Mobile Price Range Prediction

Chandrashekhar Awate

Data science Trainee

Almabetter, Banglore

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.**

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).**

Exploratory Data Analysis

Exploratory data analysis was done to know the insights of the data and which features have there more influence on Mobile price .

Null value Treatment

Our dataset was not having any null values and duplicates too.

Fitting different models

1)Support vector Machine
2)Logistic Regression
3)Decision Tree Classiefier
4)Naïve Bayes

Hyperparameter Tuning and regularization

Hyperparameter tuning and regularization is important for Classification problems. It help us find the optimal Classification model which is free from errors and can be used efficiently for prediction.

Algorithms

1)Support Vector Machine

The Maximal-Margin Classifier is a hypothetical classifier that best explains how SVM works in practice.

The numeric input variables (x) in your data (the columns) form an n-dimensional space. For example, if you had two input variables, this would form a two-dimensional space.

A hyperplane is a line that splits the input variable space. In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions you can visualize this as a line and let's assume that all of our input points can be completely separated by this line. For example:

B0 + (B1 * X1) + (B2 * X2) = 0

Where the coefficients (B1 and B2) that determine the slope of the line and the intercept (B0) are found by the learning algorithm, and X1 and X2 are the two input variables.

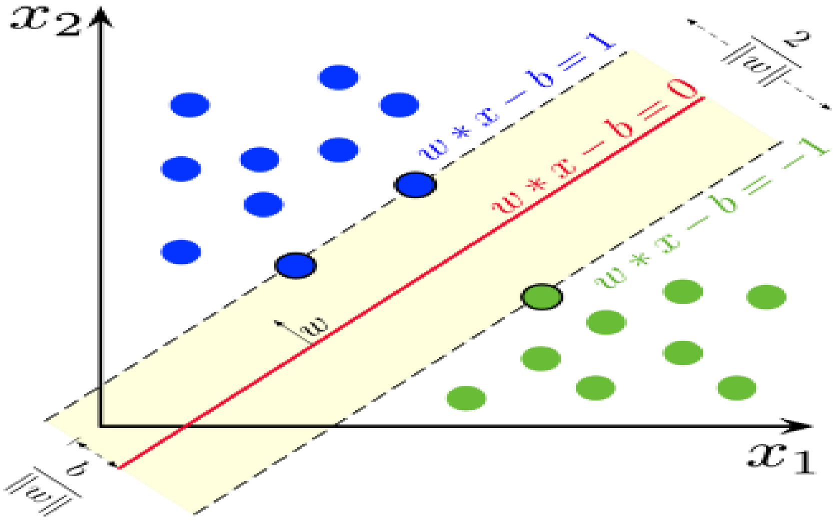
You can make classifications using this line. By plugging in input values into the line equation, you can calculate whether a new point is above or below the line.

- Above the line, the equation returns a value greater than 0 and the point belongs to the first class (class 0).
- Below the line, the equation returns a value less than 0 and the point belongs to the second class (class 1).
- A value close to the line returns a value close to zero and the point may be difficult to classify.
- If the magnitude of the value is large, the model may have more confidence in the prediction.

The distance between the line and the closest data points is referred to as the margin. The best or optimal line that can separate the two classes is the line that as the largest margin. This is called the Maximal-Margin hyperplane.

The margin is calculated as the perpendicular distance from the line to only the closest points. Only these points are relevant in defining the line and in the construction of the classifier. These points are called the support vectors. They support or define the hyperplane.

The hyperplane is learned from training data using an optimization procedure that maximizes the margin.



2)Logistic Regression

Logistic regression uses an equation as the representation, very much like linear regression.

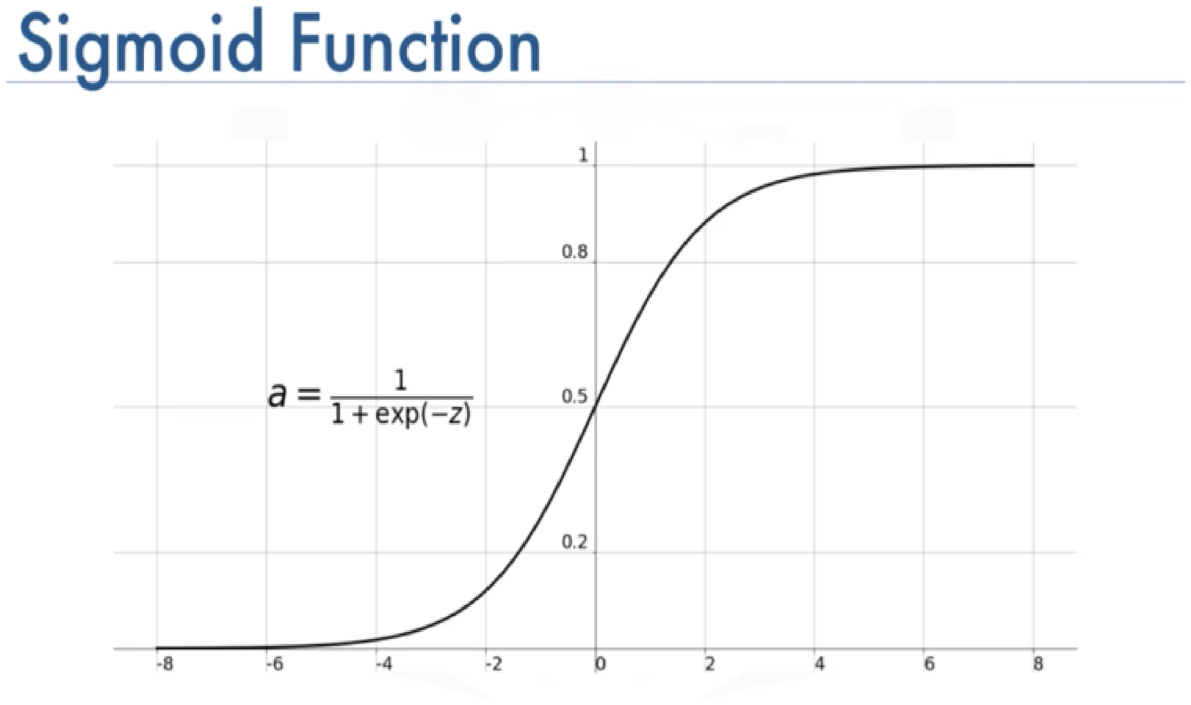
Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's).



3)Decision Tree Classifier

It is a tool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems. The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

Before learning more about decision trees let's get familiar with some of the terminologies.

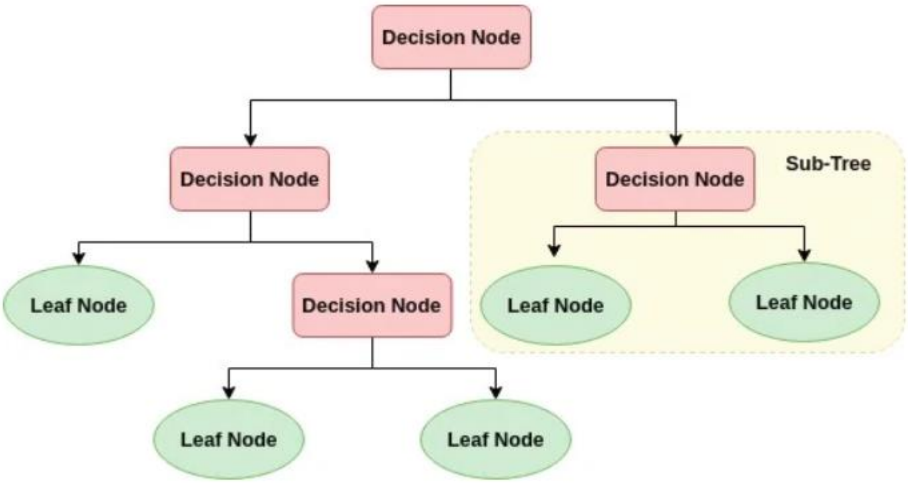
***Root Nodes* – It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.**

***Decision Nodes* – the nodes we get after splitting the root nodes are called Decision Node**

***Leaf Nodes* – the nodes where further splitting is not possible are called leaf nodes or terminal nodes**

***Sub-tree* – just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.**

***Pruning* – is nothing but cutting down some nodes to stop overfitting.**



4)Naive Bayes

Bayes theorem provides a way that we can calculate the probability of a piece of data belonging to a given class, given our prior knowledge.

Bayes’ Theorem is stated as:

- **P(class|data) = (P(data|class) * P(class)) / P(data)**

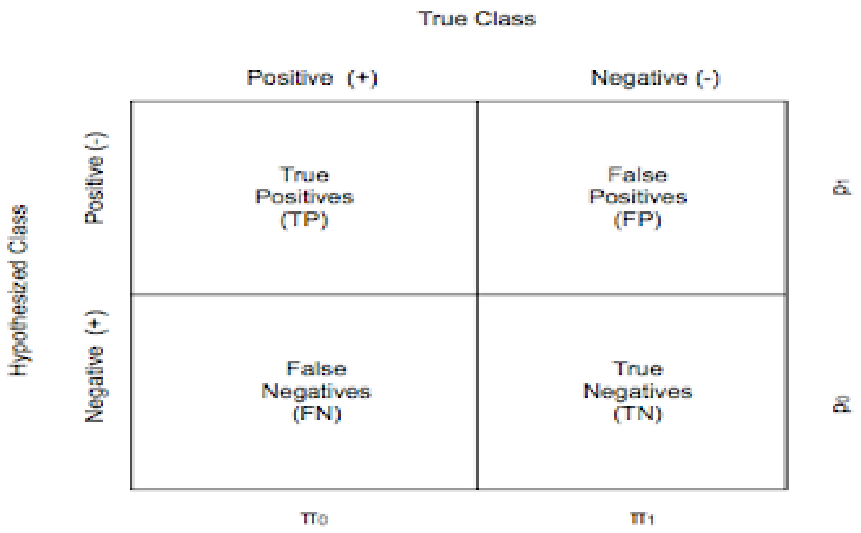
Where P(class|data) is the probability of class given the provided data.

Naive Bayes is a classification algorithm for binary (two-class) and multiclass classification problems. It is called Naive Bayes or idiot Bayes because the calculations of the probabilities for each class are simplified to make their calculations tractable.

Rather than attempting to calculate the probabilities of each attribute value, they are assumed to be conditionally independent given the class value.

This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

Model Performance and Evaluation Matrics



		True Class		
		Positive (+)	Negative (-)	
Hypothesized Class	Positive (-)	True Positives (TP)	False Positives (FP)	p_1
	Negative (+)	False Negatives (FN)	True Negatives (TN)	p_0
		π_0	π_1	

Metrics that can provide better insight are:

- **Confusion Matrix: a table showing correct predictions and types of incorrect predictions**

- **Precision: the number of true positives divided by all positive predictions. Precision is also called Positive Predictive Value. It is a measure of a classifier’s exactness. Low precision indicates a high number of false positives.**

- **Recall: the number of true positives divided by the number of positive values in the test data. The recall is also called Sensitivity or the True Positive Rate. It is a measure of a classifier’s completeness. Low recall indicates a high number of false negatives.**

- **F1 Score: the weighted average of precision and recall.**

Hyper parameter tuning:

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs. We used gridsearch cv.

Grid Search CV-

Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

Conclusions:

From Exploratory Data Analysis

1)Majorly impacting properties on price of phone are Ram, Battery power and pixel resolution. while clock speed ,mobile weight and whether it is screen touch phone or not impacting price of mobile negatively.

2) The battery power ranges between 600-2000.

3) Clock speed ranges between 0.6 to 3.

4) The random access memory of the phone ranges between 250 megabytes to 4000 megabytes.

5) The hight of screen of mobile phones ranges between 3cm to 18cm.

6) The width of screen of mobile phones ranges between 2cm to 17cm.

7) The depth or thickness of mobile ranges between something around 0.2cm to 1cm.

8) The camera megapixels ranges between 1 to 20.

9) The front camera of mobile phones ranges between 1 mp to 17mp.

10)There is no class imbalance in Target Variable.

From Models

1)The best performing model is Logistic Regression as it stands outstanding in all Four algorithms with respect to Accuracy, Precision,Recall and F-1 Score.

2)After Logistic Regression the support vector classifier is the best model according to all evaluation Scores.

3)After hyperparameter tuning on logistic and Decision Tree model no outstanding performance noticed.

4)So we will accept the both models that is Logistic Regression Model and Support Vector Machine