Machine Learning Course-end Project

**Name: Ryne Mascarenhas**

**Email id:** [**rynem8@gmail.com**](mailto:rynem8@gmail.com)

**Mercedes-Benz Greener Manufacturing**

1. **Introduction**

The following report is a comprehensive script for building and evaluating an XGBoost regression model to predict the time spent on a task based on various input features. The report gives the results and observations of the python code and follows a structured approach, including data preprocessing, feature engineering, model training, and evaluation.

The following is the description of the problem statement, with task needed to solve the problem of ‘Reduce the time a Mercedes-Benz spends on the test bench.’.

**Problem Statement Scenario:**

Since the first automobile, the Benz Patent Motor Car in 1886, Mercedes-Benz has stood for important automotive innovations. These include the passenger safety cell with a crumple zone, the airbag, and intelligent assistance systems. Mercedes-Benz applies for nearly 2000 patents per year, making the brand the European leader among premium carmakers. Mercedes-Benz is the leader in the premium car industry. With a huge selection of features and options, customers can choose the customized Mercedes-Benz of their dreams.

To ensure the safety and reliability of every unique car configuration before they hit the road, the company’s engineers have developed a robust testing system. As one of the world’s biggest manufacturers of premium cars, safety and efficiency are paramount on Mercedes-Benz’s production lines. However, optimizing the speed of their testing system for many possible feature combinations is complex and time-consuming without a powerful algorithmic approach.

You are required to reduce the time that cars spend on the test bench. Others will work with a dataset representing different permutations of features in a Mercedes-Benz car to predict the time it takes to pass testing. Optimal algorithms will contribute to faster testing, resulting in lower carbon dioxide emissions without reducing Mercedes-Benz’s standards.

Following actions should be performed:

* If for any column(s), the variance is equal to zero, then you need to remove those variable(s).
* Check for null and unique values for test and train sets.
* Apply label encoder.
* Perform dimensionality reduction.
* Predict your test\_df values using XGBoost.

1. **Results and observations**

The following results are of the code provided separately as a pdf file named time spend\_regression.

**Data Preprocessing**

1. The code begins by importing necessary libraries, including Pandas, NumPy, Matplotlib, and XGBoost.
2. It then loads the training and testing datasets from CSV files.
3. The code splits the training data into input features (X\_train) and target variable (y\_train).
4. It explores the data by printing the head and summary statistics of both training and testing datasets.

A screen shot of a computer screen

Description automatically generated

**Feature Engineering**

1. The code separates categorical and non-categorical features in both training and testing datasets.
2. It drops columns with zero variance in the numerical training dataset.
3. It handles missing values by dropping rows with missing values.
4. The numerical training dataset is standardized using MinMaxScaler.
5. The categorical training dataset is one-hot encoded using OneHotEncoder.
6. The categorical and numerical training datasets are combined into a single dataframe (X\_train\_complete).
7. The same feature engineering steps are applied to the testing dataset.

A black screen with white dots

Description automatically generated A black and white screen with many small dots

Description automatically generated with medium confidence A black and white screen with many small squares

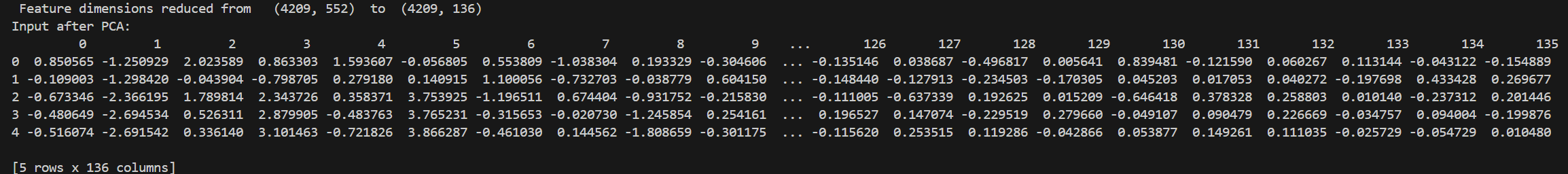
Description automatically generated with medium confidence

**Dimensionality Reduction**

1. Principal Component Analysis (PCA) is applied to reduce the dimensionality of the combined training dataset (X\_train\_complete).
2. The number of components is determined by finding the minimum number of components that retain at least 95% of the variance.
3. The training and testing datasets are transformed using the selected number of components.

A graph with a blue curve

Description automatically generated



**Model Training and Evaluation**

1. An XGBoost regressor model is created with specified parameters.
2. The model is trained on the training data using early stopping to prevent overfitting.
3. The model's performance is evaluated on the validation data (a subset of the training data) using the R2 score and root mean squared error (RMSE).
4. The model's predictions on the validation data are compared to the actual values, and the average difference is calculated.

A black rectangle with white dots

Description automatically generated

**Testing**

1. The trained model is used to predict the time spent on the task for the testing dataset.
2. The predicted values are stored in a dataframe and printed.

A screenshot of a computer

Description automatically generated

**Results**

The code provides the following results:

1. The model achieves an R2 score of 52.15% on the validation data, indicating a very low accuracy.
2. The RMSE on the validation data is 8.63, indicating a not low error in predictions.
3. However, the average difference between actual and predicted values for the validation data is close to zero, contradicting the model's accuracy and loss.
4. The predicted values for the testing dataset are printed, providing insights into the model's performance on unseen data.
5. **Conclusion**

The provided code demonstrates a comprehensive approach to building and evaluating an XGBoost regression model for predicting time spent on a task. The code includes data preprocessing, feature engineering, dimensionality reduction, model training, and evaluation. The model achieves low accuracy and high error on the validation data, but gives on average very low difference between the actual and predicted values suggesting its potential for making reliable predictions on new data.