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.