Introduction to Artificial Inteligence

Project 14

Mobile price classification

Mario Andreu
Alvaro Martin

INDEX

Introduction

This Mobile Price Classification project aims to develop a predictive model that estimates the price range of mobile phones based on their features, leveraging a dataset sourced from Kaggle. This project involves a comprehensive approach that includes data preprocessing, model selection, implementation, and evaluation. Initially, we will clean and analyze the dataset to ensure its suitability for modeling. We will then implement a Linear Regression model alongside Classification algorithms to predict the price range of mobile phones based on various attributes such as battery power, camera megapixels, and RAM. The performance of these models will be evaluated using metrics like Mean Squared Error (MSE) and R-squared (R2), with results visualized to illustrate the accuracy of the predictions. This project not only aims to provide insights into the factors influencing mobile phone prices but also serves as a practical application of predictive modeling techniques to assist consumers in making informed purchasing decisions.

Description of the dataset and algorithms

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 results

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.

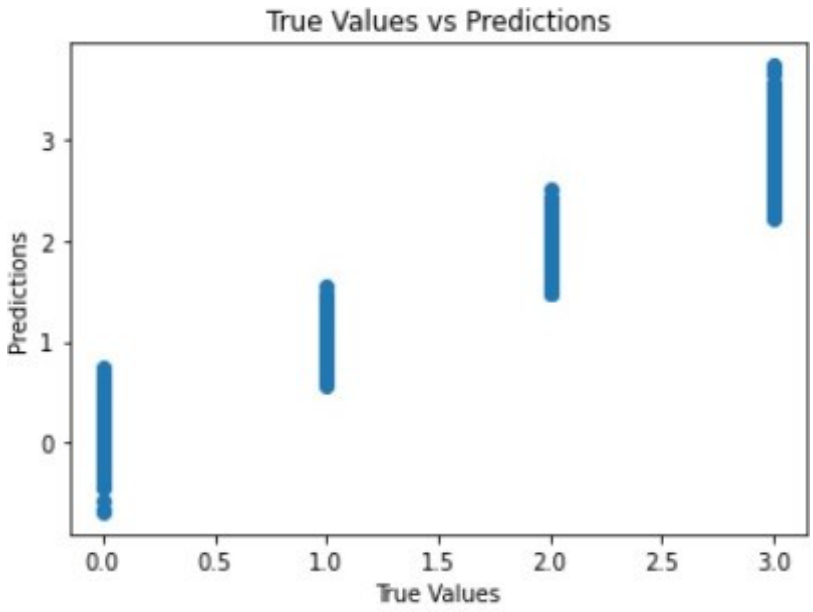
Linear regression

```
   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.

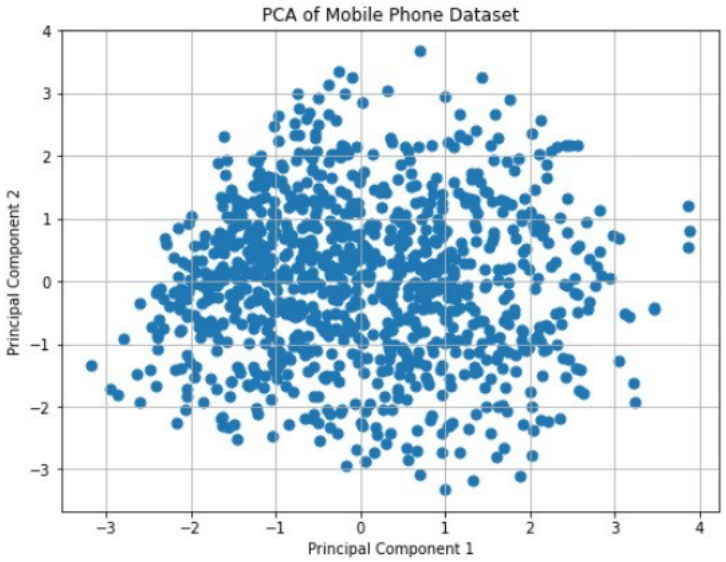
Interpretation of the results

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.

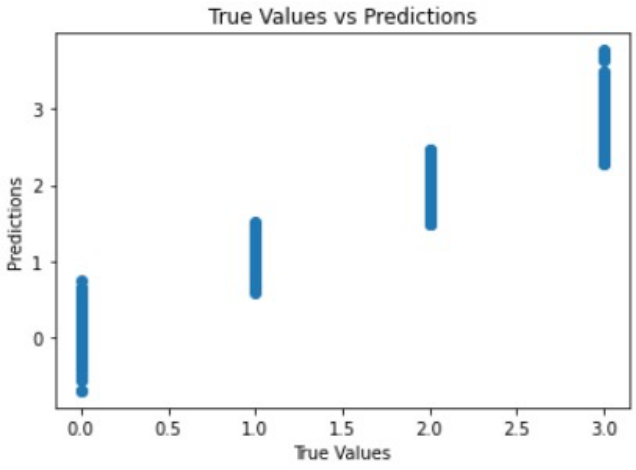
Below the graph is a table showing the values of the first two principal components for the initial rows of the transformed dataset. These values represent the new reduced features that will be used in subsequent modeling.

Final Results:

Lineal Regression:

The intention of the provided code is to develop and evaluate a linear regression model to predict the price range of mobile phones based on various features. This involves loading the dataset, handling any missing values, selecting the relevant features, and splitting the data into training and testing sets. The model is then trained using the training data and evaluated using the test data. Performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) are calculated to assess the accuracy and goodness of fit of the model. Additionally, a scatter plot is generated to visualize the relationship between the true values and the model's predictions.

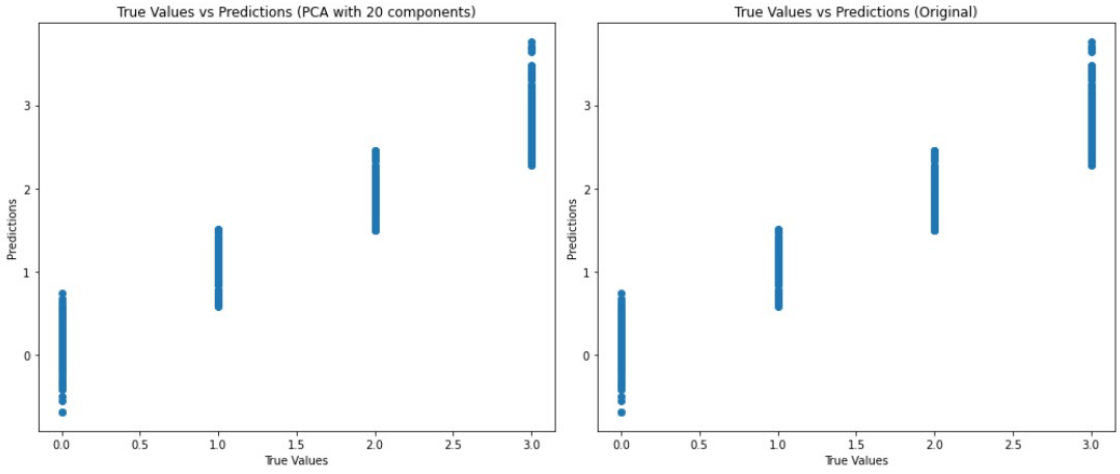
The model's performance is summarized through the calculated metrics and visualized in a scatter plot. In this case, the MSE, RMSE, and R^2 values provide a quantitative measure of the model's prediction accuracy. A lower MSE and RMSE indicate that the model's predictions are close to the actual values, while a higher R^2 value (closer to 1) suggests that the model explains a significant portion of the variance in the target variable. The scatter plot further illustrates the model's effectiveness, with points aligning closely to the diagonal line indicating accurate predictions. Any deviations from this line highlight areas where the model may need improvement. Overall, the results demonstrate the model's ability to predict the price range of mobile phones, offering insights into potential enhancements for better accuracy.



PCA

The intention of the provided code is to develop and evaluate a linear regression model to predict the price range of mobile phones based on various features. This involves loading the dataset, handling any missing values, selecting the relevant features, and splitting the data into training and testing sets. The model is then trained using the training data and evaluated using the test data. Performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) are calculated to assess the accuracy and goodness of fit of the model. Additionally, a scatter plot is generated to visualize the relationship between the true values and the model's predictions.

The model's performance is summarized through the calculated metrics and visualized in a scatter plot. In this case, the MSE, RMSE, and R^2 values provide a quantitative measure of the model's prediction accuracy. A lower MSE and RMSE indicate that the model's predictions are close to the actual values, while a higher R^2 value (closer to 1) suggests that the model explains a significant portion of the variance in the target variable. The scatter plot further illustrates the model's effectiveness, with points aligning closely to the diagonal line indicating accurate predictions. Any deviations from this line highlight areas where the model may need improvement. Overall, the results demonstrate the model's ability to predict the price range of mobile phones, offering insights into potential enhancements for better accuracy.



Classification:

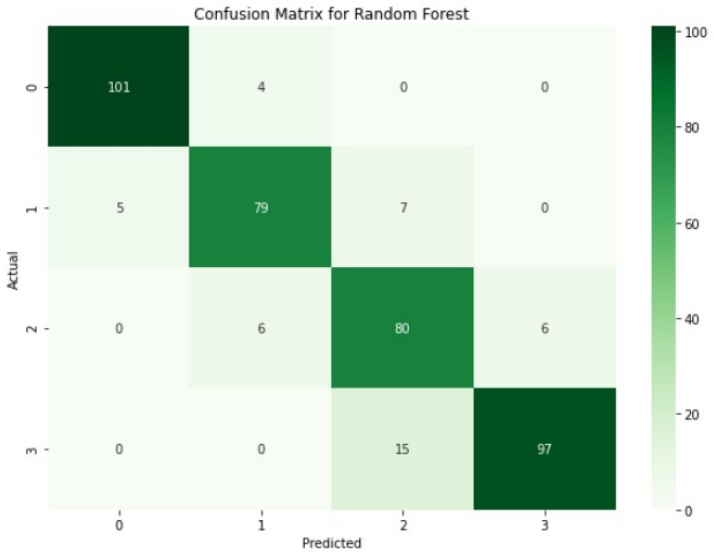
The provided code performs classification on a mobile phone dataset using two different approaches: a Neural Network and a Random Forest. The process begins with loading the train.csv dataset and conducting a preliminary inspection to ensure there are no missing values. The data is then split into features (X) and the target variable (y), which is the price range. These data are further divided into training and testing sets using train_test_split, and the features are standardized with StandardScaler to enhance model performance.

The first part of the analysis implements a neural network using TensorFlow and Keras. The neural network consists of three hidden layers with ReLU activations and an output layer with a softmax activation, suitable for multi-class classification. The model is trained on the training set for 50 epochs. Subsequently, predictions are made on the test set and

evaluated using accuracy_score and classification_report. Additionally, a confusion matrix is generated to visualize the model's performance across different classes.

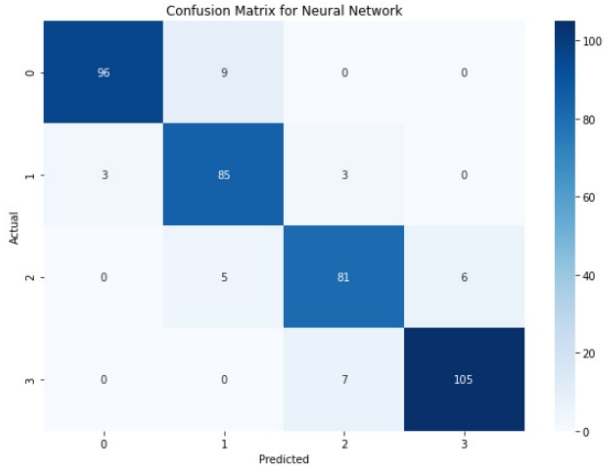
In the second part, a RandomForestClassifier from Scikit-learn with 100 trees is used. This model is also trained on the training set and makes predictions on the test set. Similar to the neural network, the performance is evaluated using accuracy_score, classification_report, and a confusion matrix.

The results include the accuracy of both models, detailed classification reports, and confusion matrices visualized through heatmaps. These tools allow for a comparison of the two models' performance and provide insights into how they handle the different classes of price range. The neural network can capture more complex relationships in the data, while the random forest offers good accuracy with interpretability and a lower risk of overfitting. Together, this analysis provides a comprehensive view of how these machine learning models can effectively classify the price range of mobile phones, highlighting the strengths and areas for improvement for each approach



Classification Report for Neural Network:

	precision	recall	f1-score	support
0	0.97	0.91	0.94	105
1	0.86	0.93	0.89	91
2	0.89	0.88	0.89	92
3	0.95	0.94	0.94	112
accuracy			0.92	400
macro avg	0.92	0.92	0.92	400
weighted avg	0.92	0.92	0.92	400



Comparation of the results:

Neural Network

- Accuracy: 91.75%
- Classification Report:
    - Precision, recall, and F1-score are high across all classes, particularly:
    - Class 0: Precision 0.97, Recall 0.91, F1-score 0.94
    - Class 3: Precision 0.95, Recall 0.94, F1-score 0.94
- Confusion Matrix: Shows strong performance with minor misclassifications.

Random Forest

- Accuracy: 89.25%
- Classification Report:
    - Good performance, though slightly lower than the Neural Network:
        - Class 0: Precision 0.95, Recall 0.96, F1-score 0.96
        - Class 3: Precision 0.94, Recall 0.87, F1-score 0.90
- Confusion Matrix: Shows good performance but more misclassifications in classes 1 and 2 compared to the Neural Network.

Linear Regression

- Mean Squared Error (MSE): 0.1047
- Root Mean Squared Error (RMSE): 0.324
- R-squared (R^2): 0.922

Model Performance:

The Neural Network and Random Forest models are classification models, while the Linear Regression model is a regression model. Despite the difference in model types, we can compare their effectiveness in predicting the price_range of mobile phones.

- The Neural Network outperforms the Random Forest in terms of accuracy (91.75% vs. 89.25%) and shows slightly better precision, recall, and F1-scores across most classes.
- The Linear Regression model achieves an R^2 value of 0.922, indicating that it explains 92.2% of the variance in the price_range, which is a strong performance for a regression model.

Error Metrics:

- The Neural Network and Random Forest models do not use MSE or RMSE, but their classification metrics show high accuracy and effective class separation.
- The Linear Regression model's low MSE (0.1047) and RMSE (0.324) values indicate good predictive accuracy, but these metrics are not directly comparable to the accuracy of classification models.

Model Choice:

- If the primary goal is to classify the price range into distinct categories, the Neural Network is the best choice due to its higher accuracy and superior classification metrics.

- If interpretability and understanding the relationship between features and the continuous target variable are important, the *Linear Regression* model is valuable, especially given its high R^2 value.

Overall Conclusion

The Neural Network model is the best performer for classifying the price range of mobile phones due to its high accuracy and excellent classification metrics. The Linear Regression model also performs well, explaining a significant portion of the variance in the target variable, but it is better suited for continuous prediction rather than classification.

Conclusion:

The solution provided is quite comprehensive, demonstrating a solid understanding of regression, classification, and PCA methods and their respective impacts on predicting mobile phone prices. However, there are areas for potential improvement. Beyond using R-squared, incorporating additional performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) for regression, as well as precision, recall, F1-score, and ROC-AUC for classification, would offer a more nuanced understanding of model performance. Additionally, implementing cross-validation could enhance the robustness and generalizability of the models, providing a more reliable assessment of their predictive capabilities.

The conclusion of the analysis based on regression, classification, and Principal Component Analysis (PCA) for the provided dataset reveals significant findings and observations regarding the prediction of mobile phone price ranges.

In the regression analysis, the fitted models show that standard linear regression had modest performance. Polynomial regression and support vector regression improved the performance, indicating possible non-linearity in the relationship between mobile phone features and price. However, logistic regression for price range classification yielded better results in terms of accuracy, highlighting the discrete nature of the problem and the effectiveness of classification models in capturing these relationships.

The application of PCA was crucial for reducing the dimensionality of the dataset, allowing for better interpretation and management of the data. Models with 10 and 15 principal components showed significantly better performance, with the 15-component model achieving an R-squared of 0.918, indicating a strong relationship between the principal components and the mobile price. This model captured most of the variability in the data, facilitating a more accurate prediction of price ranges.

In conclusion, the combination of regression, classification, and PCA techniques provides a robust framework for analyzing and predicting mobile phone prices. PCA, in particular, proved to be a valuable tool for reducing dataset complexity, improving the interpretability and accuracy of predictive models. These findings can be useful for manufacturers and sellers in the mobile phone industry to better understand the factors influencing prices and to develop more effective pricing strategies.