**Abstract and Introduction**

Despite numerous advancements in the fields of Science and Medicine, stroke is considered the 2nd leading cause of death globally, accounting for roughly 11% of total deaths. This is the statistic provided by the World Health Organization (WHO). A stroke usually occurs when the blood supply to the brain is interrupted, preventing the brain tissue from getting essential nutrients and oxygen. There are multiple symptoms observed when people suffer from stroke such as blurred vision, headache, unstable walking, slurry speech, paralysis, etc. This is more applicable to people who are in their fifties. Very seldom do young people suffer from stroke unless it is from excessive heat which could result in sun stroke.

To understand the factors strongly influencing the attack of stroke, the dataset “healthcare-dataset-stroke-data.csv” was downloaded from Kaggle (<https://www.kaggle.com/fedesoriano/stroke-prediction-dataset>). Using this data, the various parameters that could estimate the likelihood of someone getting a stroke have been explored. The main objective here is to analyse the variables responsible for stroke detection of a patient for future prediction. First, the univariate analysis was done to understand the behaviour of each predictor. Bivariate analysis was then done among the numeric data to understand the strength of the relationship. Due to the presence of multiple predictors, bivariate analysis between numeric and categorical variables was not done. Once the initial Analysis was complete the pre-processing streps such as dealing with outliers, dealing with missing values, handling categorical data, and scaling of data were carried out. Post the pre-processing stages, predictive model modelling was carried out using Naïve Bayes classifier, Decision Tree Classifier, Random Forests and Support Vector Machines. Out of these model, Random Forests Classifiers displayed the highest accuracy of 96.3%. K = 10 Cross Fold Validation was incorporated on this model to gauge the overall accuracy. Post Model Building, tuning of the Hyper Parameters were carried out on the Random Forest Model to bring out best parameters.

The Stroke Prediction dataset contains 5110 observations and 12 attributes as below.

1) id: unique identifier  
2) gender: "Male", "Female" or "Other"  
3) age: age of the patient  
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension  
5) heart\_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease  
6) ever\_married: "No" or "Yes"  
7) work\_type: "children", "Govt\_jov", "Never\_worked", "Private" or "Self-employed"  
8) Residence\_type: "Rural" or "Urban"  
9) avg\_glucose\_level: average glucose level in blood  
10) bmi: body mass index  
11) smoking\_status: "formerly smoked", "never smoked", "smokes" or "Unknown"\*  
12) stroke: 1 if the patient had a stroke or 0 if not

Note: "Unknown" in smoking\_status means that the information is unavailable for this patient.

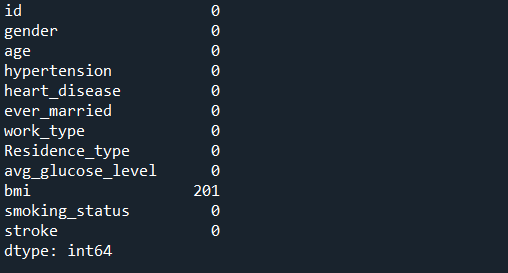
The last column represents the response, while the rest are the predictors.

The columns 1,3,9,10 hold the numerical/continuous data.

The data in columns 4,5,6,12 is binary in nature. There is no mention of ordinality in any of the columns.

Columns 2,7,8,11 hold the nominal data which is categorical by default.

There are totally 201 missing values and all of them come from the ‘bmi’ field as shown below.



The target field is the ‘stroke’ variable which holds binary categorical information. If the stroke value is equal to 0, then the patient has not suffered from a stroke; else if it is equal to 1, then the patient has suffered from it.

The motivation here is to build analytical classification models that can predict the likelihood of a person getting a stroke attack. This can be done by thoroughly identifying the important factors that influence the response variable. If the models yield high accuracy, they can be used in the medical domains which can help the doctors in their diagnostics. This will have a dominos effect overall and can help lots of people in undergoing treatment. If this is made into a Mobile App, then it could alert people from getting strokes.

There is no explicit mention about previous studies or papers that have worked on this dataset in the Kaggle Link. However, around 300 people have written code for building these models, though their intent behind it is not known.

**Research**

Due to the complexity in the data, focus on a specific stage of pre-processing or model building was difficult and hence it was carried out in chunks.

Pre-processing steps such as dealing with outliers, dealing with missing values, handling categorical data, and scaling of data were carried out extensively.

However, steps such as handling imbalance, feature selection and dimensionality reduction were not applied due to the dataset size. For performing feature selection there should be a minimum of 15-20 columns available in the dataset and for dimensionality reduction, the relationship between the variables plays a significant role. One of the major steps involved in handling data imbalance is inserting new rows in the original dataset. Due to lack of similar datasets, this was not possible and hence, sampling techniques were used as compensation.

Also, the number of records were quite less (around 5000) which made the pre-processing a bit tricky. Hence, a minor fraction of research was put into data pre-processing and most of the effort was put into hyper parameter tuning and optimization.

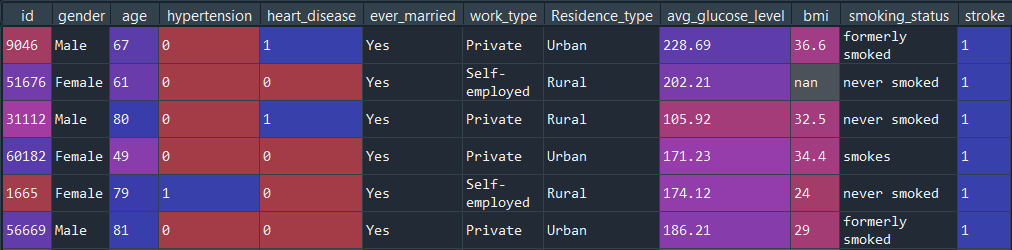
First, the model building was carried out to find the most suited model. Once the best model was selected, K = 10 cross fold validation was performed for evaluating the model performance. Post this, Grid Search CV method was used to find the best fit and model parameters. To validate this, the initial analysis of the dataset and the understanding from it played a major role in determining the best model parameters. The goodness of fit was then finalized by using this model on the training dataset and assessing the performance scores. Tuning can also be considered as the addition of multiple layers in the dataset until the maximum accuracy saturates.

This was implemented using Python’s Sklearn library which contains an array of ML algorithms and models.

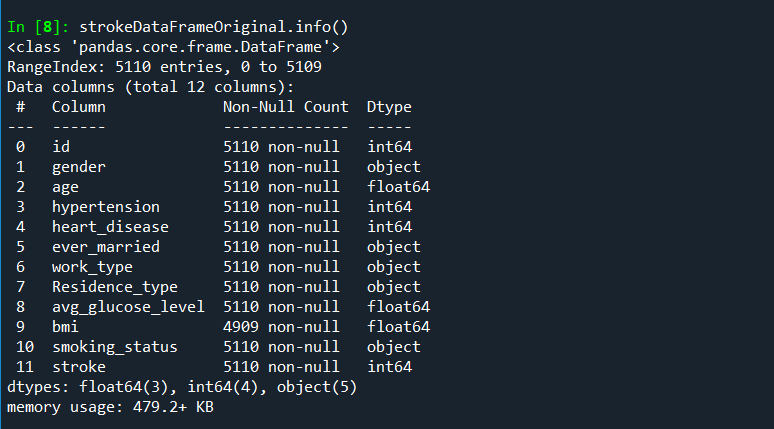
**Methodology**

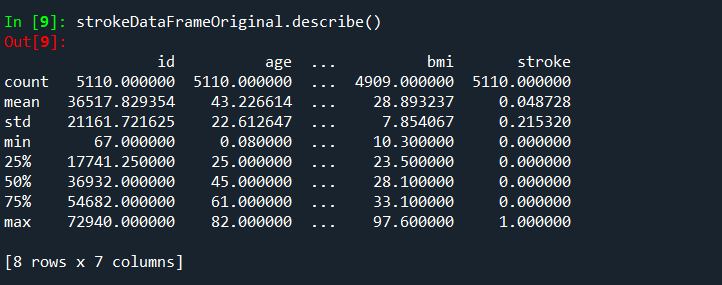
Before we start with pre-processing and Exploratory Data Analysis (EDA) we need to identify the data that has categorical information. In the dataset, the fields "hypertension", "heart\_disease" and "stroke" are binary categorical data despite holding numeric values. These fields must be converted to string type.

**Sample Input Data**

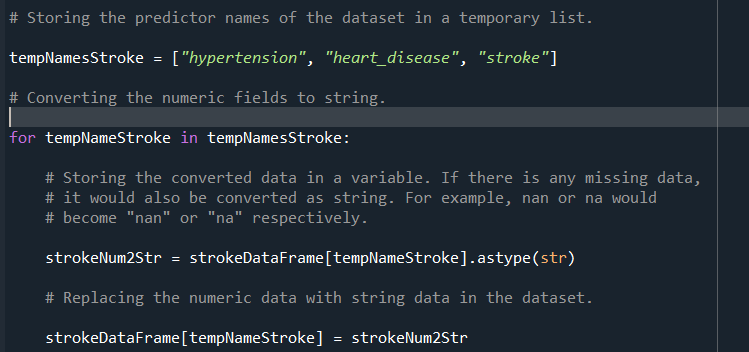


**Summary and Structure of Input Data**





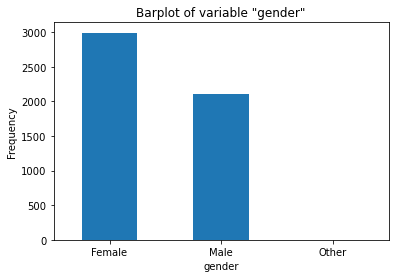
**Sample Input Data After Transformation**



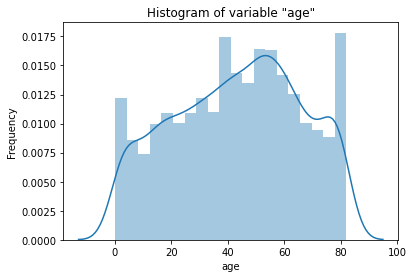
1. **Exploratory data analysis (EDA)**

EDA is done to understand the relationship between the variables. This gives us insights on the data which can be used for building predictive models. This can be done by plotting Histograms, Boxplots and Bar plots for understanding the distribution of data.

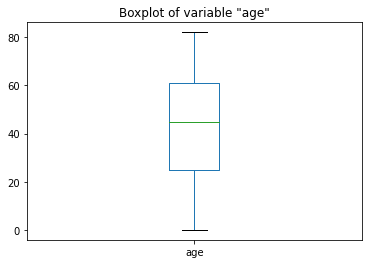
* 1. **Univariate Analysis:-** Univariate plots are created to understand the of each field.



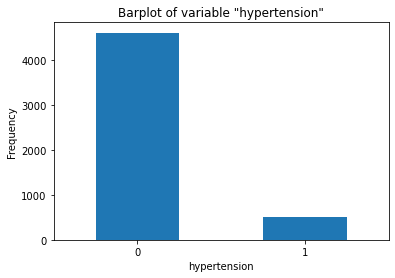
From the barplot of the "gender" variable, it is observed that majority of the patients are female while the rest are male with one patient's gender as 'other'.

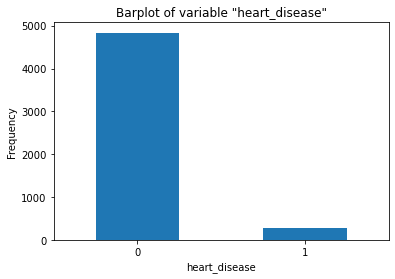


The histogram of the "age" variable is quite resemblant to a normal distribution with a minor negative skew.

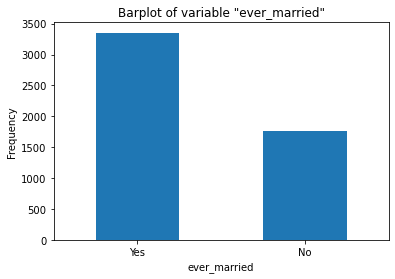


The boxplot shows that the age has zero outliers.

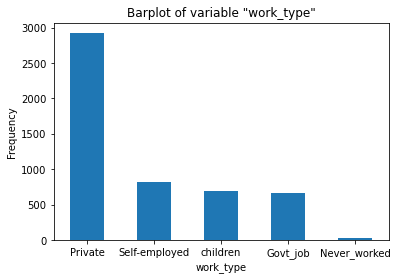




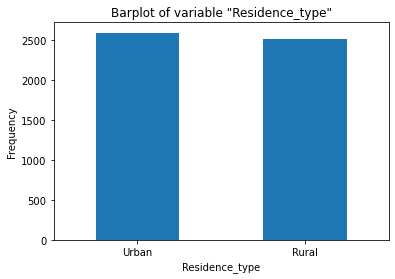
From the barplots of the "hypertension" and "heart\_disease" variables, it is observed that most of the patients do not suffer from both hypertension as well as any heart disease. A small percentage of people suffer from both.



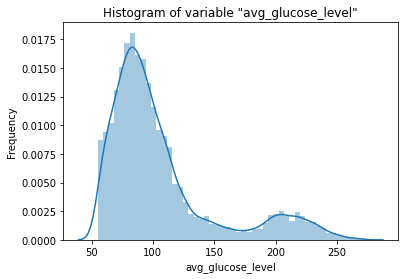
From the barplot of the "ever\_married" variable, it is observed that most of the patients are married except for a small fraction of people.

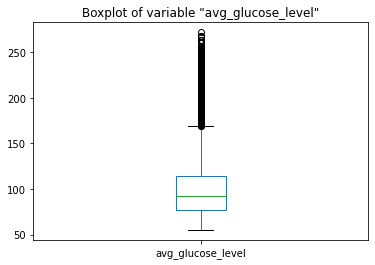


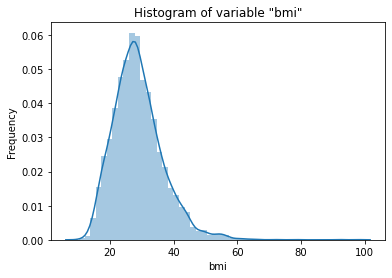
From the barplot of the "work\_type" variable, it is observed that a high percentage of the people work in private organizations. Around 16% of the population are self-employed workers. Roughly 13% of the population consist of children and people working for the goverment. Only 4% of the patients have never worked.

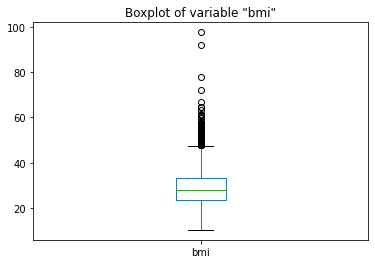


From the barplot of the "Residence\_type" variable, it is observed that the percentage of people residing in rural areas is almost equal to the percentage of people residing in urban localities.

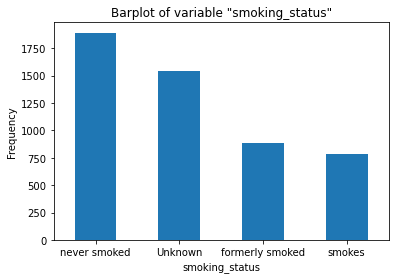




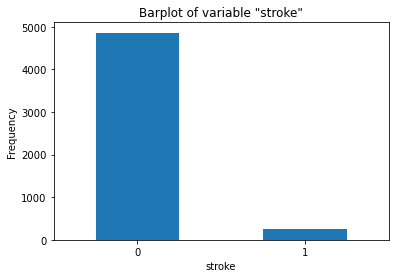




The histograms of the "avg\_glucose\_level" and "bmi" variables are quite resemblant to a normal distribution with a strong positive skew. The boxplots show that both the glucose level and bmi have multiple outliers which needs to be handled in the pre-processing stage.



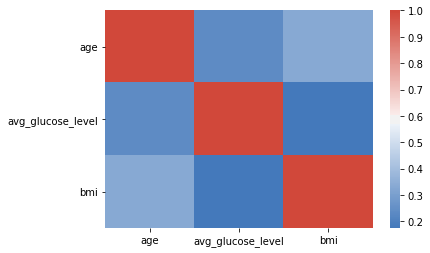
From the barplot of the "smoking\_status" variable, it is observed that a high percentage (37%) of the people have never smoked. Around 17% of the patients used to smoke before quitting. Roughly 16% of the people still smoke while it is still unknown whether the remainder of the people smoke or not.



From the barplot of the "stroke" variable, it is observed that a high percentage (95%) of the people have not experienced any stroke. Only 5% of the patients have suffered from stroke.

* 1. **Bivariate Analysis.**

A Correlation Matrix can be used to understand the bivariate relationship among the numerical fields.



From the Correlation Matrix, we can see that there is a weak relationship between the variables 'age' and 'avg\_glucose\_level' and a slightly weaker relationship between the pairs 'age' and 'bmi'. Compared to these relationships, the relationship between variables 'avg\_glucose\_level' and 'bmi' is slightly stronger.

Due to the presence of multiple predictors, bivariate analysis between numeric and categorical variables was not done.

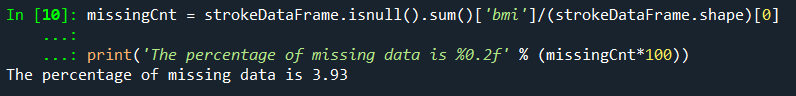
1. **Data Pre-Processing**

Before model building, pre-processing is the most important step. This involves dealing with outliers, dealing with missing values, handling categorical data, scaling data, handling imbalance, feature selection and dimensionality reduction. Due to the nature of the dataset, steps such as handling imbalance, feature selection and dimensionality reduction were not applied.

* 1. **Dealing with missing values:-**

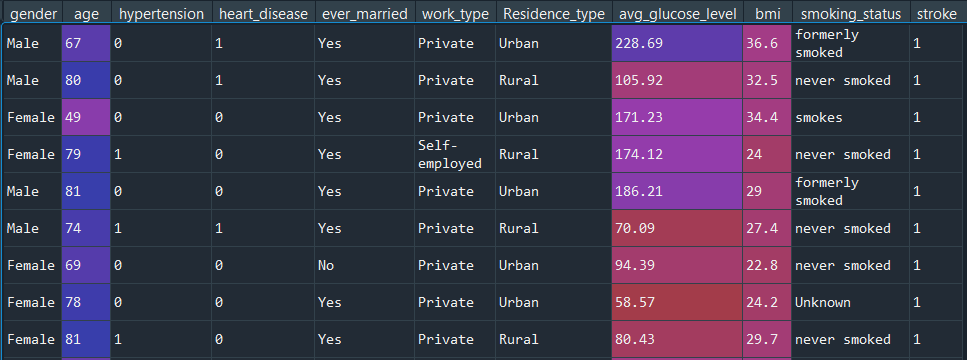
It is known that only the 'bmi' field contains missing values.

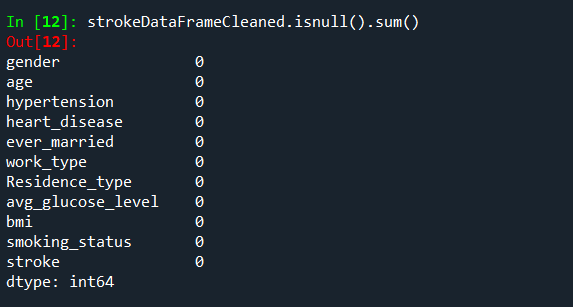
Only 3.93% percentage of the overall data is found missing.



Since the percentage is quite less (4%), there would not be much impact if those rows are removed.

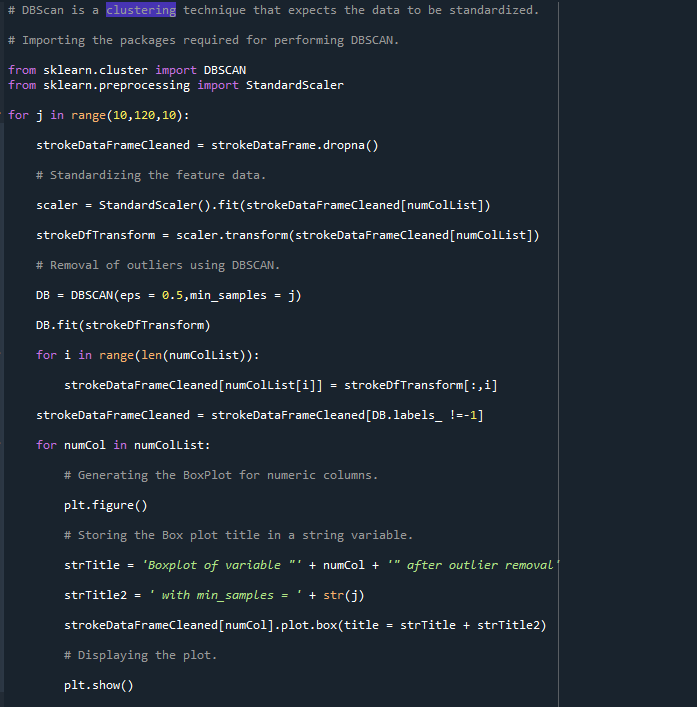
**Input Data after removal of missing values.**





* 1. **Dealing with outliers:-**

Outliers are data that differ significantly from other data in a sample. Outliers skew the data distributions and impacts the basic statistical measures and can be responsible for underperformance of certain algorithms. Some ML algorithms are relatively robust to outliers. Other ML algorithms (such as multiple linear regression) are much more sensitive to outliers. The boxplots of the "avg\_glucose\_level" and "bmi" variables show that both the glucose level and bmi have multiple outliers which needs to be handled in the preprocessing stage. Since there are two variables possessing outliers, it is better to use a common clustering technique called DBScan. DBScan relies on the idea that clusters are dense, so it starts exploring the data space in all directions and marks a cluster boundary when the density of points decreases. Areas of data space with insufficient density of points are just considered to be outliers or noise. DBScan is a clustering technique that expects the data to be standardized.



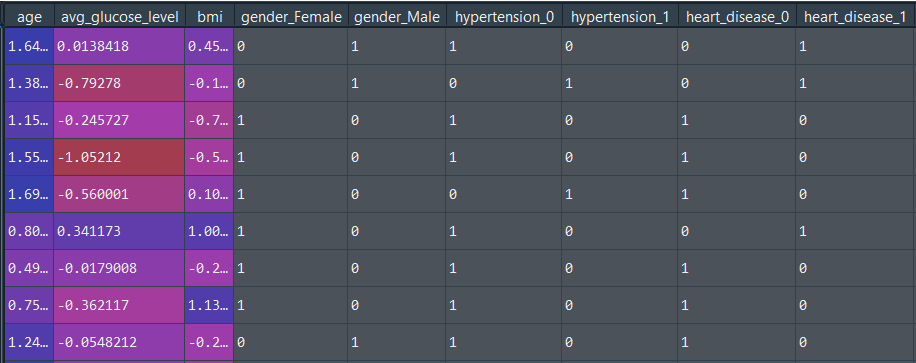
The 'min\_samples' refers to the number of samples (or total weight) in a neighborhood for a point to be considered as a core point. This includes the point itself. An array of integers ranging from 10-110 in multiples of 10 are passed as input 'min\_samples'.

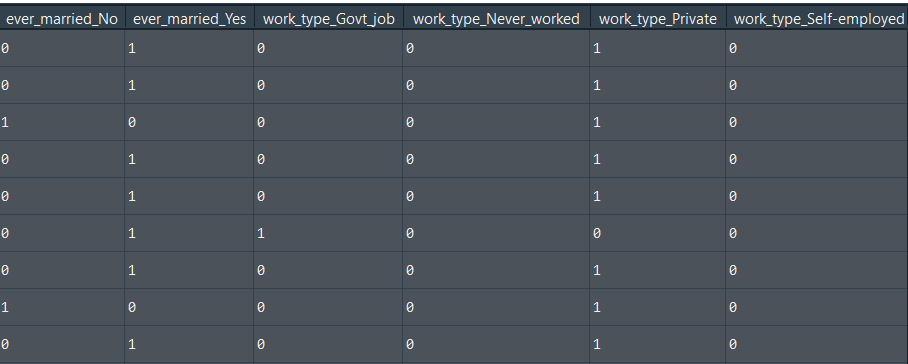
It is observed that as 'min\_samples' increases, the samples get closer which removes the outliers. The outliers are removed when min\_samples = 110. This is not saved in this report due to the multitude of graphs generated iteratively.

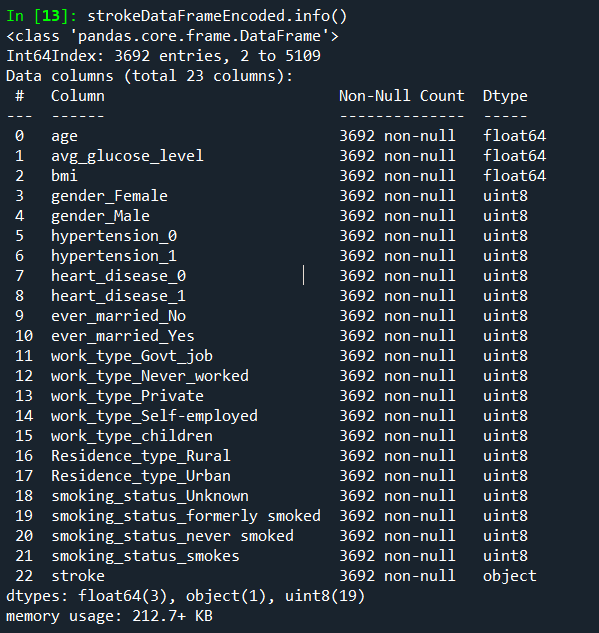
* 1. **Handling Categorical Data:-**

For building models, it is essential to convert categorical data into numeric data. Categorical data can be either ordinal or nominal. Ordinal data have logical ordering and there is no automatic mechanism for encoding this. Nominal data have no inherent ordering and can be encoded easily using algorithms such as 'one-hot encoding'. This technique will create new dummy features for each unique value in the nominal feature column. Here, we don’t have any ordinal data hence ordinal encoding is not carried out.

**Sample Data After Encoding**





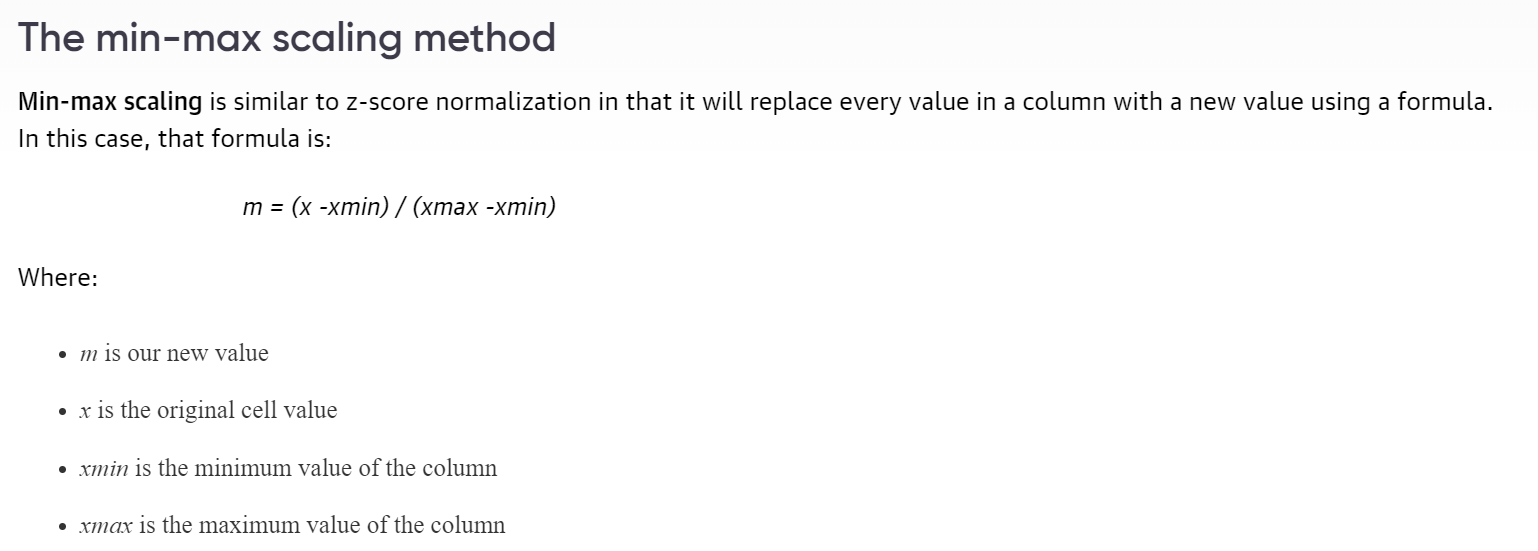


There is an increase in the number of predictors due to the levels in each categorical field. This becomes a major drawback when there are multiple levels in a single field.

* 1. **Scaling Data:-**

Scaling is done to standardize the range of data. This is a necessary aspect of pre-processing while building machine learning algorithms.

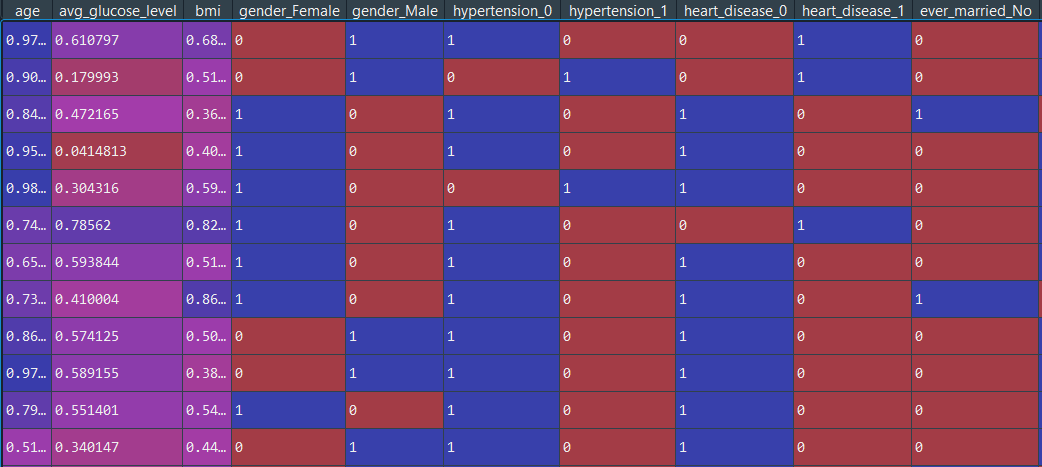
Here, minmax scaling has been performed.



This image has been taken from the below link:

<https://www.oreilly.com/library/view/feature-engineering-made/9781787287600/aa5580ee-6fb7-4ac2-a1fe-369d95b70168.xhtml>

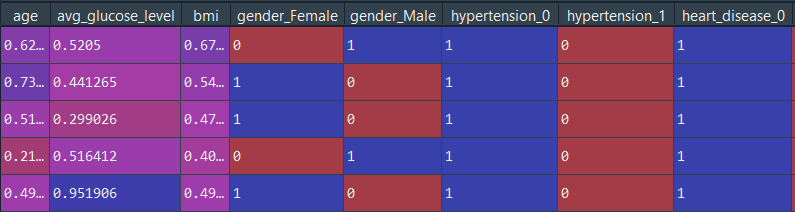
**Sample Scaled Data (restricted due to multiple columns)**



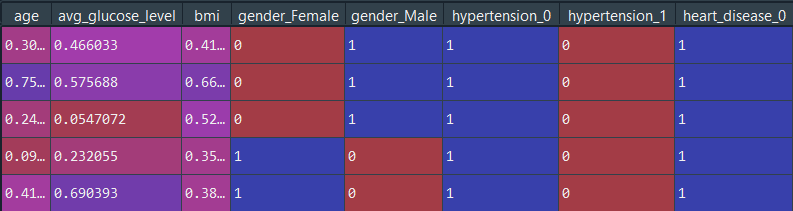
1. **Model Building**

The next step after pre-processing is model building. The models used in the initial building stage were Naïve Bayes classifier, Decision Tree Classifier, Random Forests and Support Vector Machines classifiers. Before model building, the data was is split into train and test in the ratio 70%:30%. Here 70% of the data is considered training and the remainder test.

**Sample Training Data**



**Sample Testing Data**

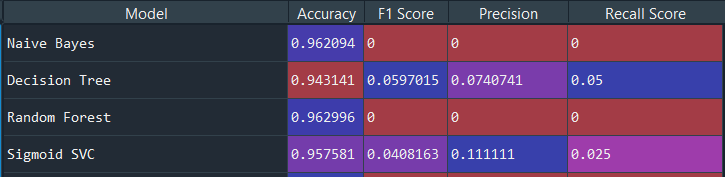


The Naïve Bayes model was built using default arguments. The Decision Tree Classifier model was built using the entropy splitting criterion. The similar thing applies to the Random Forest model with the number of estimators = 1000. And finally, a Support Vector Machine (SVM) was built using the Sigmoid Kernel. Parameters such as confusion matrix, accuracy score, f1 score, precision score and recall score were examined for each of the best performing models. K = 10 cross fold validation was performed for evaluating the model performance and post that, Grid Search CV method was used to find the best fit and model parameters. This was the hyper-parameter optimization technique employed.

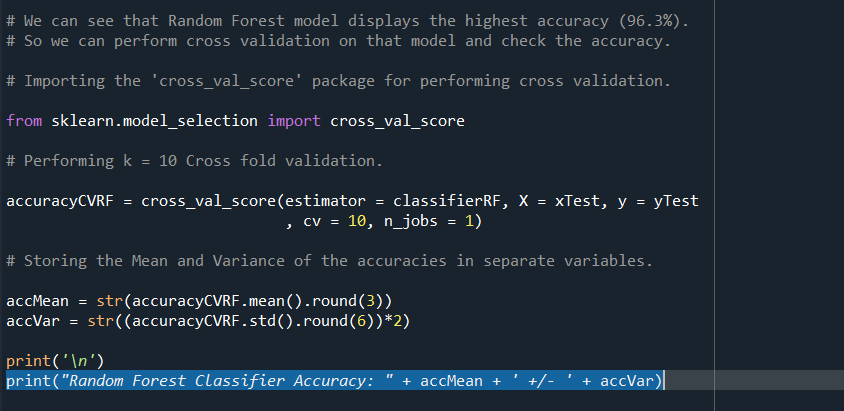
GridSearchCV implements a "fit" and a "score" method. It also implements "score\_samples", "predict", "predict\_proba", "decision\_function", "transform" and "inverse\_transform" if they are implemented in the estimator used. The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

**Evaluation and Conclusions**

The results of the initial model building are shown below.



The Random Forest classifier shows the best accuracy (96.3%) followed by Naïve Bayes Classifier (96.2%). This is due to nominal feature scaling which resulted in the creation of multiple columns and since most of them held binary values, this favoured the random forest model. Naïve Bayes algorithm also works wonders when most of the predictors hold binary values as it helps in simplifying the probability calculations. There is not much work required when most of the data are numeric. For model evaluation, K = 10 cross fold validation was used. Before performing cross validation, the model must be trained manually. k-cross fold validation is the process in which the training data is randomly split into 'k' folds without replacement, where 'k-1' folds are used for model training and one-fold is used for testing. This algorithm is applied k-times to obtain 'k' models and performance estimates. The models are independent to each other and hence, the average performance is calculated to obtain a performance estimate. This is extremely reliable while processing unseen data and balances the variance-bias trade off.



The Random Forest Classifier Accuracy: 0.962 +/- 0.007146.

There is a minor decrease in accuracy after performing k = 10 CV.

For hyper parameter optimization, two types of RF models were created using GridSearchCV: one using entropy as splitting criterion and the other using Gini Indexing.

For the model with entropy as splitting criterion, the number of estimators took the values 10, 100 and 150 with max depth as 5 or None and optional usage of Bootstrapping. The accuracy and best parameters are given below.

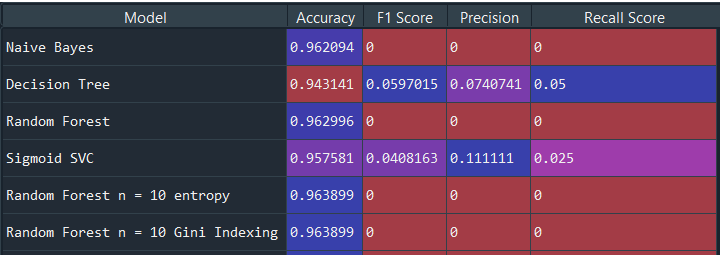
**Random Forest Best Accuracy: 0.9698153302804465**

**Random Forest Best Parameters: {'bootstrap': True, 'criterion': 'entropy', 'max\_depth': 5, 'n\_estimators': 10}**

For the model with Gini Indexing as splitting criterion, the number of estimators took the values 10, 100 and 150 with max depth as 5 or None and optional usage of Bootstrapping. The accuracy and best parameters are given below.

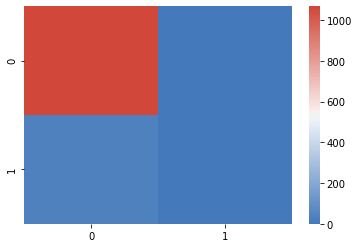
**Random Forest Best Accuracy: 0.9698153302804465**

**Random Forest Best Parameters: {'bootstrap': True, 'criterion': 'gini', 'max\_depth': 5, 'n\_estimators': 10}**



There is a slight increase in model accuracy due to hyper parameter tuning.

**Confusion Matrix of the Grid Search RF Model.**



From the heat map we can say that the True Positive (TP) value is much higher than the test of values and hence the accuracy is quite high. We can conclude saying that the Random Forest Classifier is the most ideal one when comes to predicting strokes.

For future work, it would be better to explore the pre-processing techniques after learning the mathematical concepts behind them in detail. This would help in the model building aspect as well.