Problem Set 3 Part 1

By: Kali Benavides

Collaborated with: Alexa Canaan

Problem 3.1

A)

To perform linear regression on this data I split the data set into a training set, validation set and test set.

Training set was 550 points from 1958-2003, the validation set was 74 data points (2003-2010), and the test set was also 74 data points (2010-2016).

Coefficients: Intercept =0 and Slope = 1.368 (Fitted to the training data)

Metrics selected to quantify goodness of fit:

Root Mean Square Error: Measures the accuracy of the prediction of the model. The closer the value is to zero, the more accurate the model is at predicting the correct value. But can be affected by outliers because the residual is squared before taking the average.

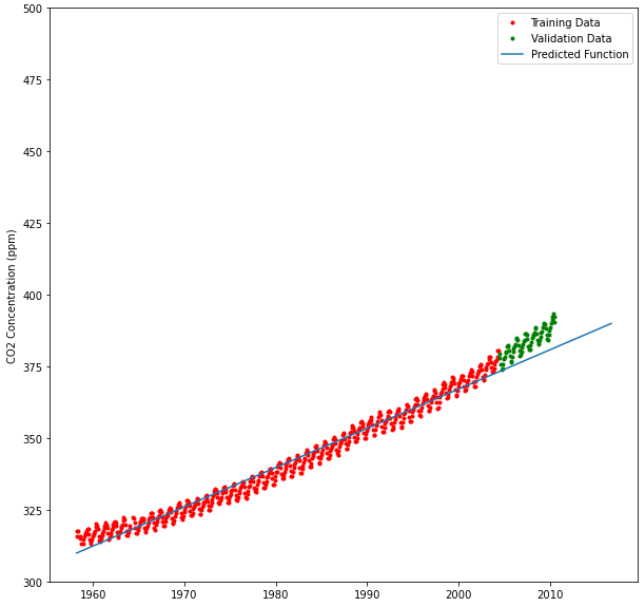
Mean Absolute Percentage Error: This calculation is more robust to outliers because it looks at the absolute value of the difference. Also, because it is divided by the actual output variable, it is incorporating the magnitude of the data. This calculation would be undefined for data points that equal zero, but we don't have that in this data set.

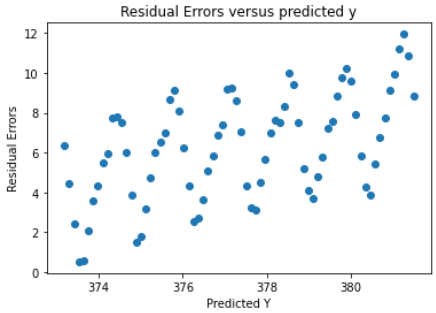
Goodness of Fit (with validation data set)

- Root Mean Square Error 6.783
- Mean Absolute Percentage Error= 1.625%

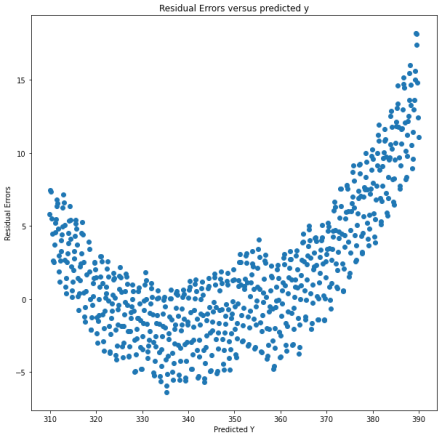
Goodness of Fit with Training Data Set:

- Root Mean Square Error 2.887
- Mean Absolute Percentage Error 0.691%
  *The linear model fit the training data far better than the validation data set. This shows that this model is not very accurate at predicting values from this data set.





*Residual error of just the validation training set shows that the residuals are not randomly distributed about zero. We see the seasonality of the data coming through into the residual values.



*Residual error of entire training set shows that it is being influenced by a variable other than random error. The model is also worse at predicting the lowest and highest values for Y, this makes sense because it is a linear model and so is fitting an average slope to the entire model which doesn't take into account that the lower value data points taken in the beginning of the time period have a different slope than the data at the end of the time period which are of higher values.

```
#Plot the residual error and comment about it.
Residual_1 = yval - y_pred_val


MSE1 = (Residual_1**2).mean()
RMSE1 = np.sqrt(MSE1)
MAPE1 = ((np.absolute(Residual_1)/yval)*100).mean()
print('Mean Square Error', MSE1)
print('Root Mean Square Error', RMSE1)
print('Mean Absolute Percentage Error', MAPE1)
```

B) Quadratic Fit to data

Fitted using training data set.

Coefficients =[0.000e+00, -4.585e+01, 1.191e-02]

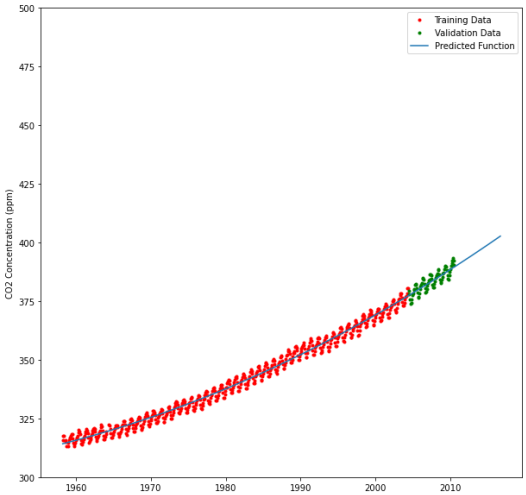
This model fit the data better than the linear model. There was also less of a difference between the goodness of fit of the model on the training data and the validation data.

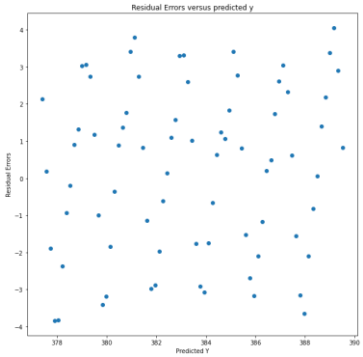
<u>Goodness of Fit with validation data set:</u>

- Root Mean Square Error 2.244
- Mean Absolute Percentage Error 0.509%

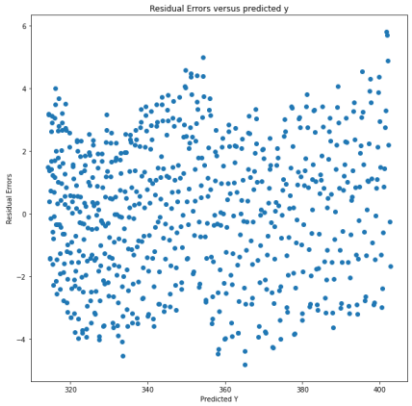
<u>Goodness of Fit with Training Data set:</u>

- Root Mean Square Error 2.183
- Mean Absolute Percentage Error 0.539%





Residuals for Validation Data Set plotted versus predicted values for the validation data set.



Residuals for entire data set plotted against the predicted y value for the entire data set. This shows that while the residuals are more centered around zero than in the linear model, there is still some behavior not explained just by random error. This is likely related to the seasonality of the data which we are not accounting for in this model.

C) Fitting data with a quartic polynomial model
Coefficients = [0.000e+00, -5.067e-04, -6.676e-01, 4.456e-04, -8.360e-08]

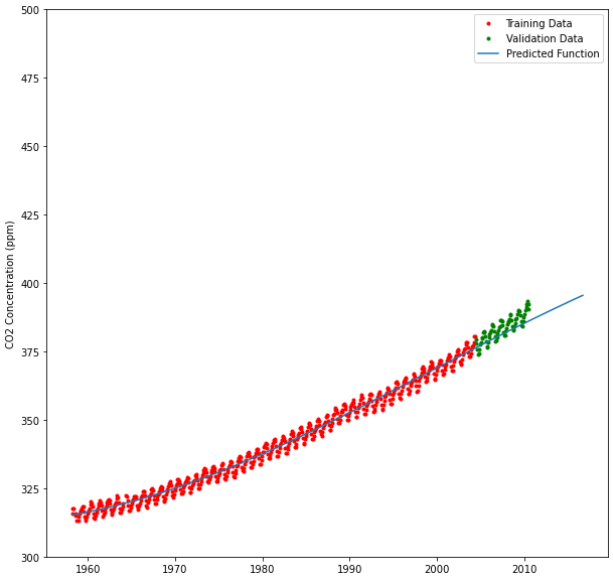
Trained with the training data set.
Tested with the validation data set.
<u>Goodness of Fit of Quartic Model with validation data set:</u>

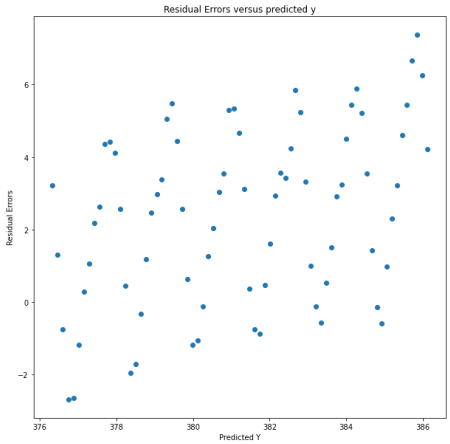
- Root Mean Square Error 3.366
- Mean Absolute Percentage Error 0.725%

<u>Goodness of fit with training data set:</u>

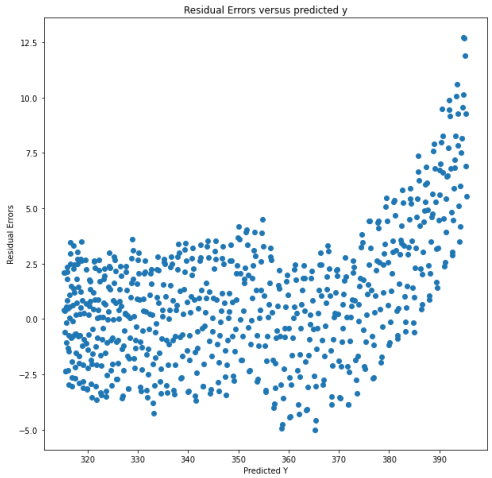
- Root Mean Square Error 2.145
- Mean Absolute Percentage Error 0.533%

The model fit the training data set better than the validation d ata set. It was not as good at predicting the validation data set as the quadratic model.





Above residuals for validation data set plotted against predicted values for the validation data set
Below, residuals for entire data set plotted against predicted values for the entire data set. Here we see that as we move to the higher predicted y values which make up our validation/testing portion of the data set, the residuals become more spread out. This indicates the poorer fit that this model has on the validation/testing portion of the data because it has overfit the training data set.



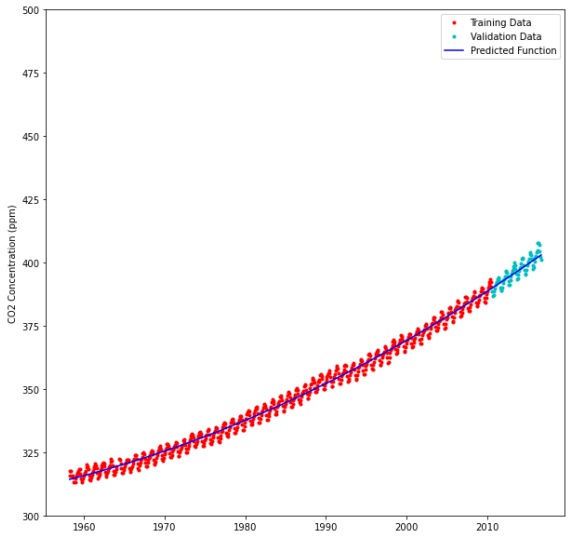
C) I would select the quadratic fit because it's goodness of fit for the validation data set is the best. The Root Mean Square Error and Mean Absolute Percentage Error for this method using the validation data set are lower than for the other methods.

The pros of using a higher order model are that they can more closely fit the training data set but are therefore prone to overfitting. This makes them poor models for new data if there is a trend that deviates from the training data set.

To select the order of the model, I would test all different models of varying orders using a portion of the data as a training data set. I would then determine the goodness of fit of each model using another portion of the data as a validation data set. The polynomial order which greatly reduces the error associated with the fit without overfitting is the model to select. You could graph the order of the polynomial against a goodness of fit metric to identify the point at which the fit has decreased the most similar to an elbow plot for the Kmeans clustering method.

Then once selecting the model, I would retrain the data set on the original training data set plus the validation data set. I can then test the data set on the test data set which the model has not yet seen.

See below graph of the quadratic model retrained on the training and validation data set.



<u>Goodness of Fit of Model with Training Data:</u>

- Root Mean Square Error 2.189
- Mean Absolute Percentage Error 0.535%

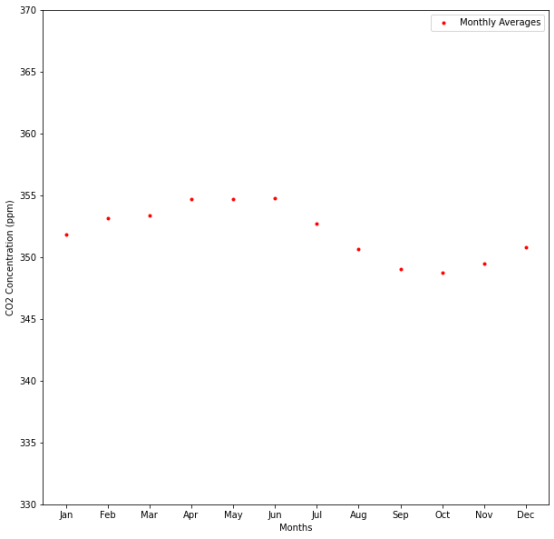
<u>Goodness of Fit of Model on Test Data:</u>

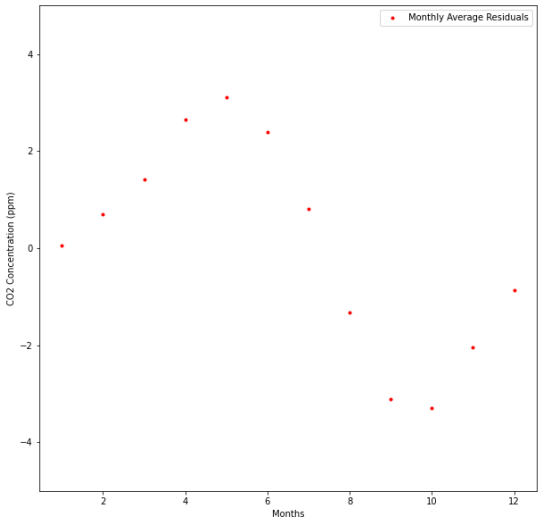
- Root Mean Square Error 2.419
- Mean Absolute Percentage Error 0.518%

D) Average value for each month:

Month	Average CO2 Concentration (ppm)
January	351.828
February	353.177
March	353.349
April	354.713
May	354.735
June	354.795
July	352.692
August	350.692
September	349.033
October	348.736
November	349.478
December	350.775

Seasonality graphed below.



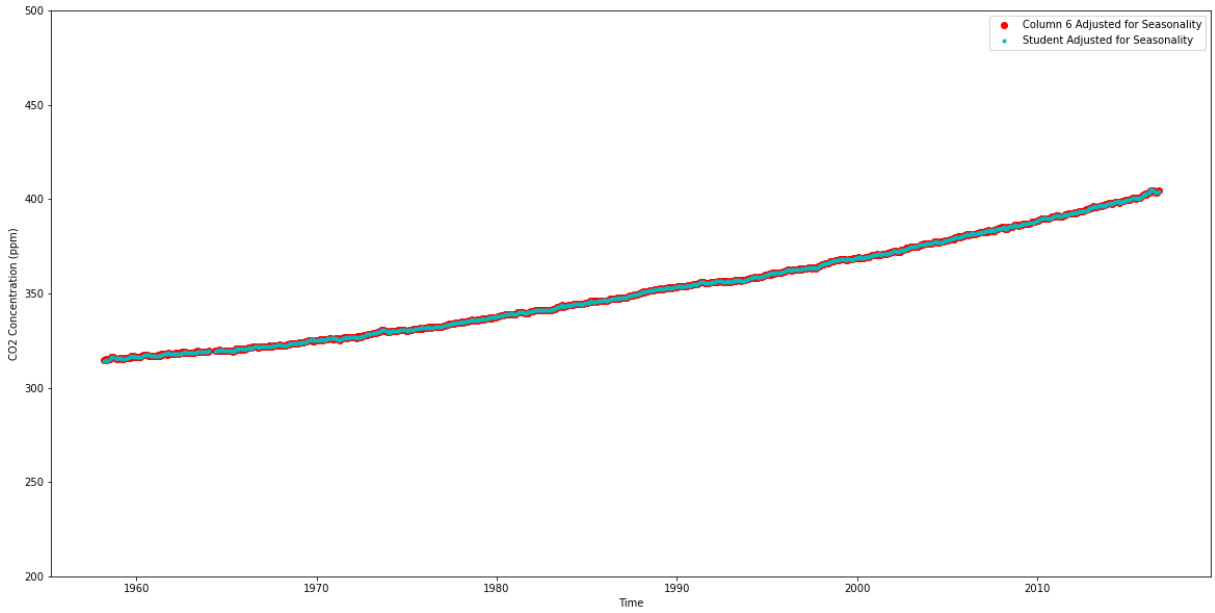


```
def CalcMonAvg(num):

    Mon_values = []
    Mon_count = 0
    for row in range(len(df_ML)):
        month = df_ML.iloc[row,1]
        val = df_ML.iloc[row,4]
        if month == num:
            Mon_values.append(val)
            Mon_count = Mon_count + 1

    Mon_average = sum(Mon_values)/Mon_count
    return Mon_average
```

```
def CalcResAvg(num, df, Monavg):
    Residual_values = []
    Residual_count = 0
    for row in range(len(df_ML)):
        month = df.iloc[row,1]
        val = df.iloc[row,4]
        if month == num:
            Res = Monavg[month-1]- df.iloc[row,11]
            Residual_values.append(Res)
            Residual_count = Residual_count + 1

    Res_average = np.mean(Residual_values)
    return Res_average
```

```
Monavg = []
for i in range(12):
    print(i)
    Monavg.append(CalcMonAvg(i+1))
```

```
Resavg = []
for i in range(12):
    print(i)
    Resavg.append(CalcResAvg(i+1,df_ML, Monavg))
```



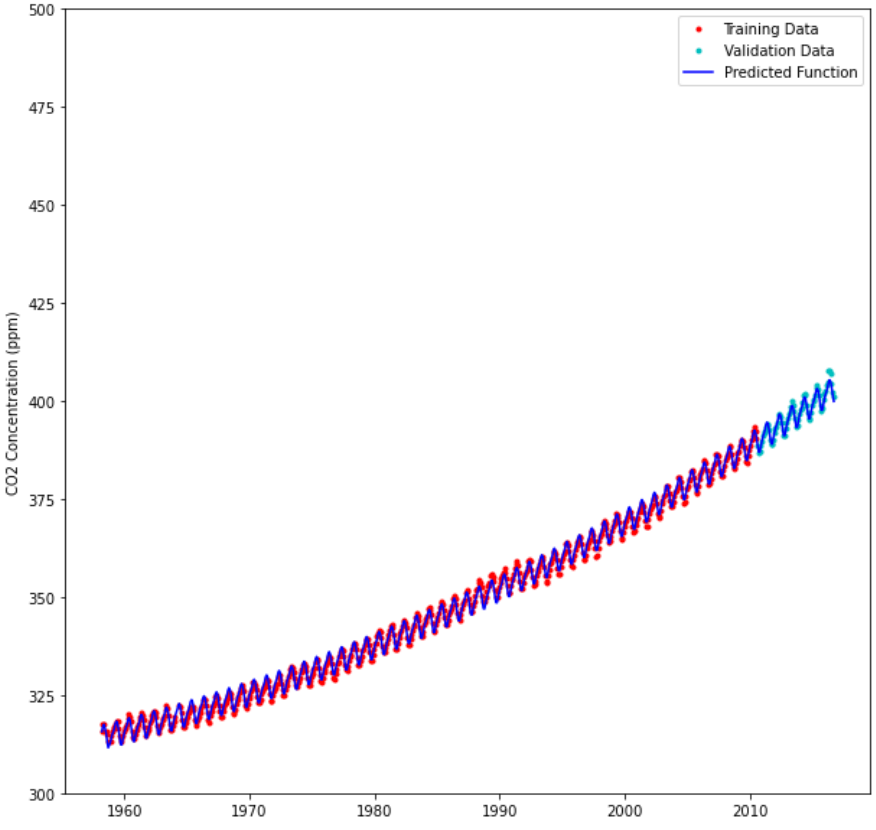
Above is the data in blue that I have adjusted to remove the seasonality compared to the data from Column 6 (in red) from the data set which was adjusted by the researchers. Both now show only the long-term trend in the data.

```python
#Remove seasonality

def SubMonAvg(Res, df):
    AllVal = []
    for row in range(len(df)):
        month = df.iloc[row,1]
        val = df.iloc[row,4]
        newval = val - Resavg[month-1]
        AllVal.append(newval)

    df['CO2 minus seasonality'] = AllVal
    return df
```

```python
SubMonAvg(Resavg, df_ML)
```

E) Plot the fit for $F_2(t) + P_i$

Below plotted is the quadratic model with the periodicity added to the model. Now the model is showing both the long-term trend and the seasonality trend. We can see the model now fits the data better (See below goodness of fit metrics). There is seasonal variation in the CO2 levels due to the lower levels of photosynthesis of plants in the winter. During the winter months, the respiration of CO2 outweighs the oxygen released during photosynthesis. Then during the summer months, there are lower CO2 levels as the plants grow more and utilize more of the CO2 in the air. Although the hemispheres have opposite cycles, because there is more land in the Northern hemisphere, it's growing cycle dominates producing the seasonal trend in CO2 we observe from the data. But, compared to the seasonal variation, the overall long-term trend is showing an increase in CO2 levels.



<u>Fit on the training data set for the model:</u>

- Root Mean Square Error 0.701
- Mean Absolute Percentage Error 0.164%

<u>Fit on the testing data set for the model:</u>

- Root Mean Square Error 0.774
- Mean Absolute Percentage Error 0.136%

<u>Fit on the entire data set:</u>

- Root Mean Square Error 0.709
- Mean Absolute Percentage Error 0.161%