Quantitative Foundations
Project 1
Linear Feature Engineering

Image Adhikari and Suraj Poudel

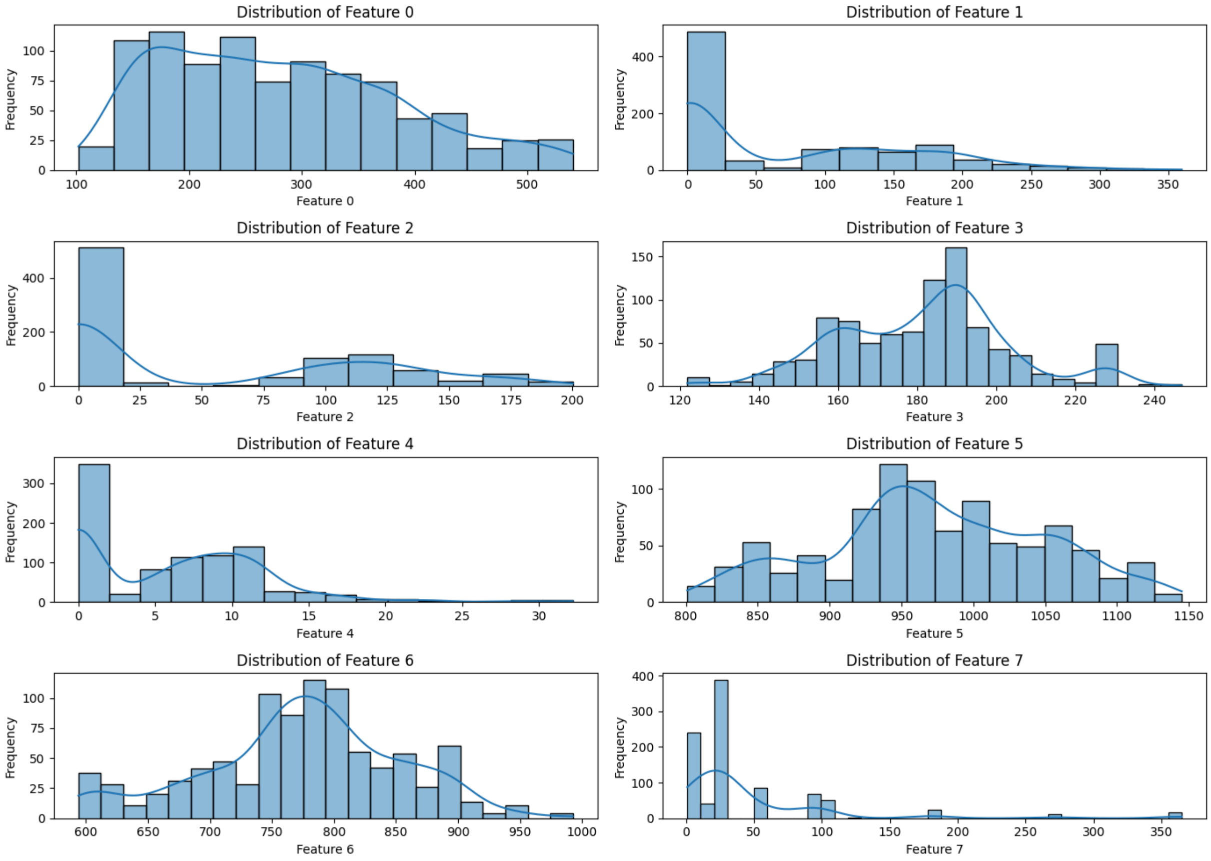
Code Repo: https://github.com/SrjPdl/Linear-Feature-Engineering

Train error: 35.42
Predicted Test Error: 67.59

Introduction:

In this project, we aimed to predict the output variable *y* from the given 8 predictor variables *x* using linear regression. We were given a dataset with 926 rows. We performed some data analysis and explored various techniques, including feature engineering and cross validation for managing overfitting, to improve model performance. Finally, we selected the best performing model to test the values given as test inputs.

Data Analysis:



From the data distribution, we can see that most of the datasets are right-skewed(Feature 1, 2, 4, and 7). This means most of the data points are aggregated in the range of lower values. Features 3, 5, and 6 have near normal distributions. Feature 0 has a more moderate right-skew and covers a larger range.

Feature Selection:

For each feature, we fit a polynomial model and calculate the mean squared error (MSE) between the predicted and actual values. The features are then ranked based on their associated errors, and the ones with the lowest errors are chosen as the best features. This method identifies which features contribute most effectively to reducing prediction error, ultimately selecting a specified number of top-performing features for further use. We have used 5 features (7, 0, 3, 4, 1) with lowest MSE as choosing either more or less number of features worsened the model's performance.

Search for best degree of polynomial

We tried to fit the model using various polynomial degrees and all of the features. This led us to a conclusion that the best performing degree of polynomial was degree 7. However, once we did some feature engineering and selected the top 5 features with the least amount of MSE, we concluded that the best performing polynomial degree was degree 6.

Prediction of Test Error:

We estimate the mean squared test error to be 67.59. This is the value of our validation error obtained using K-fold cross-validation. After selecting the optimal model features, we achieved this error on the validation set and this would be a good approximation of the mean squared error on the test set.

Overfitting Mitigation:

1. **Cross Validation:**
    - To make sure that the model's performance generalized well over various data subsets, we used k-fold cross-validation. This was done to reduce the possibility of the model overfitting to any particular subset of the data by dividing the data into separate portions and training it on several subsets.
2. **Feature Selection**
    - By carefully choosing the most relevant features and avoiding excessive polynomial terms, we tried to improve the generalization ability of the model. We observed that adding too many high-degree polynomial terms tended to increase training performance while degrading the validation performance, indicating overfitting.