Project: Mercedes-Benz Greener Manufacturing

Data Description:

1. This dataset contains set of variables, each representing a custom feature in a Mercedes car. For example, a variable could be 4WD, added air suspension, or a head-up display.
2. The ground truth is labeled 'y' and represents the time (in seconds) that the car took to pass testing for each variable.
3. File descriptions: Variables with letters are categorical. Variables with 0/1 are binary values.
4. train.csv — the training set
5. test.csv — the test set, we must predict the 'y' variable for the 'ID's in this file

Understanding the Dataset:

Checked for null and missing values in the dataset
Found the following features of data:

There are 378 entries, ID to X385

Datatypes are : float64(1), int64(369), object(8)

Also found that there are no missing values. So, now we will check for different datatypes available

Next check for different data types present in the dataset, this will give an idea of what types of features are present.

There are 8 Categorical Columns each representing a custom feature in a Mercedes car, 1 float column represents the time (in seconds) that the car took to pass testing for each variable (in our case target values), 369 integer values which indicates different tests performed on the car.

Check for null and unique values for test and train sets:

We have 12 features which only have a single/unique (Zero Variance) value in them — these are pretty useless for supervised algorithms, and should probably be dropped found this by applying correlation on the datasets and plotting an heatmap for correlation on the training set.

Data Preprocessing:

Applied One-Hot Encoder:

As mentioned earlier, we have 8 categorical columns and 369 integer values. These 369 integer values are already in '0' and '1' format, so we will convert other 8 categorical columns into one hot encoded columns to have entire train data in same format.

Applied Label Encoding :

Used Label Encoding to converting the **labels** into numeric form so as to convert it into the machine-readable form. **Machine learning** algorithms can then decide in a better way on how those **labels** must be operated.

It is an important pre-processing step for the structured dataset in supervised learning.

Used it for encoding the levels of **categorical** features into numeric values. **LabelEncoder** encode labels with a value between 0 and n_classes-1 (normalise) where n is the number of distinct labels.

Perform dimensionality reduction.

Used **PCA**(Principal Component Analysis) to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent.

After PCA   performed a train test split and checked the test and train set shape:

```
Train set shape :(3367, 364)
Test set shape: (842, 364)
```

Predict your test_df values using XGBoost

Applied XGBoost to train my model. Here I have used DMatrix as it is an internal data structure that is used by **XGBoost** which is optimized for both memory efficiency and training speed.

These are my Implications and the Results from this Project.
I have used RMSE score as an Evaluation Metric for this Project.

Results: After 150 num_boost_round of cross-validation:
train-rmse:7.61125
test-rmse: 8.55428

We have pretty decent values of Train & Test error parameter(RMSE)
.
Model is performing nicely & not over-fitting.