Machine Learning (STQD 6024) Assignment 2

Muhammad Hilman Bin Rozaini (P121535)

## Objective and dataset description

The objective of the project is to find out the best Multi Linear Regression model to predict the Unified Parkinson's Disease Rating Scale (UPDRS) score, which measures the severity of the Parkinson's disease based on various features.

From the dataset, firstly it contains information from 20 individuals diagnosed with Parkinson's disease and 20 healthy individuals. The dataset also contains various sound recordings, which includes sustained vowels, numbers, words, and short sentences, taken from each subject. Then, a group of 26 linear and time-frequency based features is extracted from the aforementioned recordings. It captures different aspects of voice characteristics such as jitter, shimmer, pitch, and other relevant measurements.

Therefore, the primary objective of this dataset is to develop a predictive model that can predict UPDRS scores based on the extracted voice features. This can be done by analyzing the relationship between the voice features against the UPDRS scores. Thus, the aim is to identify relevant patterns and build model that can assist the diagnosis of Parkinson's disease to gain insights into the Parkinson's diseases' severity and management of the diseases' condition.

## Data preprocessing

The dataset contains no null values in all of their variables, and the data types of these variables are either numerical or integer. Thus, data preprocessing is not required in this dataset since it is clean.

Next, the data is split into 80:20 ratio of training data and testing data respectively. This is needed to test the best model and test the consistency of the model in training data and testing data.

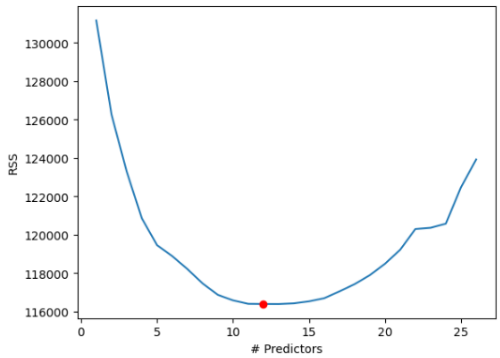
## Result summary

### Forward selection method

The method selected for predicting the UPDRS score is by using forward selection method as it offers incremental improvement, reduces complexity, provides interpretability, and computationally efficient to select relevant predictors.

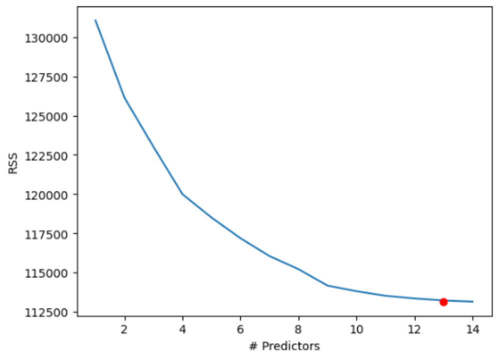
```
plt.plot(models_train["RSS"])
plt.xlabel('# Predictors')
plt.ylabel('RSS')
plt.plot(models_train["RSS"].argmin(), models_train["RSS"].min(), "or")
```

[<matplotlib.lines.Line2D at 0x2a52899adf0>]



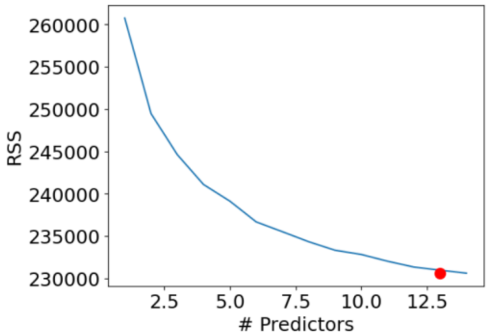

```
plt.plot(models_test["RSS"])
plt.xlabel('# Predictors')
plt.ylabel('RSS')
plt.plot(models_test["RSS"].argmin(), models_test["RSS"].min(), "or")
```

[<matplotlib.lines.Line2D at 0x2a528a12640>]




```
plt.plot(models_all["RSS"])
plt.xlabel('# Predictors')
plt.ylabel('RSS')
plt.plot(models_all["RSS"].argmin(), models_all["RSS"].min(), "or")
```

[<matplotlib.lines.Line2D at 0x15ca0a863a0>]




Based on the dataset, the best MLR model is 13 predictors in both training and testing data. In full dataset as well, the best model consists approximately 13 predictors.

## Mean Squared Error (MSE):

```
models_train["RSS"]
```

```
1     131151.778631
2     126268.815874
3     123298.119762
4     120857.707340
5     119462.441125
6     118891.988475
7     118217.167977
8     117469.598595
9     116878.460013
10    116589.791119
11    116409.844934
12    116402.931241
13    116399.275240
14    116435.207129
15    116538.272429
16    116703.035615
17    117056.258499
18    117433.667239
19    117901.744142
20    118495.050732
21    119224.725691
22    120303.843844
23    120367.163649
24    120578.051271
25    122472.958545
26    123922.005604
Name: RSS, dtype: float64
```

```
models_test["RSS"]
```

```
1     131067.471904
2     126141.589957
3     123010.316233
4     119988.248075
5     118502.868451
6     117181.587727
7     116045.172305
8     115210.050312
9     114162.595288
10    113803.111088
11    113514.101632
12    113343.324204
13    113213.329838
14    113138.404551
Name: RSS, dtype: float64
```

As MSE measures the average squared difference between the predicted UPDRS score and actual UPDRS score. We can infer the value of MSE through the value of RSS (Residual Sum of squares) divided by the total samples or data points in the dataset.

From the table, we can observe that the testing data has lower RSS than training data. Thus, we can infer that the lower RSS in training data suggest that the model performs better on test data. Although the lower RSS value compare to train data might indicate overfitting of data, but the variance of RSS in train data and test data is small, thus we can infer that overfitting would not be a major concern based on the MSE.

MSE value:

```
Mean Squared Error (MSE) - Train Data: 221.15661410980402
Mean Squared Error (MSE) - Test Data: 213.69203730240528
```

```
full dataset Mean Squared Error (MSE): 219.03723175289304
```

When we compute the MSE, we can observe that the MSE for test data is  lower than train data and full dataset. The reason for higher MSE in full dataset because it include all the dataset instead of a proportion of data. But the MSE in full dataset is still lower than train data,  which we can infer that 13 predictors is the best model.

<u>Other performance indicators:</u>

```
print(models_test.loc[13,"model"].summary())
```

```
                            OLS Regression Results
=======================================================================================
Dep. Variable:            UPDRS score   R-squared (uncentered):                   0.480
Model:                            OLS   Adj. R-squared (uncentered):              0.467
Method:                 Least Squares   F-statistic:                              36.66
Date:                Sun, 28 May 2023   Prob (F-statistic):                    6.21e-65
Time:                        17:10:51   Log-Likelihood:                         -2169.9
No. Observations:                 529   AIC:                                      4366.
Df Residuals:                     516   BIC:                                      4421.
Df Model:                          13
Covariance Type:            nonrobust
=====================================================================================================
                                       coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Shimmer (apq11)                      0.4549      0.154      2.962      0.003       0.153       0.757
HTN                                  0.0361      0.134      0.269      0.788      -0.227       0.299
Jitter (local, absolute)          2.721e+04   9700.646      2.805      0.005    8152.904    4.63e+04
Degree of voice breaks              -0.1428      0.071     -2.007      0.045      -0.283      -0.003
NTH                                -39.0199      8.607     -4.533      0.000     -55.929     -22.111
Jitter (rap)                         4.1180      2.295      1.794      0.073      -0.390       8.626
Shimmer (local, dB)                  2.8402      5.427      0.523      0.601      -7.822      13.502
Number of periods                    0.0129      0.005      2.569      0.010       0.003       0.023
Fraction of locally unvoiced frames  0.0840      0.041      2.049      0.041       0.003       0.165
Shimmer (apq3)                      -0.7536      0.424     -1.777      0.076      -1.587       0.080
Shimmer (local)                      0.5867      0.512      1.147      0.252      -0.418       1.592
Number of voice breaks              -0.5118      0.640     -0.800      0.424      -1.769       0.745
Jitter (ppq5)                       -1.4640      1.902     -0.770      0.442      -5.201       2.273
==============================================================================
Omnibus:                       89.005   Durbin-Watson:                   0.288
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              131.581
Skew:                           1.152   Prob(JB):                     2.68e-29
Kurtosis:                       3.815   Cond. No.                     2.83e+06
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 2.83e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
print(models_train.loc[13,"model"].summary())
```

```
                            OLS Regression Results
=======================================================================================
Dep. Variable:            UPDRS score   R-squared (uncentered):                   0.471
Model:                            OLS   Adj. R-squared (uncentered):              0.457
Method:                 Least Squares   F-statistic:                              34.15
Date:                Mon, 29 May 2023   Prob (F-statistic):                    1.12e-60
Time:                        01:11:28   Log-Likelihood:                         -2112.4
No. Observations:                 511   AIC:                                      4251.
Df Residuals:                     498   BIC:                                      4306.
Df Model:                          13
Covariance Type:            nonrobust
=====================================================================================================
                                       coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Shimmer (apq11)                      0.5099      0.156      3.261      0.001       0.203       0.817
HTN                                  0.1127      0.435      0.259      0.796      -0.742       0.967
Jitter (local, absolute)          2108.9332   1.45e+04      0.146      0.884   -2.64e+04    3.06e+04
NTH                                -19.7522      9.435     -2.093      0.037     -38.290      -1.214
Jitter (rap)                         4.3216      1.581      2.734      0.006       1.216       7.428
Shimmer (local, dB)                  5.4207      4.262      1.272      0.204      -2.953      13.794
Degree of voice breaks              -0.0230      0.059     -0.389      0.698      -0.139       0.093
Number of periods                    0.2175      0.142      1.526      0.128      -0.062       0.497
Fraction of locally unvoiced frames  0.0303      0.045      0.666      0.506      -0.059       0.120
Number of pulses                    -0.2052      0.142     -1.448      0.148      -0.484       0.073
Shimmer (dda)                       -0.2048      0.156     -1.309      0.191      -0.512       0.103
AC                                   4.5026      9.679      0.465      0.642     -14.514      23.519
Median pitch                        -0.0192      0.021     -0.915      0.360      -0.060       0.022
==============================================================================
Omnibus:                       60.484   Durbin-Watson:                   0.215
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               80.192
Skew:                           0.966   Prob(JB):                     3.86e-18
Kurtosis:                       3.174   Cond. No.                     6.04e+06
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 6.04e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

<u>R-squared (R2):</u>

From the table, we observe that the test data has higher R-squared value than in train data, this indicates that the model's performance is better at explaining the variance of the unseen test data.

A higher R-squared value in test data suggest that the model is able to **generalise better beyond** the patterns observed in the train data, it also shows that the model is able to capture the underlying relationship between the features (dependent variables) and the UPDRS score (independent variable). Plus, it further indicates that this model is **more robust** and reliable as it does not limit to the specific variations present in the train data, and thus, it can effectively handle the diverse and unseen data better.

Furthermore, the model is able to **avoid overfitting** concern from the RSS argument as mentioned above since it is not overly sensitive to the specific characteristic of the train data. This further indicates that the model's prediction are more accurate and aligned with true values where the model's estimated regression line fits the test data point better, which results in lesser variability in prediction to reduce prediction errors.

<u>Adjusted R-squared:</u>

Adjusted R-squared takes into account the number of predictors in the model, and penalise the inclusion of unnecessary features. Since the adjusted R-squared is higher in test data as well, it suggest that the model is able to explain a larger proportion of the variance in the test data while utilising a more **parsimonious predictors**. Thus, it indicates that the model has effectively identifies and included the most informative predictors which results to a more concise model.

Similar to R-squared, a higher value in test data suggest that the model can **generalise well** beyond the train data while managing the complexity of the predictors. It is **more robust** to handle the diverse and unseen data while maintaining the model from over-fitting which affect the interpretability.

Higher adjusted R-squared in test data is crucial as it improve the model prediction accuracy, and that it is realiable for prediction of new and unseen data. Thus, this further suggest that the model's estimated regression line fits the test data point better which results to reducing biasness and improve prediction accuracy.

In summary, a higher adjusted R-squared value in the test data suggests that the model is more parsimonious, generalizes well beyond the train data, improves prediction accuracy, and demonstrates robustness and general applicability. This highlights the model's ability to provide meaningful insights and reliable predictions when applied to new, unseen data.

<u>Akaike Information Criterion (AIC)</u>

A lower AIC value indicates that the model's performance has a better balance between model fit and complexity. However, the AIC for test data is higher than train data, which suggest that the model's performance in terms of balancing good fit and model complexity is relatively poor for test data.

It indicates that the train data may overfit. It occurs when the model captures noise or random fluctuations in the training data that leads to poor generalisation of unseen data in test data. This also suggests that the model might be too complex or may have included unnecessary predictors that do not contribute to a better fit on new data.

When there is a lack in generalisation, it indicates that the model may not be able to capture the relationships between the predictors against the UPDRS score as it relies on specific characteristics or pattern present in the training data.

**Conclusion**

As a whole, the model has a positive outlook on its MSE, R-squared and adjusted R-squared. And all of these performance indicators suggest that the model is not overfit and has great generalisation to manage the complexity of the predictors. It also suggests that the model is able to be robust in handling the diverse and unseen data while maintaining the model from over-fitting. This will further improve the prediction accuracy as it is not sensitive to a specific characteristics to any of the features in the dataset.

However, although there is a higher AIC value in the model, which concerns that the model may be unstable, and may face with decrease in predictive performance, it should only be noted on this potential issue as the other three performance indicators are proving otherwise observation.

Therefore, a slight precaution is to be noted when applying this model for predicting the UPDRS score.