INDEX

1. INTRODUCTION

Daimler is one of the world's biggest manufacturers of premium cars. Safety and efficiency are paramount on Daimler's production lines to maintain this high standard and assure the customers high quality cars. Mercedes-Benz, a division of Daimler automotive must ensure the safety and reliability of each unique car configuration before they hit the road. Quality testing is an integral part of any manufacturing facility. Daimler's wants to achieve supreme quality in order to best prepare their vehicles for rigorous urban and rural operation. In order to meet these standards, there are various quality checks that Daimler's vehicles need to pass through production, before they are declared as 'Conforming or 'Non-Conforming'. These quality checks could include, Hardware in loop (HIL), component strength analysis, evaluation of vehicle vibrations, crash test simulations, etc. and many more. The speed of the testing system for many such possible feature combinations (safety tests) is complex and very time consuming. Added to this every test has its own individual energy requirements, and more the number of tests implies more energy consumed and therefore more carbon emissions during the energy generation process. In order to adhere to the required standard production rules, safety and efficiency for Daimler's production lines, to reduce the time spent on test bench and thus to cut down the energy usage, it is important to optimize the testing procedure. This project tries to build an algorithm-based optimizing model by employing machine learning techniques that would reduce the testing times required for Daimler cars to pass the test bench.

2. PROBLEM STATEMENT

The data set encompasses around 373 attributes and approximately 8400 instances. The objective is to accurately predict the time a car will spend on the test bench based on vehicle configuration. The objective is to develop an accurate model that will be able to reduce the total time spent testing vehicles by allowing cars with similar testing configurations to be run successfully. This

problem involves the use of machine learning regression algorithm since, it requires predicting a continuous target variable (the duration of test) based on the model (categorical features) of the car [X1-X9]. This problem is a supervised learning model, since the labels are available in the form of 'Y' which is in seconds. Model solutions will provide insight to what extent and how the car configuration affects the process time. One of the main aspects of machine learning is that the efficiency of current systems can be improved using massive quantities of data, which is routinely collected by companies. Reducing the process time via reducing the "standard deviation", carbon dioxide emissions can be lowered. The primary challenge was to tackle the curse of dimensionality. Resulting in speedier testing, resulting in lower carbon dioxide emissions without reducing Daimler's standards.

3. DATASET

There are 2 files available to use. 'train.csv' and 'test.csv' both containing 4209 rows of data. These rows essentially indicate the number of vehicles provided to us and their various configurations. The 'train.csv' also contains the target variable which is available as the first column in 'Y' which are numeric values in seconds. Both the test and train contain 8 different vehicle features with labels such as X0 – X8 and 369 tests performed for the corresponding vehicle models. All the features have been anonymized and they do not come with any physical representation. These 8 categorical features with values encoded as strings such as 'aa','az', 'bb' and so on. Since this information is proprietary, Mercedes-Benz holds the rights over them and they chose not to disclose them. There are 368 tests(X10-X385) which are integer values and only indicate 0 or 1 i.e. it indicates whether the tests are performed or not. Data has been empirically collected and Mercedes-Benz claims by that, the time measurement error is almost zero for time spent by each test on the testing bench.

4. DATA VISUALIZATION

Data visualization holds great importance in a Machine Learning problem. It helps to better understand the significance of data by visualizing them in terms of graphs, charts or distributions. Patterns, trends and correlations that might otherwise go undetected in text-based data can be easily identified and understood. One of the basic plots is the distribution plot of the time data. It helps us to understand how the train dataset is spaced.

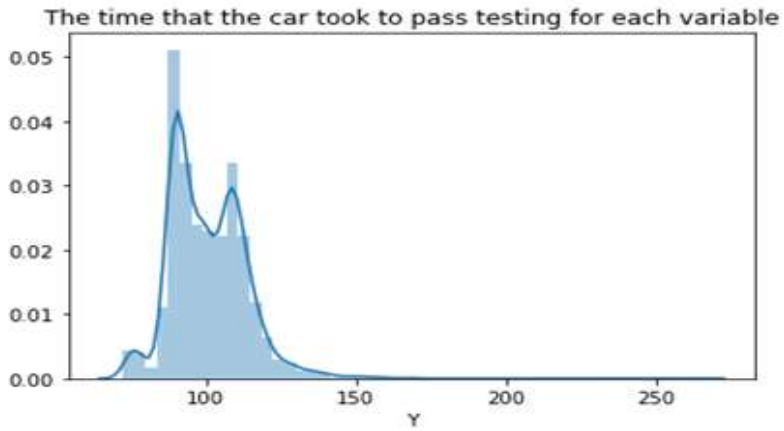

Fig 1. Distribution Plot

Fig 1. Distribution plot shows that we are dealing with multimodal data and that it is skewed towards the right.

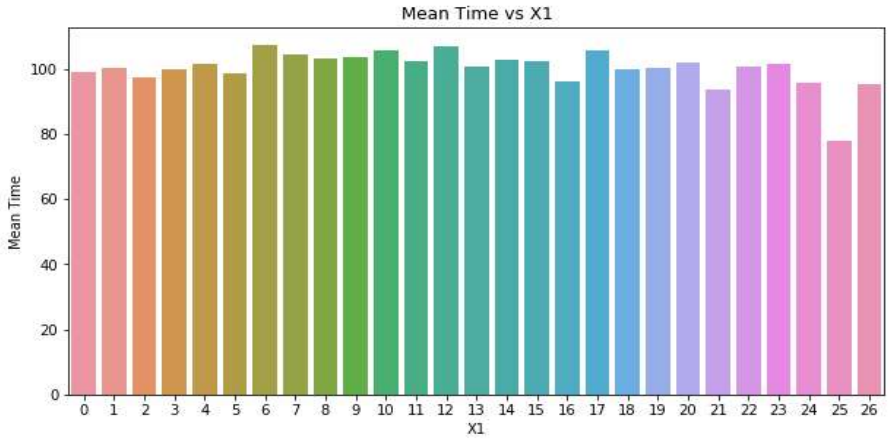

Fig 2. Mean test time for X1

Fig 2 shows mean testing time of every different instance of test feature X1. This helps us understand the impact of X1 feature on the testing time.

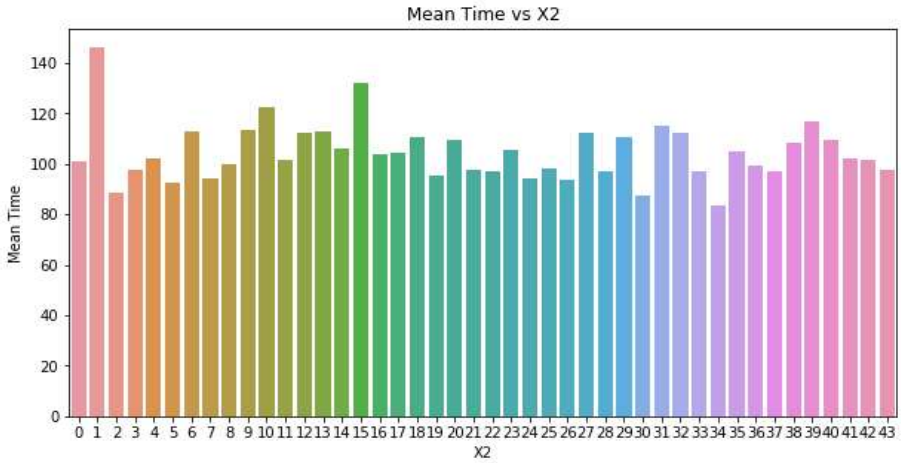

Fig 3. Mean test time for X2

Fig 3 shows mean testing time of every different instance of test feature X2. This helps us understand the impact of X2 feature on the testing time.

5. DATA PREPROCESSING

Since Mercedes-Benz hosted this project, the dataset was complete and there were no missing data. But upon close inspection, there were few variables (tests) that weren't run for any of the car features and hence they were dropped from both the training and test set. As mentioned earlier, we have 8 categorical variables in string format, since the regression models cannot process string information, they had to be converted into One Hot Encoded information. We used Label binarizing technique to One Hot Encode the categorical features. After One Hot Encoding the variables increased from 365 to 552 in both train and test set.

6. FEATURE ENGNINEERING AND FEATURE SELECTION TECHNIQUES

Often in Machine Learning problems, the dataset will have lot of features and each contributing to varied extent to the model. Some of the features would be very important and would contribute heavily to the model, while some features would not be so important and would act as noise overfitting or underfitting the model. In such cases to improve the accuracy of the training

model we must determine the importance of the features in the dataset. Accordingly following techniques were used -

6.1. Principal Component Analysis (PCA):

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. It emphasizes variation and brings out strong patterns in a dataset. PCA is implemented as a transformer object that learns n-components in its 'fit' method, and can be used on new data to project it on these components.

6.2. Singular Value Decomposition (SVD):

This transformer performs linear dimensionality reduction, contrary to PCA. This estimator does not center the data before computing the singular value decomposition. SVD makes it easy to eliminate the less important parts of the representation to produce the approx representation with any desired number of dimensions.

6.3. Independent Component Analysis (ICA):

Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear or nonlinear mixtures of some unknown latent variables, and the mixing system is also unknown. ICA can be seen as an extension to principal component analysis and factor analysis. ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely.

7. REGRESSION MODELS

This project is a supervised Machine learning problem where the algorithm receives a set of inputs and the corresponding outputs, so the algorithm compares its outputs with the correct ones then finds errors. Regression is a method of modeling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables. Accordingly, we would like to explore the following analytics methods of regression analysis.

1. Linear Regression
2. Ridge Regression
3. Lasso Regression
4. Decision Tree Regression
5. KNN Regressor
6. Artificial Neural Network (ANN)

7.1. Linear Regression

Linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. This is usually a naïve method and used as a basis for reference to other models. Based on the given data points, we try to plot a line that models these points with linear best fit. The line can be modeled based on the linear equation, for Eg.: $y = a_0 + a_1 * x$, here the motive of the linear regression algorithm is to find the best values for a_0 and a_1.

7.2. Ridge Regression

Ridge regression works by penalizing the magnitude of coefficients of features along with minimizing the error between predicted and actual observations. It is called 'regularization'. Ridge regression performs '**L2 regularization**', i.e. it adds a factor of sum of squares of coefficients in the optimization objective. The main features of ridge regression are:

- Performs L2 regularization, i.e. adds penalty equivalent to square of the magnitude of coefficients
- Minimization objective = LS Obj + α * (sum of square of coefficients).

7.3. Lasso Regression

LASSO stands for 'Least Absolute Shrinkage and Selection Operator'. Lasso regression performs L1 regularization, i.e. it adds a factor of sum of absolute value of coefficients in the optimization objective.

- Performs L1 regularization, i.e. adds penalty equivalent to absolute value of the magnitude of coefficients.
- Minimization objective = LS Obj + α * (sum of absolute value of coefficients).
- This project use alpha value of 0.001

7.4. Decision Tree Regression

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner. The recursion is

completed when the subset at a node has all the same value of the target variable. This project implements decision tree regressor with '1' random states

7.5. KNN Regressor

K nearest neighbors is a simple algorithm that stores all available cases and predicts the numerical target based on a similarity measure (e.g., distance functions). A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the K nearest neighbors. KNN regression uses the same distance functions as KNN classification. This project implements 10 neighbors

7.6. ANN (KERAS):

Keras has a methodology to organize layers. The simplest way to do this is using the 'Sequential' feature. This project tries to implement sequential layers of neurons and activation functions for each layers of model. It uses 'Relu' activation function since, the time in seconds or the 'Y' columns doesn't have any negative values. We have used 4 layers with 500, 300, 150 and 75 neurons each. The last layer is the output layer with 1 neuron which is to estimate the 'time in seconds. 'Adam' optimizer was used.

8. EVALUATION METRIC

The choice of evaluating metrics is very important especially when dealing with multiple models. This project uses R^2 and Mean Squared Error (MSE) as the main evaluation metrics. R^2 describes the amount of variation in the dependent variable, in this case the testing time of vehicles in seconds, based on independent variables which, in this case is the combination of vehicle custom feature. The co-efficient of determination is expressed as:

$$R^2 = \left(\frac{n * (\sum x * y) - (\sum x)(\sum y)}{\sqrt{n * [(\sum x^2) - (\sum x)^2] * [n * (\sum y^2) - (\sum y)^2]}} \right)^2$$

Where 'n', is the number of instances (which tests), 'x' is the prediction for the instance (the predicted time in seconds) and 'y' is the actual time (given in the dataset, in seconds).

Mean Square Error computes, a risk metric corresponding to the expected value of the squared error or loss. If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over $n_{samples}$ is defined as

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2$$

9. CLASSIFICATION

Logistic Regression

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured where there are only two possible outcomes. The dependent variable is binary or dichotomous, i.e. it only contains data coded as 1 (TRUE, success, present, etc.) or 0 (FALSE, failure, non-present, etc.). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest and a set of independent variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest

10. RESULTS

The train scores after implementing different regression models:

USING PCA		
Model	**R^2 Values**	**MSE**
Linear Regression	0.5357	76.2435
Ridge Regression	0.5483	74.2217
Lasso Regression	0.5460	74.6026
Decision Tree Regressor	-0.1300	181.2991
KNN Regressor	-0.051	167.8394

USING SVD		
Model	**R^2 Values**	**MSE**
Linear Regression	0.5068	81.20278
Ridge Regression	0.5482	74.22567
Lasso Regression	0.5461	74.58861
Decision Tree Regressor	-0.0511	166.8426
KNN Regressor	-0.0517	167.8498

USING ICA		
Model	**R^2 Values**	**MSE**
Linear Regression	0.462693	88.47291
Ridge Regression	0.39009	99.0419

Lasso Regression	0.532907	76.76558
Decision Tree Regressor	-0.057	169.1616
KNN Regressor	0.343127	106.8386

Model	**R^2 Values**	**MSE**
Logistic Regression	0.3352	-
ANN	0.5639	59.1438

Since the dataset was anonymous, different feature engineering and selection techniques was used, subsequently each of them was run on every model producing different R^2 score and MSE values. Based on this comparison metric Ridge regression model run using PCA dataset proved to the best model, as it gave the best R2 score of 0.5483 and the least MSE of 74.22567. The predicted values generated an accuracy of 0.5277.

11. FUTURE SCORE

There are plenty of way to expand on the work done in this project. Since the data is proprietary most the features were anonymous, that they had generic labels and not real features making it difficult to understand the feature set. This also limited the elimination of many features, resulting in lot of noise in the model. The length to breadth ratio of data is very large. By encoding some attributes, we are adding more attributes to the dataset. This could add a lot of noise to the data as well. It gets more challenging to model the same dataset with more attributes. Most of the attributes are available in binary format. The model may be prone to 'overfitting'.

Should Mercedes-Benz decide to provide more information about the dataset, then feature selection would have been even better resulting a better model and R^2 score.

ANN model could also be further developed, by fine tuning the activation function with different number of neurons and layers. One or more variable reduction techniques can be added to the ANN model thus improving its learning rate. This ultimately would have improved the model and provide better R^2 score and better prediction.

Implement more advanced and better model like 'XGBoost', Random Forest, Bagging, Boosting, and Gradient Boosting is always an option, which are proven to get better results.

12. CONCLUSION

The main objective of this project was to learn, understand and explore the different possibilities with basic regression models like Linear, Ridge, Lasso and others. Using the Mercedes-Benz data set the regression model were trained, cross-validated and tested before making the final predictions. According to evaluation metrics and with type of available data our regression model has pretty good prediction performance.

Thus, from Table 1 it can be concluded that our Model optimizes and predicts the testing time with a R^2 score of 0.5483 after applying PCA using the Ridge Regression Technique.