Mobile Price Range Classification

Documentation

- By Prasad Khedkar
(AlmaBetter Student)

Abstract:

Today mobile phones are integral part of our day-to-day life, be it for communications, digital finances, booking a flight ticket or any other reasons. The features in phones make our task easy and simple. We can hardly imagine a life without a phone.

Companies give some amazing features to attract customers to buy their products. But there is one task every company must do and that is to predict the correct price of the phone and classify it, like whether it is low cost, high cost or expensive phone.

We are given such classification problem in which we have to do classification of price into 4 different categories namely,
Low cost(0), Moderate cost(1),
High cost(2) and Expensive (3).
For that dataset is given which consists of different features.
Our current dataset has 2000 rows and 21 columns(or features)

Problem Statement:

The main objective is to build a classififcation model in which we have to select correct algorithm which will classify the mobile phones having different features into 4 given categories namely,
Low cost(0), Moderate cost(1),
High cost(2) and Expensive (3).

Dataset Information

- Number of instances: 2000
- Number of attributes: 21

Features information:

The dataset contains features like:

- **Battery_power** : Battery capacity of phone 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 in Megapixels
- **Four_g** : Has 4G network or not
- **Int_mem** : Internal memory capacity of phone

- **M_dep** : Mobile Depth
- **Mobile_wt** : Weight of mobile phone
- **N_cores** : No. of cores present in phone
- **Pc** : Primary(rear) camera
- **Px_height** : Pixel resolution height
- **Px_width** : Pixel resolution in width
- **Ram** : Random access memory in MB
- **Sc_h** : Screen height
- **Sc_w** : Screen width
- **Talk_time** : Longest time that battery can withstand for a call
- **Three_g** : Has 3G network support or not
- **Wifi** : Has Wifi support or not
- **Price_range** : Target variable with value-
  0- Low cost
  1- Moderate cost
  2- High cost
  3- Expensive

Steps involved:

- **<u>Data Collection</u>**

  To proceed with the problem dealing first dataset is loaded that is given, into .csv file into a dataframe then mount the drive and load the csv file into a dataframe.

- **<u>Exploratory Data Analysis</u>**
  After loading the dataset duplicate values have to be look after. There were none. So EDA was performed by comparing variable target variable 'price_range' with other independent variables. This process helped to figure out various aspects and relationships among the target and the independent variables. It gives a better idea of which feature behaves in which manner compared to the target variable.

  ❖ **Numerical Variables:**
  - **Battery_power**
  - **Clock_speed**
  - **Fc**
  - **Int_mem**
  - **M_dep**
  - **Mobile_wt**
  - **Pc**
  - **Px_height**
  - **Px_width**
  - **Ram**
  - **Sc_h**
  - **Sc_w**
  - **Talk_time**

  ❖ **Categorical Variables:**
  - **Blue**
  - **dual_sim**
  - **Four_g**
  - **N_cores**
  - **Three_g**
  - **Wifi**
  - **Price_range**

- **Null & Duplicate values Treatment:**

  There were no null values present in our dataset.
  Neither any duplicates found in the dataset,
  Dataset is already pretty clean.

- **Data Processing**

New columns added are:

- pixel_area(Actual display)
- screen(Total display)

Columns removed are :

- 'px_height'
- 'px_width'
- 'sc_h'
- 'sc_w'

- **Data Preparation & Feature Selection**

In this features and labels have been selected those are:

Independent features :-

- *Battery_power*
- *Clock_speed*
- *Fc*
- *Int_mem*
- *M_dep*
- *Mobile_wt*
- *Pc*
- *Px_height*
- *Px_width*
- *Ram*
- *Sc_h*
- *Sc_w*
- *Talk_time*
- *Blue*
- *dual_sim*
- *Four_g*
- *N_cores*
- *Three_g*
- *Wifi*

Dependent feature :-

- *price_range*

- **Feature Scaling & Train-Test split**

Standard scaler is used to scale the data

Splitting :

Training set - 70% share
Test set – 30% share

Fitting different models

Different Models tried are:

- **Logistic Regression**
- **K-NN Classifier**
- **Random-Forest Classifier**

Out of these 3 classifiers only Logistic Regressor performed well and others either overfit on the data or underfit the data

Model performance:

Performance of Logistic Regressor which is our algorithm selected for final model implementation is:

Training score : 0.947
Test score : 0.905

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.

GridSearchCV is used for hyperparameter tuning. This also results in cross validation and in this case , the dataset is divide into 5- folds.

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 hyperparameters which were extracted : 'C' , 'max_iter' & 'random_state'

Scores after tuning :
Train score: 0.95928
Test score : 0.915

Clearly , upon tuning the hyper-parameters performance of model increased significantly.
Hyper-parameter tuning boost the final output of our classification.

Conclusion:

That's it! For this documentation. Starting with loading the data so far I have done EDA , null values treatment, encoding of categorical columns, feature selection and then model building.
After hyper parameter tuning, I have prevented overfitting and decreased errors.
Out of three models, Logistic Regression performed well and classified precisely more than 90%+ accuracy score on both train and test set & also performed very well on unseen data due to various

factors like feature selection,correct model selection,etc.

Future work:

1. We can do a dynamic classification with time series modelling considering time as co-variate and linearly dependent.
2. We can adjust the ram feature to affect the classification of mobile's price according to customer's budget.
3. Naïve Bayes can also be used but it is prone to noisy data inclusion in future.

References:

1. AlmaBetter.
2. Towards data science website
3. GeeksforGeeks
4. Analytics Vidhya