Linear Feature Engineering

Abhinab Acharya, Jeevan Thapa

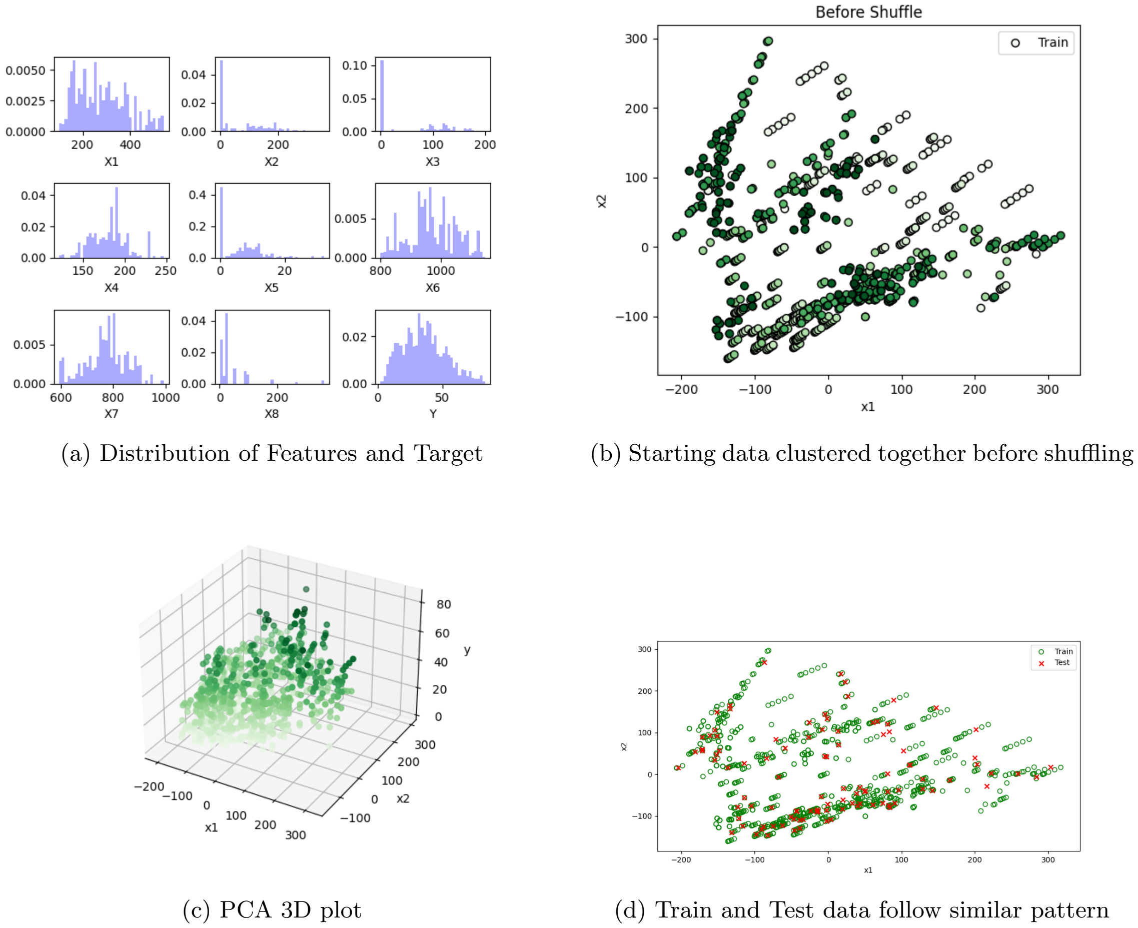
Sept 2022

Training Error: 32.59
Predicted Test Error: 37.46

1 Introduction

In this project, we fit a given dataset using Least Squares. We begin by analyzing the data, performing basis expansion, selecting the combination that performed best on validation data, and finally, reporting the overall training and test error.

2 Data Analysis



(a) Distribution of Features and Target

(b) Starting data clustered together before shuffling

(c) PCA 3D plot

(d) Train and Test data follow similar pattern

Figure 1: Data Analysis

There are 8 features and 926 data points. We plotted the distribution of each feature and target which helped us to get a general idea about the data. We also used Principal Component Analysis to check how many hidden features or components contribute to the information available in the data.

From figure 1a, we can see that for some features, there are a lot of data with zero value. Some features are somewhat uniformly distributed whereas the distribution is skewed for others. Figure 1b shows that the data is not well-shuffled before since the data points at the start of the rows are clustered together. Thus, this gives us the hint that shuffling the data might improve the performance of our model. The 3D plot of two high variance PCA components with the target is given in the figure 1c. It helps to visualize that the data points are not random and have some relationship with the target. Finally, the fourth figure 1d indicates that the test and train data points are from similar distributions and follow a similar pattern.

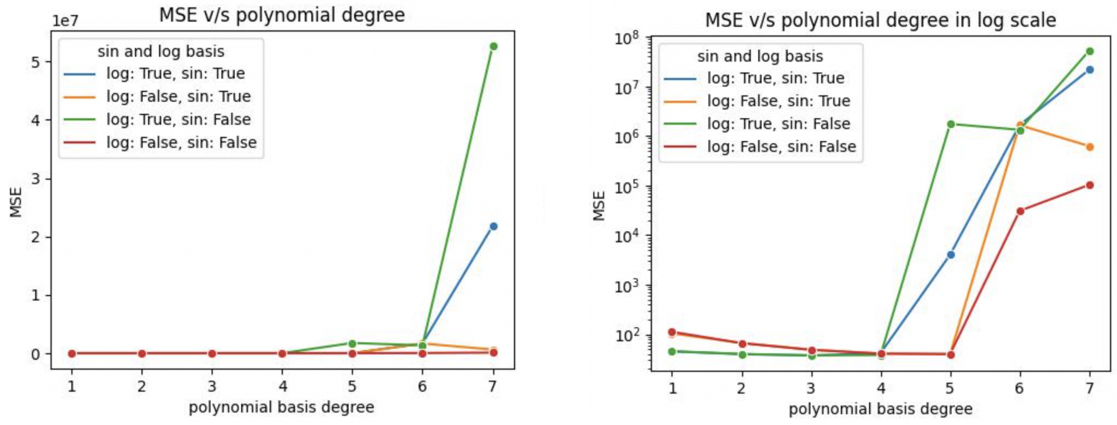
3 Feature Selection

We use three functional expansions for linear regression: polynomial with integer degrees, sin, and logarithm transformation. For polynomial expansion with degree d, given a feature x_i, we generate p polynomial features as in $\{x_i^p\}_{p=1}^d$. Similarly, for sin expansion, we simply use $\sin(x_i)$. We don't use the tan function since its output is not constrained as in sin; tan outputs infinity for $\pi/2$. And, for logarithm, we use a function $l(x_i)$ defined as $l(x_i) = \begin{cases} 0, & x_i \le 0 \\ \log_e(1+x_i), & x_i > 0 \end{cases}$.

We finally run experiments with different polynomial orders from 1 to 7, and with and without sin and log transformations and select the expansion with the least cross-validation error as the best model as described in 4 and 5.

4 Test Error Prediction

The training data was split into training and validation using K-fold cross-validation. And regression models were fit on the engineered features generated using the feature expansion described in 3. We obtained a minimum mean cross-validation error of 37.46 while using polynomial expansion of degree 3 along with sin and log transformations. This validation error has been reported as the prediction for test error.



(a) MSE v/s polynomial degree

(b) MSE v/s polynomial degree in log scale

Figure 2: Test losses for Linear regression for different expansions

5 Dealing with Overfitting

The training dataset consists of 926 data points only, which is less. Hence, a model with higher data complexity can easily overfit the training data and learn the noise components resulting in a lack of generalization. Hence, we rely on K-fold cross-validation to mitigate the issue. We used 5 as the number of folds for the task. It should be noted that the training data has been randomly shuffled before cross-validation so that the cross-validation results do not depend on the original ordering as shown by 1b. As observed in figure 2, the models with polynomial degree greater than 5 and 5 (with log term) have high test losses, showing the region of overfitting. The models with polynomial degree 3, 4 and 5 (without log term) perform relatively better. Finally, we chose for polynomial degree 3 along with sin and log transformations based on validation MSE results.