# HOMEWORK REPORT

**By Beltus Nkwawir - 704181021**

## Problem 1: Select important features affecting calories Burned.

This problem uses the dataset “fitbit.csv” to find the top-4 features having the highest relationship with the target ("calories burned" column).

Below is a figure showing the first five rows of the dataset dataframe using pandas library.

| | Date | Calories Burned | Steps | Distance | Floors | Minutes Sedentary | Minutes Lightly Active | Minutes Fairly Active | Minutes Very Active | Activity Calories |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7/07/2016 | 2,682 | 12,541 | 9.02 | 13 | 667 | 171 | 18 | 60 | 1,248 |
| 1 | 8/07/2016 | 2,423 | 8,029 | 5.70 | 35 | 760 | 208 | 13 | 6 | 928 |
| 2 | 9/07/2016 | 2,875 | 10,801 | 7.67 | 3 | 496 | 148 | 18 | 46 | 1,040 |
| 3 | 10/07/2016 | 2,638 | 11,997 | 8.52 | 22 | 771 | 248 | 3 | 27 | 1,285 |
| 4 | 11/07/2016 | 2,423 | 9,039 | 6.42 | 12 | 714 | 232 | 10 | 16 | 1,044 |

*Figure: First five columns of the dataset.*

A preprocessing step was performed to convert the Calories Burned, Steps and Activity Calories columns’ to float64. Then the input feature matrix, X, and the target output, Y were extracted from the dataset.

### Mutual Information Score.

By using the sklearn mutual_info_regression module, the mutual information score was computed for every input feature and the target(“calories burned”). The scores where then sorted to obtain the first four features corresponding to the highest mutual information score.

The results obtained are shown below.

*Selected Features having top mutual information scores*
*['Activity Calories', 'Minutes Fairly Active', 'Steps', 'Distance']*

### Recursive Feature Elimination  with Ridge Regressor

For the RFE method, the Ridge regressor as the estimator together with the sklearn *RFE()* method was used to determine the first four features that are most important. The results obtained are shown below.

Selected features by Recursive Feature Elimination
['Distance', 'Minutes Lightly Active', 'Minutes Fairly Active', 'Minutes Very Active']

## Problem 2: Cluster customer information using K-means algorithm.

In this problem, we used the “customer.csv” dataset to segment customers. The first five columns of the dataset can be seen in the figure below

| | ID | Visit.Time | Average.Expense | Sex | Age |
|---|---|---|---|---|---|
| **0** | 1 | 3 | 5.7 | 0 | 10 |
| **1** | 2 | 5 | 14.5 | 0 | 27 |
| **2** | 3 | 16 | 33.5 | 0 | 32 |
| **3** | 4 | 5 | 15.9 | 0 | 30 |
| **4** | 5 | 16 | 24.9 | 0 | 23 |

*Figure: First Five columns of customer.csv dataset*

Before computing the Distortions or Silhouette coefficients, the features were normalized using the ***StandardScaler()*** method from the ***sklearn preprocessing*** class. The results of the normalizing the dataset is shown below

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | -1.703420 | -1.212336 | -1.363789 | -1.468977 | -1.241735 |
| **1** | -1.645677 | -0.763323 | -0.307178 | -1.468977 | 0.604577 |
| **2** | -1.587934 | 1.706250 | 1.974142 | -1.468977 | 1.147610 |
| **3** | -1.530191 | -0.763323 | -0.139080 | -1.468977 | 0.930396 |
| **4** | -1.472448 | 1.706250 | 0.941545 | -1.468977 | 0.170150 |

*Figure: Normalized dataset*

Finally, the distortion score and silhouette score was then computed for cluster numbers ranging from 2 to 9 and a graph plotted as can be seen below.

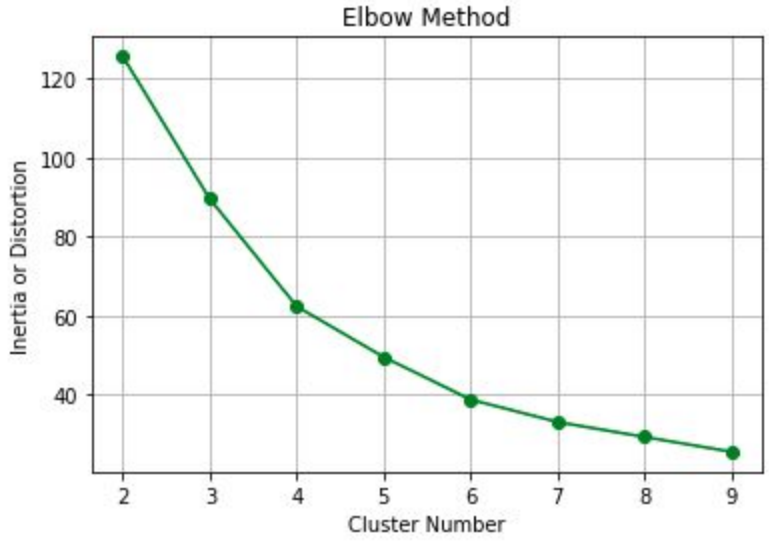


*Figure: Elbow Plot for K-Means Clustering Algorithm*

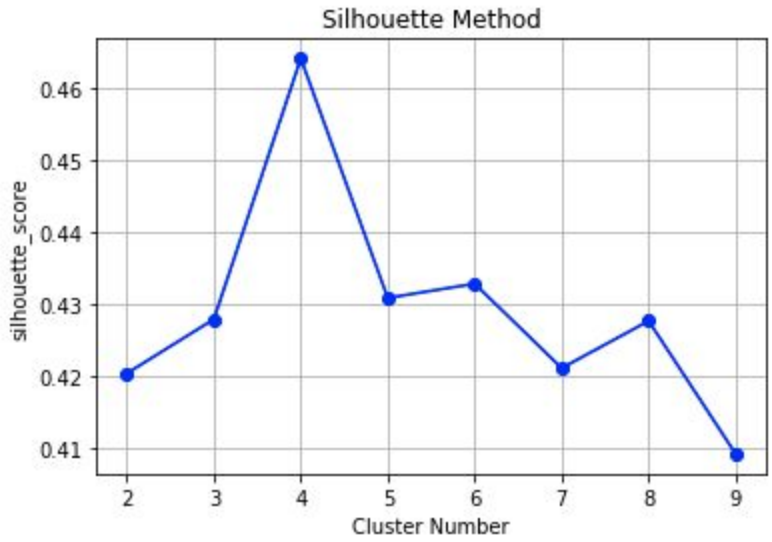


*Figure: Silhouette Plot for K-Means Clustering Algorithm*

# Problem 3. Predict Customer Churn

In this problem, the task was to predict customer churn using Logistic Regression, Decision Tree, Support Vector Machine, K-Nearest Neighbor and Neural Network methods.

The dataset used is the “telco-customer-churn”. In the figure below the first five columns of this dataset is shown

| DeviceProtection | TechSupport | StreamingTV | StreamingMovies | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalCharges | Churn |
|---|---|---|---|---|---|---|---|---|---|
| No | No | No | No | Month-to-month | Yes | Electronic check | 29.85 | 29.85 | No |
| Yes | No | No | No | One year | No | Mailed check | 56.95 | 1889.5 | No |
| No | No | No | No | Month-to-month | Yes | Mailed check | 53.85 | 108.15 | Yes |
| Yes | Yes | No | No | One year | No | Bank transfer (automatic) | 42.30 | 1840.75 | No |
| No | No | No | No | Month-to-month | Yes | Electronic check | 70.70 | 151.65 | Yes |

*Figure: telco-customer-churn dataset.*

## Basic Statistics of the Dataset.

To gain a deeper insight into the contents of the dataset, some statistical information was computed using the *pandas describe() method* and the results shown below.

| | SeniorCitizen | tenure | MonthlyCharges |
|---|---|---|---|
| **count** | 7043.000000 | 7043.000000 | 7043.000000 |
| **mean** | 0.162147 | 32.371149 | 64.761692 |
| **std** | 0.368612 | 24.559481 | 30.090047 |
| **min** | 0.000000 | 0.000000 | 18.250000 |
| **25%** | 0.000000 | 9.000000 | 35.500000 |
| **50%** | 0.000000 | 29.000000 | 70.350000 |
| **75%** | 0.000000 | 55.000000 | 89.850000 |
| **max** | 1.000000 | 72.000000 | 118.750000 |

*Figure: Statistical information of the dataset*

## Preprocessing of the Dataset.

Before the different models were trained on the dataset, some preprocessing was done on the dataset. All missing values were removed from columns that had missing values such as the TotalCharges Column. Categorical features were then label encoded to numeric features.

Based on the model and the performance obtained, feature scaling was adopted as a means to improved model accuracy.

## Model Training and Evaluation

The training and test results obtained using the Logistic Regression, Decision Tree, Support Vector Machine, K-Nearest Neighbor, and Neural Network models is shown below.

| model | train | test |
|---|---|---|
| LogisticRegression | 0.8052677195339866 | 0.802073319246403 |
| DecisionTree | 1.0 | 0.7268206819794826 |
| LinearSVC | 0.6610416412710383 | 0.6584042720498096 |
| KNN | 0.8330966728498131 | 0.751953452642106 |
| MLPClassifier | 0.6793287616477345 | 0.6701632968256017 |

*Figure: Train and Test scores from the various models*