Stroke Prediction Using Data Mining Techniques

University of Auckland

Qiong Zhou / 365217677 / qzho906

Table of Contents

1. Abstract

Ischemic heart disease, stroke, and chronic obstructive pulmonary disease are the top three causes of death from disease, accounting for 16%, 11%, and 6 % of deaths worldwide respectively (WHO, 2020). However, 90% of stroke occurrences can be preventedanin a early stage. Although the disease itself can develop over a short period of time and without any precursors, it is still possible to classify and predict patients based on early personal history and to offer solutions and advice to patients while reducing the chances of a stroke occurring. One of the main clinical risk factors for stroke is atherosclerosis-induced hypertension, along with many other risk factors including smoking, physical inactivity, unhealthy diet, harmful use of alcohol, atrial fibrillation, elevated lipid levels, obesity, genetic predisposition, stress and depression (World Stroke Organisation, 2022). This study will use data mining techniques to predict, prevent and reduce the occurrence of stroke disease based on a potential patient's personal information, health status, and lifestyle status, reducing the likelihood of a potential patient having a stroke at an early stage.

2. Introduction

In order to achieve the goal of being able to analyze in detail the causes of stroke in stroke patients and to detect early manifestations of signs of stroke. This study should review a large number of existing patient cases and engage them as a dataset to find models, as well as relationships between signs, data mining specialists should be applied.

Personnel. It is clear and confident that it is important to consult doctors and specialists in the field of stroke research, as well as those working in the field of healthcare and rehabilitation for stroke patients, and to obtain clear and detailed information about the disease itself, including the primary, secondary and direct causes of stroke. Regarding the data and information collected from these stroke specialists, a database specialist should be consulted. Since these specialists hope the results of the study will become part of a continuing studying and research process, data warehousing and data cleaning for analysis are required. Also, the information about the patients involves their privacy, the data management must also be considered.

Data. The data required for the study analysis will come primarily from the records of doctors and healthcare professionals, where detailed desensitized data on patients is recorded. For initial studies, the data can be easily found on the internet, furthermore, the study data can be expanded as the study is conducted.

Risks. As the project is primarily concerned with healthcare, the accuracy of the model depends to a large extent on the quality and quantity of the dataset used for the study. The model production process should also be concerned with the risk of privacy breaches to users. Once generated, the models should be further validated by a team of experts who specialize in the treatment and research of stroke patients.

In the following sections, firstly, the practical problem of the research will be identified, secondly, to solve the practical problem, the research problem of this project is stated. Thirdly, the research objectives according to the research question will be clarified. Fourthly, the literature about potential solutions and methodologies that addresses the objectives is explored. Fifthly, the research methodology according to datasets and situations is adopted. Sixthly, the processes that convert data into insights are designed and data are prepared for model building. Seventhly, various algorithms and models are built for comparison and resulting in the best model for implementation. Eighthly, the interpretation of the patterns and results from the model running will be discussed. Lastly, the proposed actions based on the discovered knowledge are carried out.

3. The practical problem

In 2020, stroke is one of the top three causes of death as a disease worldwide, and although the risk of stroke increases with age, strokes can and do occur at any age (Centers for Disease Control and Prevention, 2022). In the United States, one person has a stroke every 40 seconds, and every 3.5 minutes, one person dies from a stroke. The loss of life due to stroke is damaging the business value of individuals and families (Tsao et al., 2022).

In one survey, the majority of respondents (93%) identified sudden numbness on one side as a symptom of a stroke, only 38% knew all the main symptoms and knew to call 911 if someone had a stroke (Fang et al., 2005). patients who arrived at the emergency department within 3 hours of the first symptoms usually had lower rates of disability 3 months after the stroke than those who received delayed care. The stroke should therefore be taken seriously, and early action is important for stroke.

In the field of medical research, it is difficult to give patients any warning before a stroke occurs. To maximize understanding of the warning signs and symptoms of stroke, it is important to act quickly when a potential patient is likely to have a stroke or after a stroke. When emergency treatment is started quickly, the chances of survival are greater.

Therefore, combining data mining with medical information for the prediction can help to slow down and provide accurate warnings and help individuals to gain awareness of the disease. In addition, medical research institutes are looking for the importance of each risk factor. Based on this, database platforms such as Kaggle exist to address such problems, which allow development practitioners to perform a data mining process on the given data in order to discover hidden values and patterns through the data mining process. The main focus of this study is to predict the level of risk of stroke and to discover the importance of each risk factor and its relationship to the occurrence of stroke.

4. The research problem

How can data mining techniques be used to predict a potential patient's risk of stroke based on three aspects: personal information, health status, and lifestyle status?

5. The research objectives

Although the disease of the stroke itself can develop in a very short period of time and without any precursors, it is still possible to classify and predict patients based on their early personal history and to offer solutions and advice to patients while reducing the chances of a stroke.

In this case, the research was commissioned with the following objectives:

- Identify a clear relationship between the patient's current health Index (current illness and BMI) and the stroke.
- Find out the relationship between the patient's age and the chances of stroke.
- Find the relationship between the patient's current life status (smoking status, marital status, work type, living environment, etc.) and the stroke.

- Reduce the chances of stroke by providing predictions based on the patient's current illness and age.
- Use the dataset to train and fit the model to make it suitable for use in data prediction.
  The validation phase uses a selection of healthcare records as a validation process to see the prediction results of a given model, and if the model successfully passes a pre-determined threshold, validation by healthcare professionals is continued.
  The initial use of the model phase allows the healthcare professional to enter a new patient's medical history, view the patient's stroke risk rating, and have the doctor diagnose to see if the system passes the threshold.
  The enhancement and optimization phase should continue to collect the required data and retrain and improve the model to increase accuracy.

The expected outcome of the research:

- Provide detailed information on the causes of the stroke to current patients.
- Provide useful information and advice to people who are at risk of stroke.
- Reduce the likelihood of stroke in potential patients at an early stage.

6. Literature review

Ischemic heart disease, stroke, and chronic obstructive pulmonary disease are the top three causes of death from disease, accounting for 16%, 11%, and 6 % of deaths worldwide respectively (WHO, 2020). Stroke is a condition in which the blood supply to the brain is interrupted, resulting in a lack of oxygen, brain damage, and loss of function. Stroke can lead to permanent damage, including partial paralysis and impairments in speech, understanding, and memory, all of which affect the type and severity of disability depending on the part of the brain affected and the length of time the blood supply is stopped (World Stroke Organization, 2022). The prevalence of stroke has reached epidemic proportions beyond what is thought possible. Globally, one in four adults over the age of 25 will have a stroke in their lifetime (World Stroke Organization, 2022). This year 12.2 million people worldwide will have their first stroke and 6.5 million will die as a result. Worldwide, more than 110 million people have experienced a stroke (World Stroke Organization, 2022). The incidence of stroke increases significantly with age, but over 60% of strokes occur in people under the age of 70 and 16% in people under the age of 50 (World Stroke Organization, 2022).

Although stroke is an acute cerebrovascular disease with high morbidity, mortality, and disability rate. Many factors contribute to stroke, including age, race, gender, geography, and environment, which are not controllable. However, according to a review article published in the Journal of the American College of Cardiology, 90% of strokes are preventable and the key is to manage and treat controllable risk factors (Kleindorfer DO, Towfighi A, Chaturvedi S, et al., 2021). The controllable risk factors are blood pressure, blood lipids, blood sugar, and lifestyle (smoking, alcohol, etc.).

One of the main clinical risk factors for stroke is atherosclerosis-induced hypertension, also, there are many other risk factors including smoking, physical inactivity, unhealthy diet, harmful use of alcohol, atrial fibrillation, elevated blood lipid levels, obesity, genetic predisposition, stress, and depression (World Stroke Organization, 2022).

From the perspective of personal information, age and gender are key factors in the pathology of ischaemic stroke. Age and gender have a complex interactive effect on ischaemic stroke risk and pathophysiology (Roy-O'Reilly & McCullough, 2018). Aging is the strongest non-modifiable risk factor for ischaemic stroke, with older stroke patients having higher mortality and morbidity and poorer functional recovery than younger patients. Importantly, patient age modifies the impact of gender in patients with ischaemic stroke. Early in life, the burden of ischaemic stroke is higher in men, but in the older population, strokes become more common and debilitating in women (Roy-O'Reilly & McCullough, 2018). Lots of papers identified that aging is the most powerful immutable risk factor for stroke events, doubling every 10 years after age 55. Approximately three-quarters of strokes occur in people aged ⩾ 65 years. With the number of people aged ⩾ 65 years expected to increase, the number of strokes in older people is expected to rise, posing a major challenge to clinicians and policymakers in the foreseeable future (Yousufuddin & Young, 2019; Kim et al., 2020; Grefkes & Fink, 2020).

Multimorbidity, ageing, and comorbidities defined as the presence of two or more chronic conditions (e.g., diabetes, hypertension, atrial fibrillation, and coronary and peripheral artery disease), is a constant in older adults and is highly prevalent among those who experience a stroke event, estimated at 89% for those aged 65+ and 60% for those aged <65 years. These chronic conditions, in turn, steadily increase with age, so that the risk of stroke increases progressively with age (Yousufuddin & Young, 2019).

From the perspective of health status, ischemic stroke is far less common in younger adults than in older adults, but the potential causative and risk factors are more diverse. Approximately 10% to 15% of strokes occur in adults aged 18 to 50 years (Singhal et al., 2013; Maaijwee et al., 2014; Ji et al., 2013). The TOAST (Trial of ORG 10172 in Acute Stroke Treatment) classification system is parsimonious, but the pathogenesis of many young stroke patients is more likely to fall into the categories of cardiac embolism, other the pathogenesis of many young stroke patients is more likely to be cardiac embolism, other identified pathogenesis or unidentified pathogenesis rather than large artery atherosclerosis or small vessel occlusion (Adams et al., 1993).

Recent studies in the USA and Europe suggest that ischaemic stroke is increasing in young adults and demonstrate that traditional stroke risk factors common in older people (hypertension, dyslipidaemia, diabetes, smoking and obesity) are also common in young acute stroke patients (Putaala et al., 2009; Singhal et al., 2013). The combination of traditional cardiovascular disease risk factors is increasing in young adults presenting with acute stroke, but there is controversy as to whether or how much these traditional risk factors contribute to the cause of stroke, particularly in those aged less than 40 years.

From the perspective of lifestyle status, the meta-analysis included 14 studies involving 303134 participants (Pan et al., 2019). According to the meta-analysis, smokers had an overall increased risk of stroke compared to non-smokers. Subgroup analyses based on smoking status showed that current smokers had a higher risk of stroke than ex-smokers. There was also a strong association between any type of stroke and smoking status, with current smokers having an increased risk of stroke compared to non-smokers, which was influenced by gender and was higher in men than in women. From the analysis, we also observed that passive smoking increased the overall risk of stroke by 45%, and according to a dose-response meta-analysis, each additional 5 cigarettes per day was associated with a 12% increase in stroke risk (Pan et al., 2019).

7. Research methodology

We will use quantitative methods to look at situations or events that affect people, in this case how a potential patient's personal information, health indices or lifestyle status affects the occurrence of a stroke. Quantitative studies produce objective data that can be clearly communicated through statistics and numbers. We do this in a systematic scientific way, so the research can be replicated by others. Essentially, the goal of quantitative research is to understand the relationship between an independent variable and one or more dependent variables in a population.

The data used for this research task is published on the Kaggle platform. It records more than 43,400 instances of individual records, each with a label indicating whether the person has a stroke condition. And, there are 12 attributes in the dataset. The dataset itself is suitable for the subject of the study, as it covers three areas: lifestyle habits, health conditions, and personal information.

The healthcare information document is collected from the Internet, Kaggle. The data used will be accessed from a website link https://www.kaggle.com/datasets/lirilkumaramal/heart-stroke. The API command is kaggle datasets download -d lirilkumaramal/heart-stroke. It was last updated on 14 August 2021 (Amal, 2020).

The detailed information covered in the dataset is as below:

- **Personal information.** The raw data contain basic information about individuals but does not relate to the disease itself, which includes age and gender.
- **Health Index.** The raw data contain a survey of diseases and basic physical indicators. Diseases investigated include the presence of hypertension and heart disease. The underlying physical indicators include mean glucose levels and BMI.
- **Life status.** The raw data contains living status and habits, whether married or not, type of work, type of residence, and smoking status.

In terms of data quality assumptions, firstly, the dataset is a desensitization dataset, as obviously the dataset did not include sensitive data, for example, name, address, postal code, and so on. Secondly, the dataset is rich enough for training a model for stroke prediction. Thirdly, the dataset is correctly collected from stroke patients or potential patients.

The main problem with this dataset is that it provides limited information about the origin of the dataset itself and only states that this dataset is widely used. This makes the source of this dataset, and in which region it was collected and generated, ambiguous in the study. Once the dataset has been fabricated, all of the findings and predictions in this article will be meaningless. After searching through the discussion section of this dataset, it was found that the data was actually collected by the clinic and the source of the data was deliberately removed due to privacy concerns. After confirming the authenticity of the data sources, the reliability of this dataset was verified and approved for data mining.

8. Design of the processes

The design process used in this study is an iterative machine learning process. It consists of several parts, which are described below.

1. **Preparing the data**
   in this step, the data is collected and cleaned. The main task is to fill in missing values and remove potential data errors (mainly outliers and extreme values). This step includes data

cleaning and improving the quality of the dataset using python pandas or SPSS modeler or pySpark. In this step, all missing values will be removed, and extremes will also be processed to ensure the quality of the dataset. In addition, feature combination or integration will also take place in this step. The data cleaning and processing process will influence the final model selection and prediction discovery.

2. **Extracting features**
   With the data prepared, features can now be selected. In the data mining process, a lot of unnecessary, unbalanced or unimportant data can mislead the algorithm and reduce its overall performance. The task of extracting features is used to identify those key features and to process the key symbology and use them in the following task of training the model. In this step, imbalances in the data will be eliminated, especially in the target attributes. All unnecessary attributes with low relevance to the final result will be filtered out and removed and the unbalanced data will be rebalanced.
3. **Train/Build the model**
   In this step, we need to make a judgment call to determine the data mining method based on the dataset and the expected results. The choice of data mining method will influence the choice of data mining algorithm. And before we proceed with data mining, the training and test sets will be partitioned. Based on the partitioning of the training and test datasets, we perform the selection and construction of models/algorithms.
   After extracting the features, all the features can now be loaded into the model to see how the model performs. This process usually involves multiple data mining models and the best-performing one is selected as the data model for parameter tuning. The parameter tuning process is used to make the model more suitable for the features selected in the previous steps (since trends will be different for each dataset, parameter tuning is necessary to prevent the algorithm from performing below expectations). The features will then be trained on the selected model. Data mining algorithms will be evaluated and selected based on business objectives and data mining goals. The data mining models/algorithms will be run and patterns identified.
4. **Evaluate the Model**
   After training the model, the model can now be evaluated. In this step, metrics will be used to identify the performance of the model. The results of data mining, models, and patterns are evaluated, and the results are analyzed for validity and correctness. The most convenient and easy update to try to improve the accuracy of the model can occur in the training/testing data split. As the number of training sets increases, so does the efficiency of the algorithm. The method includes accuracy measures, time consumption measures, etc. Once the performance of the model is unsatisfactory, the process can jump back to any of the previously mentioned processes and modify the settings to change the performance of the model.

8.1 Data cleaning

The original files of the dataset had gaps and unfinished areas and these errors were often missed or deliberately ignored by those who did not record them in the dataset due to privacy concerns or other reasons of consideration. Two types of errors were found in the processed data, missing values and data errors. These values will be removed or replaced in preparation for further use of the data.

Clean missing data:

A total of two attributes were found to contain missing values, BMI with an integrity value of 96.631% and smoking status with 69.373%. These missing values can reduce the efficiency of the module's execution. Therefore, they need to be removed or averaged to fill them in before being applied to any model. This step eliminates the problem when applying data to certain algorithms and makes the algorithm less skewed in the wrong direction.

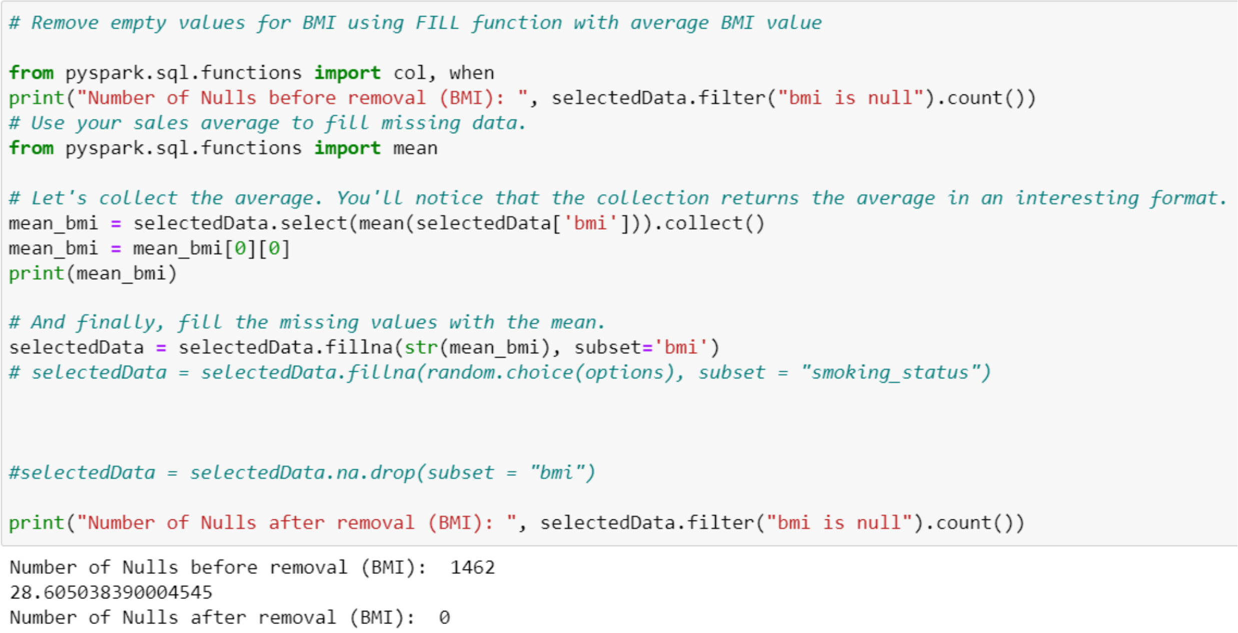
A null or empty value is not very compatible with some algorithms. It is usually represented in many different forms, such as empty cells as well as N/A markers. In this dataset, both smoking status and BMI contain these missing values. There are three ways to remove them: directly from the record or populated according to some pattern (usually copied from the last populated instance) or populated with a fixed number derived from the averaging algorithm.

For the missing BMI and smoking status values in this project, we took 2 approaches to deal with the missing BMI and smoking status values separately.

For the smoking status values, the best approach is to simply delete these instances as there are three smoking statuses and the missing values take up 31% of the existing dataset, which is too large a proportion, and randomly populating these values or replicating them from later instances is likely to result in the dataset losing its original characteristics, resulting in much less accurate predictions.

The BMI values could instead be populated using a fixed number derived from the averaging algorithm. As the BMI has roughly 4% missing values, using the mean padding would not affect the overall dataset too much and would preserve as many instances as possible to mitigate the problem of shortage of stroke cases.

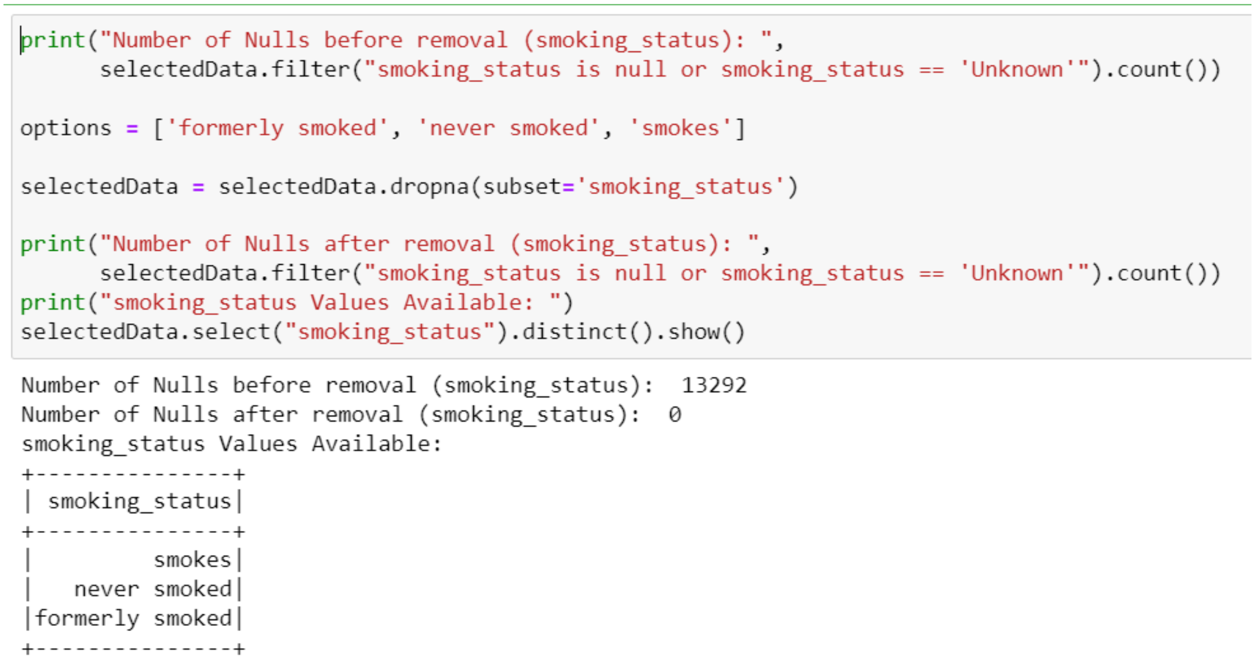
Therefore, we decided to remove those instances with missing values in the smoking status and use the mean to populate those instances with missing values in the BMI as below in Figure 1 and Figure 2.



```
# Remove empty values for BMI using FILL function with average BMI value

from pyspark.sql.functions import col, when
print("Number of Nulls before removal (BMI): ", selectedData.filter("bmi is null").count())
# Use your sales average to fill missing data.
from pyspark.sql.functions import mean

# Let's collect the average. You'll notice that the collection returns the average in an interesting format.
mean_bmi = selectedData.select(mean(selectedData['bmi'])).collect()
mean_bmi = mean_bmi[0][0]
print(mean_bmi)

# And finally, fill the missing values with the mean.
selectedData = selectedData.fillna(str(mean_bmi), subset='bmi')
# selectedData = selectedData.fillna(random.choice(options), subset = "smoking_status")



#selectedData = selectedData.na.drop(subset = "bmi")

print("Number of Nulls after removal (BMI): ", selectedData.filter("bmi is null").count())
```

```
Number of Nulls before removal (BMI):  1462
28.605038390004545
Number of Nulls after removal (BMI):  0
```

Figure 1. Cleaning missing values in BMI

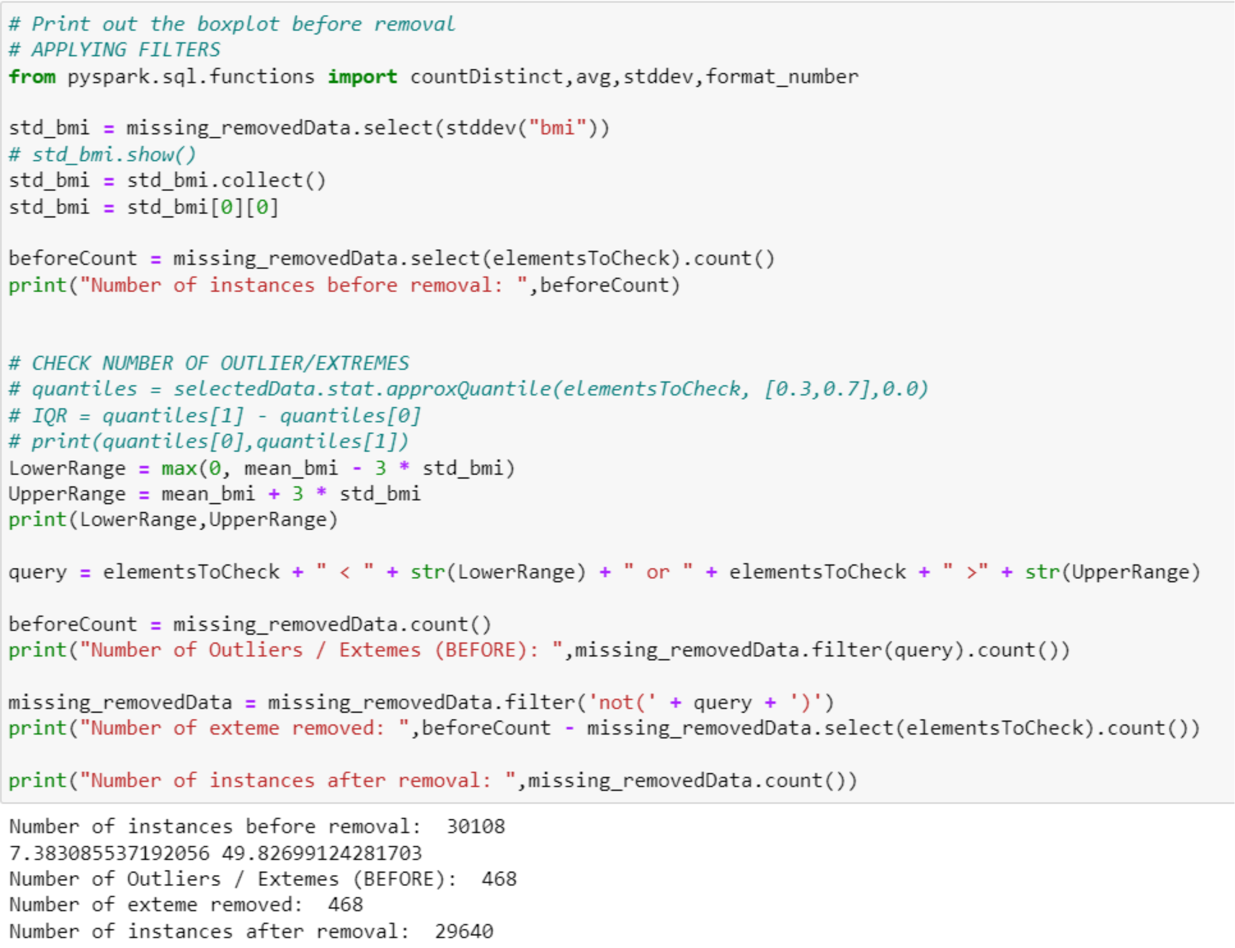


```
print("Number of Nulls before removal (smoking_status): ",
      selectedData.filter("smoking_status is null or smoking_status == 'Unknown'").count())

options = ['formerly smoked', 'never smoked', 'smokes']

selectedData = selectedData.dropna(subset='smoking_status')

print("Number of Nulls after removal (smoking_status): ",
      selectedData.filter("smoking_status is null or smoking_status == 'Unknown'").count())
print("smoking_status Values Available: ")
selectedData.select("smoking_status").distinct().show()
```

```
Number of Nulls before removal (smoking_status):  13292
Number of Nulls after removal (smoking_status):  0
smoking_status Values Available:
+---------------+
| smoking_status|
+---------------+
|         smokes|
|   never smoked|
|formerly smoked|
+---------------+
```

Figure 2. Cleaning missing values in smoking_status

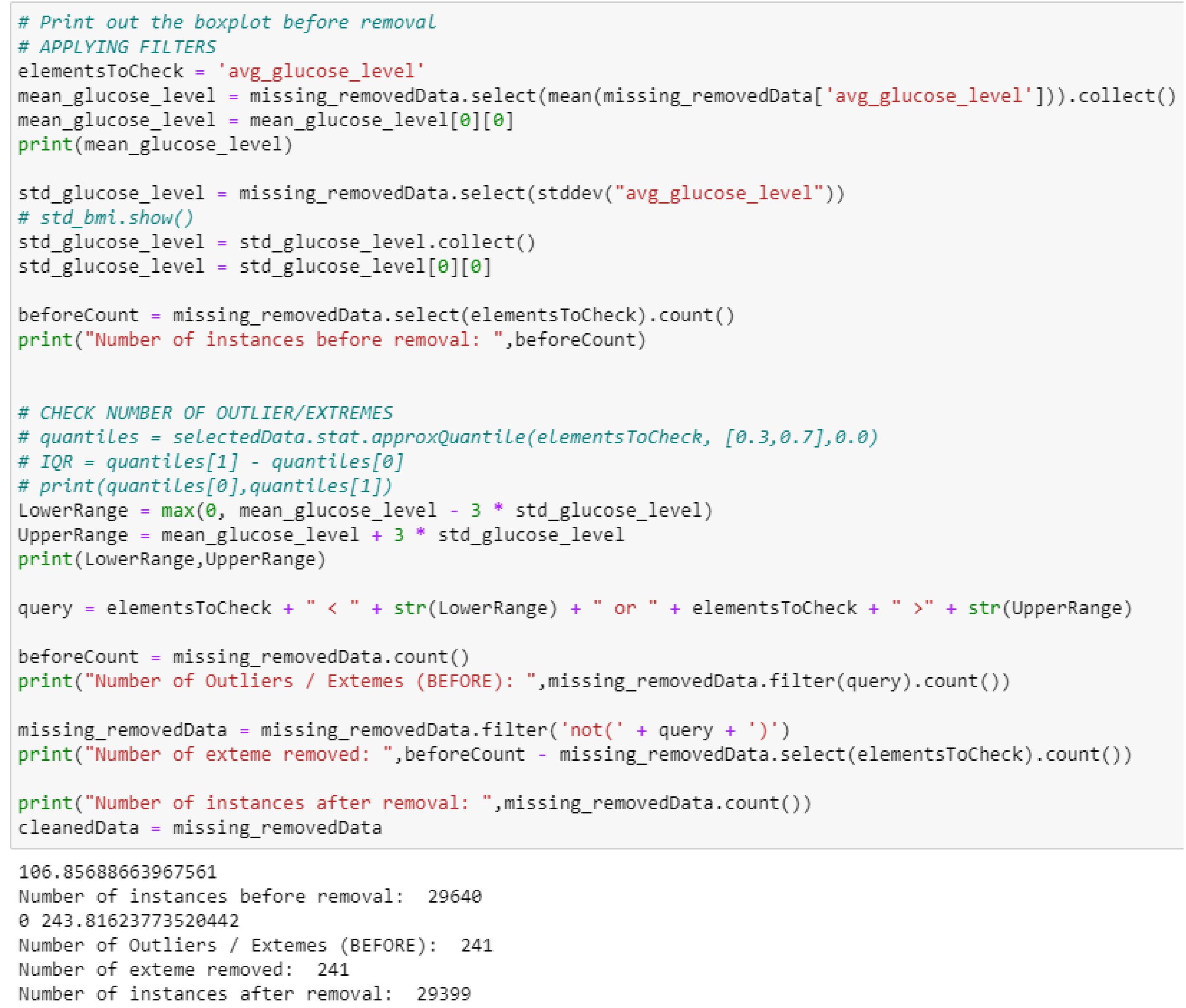
Clean data error:
The only data errors contained in the file are extreme/outlier values, which can affect the module's predictions as the module may be biased towards outliers as it tries to cover many different possible values. The method used to remove the outliers is to calculate the deviation of the value with normal distribution in BMI and average glucose level. We have identified that the values are out of 3 times the standard deviation as outliers. then, we select all values in the range of normal distribution and filter out those outlier values in age, BMI, and average glucose(rows). These extreme values are removed using forced deletion as below in Figure 3 and Figure 4. This method will attempt to drag the values from the extreme range to a reasonable range within the range.



```python
# Print out the boxplot before removal
# APPLYING FILTERS
from pyspark.sql.functions import countDistinct,avg,stddev,format_number

std_bmi = missing_removedData.select(stddev("bmi"))
# std_bmi.show()
std_bmi = std_bmi.collect()
std_bmi = std_bmi[0][0]

beforeCount = missing_removedData.select(elementsToCheck).count()
print("Number of instances before removal: ",beforeCount)


# CHECK NUMBER OF OUTLIER/EXTREMES
# quantiles = selectedData.stat.approxQuantile(elementsToCheck, [0.3,0.7],0.0)
# IQR = quantiles[1] - quantiles[0]
# print(quantiles[0],quantiles[1])
LowerRange = max(0, mean_bmi - 3 * std_bmi)
UpperRange = mean_bmi + 3 * std_bmi
print(LowerRange,UpperRange)

query = elementsToCheck + " < " + str(LowerRange) + " or " + elementsToCheck + " >" + str(UpperRange)

beforeCount = missing_removedData.count()
print("Number of Outliers / Extemes (BEFORE): ",missing_removedData.filter(query).count())

missing_removedData = missing_removedData.filter('not(' + query + ')')
print("Number of exteme removed: ",beforeCount - missing_removedData.select(elementsToCheck).count())

print("Number of instances after removal: ",missing_removedData.count())
```

```
Number of instances before removal:  30108
7.383085537192056 49.82699124281703
Number of Outliers / Extemes (BEFORE):  468
Number of exteme removed:  468
Number of instances after removal:  29640
```

Figure 3. Cleaning BMI outliers process



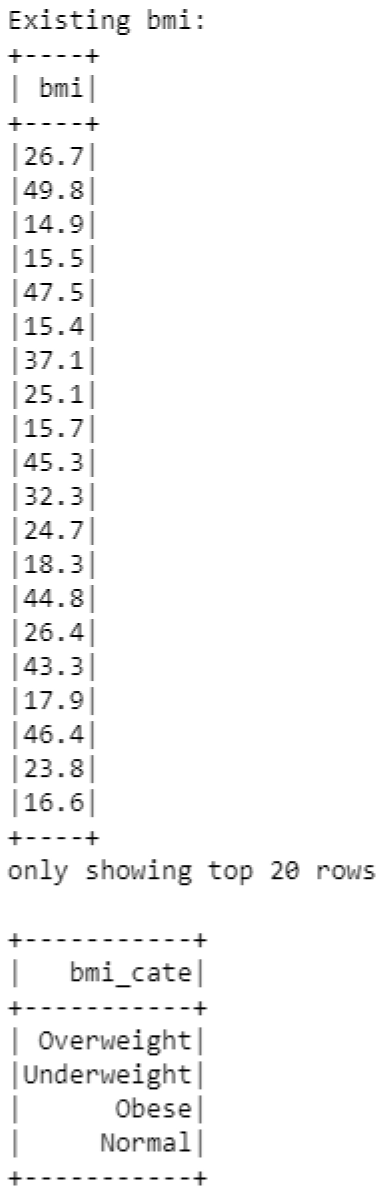
```
# Print out the boxplot before removal
# APPLYING FILTERS
elementsToCheck = 'avg_glucose_level'
mean_glucose_level = missing_removedData.select(mean(missing_removedData['avg_glucose_level'])).collect()
mean_glucose_level = mean_glucose_level[0][0]
print(mean_glucose_level)

std_glucose_level = missing_removedData.select(stddev("avg_glucose_level"))
# std_bmi.show()
std_glucose_level = std_glucose_level.collect()
std_glucose_level = std_glucose_level[0][0]

beforeCount = missing_removedData.select(elementsToCheck).count()
print("Number of instances before removal: ",beforeCount)


# CHECK NUMBER OF OUTLIER/EXTREMES
# quantiles = selectedData.stat.approxQuantile(elementsToCheck, [0.3,0.7],0.0)
# IQR = quantiles[1] - quantiles[0]
# print(quantiles[0],quantiles[1])
LowerRange = max(0, mean_glucose_level - 3 * std_glucose_level)
UpperRange = mean_glucose_level + 3 * std_glucose_level
print(LowerRange,UpperRange)

query = elementsToCheck + " < " + str(LowerRange) + " or " + elementsToCheck + " >" + str(UpperRange)

beforeCount = missing_removedData.count()
print("Number of Outliers / Extemes (BEFORE): ",missing_removedData.filter(query).count())

missing_removedData = missing_removedData.filter('not(' + query + ')')
print("Number of exteme removed: ",beforeCount - missing_removedData.select(elementsToCheck).count())

print("Number of instances after removal: ",missing_removedData.count())
cleanedData = missing_removedData
```

```
106.85688663967561
Number of instances before removal:  29640
0 243.81623773520442
Number of Outliers / Extemes (BEFORE):  241
Number of exteme removed:  241
Number of instances after removal:  29399
```

Figure 4. Cleaning average glucose level outliers process

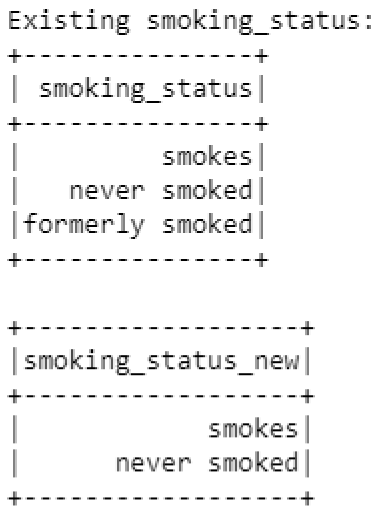
In addition, further steps such as feature creation and data formatting are carried out to make the data more suitable for the data mining process.

The feature creation process follows the CDC control guidelines for creating BMI categories as it is more useful than the actual numbers. Smoking status and job types were also recreated as there were duplicate values in these attributes. Figure 5 - 7 shows this process.



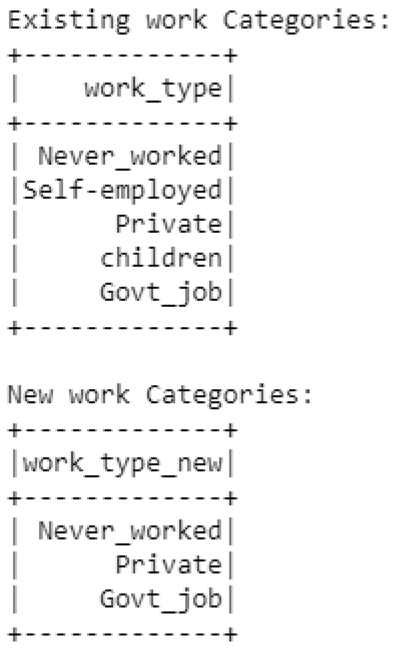
```
Existing bmi:
+----+
| bmi|
+----+
|26.7|
|49.8|
|14.9|
|15.5|
|47.5|
|15.4|
|37.1|
|25.1|
|15.7|
|45.3|
|32.3|
|24.7|
|18.3|
|44.8|
|26.4|
|43.3|
|17.9|
|46.4|
|23.8|
|16.6|
+----+
only showing top 20 rows

+-----------+
|   bmi_cate|
+-----------+
| Overweight|
|Underweight|
|      Obese|
|     Normal|
+-----------+
```

Figure 5. New BMI groups



```
Existing smoking_status:
+---------------+
| smoking_status|
+---------------+
|         smokes|
|   never smoked|
|formerly smoked|
+---------------+

+------------------+
|smoking_status_new|
+------------------+
|            smokes|
|      never smoked|
+------------------+
```

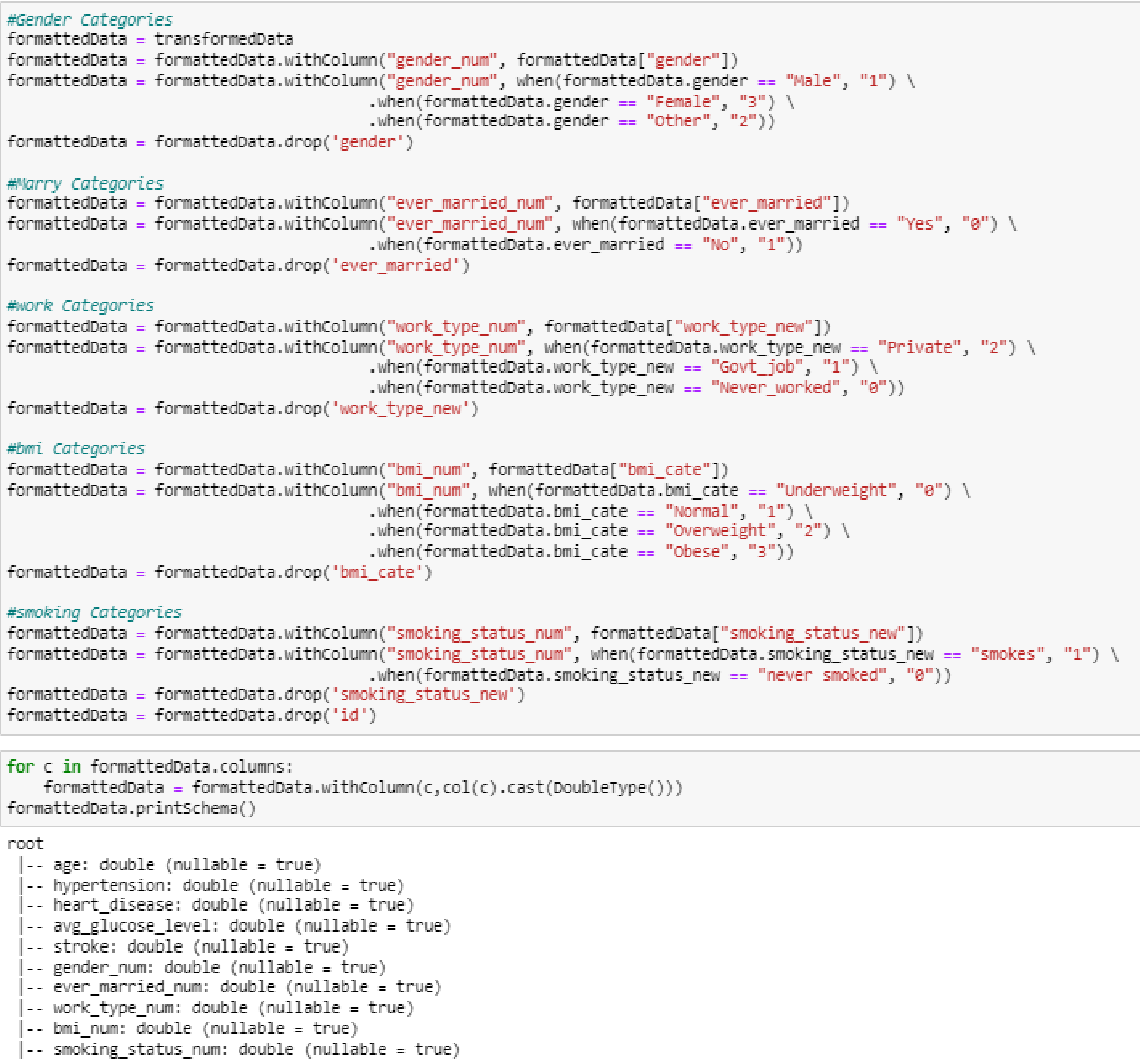
Figure 6. New Smoking status after set a flag



```
Existing work Categories:
+-------------+
|    work_type|
+-------------+
| Never_worked|
|Self-employed|
|      Private|
|     children|
|     Govt_job|
+-------------+

New work Categories:
+-------------+
|work_type_new|
+-------------+
| Never_worked|
|      Private|
|     Govt_job|
+-------------+
```

Figure 7. New Work categories

The data formatting was done in this dataset, which covers the data labels to the corresponding numbers, as most algorithms prefer to accept the results as numbers rather than actual strings. Figure 8 shows the result of its implementation.



```
#Gender Categories
formattedData = transformedData
formattedData = formattedData.withColumn("gender_num", formattedData["gender"])
formattedData = formattedData.withColumn("gender_num", when(formattedData.gender == "Male", "1") \
                                        .when(formattedData.gender == "Female", "3") \
                                        .when(formattedData.gender == "Other", "2"))
formattedData = formattedData.drop('gender')

#Marry Categories
formattedData = formattedData.withColumn("ever_married_num", formattedData["ever_married"])
formattedData = formattedData.withColumn("ever_married_num", when(formattedData.ever_married == "Yes", "0") \
                                        .when(formattedData.ever_married == "No", "1"))
formattedData = formattedData.drop('ever_married')

#work Categories
formattedData = formattedData.withColumn("work_type_num", formattedData["work_type_new"])
formattedData = formattedData.withColumn("work_type_num", when(formattedData.work_type_new == "Private", "2") \
                                        .when(formattedData.work_type_new == "Govt_job", "1") \
                                        .when(formattedData.work_type_new == "Never_worked", "0"))
formattedData = formattedData.drop('work_type_new')

#bmi Categories
formattedData = formattedData.withColumn("bmi_num", formattedData["bmi_cate"])
formattedData = formattedData.withColumn("bmi_num", when(formattedData.bmi_cate == "Underweight", "0") \
                                        .when(formattedData.bmi_cate == "Normal", "1") \
                                        .when(formattedData.bmi_cate == "Overweight", "2") \
                                        .when(formattedData.bmi_cate == "Obese", "3"))
formattedData = formattedData.drop('bmi_cate')

#smoking Categories
formattedData = formattedData.withColumn("smoking_status_num", formattedData["smoking_status_new"])
formattedData = formattedData.withColumn("smoking_status_num", when(formattedData.smoking_status_new == "smokes", "1") \
                                        .when(formattedData.smoking_status_new == "never smoked", "0"))
formattedData = formattedData.drop('smoking_status_new')
formattedData = formattedData.drop('id')
```

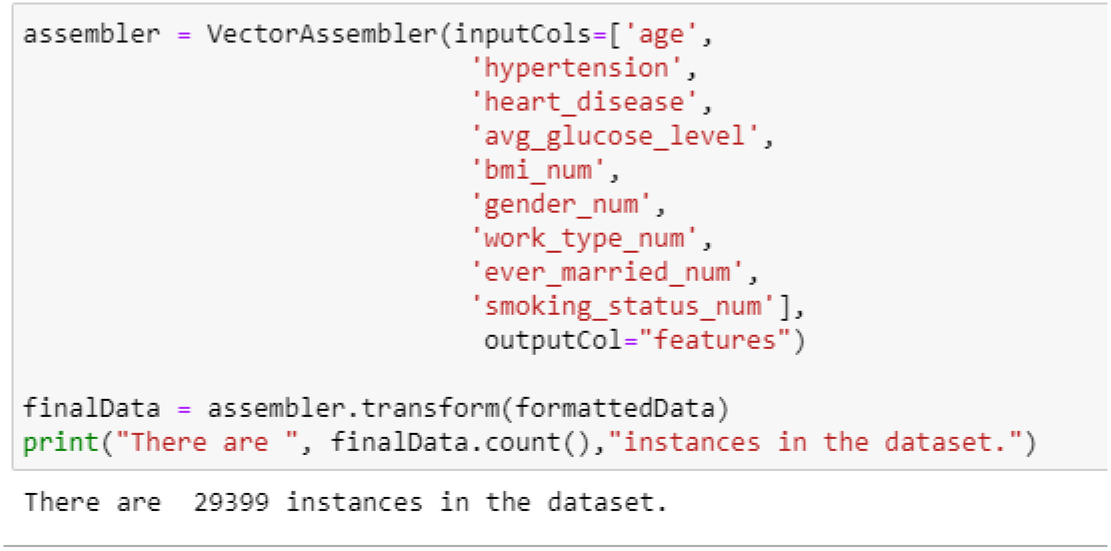
```
for c in formattedData.columns:
    formattedData = formattedData.withColumn(c,col(c).cast(DoubleType()))
formattedData.printSchema()
```

```
root
 |-- age: double (nullable = true)
 |-- hypertension: double (nullable = true)
 |-- heart_disease: double (nullable = true)
 |-- avg_glucose_level: double (nullable = true)
 |-- stroke: double (nullable = true)
 |-- gender_num: double (nullable = true)
 |-- ever_married_num: double (nullable = true)
 |-- work_type_num: double (nullable = true)
 |-- bmi_num: double (nullable = true)
 |-- smoking_status_num: double (nullable = true)
```

Figure 8. Python labeling process

8.2 Data Selection

Following the above process, the dataset is now ready for feature selection and feature projection. The purpose of feature selection is to use the algorithm to filter out unimportant or unnecessary attributes that may affect the performance of the model. This process ultimately selected nine attributes to perform the data mining task, while the other attributes were removed as they had less impact on the trip. Figure 9 shows the process of feature selection.



```
assembler = VectorAssembler(inputCols=['age',
                             'hypertension',
                             'heart_disease',
                             'avg_glucose_level',
                             'bmi_num',
                             'gender_num',
                             'work_type_num',
                             'ever_married_num',
                             'smoking_status_num'],
                              outputCol="features")

finalData = assembler.transform(formattedData)
print("There are ", finalData.count(),"instances in the dataset.")
```

```
There are  29399 instances in the dataset.
```

Figure 9. dataset after deleting unneeded attributes

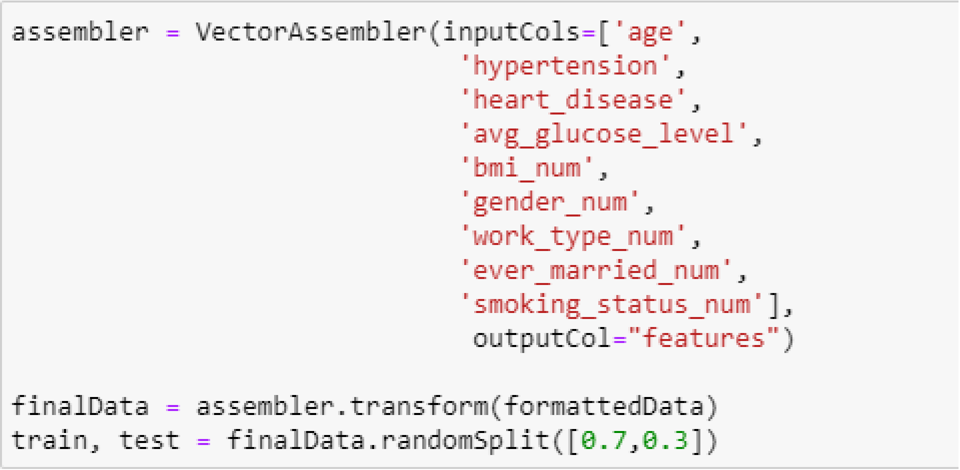
8.3 Train dataset / Test dataset split

Following the data cleaning process, and constructing new features and data selection, the data cleaning process succeeded in removing the existing extreme/outlier and missing values. However, the imbalanced label problem is not solved, and this problem needs to be handled before building a model and doing data mining.

After the data cleaning process and the construction of new features, the data cleaning process successfully eliminated the existing extreme/outliers and missing values. However, the imbalanced labelling problem was not solved, and this issue needs to be addressed before building the model and performing data mining.

Before balancing the stroke and non-stroke samples, we need to perform data partitioning first, because we only want to balance the data inside the training set so that the model can have enough balanced data for training and can predict the results of the real data more accurately. The data in the test set maintains the characteristics of the original real data set as much as possible, which can also better test the results of the model training. Therefore, we split the dataset to train and test datasets based on the 7:3 ratio.

The main idea of partitioning the dataset into validation sets is to prevent our model from overfitting, i.e., the model becomes very good at classifying samples in the training set but cannot generalize and accurately classify data that it has not seen before.

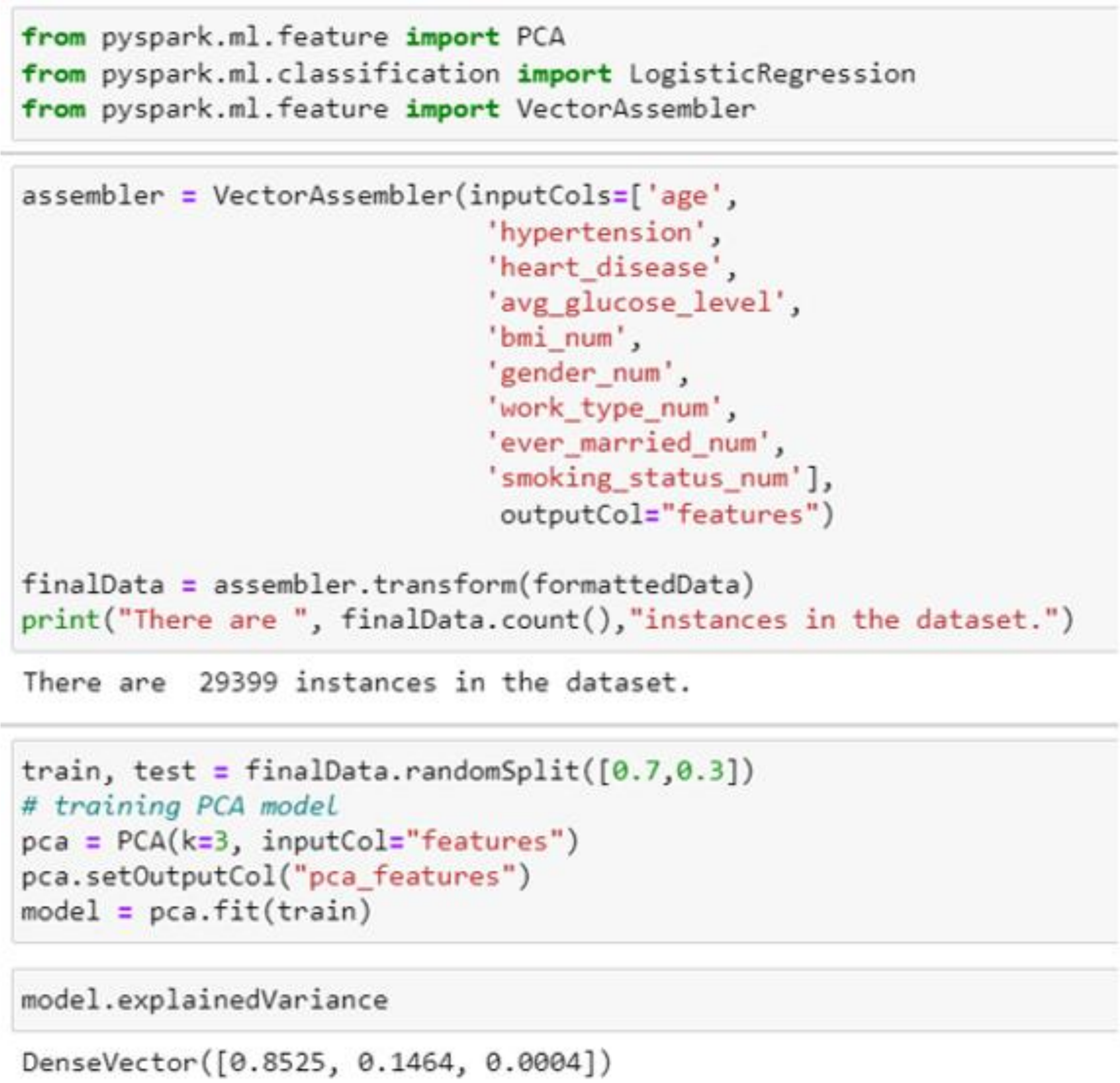


```
assembler = VectorAssembler(inputCols=['age',
                             'hypertension',
                             'heart_disease',
                             'avg_glucose_level',
                             'bmi_num',
                             'gender_num',
                             'work_type_num',
                             'ever_married_num',
                             'smoking_status_num'],
                              outputCol="features")

finalData = assembler.transform(formattedData)
train, test = finalData.randomSplit([0.7,0.3])
```

Figure 10. Train, test split to 7:3

8.4 Data reduction & Projection

In the data reduction process, among these nine attributes, there are also attributes that need to be evaluated for some of the smaller contributions to the final results. Most of the time, these attributes can mislead the algorithm and ultimately reduce the overall accuracy of the model. Therefore, some amount of data reduction/dimensionality reduction is also required to eliminate the number of these attributes, using a method called PCA. PCA is used for exploratory data analysis and building predictive models. It is typically used for dimensionality reduction, projecting each data point onto only the first few principal components to obtain low-dimensional data while preserving as much variation in the data as possible. We do the data reduction using PCA as shown in Figure 11. We project the 9 latitudes to a 3-dimensional data graph.



```
from pyspark.ml.feature import PCA
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import VectorAssembler
```

```
assembler = VectorAssembler(inputCols=['age',
                              'hypertension',
                              'heart_disease',
                              'avg_glucose_level',
                              'bmi_num',
                              'gender_num',
                              'work_type_num',
                              'ever_married_num',
                              'smoking_status_num'],
                              outputCol="features")

finalData = assembler.transform(formattedData)
print("There are ", finalData.count(),"instances in the dataset.")
```

```
There are  29399 instances in the dataset.
```

```
train, test = finalData.randomSplit([0.7,0.3])
# training PCA model
pca = PCA(k=3, inputCol="features")
pca.setOutputCol("pca_features")
model = pca.fit(train)
```

```
model.explainedVariance
```

```
DenseVector([0.8525, 0.1464, 0.0004])
```

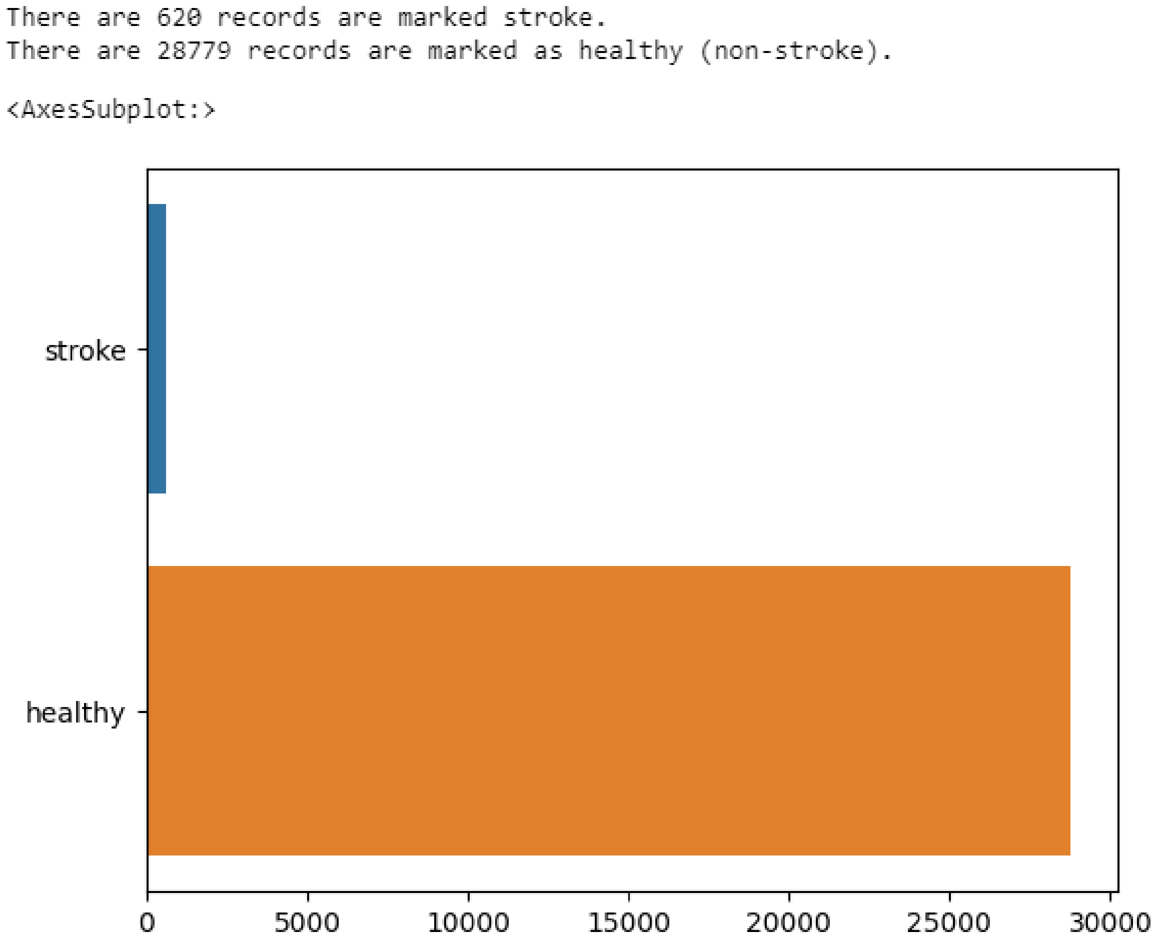
Figure 11. Use PCA to do data reduction by reducing latitudes

In the data projection process, we solve the problem of unbalanced labels in the training dataset, because we try to keep the test dataset the same as the raw dataset possible, and always invisible. Training only the training set equal to that of stroke patients and non-stroke patients allows the model to work better when real data is available.

Unbalanced labels, with too large a difference in the number of stroke patients and non-stroke patients, can lead to the prior probability of the data itself, which can have a bias towards negative samples of stroke. And according to Bayesian theory, the prior probability is the probability that can be obtained before the model experiment or sampling based on previous experience and analysis. To put it in layman's terms, there are now 620 samples labeled as stroke and 28,779 samples labeled as not stroke in the dataset as shown in Figure 12. Using this dataset to train the model, the model will most likely predict the patients who do not have a stroke accurately, while the patients with a high risk of stroke are predicted to have a high probability of not having a stroke. This model, however, has a high risk of not being able to accurately predict the likelihood of stroke for patients at an early stage in practical use, which is contrary to our goal of doing this project and modeling each other. To address the potential dangers of such label imbalances, we need to rebalance the data set between stroke and healthy (non-stroke) patients.

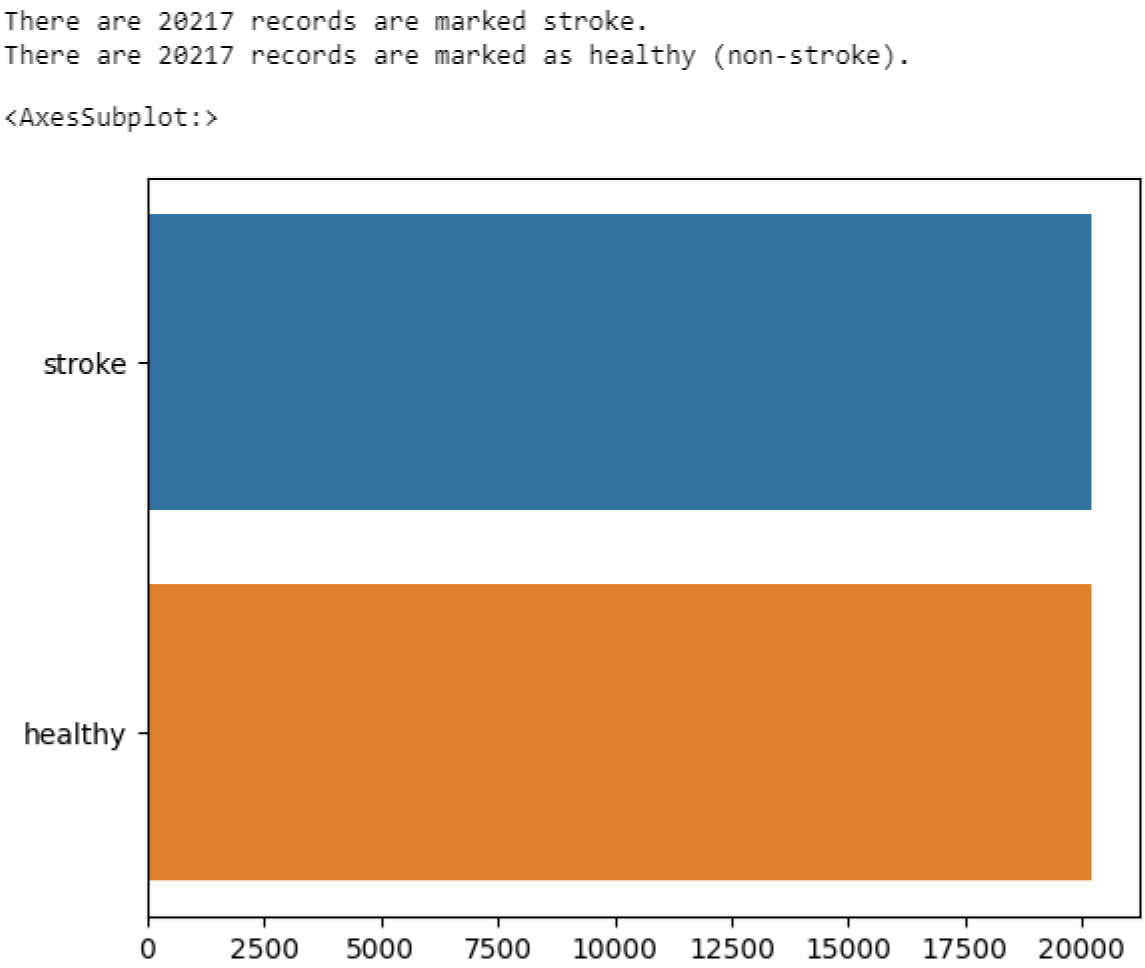
We use the algorithm of repeating to solve the imbalance label problem in the training dataset, and generate more data for stroke, which makes the stroke patients and non-stroke patients' ratio around 1: 1. We repeat stroke instances 50 times to make stroke patients and non-stroke patients' ratio around 1: 1, which solves the imbalance label problem. And we are integrating newly generated stroke instances into the dataset.

In this way, we have generated new data on stroke patients to balance the unevenness of stroke and non-stroke distribution and integrate new data records into the dataset. We can see the new stroke distribution in the dataset is balanced as shown in Figure 13.



```
There are 620 records are marked stroke.
There are 28779 records are marked as healthy (non-stroke).

<AxesSubplot:>
```

Figure 12. Stroke distribution in training dataset before re-balance



```
There are 20217 records are marked stroke.
There are 20217 records are marked as healthy (non-stroke).

<AxesSubplot:>
```

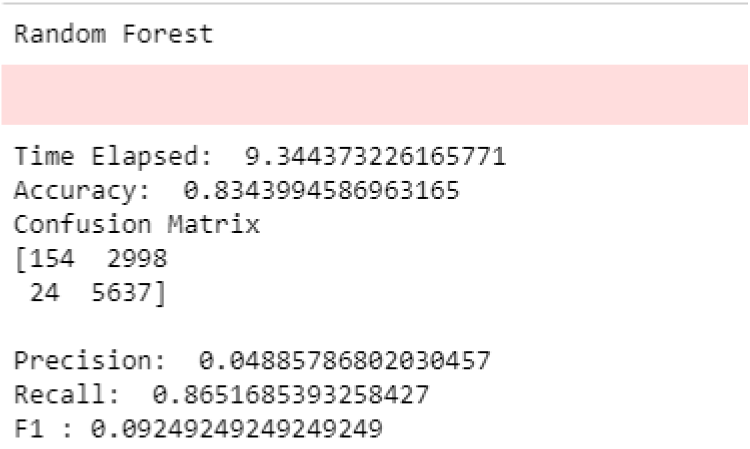
Figure 13. Stroke distribution in training dataset after re-balance

After this step, we get our final dataset ready for the data mining process.

9. Implementation

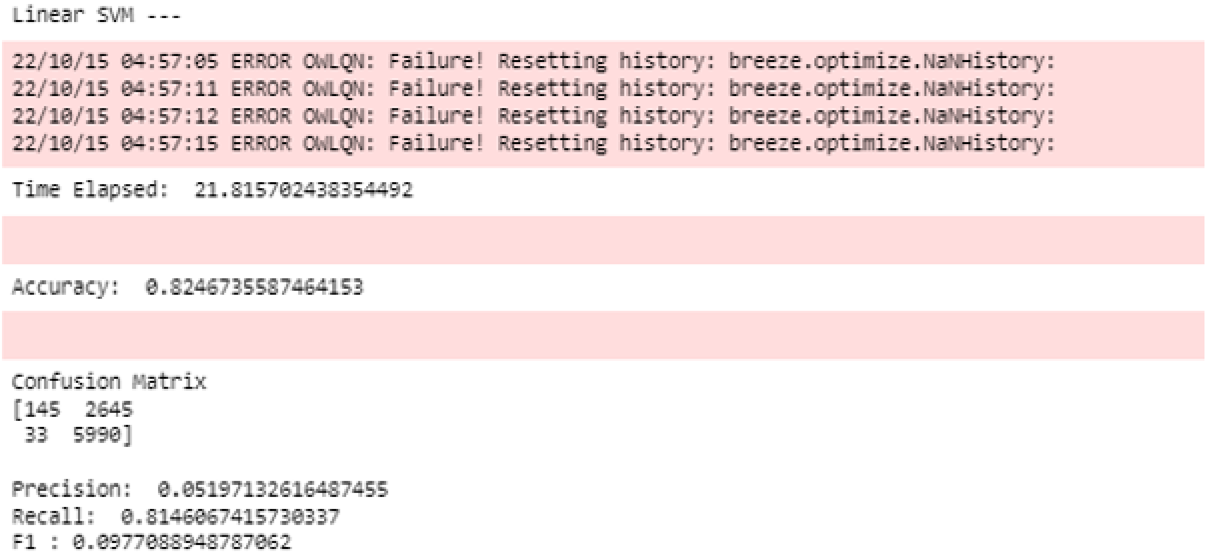
9.1 Choose data mining algorithms

The data mining algorithm for this research can be narrowed down to supervised learning, and it should use classification as it is used to distinguish items in a dataset into classes or groups. It helps to accurately predict the behavior of entities within the group, which aligns with the data mining methodology as it suits and follows the requirement of research objectives. To find the most suitable data mining algorithm, the modeling process is performed in three candidate algorithms, Random Forest, SVM, and Neural Networks.



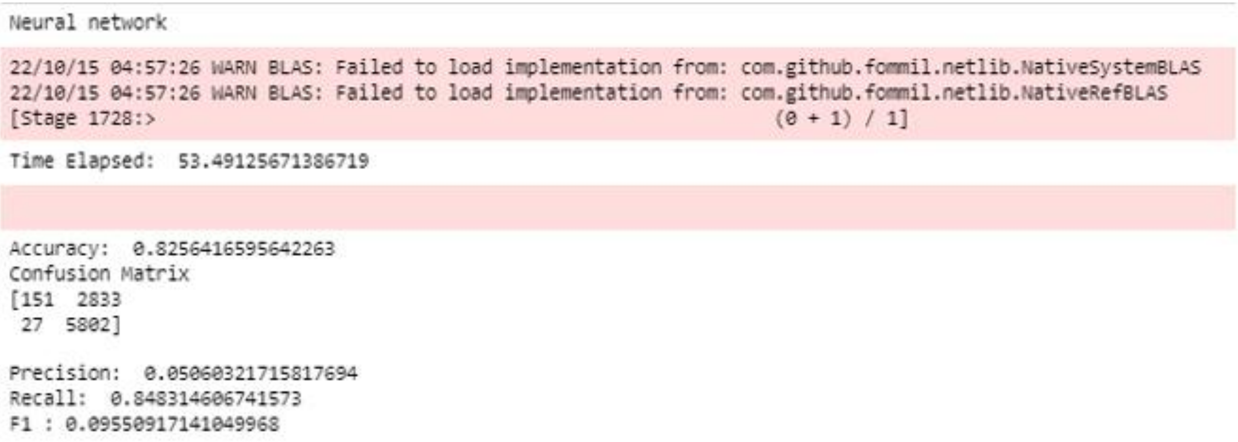
```
Random Forest

Time Elapsed:  9.344373226165771
Accuracy:  0.8343994586963165
Confusion Matrix
[154  2998
 24  5637]

Precision:  0.04885786802030457
Recall:  0.8651685393258427
F1 : 0.09249249249249249
```

Figure 14. Random forest model result



```
Linear SVM ---
22/10/15 04:57:05 ERROR OWLQN: Failure! Resetting history: breeze.optimize.NaNHistory:
22/10/15 04:57:11 ERROR OWLQN: Failure! Resetting history: breeze.optimize.NaNHistory:
22/10/15 04:57:12 ERROR OWLQN: Failure! Resetting history: breeze.optimize.NaNHistory:
22/10/15 04:57:15 ERROR OWLQN: Failure! Resetting history: breeze.optimize.NaNHistory:
Time Elapsed:  21.815702438354492

Accuracy:  0.8246735587464153

Confusion Matrix
[145  2645
 33  5990]

Precision:  0.05197132616487455
Recall:  0.8146067415730337
F1 : 0.0977088948787062
```

Figure 15. SVM model process and result



```
Neural network
22/10/15 04:57:26 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
22/10/15 04:57:26 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
[Stage 1728:>                                                   (0 + 1) / 1]
Time Elapsed:  53.49125671386719

Accuracy:  0.8256416595642263
Confusion Matrix
[151  2833
 27  5802]

Precision:  0.05060321715817694
Recall:  0.848314606741573
F1 : 0.09550917141049968
```

Figure 16. Neural network model process and result

Random Forest and Neural Network models perform well and have similar accuracy scores and recall scores. We would choose the Neural Network model as our decision for this project, as the deep learning model is more suitable when we do predictions for medical cases because it is more stable than other models.

We can improve the prediction accuracy score and recall score by adjusting the parameters of this algorithm in the following section.

9.2 Parameter Tunning

Most machine learning and deep learning algorithms have some parameters that can be tuned, called hyperparameters. Before training the model, we need to set the hyperparameters. Hyperparameters are crucial for building robust and accurate models. By helping us find a balance between bias and variance, they prevent the model from being overfitted or under fitted.

The algorithm used to train the model is now being set to the Neural Network, in order to give the best and most accurate result, the parameter inputs are needed to be turned several times and uses the different combinations to evaluate the best combination which can be applied in the final data mining processes. Gradient Descent is one of the most commonly used methods for solving model parameters of machine learning algorithms, i.e. unconstrained optimization problems, and another commonly used method is least squares. When solving for the minimum of the loss function, the gradient descent method can be used to solve the problem iteratively, step by step, to obtain the minimized loss function and model parameters. Conversely, if we need to solve for the maximum value of the loss function, it is then necessary to iterate with gradient ascent.

In this research, we use stochastic gradient descent in this case, the maximum number of runs of gradient descent, step size, and the structure of the neural network will be evaluated and tested to see the influence of each.

The stochastic gradient descent algorithm updates the model parameters one sample at a time, so each learning is very fast and online updates can be performed. The biggest disadvantage is that each update may not go in the right direction, resulting in sharp fluctuations (perturbations) in the objective function, but on the other hand, the fluctuations caused by stochastic gradient descent have the advantage that for basin-like regions (i.e. many local minima) then the fluctuations may lead to a jump from the current local minima to another better local minima, eventually converging at a better local minima, or even a global minima.

Due to the fluctuations, the number of iterations of learning will increase, i.e. convergence becomes slower.

The maximum number of runs of gradient descent：

The maximum number of runs of gradient descent stops when the maximum number of runs is reached, even if there is no convergence. We can do this by limiting the maximum number of runs of gradient descent so that the effect of a certain excessive error is limited, and this avoids excessive changes in the direction of the iteration.

step size：

the step size of each parameter updates for gradient descent. To minimize the fluctuations, we can adjust the step size. A larger step size avoids oscillations to some extent and allows walking over the local minima toward the larger extreme value points.

The structure of the neural network:

The structure of the neural network, how many layers there are, and how many neurons are in each layer. The length of the list means how many layers the neural network has, and the value in the list means how many neurons are in the layer. The structure of the neural network affects the capacity, the more complex the neural network, the more complex the function that can be fitted.

The parameters of the max number of runs, step size, and the structure of the neural network is being chosen and set up a test combination to evaluate both the accuracy, recall score, and time efficiency of the algorithm. Table 1 is showing the result of that.

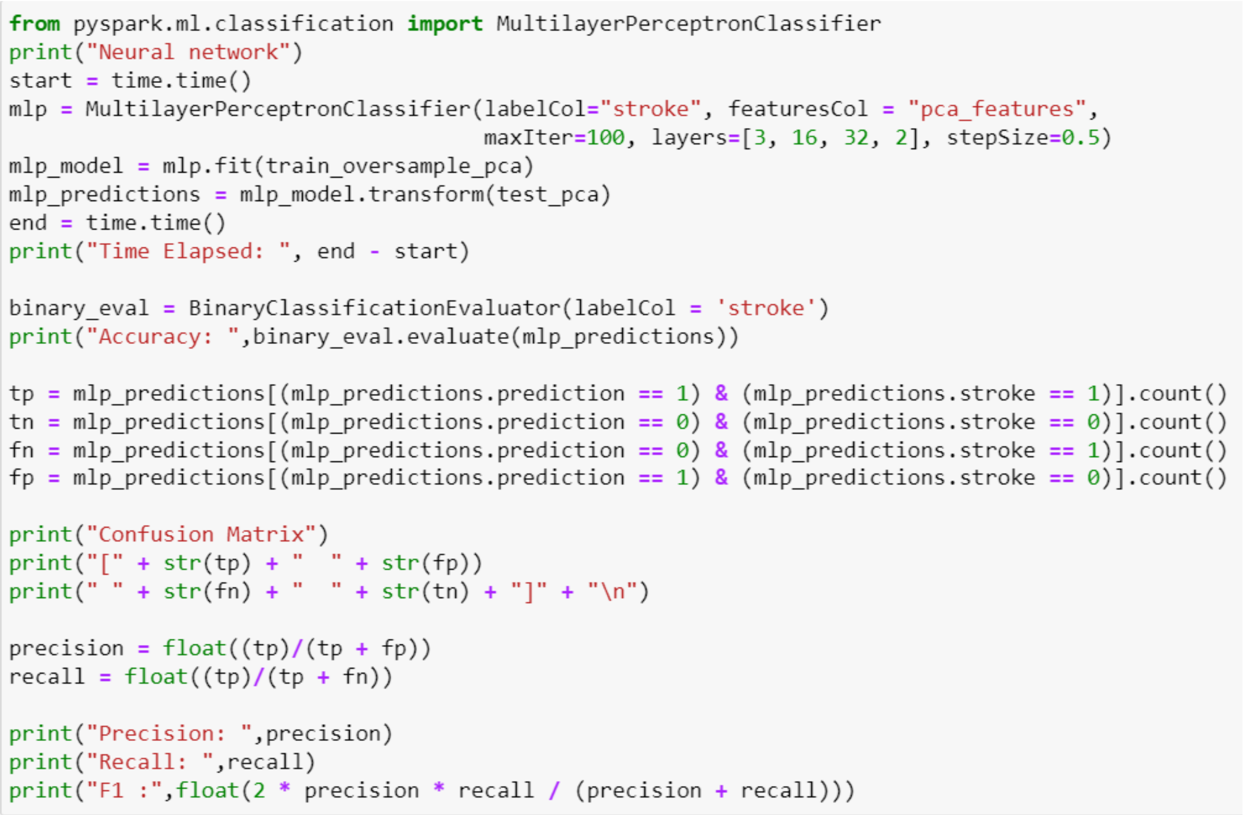
Max number of runs	**Step size**	**Structure of the neural network**	**Accuracy%**	**Recall%**
300	0.5	[3, 8, 2]	64.34	36.32
300	0.5	[3, 32, 2]	80.79	81.05
300	0.5	[3, 8, 32, 2]	81.5	83.68
100	0.5	[3, 16, 32, 8, 2]	81.57	82.63
100	0.5	[3, 6, 62, 2]	83.34	86.55

Table 1. Neural network model parameter pruning

The best parameter of the model is highlighted and chosen for further data mining steps.

9.3 Final model with defined parameters

After applying the algorithm selection as well as the parameter tuning process, the best combination of the parameter is identified, Figure 17 is showing the final data mining model that was used to discover the patterns.



```
from pyspark.ml.classification import MultilayerPerceptronClassifier
print("Neural network")
start = time.time()
mlp = MultilayerPerceptronClassifier(labelCol="stroke", featuresCol = "pca_features",
                                     maxIter=100, layers=[3, 16, 32, 2], stepSize=0.5)
mlp_model = mlp.fit(train_oversample_pca)
mlp_predictions = mlp_model.transform(test_pca)
end = time.time()
print("Time Elapsed: ", end - start)

binary_eval = BinaryClassificationEvaluator(labelCol = 'stroke')
print("Accuracy: ",binary_eval.evaluate(mlp_predictions))

tp = mlp_predictions[(mlp_predictions.prediction == 1) & (mlp_predictions.stroke == 1)].count()
tn = mlp_predictions[(mlp_predictions.prediction == 0) & (mlp_predictions.stroke == 0)].count()
fn = mlp_predictions[(mlp_predictions.prediction == 0) & (mlp_predictions.stroke == 1)].count()
fp = mlp_predictions[(mlp_predictions.prediction == 1) & (mlp_predictions.stroke == 0)].count()

print("Confusion Matrix")
print("[" + str(tp) + "  " + str(fp))
print(" " + str(fn) + "  " + str(tn) + "]" + "\n")

precision = float((tp)/(tp + fp))
recall = float((tp)/(tp + fn))

print("Precision: ",precision)
print("Recall: ",recall)
print("F1 :",float(2 * precision * recall / (precision + recall)))
```

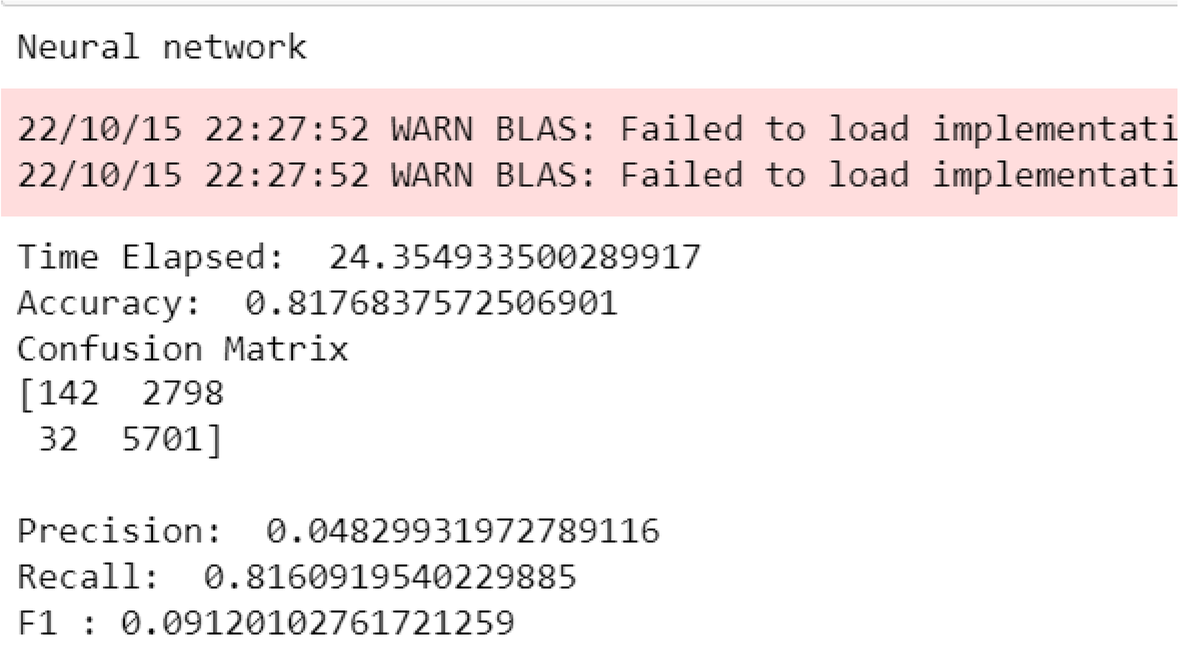
Figure 17. Selected Neural Network model with predefined parameters

10. Interpretation of the patterns and results

After fitting the data mining model and using the test data to evaluate the model, the following patterns and results are produced. The input attributes will be gender, age, hypertension, heart disease, ever married, average blood glucose level, new BMI, new smoking status, and new job type, and the target is set to be stroke. The total amount of data used was 49,227 instances, which were split into training/testing datasets. The split ratio was set to 7:3. Training dataset after using the rebalance method has 40,434 records, and the testing dataset has 8,793 records.

10.1 Interpretation of result

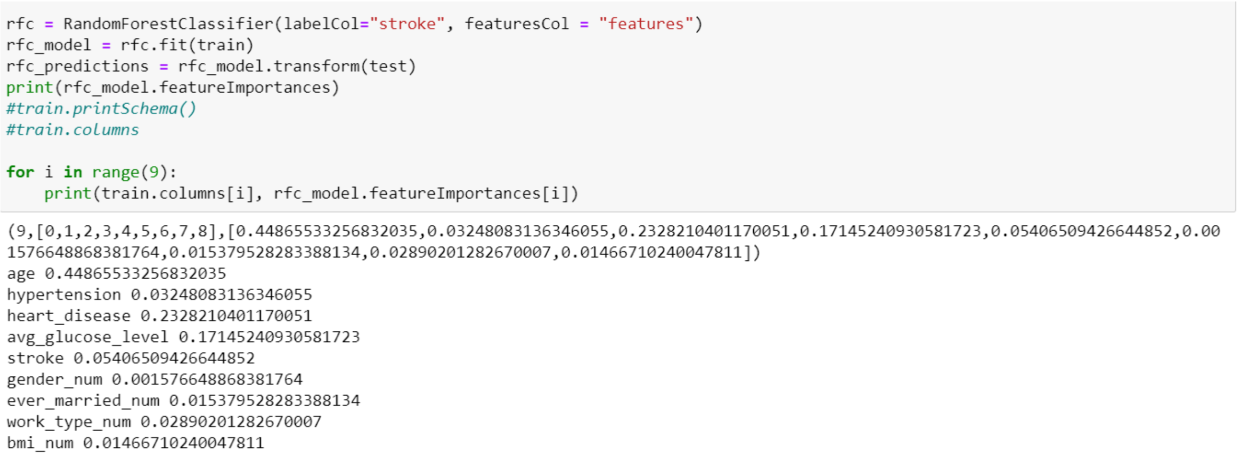
Figure 18 is showing the execution result of the model. This image includes the accuracy, confusion matrix, summary report, and time measurements for the model.



```
Neural network

22/10/15 22:27:52 WARN BLAS: Failed to load implementati
22/10/15 22:27:52 WARN BLAS: Failed to load implementati

Time Elapsed:  24.354933500289917
Accuracy:  0.8176837572506901
Confusion Matrix
[142  2798
 32  5701]

Precision:  0.04829931972789116
Recall:  0.8160919540229885
F1 : 0.09120102761721259
```

Figure 18. Selected Neural Network model running result

Figure 19 shows the importance of each attribute in the model.

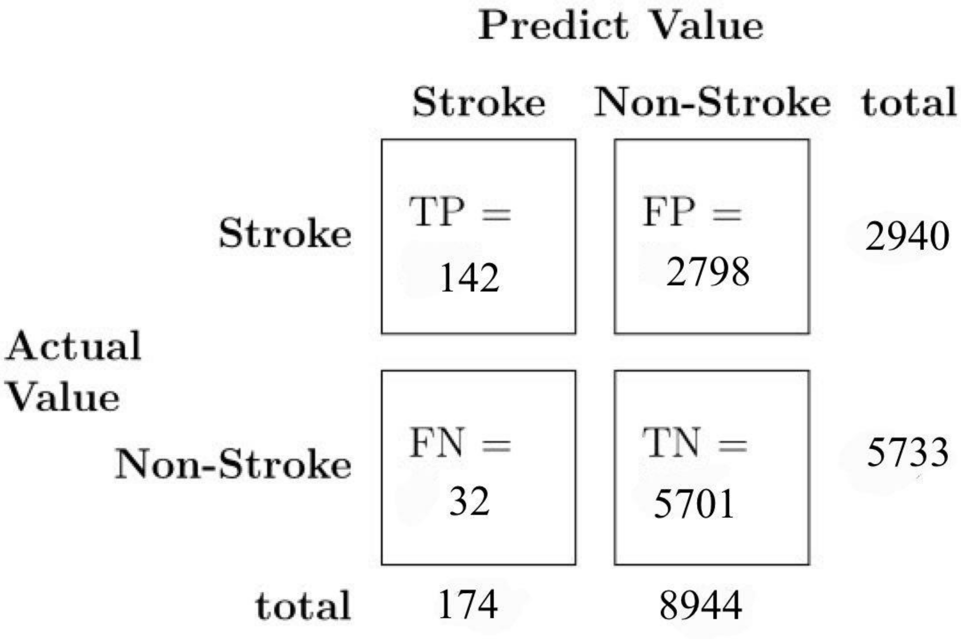


```
rfc = RandomForestClassifier(labelCol="stroke", featuresCol = "features")
rfc_model = rfc.fit(train)
rfc_predictions = rfc_model.transform(test)
print(rfc_model.featureImportances)
#train.printSchema()
#train.columns

for i in range(9):
    print(train.columns[i], rfc_model.featureImportances[i])
```

```
(9,[0,1,2,3,4,5,6,7,8],[0.44865533256832035,0.03248083136346055,0.2328210401170051,0.17145240930581723,0.05406509426644852,0.00
1576648868381764,0.015379528283388134,0.02890201282670007,0.01466710240047811])
age 0.44865533256832035
hypertension 0.03248083136346055
heart_disease 0.2328210401170051
avg_glucose_level 0.17145240930581723
stroke 0.05406509426644852
gender_num 0.001576648868381764
ever_married_num 0.015379528283388134
work_type_num 0.02890201282670007
bmi_num 0.01466710240047811
```

Figure 19. Feature Importance

From the result, the algorithm accuracy was 81.77%, and the recall score is 81.6%, which indicated a good performance of the used Neural Network algorithm for stroke prediction in the project. The recall score means the accuracy (positive predictive value) of classifying the data instances was very good, meaning that the model was sensitive enough to predict the sample of stroke, the probability of predicting the sample of stroke as the correct stroke was 81.6%.



		Predict Value		
		Stroke	Non-Stroke	total
Actual Value	Stroke	TP = 142	FP = 2798	2940
	Non-Stroke	FN = 32	TN = 5701	5733
	total	174	8944	

Figure 20. Model running result confusion matrix

The overall performance of the model was quite satisfactory, in dealing with an unbalanced database, we had to perform a rebalancing, meaning that noise was added to the dataset. However, the model still achieves an average accuracy of 82%. At the same time, the recall rate (calculated by the confusion matrix above Figure 20) reached 82%, which is quite satisfactory. Figure 21 shows the ROC curve, indicating that the model performed well. The model is able to perform data prediction for both stroke and non-stroke, and the model itself is usable and valid.

However, overall, the accuracy and recall of this model are still not high enough to accurately achieve most predictions, but it is far from being ready for use for medical purposes, and more data needs to be collected for input and training, and more data is needed for testing.

However, this model needs further improvement and involves more predictors and adjusting more parameters to make the model more accurate and usable for medical purposes in the future. From a data perspective, the model still sees relatively little information about the candidates for stroke. Even with the test dataset for detection, overfitting problems may still occur, resulting in an insensitive model.

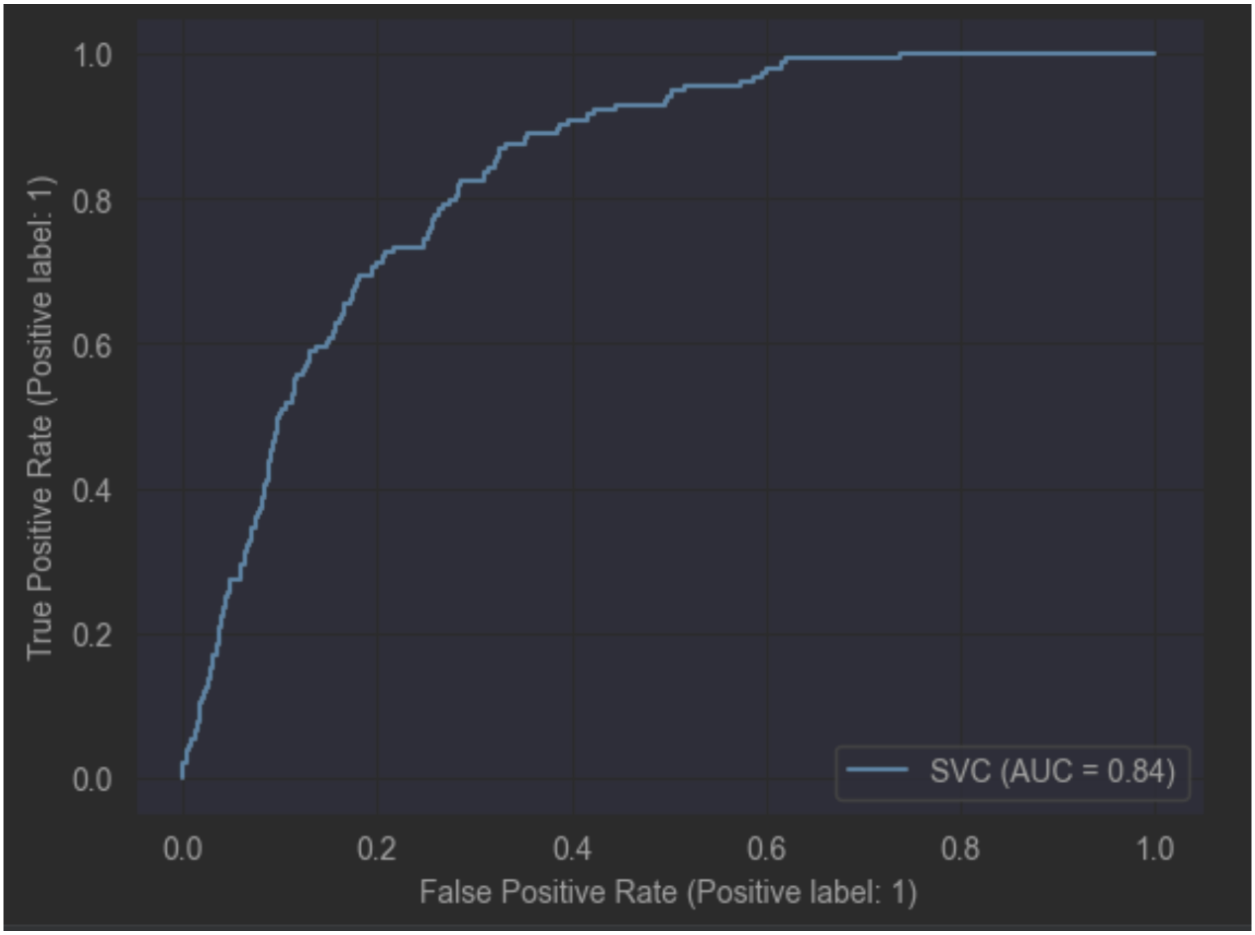
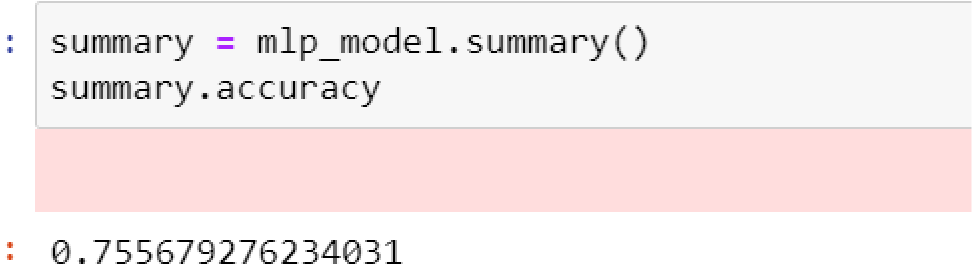


Figure 21. ROC curve

The further evaluation of the mode, we get the accuracy when the model is training using a training dataset, the accuracy is 75.57% as shown in Figure 22, which is lower than the model running for the test dataset (82%), which means the model has a good generalization.



```
summary = mlp_model.summary()
summary.accuracy
```

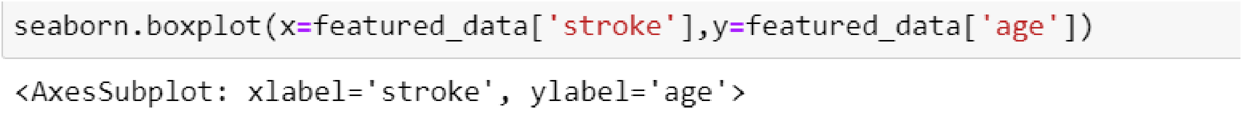
```
0.755679276234031
```

Figure 22. Model training dataset accuracy

Recall that the business goal of this data mining process is to discover patterns among attributes and their impact on stroke. This process is also considered to assess the importance of the attributes and draw a conclusion about the risk factors for each feature.

The main findings are listed below:

1. The age factor is much more influential than other medical conditions (hypertension, heart disease). As seen in Figure 20, age is much more important than other medical-related information. Most patients get stroke within the age range of 60 – 80 years old.



```
seaborn.boxplot(x=featured_data['stroke'],y=featured_data['age'])
```

```
<AxesSubplot: xlabel='stroke', ylabel='age'>
```

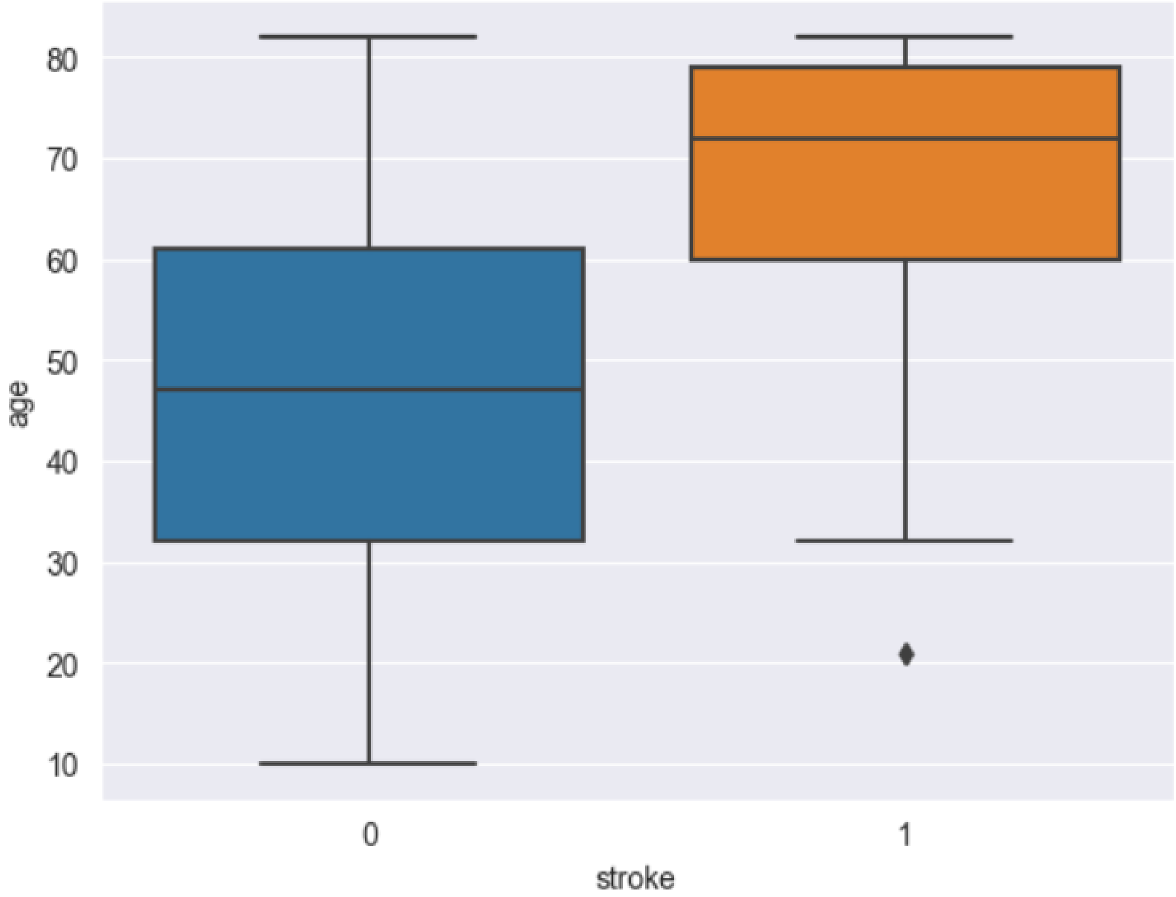
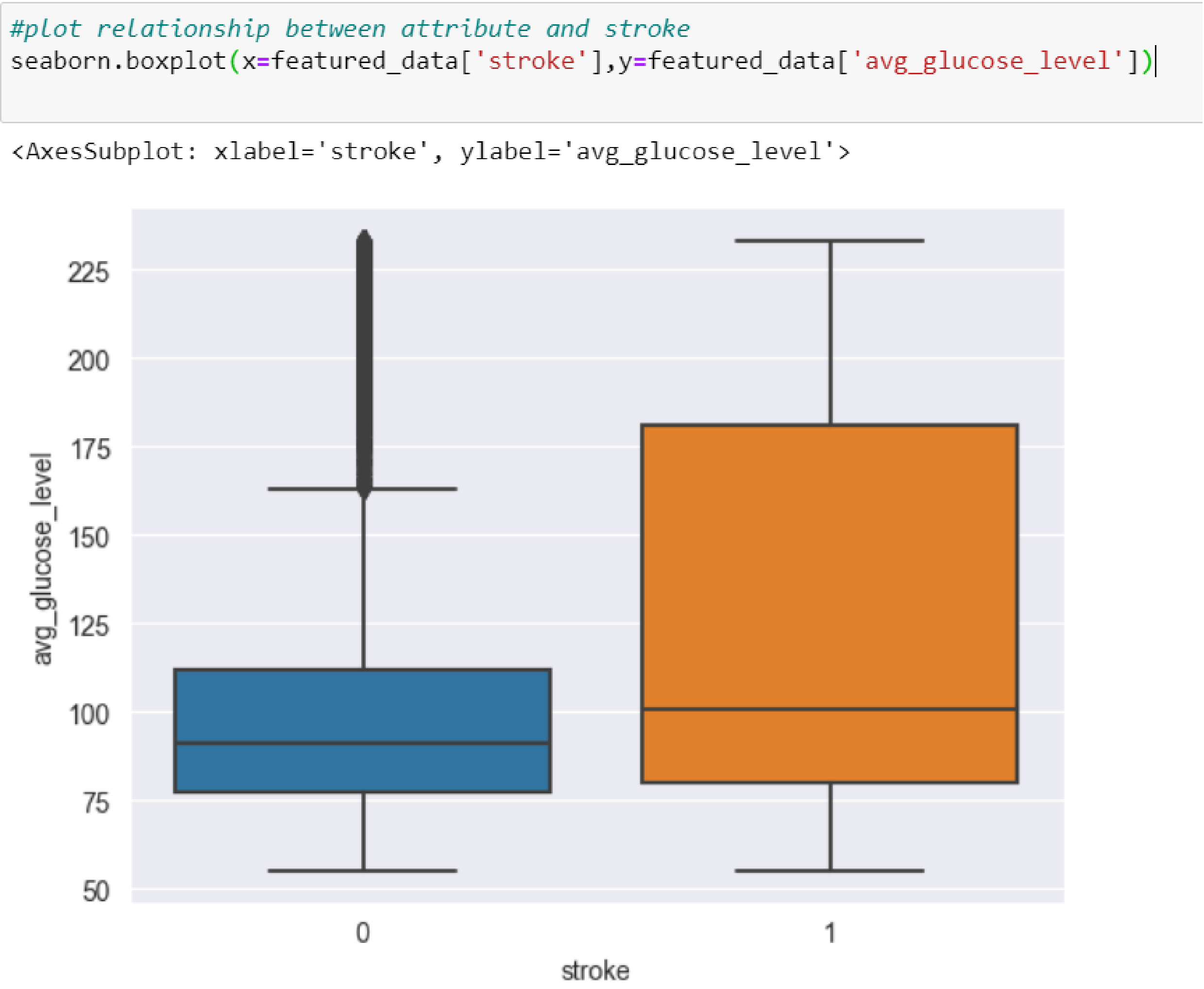


Figure 23. relationship of age and stroke

2. The average glucose is the second important predictor as the medical condition of stroke. Patients with higher average glucose level is more likely to get stroke. Most patients get stroke with the average glucose level in the range 113 – 180 mg/dL as shown in Figure 21. The normal blood glucose range is 70 to 99* mg/dL (Cleveland Clinic medical professional, 2018).



```
#plot relationship between attribute and stroke
seaborn.boxplot(x=featured_data['stroke'],y=featured_data['avg_glucose_level'])
```

```
<AxesSubplot: xlabel='stroke', ylabel='avg_glucose_level'>
```

Figure 24. relationship of average glucose level and stroke

10.2 Interpretation of Patterns:

The quantity and quality of the dataset are good because it provides a sufficient number of cases for training and the test sample covers most of the cases. The use of the oversampling technique repeating algorithm balanced the difference between the two labels without causing overfitting problems.

The model used for this dataset is a neural network. It addresses the shortcomings of random trees and random forests for space-based classification, as the characteristics of stroke patients and non-stroke patients are similar. It also addresses the disadvantage of SVM, which projects data to higher latitudes for classification and is not very efficient when there are many observed samples. As well as the shortcomings of logistic regression, when the feature space is large, logistic regression does not perform very well and is prone to underfitting and generally less accurate.

Neural networks were able to predict outcomes more accurately. 82% recall and accuracy proved that the model was usable and that the model matched the dataset well. Therefore, it is a good model.

Pattern 1：

Age is a much more important factor than other medical conditions (hypertension, heart disease) As shown in Figure 60, age is much more important than other medically relevant information. Most patients have a stroke between the ages of 60 and 80. It is indisputable that, when combined with blood glucose, older people are more likely to have high blood glucose and therefore more likely to cause a stroke.

Pattern 2：

The age factor is much more influential than other medical conditions (hypertension, heart disease). As shown in Figure 60, age is much more important than other medically relevant information. Most patients had a stroke between the ages of 60 and 80 years. This is indisputable, and in combination with blood glucose, the probability of occurrence of high blood glucose is higher in the elderly and therefore causes a greater chance of stroke.

Pattern 3：

The analysis of the importance of different attributes concluded that the average blood glucose level and BMI had a greater degree of influence than other medically relevant factors. the BMI value as well as the average blood glucose level showed the lifestyle of a person. According to the research paper, a person who is obese has a higher chance of getting a stroke than others, and a person with a lower BMI and average blood sugar maintains a good lifestyle. Therefore, they are less likely to have a stroke.

Pattern 4：

Heart disease is also an important factor in influencing and predicting stroke, and patients with a history of heart disease or related disorders are more likely to have a stroke. It is therefore clear that the study of factors that impress stroke should also focus on the patient's own underlying disease.

Pattern 5：

According to model 4, we can know that stroke patients and non-stroke patients have similar characteristics. So, in addition to that, the combination of several factors will have an impact on the final stroke situation. This is because patients with higher-than-normal blood glucose levels and older age will have a higher chance of having a stroke.

Pattern 6：

The type of work and smoking habits has a smaller impact on stroke. Figure 63 shows the importance of the attributes. It is clear from the figure that job category and smoking habits influence stroke, but the effect is limited. As a factor, these two attributes have no direct effect on stroke. However, these attributes as part of lifestyle, job type represents the stress and environment a person is exposed to, while smoking status influences the presence of hypertension and heart disease. Thus, the effects of job type and smoking habits on stroke are indirect.

The patterns discussed above suggest that medical-related information is not the most risk factor for stroke. The original hypothesis focused on the influence of medical information, but the factors found to be more influential are now blood glucose and age. Based on the research, it is recommended that potential patients with blood glucose and age above a certain threshold undergo regular physical examinations to reduce the risk of stroke.

11. Proposed actions

11.1 Limitation

The limitations of the current data mining model are listed below.

1. The model is not accurate enough. However, the current model has an accuracy rate of over 80%. For other areas, such as map route prediction or the field of weather forecasting, this is usually sufficient. However, in medicine, the results need to be accurate and precise. The model should be further improved to enhance its overall performance.

2. The data labels are unbalanced. There is a significant difference in labeling between stroke patients and non-stroke patients in this database, as most instances are labeled as non-stroke patients. In addition, the process of data balancing added noise to the original dataset by duplicating the collection of stroke patient cases, which is highly likely to affect the performance of the model. Based on the medical research aspect, the large amount of information on stroke patients should be further investigated.

3. The amount of data collected. The records collected are quite limited in the overall dataset, as it is slightly more than 43,400 cases. Such a volume of data makes it difficult to find valuable and meaningful patterns and to train models that meet the standards of medical use in terms of accuracy and recall. Therefore, more patient information should be studied.

4. The single source of data This dataset was collected from a limited number of sources. There are several factors that could potentially influence stroke. Location, different countries, long-term dietary habits, etc. could potentially have an impact. Therefore, records from multiple sources should be collected and evaluated to determine and validate the reliability of the data.

5. The number of attributes is limited, and the dataset contains a fairly limited number of attributes. The underlying diseases and chronic conditions associated with medical information are not fully covered. Further attributes may influence the occurrence of stroke. Therefore, several additional attributes should be added that could determine stroke risk more accurately.

11.2 Improvement

Based on the above limitations, the following improvements can be given.

1. further parameter tuning of the algorithm should be carried out to improve the overall performance of the model.

2. More records of stroke patients should be included in the study to make the difference between the two labels smaller and to try not to artificially balance the data by adding noise.

3. further records should be collected to allow for a larger sample space in the dataset.

4. more institutions should also be considered for inclusion in the current research programme so that records can be collected from multiple sources.

5. Further and more attributes should be provided to the existing model to provide a more accurate model.

6. Instead of directly determining whether a patient has had a stroke, the patient's level of risk should be flagged because early prevention of stroke occurrence is more important.

12. Reference

Adams, H. P., Bendixen, B. H., Kappelle, L. J., Biller, J., Love, B. B., Gordon, D. L., & Marsh, E. E. (1993). Classification of subtype of acute ischemic stroke. definitions for use in a multicenter clinical trial. toast. trial of ORG 10172 in acute stroke treatment. *Stroke*, *24*(1), 35–41. https://doi.org/10.1161/01.str.24.1.35

Centers for Disease Control and Prevention. (2022, October 14). *Stroke facts*. Centers for Disease Control and Prevention. Retrieved October 19, 2022, from https://www.cdc.gov/stroke/facts.htm#:~:text=Disease%20and%20Stroke-,Stroke%20statistics,disease%20was%20due%20to%20stroke.&text=Every%2040%20seconds%2C%20someone%20in,minutes%2C%20someone%20dies%20of%20stroke.&text=Every%20year%2C%20more%20than%20795%2C000,United%20States%20have%20a%20stroke.

Chong, J. Y. (2022, August 4). *Overview of stroke - brain, spinal cord, and nerve disorders.* MSD Manual Consumer Version. Retrieved August 15, 2022, from https://www.msdmanuals.com/home/brain,-spinal-cord,-and-nerve-disorders/stroke-cva/overview-of-stroke

Cleveland Clinic medical professional. (2018, February 21). *Blood glucose test: Levels & What They mean*. Cleveland Clinic. Retrieved September 19, 2022, from https://my.clevelandclinic.org/health/diagnostics/12363-blood-glucose-test#:~:text=A%20blood%20glucose%20test%20is,indicate%20pre%2Ddiabetes%20or%20diabetes.

Grefkes, C., & Fink, G. R. (2020). Recovery from stroke: Current concepts and future perspectives. *Neurological Research and Practice*, *2*(1). https://doi.org/10.1186/s42466-020-00060-6

Fang J, Keenan NL, Ayala C, Dai S, Merritt R, Denny CH. Awareness of stroke warning symptoms—13 states and the District of Columbia, 2005. *MMWR*. 2008;57(18):481–5.

Ji, R., Schwamm, L. H., Pervez, M. A., & Singhal, A. B. (2013). Ischemic stroke and transient ischemic attack in young adults. *JAMA Neurology*, *70*(1), 51. https://doi.org/10.1001/jamaneurol.2013.575

Kim, J., Thayabaranathan, T., Donnan, G. A., Howard, G., Howard, V. J., Rothwell, P. M., Feigin, V., Norrving, B., Owolabi, M., Pandian, J., Liu, L., Cadilhac, D. A., & Thrift, A. G. (2020). Global stroke statistics 2019. *International Journal of Stroke*, *15*(8), 819–838. https://doi.org/10.1177/1747493020909545

Kleindorfer DO, Towfighi A, Chaturvedi S, Cockroft KM, Gutierrez J, Lombardi-Hill D, Kamel H, Kernan WN, Kittner SJ, Leira EC, Lennon O, Meschia JF, Nguyen TN, Pollak PM, Santangeli P, Sharrief AZ, Smith SC Jr, Turan TN, Williams LS. 2021 Guideline for the Prevention of Stroke in Patients With Stroke and Transient Ischemic Attack: A Guideline From the American Heart Association/American Stroke Association. Stroke. 2021

Jul;52(7):e364-e467. doi: 10.1161/STR.0000000000000375. Epub 2021 May 24. Erratum in: Stroke. 2021 Jul;52(7):e483-e484. PMID: 34024117.

Maaijwee, N. A., Rutten-Jacobs, L. C., Schaapsmeerders, P., van Dijk, E. J., & de Leeuw, F.-E. (2014). Ischaemic stroke in young adults: Risk factors and long-term consequences. *Nature Reviews Neurology, 10*(6), 315–325. https://doi.org/10.1038/nrneurol.2014.72

Pan, B., Jin, X., Jun, L., Qiu, S., Zheng, Q., & Pan, M. (2019). The relationship between smoking and stroke. *Medicine*, *98*(12). https://doi.org/10.1097/md.0000000000014872

Putaala, J., Curtze, S., Hiltunen, S., Tolppanen, H., Kaste, M., & Tatlisumak, T. (2009). Causes of death and predictors of 5-year mortality in young adults after first-ever ischemic stroke. *Stroke*, *40*(8), 2698–2703. https://doi.org/10.1161/strokeaha.109.554998

Roy-O'Reilly, M., & McCullough, L. D. (2018). Age and sex are critical factors in ischemic stroke pathology. *Endocrinology*, *159*(8), 3120–3131. https://doi.org/10.1210/en.2018-00465

Singhal, A. B., Biller, J., Elkind, M. S., Fullerton, H. J., Jauch, E. C., Kittner, S. J., Levine, D. A., & Levine, S. R. (2013). Recognition and management of stroke in young adults and adolescents. *Neurology*, *81*(12), 1089–1097. https://doi.org/10.1212/wnl.0b013e3182a4a451

Tsao CW, Aday AW, Almarzooq ZI, Alonso A, Beaton AZ, Bittencourt MS, et al. Heart Disease and Stroke Statistics—2022 Update: A Report From the American Heart Association. *Circulation*. 2022;145(8):e153–e639.

WHO. (2020, December 9). *The top 10 causes of death*. World Health Organization. Retrieved July 29, 2022, from https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

Yousufuddin, M., & Young, N. (2019). Aging and ischemic stroke. *Aging*, *11*(9), 2542–2544. https://doi.org/10.18632/aging.101931