Prediction Of Stroke Based on Real-Time Patient Data

DIC Project Phase 2

Team:

Name	UB ID
Venkata Rohil Wardhan Kancharla	vkanchar
Mahendra Nallabothu	mnallabo

Types of Model Algorithms Used:

1. Neural Network
2. Ridge Regression
3. KNN
4. Decision Tree
5. K-Means
6. Random Forest

Various Metrics Used:

1. **Mean Squared Error (MSE) -** Mean Squared error is the average squared distance between true or actual value vs predicted value. A model is perfect if MSE is 0.0.
2. **Root Mean Squared Error (RMSE) -** Root mean squared error is the root of MSE. the error can be interpreted better since the scale is the same as the random variable. If RSME is 0.0, the model is said to be ideal.
3. **Mean Absolute Error (MAE) -** Mean absolute error is the average difference between actual and predicted output values. It shows the error of the predicted vs actual value but cannot show if the data is underfitting or overfitting. A model is ideal if MAE is 0.0.
4. **R2 -** R2 is inversely proportional to the sum of the squared error of the regression line. The range of R2 is (-∞,1). If R2 is 1, the model is said to be ideal. R2 is written as R^2 (R Squared).
5. **Accuracy -** The accuracy is calculated by comparing the actual data with the predicted data obtained after training the model. The accuracy is calculated to be 100%, and the model with an accuracy close to 100% is better suited for prediction.

1. Neural Network - A neural network is an artificial intelligence that allows the computer to predict the data's accuracy, loss, and other predictions based on the test and training data split from the data frame. The neural network functions as a human brain for the computer. Neural Networks is can be classified as regression under supervised learning of machine learning models.

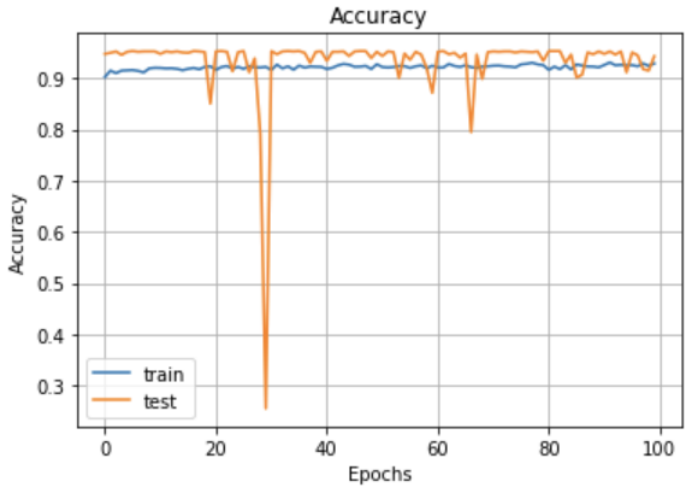
Justification - Neural Network has been used to know how accuracy does the human brain function based on the given data. The accuracy obtained was 94%, which might be good for a small dataset but is not so efficient for a larger dataset, like around 100,000 patient data.

Accuracy and Loss of Test Data -
Loss - 2.44
Accuracy - 94.29%

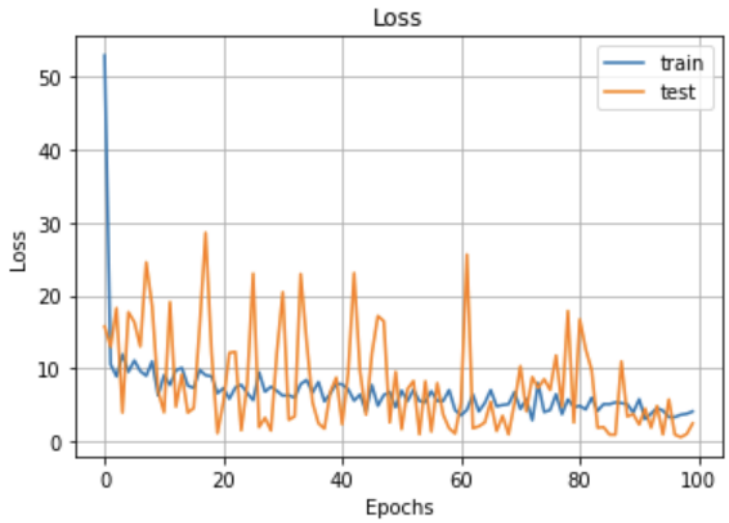
```
[2.4498958587646484, 0.9429734945297241]
```

Visualization -

Accuracy of Train vs Test Data -



Loss of Train vs Test Data -



2. Ridge Regression - Ridge regression is a type of regression under the supervised learning of a machine learning model. Ridge regression is done to prevent overfitting data and reduce error by adding some bias to the regression estimates. Ridge regression shows the error present as the distance between the present point to the linear line drawn as y=mx+b. This bias is taken as lambda, which can be taken as a random value between 0 and 1 and going by trial and error method to find the best possible value of lambda to get better evaluation metrics.

Justification -
Ridge regression can be used as the metrics obtained are close to an ideal dataset which helps in better prediction of the stroke of a patient.

MSE - 0.04
RMSE - 0.20
MAE - 0.08
R2 for the model - 0.05

```
mse_ridge = mean_squared_error(y_test, y_pred_ridge)
mse_ridge
```

```
0.04258699293448754
```

```
rmse_ridge = math.sqrt(mse_ridge)
rmse_ridge
```

```
0.20636616228075652
```

```
mae_ridge = mean_absolute_error(y_test, y_pred_ridge)
mae_ridge
```

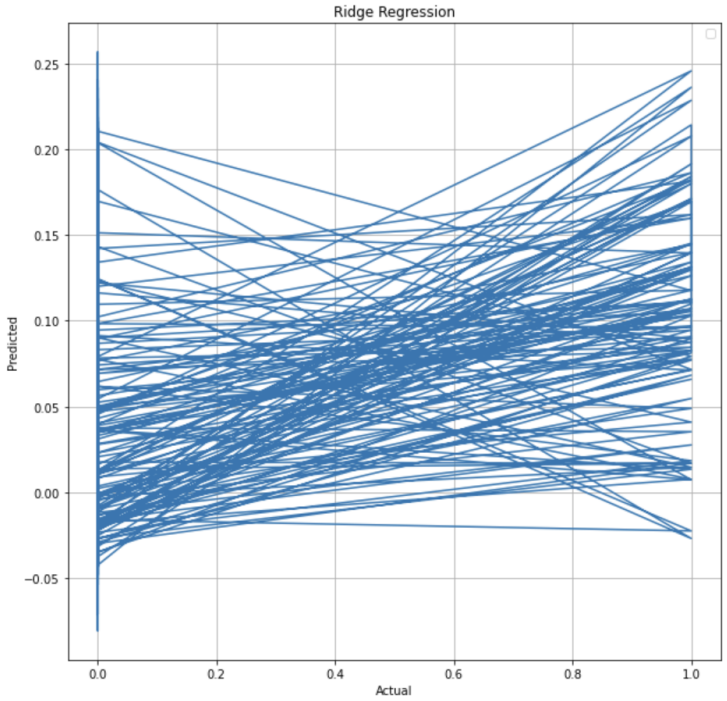
```
0.0897305839339679
```

```
r2_ridge = r2_score(y_test, y_pred_ridge)
r2_ridge
```

```
0.05913620208988157
```

Visualization -

The actual vs predicted data of the ridge regression model can be observed from the graph below. The error of actual data is more compared to the predicted data.



3. KNN - KNN is known as K-Nearest Neighbour. KNN is a classification model under supervised learning of machine learning models. The data is classified to its nearest neighbors based on the model trained. KNN does not perform training on the training data. It stores the data to classify the new data to its nearest neighbors.

Justification -
KNN can only be used for classification, and the model is close to ideal as the nearest neighbors are almost classified properly.

MSE - 0.04
RMSE - 0.21
MAE - 0.04
R2 - -0.04

```
mse_knn = mean_squared_error(y_test, y_pred_knn)
mse_knn
```

```
0.04412763068567549
```

```
rmse_knn = math.sqrt(mse_knn)
rmse_knn
```

```
0.21006577704537094
```

```
mae_knn = mean_absolute_error(y_test, y_pred_knn)
mae_knn
```

```
0.04412763068567549
```

```
r2_knn = r2_score(y_test, y_pred_knn)
r2_knn
```

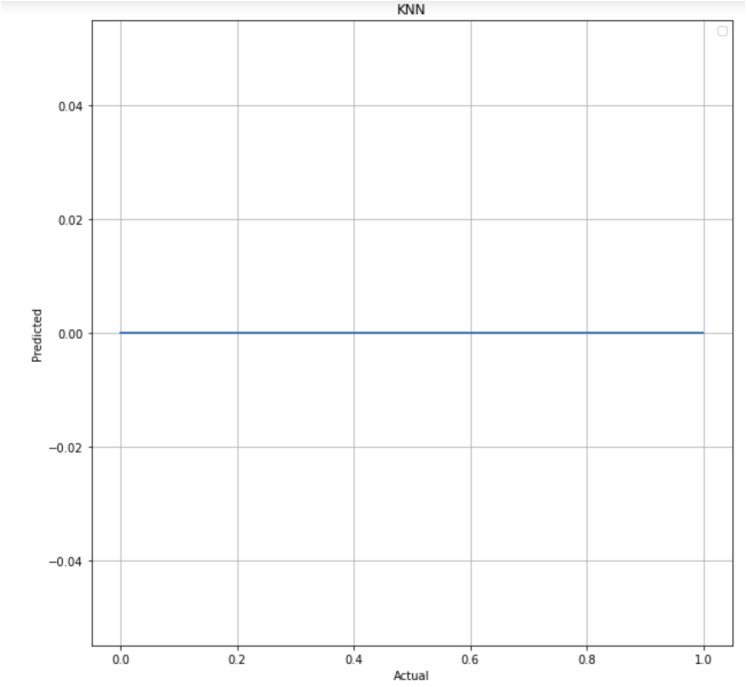
```
-0.04616477272727271
```

Accuracy - 95%

```
print(accuracy_score(y_test, y_pred_knn))
```

```
0.9558723693143245
```

Visualization -
As observed from the graph, the actual vs predicted data is on a straight line, which shows that the model's accuracy is constant at 0 and at 1. This shows the model is stable and can classify the test data effectively.



4. Decision Tree - The decision-making process of a model through the branching of the data to form a tree-like structure is decision-making. Decision tree comes under both classification and regression, and I have taken decision tree for data regression. The regression model of a decision tree predicts the output of continuous values, which is useful for stroke prediction.

Justification -
The obtained metrics are close to ideal but, R2 is below 0, which makes this model the third best from the different algorithms trained for the stroke prediction dataset as the accuracy is also 92%.

MSE - 0.07
RMSE - 0.27
MAE - 0.07
R2 - -0.83

```
mse_decision = mean_squared_error(y_test, y_pred_decision)
mse_decision
```

```
0.07739307535641547
```

```
rmse_decision = math.sqrt(mse_decision)
rmse_decision
```

```
0.27819610952782114
```

```
mae_decision = mean_absolute_error(y_test, y_pred_decision)
mae_decision
```

```
0.07739307535641547
```

```
r2_decision = r2_score(y_test, y_pred_decision)
r2_decision
```

```
-0.8348120629370628
```

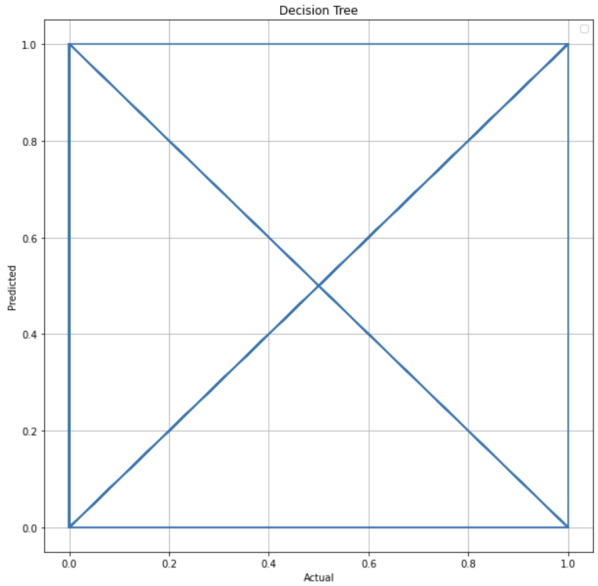
Accuracy - 92%

```
acc = accuracy_score(y_test, y_pred_decision)
acc
```

```
0.9226069246435845
```

Visualization -

As observed from the graph, the actual vs predicted output is shown as an interlinkage between different attributes taken by the output of continuous values to determine the required output.



5. K-Means - K-Means is a clustering of data based on the attributes provided with a centroid for each cluster which is the median value of that cluster. K-Means comes under unsupervised learning of the machine learning model. The centroid is calculated by repetition of the K-means algorithm until an optimal centroid is found. The number of clusters is the K defined by us in the algorithm.

Justification -
Since the model is unsupervised learning, it is highly unstable and unreliable. This shows that the clusters formed are not accurate and the model cannot be used to predict stroke.

MSE - 5.98
RMSE - 2.44
MAE - 2.02
R2 - -140.82

```
mse_kmeans = mean_squared_error(y_test, y_pred_kmeans)
mse_kmeans
```

```
5.98234894772573
```

```
rmse_kmeans = math.sqrt(mse_kmeans)
rmse_kmeans
```

```
2.4458840830517152
```

```
mae_kmeans = mean_absolute_error(y_test, y_pred_kmeans)
mae_kmeans
```

```
2.021724372029871
```

```
r2_kmeans = r2_score(y_test, y_pred_kmeans)
r2_kmeans
```

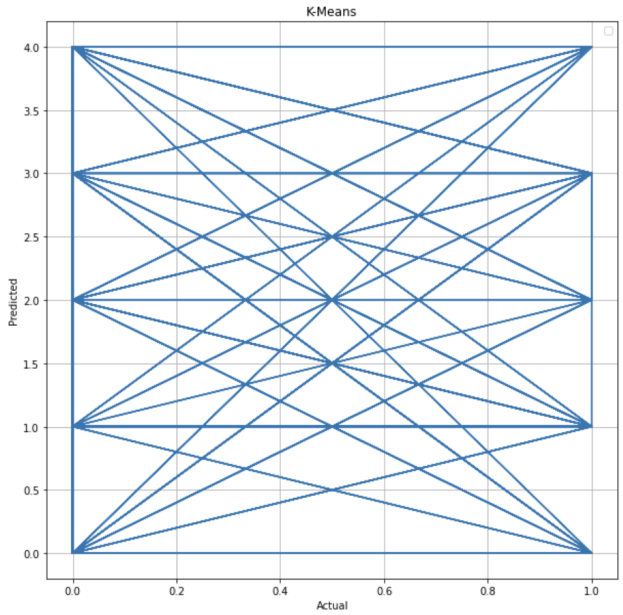
```
-140.82775349650348
```

Accuracy - 17%

```
acc = accuracy_score(y_test, y_pred_kmeans)
acc
```

```
0.17175831636116767
```

Visualization -
The k-means form clusters which have been interlinked, as observed in the graph. Clustering is done, but the model's accuracy shows that it is unreliable since accuracy is only 17%.



6. Random Forest - Random forest is a classification algorithm in supervised learning in machine learning. The random forest also classifies the data based on the training data but does not perform any training. It is used to find the behavior of the data. Random forest is constructed on top of the decision tree model, so, it also forms trees to show the behavior of data.

Justification -
Random forest data cannot be used as it does not perform any training to the data but can only be used to perform classification of data to study its behavioral pattern.

MSE - 0.04
RMSE - 0.21
MAE - 0.04
R2 - -0.04

```
mse_random = mean_squared_error(y_test, y_pred_random)
mse_random
```

```
0.04412763068567549
```

```
rmse_random = math.sqrt(mse_random)
rmse_random
```

```
0.21006577704537094
```

```
mae_random = mean_absolute_error(y_test, y_pred_random)
mae_random
```

```
0.04412763068567549
```

```
r2_random = r2_score(y_test, y_pred_random)
r2_random
```

```
-0.04616477272727271
```

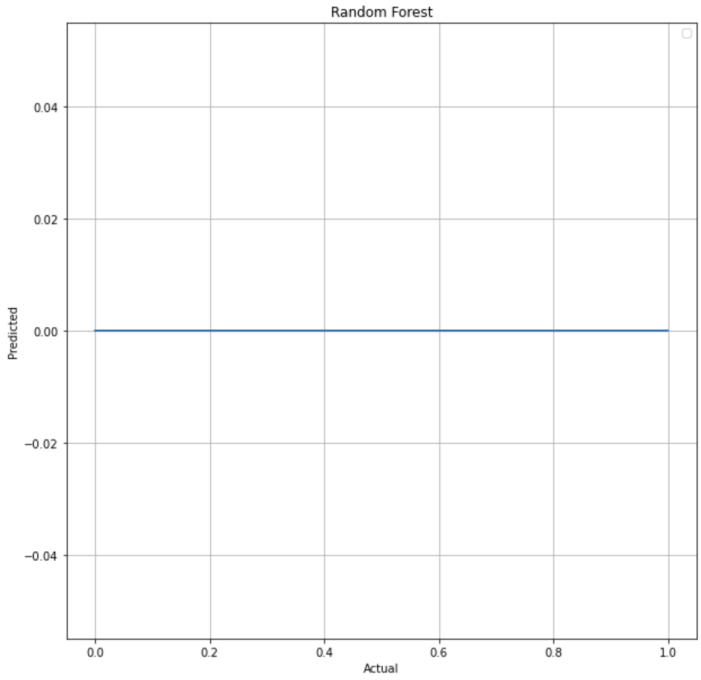
Accuracy - 95%

```
acc = accuracy_score(y_test, y_pred_random)
acc
```

```
0.9558723693143245
```

Visualization -

The accuracy of the model is 95%, which shows the behavior of the dataset is close to ideal and the dataset is constant from start to end. Even though the model cannot be used, it shows that the dataset is stable to perform machine learning algorithms to predict the stroke of a patient.



Reference -

For Different Model Algorithms taken -
https://scikit-learn.org/stable/
https://www.analyticsvidhya.com/blog/2021/03/everything-you-need-to-know-about-machine-learning/