e-ISBN : 978-629-98755-1-2

SIG : e-Learning@CS
https://appspenang.uitm.edu.my/sigcs/
Publication Date : 5-Apr-2024

PREDICTION OF STROKE DISEASE USING MACHINE LEARNING TECHNIQUES

*Syarifah Adilah Mohamed Yusoff[1], Saiful Nizam Warris[2], Mohd Saifulnizam Abu Bakar[3] and Rozita Kadar[4]
**syarifah.adilah@uitm.edu.my[1], saifulwar@uitm.edu.my[2], mohdsaiful071@uitm.edu.my[3], rozita231@uitm.edu.my[4]*

[1,2,3,4]Jabatan Sains Komputer & Matematik (JSKM),
Universiti Teknologi MARA Cawangan `Pulau Pinang, Malaysia

**Corresponding author*

ABSTRACT

Stroke is a global disease that is reported to increase annually and is a leading cause of mortality worldwide. The advancement of data analytics and machine learning has made it possible to foretell future patterns and insights, which could lead to the discovery of novel treatments for this condition. This study has investigated five commonly used machine learning algorithm to be constructed as potential models for predicting stroke dataset. Jupyter Notebook, a phyton-base engine, was employed as a data analytic tool for the purpose of analysing and evaluating all of the models. The five models were Decision Tree, Logistic Regression, Linear Discriminant Analysis, Gaussian Naïve Bayes and Support Vector Machine, have being implemented to predict binary outcome of stroke and no stroke. The accuracy percentage reported that Logistic regression outperformed other models with 50.93%.

Keywords: prediction, stroke, machine learning, data analytic, algorithm

Introduction

The number of people suffering from a stroke is increasing every day. Stroke or also known as "angin ahmar" among local Malaysian community is a cerebrovascular disease that happen when blood flow to the brain is restricted, blocked or reduced. According to Department of Statistic Malaysia (2022), stroke is third leading death among Malaysian, after Ischaemic heart disease and Lower repository infections. Several studies recently indicate the increasing trends among low and middle income countries, compared to high-income countries and rising trend among young adults (Wen et al. (2021); Kay & Venketasubramaniah(2022)).

Artificial intelligence is widely used in healthcare and medicine to assist doctors in making decisions by offering enhanced and accurate diagnosis and prognoses. Thanks to advancements in machine learning algorithms and statistical understanding, data mining and analytics have evolved beyond descriptive analysis to encompass predictive and prescriptive analytics. Figure 1 demonstrates the association of data mining with other disciplines, mutually union and complement each other to exploits three promising analytical approaches driven by information. The potential of these emerging technologies goes beyond the medical field and permeates several aspects of real-life situations, including science and technology, as well as humanities and social sciences.

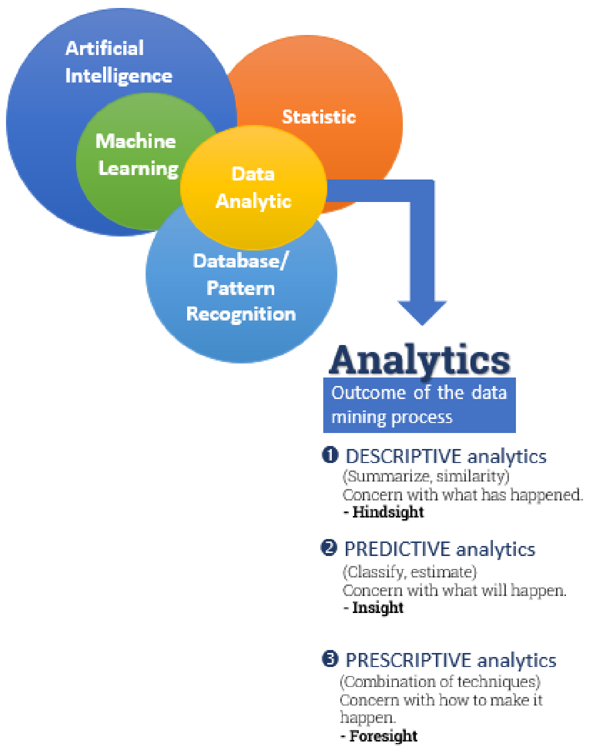


Figure 1: Analytics outcome from the emerging of data mining technology

In this era of digitalization, information is regarded as valuable as oil. Data analytics have the potential to greatly impact current and future endeavours as a result of the extensive use of relevant technology. Data analytics has the ability to transform raw data into valuable insights, identifying trends and solving problems. It can enhance corporate processes, optimise decision-making, and stimulate economic expansion.

Research Methodology

The general framework of supervised machine learning implementation is illustrated in the Figure 2, where the data of consist of stroke patients' attributes were first went through some steps of pre-processing process in order to prepare prominent features for further analysis. Then, the data are split into testing and training randomly using 10-fold cross validation. The training data was used to construct the model based on selected machine learning algorithms. Meanwhile, testing data was used to validate the model when the prediction was done.

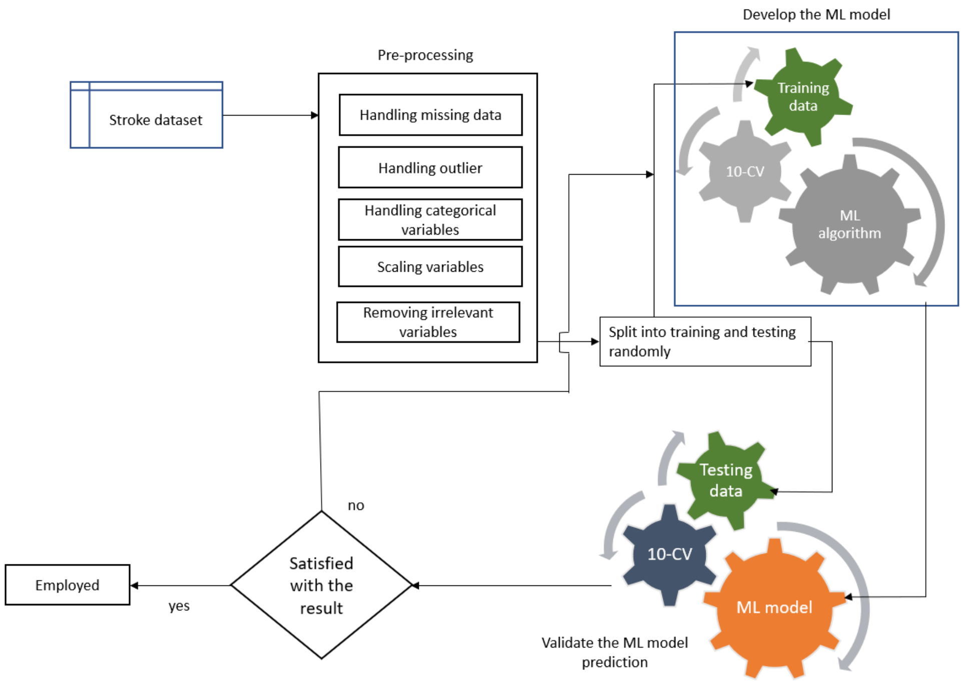
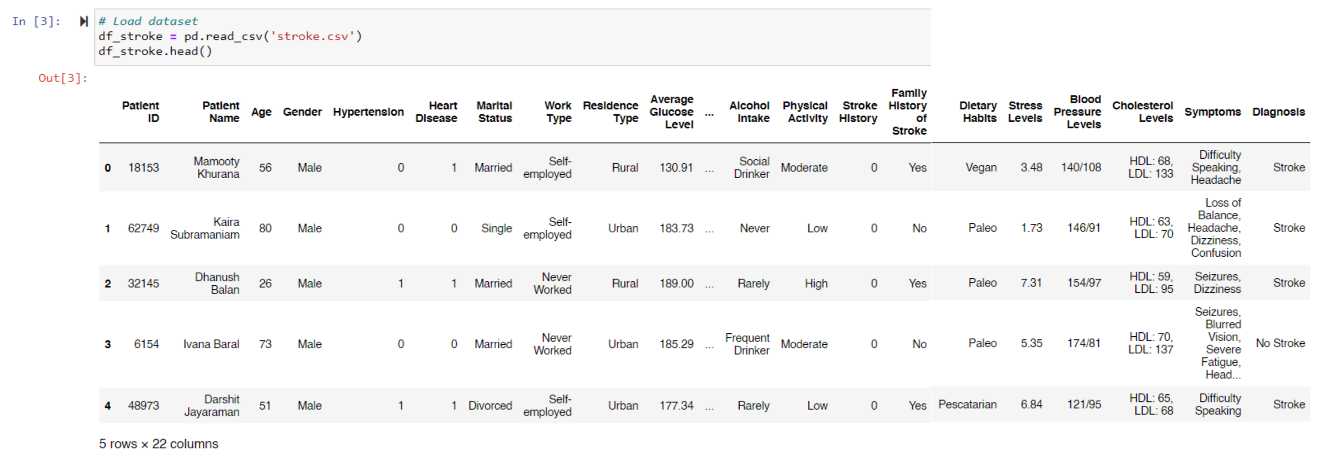


Figure 2: Research framework of stroke analysis

Understanding the Dataset

The stoke dataset is syntactic data from https://github.com/incribo-inc/stroke_prediction that classify patients from demographic and medical information to be diagnosed as stroke or non-stroke.

The datasets consist of 15,000 of record of patients and 22 attributes in total. The attributes are list of demographic and medical history information which are *'Patient ID', 'Patient Name', 'Age', 'Gender', 'Hypertension', 'Heart Disease', 'Marital Status', 'Work Type', 'Residence Type', 'Average Glucose Level', 'Body Mass Index (BMI)', 'Smoking Status', 'Alcohol Intake', 'Physical Activity', 'Stroke History', 'Family History of Stroke', 'Dietary Habits', 'Stress Levels', 'Blood Pressure Levels', 'Cholesterol Levels', 'Symptoms', 'Diagnosis'*. From these 22 attributes the first 21 are all features and the last one is considered as target attributes. The target attribute is two-class classification problem of stroke and non-stroke disease. Figure 3 illustrates the first five records of patients for all attributes (5 rows x 22 columns).



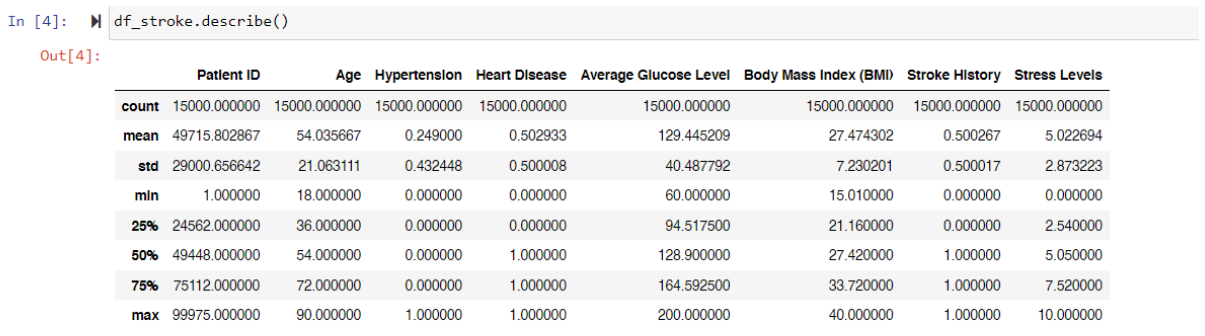
```
In [3]:  # Load dataset
         df_stroke = pd.read_csv('stroke.csv')
         df_stroke.head()
Out[3]:
```

	Patient ID	Patient Name	Age	Gender	Hypertension	Heart Disease	Marital Status	Work Type	Residence Type	Average Glucose Level	...	Alcohol Intake	Physical Activity	Stroke History	Family History of Stroke	Dietary Habits	Stress Levels	Blood Pressure Levels	Cholesterol Levels	Symptoms	Diagnosis
0	18153	Mamooty Khurana	56	Male	0	1	Married	Self-employed	Rural	130.91	...	Social Drinker	Moderate	0	Yes	Vegan	3.48	140/108	HDL: 68, LDL: 133	Difficulty Speaking, Headache	Stroke
1	62749	Kaira Subramaniam	80	Male	0	0	Single	Self-employed	Urban	183.73	...	Never	Low	0	No	Paleo	1.73	146/91	HDL: 63, LDL: 70	Loss of Balance, Headache, Dizziness, Confusion	Stroke
2	32145	Dhanush Balan	26	Male	1	1	Married	Never Worked	Rural	189.00	...	Rarely	High	0	Yes	Paleo	7.31	154/97	HDL: 59, LDL: 95	Seizures, Dizziness	Stroke
3	6154	Ivana Baral	73	Male	0	0	Married	Never Worked	Urban	185.29	...	Frequent Drinker	Moderate	0	No	Paleo	5.35	174/81	HDL: 70, LDL: 137	Seizures, Blurred Vision, Severe Fatigue, Head...	No Stroke
4	48973	Darshit Jayaraman	51	Male	1	1	Divorced	Self-employed	Urban	177.34	...	Rarely	Low	0	Yes	Pescatarian	6.84	121/95	HDL: 65, LDL: 68	Difficulty Speaking	Stroke

5 rows × 22 columns

Figure 3: The first 5 patients' information from the datasets

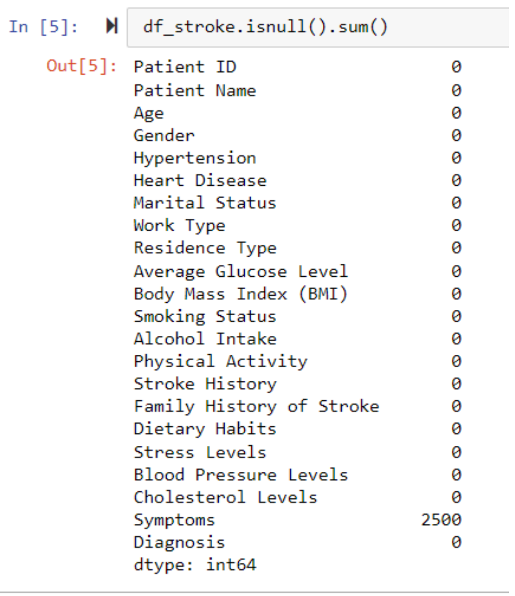
Figure 4 reveals that among the 21 attributes, 8 are categorized as numeric types. The pattern of the values can be described using statistical measures such as standard deviation, minimum, maximum, and the quartiles (25%, 50%, and 75%) of the data. Example, the magnitude of the data on each feature represent by standard deviation (std) and further the smallest and largest value given by min and max consecutively. Meanwhile the rest of features, illustrates in Figure 5 are from nominal types except *'Cholesterol Levels'*, *'Blood Pressure Levels'* and *'Symptoms'* are string types.



```
In [4]:  df_stroke.describe()
Out[4]:
```

	Patient ID	Age	Hypertension	Heart Disease	Average Glucose Level	Body Mass Index (BMI)	Stroke History	Stress Levels
count	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000
mean	49715.802867	54.035667	0.249000	0.502933	129.445209	27.474302	0.500267	5.022694
std	29000.656642	21.063111	0.432448	0.500008	40.487792	7.230201	0.500017	2.873223
min	1.000000	18.000000	0.000000	0.000000	60.000000	15.010000	0.000000	0.000000
25%	24562.000000	36.000000	0.000000	0.000000	94.517500	21.160000	0.000000	2.540000
50%	49448.000000	54.000000	0.000000	1.000000	128.900000	27.420000	1.000000	5.050000
75%	75112.000000	72.000000	0.000000	1.000000	164.592500	33.720000	1.000000	7.520000
max	99975.000000	90.000000	1.000000	1.000000	200.000000	40.000000	1.000000	10.000000

Figure 4: The eight features with numeric type

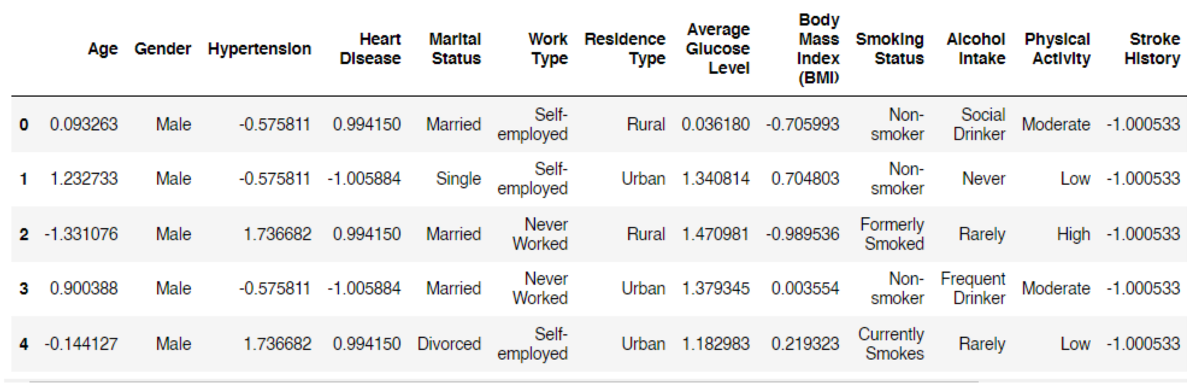


```
In [5]:   df_stroke.isnull().sum()

Out[5]:   Patient ID                     0
          Patient Name                   0
          Age                            0
          Gender                         0
          Hypertension                   0
          Heart Disease                  0
          Marital Status                 0
          Work Type                      0
          Residence Type                 0
          Average Glucose Level          0
          Body Mass Index (BMI)          0
          Smoking Status                 0
          Alcohol Intake                 0
          Physical Activity              0
          Stroke History                 0
          Family History of Stroke       0
          Dietary Habits                 0
          Stress Levels                  0
          Blood Pressure Levels          0
          Cholesterol Levels             0
          Symptoms                    2500
          Diagnosis                      0
          dtype: int64
```

Figure 5: The list of all attributes provides by the stroke dataset

Pre-process the Dataset

Once the data has been comprehended, the pre-processing steps have been taken over to perform or reconstruct the pattern of the data for the desired analysis. The list of pre-processing steps has been proposed in the framework in Figure 2. Begin by addressing the issue of missing values. The *'Symptoms'* feature has 2500 missing data points, therefore will be eliminated. Additionally, the other three string properties, namely *'Cholesterol Levels'* and *'Blood Pressure Levels'* also being eliminated. Next, features that have unique values, such as *'Patient ID'* and *'Patient Name'*, as these unique values are not related to our target class. Scaling the numeric features are important due to different and varies magnitude of the data may cause the bigger magnitude will over shadow the small magnitude in modelling process. Figure 6 shows the scaling implementation to the first 6th numeric features using average scaling method and the implementation will not lost any original properties of the data.

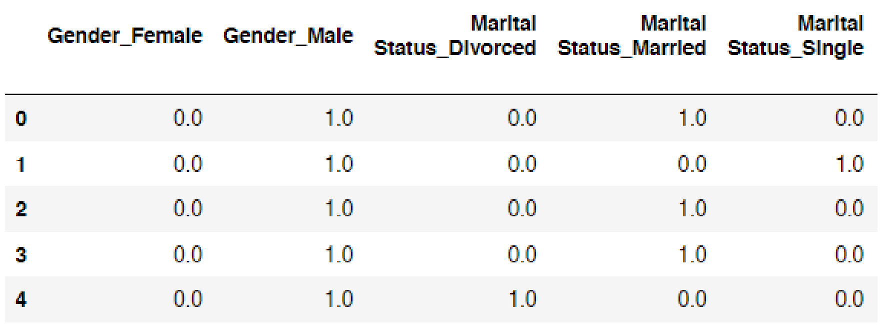


	Age	Gender	Hypertension	Heart Disease	Marital Status	Work Type	Residence Type	Average Glucose Level	Body Mass Index (BMI)	Smoking Status	Alcohol Intake	Physical Activity	Stroke History
0	0.093263	Male	-0.575811	0.994150	Married	Self-employed	Rural	0.036180	-0.705993	Non-smoker	Social Drinker	Moderate	-1.000533
1	1.232733	Male	-0.575811	-1.005884	Single	Self-employed	Urban	1.340814	0.704803	Non-smoker	Never	Low	-1.000533
2	-1.331076	Male	1.736682	0.994150	Married	Never Worked	Rural	1.470981	-0.989536	Formerly Smoked	Rarely	High	-1.000533
3	0.900388	Male	-0.575811	-1.005884	Married	Never Worked	Urban	1.379345	0.003554	Non-smoker	Frequent Drinker	Moderate	-1.000533
4	-0.144127	Male	1.736682	0.994150	Divorced	Self-employed	Urban	1.182983	0.219323	Currently Smokes	Rarely	Low	-1.000533

Figure 6: The scaling of the numeric types of features

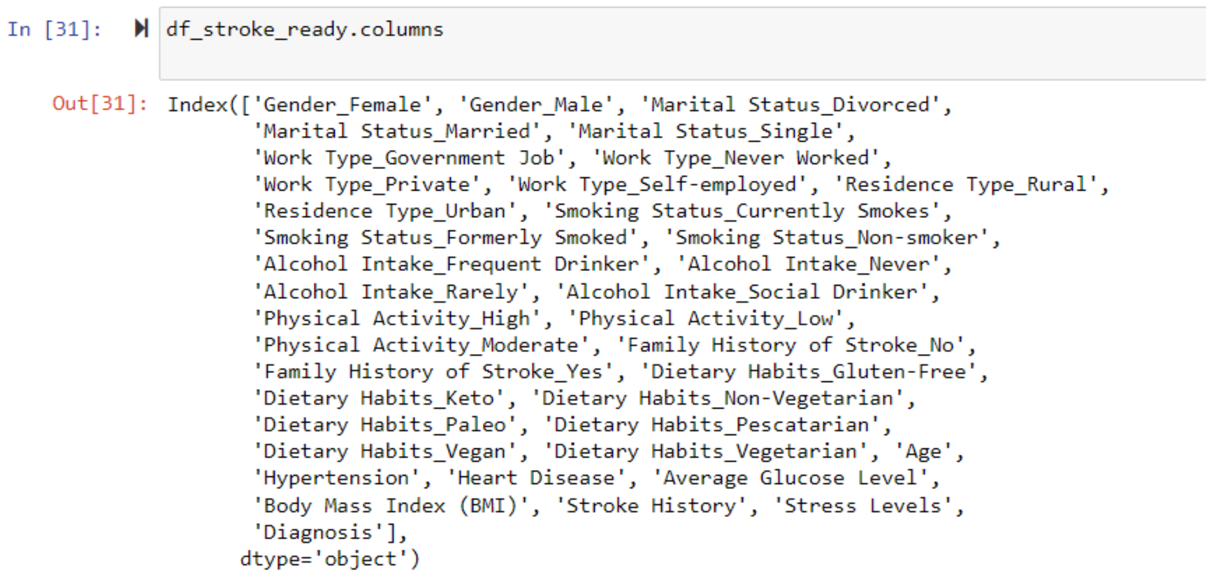
In data mining, encoding categorical data is necessary in order to make easier for computer to understand. The implementation using *one-hot encoding()* by converting each value of categorical features into a column and assign it a 1 or 0 value. Figure 7 illustrates the outcome of encoding process of two features Gender and Marital Status. Gender is a type of nominal features that consist value Female or Male. For example, since the instance #1st is Male, thus Gender_male is 1 and the Gender_Female is 0. For second feature, Marital Status consists of values Married, Single or Divorced. After encoding, 3 new columns appear for Marital Status and value 1 indicate the instance has true value for that column and the rest are 0. For example, instance #2nd Marital_Status_Single is 1, while Marital_Status_ Divorced and Marital_Status_Married are 0.

Figure 8 depicts all the final attributes selected for the next process which is predictive modelling classification using several Machine Learning algorithms.



	Gender_Female	Gender_Male	Marital Status_Divorced	Marital Status_Married	Marital Status_Single
0	0.0	1.0	0.0	1.0	0.0
1	0.0	1.0	0.0	0.0	1.0
2	0.0	1.0	0.0	1.0	0.0
3	0.0	1.0	0.0	1.0	0.0
4	0.0	1.0	1.0	0.0	0.0

Figure 7: Encoding two features of Gender and Marital Status



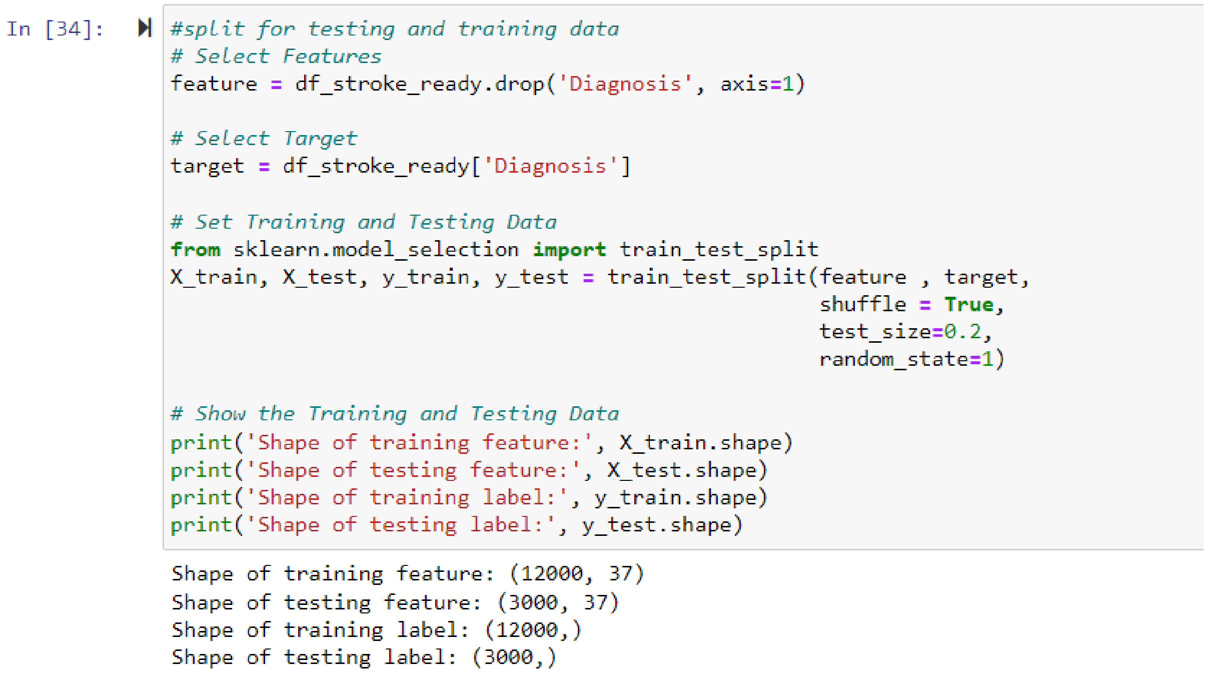
```
In [31]:   df_stroke_ready.columns

Out[31]: Index(['Gender_Female', 'Gender_Male', 'Marital Status_Divorced',
                'Marital Status_Married', 'Marital Status_Single',
                'Work Type_Government Job', 'Work Type_Never Worked',
                'Work Type_Private', 'Work Type_Self-employed', 'Residence Type_Rural',
                'Residence Type_Urban', 'Smoking Status_Currently Smokes',
                'Smoking Status_Formerly Smoked', 'Smoking Status_Non-smoker',
                'Alcohol Intake_Frequent Drinker', 'Alcohol Intake_Never',
                'Alcohol Intake_Rarely', 'Alcohol Intake_Social Drinker',
                'Physical Activity_High', 'Physical Activity_Low',
                'Physical Activity_Moderate', 'Family History of Stroke_No',
                'Family History of Stroke_Yes', 'Dietary Habits_Gluten-Free',
                'Dietary Habits_Keto', 'Dietary Habits_Non-Vegetarian',
                'Dietary Habits_Paleo', 'Dietary Habits_Pescatarian',
                'Dietary Habits_Vegan', 'Dietary Habits_Vegetarian', 'Age',
                'Hypertension', 'Heart Disease', 'Average Glucose Level',
                'Body Mass Index (BMI)', 'Stroke History', 'Stress Levels',
                'Diagnosis'],
               dtype='object')
```

Figure 8: Final attributes constructed for next process

Training the Prediction Model

The first step of developing the model using machine learning algorithm split the attributes into features and target. The target attribute is typically an independent attribute that serves as the ultimate goal of the investigation. Meanwhile, features consist all attributes that are dependents to the target (please refer to Figure 8).

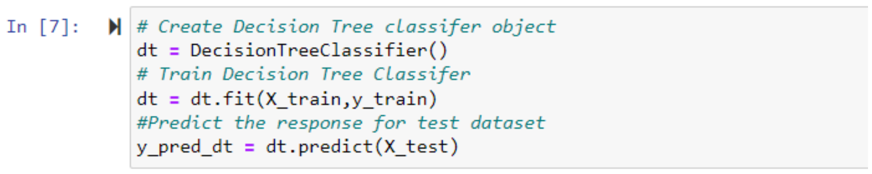
In this study, the target attribute is *'Diagnosis'* that consisting two-class values of stroke and no stroke. The coding implementation in Figure 9 demonstrates the process of splitting the stroke dataset. 80% of the dataset is allocated for training the model, allowing the algorithm to learn patterns from this subset of features and target label. Once the model was created, the validation process was conducted using the remaining 20% of testing features and label to assess the quality of the model. This validation will mitigate bias as the test data utilized for evaluation was distinct from the data employed in model development.



```
In [34]:  #split for testing and training data
          # Select Features
          feature = df_stroke_ready.drop('Diagnosis', axis=1)

          # Select Target
          target = df_stroke_ready['Diagnosis']

          # Set Training and Testing Data
          from sklearn.model_selection import train_test_split
          X_train, X_test, y_train, y_test = train_test_split(feature , target,
                                                              shuffle = True,
                                                              test_size=0.2,
                                                              random_state=1)

          # Show the Training and Testing Data
          print('Shape of training feature:', X_train.shape)
          print('Shape of testing feature:', X_test.shape)
          print('Shape of training label:', y_train.shape)
          print('Shape of testing label:', y_test.shape)

          Shape of training feature: (12000, 37)
          Shape of testing feature: (3000, 37)
          Shape of training label: (12000,)
          Shape of testing label: (3000,)
```

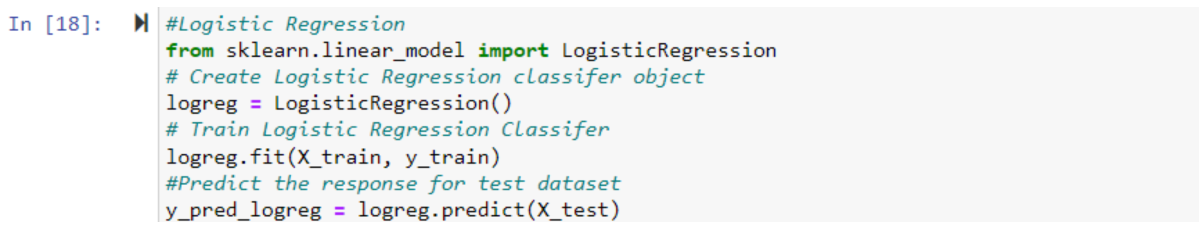
Figure 9: The implementation of split the attributes of stroke dataset to features and target class

In this study, several algorithms were chosen to be implemented with the training data to construct models, the implementation and construction of Decision Tree Model was depicted in Figure 10, Logistic Regression model in Figure 11, Linear Discriminant Analysis model in Figure 12, Gaussian Naïve Bayes model in Figure 13 and Support Vector Machine model in Figure 14. After the models was constructed, prediction step was generated to calculate the prediction results from testing dataset.



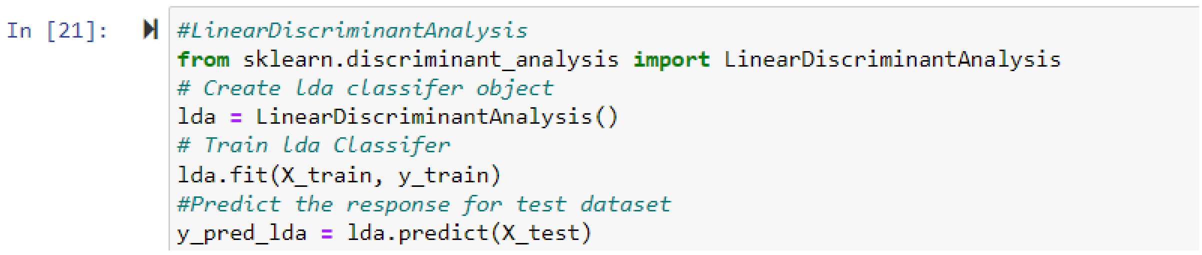
```
In [7]:  # Create Decision Tree classifer object
         dt = DecisionTreeClassifier()
         # Train Decision Tree Classifer
         dt = dt.fit(X_train,y_train)
         #Predict the response for test dataset
         y_pred_dt = dt.predict(X_test)
```

Figure 10: The implementation and construction for Decision Tree Model



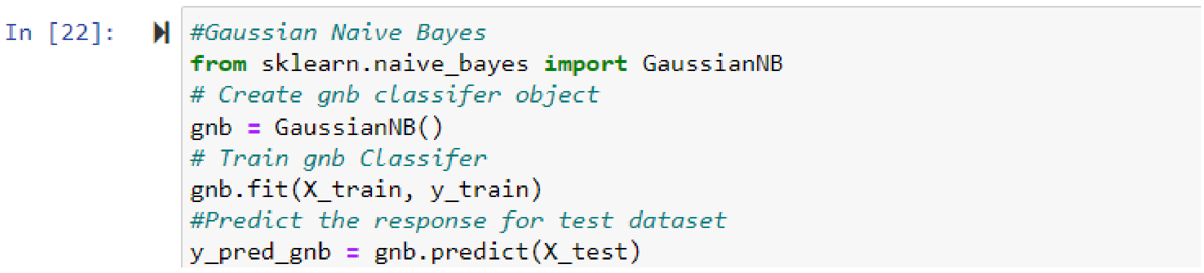
```
In [18]:  #Logistic Regression
          from sklearn.linear_model import LogisticRegression
          # Create Logistic Regression classifer object
          logreg = LogisticRegression()
          # Train Logistic Regression Classifer
          logreg.fit(X_train, y_train)
          #Predict the response for test dataset
          y_pred_logreg = logreg.predict(X_test)
```

Figure 11: The implementation and construction for Logistic Regression Model



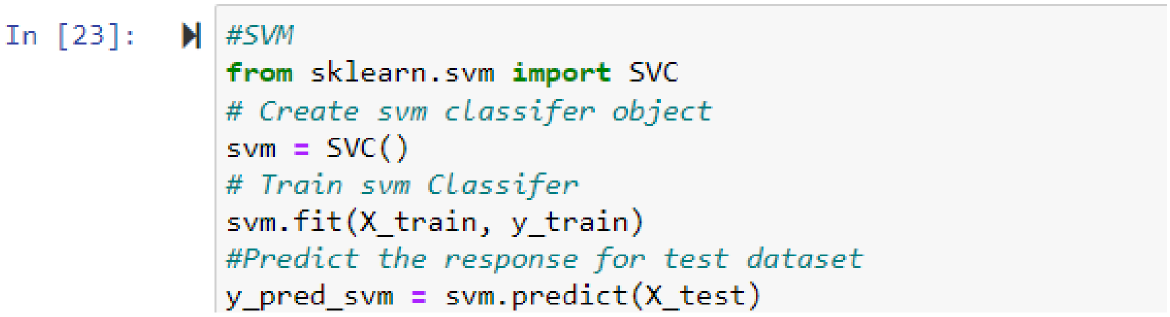
```
In [21]:  #LinearDiscriminantAnalysis
          from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
          # Create lda classifer object
          lda = LinearDiscriminantAnalysis()
          # Train lda Classifer
          lda.fit(X_train, y_train)
          #Predict the response for test dataset
          y_pred_lda = lda.predict(X_test)
```

Figure 12: The implementation and construction for Linear Discriminant Analysis Model



```
In [22]:  #Gaussian Naive Bayes
          from sklearn.naive_bayes import GaussianNB
          # Create gnb classifer object
          gnb = GaussianNB()
          # Train gnb Classifer
          gnb.fit(X_train, y_train)
          #Predict the response for test dataset
          y_pred_gnb = gnb.predict(X_test)
```

Figure 13: The implementation and construction for Gaussian Naïve Bayes Model



```
In [23]:  #SVM
          from sklearn.svm import SVC
          # Create svm classifer object
          svm = SVC()
          # Train svm Classifer
          svm.fit(X_train, y_train)
          #Predict the response for test dataset
          y_pred_svm = svm.predict(X_test)
```

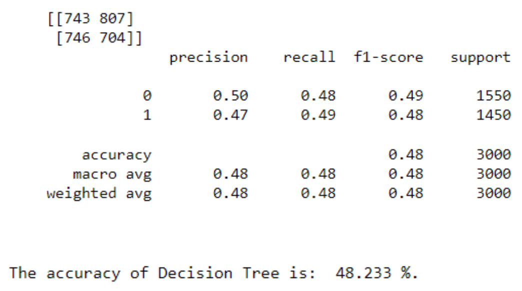
Figure 14: The implementation and construction for Support Vector Machine Model

Result and Discussion

The prediction results were gathered from all different models and evaluated using various metrics to determine how good is the predictive analytics models and find the most suited for the stroke prediction analysis. Confusion Matrix, classification report and accuracy scores were output result from the analysis and varies from all different models.

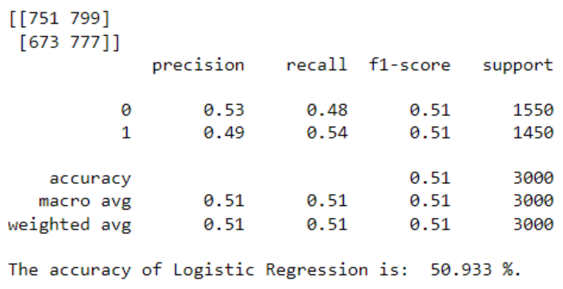
A confusion matrix has the count of the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) among actual data and prediction from the ML model. A classification reports gives the full report of the classification with parameters like recall, precision, f1-score and support for the outcome class. Meanwhile, accuracy score gives the accuracy of the trained model after evaluating it using test data.

The analysis result in Figure 15, 16, 17, 18 and 19 indicate the confusion matrix of each models were the table on the left top, followed by classification report of precision, recall and f1-score in the middle where 0 and 1 represent stroke (1) and non-stroke (0) class subsequently and the last line were the result of accuracy percentage. In this study the main objective is to compare the accuracies of stroke prediction across different machine learning algorithm models. Comparing all accuracy percentage for all models, the Logistic Regression model give the highest performance of accuracy with 50.933% and outperform other 4 ML models.



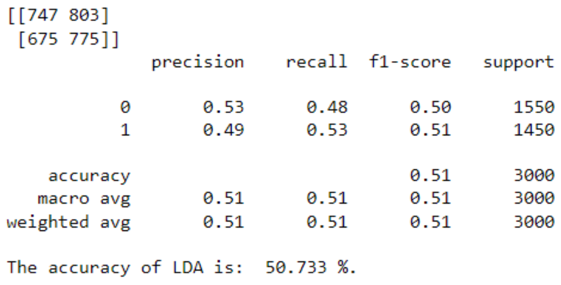
```
[[743 807]
 [746 704]]
              precision    recall  f1-score   support

           0       0.50      0.48      0.49      1550
           1       0.47      0.49      0.48      1450

    accuracy                           0.48      3000
   macro avg       0.48      0.48      0.48      3000
weighted avg       0.48      0.48      0.48      3000


The accuracy of Decision Tree is:  48.233 %.
```

Figure 15: Prediction reports from confusion matrix, classification report and accuracy of Decision Tree Model.



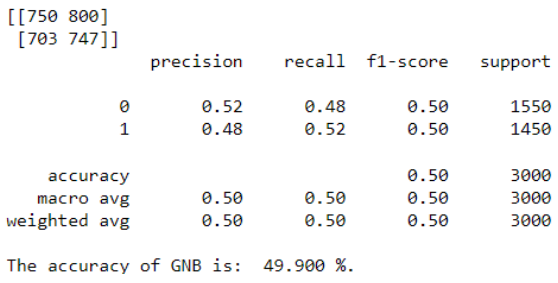
```
[[751 799]
 [673 777]]
              precision    recall  f1-score   support

           0       0.53      0.48      0.51      1550
           1       0.49      0.54      0.51      1450

    accuracy                           0.51      3000
   macro avg       0.51      0.51      0.51      3000
weighted avg       0.51      0.51      0.51      3000

The accuracy of Logistic Regression is:  50.933 %.
```

Figure 16: Prediction reports from confusion matrix, classification report and accuracy of Logistic Regression Model.



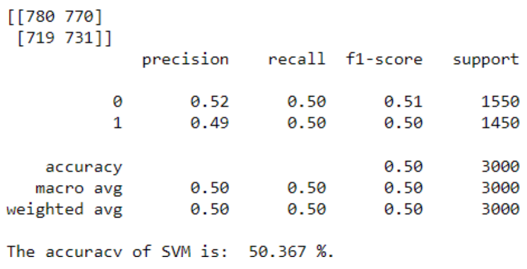
```
[[747 803]
 [675 775]]
              precision    recall  f1-score   support

           0       0.53      0.48      0.50      1550
           1       0.49      0.53      0.51      1450

    accuracy                           0.51      3000
   macro avg       0.51      0.51      0.51      3000
weighted avg       0.51      0.51      0.51      3000

The accuracy of LDA is:  50.733 %.
```

Figure 17: Prediction reports from confusion matrix, classification report and accuracy of Linear Discriminant Analysis Model.



```
[[750 800]
 [703 747]]
              precision    recall  f1-score   support

           0       0.52      0.48      0.50      1550
           1       0.48      0.52      0.50      1450

    accuracy                           0.50      3000
   macro avg       0.50      0.50      0.50      3000
weighted avg       0.50      0.50      0.50      3000

The accuracy of GNB is:  49.900 %.
```

Figure 18: Prediction reports from confusion matrix, classification report and accuracy of Gaussian NaiveBayes Model.



```
[[780 770]
 [719 731]]
              precision    recall  f1-score   support

           0       0.52      0.50      0.51      1550
           1       0.49      0.50      0.50      1450

    accuracy                           0.50      3000
   macro avg       0.50      0.50      0.50      3000
weighted avg       0.50      0.50      0.50      3000

The accuracv of SVM is:  50.367 %.
```

Figure 19: Prediction reports from confusion matrix, classification report and accuracy of Support Vector Machine Model.

Conclusion

In this study 5 different machine learning algorithms were chosen to be implemented with stroke dataset after rigorous pre-processing steps applied to the raw data. All of the algorithms were choosen from some variation of mechanism. For example, Decision Tree and Logistic Regression algorithms are both tree-based mechanism; Gaussian Naive Bayes and Linear Discriminant Analysis both are probabilistic statistical mechanism; and lastly Suport Vector Machine is functional-based algorithm. In the end, Logistic Regression Model gives the highest performance of accuracy which technically known as simple algorithm that suitable for binary class dataset and very similar to simple neural network. Future study will explore more possibilities of algorithm mechanism and behaviour that give strong indication of the performance on specific dataset.

References:

Atha, R. (2022, January 30). *Building Calssification Model with Python*. Analytics Vidhya. https://medium.com/analytics-vidhya/building-classification-model-with-python-9bdfc13faa4b

Department of Statistics Malaysia. *Statistics on Causes of Death Malaysia 2020*. https://www.dosm.gov.my/v1/index.php?r=column/cthemeByCat&cat=401&bul_id=QTU5T0dKQ1g4MHY xd3Zp MzhEMzdRdz09&menu_id=L0pheU43NWJwRWVSZklWdzQ4TlhUUT09 (accessed 1 March 2021).

Hwong, W. Y., Ang, S. H., Bots, M. L., Sivasampu, S., Selvarajah, S., Law, W. C., Latif, L. A., & Vaartjes, I. (2021). Trends of stroke incidence and 28-day all-cause mortality after a stroke in Malaysia: A linkage of national data sources. *Global Heart*, *16*(1). https://doi.org/10.5334/GH.791

Kavish (2024, Februari 23). *Handling Imbalanced Data with Imbalance-Learn in Phyton*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/05/handling-imbalanced-data-with-imbalance-learn-in-python/

Remi, M. (2021, April 6). *How to Classify Data in Phyton using Scikit-learn*. ActiveState. https://www.activestate.com/resources/quick-reads/how-to-classify-data-in-python/

Tan, K. S., & Venketasubramanian, N. (2022). Stroke Burden in Malaysia. *Cerebrovascular Diseases Extra*, *12*(2), 58–62. https://doi.org/10.1159/000524271