BRAC
UNIVERSITY

Inspiring Excellence

Course: CSE449

Project Name: Mobile Price Classification(Artificial Intelligence Base)

Group:Individual
Name: MD ABRAR MAHIB PRITOM
ID: 24141155
SECTION: 01

Submitted to

Annajiat Alim Russel
Lecturer

BRAC University

Submission Date:

October 6, 2024

Introduction

We have executed a Mobile Price Prediction utilizing diverse Machine Learning Calculations. This project will classify the cost run of the mobile price. The cost ranges from 0-3. We'll examine the cost extend within the dataset. It's a classification issue. Presently I have prepared a mobile price classification utilizing 3 ML calculations. This demonstrates the extent of the mobile based on distinctive parameters like front camera, touch screen, centers, battery, clock speed, inside memory, battery capacity, etc. After preparing the show utilizing 3 calculations, I compared all the models utilizing the chart.

Methodology

In this mobile price prediction project, we have used several python libraries for example pandas,sklearn, matplotlib. To retrieve the CSV file we have used pandas. Then we shrunk the dataset to an ideal dataset for feeding the algorithm then we used visualization for better processing. Then we split the dataset into train and test datasets. We have used three training models using the training dataset. The models we have used are decision trees, KNN, and Logistic regression. Then we found several accuracy rates for different training models. In the end, we visualized and compared the three different models' algorithm scores using matplotlib.

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

We analyzed the dataset in the first place using data_train.describe() and also checked the rows and columns using data_train.shape(). We also check null values using data_train.isnull().sum() and found no null in dataset. Then we tried to visualize the data using some of the dataset's features with labels. For example, we compared them using matplot plotting. We have checked the correlation of the dataset but in this dataset, we couldn't find any major correlation. All the features are necessary. We also check the outlier of the dataset and we found a small outlier in px_height and fc. 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:**

Now we have loaded the Decision Tree Classifier from sklearn library and defined the DecisionTreeClassifier and trained with the X_train and Y_train datasets. Then test the model using the X_test dataset.

- **Logistic regression**

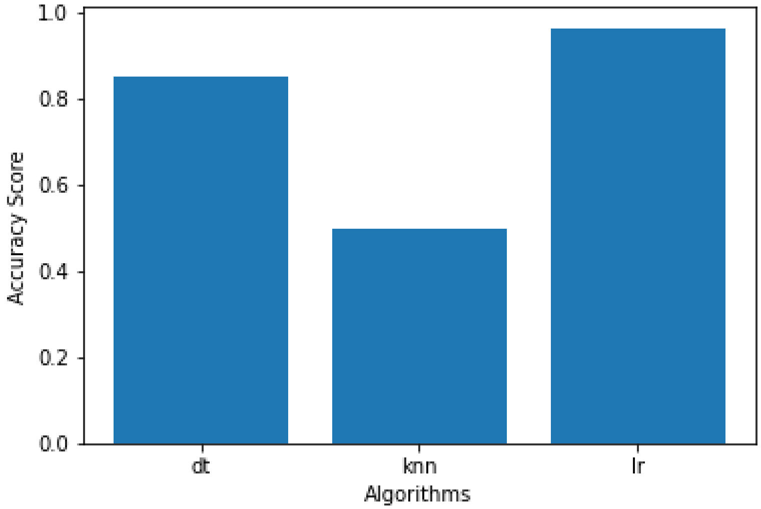
Now we have loaded the Logistic Regression and defined the LogisticRegression and train with the X_train and Y_train dataset. Then test the model using the X_test dataset.

- **KNN**

Now we have loaded the KNN algorithm KNeighborsClassifier() and used the classifier knn.fit(X_train_std,Y_train) to train the model. then we used knn.predict(X_test) to test the dataset.

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