# Quantitative Foundation

## Project 1: Report

Group Members:
Moiz Arif
Nilesh Kumar

**Mean Training Error = 36.029588**
**Mean Test Error = 40.369052 (For outliers, see Approach, point 7)**

## Problem Statement

You are given a dataset, consisting of 926 examples. Each example has 8 real-valued predictor attributes x ("regressors"), and a single real-valued dependent value y to be predicted. Your goal is to use this data to build a function that will predict values y for new data x. Specifically, we have more data that was generated from the same source. We will measure how well your predictor does on this new data.

## Approach

Here is the approach we took in order to solve the problem listed in the problem statement:

1. We start off the project by loading the training data (*traindata.txt*) into our system and separate out X and Y from the given dataset.
2. Once we have the data we try to figure out the best degree for our data taking care of the fact that we do not choose a very simple model that under fits our data and also avoid overfitting, we do so by performing K-fold cross validation and checking degrees starting from 1 up to 10.
3. As we ran this experiment for K=5, we realize that the best fit for this dataset is polynomial of order 3.
4. However, after experimenting with more values of K we realized that for k >= 7, polynomial of degree 4 is the best fit. In our understanding it is because with a bigger value of k means that there is more data for training and hence the overfitting starts at a higher order polynomial.
5. After establishing the above results, we tried to test some other functions along with a bias term. Some functions that we tried include: sin, cos, tan, log, sigmoid, relu, roots, negative of x and absolute. Out of these functions only sin gave a slight improvement in our mean cross validation error, Relu and absolute did not contribute anything because all the features are already positive so it was basically equally to passing X.
6. After this we use the finalized model ($ax + bx^2 + cx^3 + \sin(x) + 1$) to find out the predicted values (y) on the test data (*testinputs.txt*).
7. We were inclined to choose our test error using our cross validation error but we realized that we were overestimating our test error because we were not looking at the entire dataset when we had a lower value of K. So we choose this approach called **LOOCV (Leave One Out Cross Validation)**. However, the error we got was averaged over 926 examples but in our test set there are only 103 examples so any outlier in our test set will have more effect on our error because of fewer example so we expect a mean test error +2-3 the error we got with 926 examples.

## Experiments and Results

In order to find the features that minimize the error we performed the following experiments:

| Experiment | Experiment Description | Mean Cross Validation Error (K=7 otherwise specified) |
|---|---|---|
| 1 | $ax + b + \sin(x)$ | 127.34 |
| 2 | $ax + bx^2 + c + \sin(x)$ | 79.87 |
| 3 | $ax + bx^2 + cx^3 + d + \sin(x)$ | 62.93 |
| 4 | $ax + bx^2 + cx^3 + dx^4 + e + \sin(x)$ | 54.17 |
| 5 | $ax + bx^2 + cx^3$ | 64.272954 (K = 5) |
| 6 | $ax + bx^2 + \sin(x) + x^{1/2}$ | 67.184519 (K = 5) |
| 7 | $ax + bx^2 + cx^3 + \sin(x) + d$ | 62.848697 (K=5) |
| **8** | $\mathbf{ax + bx^2 + cx^3 + \sin(x) + d}$ | **40.369052**<br>**LOOCV (Leave One Out Cross Validation)** |

We ran numerous other experiments with different function as mentioned in our approach but here we have quoted the results that gave us comparable results. We also choose different values of K but we realized that after K=7 the best fit order did not change.

## Conclusion

In this project we learned how to deal with hyper parameters. While choosing order for the polynomial and using other functions along with value or ‘k’, we realized that things are quite different in theory and as compared to practice. We ran into some problems like numerical instability, overfitting and even changing trends of fitting with varying values of ‘k’ in cross validation which helped in understanding the tradeoff that we make when choosing different hyper-parameters for our models. We also tried **feature engineering** by eliminating features that had lower correlation coefficient with y, starting with the lowest but that did not give any performance gains.

*Note: We have added graphs on the next page for reference only, these will be displayed when our code is run.*

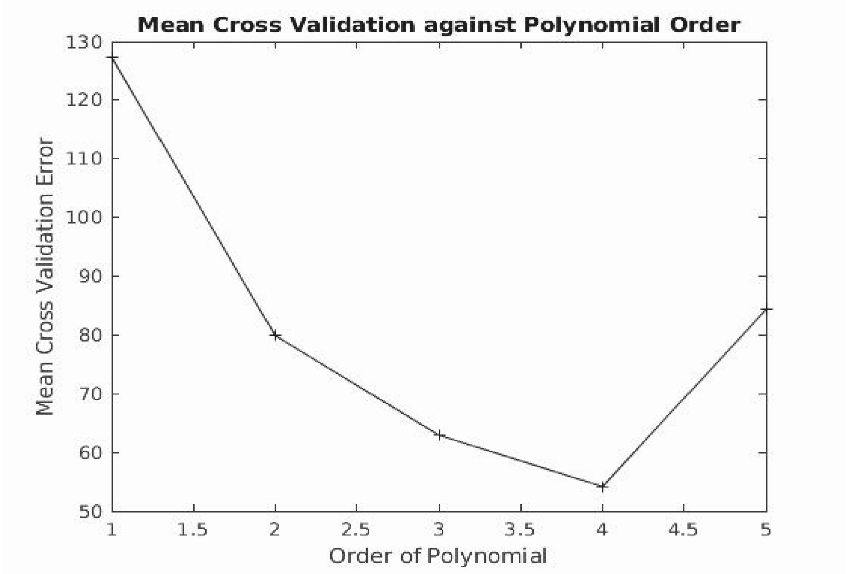

*Figure 1: Cross Validation Error trend on increasing order of polynomials K=5*

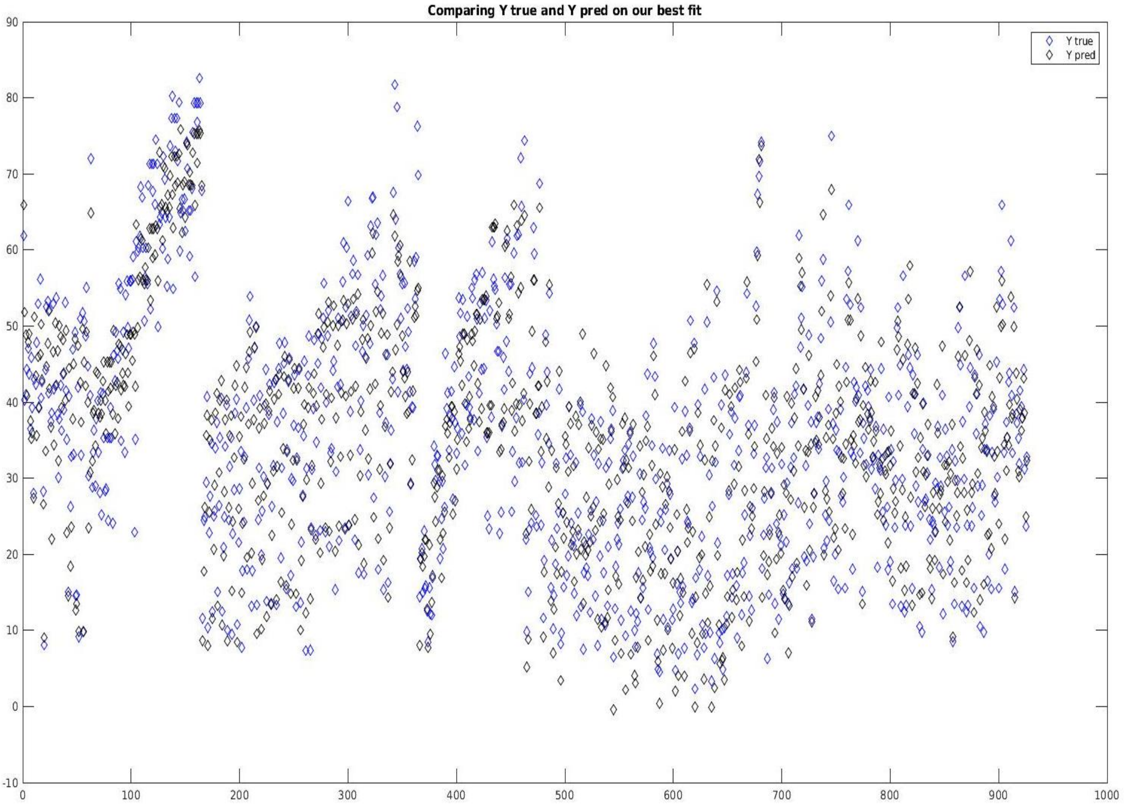

*Figure 2: Comparing our predicted value of y for training set against y predicated*