MOBILE PRICEPREDICTION

USING RANDOM FOREST

REGRESSION

# Introduction

* Mobile phones come in all sorts of prices, features, specifications and all.
* 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

* Mobile prices are an important reflection of the Humans and some ranges are of great interest for both buyers and sellers.
* Ask a mobile buyer to describe their dream Mobile or Branded Mobile Phones.
* So in this blog we are going to see about how the prices are segregated based on the some of the features.
* As well as the target feature prediction based on the same features.
* The data contains information regarding mobile phone features, specifications etc and their price range.
* The various features and information can be used to predict the price range of a mobile phone.

# Objective

* Mobile prices are an important reflection of the Humans and some ranges are of great interest for both buyers and sellers ask a mobile buyer to describe their dream Mobile or Branded Mobile Phones. So in this blog we are going to see about how the prices are segregated based on the some of the features. As well as the target feature prediction based on the same features
* The solution is predict the mobile price range compare with Battery power, 4G enabled, wife, Bluetooth, Ram etc.

# Attibutes

* 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 megapixels
* five\_g : Has 5G 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 megapixels
* px\_height : Pixel Resolution Height
* px\_width : Pixel Resolution Width
* ram : Random Access Memory in Megabytes
* 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
* four\_g : Has 4G or not
* touch\_screen : Has touch screen or not
* wifi : Has wifi or not
* price\_range : This is the target variable with values of 0(Poor), 1(Average), 2(Good) and 3(Excellent)

# Proposed Algorithm

**5.1 Random Forest Regression:**

* Before understanding the working of the random forest algorithm in machine learning, we must look into the ensemble technique.
* Ensemble simply means combining multiple models.
* Thus a collection of models is used to make predictions rather than an individual model.

**5.2 Steps involved in random forest algorithm:**

* Step 1: In Random forest n number of random records are taken from the data set having k number of records.
* Step 2: Individual decision trees are constructed for each sample.
* Step 3: Each decision tree will generate an output.
* Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

**5.3 Calculation**:

Decision trees learn how to best split the dataset into smaller and smaller subsets to predict the target value. The condition, or test, is represented as the “leaf” (node) and the possible outcomes as “branches” (edges). This splitting process continues until no further gain can be made or a present rule is met, e.g. the maximum depth of the tree is reached.

* Gini index 

* E(s)=p(+) log(+) –p(-) log(-)

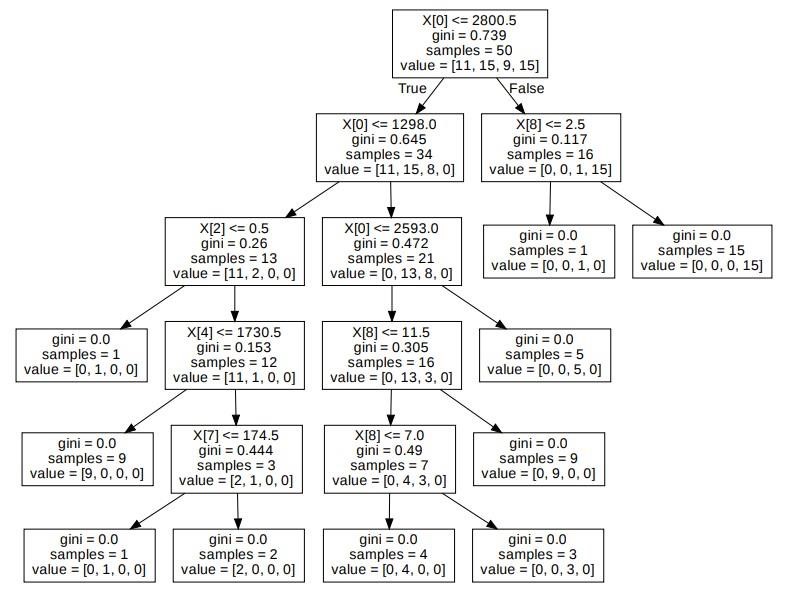


Fig 1: Mathematical calculation for Random forest method

**5.4 Advantages of Random Forest Algorithm:**

* It can be used in classification and regression problems.
* It solves the problem of overfitting as output is based on majority voting or averaging.
* It performs 1well even if the data contains null/missing values.
* Each decision tree created is independent of the other thus it shows the property of parallelization.
* It is highly stable as the average answers given by a large number of trees are taken.
* It maintains diversity as all the attributes are not considered while making each decision tree though it is not true in all cases**.**

**5.5 Disadvantages of Random Forest Algorithm:**

* Random forest is highly complex when compared to decision trees where decisions can be made by following the path of the tree.
* Training time is more compared to other models due to its complexity. Whenever it has to make a prediction each decision tree has to generate output for the given input data.

**5.6 Application of random forest:**

* This algorithm is used to forecast behaviour and outcomes in a number of sectors, including banking and finance, e-commerce, and healthcare.
* It has been increasingly employed thanks to its ease of application, adaptability, and ability to perform both classification and regression tasks.

# Input code for Mobile Prediction

From google colab

import files Uploaded=files.upload() import pandas as pd

df=pd.read\_csv('/content/train.csv')

!pip install chart\_studio import numpy as np import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns %matplotlib inline

import chart\_studio.plotly as py

from plotly.offline import init\_notebook\_mode,iplot

init\_notebook\_mode(connected=True)

import plotly.graph\_objs as go

import warnings warnings.filterwarnings("ignore")

import os

file = open('/content/train.csv') df.head(10)

|  |  |
| --- | --- |
| import seaborn as sns import matplotlib.pyplot as pd corr=data.corr() fig = plt.figure(figsize= (15,12)) r = sns.heatmap(corr, cmap='Purples') |  |
| r.set\_title("Correlation ") | |
| X = df[['ram','int\_memory','four\_g','five\_g','battery\_power','fc','pc','mobile\_wt','talk\_time']] | |

y=df['price\_range'] from sklearn.ensemble import model=RandomForestRegressor() model.fit(X,y) r\_sq = model.score(X, y) print(r\_sq) pip install sklearn pip install gradio import gradio as gr outputs = gr.outputs.Textbox()

app = gr.Interface(fn=price\_range, inputs=['number','number','number','number','number','number ','number','number','number'], outputs=outputs,description="This is a Mobile price prediction mo del")

demo = gr.Interface(fn=price\_range , inputs=[ gr.Slider(250, 8000),gr.Slider(6, 128),gr.Slider(0, 1),gr.Slider(0, 1),gr.Slider(500, 45000),gr.Slider(0, 60),gr.Slider(0, 108),gr.Slider(0, 170),gr.Slid er(2,20)], outputs=outputs,)

demo.launch()

# Output for Mobile price Prediction

## Correlation matrix

Accuracy for random forest regression: 98.44%

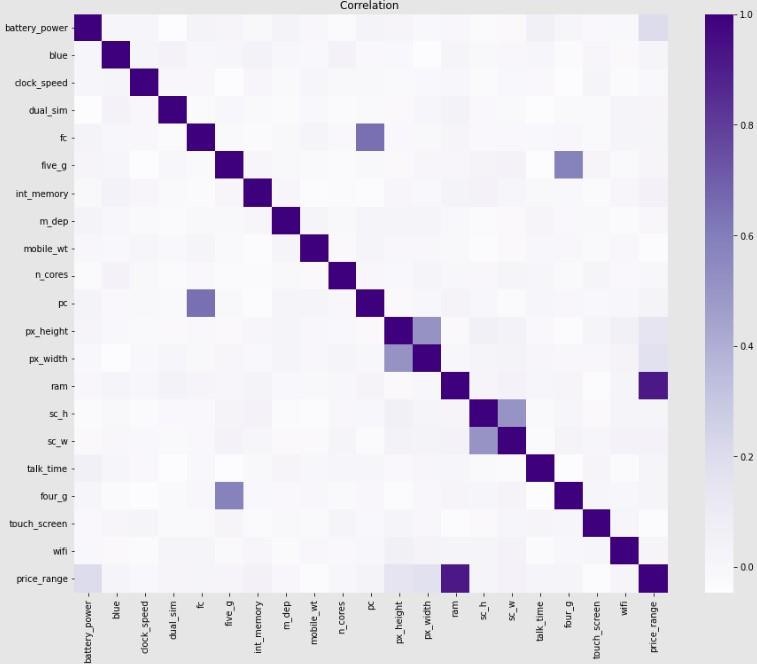


Fig 2: Correlation matrix

Correlation between battery\_power , blue,clock\_speed , dual\_sim , fc , four\_g , int\_memory , m\_dep , mobile\_w t, n\_cores , pc , px\_height , ram,sc\_h , sc\_w , talk\_time , three\_g , touch\_scre en , wifi , price\_range.

**7.1 GUI:**

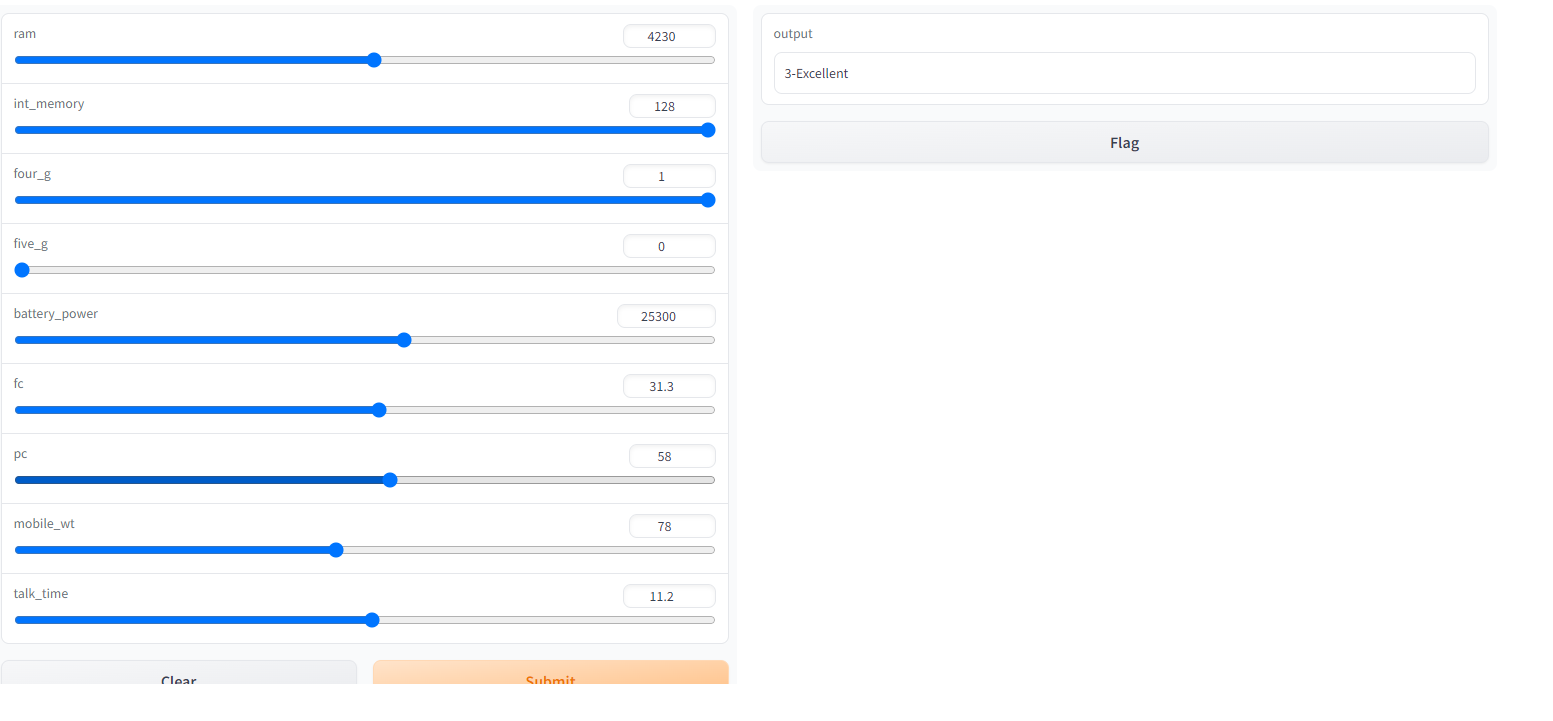


Fig 3: output using Graphical user interface (GUI)

# 8.Conclusion

We looked at classification. Classifiers represent the intersection of advanced machine theory and practical application. These algorithms are more than just a sorting mechanism for organising unlabelled data instances into distinct groupings. Classifiers include a unique set of dynamic rules that include an interpretation mechanism for dealing with ambiguous or unknown values, all of which are suited to the kind of inputs being analysed. Most classifiers also utilise probability estimates, which enable end-users to adjust data categorization using utility function

**9.Result**

We have successfully predicted the Mobile Price Prediction with the Random Forest Regression Model Using Python.