**Author:** Divya Pardeshi

**Date:** 13 -11- 2022

**Mercedes-Benz Greener Manufacturing Project**

**Introduction:**

Mercedes-Benz, a pioneer in automotive innovation, is committed to ensuring the safety and reliability of their cars through a rigorous testing system. The challenge lies in optimizing the testing process, specifically reducing the time cars spend on the test bench. A more efficient testing system not only upholds Mercedes-Benz's high standards but also contributes to lower carbon emissions.

**Problem Statement:**

The dataset provided represents different permutations of features in a Mercedes-Benz car. The goal is to predict the time it takes for a car to pass testing based on these features. This prediction task aims to leverage algorithmic approaches to optimize testing speed and enhance overall efficiency.

**Actions Taken:**

1. **Importing Necessary Libraries:**

* Libraries such as NumPy, Pandas, Seaborn, Matplotlib, and XGBoost are imported for data manipulation, visualization, and building the predictive model.

1. **Loading and Exploring Data:**

* Train and test datasets are loaded into Pandas DataFrames (train\_df and test\_df).
* Basic exploratory data analysis is performed, including checking data types, information about the dataset, and identifying categorical features.

1. **Data Preprocessing:**

* Columns 'ID' and 'y' are removed from the training data as they are not crucial for model learning.
* Numerical and categorical features are separated for further processing.

1. **Handling Zero Variance Features:**

* Identified columns with zero variance and removed them from the dataset to enhance model efficiency.

1. **Label Encoding for Categorical Features:**

* Categorical features are label-encoded to convert them into numerical format for model training.

1. **Concatenating Features:**

* Categorical and numerical features are concatenated to create the final data

1. **Train-Test Split:**

* The dataset is split into training and testing sets.

1. **Model Building and Evaluation:**

* An XGBoost regression model is built using the training set.
* Model performance is evaluated using metrics such as Root Mean Squared Error (RMSE) and R-squared.

1. **Visualizing the Distribution of Actual and Predicted Values:**

* A distribution plot is created to visualize the actual and predicted values on the test set.

1. **Hyperparameter Tuning Using Grid Search:**

* Grid Search is performed to find the best hyperparameters for the XGBoost model.

1. **Predictions on Test Data:**

* The model is used to make predictions on the test set.
* Predictions are stored in a DataFrame and saved to a CSV file.

1. **K-fold Cross Validation with RandomizedSearchCV:**

* RandomizedSearchCV is employed to perform K-fold cross-validation and find optimal hyperparameters.

1. **Final Predictions on Test Data Using XGBoost:**

* The tuned XGBoost model is used to make final predictions on the test set.
* Predictions are stored in a DataFrame and saved to a CSV file.

**Conclusion:**

The code provides a comprehensive example of building, tuning, and evaluating an XGBoost regression model for predicting a continuous target variable in a tabular dataset. The use of various preprocessing techniques and hyperparameter tuning contributes to achieving optimal model performance.