	0.42	0.42	0.42	400
weighted avg	0.42	0.41	0.41	400

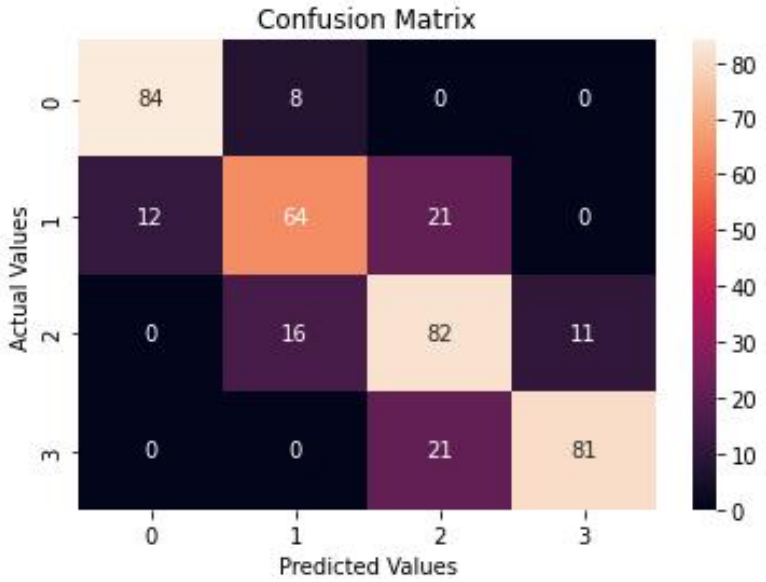


- **Implementing Decision Tree Classifier**

  Decision Tree Classifier is a structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node.

Evaluation Metrics for Decision Tree Classifier

	precision	recall	f1-score	support
0	0.88	0.91	0.89	92
1	0.73	0.66	0.69	97
2	0.66	0.75	0.70	109
3	0.88	0.79	0.84	102
accuracy			0.78	400
macro avg	0.79	0.78	0.78	400
weighted avg	0.78	0.78	0.78	400

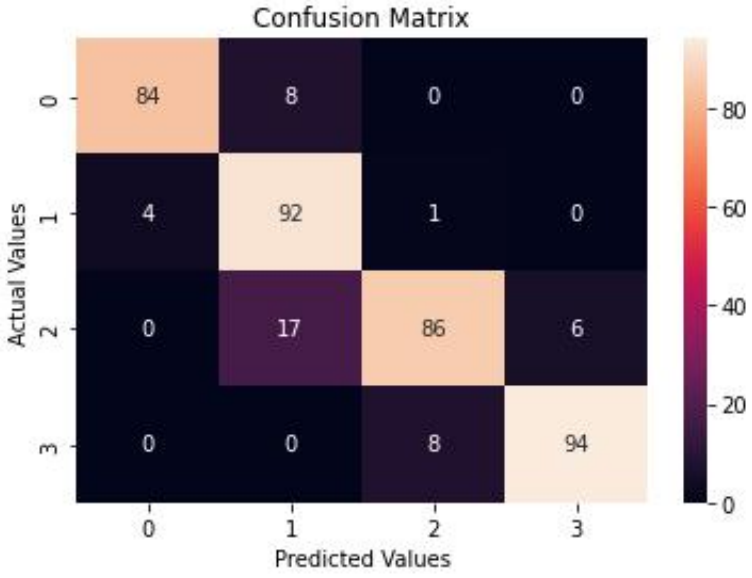


- **Implementing XGBoost Classifier**

  The XGBoost algorithm is effective for a wide range of regression and classification predictive modeling problems. It is an efficient implementation of the stochastic gradient boosting algorithm and offers a range of hyperparameters that give fine-grained control over the model training procedure.

Evaluation Metrics for XGBoost Classifier

	precision	recall	f1-score	support
0	0.95	0.91	0.93	92
1	0.79	0.95	0.86	97
2	0.91	0.79	0.84	109
3	0.94	0.92	0.93	102
accuracy			0.89	400
macro avg	0.90	0.89	0.89	400
weighted avg	0.90	0.89	0.89	400

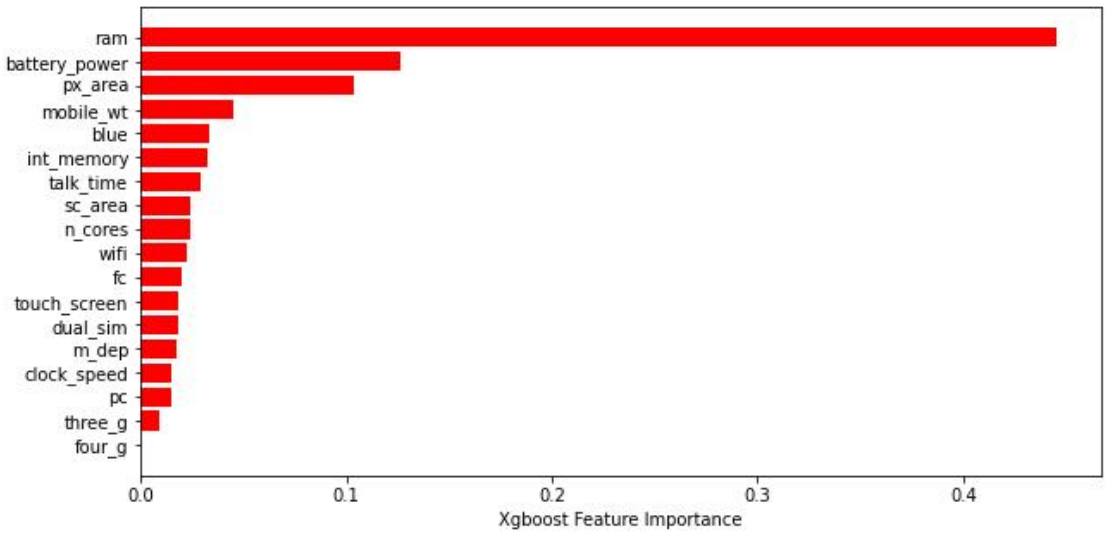


Evaluation metrics Comparison table

	KNN	Dec. T	XG
Accuracy	0.42	0.78	0.89

- **Actual Important Features by XGB Classifier**

Feature Importance refers to techniques that calculate a score for all the input features for a given model. The scores simply represent the "importance" of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable. Below we have a bar-plot representing feature importance in sorted order.



As we can see here, our top features which were important for predicting target variable are "ram", "battery power", "pixel area" and "mobile weight".

CONCLUSION

* Considering multi class classification, Logistic Classification won't fit this dataset well enough. Therefore, we used KNN, Decision Tree and XGBoost Classifier here.

* Considering outliers, our dataset is fairly clean. We have very few of them which won't disturb our analysis. We encountered outliers in features like "Front Camera megapixels" and "pixel height".

* While implementing KNN Classifier, even using best parameters and cross validation we observed,
accuracy score was - 0.41
Precision - 0.52, 0.34, 0.34, 0.47
Recall - 0.54, 0.36, 0.36, 0.40
F1 Score - 0.53, 0.35, 0.35, 0.43

which is really not acceptable. And thus we will have to reject this model.

* While implementing Decision Tree Classifier, tuning it's hyperparameters and using cross validation we observed metrics were way better than KNN. Here the model fit's well to test data and can be used to predict actual price range.
accuracy score was - 0.78
Precision - 0.88, 0.73, 0.66, 0.88
Recall - 0.91, 0.66, 0.75, 0.79
F1 Score - 0.89, 0.69, 0.70, 0.84

* And finally, Implementing XGBoost Classifier, this is where we observed the highest values of all metrics compared to Decision Tree and KNN.

accuracy score was - 0.89

Precision - 0.95, 0.79, 0.91, 0.94

Recall - 0.91, 0.95, 0.79, 0.92

F1 Score - 0.92, 0.86, 0.84, 0.93