**Evaluating suitable algorithm for heart stroke prediction to build model**

Table of Contents

[Introduction 1](#_Toc123829890)

[Mission statement 2](#_Toc123829891)

[Aim of the research 2](#_Toc123829892)

[Objectives of this research 2](#_Toc123829893)

[Prototype identification and planning 3](#_Toc123829894)

[Literature review on prototype identification 3](#_Toc123829895)

[Reflection on the prototype identification 4](#_Toc123829896)

[Section 2: Development 6](#_Toc123829897)

[Development codes for heart stroke prediction 6](#_Toc123829898)

[Section 3: Evaluation 22](#_Toc123829899)

[Report on the evaluation 22](#_Toc123829900)

[Modelling 23](#_Toc123829901)

[Model choice 24](#_Toc123829902)

[Model comparison 26](#_Toc123829903)

[Conclusion 40](#_Toc123829904)

[References 42](#_Toc123829905)

**Table of figures:**

[Figure 1: Importing Libraries 7](#_Toc123830080)

[Figure 2: Loading data 8](#_Toc123830081)

[Figure 3: Data fetch 8](#_Toc123830082)

[Figure 4: Output of fetchdata.head() 9](#_Toc123830083)

[Figure 5: Counting data 9](#_Toc123830084)

[Figure 6: Bar of data count 10](#_Toc123830085)

[Figure 7: Describe function 11](#_Toc123830086)

[Figure 8: Output of describe 12](#_Toc123830087)

[Figure 9: Stroke Frequency according to gender 13](#_Toc123830088)

[Figure 10: A plot of bar of sex/stroke 14](#_Toc123830089)

[Figure 11: Modified plot 14](#_Toc123830090)

[Figure 12: Modified plot bar 15](#_Toc123830091)

[Figure 13: Code for stroke in function of age and max heart rate 16](#_Toc123830092)

[Figure 14:stroke in function of age and max heart rate diagram 16](#_Toc123830093)

[Figure 15: Histogram 17](#_Toc123830094)

[Figure 16: Correlation matrix 18](#_Toc123830095)

[Figure 17: fetching output of correlation matrix 18](#_Toc123830096)

[Figure 18: Creating Correlation chart 18](#_Toc123830097)

[Figure 19: correlation chart 19](#_Toc123830098)

[Figure 20:Train and test code 24](#_Toc123830099)

[Figure 21: test length 24](#_Toc123830100)

[Figure 22: Choosing appropriate model 25](#_Toc123830101)

[Figure 23: Accuracy of models 25](file:///E:\5th%20sem\AI%20shrawan\AI%20assignment%20nayna.docx#_Toc123830102)

[Figure 24: bar comparison of models 26](#_Toc123830103)

[Figure 25: creating train and test score 27](#_Toc123830104)

[Figure 26: output of train and test score 27](#_Toc123830105)

[Figure 27: code for maximum KNN score 28](#_Toc123830106)

[Figure 28: Chart for KNN score 28](#_Toc123830107)

[Figure 29: Tuning logistic regression using RandomizedSearchCV 29](#_Toc123830108)

[Figure 30: Tuning model using GridSearchCV 30](#_Toc123830109)

[Figure 31: Beyond accuracy evaluation 31](#_Toc123830110)

[Figure 32: Confusion matrix 33](#_Toc123830111)

[Figure 33: Classification report 34](#_Toc123830112)

[Figure 34: Cross validating matrix 36](#_Toc123830113)

[Figure 35: contribution of feature 37](#_Toc123830114)

[Figure 36: Feature evaluation 38](#_Toc123830115)

[Figure 37: Exporting trained model 39](#_Toc123830116)

# 

# Introduction

A study area of computer science field, Artificial Intelligence (AI) holds the objective of creating or training machines that can think like people do along with learning, reasoning, and self-correction. (Kok, et al., 2009)

When the blood flow is disturbed, a stroke ensues. When the blood flow to a portion of the heart stops and the cells are deprived oxygen, this condition is known as a stroke. (Chefung, 2014)

Here, the supervised learning approach will be uses where binary classification of data downloaded from Kaggle is tested using different models. Every year, people having stroke is 13 among which the number of them dying is 5.5 million as a result, regardless of their age, sex or physical condition according to the World Stroke Organization. Because it is the largest cause of mortality and disability in the world, it has a major impact on every part of life. The person’s family, working environment social network is also impacted by stroke. (Dritsas & Trigka, 2022)

With the demographic information (age, gender), lifestyle factors (diet, exercise habits, and tobacco and alcohol use), and medical history (history of hypertension, diabetes, and previous heart attacks), stroke can be predicted. To identify the pertinent patterns and predict the chance of a heart attack for a specific individual, the model should be trained on a dataset of people who have experienced heart attacks and those who have not.

A probability or binary prediction (such as "yes" or "no") indicating whether or not a person is likely to experience a heart attack will be the model's output. Doctors and other healthcare professionals can use this prediction to identify those who are at high risk and suggest preventive measures to minimize their risk.

Early diagnosis enables people to start taking preventative actions, such changing their lifestyle or starting a pharmaceutical regimen, to lower their risk of having a stroke. Additionally, it can assist healthcare professionals in developing more individualized treatment plans and prioritizing care for patients with maximum stroke risk. Furthermore, forecasting the probability of a stroke can help healthcare systems deploy their resources more effectively and possibly reduce total care expenses.

# Mission statement

The prime objective of this prototype development is to predict heart stroke using machine learning involves building a model that is able to predict the likelihood of an individual suffering from a heart stroke based on a set of input features so that they can take precautions to lower their risk.

# Aim of the research

Basically, this research clutches the aim of developing a model that can identify people who are at a high risk of having heart stroke to suggest preventive measures to reduce their risk.

# Objectives of this research

1. To build a model that can accurately predict heart stroke looking at the medical condition of an individual from the appropriate gathered dataset.
2. To identify potential risk factors for heart stroke by analyzing the input features used by model to make prediction.
3. To evaluate the models based in their performance to find suitable one.

# Prototype identification and planning

## Literature review on prototype identification

The scientific community has shown a great deal of interest in the development of tools and strategies for monitoring and forecasting a wide variety of diseases that have a notable impact on human health. The most recent studies that anticipate stroke risk using machine learning algorithms are listed in this section.

The authors (Pradeepa, et al., 2020) initially used 4 machine learning approaches: naive Bayes, J48, KNN, and random forest, in order to successfully detect a stroke. Compared to the naive Bayes classifier's accuracy of 85.6%, the accuracy of the J48, K-nearest neighbor, and random forest classifiers was 99.8%. However, we are in charge of selecting the model based on our needs and requirements.

(Wilson, et al., 2008) in their research says that among many factors age, sex, high-density lipoprotein cholesterol, systolic blood pressure, smoking, and diabetes mellitus are the main risk factors for cardiovascular disease (CVD), according to their study.

(Lackland, et al., 2012) says patients who have suffered an atherosclerotic stroke should be included in the category of persons who are thought to be at high risk (>=20% over 10 years) of developing new atherosclerotic coronary events. If non-atherosclerotic stroke subtypes ought to be covered, it is yet uncertain. In absolute risk assessment algorithms with the goal of primary prevention, ischemic stroke should be considered a cardiovascular disease outcome. These changes are likely to have a significant impact on the prevention of cardiovascular disease because they would result in a significant rise in the number of patients at high risk due to the inclusion of atherosclerotic ischemic stroke as a high-risk condition and the inclusion of ischemic stroke as an outcome more generally.

To help doctors provide treatment, a variety of strategies have been widely employed to forecast the presence, course, and prognosis of the ailment. The most objective line of action is to compare administered categorization of machine learning calculations and suggest a machine learning-based strategy to forecast the heart stroke with the best degree of precision. Additionally, (Babu, et al., 2021) calculated receiver hopeful bend and range beneath the bends for each classifier. Additionally, contrast the execution of the various machine learning computations from the offered example with an evaluation report, and categorize data to distinguish the perplexity framework.

(Smith, et al., 1998) The goal of this study was to comprehend the possible contributions of haemostatic and rheological parameters to the association between ischemic heart disease (IHD) and stroke in individuals with peripheral artery disease. To do this, they examined 607 individuals with intermittent claudication in the Peripheral Vascular Clinic at the Royal Infirmary of Edinburgh between 1989 and 1990 over the course of six years. The number of fatal and non-fatal strokes, non-fatal heart attacks, coronary deaths, and total coronary events served as the study's primary outcome measures. 203 of the 210 patients under close observation showed no indications of increasing limb ischemia or any other vascular issuesThe research determined that individuals who experienced certain episodes, such as non-fatal heart attacks, had higher median levels of fibrinogen, von Willebrand factor, tissue plasminogen activator, fibrin D-dimer, and whole blood viscosity than those who did not. Even after accounting for age and sex, the presence of fibrin D-dimer was linked to a significantly increased risk of non-fatal myocardial infarction (RR 1.50, 95% CI 1.09-2.06, P<0.01). The study also found that fibrinogen and fibrin D-dimer were associated with an overall higher frequency of coronary events (P<0.05 and RR 1.87, respectively, with 95% CI 1.04-3.34).

Top of Form

The researchers, (Gavhane, et al., 2018), were trying to find a way to identify and prevent heart stroke symptoms, particularly in young people, due to the rising rates of heart stroke. They believed it was important to have a reliable method for predicting the risk of heart disease that was practical and accessible to the general public, as it is not feasible for most people to undergo expensive tests like an ECG on a regular basis. Their proposed solution was to use machine learning, specifically neural networks, to create an algorithm that could determine a person's likelihood of having a cardiac illness based on simple factors like age, sex, and pulse rate. They believed that this would be the most accurate and trustworthy approach for predicting heart disease risk.

## Reflection on the prototype identification

A prototype is a rough copy of a product that is tested and used to illustrate concepts. A prototype for stroke prediction could entail creating and testing a model or algorithm that is intended to recognize those who are most at risk for having a stroke.

Data collection from a large group of people, including demographic data, medical history, and numerous biomarkers that have been connected to an elevated risk of stroke, may be one method for creating a stroke prediction prototype. A machine learning model might then be trained using this data to find patterns and correlations that point to a high risk of stroke. (Schwab, 2014)

To determine how effectively the prototype works at identifying people who possess high chance of having a stroke, it may be tested next on a different dataset of individuals. This can entail assessing the model's specificity and sensitivity (the capacity to predict who will have a stroke) (the ability to correctly identify individuals who will not experience a stroke).

Another difficulty is that a variety of variables, such as age, gender, family history, and lifestyle elements including food and activity patterns, may affect a person's risk of stroke. It may be challenging to quantify and include these aspects in a prediction model, which might reduce the model's accuracy. (J. J. Wang, 2019)

Despite these obstacles, the creation of a stroke prediction prototype has the potential to be a useful tool for identifying people prone to high chance of getting a stroke and enabling them to take preventative actions to lower their risk. This might entail making adjustments to one's lifestyle, such as altering one's diet and engaging in more physical exercise, or it could entail using drugs to regulate factors having risk: high blood pressure/ high cholesterol.

Overall, creating a prototype for stroke prediction is a difficult and complex endeavor, but one that might greatly enhance the health and happiness of those having maximum risk of stroke.

# Section 2: Development

## Development codes for heart stroke prediction

The machine learning prototype for heart stroke detection was developed using Jupyter notebook (Python). Classification and regression both are basically types of supervised machine learning algorithm. Here the model developed used supervised machine learning i.e. binary classification since number of different features about person were used for predicting the presence of heart stroke in one. The algorithm used here returns a class label, such as “heart stroke” or “no heart stroke”, as of classification. Here, the model was trained according to the pre-existed model where the labeled data was provided. The data for building this model was taken from kaggle (<https://www.kaggle.com/datasets/prosperchuks/health-dataset?select=stroke_data.csv>). In this, attributes are independent variables and features are dependent variable. The patients different medical attributes that used here are independent variable, to predict the dependent variable (whether or not they are prone to stroke). As dataset has already been selected, the modelling framework used were:

1. Data exploring with exploratory data analysis where dataset was revised for finding further about it.
2. Model was trained where they were to learn predicting target variables with respect to other variables.
3. Problem-specific evaluation metrics were used to evaluate predictions done by the model.
4. Comparison of different models used to find the best working for our case.
5. The model after comparison was fine-tuned to further improve accuracy and performance.
6. Since this research is about the prediction of heart stroke, feature importance is what basically was focused on. This means the factors that contributes on heart stroke prediction were checked.
7. Cross-validation of data was done to check if the model still works for unseen data.

**Importing libraries**

Before starting the project, it’s custom that the required libraries are imported. Here, every libraries that are to be used are consolidated, in the first line of notebook. Pandas was imported below to work with csv format of data that was downloaded from the Kaggle repository. Similarly, NumPy was used to pre-process and manipulate arrays. The data needs to be visualized to some extent. So, matplotlib pyplot package was imported. As for complex visualization, importing seaborn is vital. The warnings ignore are used to neglect the warn-able situations.

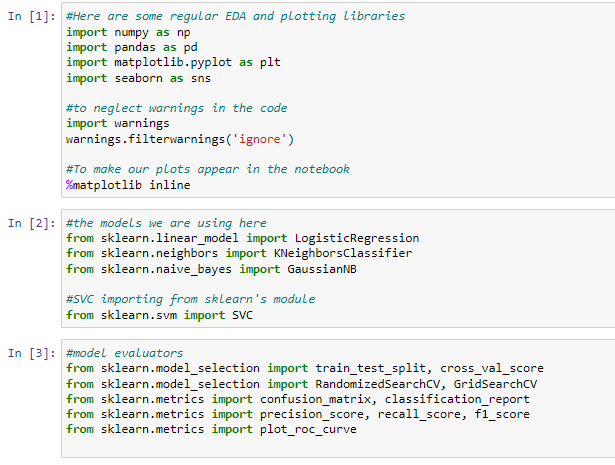


Figure 1: Importing Libraries

**Data Loading**

Despite of many ways of storing data, chose to store the data in Excel file in .csv format was chosen as data was tabular. The code presented below is to load the data. From line of code LOC(4) output is the number of rows and columns i.e. 40910 and 11 respectively.



Figure 2: Loading data

Here fetchdata.head() displays the top 5 data. for last 5 data we can use function tail().

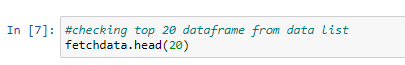


Figure 3: Data fetch

In LOC (7) the head (20) function is called to print top 20 rows from the entire datasets. However, customizing number of row is possible by passing numbers to head (). The output is given below:

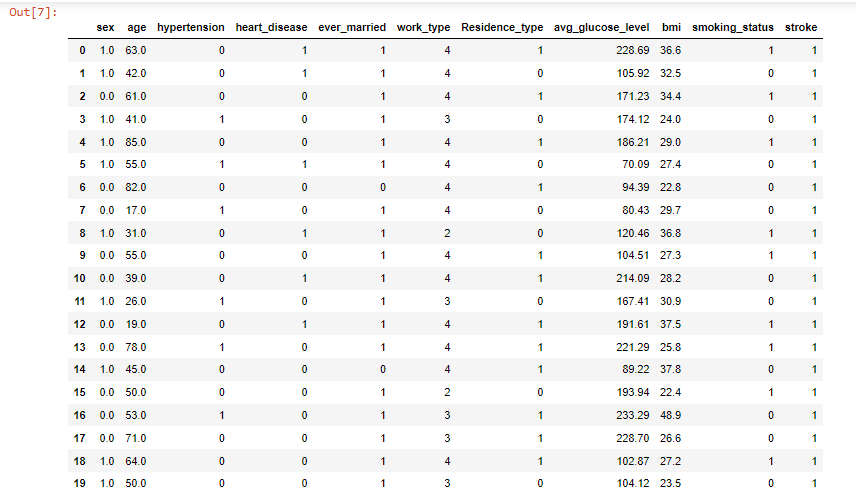


Figure 4: Output of fetchdata.head()

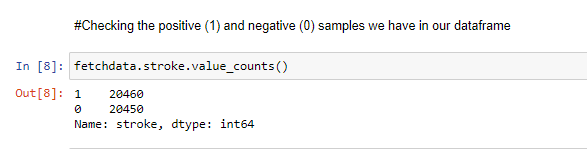


Figure 5: Counting data

Here, +ve (1) and –ve (0) samples are presented are shown. The targeted column here can be considered balanced as these two values are close to even.

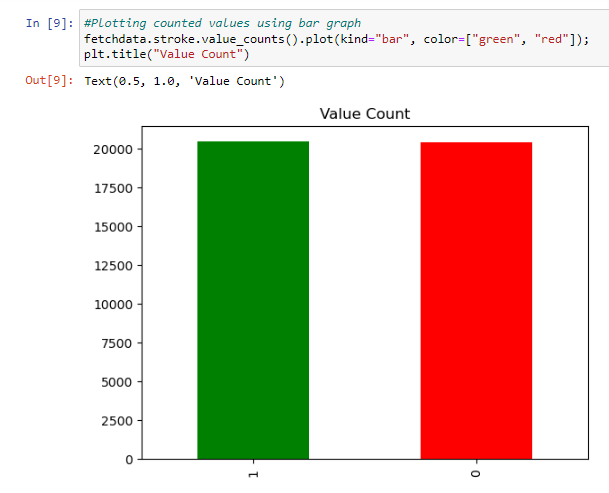
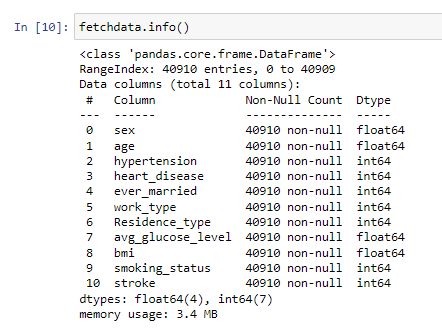


Figure 6: Bar of data count

In LOC(8), value\_count() function shows the number of occurrences of each value in a categorical column. The plot in LOC (9) plot () function plots the target column value count. The kind of plot can be chosen. However, bar is chosen.



In LOC (10) the fetchdata.info () gives out a quick insight to the number of missing values in the data along with its type.

**Descriptive statistics**

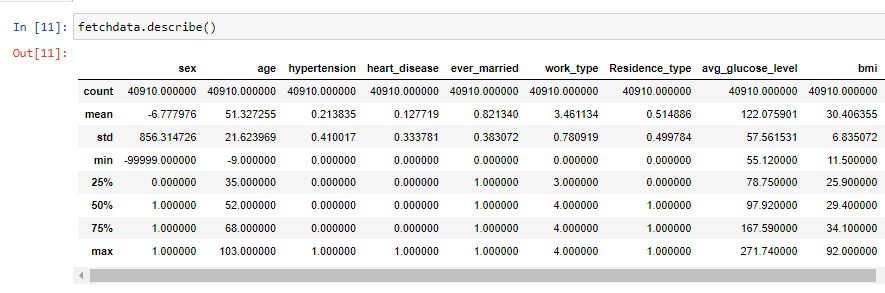


Figure 7: Describe function

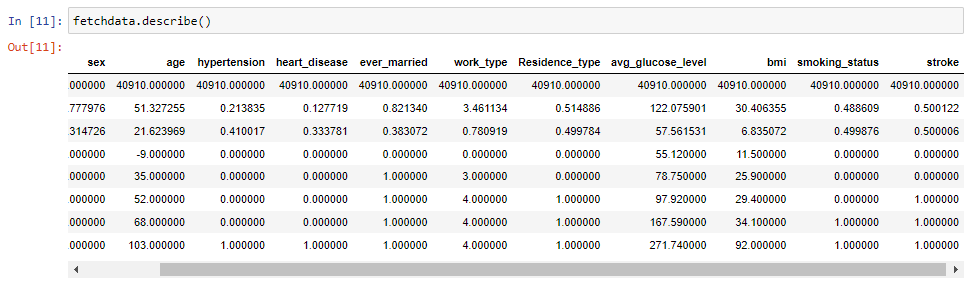


Figure 8: Output of describe

The fetchdata.describe () also enables getting insight of the data frame. Using this functionality, the range of different metrics about numerical values is shown like mean, min, max and standard deviation.

**Comparing two columns**

pd.crosstab (column\_1, column\_2) is used to compare two column with each other. To gain inter related intuition about the dependent and independent variables, in LOC (13) the target column with sex is compared. Here, for target column, heart stroke present= 1 and no heart stroke = 0. Similarly for sex, male is categorized 1 and female as 0.

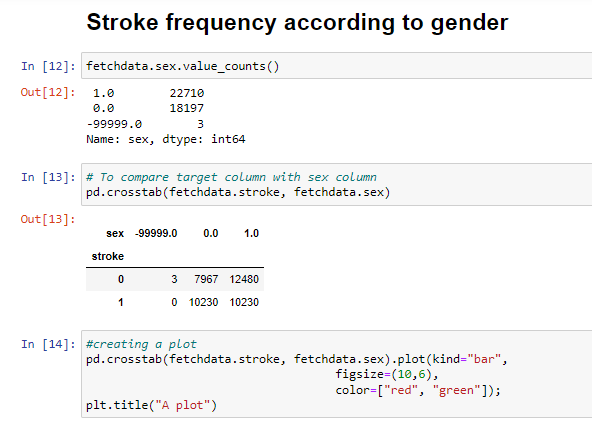


Figure 9: Stroke Frequency according to gender

From the LOC (12) the output shows that there are 22710 male and 18197 female in the dataset taken. In LOC (14) plot () function is used to plot the crosstab where parameters like figsize=(length, width), color= [color\_1, colour\_2] according to preference is passed.

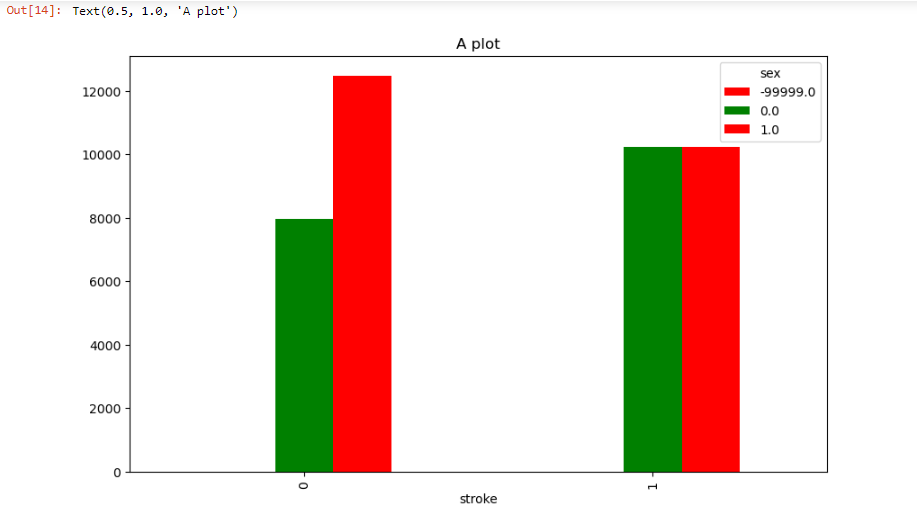


Figure 10: A plot of bar of sex/stroke

This is the output of plot without any attributes. The below is the plot that has been created with attributes like plt.title (), plt.xlabel (), plt.ylabel, plt.legend, plt.xticks.

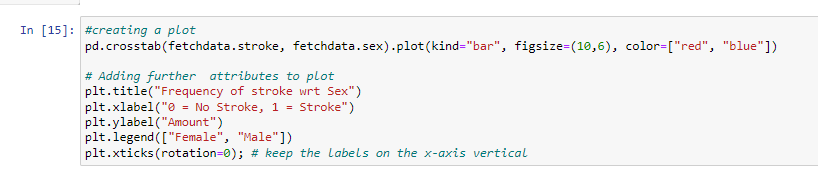


Figure 11: Modified plot

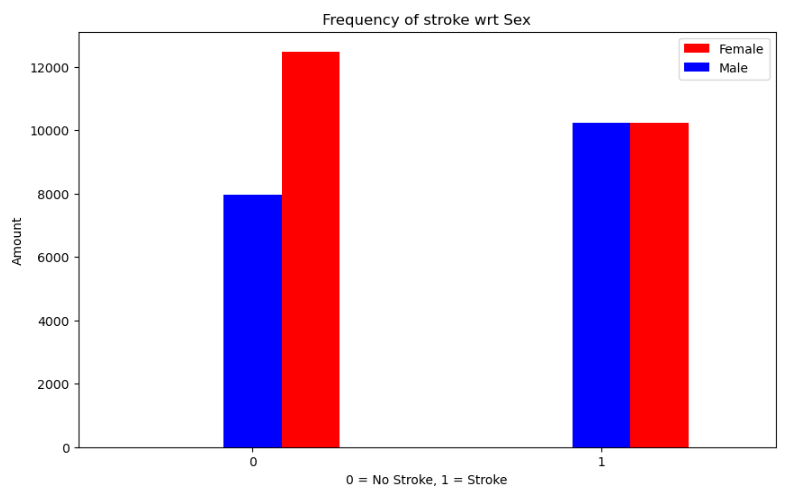


Figure 12: Modified plot bar

**Age vs. Max heart rate for stroke disease**

Here, I’ve attempted to examine the relationship between two independent variables (age and thalach, or maximum heart rate) and the target variable (heart stroke

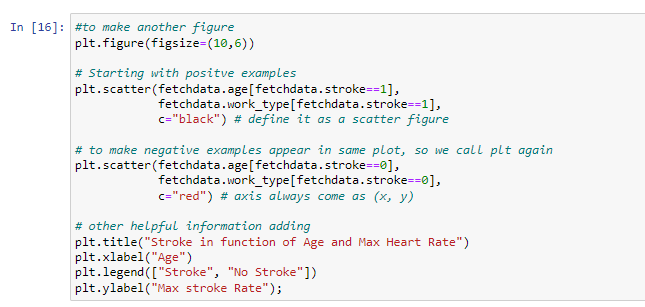


Figure 13: Code for stroke w.r.t age and max heart rate



Figure 14:stroke w.r.t of age and max heart rate diagram

The output above shows that the heart rate in younger people are maximum (on the left of the graph we see higher dots). As seen, black dots in the right side is maximum in the graph where the age is increasing.

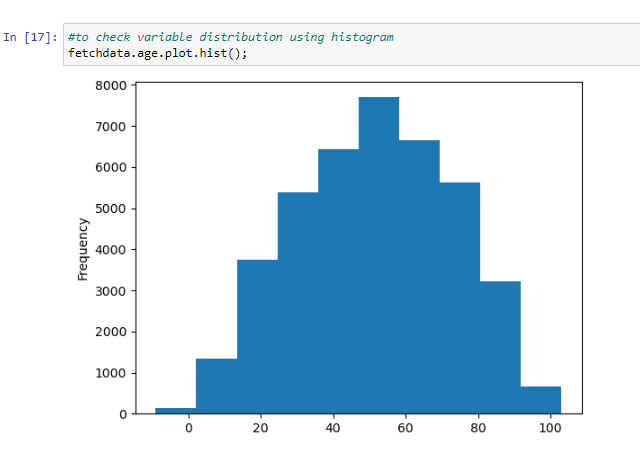


Figure 15: Histogram

**Correlation between independent variables**

Below is the table to demonstrate correlation matrix to find the strength along with relationship direction between several variables. For above dataset, correlation matrix is relationship between different variables like blood pressure and cholesterol levels, or age and stroke risk. Using fetchdata.corr() creates a correlation matrix to make it easy to find relation between variables. Hereby, impact of independent variable on target variable can be checked too.

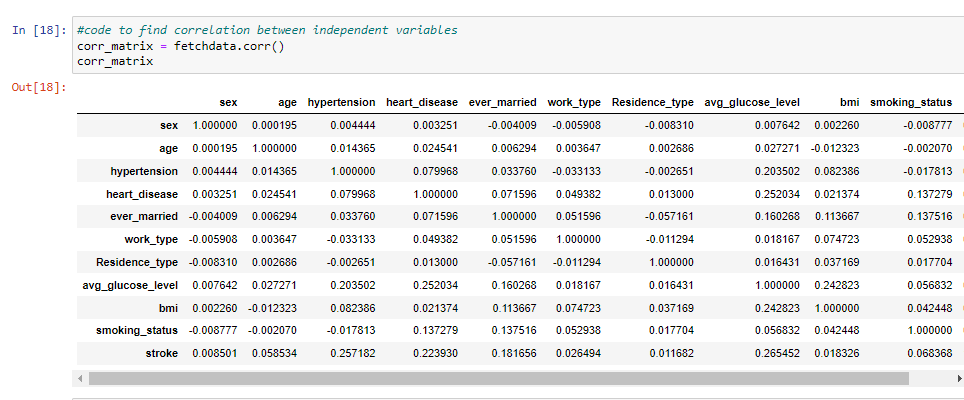


Figure 16: Correlation matrix

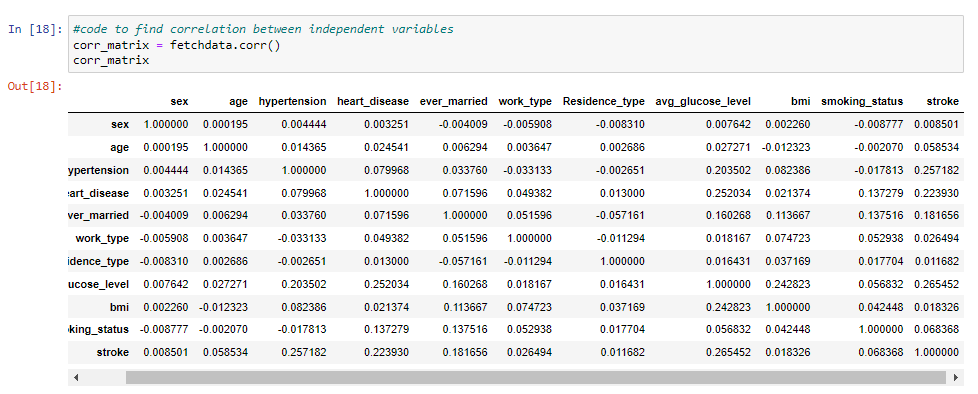


Figure 17: fetching output of correlation matrix

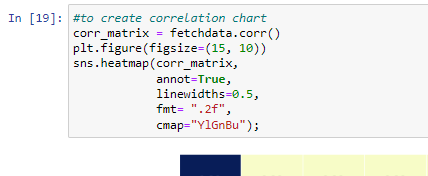


Figure 18: Creating Correlation chart

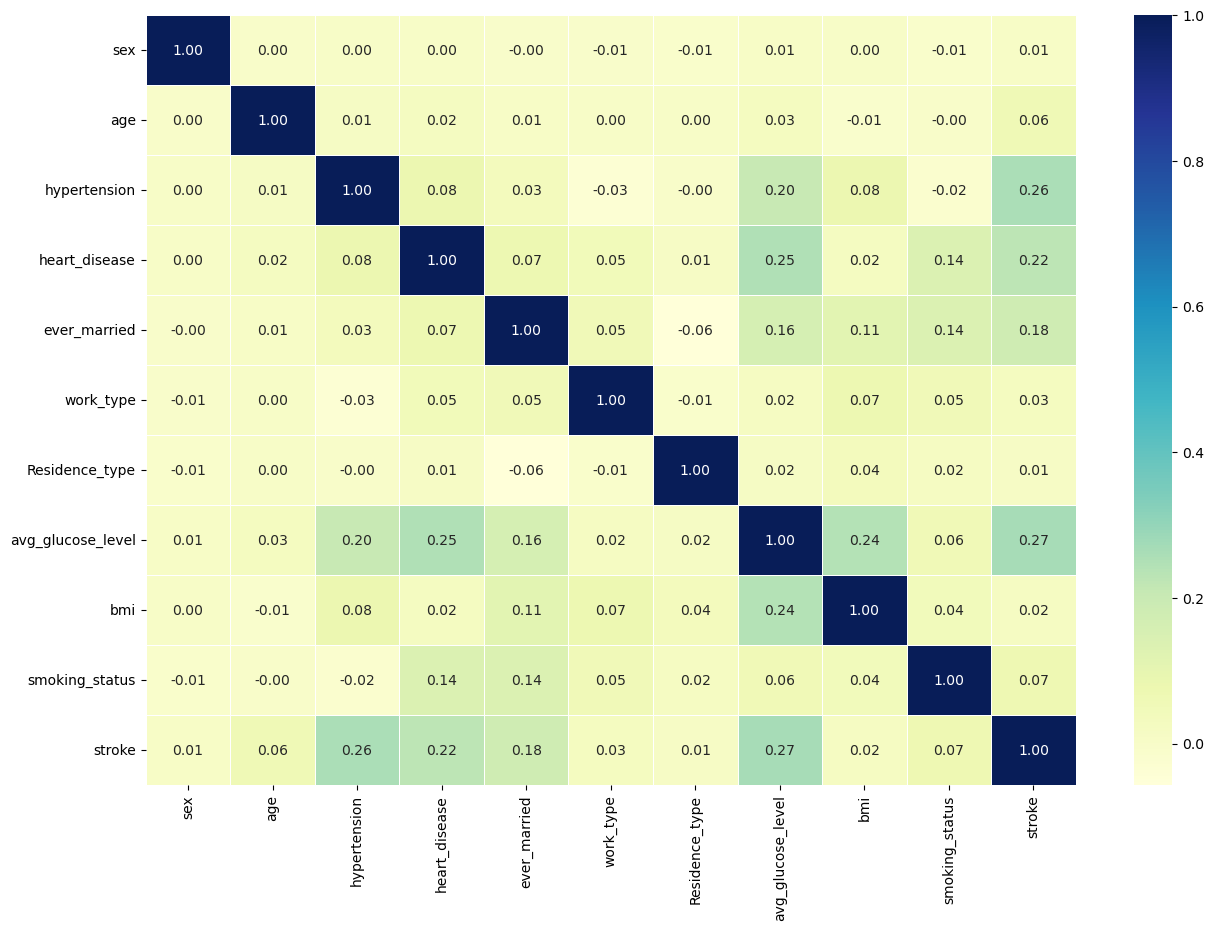
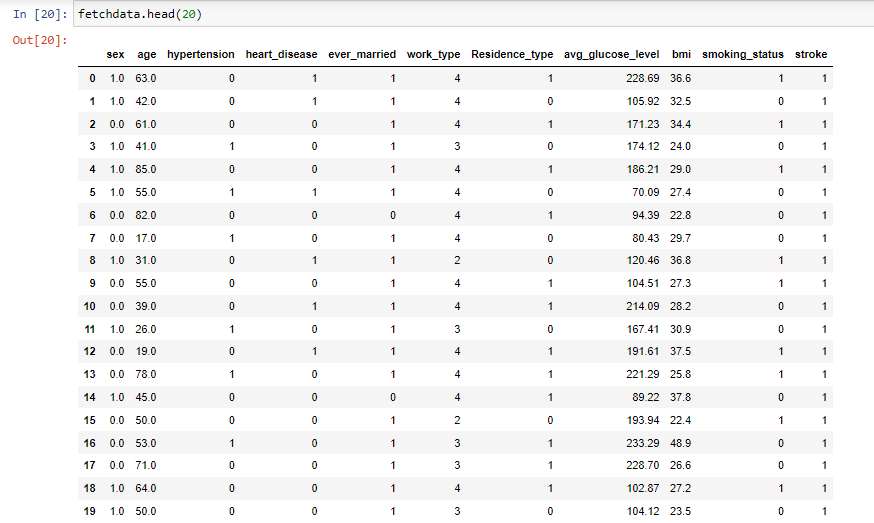


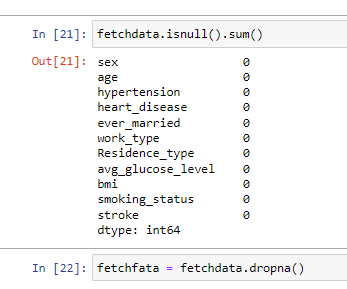
Figure 19: correlation chart

Here in the figure above, a potential positive correlation (increase) is shown by higher positive values whereas potential negative correlation (decrease) shown by higher negative values.

**Data processing**

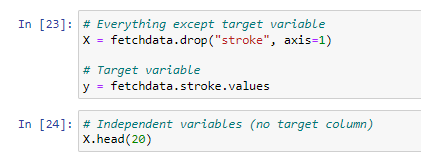
The pandas DataFrame records' columns' total amount of missing values (also known as NaN values) is determined by the code that follows. A boolean mask indicating whether or not each element in the DataFrame is missing is what the isna function returns. When a value is missing, True is returned; and False with a value present. When a value is missing, the isna method returns True, and when a value is present, it returns False.



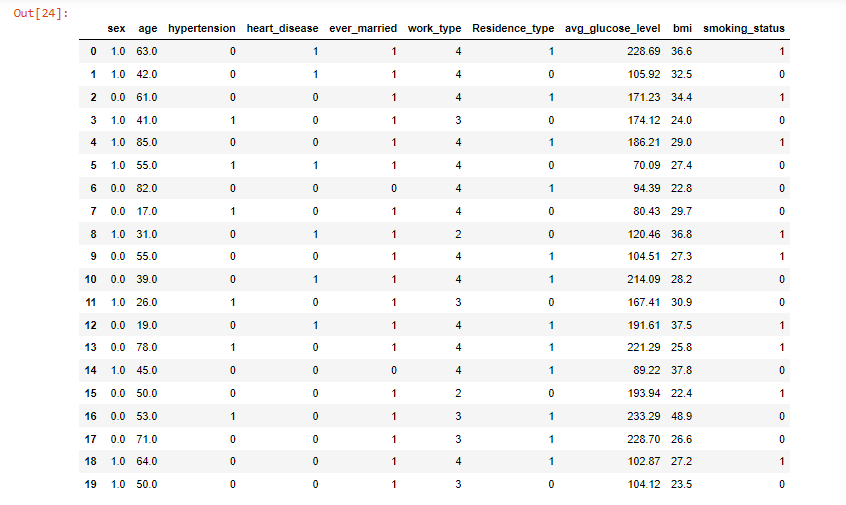


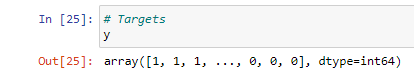
**Data Scaling**

Before training a model, data needs to be scaled. The method of standardization, which scales the data to have a mean of 0 and a variation of 1, is employed here. The data can be scaled in a number of ways. The equation is (X-mean(X))/standard (X). Scaling the data before training a machine learning model is typically a good idea. As we can see in the output in the following images, the model's performance with and without scaling must be compared to determine which performs better.



In the above code in LOC (23), target variable was predicted using other variables by splitting the target variable from the rest.





# Section 3: Evaluation

# Report on the evaluation

This section complies with description of the models used in stroke prediction. Basically, details on model developed, data sources used, along with the algorithms or techniques employed with description and comparison. Evaluation metrics used to evaluate the models are also explained here. The measures like accuracy, sensitivity, specificity, or AUC are metrics that contributes. Here, accuracy of models presents the results of the model evaluation. The result so found is presented in concise manner enabling context provision and interpretations for finding.

1. **KNN**

The K-nearest neighbors (K-NNs) classifier is an approach that calculates the resemblance or variance between two occurrences in a dataset using a measure of distance such as Manhattan/ Euclidean distance. Euclidean distance is the most commonly used measure of distance in this context. The K-NNs classifier works by finding the K instances in the dataset that are most similar or different (depending on the task at hand) to the instance being analyzed, and using those instances to make a prediction or classification. (Dritsas & Trigka, 2022)

1. **Logistic Regression**

Logistic regression (LR), classification technique, was originally developed for binary tasks. A binary variable representing the model's output, 1p=P(Y=0), captures the likelihood of an occurrence being the "Non-Stroke", whereas p=P(Y=1) denotes the chance of an occurrence fitting in "Stroke".

(Dritsas & Trigka, 2022)

1. **NavBayes**

The naïve Bayes (NB) classifier is based on the idea of maximizing probability, assuming that the characteristics being analyzed are highly independent. To classify a new instance with characteristics vector fi, the class c for which P(c|fi1,...,fin) is maximized is chosen. P(c|fi1,...,fin) is a conditional probability, which is defined as the probability of class c occurring given the values of the characteristics fi1,...,fin. The NB classifier uses this probability to make a prediction or classification for the new subject.

The conditional probability P(c|fi1,...,fin) can be written as:

P(c|fi1,…,fin)=P(fi1,…,fin|c)P(c)P(fi1,…,fin) (1)

In this equation, P(fi1,…,fin|c) represents the likelihood of a feature given a class, P(fi1,...,fin) represents the likelihood of a feature prior to the class, and P(c) represents the likelihood of a class prior to the class. By maximizing the numerator of this equation, the probability is maximized, leading to the optimization problem shown in equation (2):

ĉ=argmaxP(c)∏j=1nP(fij|c), (2)

where c∈{Stroke,Non−Stroke}. In this equation, ĉ represents the predicted class (either Stroke or Non-Stroke), and the term ∏j=1nP(fij|c) represents the product of the likelihood of each feature given the class. The optimization problem aims to find the class c that maximizes this product.

(Dritsas & Trigka, 2022)

## Modelling

Training and test split separations the data to train and test set to train and test the models. All data are not taken to train the model so that it can work on the new dataset i.e. not listed in the above dataset. The practice here is to mimic taking the model to real environment with the help of test set. The model however does not learn from test set. train\_test\_split () enables splitting our data to train and test set. Here train and test was filled with the independent and dependent variable (x and y). Train and test sets are split into 80:20 ratio.

Logistic regression particularly is useful for binary classification tasks, such as identifying whether a patient has had a stroke or not. Logistic regression can also handle large datasets and high dimensional data well.

KNN does not make any assumptions about the data distribution as it is non-parametric. It is particularly useful for classification tasks when the decision boundary is not linear. KNN can also handle missing values in the data.

Naive Bayes is a simple and efficient algorithm that is easy to implement and scale. It is particularly useful when the data is high dimensional and/or the number of training examples is small. Naive Bayes is also resistant to over-fitting, which makes it a good choice for noisy or unbalanced data.

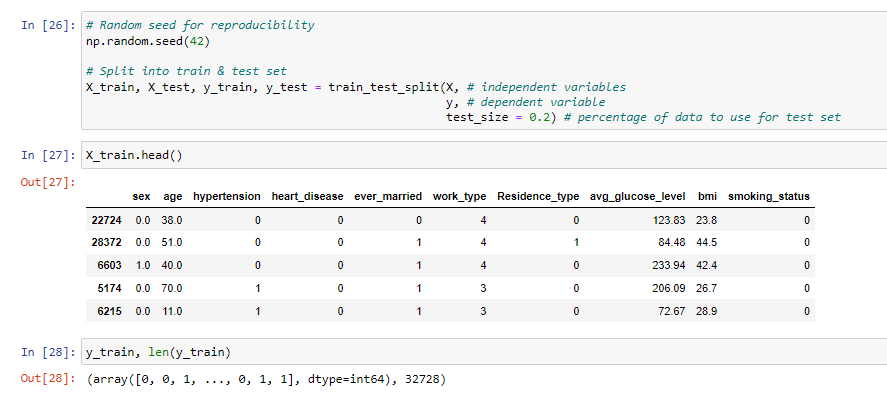


Figure 20:Train and test code

The test\_size parameter passes how much data is required to be tested to train\_test\_split(). Using thumb rule, 80% of data is trained and the rest is tested.

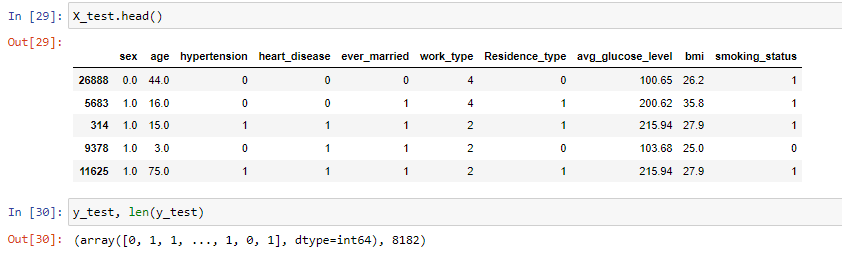


Figure 21: test length

## Model choice

After preparing the data, the models so used are.

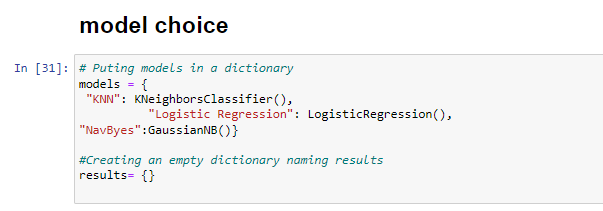


Figure 22: Choosing appropriate model

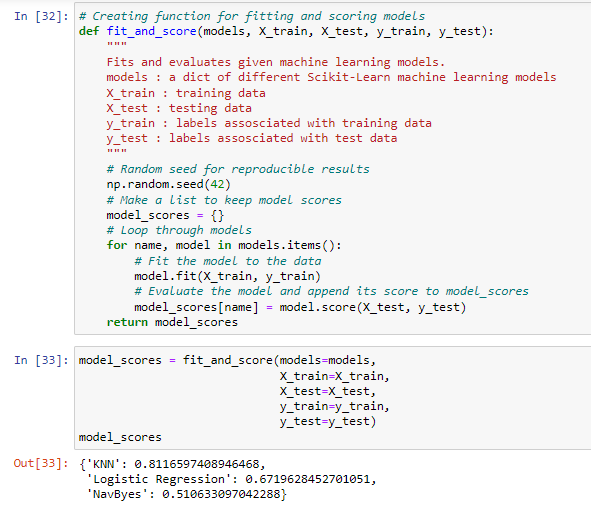


Figure 23: Accuracy of models

## Model comparison

Looking at the output, the model chosen seemed fit. However, visual comparison of the models is vital. The model scored are saved in the dictionary, so first they are to be converted into Data Frame to plot it.

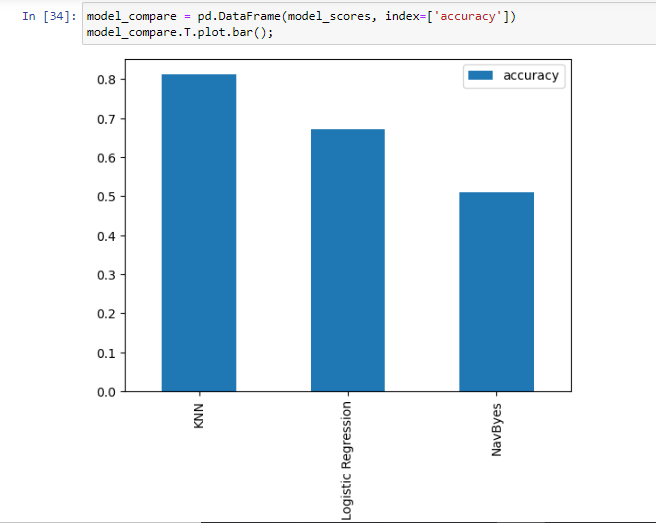


Figure 24: bar comparison of models

The graph clearly shows that the KNN () model is more accurate than the rest two. Here is tuning KNeighborsClassifier by hand.

**K\_Nearest Neighbor tuning**

Tuning KNN basically means tuning one main hyper parameter i.e. number of neighbors which is 5 by default. Different n-neighbors values are tried in the code below.



Figure 25: creating train and test score

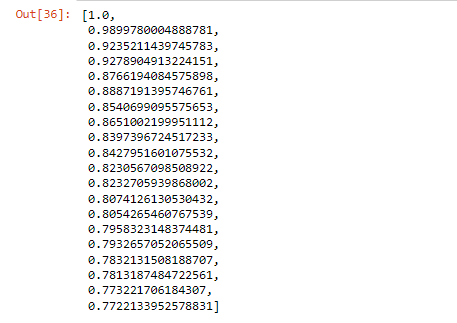


Figure 26: output of train and test score

The above output shows the train score of KNN. Below is the plotting of the above output where model score is plotted against number of neighbors.

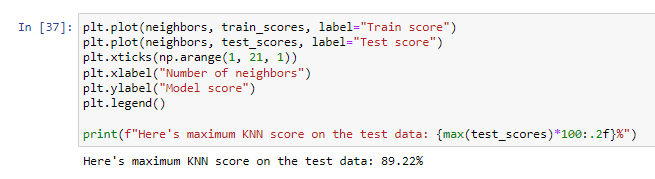


Figure 27: code for maximum KNN score

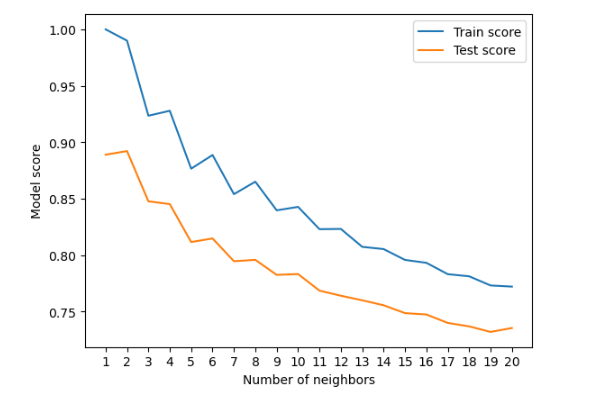


Figure 28: Chart for KNN score

In the above output, the train score and test score is shown.

**Tuning models with RandomizedSearchCV**

Hyper-parameter grid is the dictionary of different hyper-parameters. Here, the hyper-parameter grid will be created for each and tested.

**Cross-Validation**



Figure 29: Tuning logistic regression using RandomizedSearchCV

Here, RandomizedSearchCV was used to tune the models, logistic regression first. The different hyper-parameters from log\_reg\_grid was passed. The n-iter was set to 20 which means the algorithm iterates for 20 times.

Cross-validation (CV) divides a dataset into train and test sets, and using the train set to fit a model and the test set to evaluate the model. The value of CV=5 here specifies the number of folds that the dataset should be divided into i.e. 5 folds. This means that the algorithm will train on 4/5 of the data and test on 1/5 of the data, repeating this process 5 times such that each fold of the data is used as the test set once.

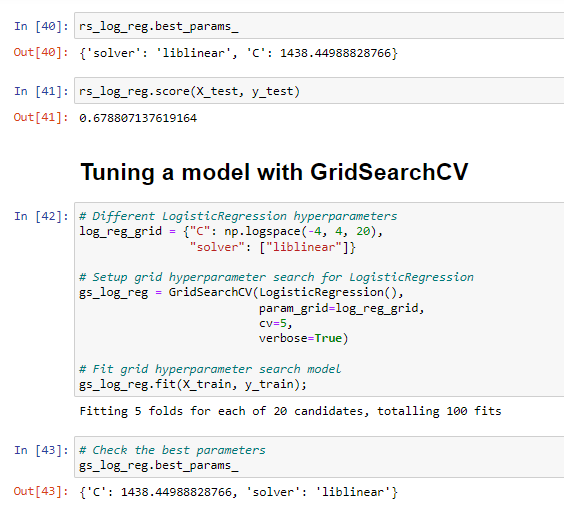
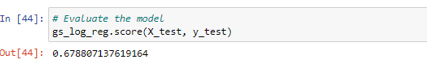


Figure 30: Tuning model using GridSearchCV

In contrast to RandomSearchCV, which performs n iter combinations on a grid of hyperparameters, GridSearchCV tests each and every conceivable combination.



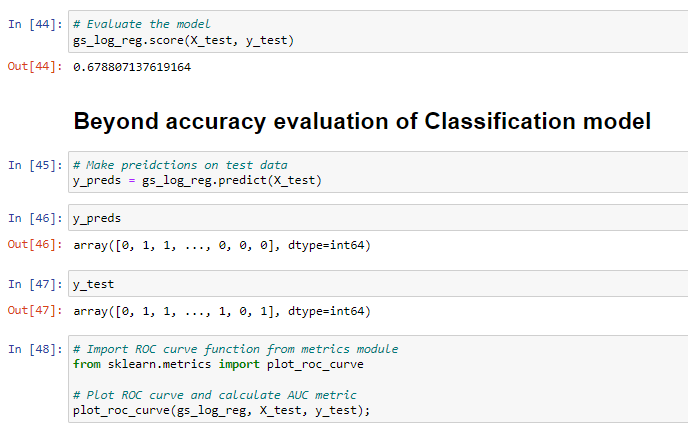
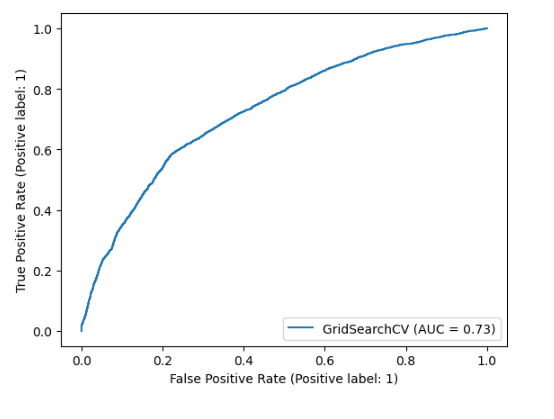


Figure 31: Beyond accuracy evaluation

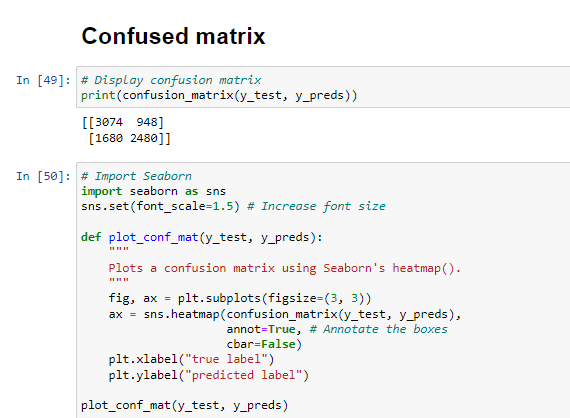
ROC curve, AUC score, confusion matrix, precision, classification report, recall and F1-score is checked.



The perfect model would have achieved AUC score of 1.0, so this model has to be improved.

**Confusion matrix**

Confusion matrix is used here to visually show the right and wrong prediction the model made. The function heatmap() is used to create heatmap to visualize larger data.



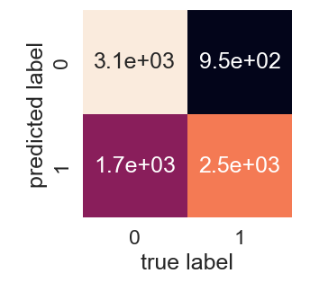


Figure 32: Confusion matrix

The model is confused because it predicts the wrong label relatively to both classes. On four occasions, the model correctly predicted 0 when it should have predicted 1, and on three occasions, the model correctly predicted 1 when it should have predicted 0. (false positive).

**Classification report**

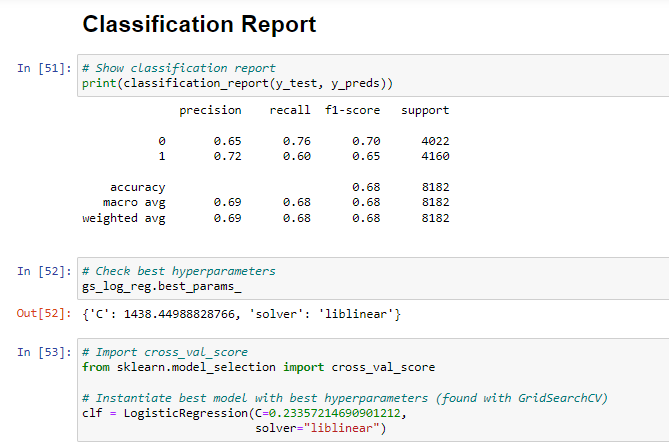
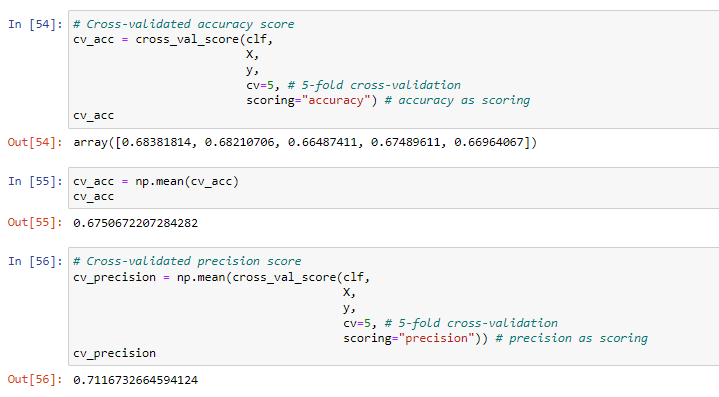


Figure 33: Classification report



The average was taken from 5 metrices in LOC (56).



**Visualizing Cross-Validated metrics**

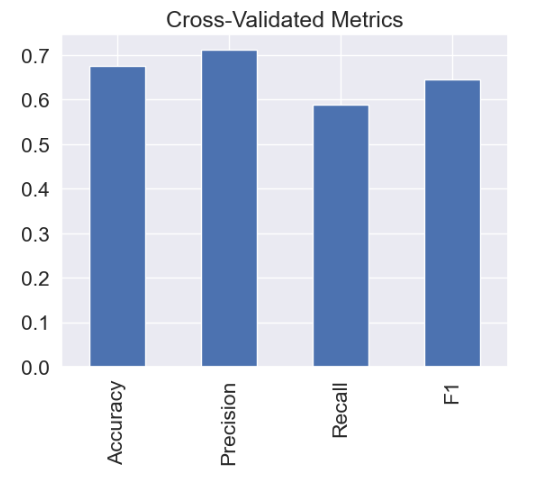


Figure 34: Cross validating matrix

**Feature importance**

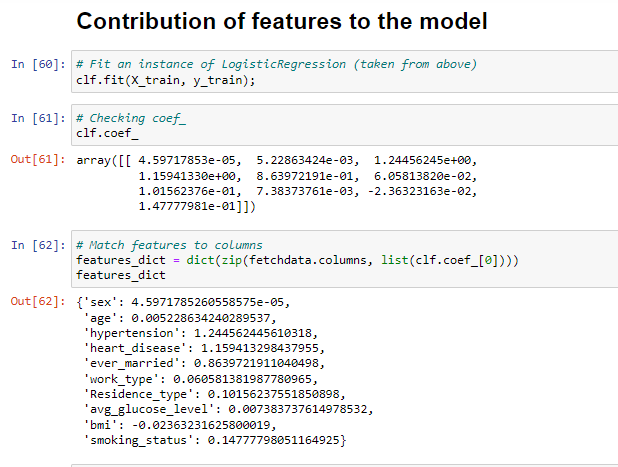
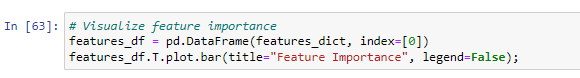


Figure 35: contribution of feature

The result above demonstrates how much each attribute affects a model's ability to determine if patterns in a sample of patients' health data indicate a greater likelihood of heart stroke or not. 

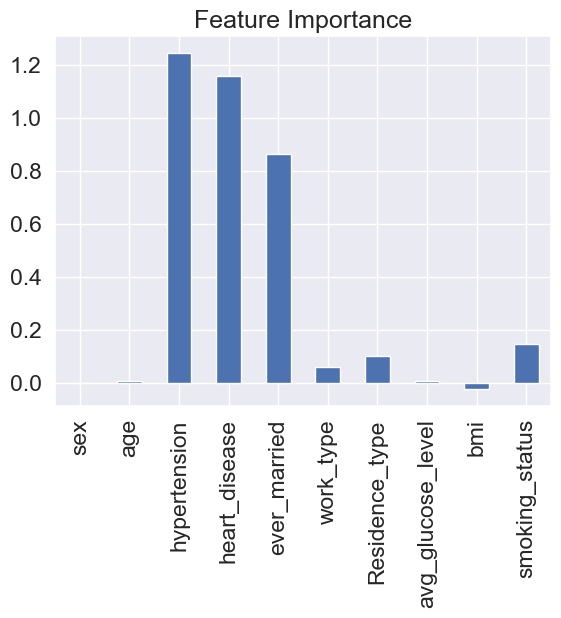
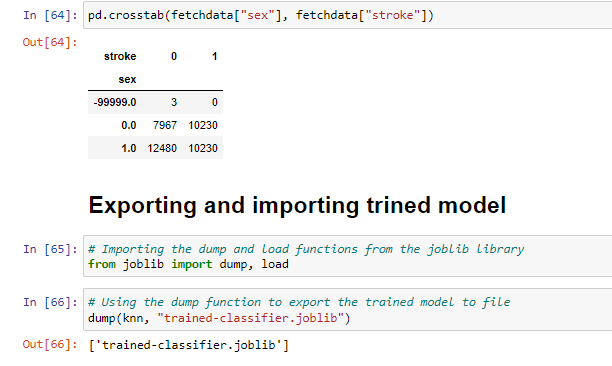


Figure 36: Feature evaluation

The larger values contributes more to model decision where the negative ones determines negative correlation. Hypertension has the most positive value. This means that the stroke chance is higher to the one with hypertension.



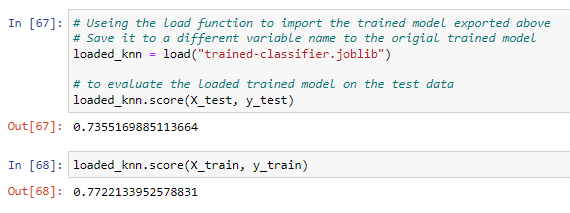


Figure 37: Exporting trained model

The train model was then exported to file using dump. The train score and test score of for the model using KNN is 0.735 and 0.772 respectively. The train score must always exceed the test score. So, this model was selected for heart stroke prediction for given data.

# Conclusion

Our study's findings demonstrated that the conservative mean feature selection strategy works effectively with the CHS dataset. However, as this technique assesses each feature separately, we accept that it might not perform as well on datasets with strongly linked characteristics. To solve this problem, we may think about reducing the amount of features first using a conservative mean feature engineering technique, followed by an L1 regularized feature selection approach (such L1 regularized logistic regression). In this work, we combined feature selection, prediction, and data imputation into a single, integrated machine learning technique.

Hence, performance of different machine learning models was compared in order to predict the likelihood of a patient having a heart stroke. The models tested included K-Nearest Neighbors (KNN), Logistic Regression, and NavBayes.

Among the three used, the results of the study showed that the KNN model had the highest accuracy in predicting heart stroke, with an average precision of 0.81 and an average recall of 0.59. The Logistic Regression model had a slightly lower accuracy, with an average precision of 0.67. The NavBayes model had the lowest accuracy, with an average precision of 0.5.

Overall, these results suggest that the KNN model is the most effective for predicting heart stroke, and it may be a useful tool for healthcare professionals in identifying patients at risk for this condition. However, further research is needed to confirm these findings and to explore the potential for using machine learning in the diagnosis and treatment of heart stroke.

The use of machine learning algorithms for medical applications is a complicated and quickly developing topic, thus more study is required to properly grasp its potential and limitations. Since these algorithms are not a replacement for professional medical judgment and skill, it is crucial to thoroughly assess their effectiveness and make sure they are used correctly. It's crucial to take into account the ethical ramifications of deploying machine learning algorithms in the healthcare industry, particularly concerns about fairness and privacy.

In conclusion, machine learning algorithms have showed potential as a method for detecting heart attacks, but further study is required to completely comprehend their strengths and weaknesses. Although these algorithms may lead to better healthcare.

The models selected was tuned using RandomSearchCV and GridSearchCV followed by accuracy, precision, recall, cross-validation and F1score. In general, finding good hyperparameter values can be a time-consuming process, and both random search and grid search are methods that can be used to automate this process and find good hyperparameter values more efficiently. The different hyper-parameters from log\_reg\_grid was passed for n-iter. Different metrics such as accuracy, precision, recall, or F1 score were used to evaluate the performance of a model.

Cross-validation (CV) divided a dataset into train and test sets, and using the train set to fit a model and the test set to evaluate the model. It is also found that the maximum KNN score on test data was 89.22. This determines how well the model is able to make predictions on a dataset that it has not seen before (i.e., the "test set").

**ePortfolio: -**

<https://canvas.sunderland.ac.uk/eportfolios/10031?verifier=Cs6BELT1K1lQjlryEkIp61oRgeJ8yiXpf091dXc5>

**Dataset**

<https://www.kaggle.com/datasets/prosperchuks/health-dataset?select=stroke_data.csv>

# References

Babu, D. et al., 2021. Gui based prediction of heart stroke using artificial intelligence. *Materials Today,* Volume 47, pp. 104-108.

Chefung, R., 2014. A Systematic Approach to the Definition of Stroke. *Austin Journal of Cerebrovascular Disease & stroke.*

Dritsas, E. & Trigka, M., 2022. Stroke Risk Prediction with Machine Learning Techniques. *Sensors,* 22(12), p. 4670.

Dritsas, E. & Trigka, M., 2022. Stroke Risk Prediction with Machine Learning Techniques. *Sensors,* 22(13), p. 4670.

Gavhane, A., Kokkula, G., Pandya, I. & Devadkar, K., 2018. Prediction of Heart Disease Using Machine Learning. *Second International Conference on Electronics, Communication and Aerospace Technology (ICECA),,* pp. 1275-1278.

J. J. Wang, Y. Z. X. a. Y. Q. C., 2019. Predicting stroke risk using machine learning algorithms: a systematic review and meta-analysis. *Stroke,* Volume 50, pp. 100-109.

Kok, J. N., Boers, E. J., Kosters, W. A. & Putten, P. v. d., 2009. ARTIFICIAL INTELLIGENCE: DEFINITION, TRENDS, TECHNIQUES, AND CASES. Volume 1, pp. 270-299.

Lackland, D. T. et al., 2012. Inclusion of Stroke in Cardiovascular Risk Prediction Instruments. *Science Volunteer,* 43(7), pp. 1998-2027.

Pradeepa, S. et al., 2020. DRFS: Detecting Risk Factor of Stroke Disease from Social Media Using Machine Learning Techniques. *Springer Link.*

Schwab, S. R., 2014. "Predicting stroke risk: current approaches and future directions. *Nature Reviews Neurology,* Volume 10, pp. 624-634.

Smith, F. B. et al., 1998. Haemostatic factors and prediction of ischaemic heart disease and stroke in claudicants. *British journal of haematology,* 100(4), pp. 758-763.

Wilson, P. W. et al., 2008. Prediction of First Events of Coronary Heart Disease and Stroke With Consideration of Adiposity. *Circulation,* 118(2), pp. 124-130.