Michael Turner

Technical Challenge

Upon visualising the initial dataset I found that there are many columns that that only contain zeros and that are duplicates of another column. Removing these columns brings us from a 517 by 7786 matrix to a 517 by 2375 matrix. Looking closer at the `‘days_to_death’` column I found that some are left blank producing a `NaN` result. I then removed these rows producing a final dataset to work on that is a 508 by 2375 matrix

Next is to assign the independent values to be the reaction flux rate, denoted `x`, and the dependent values to be the `age_to_death` column, denoted `y`. For here we are ready to begin applying learning techniques.

Supervised Learning: Linear Regression

Since we are trying to predict a real value for `age_to_death` I used `LinearRegression` from the Scikit-Learn library `sklearn.linear_model`. To split, train and test the dataset I used `train_test_split` from the Scikit-Learn library `sklearn.model_selection` as

```
xTrain, xTest, yTrain, yTest = train_test_split(x, y, test_size = 0.2,
                                                 random_state = 42)
```

After training on 80% of the dataset we have a prediction accuracy of

```
Accuracy:  0.9999995705155238
```

This is thanks to the large size of the dataset. Because of the extreme success of the linear regression method we could bring the training size down since, in most cases, it is not necessary to be this accurate. We can take advantage of the large number of features. Training on 40% of the dataset gives an accuracy of

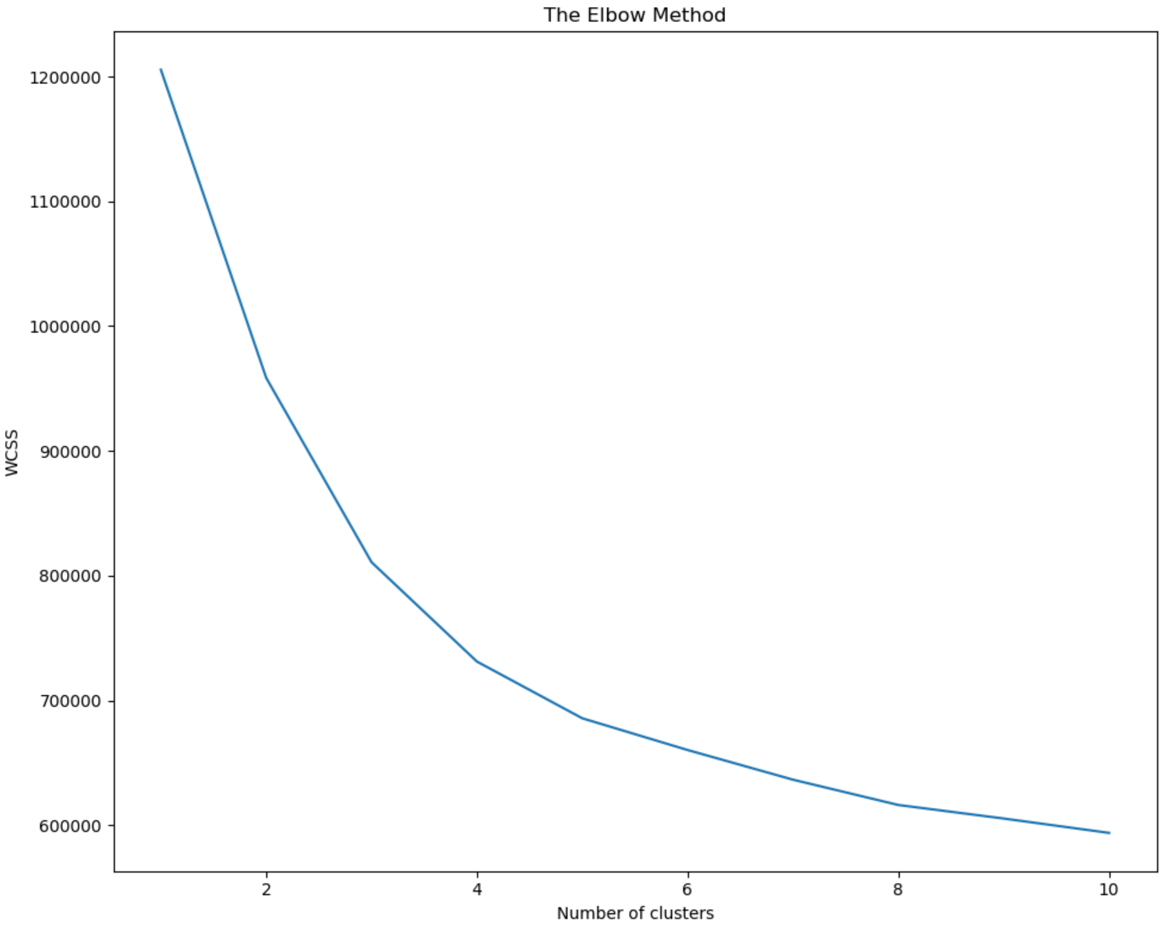
```
Accuracy:  0.9976726561327651
```

Unsupervised Learning: K-Means Clustering

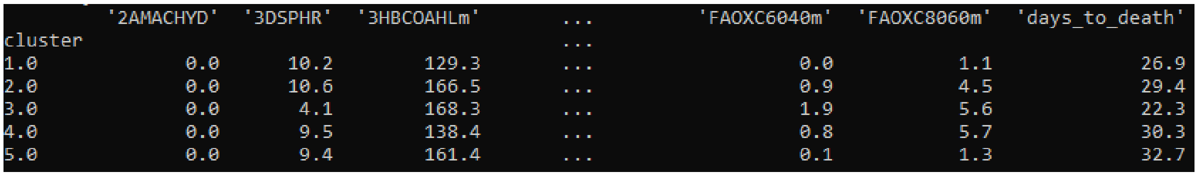
Here the goal is to identify the clusters within the data based on similarities within the cluster members. The essential stage is to standardize the dataset as the K-Means clustering depends on calculating distance between observations. Some variables may have a higher influence on the clustering output due to different scales of measurement.

After standardizing the dataset, the K-Means number of clusters needs to be defined before applying the method. To do this I looked for the number of clusters that minimises the total Within-Cluster Sum of Squares (WCSS). This can be done through the Elbow

Method which plots the total WSCC as a function of the number of clusters shown in the following Figure



Here the optimal number of clusters is the location of the bend at Number of cluster = 5. Using this value with `KMeans` from the Scikit-Learn library `sklearn.cluster` I produced a new data frame for Clusters and then we can calculate the means for each cluster shown in the following figure



```
         '2AMACHYD'  '3DSPHR'  '3HBCOAHLm'       ...          'FAOXC6040m'  'FAOXC8060m'  'days_to_death'
cluster                                          ...
1.0             0.0      10.2        129.3       ...                   0.0           1.1             26.9
2.0             0.0      10.6        166.5       ...                   0.9           4.5             29.4
3.0             0.0       4.1        168.3       ...                   1.9           5.6             22.3
4.0             0.0       9.5        138.4       ...                   0.8           5.7             30.3
5.0             0.0       9.4        161.4       ...                   0.1           1.3             32.7
```

So, given our dataset, we have that the following reaction flux rate values will on average give the corresponding `age_to_death`.