Inspiring Excellence

CSE422: Artificial Intelligence

Project Name: Mobile Price Classification

Group: 07

(Name & ID):

No.	Name	ID
1	Niaz Rahman	21301239
2	Zarin Tasnim Raisa	21301022

Section: 09

Submitted to

Rafiad Sadat Shahir , Sumaiya Aakter

Lecturer

BRAC University

Introduction

We have made a Mobile Price Prediction using various ML Calculations. This project will classify the mobile price run from 0 to 3. We will examine the cost extend in the dataset. It is a classification issue. I have prepared a mobile price classification using 3 ML calculations. It will show the extent of the mobile based on unique parameters such as front camera touch screen, center, battery, clock speed, internal memory, battery capacity etc. I have prepared the show using 3 calculations and compared all the models using the chart.

Methodology

Our mobile price prediction project makes use of many Python libraries, such as matplotlib, pandas, and sklearn. We've used pandas to get the CSV file. The dataset was subsequently reduced to the perfect size for feeding the algorithm, and visualization was employed for improved processing. Next, the dataset was divided into train and test subsets. Using the training dataset, we have employed three training models. We have employed KNN, logistic regression, and decision trees as our models. Next, we discovered a variety of accuracy rates for various training models. Finally, we used matplotlib to show and compare the algorithm scores of the three distinct models..

Dataset description

- **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 megapixels
  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 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
three_g : Has 3G or not
touch_screen : Has touch screen or not
wifi : Has wifi or not

- **Label**
  price_range: This is the target variable with values of 0(low cost), 1(medium cost), 2(high cost), and 3(very high cost).

Data preprocessing

First, we used data_train.describe() to examine the dataset, and then we used data_train.shape() to verify the rows and columns. Additionally, we checked the dataset for null values using data_train.isnull().sum() and discovered none. Next, we attempted to use some of the labeled characteristics in the dataset to display the data. For instance, we used Matplot Plotting to compare them. We looked at the dataset's correlation, but we were unable to detect any significant association. Every functionality is required. Additionally, we examined the dataset for outliers, and in px_height and fc, we discovered a little outlier. Then separate the features and save it in the X variable and label in the Y variable. We have used the split train_test_split() function we splitted 25% of the data in test and 75% in train. We have checked x_train and y_train rows are equal or not using .shape(). Moreover, for scaling the dataset we have used StandardScaler imported from sklearn preprocessing. Then we have scaled the x_train and x_test dataset and saved it in the X_test_std.

Model selection

- **Decision tree:**

The Decision Tree Classifier has now been imported from the sklearn package, defined, and trained using the datasets X_train and Y_train. Next, use the X_test dataset to test the model.

- **Logistic regression**

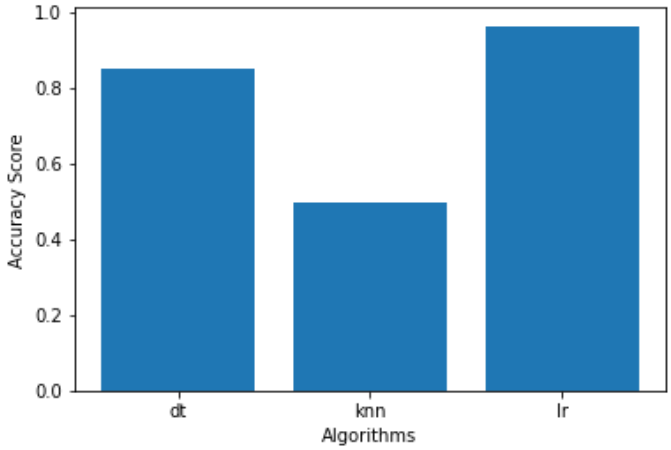
The logarithmic regression has now been loaded, created, and trained using the X_train and Y_train datasets. Next, use the X_test dataset to test the model.

- **KNN**

The KNN algorithm KNeighborsClassifier() has now been loaded, and the model has been trained using the classifier knn.fit(X_train_std,Y_train). After then, we tested the dataset using knn.predict(X_test).

Result

Model Name	Accuracy rate
Decision tree	**0.848(84%)**
Logistic regression	**0.962(96%)**
KNN	**0.496(49%)**



References

https://www.kaggle.com/datasets/iabhishekofficial/mobile-price_(SHARMA)-classification