**Project Write-Up: Reducing the Time a Mercedes-Benz Spends on the Test Bench Using Machine Learning**

**Project Objective**

The goal of this project is to reduce the time a Mercedes-Benz spends on the test bench by predicting the time required for different configurations of the car. By optimizing this time using machine learning algorithms, the testing system's efficiency can be improved, leading to lower carbon dioxide emissions without compromising safety standards.

This is achieved by developing a predictive model using the provided datasets, which include various feature permutations of Mercedes-Benz cars. The model’s task is to predict the time it takes for a car to pass testing, based on the configuration of features.

**Datasets**

* **Training dataset (train\_df)**: Includes feature variables and the target variable (y), which represents the time a car spends on the test bench.
* **Test dataset (test\_df)**: Contains only feature variables for which the model will predict the test bench time. It does not include the y (target) values.

**Step-by-Step Approach**

**Step 1: Preprocessing the Data**

**1.1 Drop the ID Column**

The ID column serves only as an identifier for the cars and does not provide any useful information for prediction. Hence, it was dropped from both the training and test datasets. However, the ID from the test dataset was saved for later use in the final submission.

**1.2 Label Encoding for Categorical Columns**

The datasets contained several categorical variables (e.g., X0, X1, etc.), which must be converted into numerical form. We used **Label Encoding** to convert these categorical columns into integers.

**1.3 Remove Zero Variance Features**

Some features may not vary across the dataset, which means they have no predictive power. We removed features with zero variance.

**Step 2: Dimensionality Reduction Using PCA**

With the pre-processed data, we further reduced the number of features using **Principal Component Analysis (PCA)**. This technique helped in retaining 95% of the variance while reducing the dimensionality of the dataset, which can improve the model's performance and prevent overfitting.

**Step 3: Train the XGBoost Model**

We used **XGBoost**, a high-performance gradient boosting algorithm, to train the model on the preprocessed and PCA-transformed training dataset

**Step 4: Make Predictions on the Test Dataset**

Once the model was trained, we used it to predict the test bench time for the cars in the test\_df dataset. The predictions were made on the PCA-transformed test set.

**Step 5: Save Predictions for Submission**

The predictions were saved in a CSV file, along with the ID column that was stored earlier.