Mercedez Benz Greener Manufacturing: Data Analysis and Optimizing Testing Time

Siddharth Jain*
Vignesh Dhanapal*
siddharthj@vt.edu
vigneshdhanapal@vt.edu
Virginia Polytechnic Institute and State University
Blacksburg, Virginia, USA

ACM Reference Format:
Siddharth Jain and Vignesh Dhanapal. 2020. Mercedez Benz Greener Manufacturing: Data Analysis and Optimizing Testing Time. In *Proceedings of Virginia Tech (Course Project 2020).* CS-55525, Blacksburg, VA, USA, 11 pages.

1 Introduction

The project deals to optimize the testing time for an automobile manufacturing company, Daimler. It is one of the world's biggest manufacturers of premium cars, safety and efficiency are paramount on their production lines. After a car is manufactured it goes through multiple testing conditions. A car consists of numerous components and each comes with multiple features. Data scientists and engineers in the organization have already developed a robust testing system according to the company standards maintaining the safety and reliability of each and every unique car configuration before the vehicle is ready to be driven on the road. This project successfully deals with contributing to the faster testing, this might result in lower carbon dioxide emissions or better performance without reducing Daimler's standards.

1.1 Motivation

The current motivation of this project is to optimize the speed of their testing system using a powerful algorithmic approach for so many possible feature combinations. For any manufacturing industry the biggest challenge that comes after their R&D designs is the time duration that their product takes from the step 1 of manufacturing till it launches in the market. If this can be reduced, the industry can make huge profits. Thus looking at the weight of this challenging problem we decided to move ahead with this project.Our project will work with a dataset representing different permutations of Mercedes-Benz car features to predict the time it takes to pass testing.

*Both authors contributed equally to this research.

Course Project 2020, December 2020, Blacksburg, VA, USA

ACM ISBN ...$

1.2 Problem Statement

This problem is to tackle the curse of dimensionality for the given data set and reduce the time that cars spend on the test bench, labeled as y in the dataset. To ensure the safety and reliability of each and every unique car configuration before they hit the road, Daimler's engineer have developed a robust testing system. But optimizing the speed of their testing system for so many possible feature combinations is complex and time consuming without a powerful algorithmic approach.

The whole project will be a combination of different algorithms used for data visualization, data pre-processing, model building and evaluation. We have also looked for any possible outliers and also modeled our algortihms in such a way to reduce overfitting.

2 Data Description

The entire data was provided in two files, test.csv and train.csv by providers Mercedes-Benz for the competition on a popular website Kaggle [1].These files have an extension of .csv which stands for Comma Seperated Values. Total amount of data including both testing and training sets consist of 378 columns and 4210 data points. This dataset contains anonymized set of variables in order to protect the Intellectual property, each representing a custom feature in a Mercedes-Benz car which defines a particular car model. For instance, a variable could be 4WD (Four Wheel Drive), added air suspension, or a head-up display. Among these 378 columns all the features are considered as categorical. The first 9 features(X0 - X8) are categorical values encoded in some manner by the Daimler. The remaining features (X10 - X385) are binary values which are either 0 or 1.

3 Data Pre-processing

Data overall can vary a lot, it can be big with some hundreds of data points or it can huge with around ten thousand data point. Talking in terms of Data mining applications every data irrespective of the source and application it usually structured in tables having rows and columns. A most likely scenario is that rows are the data points and columns can be the features, classes, or input/output variables but being said

converse can also be true where data can come in the form of images or audio/video files.But here one thing is sure that the final stage of data mining is in the form of matrices of n^{th} order.

To perform any kind of analysis, performing regression or classification on any type of data, data pre-processing is very important. It can be considered as broad area and consists of a number of strategies, techniques and algorithms. These steps consists of Data Quality Assessment, Feature Aggregation, Feature sampling, Dimensionality reduction, Feature Encoding.[6]

3.1 Feature Encoding

The whole purpose of data pre-processing is to convert available data such that it can be parsed by machine learning or deep learning algorithms. Almost all algorithms require input data in numerical form. But it is not always necessary that data is generated or created in that order. For instance an input categorical values can contain alphabets, symbols, or combination of both. In such case the algorithms would not be able to read this data and thus those features will be excluded while training.

As discussed earlier in section 2 we have 8 features that contain categorical values (X0 - X8) and in order for our models, to be discussed later, it was necessary to encode these values to numerical values. We used a method known as Label Encoder in the machine learning package known as Scikit-Learn. It converted all the values in the categorical features to numerical value as shown in Figure 1

	ID	y	X0	X1	X2	X3	X4	X5	X6	X8
0	0	130.81	k	v	at	a	d	u	j	o
1	6	88.53	k	t	av	e	d	y	l	o
2	7	76.26	az	w	n	c	d	x	j	x
3	9	80.62	az	t	n	f	d	x	l	e
4	13	78.02	az	v	n	f	d	h	d	n

	ID	y	X0	X1	X2	X3	X4	X5	X6	X8
0	0	130.81	37	23	20	0	3	27	9	14
1	6	88.53	37	21	22	4	3	31	11	14
2	7	76.26	24	24	38	2	3	30	9	23
3	9	80.62	24	21	38	5	3	30	11	4
4	13	78.02	24	23	38	5	3	14	3	13

Figure 1. Feature Encoding for categorical features

We can clearly see that each instance is assigned a unique value to each existing categorical value and convert the dataframe to the new vriable. A important point to note here is that, while encoding these features we do have to include testing data as well. A common mistake here can be be made is just encoding the training data but it can lead to decrement in the overall model performance and generality as the model would not know how to respond it if encountered a feature in testing set.

3.2 Feature Sampling

This method is used for reducing time to train the model. It is very well obvious to think that more the data we use, better is the training but it can be disastrous if one is not careful. Sometime a lot more data than required can sometimes leads to overfitting. Although there are many more factors that accounts for overfitting but that is discussed later in section 55. Therefore a certain amount data is sufficient to train a generalized model (which can be applied on a similarly structured data).

Another very important usage of this step is to training time. In many research it is found that some model algorithms do not converge easily on certain types of data. Now let's say if we have huge data and we are trying various model to find the best model for our application and every model takes around an hour to get trained, but what if even after an hour the testing error is 13%, which is more when compared with other. This can be considered as a wastage of memory and computation cost. In most cases, working with the complete dataset can turn out to be too expensive considering the memory and time constraints. Thus to reduce this problem we reduce our complete data and split it into two different sets.

For our project we split our training data into two further subsets 80-20% and used this 80% to train our model and remaining to test our model for computing mean square error.

3.3 Dimensionality Reduction

As we move forward in industrial age we have to deal with more and more data, containing hundreds of features having thousand and thousands of values. We can also call these features as dimensions and this makes the researchers to coined a term 'The Curse of Dimensionality'. This basically refers to the complexity increased due to such high number of features, thus making significantly harder to train the model. As the dimensionality increases, the number planes occupied by the data increases thus adding more and more sparsity to the data which is difficult to model and visualize.

As discussed we have a total of 368 number of features that contain binary values, they can also be classified as categorical data but contains 0 and 1. We started with statistical analysis to compute the variance of all the values across all feature variables. Since the values were binary, to make the visualization easy we plotted the count across 4209 data points for each feature [2]. Figure 2 shows some sample data as a proof of concept, complete image is included in the Appendix.It shows distribution of some sample binary features in both the training and testing set combined.

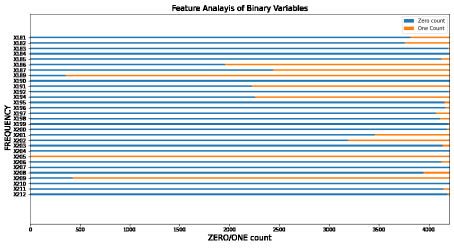

Figure 2. Feature Encoding for categorical features

Further digging into the plot it was found that there were 12 different features which contain only zeroes and as there is no variation in these features, they have no predictive power and therefore, These 12 features, ['X11','X93', 'X107', 'X233', 'X235', 'X268', 'X289', 'X290', 'X293', 'X297', 'X330', 'X347'] can be removed from both testing and training data. Although one can say that from 368 features will removing 12 features can make a difference? Definitely yes, even one feature makes a difference but measure can be low. Further discussion on this is done in section 5.

There are many other methods which are used to solve this purpose such as Correlation Matrix, Principal Component Analaysis (PCA - is explained in section Model Building 5.2)

3.4 Correlation Matrix

As the term implies Correlation Matrix relates different features among each other and compute a measure of relatability.This method is used to summarize a large amount of data where the goal is to see patterns.

Another important application are exploratory factor analysis, confirmatory factor analysis, structural equation models, and linear regression when excluding missing values pairwise. Looking at our problem, a perfect regression problem, it was very necessary to compute the correlation matrix. Talking in terms of Statistics here it is how correlation matrix is calculated.

Correlation:

$$corr\left(\vec{X},\vec{Y}\right) = \frac{covariance\left(\vec{X},\vec{Y}\right)}{std\,dev(\vec{X})\cdot std\,dev(\vec{Y})} = \frac{S_{xy}}{S_x \cdot S_y}$$

Covariance:

$$S_{xy} = \frac{1}{n-1}\cdot\sum_{k=1}^{n}(x_k-\overline{x})\cdot(y_k-\overline{y})$$

Standard Deviation:

$$S_x = \sqrt{\frac{1}{n-1}\cdot\sum_{k=1}^{n}(x_k-\overline{x})^2}$$

$$S_y = \sqrt{\frac{1}{n-1}\cdot\sum_{k=1}^{n}(y_k-\overline{y})^2}$$

Figure 4 and 3 shows a correlation matrix for both Binary Features(X10 - X385) and Categorical Features(X0 - X8)

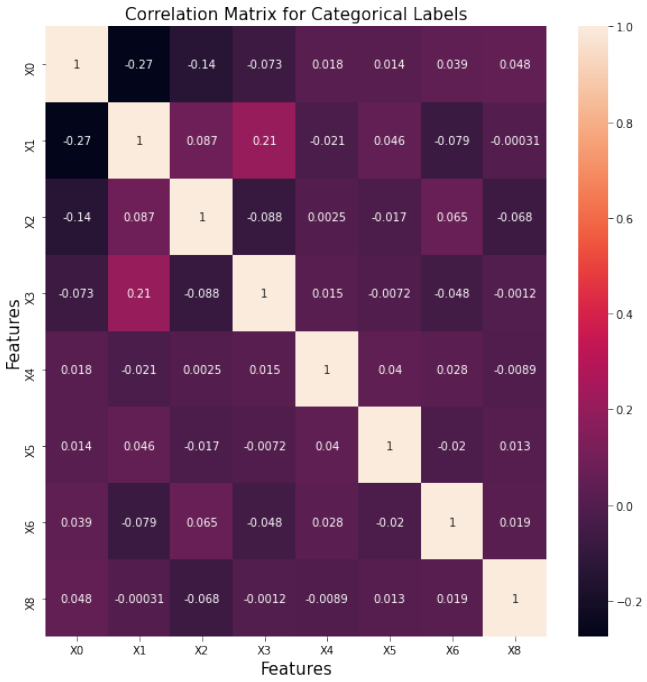

Figure 3. Feature Encoding for categorical features

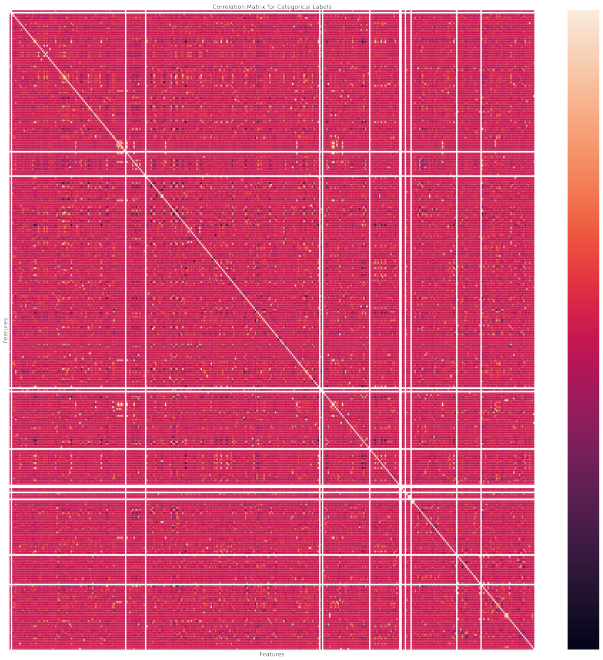

Figure 4. Feature Encoding for categorical features

We can see that all the features are related to each other and of course the diagonal elements are 1 because a feature will be related 100% to itself. Similarly for the binary features we can see that there are a few features which are not at all related to other thus showing a white line for that particular

feature. On printing those columns it was found that those were those same features that showed zero variance.

3.5 Data Quality Assessment

There are very high chances that a data might contain duplicate and inconsistent values also sometime 'NaN', stand for not a number meaning missing data. This method helps finding and removing such data points from the available data. Cleaning the outlier is a very important step in Data Processing. To train our model efficiently we performed a few data visualization on output values to find any outlier if present. Figure5 shows a scatter plot and from this we can clearly see an outlier. Outlier is defined as datapoint which does not follow the trend of the entire dataset. This can bring many discrepencies while training the model and hence must be removed to improve the performance

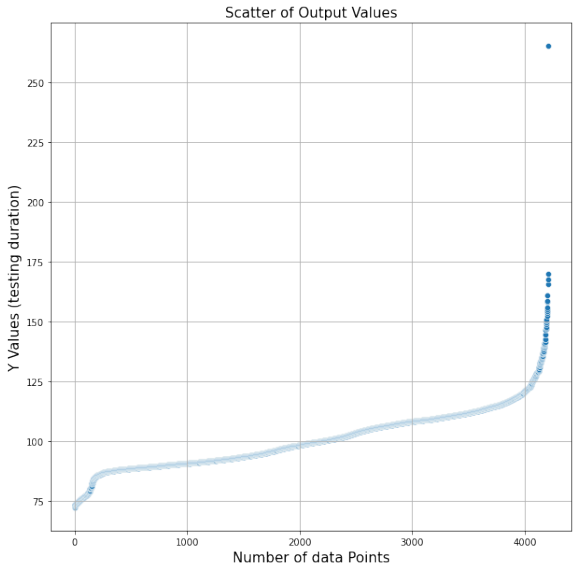

Figure 5. Scatter output of Y values

4 Data Exploration

We started with raw data which consists of outliers, not usefull features, and having values that are unreadable to ML/DL algorithms. After performing methods performed in previous section, 3.4, 3.3, 3.5, 3.1, 3.2 we were able to clean both training data and testing data. We exported these dataframes in .csv format using Pandas package library further evaluation and training the model. This data we refer in code and other material as Cleaned Data.

This cleaned data consist of no outlier, 12 feature lesser than the raw data, and categorical encoded. Figure 6 shows histogram to better visualize the distribution[5]. Distribution of data is best done with the help of Histogram. There are several conclusions to be drawn from this histogram: First, the majority of test durations are between 90 and 100 seconds. Second, there are peaks in testing times around 97–98 seconds and near 108 seconds. Third, the testing times are bi-modal, with two distinct peaks and fourth, this data is positively skewed, with a long tail stretching into the upper values.

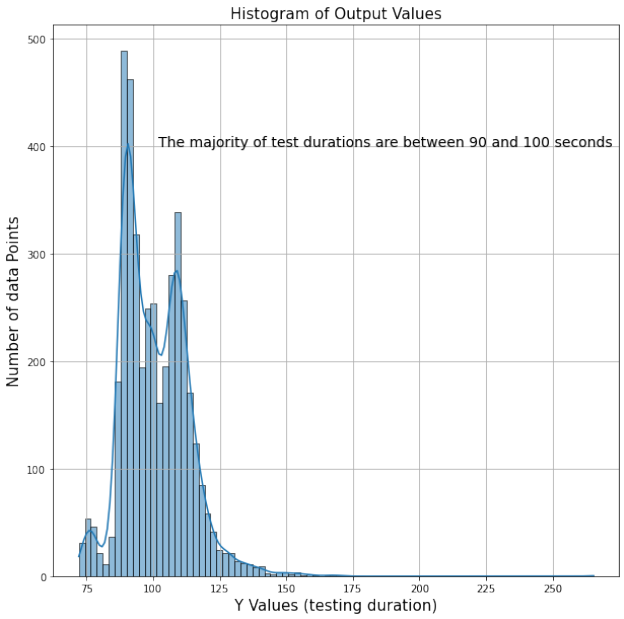

Figure 6. Distribution of output variable

5 Model Building

We have used multiple models and a thorough comparison is done among each one of them, We have used highly advanced computational software MATLAB to perform training and model evaluation. It provides various different techniques and models. With their easy syntax and self explanatory documentation [3], it was easy for us to come up with results. We also performed a comparison study for both raw and cleaned data.

5.1 Linear Regression

Linear regression is one of the most popular models to predict a continuous response variable based on the predictor variables [4]. The model fits the training data such that it generates a equation for predicting the response variable by inputting the predictor variables. The coefficients of the predictor variables are found by the model. One common approach is to generate a best fit line for the data such that it gives the least possible mean square error. A linear regression line has an equation of the form Y = a + bX, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when x = 0). The above equation would resemble prediction equation with one prediction variables. In our model we have a very large number of prediction variables. The model

we develop aims to predict the testing time based on the features of the car. The linear regression model was fitted on the training data and the performance of the model was tested on the testing data. Root Mean Square Error (RMSE) and computational time were found to evaluate the prediction accuracy and computational load of the model. The model was made for two types of data- raw data and clean data. The clean data had some of the features and samples removed based on some data visualisation observations. The 7 shows the plot of predicted vs actual values for the raw data. The RMSE value for testing on testing data is 10.145 and computational time was 0.254 seconds.The 8 shows the plot of predicted vs actual values for the clean data. The RMSE value for testing on testing data is 10.149 and computational time was 0.289 seconds.

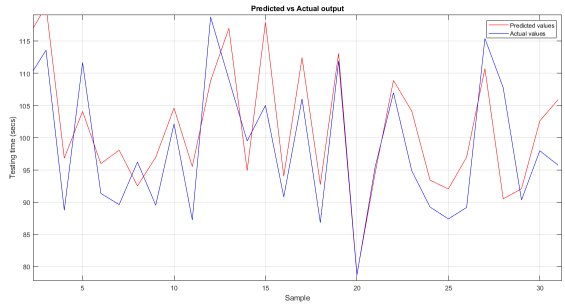

Figure 7. Predicted vs Actual Values for Regression Model (Raw Data)

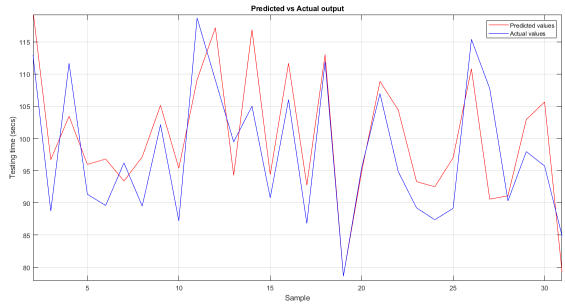

Figure 8. Predicted vs Actual Values for Regression Model (Clean Data)

From the plots and the performance values there is no significant difference between the raw and clean data sets. Both the models give a reasonably good prediction. Some of the advantages of these linear regression models are that they are easy to interpret and relate to the real world problem for decision makers and they are not computationally fast.

5.2 PCA along with Linear Regression

Data which have very high number of dimensions (predictor variables) tend to be computationally expensive. Most of the times these high dimensional variables will have many variables which are highly correlated and hence redundant. Thus reducing the dimensions would not lead to high loss of information. Also sometimes reducing the dimensions can make data which is difficult to understand and interpret more clear and interpretable. The model also becomes more stable by reducing the dimensions of the data. There are a lot of dimensional reduction techniques available. One popular technique is Principal Component Analysis (PCA) where the data is reduced to dimensions where maximum amount of variation of information is captured. This dimensionally reduced data can be used for further analysis like prediction. We are trying to reduce the dimensions of the car feature data and then use the transformed data for building a linear regression model to make predictions of the testing time. Both the raw and clean data sets were reduced to different values of lower dimensions and linear regression models were fitted on the reduced data and predictions were made on testing data set to find the RMSE values. The Figure 9 and 10 show the plots of predicted and actual values for linear regression models after reduced to 5 and 100 principal components from raw data.

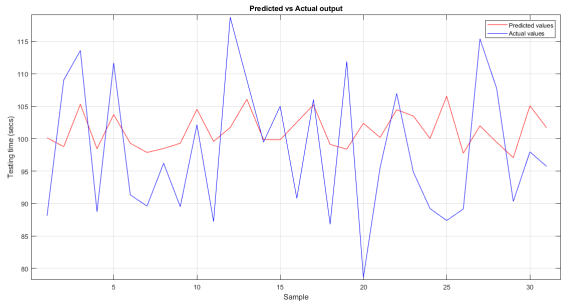

Figure 9. Predicted vs Actual Values for Regression Model after dimensional reduction (reduced to 5 principal components) using PCA (Raw Data)

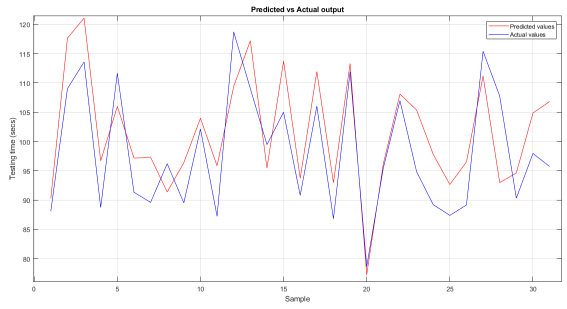

Figure 10. Predicted vs Actual Values for Regression Model after dimensional reduction (reduced to 100 principal components) using PCA (Raw Data)

The Figure 11 and 12 show the plots of predicted and actual values for linear regression models after reduced to 5 and 100 principal components from clean data.

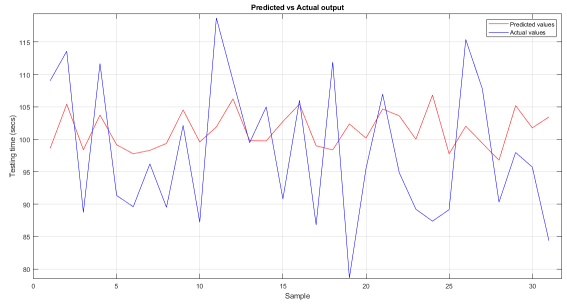

Figure 11. Predicted vs Actual Values for Regression Model after dimensional reduction (reduced to 5 principal components) using PCA (Clean Data)

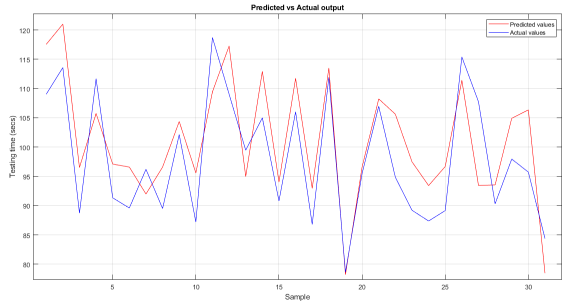

Figure 12. Predicted vs Actual Values for Regression Model after dimensional reduction (reduced to 100 principal components) using PCA (Clean Data)

The plots show that both raw and clean data models give fairly good prediction at higher values of principal components i.e. 100. For lower value of principal component i.e. 5 components both the models give a bad prediction as there is very low information available to give a good prediction. The 13 shows the RMSE testing error of linear regression model of the dimensionally reduced clean data. From the table it can be seen that as number of principal components increase the RMSE goes down. But after a certain point the RMSE increases with number of principal components. This shows that the model starts to overfit the data and should be stopped at that point. Similar observations can be made for the raw data set model. The clean data model had slightly lower RMSE when compared to the clean data models.

Number of Principal Components	Testing Error (RMSE)
5	11.6008
10	9.7568
15	9.0420
20	8.0628
25	7.9785
50	7.9426
100	7.6378
150	7.6558
200	7.7296
300	8.0361

Figure 13. Summary of RMSE error for different number of principal components for linear regression model for dimensionally reduced clean data

The 14 shows the plot of RMSE against number of principal components.

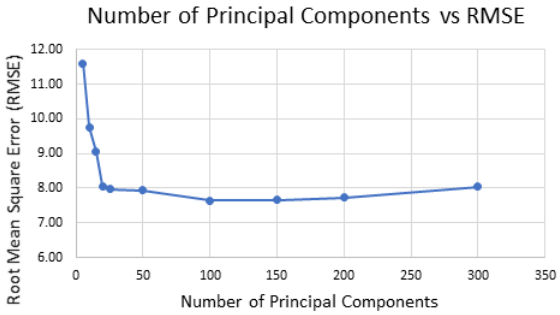

Figure 14. Number of principal components vs RMSE for dimesnionally reduced PCA model after linear regression (clean data)

It can be seen that initially there is a drastic reduction in RMSE for additional principal component added to the

model. But after some point the rate of drop in RMSE decreases. Beyond this point adding many additional principal components would make only a small difference in RMSE. Similar observations can be made for the raw data set model. Hence there has to be balance between RMSE and model simplicity (or storage) when deciding the number of principal components and RMSE. The PCA model reduces the storage space and also makes further analysis faster after dimensional reduction. It gives fairly good prediction if the right number of principal components are chosen. One disadvantage of the model is that it can be difficult to interpret and relate the solution of the model to the real world problems.

5.3 Gaussian Process Model

Gaussian Process Model is a stochastic involves a collection of collection of random variables such that every finite collection of those random variables has a multivariate normal distribution, i.e. every finite linear combination of them is normally distributed. Since they involve multiple variables they have flexibility highly varying data sets. Because of their flexibility they can be found in a variety of applications. We try to use Gaussian Process Model to predict the testing time of cars based on their features.

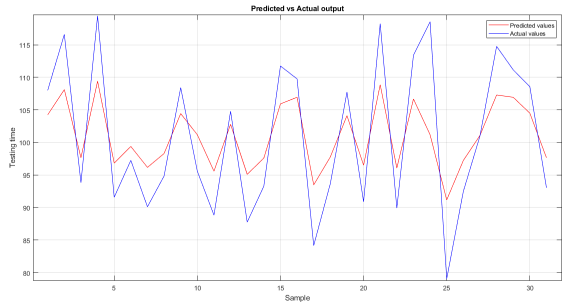

Figure 15. Predicted vs Actual Values for Gaussian process model (Raw Data)

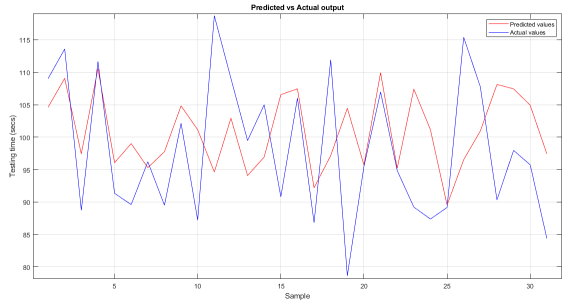

Figure 16. Predicted vs Actual Values for Gaussian process model (Clean Data)

The 15 and 16 show the predicted vs actual values for Gaussian process model for the raw and clean data set respectively. The RMSE value for the raw and clean data models were 5.008 and 4.7950 for the raw and clean data respectively. This shows that performance of the model improved because of the data cleaning. Both the models had computational time around 17-18 seconds. The good performance of the model is due to the structure of the model which allows a lot of flexibility for the variables and can fit the variation in the data very well. However it can have higher computational load and not as simple to interpret as the linear regression models.

5.4 Relieff Algorithm along with Linear Regression

One more way to address the issues discussed in Principal Component Analysis section is feature selection. Instead of reducing a high dimensional data to lower dimensions, a subset of the features in the data is selected and is then used for further analysis. This reduces computational load of the model and also sometimes improves the prediction accuracy of the model as sometimes the additional features may have negative impact on the prediction. One such feature selection method is Relieff algorithm. It computes the nearest neighbors to each data point and based on the distances assigns weights to the features in the data. The features which are important in describing the data more and hence required for prediction are given higher weights. Then a specified number of features are chosen and the chosen feature data is used for linear regression model building for our application. The models were tried for different number of features chosen for both the raw and clean data. The 17 shows the predictor ranking for the clean data model with the corresponding weight of the predictors indicated by the height of the bars.

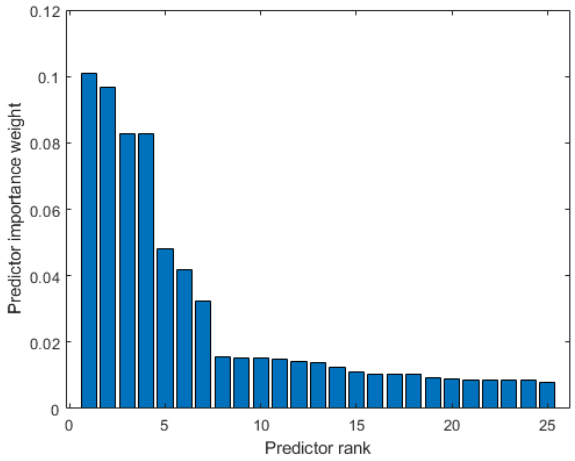

Figure 17. Predictor ranking of Relieff algorithm with the corresponding weights using Clean Data

The 18 and 19 show plot of prediction vs actual values for the linear regression model using the top 2 predictors and top 20 predictors respectively in the raw data set.

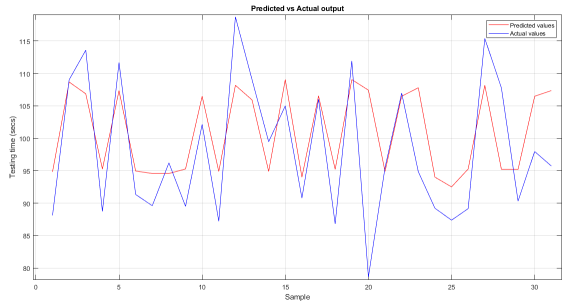

Figure 18. Predicted vs Actual Values for Regression Model using top 2 predictors chosen by Relieff Algorithm for Raw Data

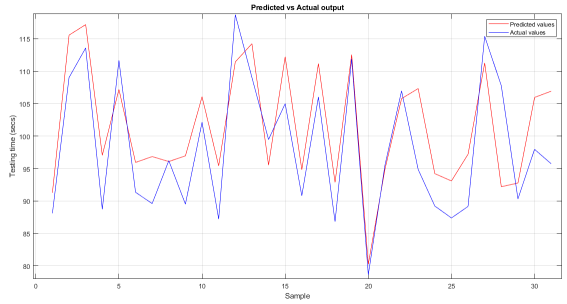

Figure 19. Predicted vs Actual Values for Regression Model using top 20 predictors chosen by Relieff Algorithm for Raw Data

The 20 and 21 show plot of prediction vs actual values for the linear regression model using the top 2 predictors and top 10 predictors respectively in the clean data set.

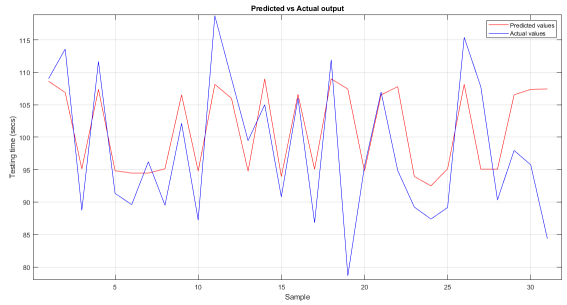

Figure 20. Predicted vs Actual Values for Regression Model using top 2 predictors chosen by Relieff Algorithm for Clean Data

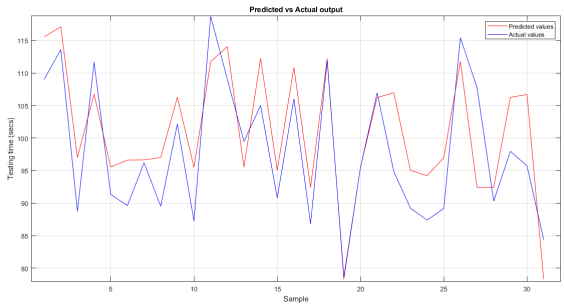

Figure 21. Predicted vs Actual Values for Regression Model using top 10 predictors chosen by Relieff Algorithm for Clean Data

It can be seen from the plots above that the RMSE reduces as number of features in the model increases for both the raw and clean data models. But after a certain point they start increasing again. This is a typical overfitting problem and the number of features to be included should be chosen properly to avoid this issue.

Number of features	Testing Error (RMSE)
1	9.9921
2	9.9938
5	9.1908
10	7.4593
15	7.4849
20	7.4842
25	7.4855
50	7.5084
100	7.5484
150	7.7160
200	7.8125

Figure 22. Summary of Number of features vs RMSE for Linear Regression with Relieff Algorithm

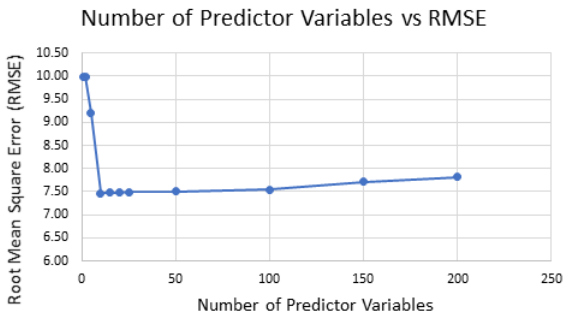

Figure 23. Number of predictor variables vs RMSE for Relieff algorithm using Clean Data

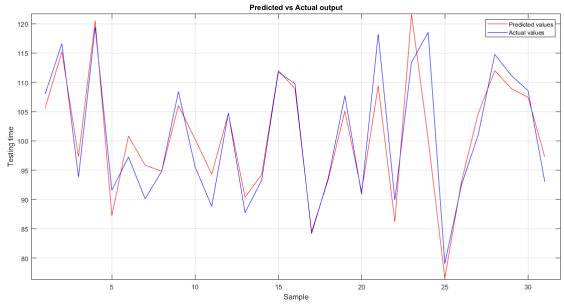

Figure 24. Predicted vs Actual Values for Regression Model for 50 learning cycles in Ensemble method algorithm for Raw Data

The 22 shows the summary of results of the models showing number of predictor variables vs RMSE for clean data. The 23 shows the plot of the same. Both clean and raw data models show the same observations. The clean data model had slightly better RMSE performance values. Relieff models can be computationally expensive as they took around 35-40 seconds for computations based on the number of predictors chosen. But the advantage is that they reduce the number of predictors drastically making the model structurally simpler while also saving storage space. Another major advantage is that the model is easily interpretable and relatable to the rel world problem unlike PCA or other dimensional reduction models.

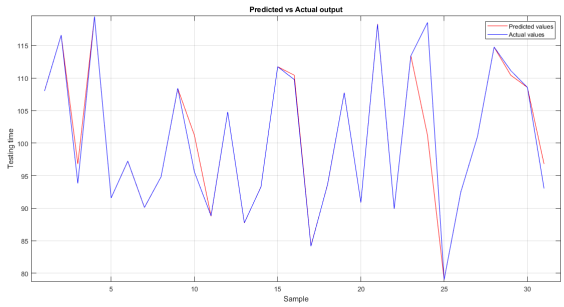

Figure 25. Predicted vs Actual Values for Regression Model for 1000 learning cycles in Ensemble method algorithm for Raw Data

The 26 and 27 show the plot of prediction vs actual values of the ensemble model for 50 and 1000 learning cycles respectively using the clean data.

5.5 Ensemble Methods

Sometimes when solving a complex problem, developing a single model may or may not give a reasonable or optimal performance. Ensemble method is one technique to overcome this issue. Ensemble methods create multiple base models and then combine them to produce improved results. The improved results can be reduced variation, increased prediction accuracy and reduced bias.The main idea is to combine many weak models so that the resulting model is powerful one. The ensemble method algorithm was tried out for both the raw and clean data for different learning cycles. The ensemble method used was Least-squares boosting (LS-Boost). The 24 and 25 show the plot of prediction vs actual values of the ensemble model for 50 and 1000 learning cycles respectively using the raw data.

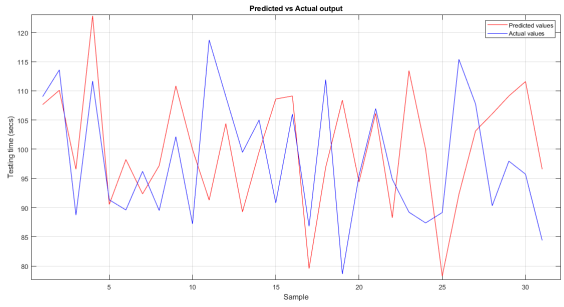

Figure 26. Predicted vs Actual Values for Regression Model for 50 learning cycles in Ensemble method algorithm for Clean Data

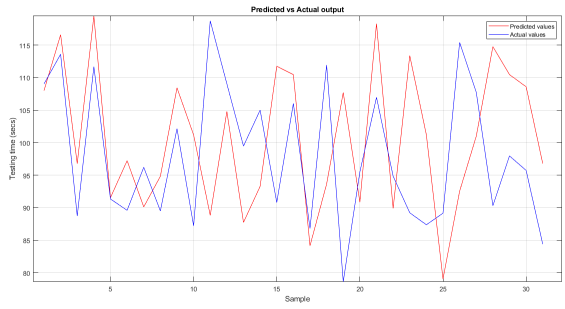

Figure 27. Predicted vs Actual Values for Regression Model for 1000 learning cycles in Ensemble method algorithm for Clean Data

The above plots show that as number of learning cycles increase the predictions become worse for both the raw and clean data models. Both the models show similar prediction accuracy for a given learning cycle.

Number of learning Cycles	Training Error (RMSE)	Testing Error (RMSE)	Computational Time (secs)
50	4.3464	8.8573	10.6471
100	3.2332	8.9817	16.9045
500	2.2052	9.0744	86.8648
1000	2.1884	9.0884	176.4678

Figure 28. Summary of results of Ensemble models with training error, testing error and computational time for different learning cycles using clean data

The 28 shows the summary of results containing training error, testing error and computational time for different learning cycles of Ensemble model using clean data. It can be seen that the computational time increases as the number of learning cycles increases.

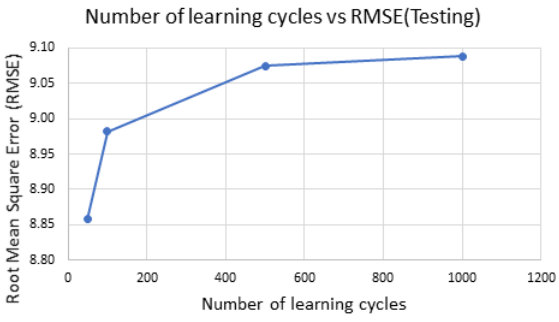

Figure 29. RMSE of testing prediction for different learning cycles of Ensemble model using clean Data

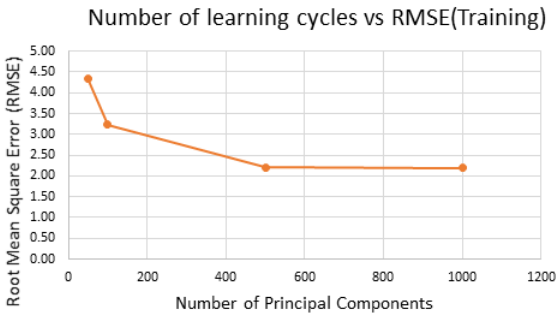

Figure 30. RMSE of training prediction for different learning cycles of Ensemble model using clean Data

The 29 and 30 shows plots of RMSE vs number of learning cycles for Ensemble model using clean data for training and testing data respectively. It can be seen from the plots that as the number of learning cycles increases the training error reduces and testing error increases. This is a typical case of over fitting seen in both the data sets (raw and clean) and the right number of learning cycles should be chosen to optimize the model. One disadvantage of the ensemble model is the computational load of the model is significantly higher than simple linear regression. It should be noted the Ensemble Regression model gives better prediction accuracy than the simple linear regression model. Another important advantage of this model is that the final model is very simple and is easily interpretable by the decision makers.

6 Model Evaluation

Based on the models discussed, output plotted, performance measures computed and results summarised the models can be evaluated against each other and comparisons made. Although the model was simple and computationally less expensive, simple linear regression models are not recommended as they gave the worst prediction accuracy among all the models. Gaussian process models gave the best possible prediction accuracy when compared to all the models. However they are significantly computationally expensive and more complex than linear regression model to interpret. PCA + Linear regression model gave good prediction results when appropriate number of components are chosen and also has reduced dimensions. However, these models may sometimes be computationally heavy and be difficult to interpret. Ensemble models gave good prediction results and the final model is also easily interpretable. However the right parameters need to be chosen to optimize the model. The Relieff Algorithm + Linear regression model performed well and gave good predictions with reduced number of predictor variables. It may sometimes be computationally more expensive than simple linear regression model, but has very

good interpretability. PCA + Linear Regression, Relieff Algorithm + Linear Regression and Ensemble methods are prone to overfitting issue and care must be taken while tuning the parameters to overcome the issue. All the models had beter or same performance when tried with the cleaned data as compared to the raw data. The discussed models could be further improved by optimally tuning the parameters, trying a combination of these models and training them on larger data sets. Among the models tried we recommend using Relieff Algorithm + Linear Regression model as it gives good prediction accuracy with reduced dimensions and is not computationally too expensive. If computational load is not a problem we recommend using Gaussian Process Model as it has very good prediction accuracy.

7 Real-world Insights

Some of the real world insights learned in this project about the problem will be discussed in this section. Initially we visualized the data and based on the observations we cleaned the data which were outliers and features which had almost the same entry for all the samples (features with low variance). Thus we can conclude that some of the outliers in the real world scenario would be due to wrong data entry or some calibration issue which needs to further investigated. Also some of the features are same for all the types of cars. So, having that information does not help in making decisions or predictions which can reduce testing time. Good prediction results were obtained with reduced dimensions which shows that by knowing only some features of the car we can make good predictions regarding the testing time. With more data being used to train the model the model prediction accuracy increased. This shows that as more cars are produced the personnel making decisions regarding testing will become more familiar with the process and have more knowledge to make good prediction.

8 Lessons learned

We learned overall understanding of manipulating data and handling of data. This project gave us hands on experience of an industrial level project that would definitely helped in understanding real world problem and how they can be solved using the knowledge of data mining and data processing. On an industrial scale many different types of graphs and plots are generated by different modern machines. Through this course we were able to learn the correct usage and applications of these different plots and how to interpret and extract useful information.

We both are from mechanical background and a course like Data Analytics is very new to us. We might be strong in our domain knowledge but in today's industrial age where everything is automated and controlled by computers. This makes it very important to us to learn new skills that can help us solving problems of this age, and this project successfully accomplish this task.

9 Appendices

The programming codes for this project are in the attached files listed below:

- A MATLAB file named "DA_Project_Raw_Data" which contains code for algorithms tried on raw data.
- A MATLAB file named "DA_Project_Clean_Data" which contains code for algorithms tried on clean data.
- File "Dhanapal_Jain_DataVisualization.ipynb" contains code and all the plots produced in section 2, 3, 4

Acknowledgments

Authors thank the support of Dr. Chandan Reddy along with the Teaching Assistants for the course CS-5525 for guiding us providing us with necessary information

References

[1] Daimler. 2017. Mercedes-Benz Greener Manufacturing. https://www.kaggle.com/c/mercedes-benz-greener-manufacturing/data.
[2] Mathplot Lib. 2007. Supervised learning. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning.
[3] Mathworks. 2007. Choose a Regression Function. https://www.mathworks.com/help/stats/introduction-to-parametric-regression-analysis.html.
[4] Scikit-Learn. 2007. Supervised learning. https://seaborn.pydata.org/.
[5] Seaborn. 2002. Visualization with Python. https://matplotlib.org/.
[6] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., Inc., USA.