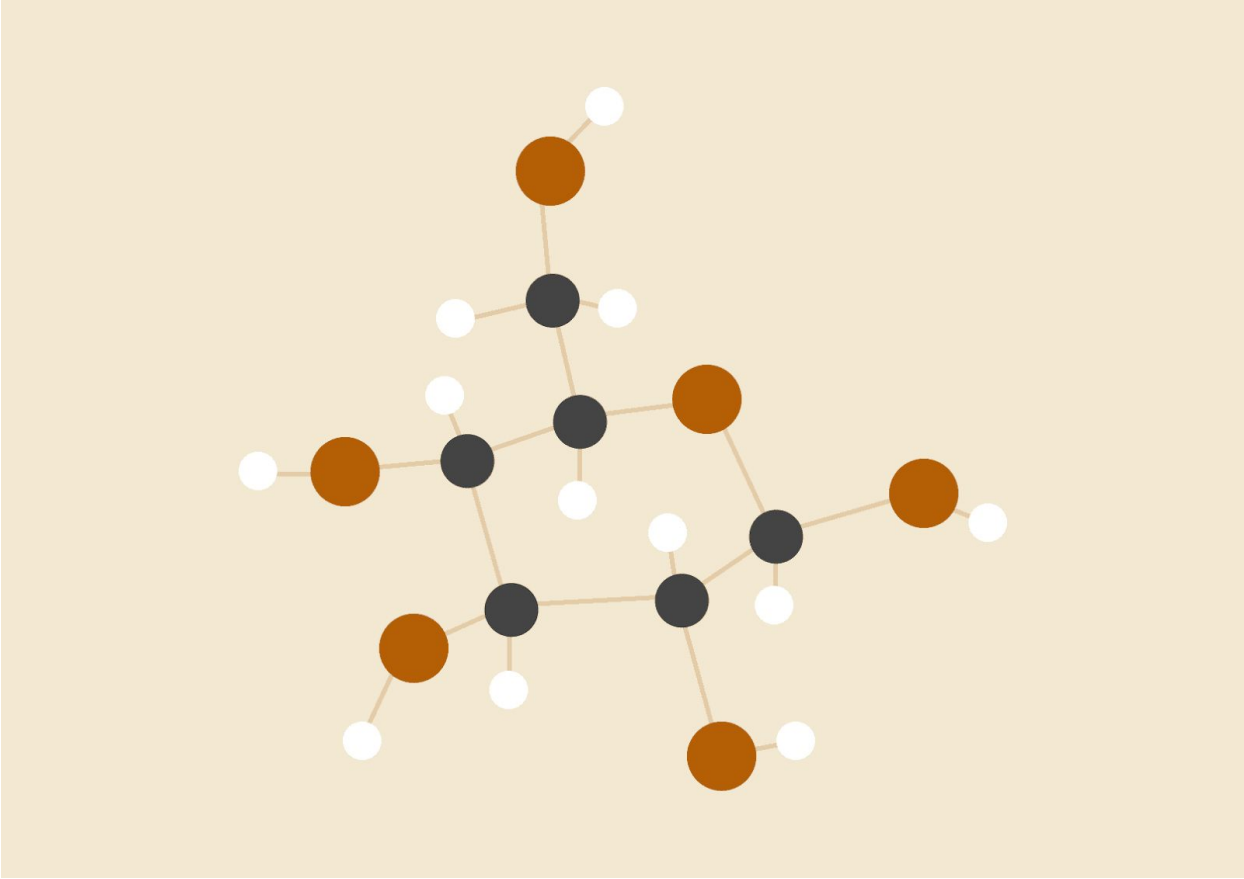
Quantitative Foundations

Project 1

Linear Feature Engineering



Ayush Kumar Shah (as1211@rit.edu)

Akib Shahriyar (as8751@rit.edu)

11.09.2020

Training Error: 33.211

Predicted Test Error: 51.664

Introduction

The target of this project is to deal with linear regression, overfitting, and feature engineering. We were provided with an unknown dataset consisting of 926 examples. We performed some analysis on the data and experimented with multiple models to fit the data using linear regression. The mean square training error on the training data using the best model was 33.211.

Data Analysis

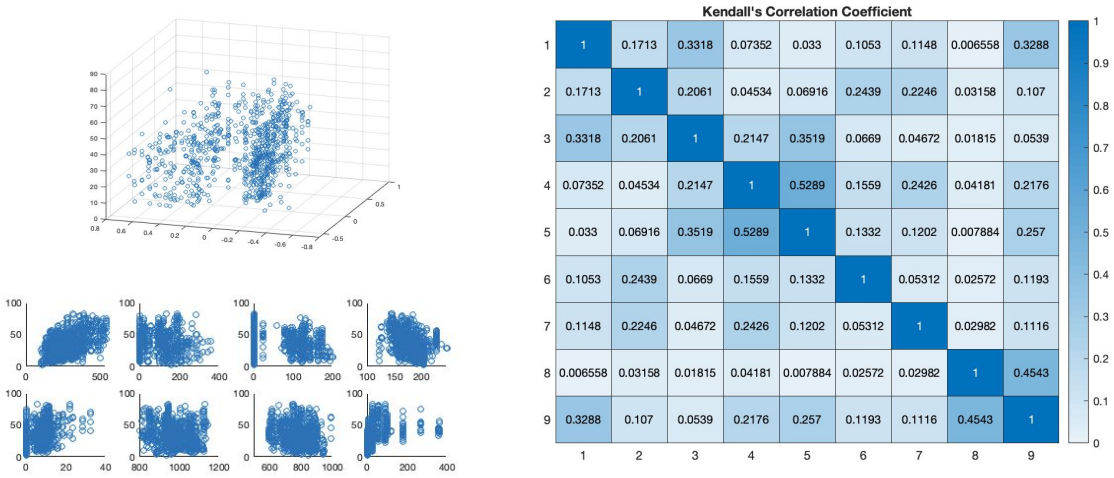


Figure 1: Data analysis results on the given dataset.

Initially, we analyzed the training data to understand the relationship between the features and the target. So, we visualized each feature's correlation with the target, tabulated Kendall's correlation coefficient values, and used PCA to reduce the 8-dimensional features to 2-dimensional, and plotted it with the target using 3D scatter plot. We could not benefit much from the data analysis due to the lack of domain knowledge.

Feature Selection

To find suitable features for our dataset, we experimented with multiple functions. We began with polynomial functions of orders 0 to 10. Using the K-fold cross-validation method (K = 10), we found that the 4th order polynomial function fit the data well as it had the least validation error.

However, we thought that other functions could improve accuracy. So, we created several custom functions to extract more features from the input data to create complex models to fit the data

well. We included fractional powers of the input features, including 1/10 to 9/10. Among them, the combination of 1/10 and 2/10 orders with the 2nd and 3rd order polynomial functions reduced the cross-validation error.

Next, we tested the negative powers of the input features and log functions. These types of models were unsuitable for our data as the resultant model was failing to learn from input data and was providing “Not a Number (NaN)” as output.

Finally, we tested different trigonometric functions such as sin(), cos(), tan() etc. Among them, sin() and cos() functions combined with the previously selected function had low validation errors. Ultimately, the final functions we selected having the least validation error was

$$\begin{bmatrix} X^2 & X^3 & X^{0.1} & X^{0.2} & 1 \end{bmatrix}^T$$

Prediction about Test Error

We have predicted the mean square test error to be **51.664.** We arrived at this prediction using K-fold cross-validation. We obtained the average mean square error of **51.664** on the data validation set using the best-selected model features. The training was done on the whole data using the same model.

However, during the K-fold cross-validation step before selecting the models, the features were divided into non-overlapping training and validation sets. The training was done on features created from the training set, and testing was done on different features built from the unseen validation set. Hence, the validation error approximates the test error on the unseen data well.

Dealing with Overfitting

We implemented K-fold cross-validation to select the best model for our given dataset. We chose the model with the least validation error as the best model and used it to train the whole data. Since we calculated the validation errors on the features not seen by the model during training, overfitting was reduced.

CONCLUSION

During this project, we implemented linear regression and performed feature engineering to the given data successfully. We also experimented with multiple models to decide on the best model to represent the data. We learned to implement different operations of linear algebra, model selection, and K-fold cross-validation efficiently in MatLab during the project.