Stroke Prediction with Data Science

by

Kelvin Ng Han Yao

Table of Contents

STROKE PREDICTION WITH DATA SCIENCE

ABSTRACT

Stroke, a disease which impact on arteries connecting to brain and occur when a blood vessel is blocked from transferring nutrients and oxygen to the brain. It is the third major leading causes of death in Malaysia, and it reach 9.80% of Malaysia's total deaths in year 2018. With early detection on stroke disease, various of preventive action can be took to reduce the damage dealt to the stroke patient, therefore, numerous of research is done all over the world to predict stroke with data. In this assignment, 4 machine learning model approach which are Naïve Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM) have been built for stroke prediction. Here, One-hot encoding has been used to convert categorical data into binary variable to provide more detail information in model training, synthetic minority over-sampling technique (SMOTE) has been used for class balancing in train dataset and min-max normalization technique is used to convert all numeric value into common scale. After analyzing and comparing between the 4-model built, the optimum stroke prediction model among the 4 model is 93.59% accuracy.

INTRODUCTION, RESEARCH GOAL & OBJECTIVES

Data science has become a trend in year 2021. A lot of industry including healthcare sector using data to improve the production rate and efficiency in their sector. Healthcare sector is one of the important sectors that used high accuracy prediction model to take preventive action on various disease and to reduce disease mortality rate of a country. Stroke disease has been selected in this assignment because it is one of the major diseases causes death over the world and the most important reason is stroke can be prevented and cure if it able to be predict and detected at early stage and therefore model accuracy is very important for a disease prediction model as early detection or prediction can lead to earlier preventive action and save a life.

The objective and research goal of this assignment is to train 4 machine learning model which are Naïve Bayes, Logistic Regression, Random Forest, and SVM then compare among the model to retrieve a stroke prediction model with the highest accuracy. In this assignment 12 clinical features including patient demography, body mass index, average glucose level, high blood pressure status and heart disease status are collected and used to train the model. Data are collected from more than 5000 individuals while all the sensitive data has been masked and removed. These datasets are integrated and transformed to remove abnormal and missing value while multiple data pre-process technique such as one-hot encoding, min-max normalization and class balancing has been applied on the dataset to train a better stroke prediction model.

SECTION 1: RELATED WORKS

While searching for stroke prediction model relevant previous work, search result shows the number of stroke prediction model is increasing year by year. Various of model are used in building stroke prediction model but the dataset used in each related work are relatively small to fully explore the potential of machine learning model and shorten the duration to train the model. This assignment also used relatively small stroke prediction dataset to build and train the model.

For handling null value or empty string, most of the relevant work will handle null value with simple method such as single imputation or complete case analysis which is also used in this assignment. Advantage of using complete case analysis is it will remove noise variable and retain only useful and accurate data, but it will also further reduce the size of the dataset and affect the efficiency of model training process.

Most used machine learning model in each relevant work were Random Forest, SVM, Artificial Neural Network and Decision Tree. It's hard to compare and decide best performance model in each relevant work as the data characteristic used is different in each relevant work but SVM perform better in most of the studies follow up with Random Forest and Artificial Neural Network. Random Forest and SVM are also used in this assignment but the model performance result shows that Random Forest perform better than SVM in this assignment.

Model validation is important in model building so that the model can have certain accuracy when dealing with real world industry data. In most of the relevant work, common internal validation methods are used such as train test data split and cross validation. Train test data split validation is used in this assignment with a ratio of 0.7 because cross validation with multiple folds will require more training time and consume more local machine resource which will slow down the process of this assignment, therefore, split validation that faster speed in training model is used in this assignment.

Performance metric is important to determine which model perform better. In most of the relevant work, F-score is used when dealing with imbalance data while accuracy is used

when dealing with balance data. In this assignment, accuracy will be the main performance metric to determine the performance of each model built.

Description	**Majority Method**
Dataset Size	Small dataset size with average of 43 attributes and 3000 observations
Handling Missing Data	Single Imputation, Complete Case Analysis
Machine Learning Model	Random Forest, SVM, Artificial Neural Network, Decision Tree
Model Validation	Split Validation, Cross Validation.
Performance Metric	Accuracy, F-score

Table 1

Summary table is created above based on review on previous relevant work. Every process built and model trained in this assignment will be done based on the reference on the above summary table.

SECTION 2: METHODS

Dataset Description

This assignment used a stroke prediction dataset which contain 5110 observations with 13 attributes. This dataset contains patient id to ensure the uniqueness of each observation and 12 other attributes which will be explained in detail. Gender, categorical data with categories of male and female. Age, continuous data with normal range between 0 to 120. Age categories, categorical data with categories of infants, adults, children, older adults, and teens. Smoking status, categorical data with categories of formerly, never and smokes. Married status, categorical data with categories of yes and no which determine whether an individual is married previously. Employment status with categories of children, government job, never worked, private and self-employed. Region type, categorical data with categories of rural and urban. Body mass index, continuous data with normal range between 20 to 60. Average glucose level, continuous data with normal range between 60 to 240. High blood pressure, categorical data with categories of 0 and 1 which determine whether an individual is having high blood pressure. Heart disease, categorical data with categories of 0 and 1 which determine whether an individual is having a heart disease. Stroke status, target variable in this dataset and categorical data with categories of yes and no which determine whether an individual is having stroke

Learning Technique and Library Used

The learning technique used in this assignment is supervised learning technique and the package and library used is listed in figure below.

```
library(psych)
library(DataExplorer)
library(dplyr)
library(ggplot2)
library(caTools)
library(mltools)
library(data.table)
library(reshape2)
library(lattice)
library(grid)
library(UBL)
library(e1071)
library(caret)
library(randomForest)
library(tictoc)
```

Figure 1

The machine learning method use in this assignment are Naïve Bayes, Logistic Regression, Random Forest, and SVM as these methods are good classifier to deal with categorical target variable. The metric used in this assignment to evaluate the performance of the model is accuracy.

SECTION 3: DATASET PREPARATION

Data Cleansing

Before start with dataset preparation, seed is set at 2021 to ensure getting back same result each time the process is rerun. When checking dataset using str(data), wrong data type is detected on body_mass_index which is character type instead of numeric type as figure below.

```
> str(data)
'data.frame':   5110 obs. of  13 variables:
 $ patient_id         : int  292 573 794 1292 1343 1747 1918 1927 2264 2408 ...
 $ gender             : chr  "Male" "Female" "Female" "Female" ...
 $ age                : int  82 75 83 69 55 80 52 64 81 68 ...
 $ age_categories     : chr  "OlderAdults" "OlderAdults" "OlderAdults" "OlderAdults" ...
 $ smoking_status     : chr  "Never" "Never" "Formerly" "Unknown" ...
 $ married_status     : chr  "Yes" "Yes" "No" "Yes" ...
 $ employment_status  : chr  "SelfEmployed" "SelfEmployed" "Private" "Private" ...
 $ region_type        : chr  "Rural" "Urban" "Rural" "Rural" ...
 $ body_mass_index    : chr  "30.4" "25.6" "25.5" "38.3" ...
 $ avg_glucose_level  : num  89.5 73 82 209.1 69.2 ...
 $ high_blood_pressure: int  0 0 1 0 0 1 1 0 1 1 ...
 $ heart_disease      : int  0 0 1 0 0 0 0 0 0 0 ...
 $ stroke_status      : chr  "Yes" "Yes" "Yes" "Yes" ...
```

Figure 2

The body_mass_index data type is character due to “N/A” text value in the column, therefore, as.numeric() function is used to automatically convert “N/A” text value to null value and convert the other data to numeric data type as figures below.

```
# use as.numeric will automatically convert non numeric value to N/A value
data$body_mass_index = as.numeric(data$body_mass_index)
```

Figure 3

```
> class(data$body_mass_index)
[1] "character"
> data$body_mass_index = as.numeric(data$body_mass_index)
Warning message:
NAs introduced by coercion
> class(data$body_mass_index)
[1] "numeric"
```

Figure 4

Patient id is dropped from the dataset and an extra category of “Other” is removed from the Gender column as figure below.

```r
# Drop ID column
df <- select(data, -patient_id )

# Remove 'Other' category from gender column
df = df[(df$gender == 'Male' | df$gender == 'Female'),]
```

Female	Male	Other
2993	2115	1

→

Female	Male
2993	2115

Figure 5

Both missing data and empty string is detected in avg_glucose_level, employment_status, married_status, heart_disease, high_blood_pressure, body_mass_index, smoking_status and removed from the dataset as figure below.

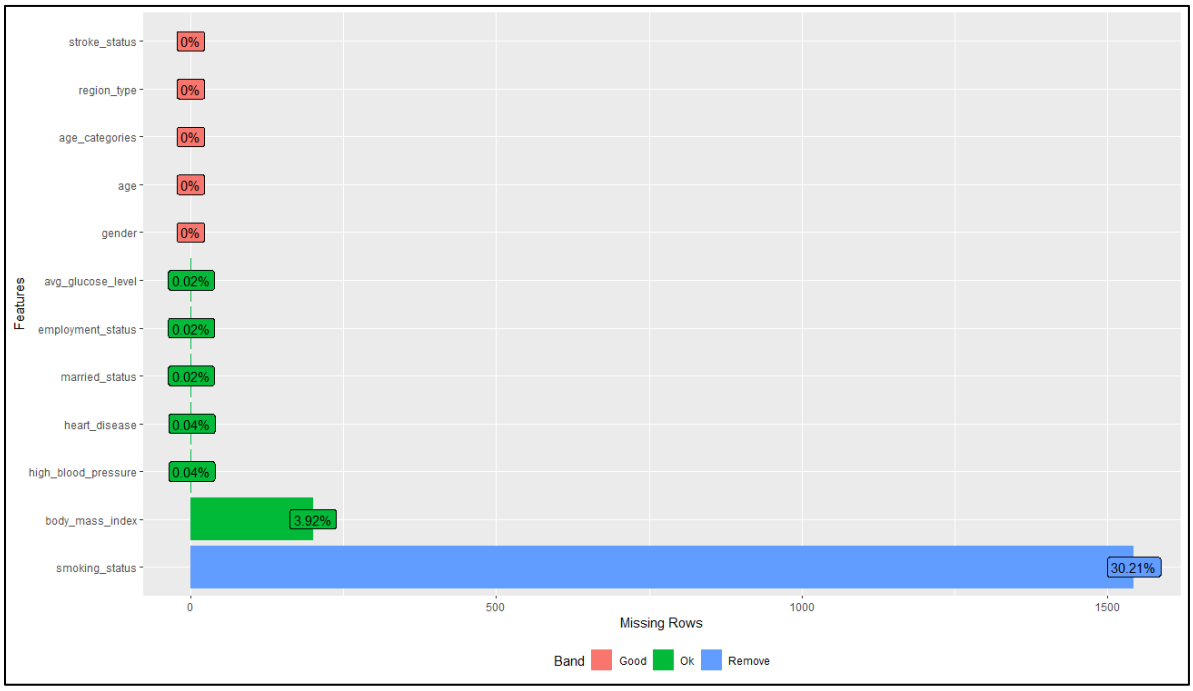

Figure 6

Outlier value and abnormal value is detected and removed from body_mass_index, avg_glucose_level and age as figure below.

```
> describe(df)
                    vars    n   mean     sd median trimmed   mad    min      max    range
gender*                1 5108   1.41   0.49   1.00    1.39  0.00   1.00     2.00     1.00
age                    2 5108  44.18  22.68  46.00   44.57 26.69 -61.00    83.00   144.00
age_categories*        3 5108   1.96   1.41   1.00    1.76  0.00   1.00     5.00     4.00
smoking_status*        4 5108   3.58   1.09   3.00    3.61  1.48   1.00     5.00     4.00
married_status*        5 5108   2.66   0.48   3.00    2.69  0.00   1.00     3.00     2.00
employment_status*     6 5108   4.49   1.28   5.00    4.62  0.00   1.00     6.00     5.00
region_type*           7 5108   1.51   0.50   2.00    1.51  0.00   1.00     2.00     1.00
body_mass_index        8 4908  28.43  39.21  27.10   27.35  6.97 -37.70  2718.00  2755.70
avg_glucose_level      9 5107 109.29 372.42  89.88   95.86 26.06  53.12 26521.76 26468.64
high_blood_pressure   10 5106   0.10   0.30   0.00    0.00  0.00   0.00     1.00     1.00
heart_disease         11 5106   0.05   0.23   0.00    0.00  0.00   0.00     1.00     1.00
stroke_status*        12 5108   1.05   0.21   1.00    1.00  0.00   1.00     2.00     1.00
```

Figure 7

```
# Check for outlier and abnormal data
describe(df)

# Remove body_mass_index with record more than 100 which is out of normal
range
df = df[!(df$body_mass_index > 100),]

# Remove avg_glucose_level with record more than 300 which is out of normal
range
df = df[!(df$avg_glucose_level > 300),]

# Remove negative value in age and body_mass_index
df = df[!( df$age < 0 | df$body_mass_index < 0),]
describe(df)
```

Figure 8

Children under age 14 in Malaysia are not allowed to be employed, therefore, data are checked for age under 14 and employment_status are not “Children” as figure below.

```
> # Check and remove value for age under 14 but employment status are not 'Children'
> subset(df,df$age < 14 & df$employment_status!="Children",select=c(age,employment_status))
    age employment_status
47    1           Private
165   1      SelfEmployed
```

Figure 9

Correlation Matrix

Correlation matrix heatmap is built to determine the correlation between all numeric attributes as figure below.

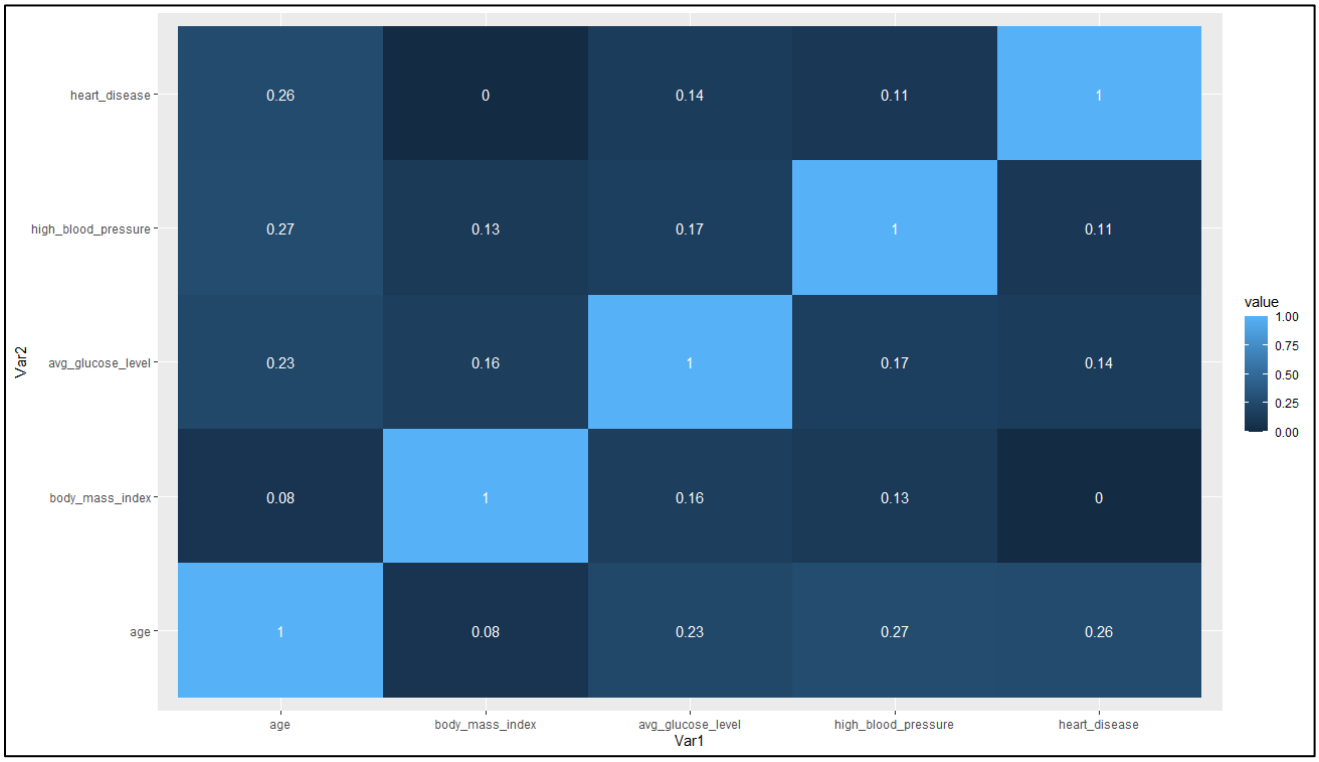

Figure 10

From the figure above, there are no correlation between attribute that is more than 0.8, therefore, no attribute is removed from dataset at this step.

One-hot Encoding

One-hot encoding technique is used to convert categorical variable into binary variable so that the variables can provide more detail information for model training. Categorical column need to be convert to factor type only able to apply with one-hot() function as figure below.

```
# One-hot encoding categorical data with data table
col_names <- c('gender','age_categories','smoking_status','married_status',
'employment_status', 'region_type')

df[,col_names] <- lapply(df[,col_names] , factor)

df2 <- one_hot(as.data.table(df))
```

Figure 11

Stroke_status which is the target variable is labelled for stroke prediction in later phase as figure below.

```
# Labeling the target variable values
# 'No' to 0 while 'Yes' to 1
df2$stroke_status <-
factor(df2$stroke_status, levels=c("No","Yes"), labels=c("0", "1"))
```

Figure 12

Train - Test Split

Dataset is split into train and test data with a ratio of 0.7 and 0.3 respectively as figure below.

```
> dim(train)
[1] 2400   24
> dim(test)
[1] 1014   24
```

Figure 13

Class Balancing

Once the dataset is split into train and test data, class balancing is performed on the train data. The reason class balancing is performed only on train data is because train data need balance class to train model while test data need to remain as similar as real world data to get a correct accuracy.

```
> table(train$stroke_status)

   0    1
2279  121
```

Figure 14

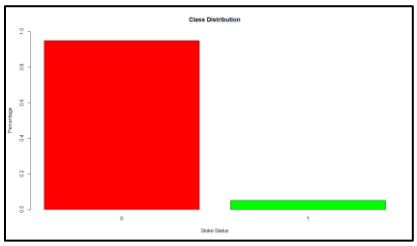

Figure 15

After perform class balancing using SMOTE technique to oversample the dataset, the distribution of target variable is balanced.

```
> table(train_bal$stroke_status)

   0    1
1200 1200
```

Figure 16

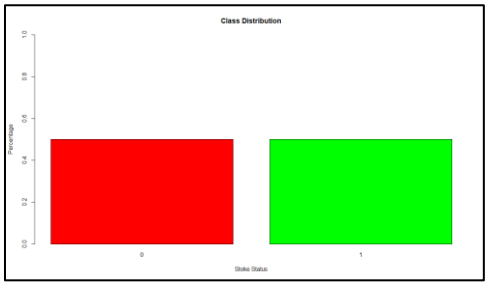

Figure 17

Min-max Normalization

Min-max normalization is performed on both train and test data as preprocess modelling need to be exactly same for train data, test data as well as real world data when the model is in production.

```
# Min- Max Normalization for numerical value
# One-hot encoded column will remain 0 and 1
normalize = function(x) { return ((x - min(x)) / (max(x) - min(x)))}

# Min- Max Normalization train data
train_bal_norm_tmp <-
as.data.frame(lapply(select(train_bal,-stroke_status), normalize))
train_bal_norm <-
cbind(train_bal_norm_tmp, stroke_status=train_bal$stroke_status)

# Min- Max Normalization test data
test_norm_tmp <-
as.data.frame(lapply(select(test, -stroke_status), normalize))
test_norm <-
cbind(test_norm_tmp,stroke_status= test$stroke_status)
```

Figure 18

SECTION 4: ALGORITHMS MODEL IMPLEMENTATION & MODEL VALIDATION

Naïve Bayes

The first model build is Naïve Bayes model. The hyperparameter of Naïve Bayes model is tuned with grid search method as figure below.

```
# Naive Bayes
# Naive Bayes model tuning using grid search method
search_grid =
expand.grid(usekernel = c(TRUE, FALSE),fL = 0:5,adjust = seq(0, 5, by = 1))

nb_tune_model =
train(xtrain, ytrain, 'nb', metric="Accuracy",tuneGrid = search_grid)
```

Figure 19

```
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

Figure 20

3 hyperparameter in Naïve Bayes were tuned which are fL, usekernel and adjust. fL hyperparameter allow user to include Laplace smoother, usekernel hyperparameter allows user to use a kernel density estimate for continuous variables against a guassian density estimate while adjust hyperparameter is referring to the bandwidth of kernel density. The final value used to build Naïve Bayes model after tuned were fL = 0, usekernel = TRUE and adjust = 1.

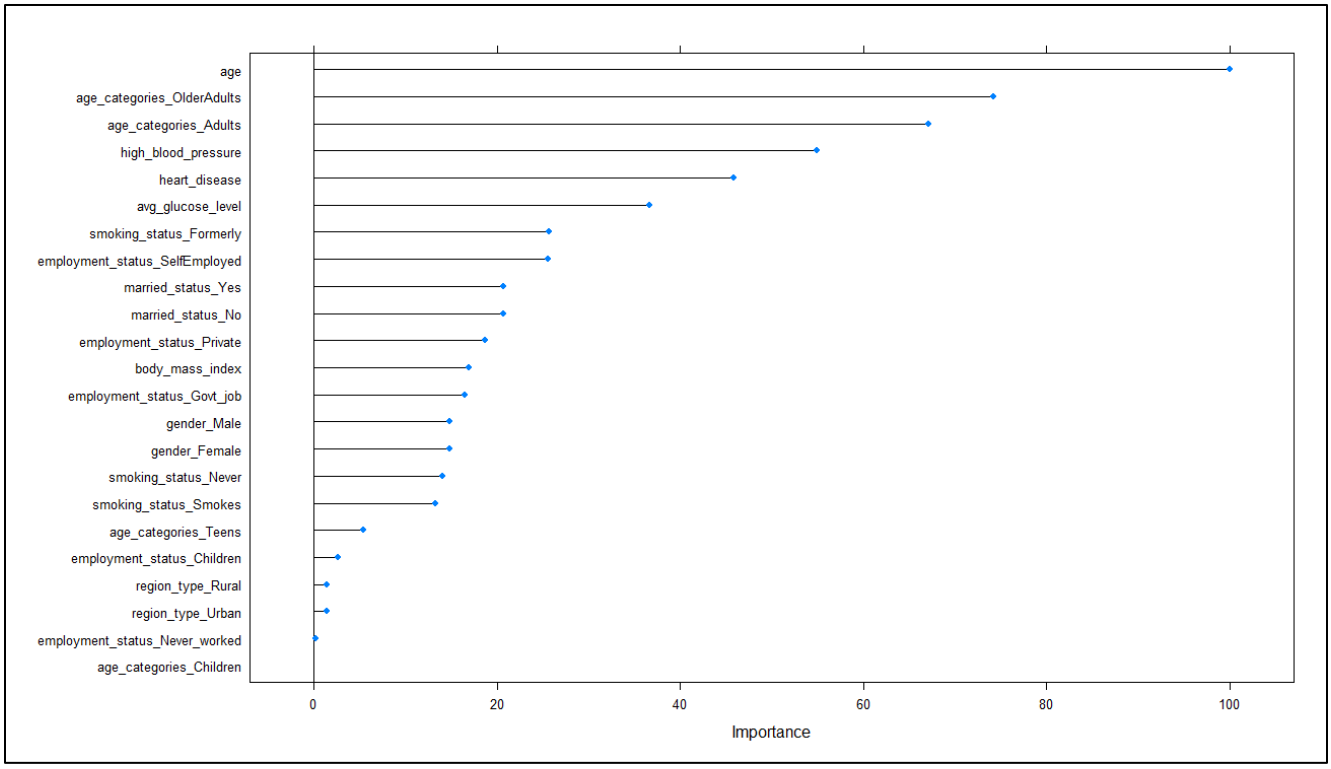

Figure 21

	Importance
age	100.000
age_categories_OlderAdults	74.171
age_categories_Adults	67.088
high_blood_pressure	54.911
heart_disease	45.894
avg_glucose_level	36.619

Figure 22

From the above figure, it's clearly shown that age attribute plays an importance role in model training. The higher the age, the older the age categories, the more importance the role in Naïve Bayes stroke prediction model building. Naïve Bayes stroke prediction model are train again under feature selection with only attributes with importance above 0.3.

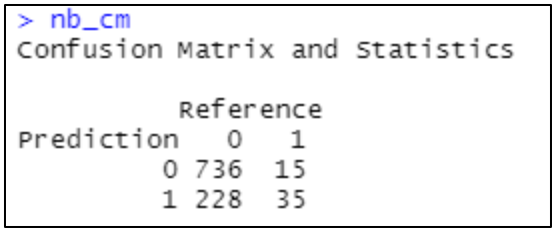

Figure 23

The above figure is the confusion matrix of Naïve Bayes model, and the precision of Naïve Bayes model can be calculated which is 35 / (35 + 15) = 0.7 while the recall of Naïve Bayes model is 35 / (35 + 228) = 0.13.

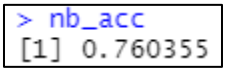

Figure 24

The accuracy of Naïve Bayes stroke prediction model is 76.03%.

Logistic Regression

The next model build is Logistic Regression model. The Logistic Regression model is train as figure below.

```
# Logistic Regression
# Logistic Regression model training
logreg_model = glm(stroke_status ~., train_final, family = binomial)
```

Figure 25

```
Call:
glm(formula = stroke_status ~ ., family = binomial, data = train_final)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.62886  -0.68491   0.08487   0.76979   2.35228

Coefficients: (6 not defined because of singularities)
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -1.648e+01  4.531e+02  -0.036  0.97099
gender_Female                 -1.376e-03  1.212e-01  -0.011  0.99094
gender_Male                           NA         NA      NA       NA
age                            8.696e+00  6.065e-01  14.336  < 2e-16 ***
age_categories_Adults          1.058e+01  4.531e+02   0.023  0.98137
age_categories_Children        2.773e-01  1.296e+03   0.000  0.99983
age_categories_OlderAdults     9.534e+00  4.531e+02   0.021  0.98321
age_categories_Teens                  NA         NA      NA       NA
smoking_status_Formerly       -6.655e-01  1.680e-01  -3.961 7.46e-05 ***
smoking_status_Never          -7.975e-01  1.564e-01  -5.099 3.41e-07 ***
smoking_status_Smokes                 NA         NA      NA       NA
married_status_No             -1.992e-01  2.166e-01  -0.920  0.35773
married_status_Yes                    NA         NA      NA       NA
employment_status_Children     2.533e-01  6.894e+02   0.000  0.99971
employment_status_Govt_job     3.655e-01  1.912e-01   1.912  0.05589 .
employment_status_Never_worked -9.684e+00  7.975e+02  -0.012  0.99031
employment_status_Private      2.916e-01  1.443e-01   2.020  0.04333 *
employment_status_SelfEmployed        NA         NA      NA       NA
region_type_Rural             -8.486e-02  1.158e-01  -0.733  0.46373
region_type_Urban                     NA         NA      NA       NA
body_mass_index                5.075e-01  4.786e-01   1.060  0.28897
avg_glucose_level              6.116e-01  2.193e-01   2.789  0.00529 **
high_blood_pressure            1.036e+00  1.461e-01   7.093 1.31e-12 ***
heart_disease                  5.806e-01  1.864e-01   3.115  0.00184 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 26

Feature selection is performed in Logistic Regression model based on significant attributes with 2 * and above based on above figure.

```
Call:
glm(formula = stroke_status ~ ., family = binomial, data = lgxtrain)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.59259  -0.69776   0.03112   0.74190   2.25310

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)             -4.1684     0.2188 -19.055  < 2e-16 ***
age                      6.1232     0.3039  20.146  < 2e-16 ***
smoking_status_Formerly -0.6485     0.1647  -3.937 8.25e-05 ***
smoking_status_Never    -0.8466     0.1503  -5.633 1.78e-08 ***
high_blood_pressure      0.9539     0.1442   6.613 3.77e-11 ***
heart_disease            0.6131     0.1828   3.355 0.000795 ***
avg_glucose_level        0.7147     0.2045   3.496 0.000473 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3327.1  on 2399  degrees of freedom
Residual deviance: 2246.2  on 2393  degrees of freedom
AIC: 2260.2

Number of Fisher Scoring iterations: 5
```

Figure 27

After feature selection is performed only 7 attributes is retained in Logistic Regression as figure above.

```
> logreg_cm
      logreg_model_predict
ytest   0   1
    0 694 270
    1  11  39
```

Figure 28

The above figure is the confusion matrix of Logistic Regression model, and the precision of Logistic Regression model can be calculated which is 39 / (39 + 270) = 0.13 while the recall of Logistic Regression model is 39 / (11 + 39) = 0.78.

```
> logreg_acc
[1] 0.7228797
```

Figure 29

The accuracy of Logistic Regression stroke prediction model is 72.28%.

Random Forest

The third model build is Random Forest model. The Random Forest model is tuned using grid search method as figure below.

```
# Random Forest
# Random Forest model tuning using grid search method
tunegrid <- expand.grid(.mtry=c(1:8))

# The final value used for the model was mtry = 6
rf_tune_model <-
train(stroke_status~.,data = train_final, method="rf", metric="Accuracy",
tuneGrid=tunegrid)
```

Figure 30

Only one hyperparameter in Random Forest is tuned which is the mtry hyperparameter as tuning on both mtry and ntree parameter will consume a lot of resource of the local machine and the duration took few hours, therefore, in this assignment only mtry hyperparameter is tuned for Random Forest. Mtry hyperparameter will randomly sample variables based on the number assigned as candidate at each split. The final value of mtry hyperparameter tune from grid search method is mtry = 6, therefore, Random Forest model are train with hyperparameter mtry = 6.

```
> rf_cm
                 ytest
rf_model_predict   0   1
               0 945  46
               1  19   4
```

Figure 31

The above figure is the confusion matrix of Random Forest model, and the precision of Random Forest model can be calculated which is 4 / (46 + 4) = 0.08 while the recall of Random Forest model is 4 / (19 + 4) = 0.17.

```
> rf_acc
[1] 0.9358974
```

Figure 32

The accuracy of Random Forest stroke prediction model is 93.59%.

Support Vector Machine (SVM)

The next model build is SVM model. The SVM model is tuned using grid search method as figure below.

```
# SVM
# SVM model tuning using grid search method
svm_tune_model =
tune(svm, stroke_status~., data=train_final, ranges = list(epsilon = seq
(0, 1, 0.1), cost = 2^(0:2)))

# SVM optimum model, epsilon = 0, cost = 4, kernel = radial
svm_opt_model = svm_tune_model$best.model
```

Figure 33

3 hyperparameter in SVM were tuned which are epsilon, cost, and kernel. Epsilon hyperparameter represent the margin that user allow and tolerate that penalty is not given to error, cost hyperparameter also referring to cost of misclassification where user decide how much data that SVM are allowed to misclassify while kernel hyperparameter is referring to the method of mathematical function used to deal with input data and transform into. The final value used to build SVM model after tuned epsilon = 0, cost = 4, kernel = radial.

```
> svm_cm
          Actual
Predicted   0   1
        0 910  43
        1  54   7
```

Figure 34

The above figure is the confusion matrix of SVM model, and the precision of SVM model can be calculated which is 7 / (7 + 43) = 0.14 while the recall of SVM model is 7 / (7 + 54) = 0.11.

```
> svm_acc
[1] 0.9043393
```

Figure 35

The accuracy of SVM stroke prediction model is 90.43%.

SECTION 5: ANALYSIS & RECOMMENDATIONS

Total of 4 models have been trained in this assignment which are Naïve Bayes, Logistic Regression, Random Forest and SVM. As small dataset gets overfitted easily, therefore, these 4 models is chosen because they can perform better when dealing with small dataset. All four model has been trained and tested with preprocessed data and preprocess model and the main performance metric which is accuracy for the 4 model are recorded as figure below.

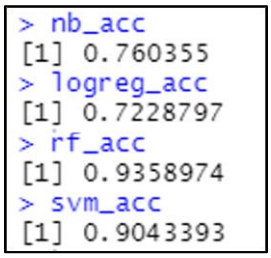

Figure 36

Based accuracy of 4 trained model in this assignment, Random Forest model will be the champion model in this assignment with the highest accuracy.

Model	Precision	Recall	Accuracy
Naïve Bayes	70%	13%	76.03%
Logistic Regression	13%	78%	72.28%
Random Forest	8%	17%	93.59%
SVM	14%	11%	90.43%

Table 2

Naïve Bayes model and Logistic Regression model have a high precision and recall respectively compared to Random Forest model and SVM model which mean Naïve Bayes model can get the true positive value and manage to determine patient who are having stroke while Logistic Regression model manage to get a high amount of false negative which mean logistic regression can't determine the patient who having stroke but it somehow predicted

those who are not having stroke might have stroke in the future which preventive action can be take in advance.

Although Random Forest model and SVM model don't have high precision rate and recall rate, but their accuracy is higher compared to Naïve Bayes model and Logistic Regression model which are 93.59% and 90.43% respectively.

SVM outperform Logistic Regression because SVM finds the optimum distance between line and support vectors to separate class and this lower risk of classification error, while logistics regression can have different decision line with various weight that close to optimal point.

Random Forest outperformed Logistic Regression because the explanatory variable in this dataset is more while Logistic Regression can only perform better when the explanatory variable is more than noise variable.

SVM outperform Naïve Bayes because SVM deal with the interaction between attribute until certain level, but Naïve Bayes treat all attributes independently which cant interpret a deeper level of relationship between attributes but Naïve Bayes train faster than SVM as only probability of each class needed to be calculated in Naïve Bayes.

Random Forest outperformed Naïve Bayes because it's have a much more complex and large model size compare to Naïve Bayes while Naïve Bayes is simple model and cannot cater complicated data behavior, therefore, Random Forest have better performance than Naïve Bayes with a dataset with complex behavior but advantage of Naïve Bayes is that it can quickly adapt to the changes in any new dataset while Random Forest need to rebuild whenever there's changes in dataset else it will lead to overfitting.

In most of the related work, SVM and Random Forest are used more in stroke prediction model building compared to Naïve Bayes and Logistic Regression as in the real world industry, stroke is a critical disease that can ruin a patient life, therefore, a model with higher accuracy will be prioritize in the real world situation and same case in this assignment where Random Forest stroke prediction model will be recommended as it is the champion model and it has the highest accuracy compare to the other model.

CONCLUSION

This is assignment is a great exposure as it's my first time of building machine learning model from end to end by myself. It's a great boost in knowledge about applied machine learning in real world use case and improve my understanding on how to train machine to deal with industry problem. The great thing in this assignment is gaining knowledge about data transformation, data handling, feature selection, correlation matrix, one hot encoding, class balancing, data splitting, training various type of model and determine model performance with different metric. Although the model built is at a basic level and model is not well trained in this assignment due to blind spot of knowledge and size of dataset but it's still a great experience for me.

For future work, there are a lot of different technique and model that I wish to try with better machine spec or even in server like Artificial Neural Network (ANN), Deep Learning and train better model with high precision and accuracy that can tackle real world industry problem.

BIBLIOGRAPHY

Authors, T. (2019). Comparing regression, naive Bayes, and random forest methods in the prediction of individual survival to second lactation in Holstein cattle. *Journal of Dairy Science*, 13.

Benjamin Letham, C. R. (2015). Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.* , 22.

Combi, C. (2020). Artificial Intelligence in Medicine. *Artificial Intelligence in Medicine*, 568.

Dave, P. (2021). *How to create a Stroke Prediction Model?* Retrieved 12 8, 2021, from https://www.analyticsvidhya.com/blog/2021/05/how-to-create-a-stroke-prediction-model/

Kuo-Liong Chien, M. P., Ta-Chen Su, M. P., Hsiu-Ching Hsu, P., Wei-Tien Chang, M. P., Pei-Chun Chen, P., Fung-Chang Sung, P., . . . Yuan-Teh Lee, M. P. (2021). *Constructing the Prediction Model for the Risk of Stroke in.* Taiwan: American Heart Association, Inc.

Lemons, K. (2020). A Comparison Between Naïve Bayes and Random Forest to Predict Breast Cancer. *International Journal of Undergraduate Research and Creative Activities*, 6.

Rich Caruana, Y. L. (2015). Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. *RESEARCH-ARTICLE*, 10.

Risk Factors and Prediction of Stroke in a Population with High Prevalence of Diabetes: The Strong Heart Study. (2017, May). Retrieved from Scientific Research: https://www.scirp.org/journal/paperinformation.aspx?paperid=76573

Sailasya, G., & Kumari, G. L. (2021). Analyzing the Performance of Stroke Prediction using. *International Journal of Advanced Computer Science and Applications*, 7.

Sida Wang, C. D. (n.d.). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. 5.

Stroke Prediction using Data Analytics and Machine Learning. (2021, June 22). Retrieved from TechTarget: https://www.datasciencecentral.com/profiles/blogs/stroke-prediction-using-data-analytics-and-machine-learning

Sundaresan, B. (2021, July 3). *Stroke Prediction*. Retrieved from RPubs: https://rpubs.com/bharath2925/strokeprediction

Teoh, D. (2018). Towards stroke prediction using. *Teoh BMC Medical Informatics and Decision Making*, 11.

Veena Potdar, L. S. (2021). A Survey on Stroke Disease Classification and Prediction using Machine Learning Algorithms. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*.

Vida, Avula, V., Chaudhary, D., Shahjouei, S., Khan, A., Griessenauer, C. J., . . . Ramin. (2021). Prediction of Long-Term Stroke Recurrence Using Machine Learning Models. *JCM*.

Zdrodowska, M. (2019). ATTRIBUTE SELECTION FOR STROKE PREDICTION. 4.

Zhang, R. (2019). Interpretable Machine Learning Methods for Stroke Prediction. 75.