Project 1

Team Name: bitnetwork

## Members:

|  |  |  |
| --- | --- | --- |
| Name | Email | % Contribution |
| Diego Franchi | [diegofranchi@csu.fullerton.edu](mailto:diegofranchi@csu.fullerton.edu) | 100 |
| Imran Gosla | [Igosla0@csu.fullerton.edu](mailto:Igosla0@csu.fullerton.edu) | 100 |

Software and Tools used:  
For our project, we used the **Python Machine Learning Package**. By using **Scikit Learn** for the machine learning library, **Numpy** for math manipulation, and **pandas** for data containers we generated two models, the linear and non-linear models for the given dataset. We used **Cross Validation** to find the **optimal order of polynomial for the given dataset**.

## Analysis:

## Linear Model:

### Uncertainty of Model:

Mean Absolute Error: 0.833082971233712

Mean Squared Error: 1.2320182268783093

Root Mean Squared Error: 1.109963164649309

## Non-Linear Model:

### Uncertainty of Model:

## Process used for developing the model

We developed our model using Sklearn, Numpy, Pandas, and Scipy within Python. We began by importing the dataset into a pandas data frame object. After importing the data, we separate the feature columns from the dataset and split the data into a training and test set to create a model and test the accuracy. We then apply the linear regression and fit the training data to it. From there, our program predicts using the training data and we calculate the root mean squared error in order to find the uncertainty of the model (lower is better).

## Justification for why we have the best models

Linear Regression model without removing any of the features results in a more accurate model because it minimizes the root mean squared error (using cross validation reduces the variance associated with a single trial of train/test split). Also, by looping through different order of polynomials when testing, we chose the one with the lowest root mean squared error.