Name: Rohit Seelam

Email address: shreerohit24@gmail.com

Contact number: 9849144286

Anydesk address:

Years of Work Experience: **0.**

Date: **12th Dec 2020**

Self Case Study -1 :

MERCEDES-BENZ GREENER MANUFACTURING

OVERVIEW:

INTRODUCTION:

To ensure safety and reliability, Daimler (Mercedes-Benz) has developed a Robust testing system for their automobiles with different configurations or features before they reach their customers. With the power of machine learning we predict the process time and help the manufacturers optimize the time a car spends on the test bench.

BUSINESS PROBLEM:

Optimizing the speed of their testing system for so many possible feature combinations is complex and time-consuming without a powerful algorithmic approach. The model predicted provides insight to what extent and how the car configuration affects our process time and it will be helpful for the engineers to optimize and reduce the time that cars spend on the test bench resulting in speedier testing.

DATASET :

The train dataset has 4209 data-points each point having various configurations or features of the cars. There are 378 columns with 377 features and a column representing the time the car took for testing. There are 8 categorical features and 369 binary features. There are no NaN values present in the data set.

ML FORMULATION & PERFORMANCE METRIC:

This is a Machine learning Regression task ie. The variable we need to predict in this problem is a continuous variable. Here, The R^2 metric (coefficient of determination) is used as it is a good measure to determine the quality of a model in regression tasks. It provides the variation of the process time (dependent variable) based on the vehicle's configurations (independent variables).

Research-Papers/Solutions/Architectures/Kernels:

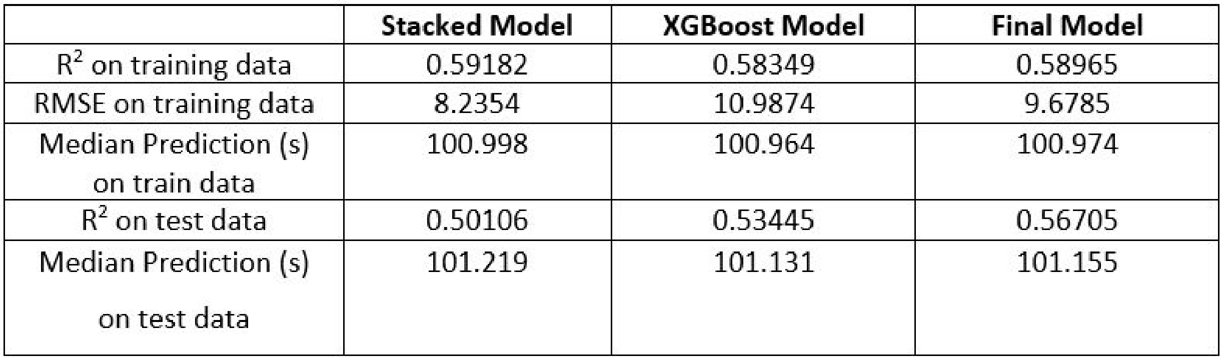
1. https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/car-testing.htm

   This blog makes us understand the importance of our problem better. Since the dataset did not describe what the features meant physically, we get some idea about the various tests performed on a car before productionizing it and reaching the consumer market by referring to the above mentioned Blog.

2. https://medium.com/@williamkoehrsen/capstone-project-mercedes-benz-greener-manufacturing-competition-4798153e2476

The above mentioned blog explains detailed problem overview, data cleaning, Exploratory Data analysis on the categorical and binary variables is done and comparison between various models on the problem . This blog helped me to understand the objective of the problem better .

Many algorithms are used by William Kohersen to solve the regression problem. The results he got from various models is shown in Fig.1 below.



	Stacked Model	**XGBoost Model**	**Final Model**
R^2 on training data	0.59182	0.58349	0.58965
RMSE on training data	8.2354	10.9874	9.6785
Median Prediction (s) on train data	100.998	100.964	100.974
R^2 on test data	0.50106	0.53445	0.56705
Median Prediction (s) on test data	101.219	101.131	101.155

Figure.1 Performance of various models

The baseline model used in this solution is linear regression . Baseline model is stacked with random forest model and a weighted average between Xgboost model and stacked model is taken finally.

3. https://www.kaggle.com/sudalairajkumar/simple-exploration-notebook-mercedes

   In the above mentioned code data cleaning , Exploratory Data Analysis is done on the Categorical Features and the Binary Features. Feature importance is determined using xgboost as well as Random forest regressor.

Feature importance is determined using two models in the above solution, namely Xgboost and Random forest model. The feature importances are shown in below figures.

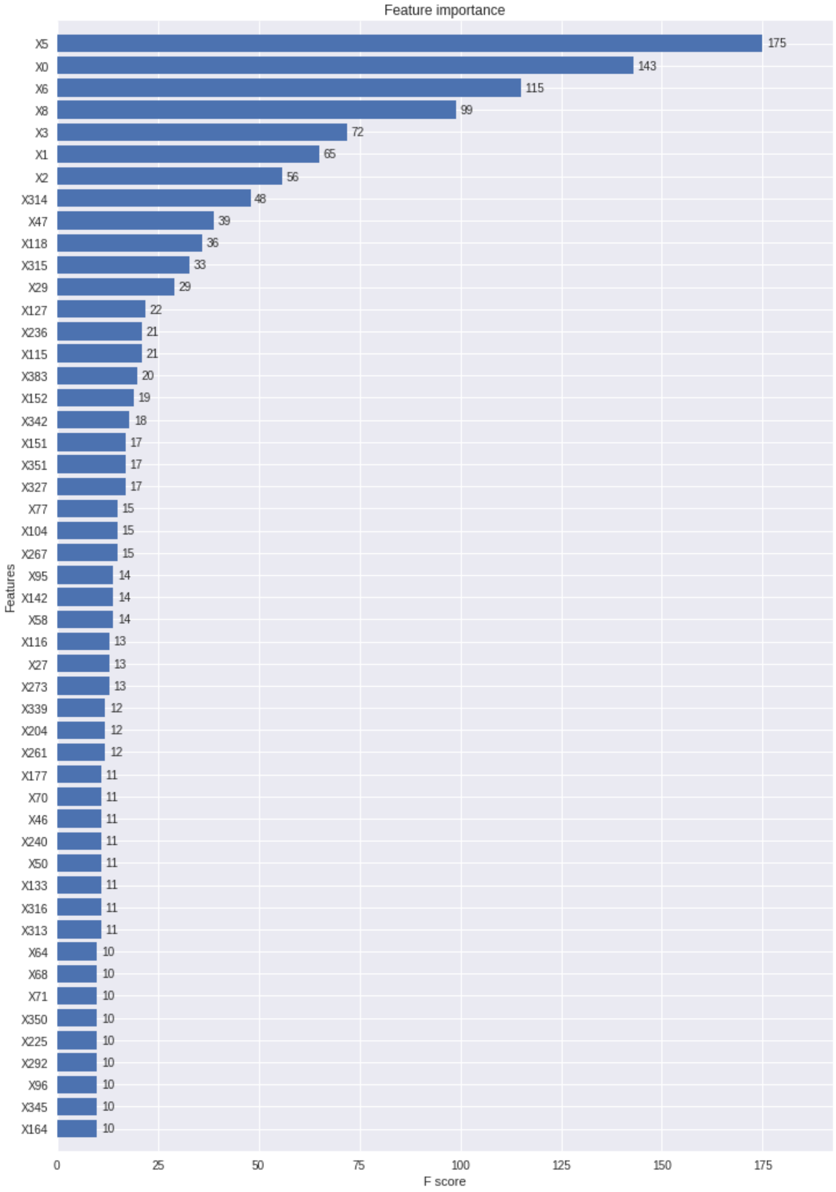


Fig.2 Feature importance using Using Xgboost

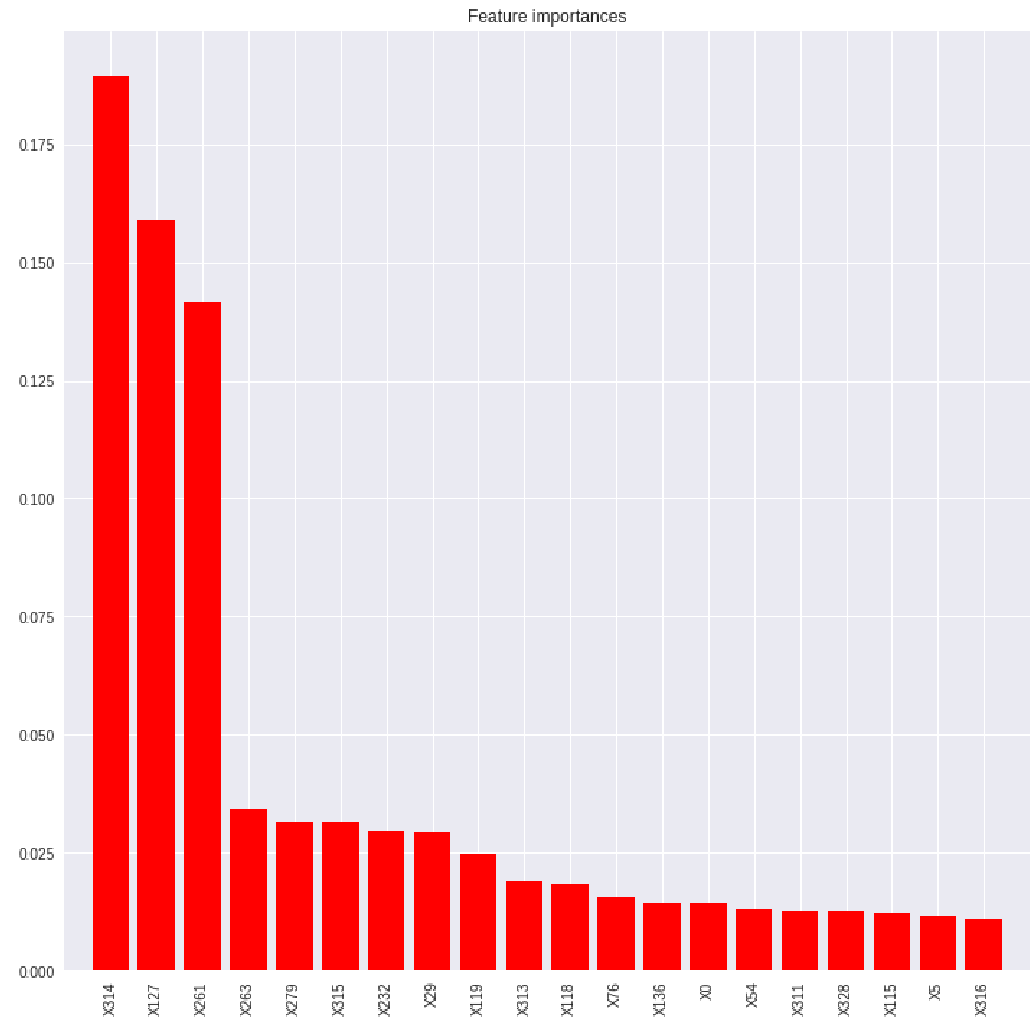


Fig.2 Feature importance using Using Random Forest Model

The categorical features had highest importance. The important features deduced from both the models varied slightly .

4. **https://www.kaggle.com/oysteijo/pca-and-t-sne**

There are 4200 samples and 350 features, To solve the problem of Curse of Dimensionality. Dimensionality Reduction techniques like PCA and T-sne are used by the author in the above solution.

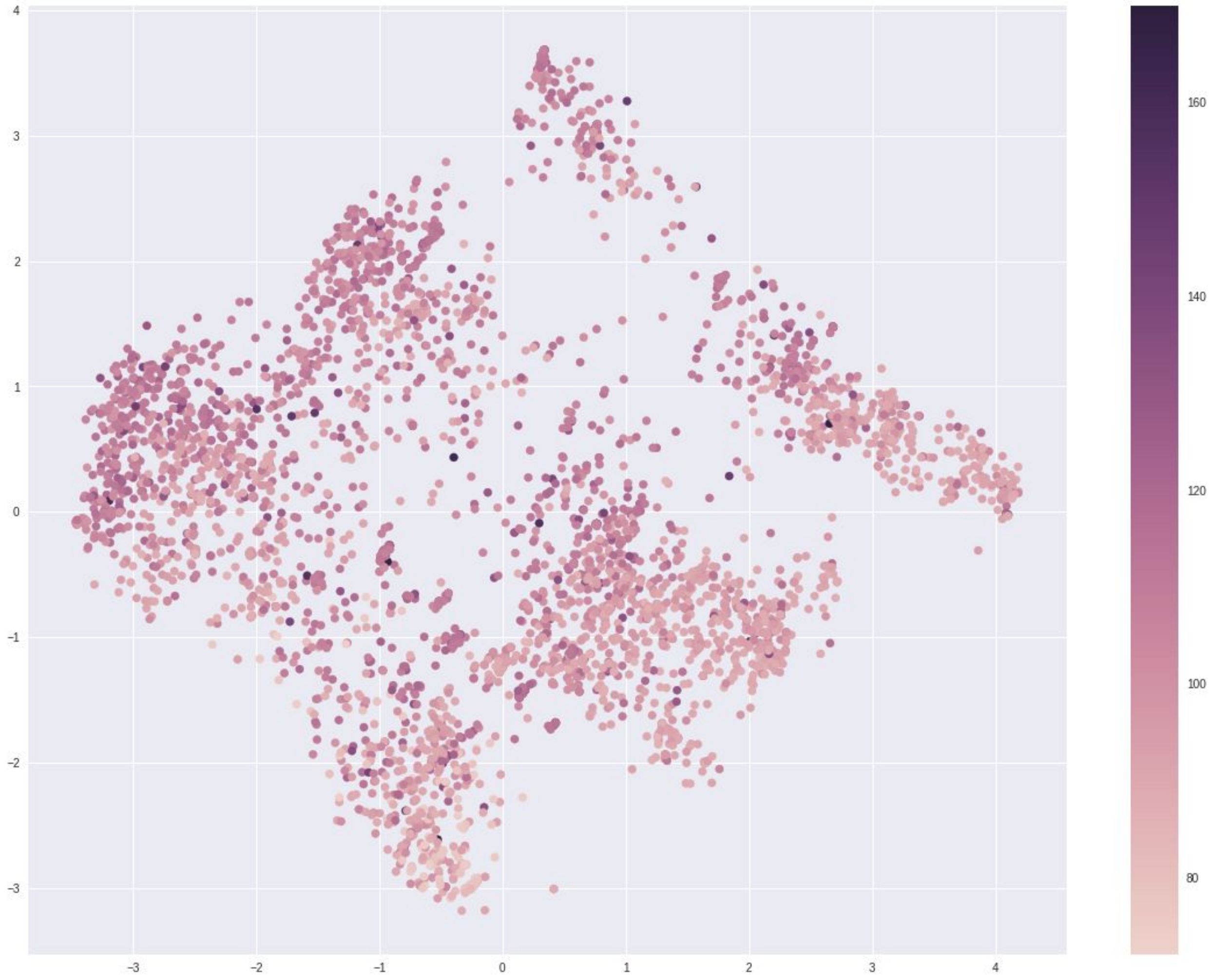


Figure.4 Dimensionality Reduction Using PCA.

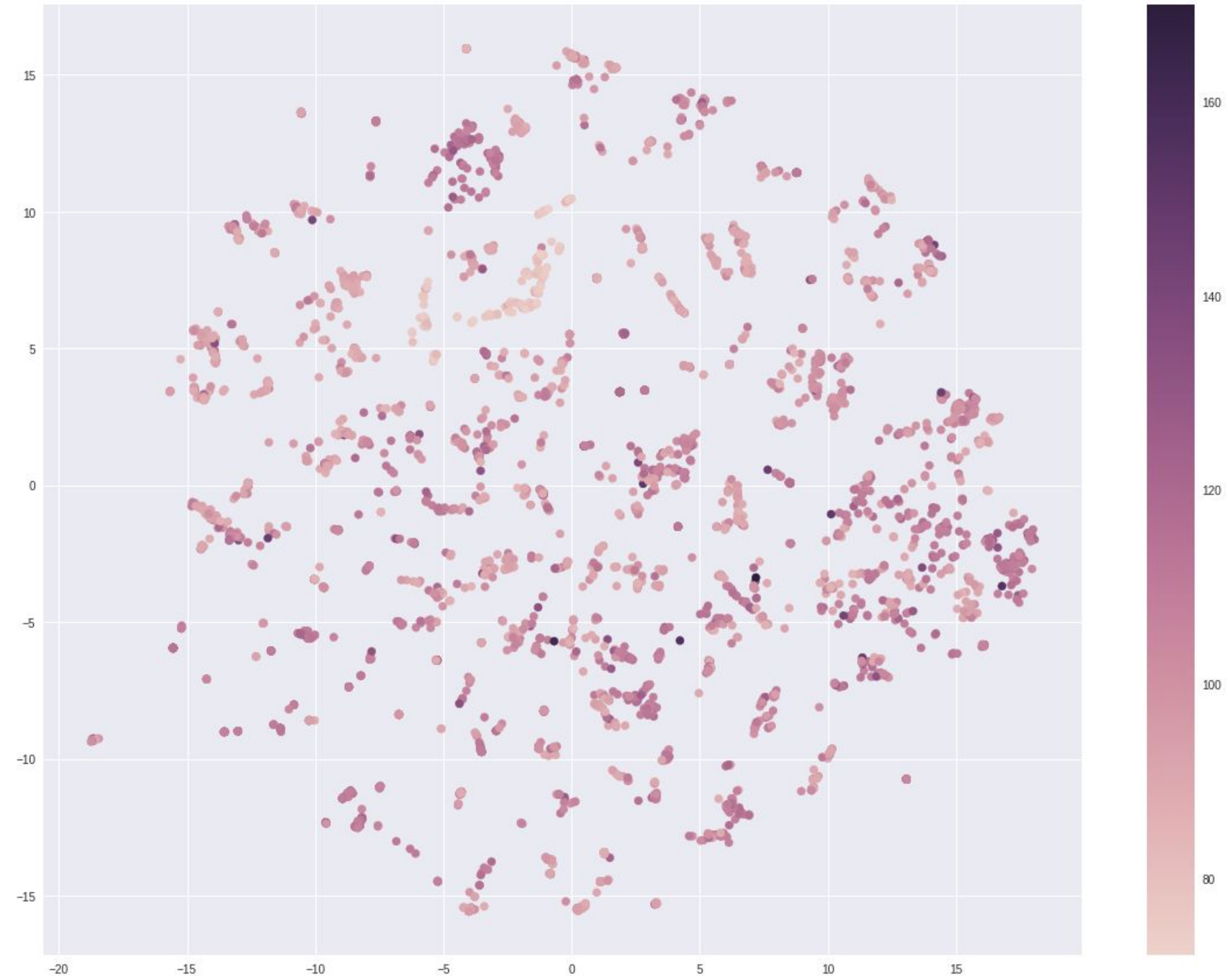
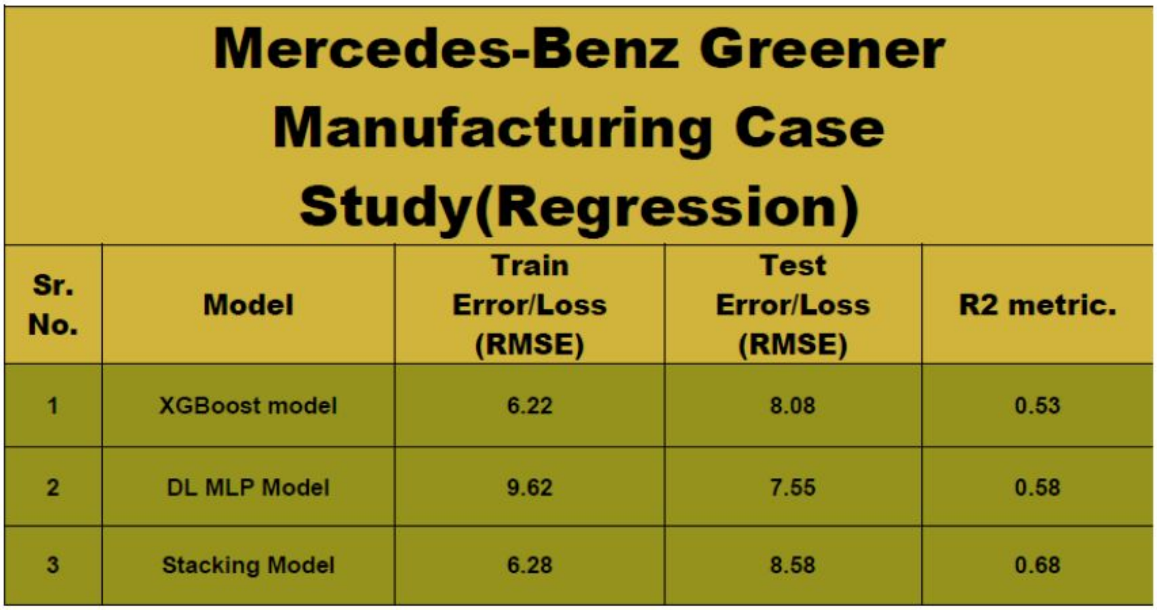


Figure.4 Dimensionality reduction using T-sne.

By converting the points to 2-dimensions the author plotted the distribution of the target variable and observed patterns in both techniques.

5. https://blog.goodaudience.com/stacking-ml-algorithm-for-mercedes-benz-greener-manufacturing-competition-5600762186ae

In this solution, Stacking gave good results compared to the Xgboost and DL models. Overfitting problem is not seen while using the stacked model. Figure.5 Shows the results using the various models.



Mercedes-Benz Greener Manufacturing Case Study(Regression)				
Sr. No.	Model	Train Error/Loss (RMSE)	Test Error/Loss (RMSE)	R2 metric.
1	XGBoost model	6.22	8.08	0.53
2	DL MLP Model	9.62	7.55	0.58
3	Stacking Model	6.28	8.58	0.68

Figure.5

First Cut Approach

1. **Data acquisition**: Train and test data is readily Available in kaggle, So is quite easy and we can proceed with further steps.
2. **Data cleaning** is the first step : checking if there are any null values and removing the binary features with unique values i.e Either completely 1 or 0. Removing outliers based on the target value because it is the only continuous variable in the dataset. Removal of any Duplicates present in the dataset.
3. **Exploratory Data analysis:** on the Target variable, categorical features and binary features need to be done and check if we can deduce some relationship between the input variables and the target values.

4. Perform **Dimensionality reduction techniques** like PCA and T-sne to see how the data is spread.
5. **Checking Feature Importance:** we remove the features which have less importance and try to add some new features.
6. **Implementing ML models:** Building various Regression models and train the dataset to see which model gives the best R^2 value.

Notes when you build your final notebook:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument "X" (a single data points i.e 1*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
    a. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
    b. so in your final notebook, you need to pass only those two values
    c. def final(X):
    ```
            preprocess data i.e data cleaning, filling missing values etc
            compute features based on this X
            use pre trained model
            return predicted outputs
    final([time, location])
    ```
    d. in the instructions, we have mentioned two functions one with original values and one without it
    e. final([time, location])   # in this function you need to return the predictions, no need to compute the metric
    f. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y_predict)
4. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data

5. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
6. Check this live session: https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models