Introduction to Artificial Inteligence

Project 14

Mobile price classification

Mario Andreu
Alvaro Martin

Preliminary assumptions

Description of the task

The task aims to predict the price range of mobile phones based on their attributes. It falls under the category of regression problems, as the model needs to predict a continuous value (the price range) based on input features like battery power, camera megapixels, etc. This task is essential for consumers, retailers, and manufacturers to understand the pricing dynamics of mobile devices.

Description of the dataset

The dataset contains a variety of attributes for mobile phones along with their corresponding price ranges. Attributes may include battery power, camera quality, internal memory, etc. It's crucial to analyze the dataset to understand its size, feature distribution, and the range of prices.

Additionally, examining potential correlations between features and prices can provide insights into the data.

Preprocessing and cleaning:

This step involves handling missing values, encoding categorical variables (if any), and scaling numerical features. For instance, missing values can be imputed using methods such as mean, median, or mode imputation. Categorical variables may need to be one-hot encoded or label encoded, depending on the algorithm used. Numerical features can be scaled to ensure that they have similar ranges.

It's crucial to check if the labels (price ranges) are balanced. If there's a class imbalance, techniques such as oversampling , undersampling, or using weighted loss functions can be employed to address it. Oversampling involves creating additional samples from the minority class, whereas undersampling involves reducing the number of samples from the majority class. Class weighting assigns higher weights to the minority class during training to give it more importance. The specific technique chosen would depend on the characteristics of the dataset and the chosen modeling technique. However, the exact split ratio can vary depending on the dataset size and complexity

The dataset can be split into training, validation, and test sets. The training set is used to train the model, the validation set is used for hyperparameter tuning and model selection, and the test set is used to evaluate the final model's performance.

To evaluate the performance of the solution, metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or R-squared can be used. These metrics quantify the difference between predicted and actual prices.

Additionally, considering data splits is crucial for robust evaluation. An upper bound for performance can be estimated by referring to the best scores achieved on similar datasets, such as the highest score on Kaggle if available.

Description of the solution:

For price prediction tasks, we are going to use the methods of Linear Regression, Random Forest and Neutral Networks

Linear Regression: Linear regression assumes a linear relationship between the input attributes and the target variable (price). It seeks to fit a linear equation to the data, where the coefficients represent the weights assigned to each input attribute. The model predicts the price based on the weighted sum of the input attributes. Linear regression can be implemented using various techniques, such as ordinary least squares (OLS), gradient descent, or regularized regression (e.g., Lasso or Ridge regression).

Random Forests: Decision trees recursively split the data based on attribute values to create a predictive model. Each internal node of the tree represents a decision based on an attribute, and each leaf node represents a predicted price range. Random forests are an ensemble of decision trees, where multiple trees are trained on random subsets of the data. The final prediction is obtained by aggregating the predictions of individual trees. Decision trees and random forests can handle both numerical and categorical features without the need for extensive preprocessing.

Neural Networks: Neural networks, particularly deep learning models, can capture complex relationships between features and the target variable. They consist of interconnected layers of nodes (neurons) that perform computations on the input data. Deep learning models can have various architectures, such as feedforward neural networks, convolutional neural networks (CNNs), or recurrent neural networks (RNNs). Data preprocessing for neural networks may include feature scaling, handling categorical variables (e.g., one-hot encoding), and handling missing values.

Linear regression is a simple and interpretable method, but it assumes a linear relationship between features and may not capture complex patterns in the data. Decision trees and random forests can handle non-linear relationships and interactions between features. They also provide feature importance rankings. Neural networks are highly flexible and can learn intricate patterns in the data, but they require more computational resources and larger amounts of data for training.

Data preprocessing required for these methods:

- Linear Regression: Data preprocessing for linear regression may involve handling missing values, removing outliers, feature scaling, and encoding categorical variables (if applicable).

- Random Forests: Decision trees and random forests can handle categorical variables without extensive preprocessing. However, if there are missing values or outliers, they may need to be addressed. Feature scaling is not necessary for decision trees/random forests.
- Neural Networks: Data preprocessing for neural networks may include handling missing values, removing outliers, feature scaling (e.g., normalization or standardization), and encoding categorical variables (e.g., one-hot encoding).

It is important to experiment with different preprocessing techniques and compare their impact on the performance of each method. The specific preprocessing steps and hyperparameter settings should be determined based on the characteristics of the dataset and the chosen modeling technique.

Midterm solution

At this stage, we have implemented Principal Component Analysis (PCA) for dimensionality reduction and Linear Regression for initial predictive modeling of mobile phone prices. These methods help manage the high-dimensional dataset and ensure efficient modeling. In the next stage, we will compare the performance of this regression model with classification methods.

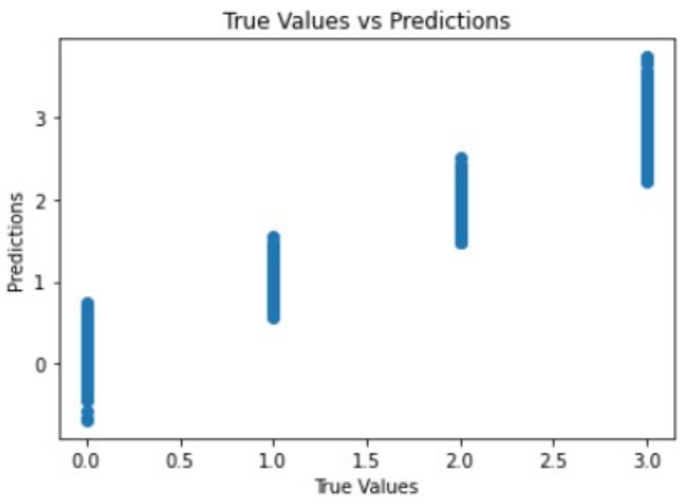
Linear regression

```
   battery_power  blue  clock_speed  dual_sim  fc  four_g  int_memory  m_dep  \
0            842     0          2.2         0   1       0           7    0.6
1           1021     1          0.5         1   0       1          53    0.7
2            563     1          0.5         1   2       1          41    0.9
3            615     1          2.5         0   0       0          10    0.8
4           1821     1          1.2         0  13       1          44    0.6

   mobile_wt  n_cores  ...  px_height  px_width   ram  sc_h  sc_w  talk_time  \
0        188        2  ...         20       756  2549     9     7         19
1        136        3  ...        905      1988  2631    17     3          7
2        145        5  ...       1263      1716  2603    11     2          9
3        131        6  ...       1216      1786  2769    16     8         11
4        141        2  ...       1208      1212  1411     8     2         15

   three_g  touch_screen  wifi  price_range
0        0             0     1            1
1        1             1     0            2
2        1             1     0            2
3        1             0     0            2
4        1             1     0            1

[5 rows x 21 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  2000 non-null   int64
 1   blue           2000 non-null   int64
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64
 4   fc             2000 non-null   int64
 5   four_g         2000 non-null   int64
 6   int_memory     2000 non-null   int64
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64
 9   n_cores        2000 non-null   int64
 10  pc             2000 non-null   int64
 11  px_height      2000 non-null   int64
 12  px_width       2000 non-null   int64
 13  ram            2000 non-null   int64
 14  sc_h           2000 non-null   int64
 15  sc_w           2000 non-null   int64
 16  talk_time      2000 non-null   int64
 17  three_g        2000 non-null   int64
 18  touch_screen   2000 non-null   int64
 19  wifi           2000 non-null   int64
 20  price_range    2000 non-null   int64
```



The first graph shows a comparison between the true values and the predictions made by the Linear Regression model. The axes of the graph represent:

- X-axis: True Values
- Y-axis: Predictions

The graph indicates that the model's predictions align with the true values. However, there is some dispersion, suggesting that the predictions are not perfect. The overall trend shows a positive correlation, meaning that as the true values increase, the predictions also increase, albeit with variations.

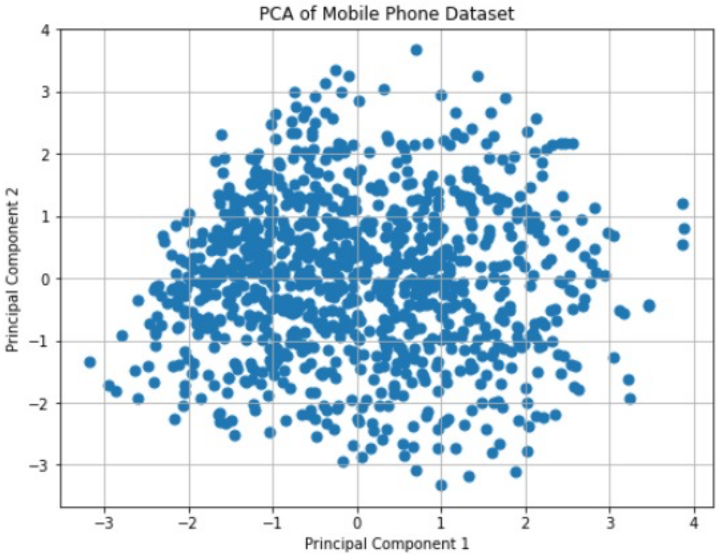
Interpretation of the results

The "True Values vs Predictions" graph shows that the model's predictions follow a similar trend to the true values, though there are variations and not all predictions are accurate.

The evaluation metrics (MSE and R^2) provide a quantitative measure of the model's performance, indicating how well the model fits the data.

PCA

[8 rows x 21 columns]



	principal_component_1	principal_component_2	id
0	1.539376	-2.434085	1
1	0.005105	0.402862	2
2	-1.075417	0.382337	3
3	3.453739	-0.413990	4
4	1.691045	-0.640134	5

The second graph shows the projection of the dataset into the space of two principal components obtained through PCA. The axes of the graph represent:

- X-axis: Principal Component 1
- Y-axis: Principal Component 2

This graph helps visualize how the high-dimensional data is distributed when reduced to two dimensions. The distribution appears fairly scattered, suggesting that the original data has high variability and that PCA has captured a significant portion of this variability.

Below the graph is a table showing the values of the first two principal components for the initial rows of the transformed dataset. These values represent the new reduced features that will be used in subsequent modeling.

<u>Challenges and Next Steps:</u>

The primary challenge encountered has been understanding the dataset, its structure, and features.

For the final stage, we want to Implement Classification Algorithms. We plan to implement classification methods such as Decision Trees or Random Forest for comparison.