Stroke Prediction

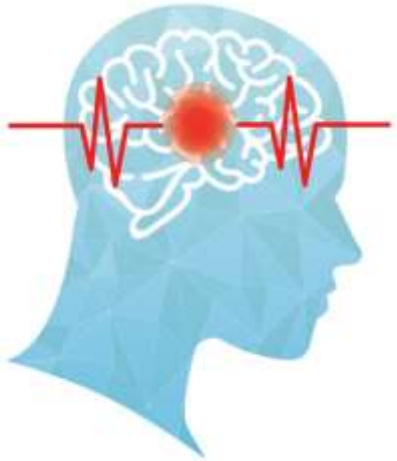
BBM469 Data Intensive Applications Laboratory

Data Science Capstone Project

Burak Yılmaz, Mehmet Sezer

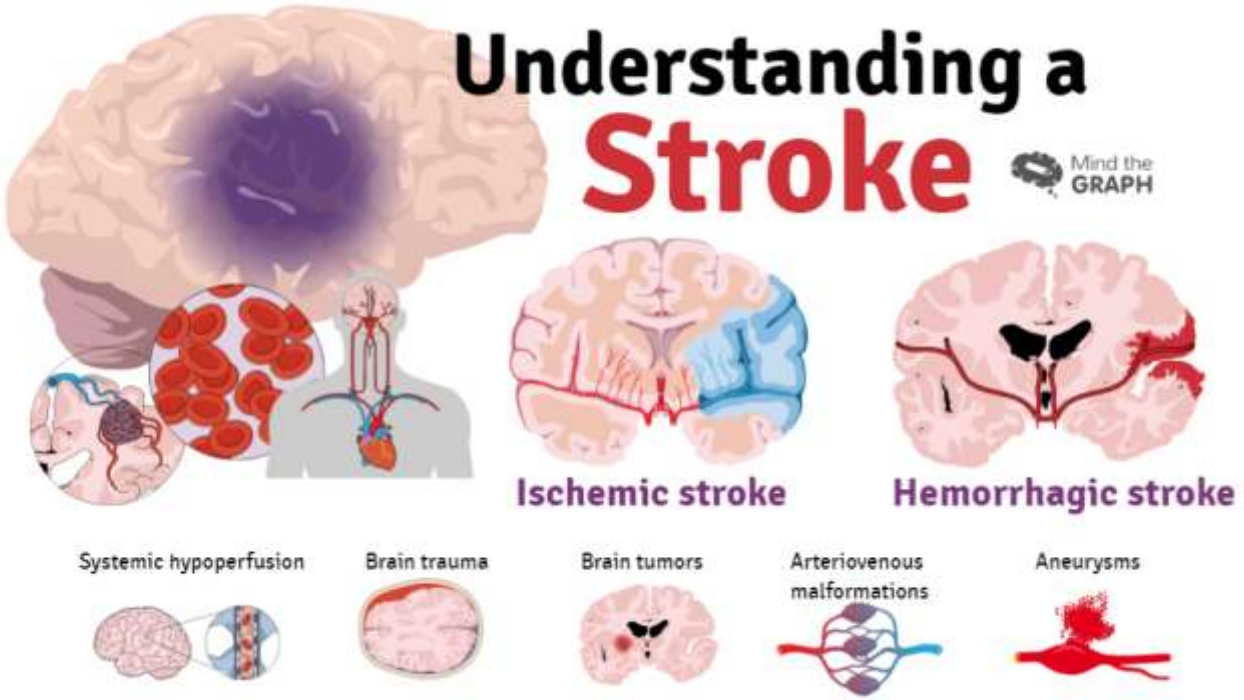
Problem

- First let's start with what is stroke?



A stroke is a medical condition in which poor blood flow to the brain causes cell death. There are two main types of stroke: ischemic, due to lack of blood flow, and hemorrhagic, due to bleeding. Both cause parts of the brain to stop functioning properly. Signs and symptoms of a stroke may include an inability to move or feel on one side of the body, problems understanding or speaking, dizziness, or loss of vision to one side. Signs and symptoms often appear soon after the stroke has occurred. If symptoms last less than one or two hours, the stroke is a transient ischemic attack (TIA), also called a mini-stroke. A hemorrhagic stroke may also be associated with a severe headache. The symptoms of a stroke can be permanent. Long-term complications may include pneumonia and loss of bladder control.

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.This dataset is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relevant information about the patient.



Data Understanding

Dataset Information

The csv contains data related to patients who may have heart disease and various attributes which determine that :

id: unique identifier

gender: "Male", "Female" or "Other«

age: age of the patient

hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension.

heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease.

ever_married: "No" or "Yes«

work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed".

Residence_type: "Rural" or "Urban«.

avg_glucose_level: average glucose level in blood.

bmi: body mass index.

smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown".

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.

Let’s look at what our data looks like

id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked	0
44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked	0
19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0
44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown	0

Since our data is consists of mixed data types, for the numerical ones we can examine the descriptive statics information.

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

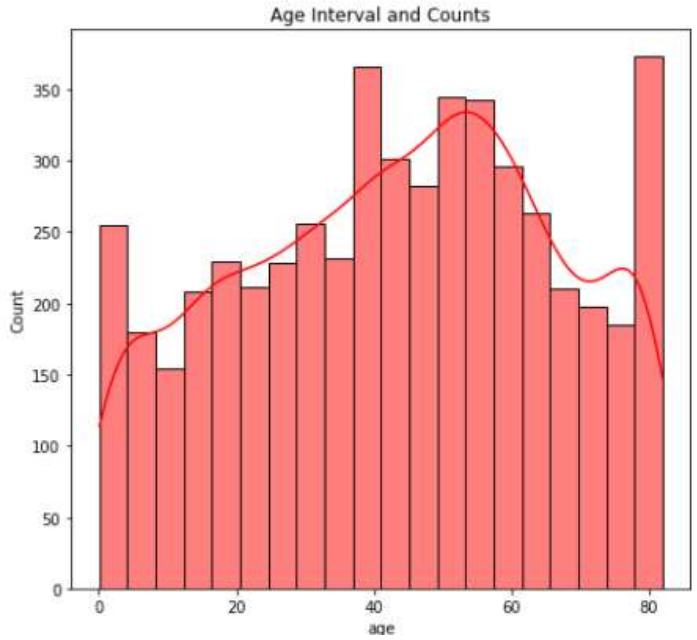
Data Visualization

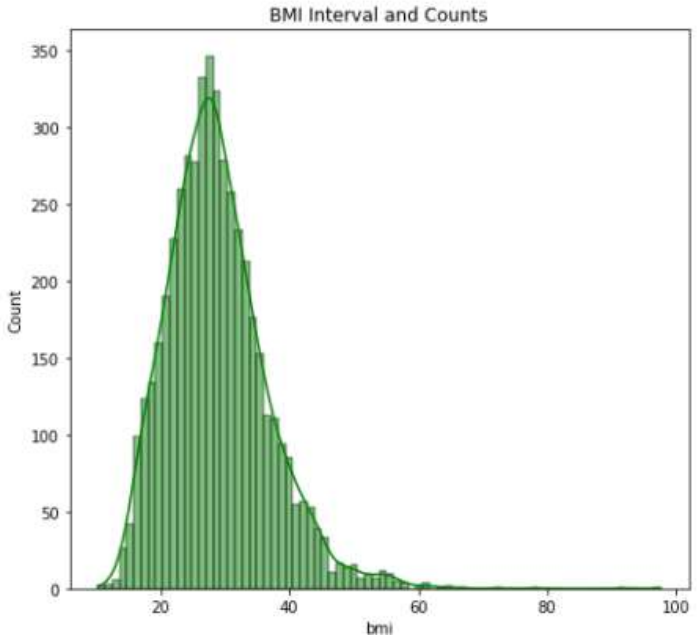
By looking at the graphs we can say,

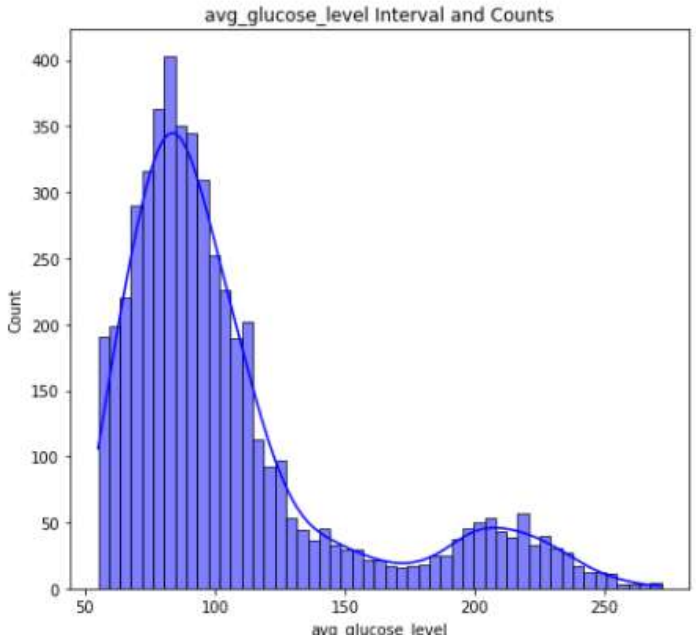
Age distribution is normal distribution.

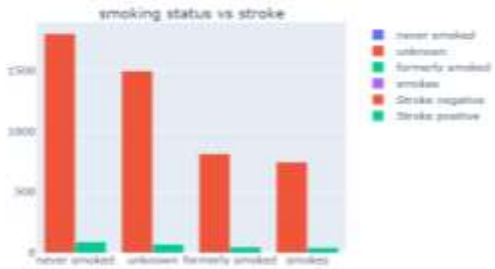
Bmi field may have outliers and most of the values gathered between 15-55. The outliers make the distribution curve highly skewed towards right.

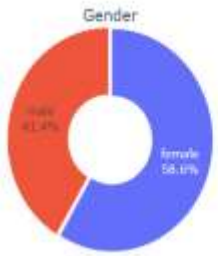
Avg glucose level distribution looks like semi normal distribution and most of the data are below 150, moreover like bmi field it has outliers.

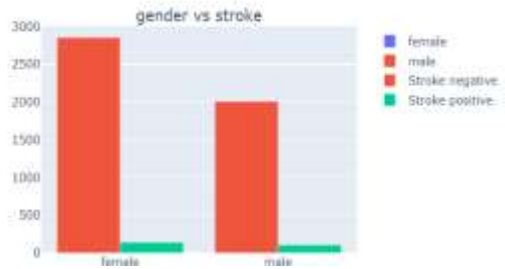


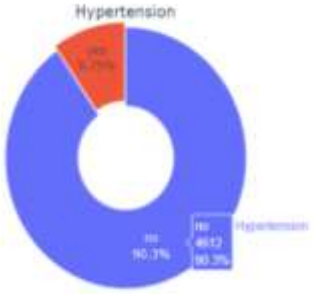


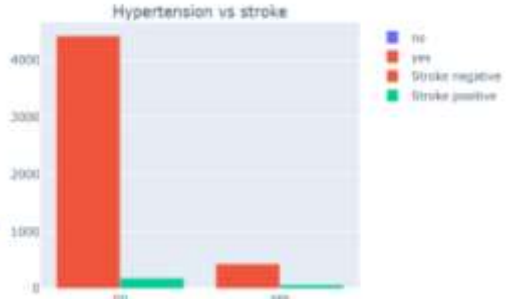


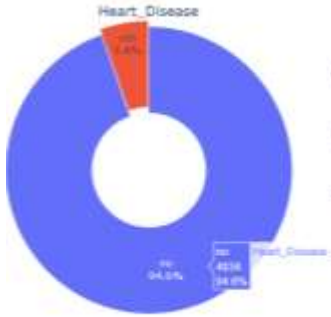


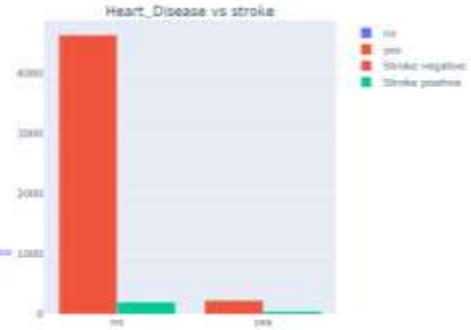


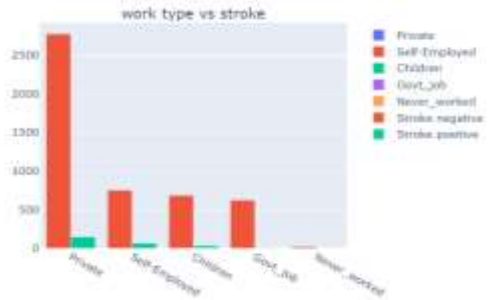


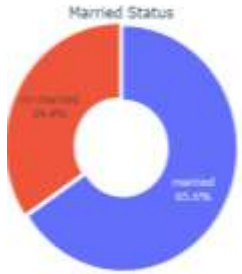


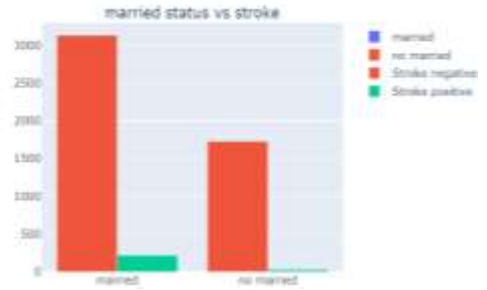








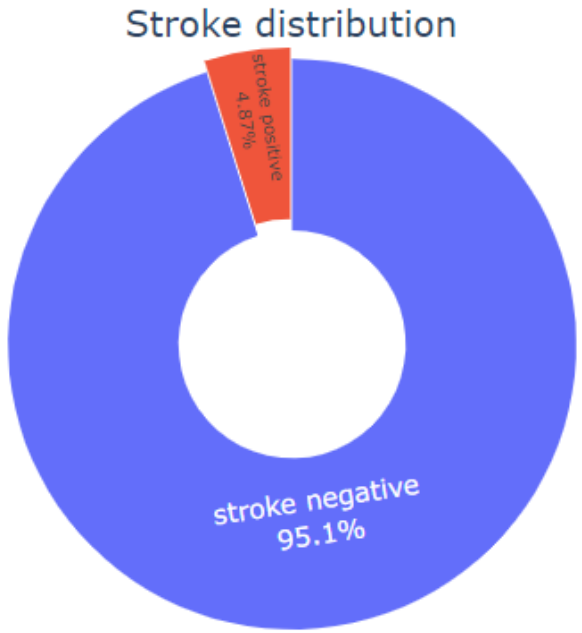


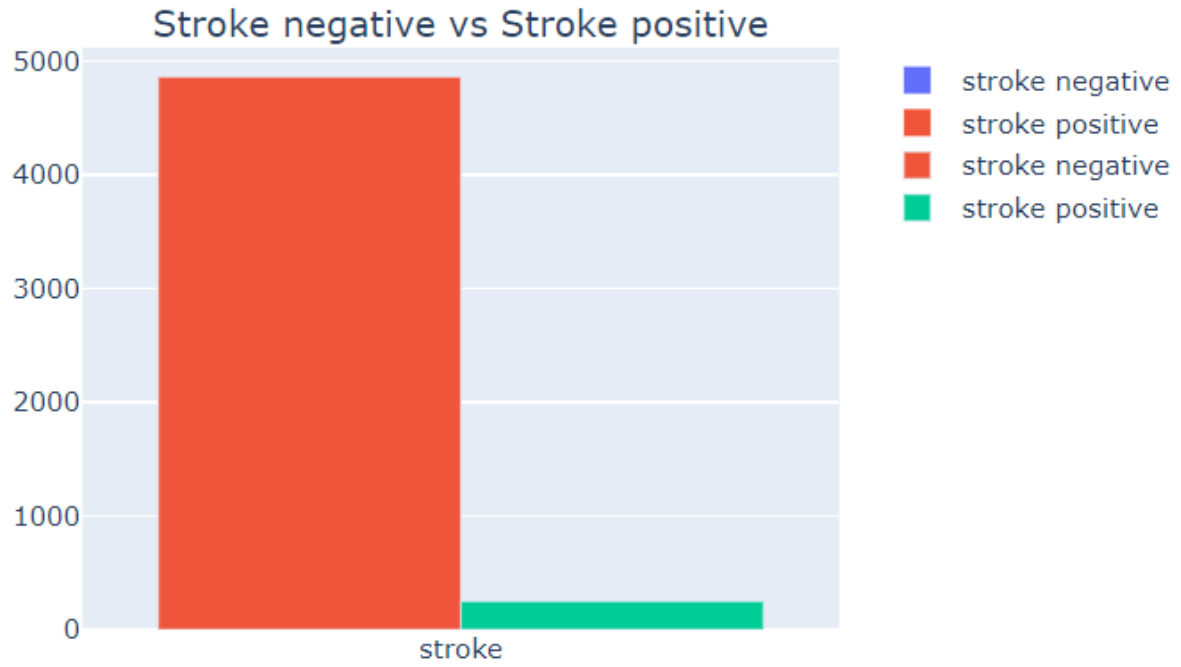


We examine the relationship between categorical features and stroke.

In conclusion for some features we can say the proportion of people with stroke positive is about the same among other types of that feature. Therefore that feature may not be important for us.

Our data is highly imbalanced. We can see that we have mostly stroke negative cases.

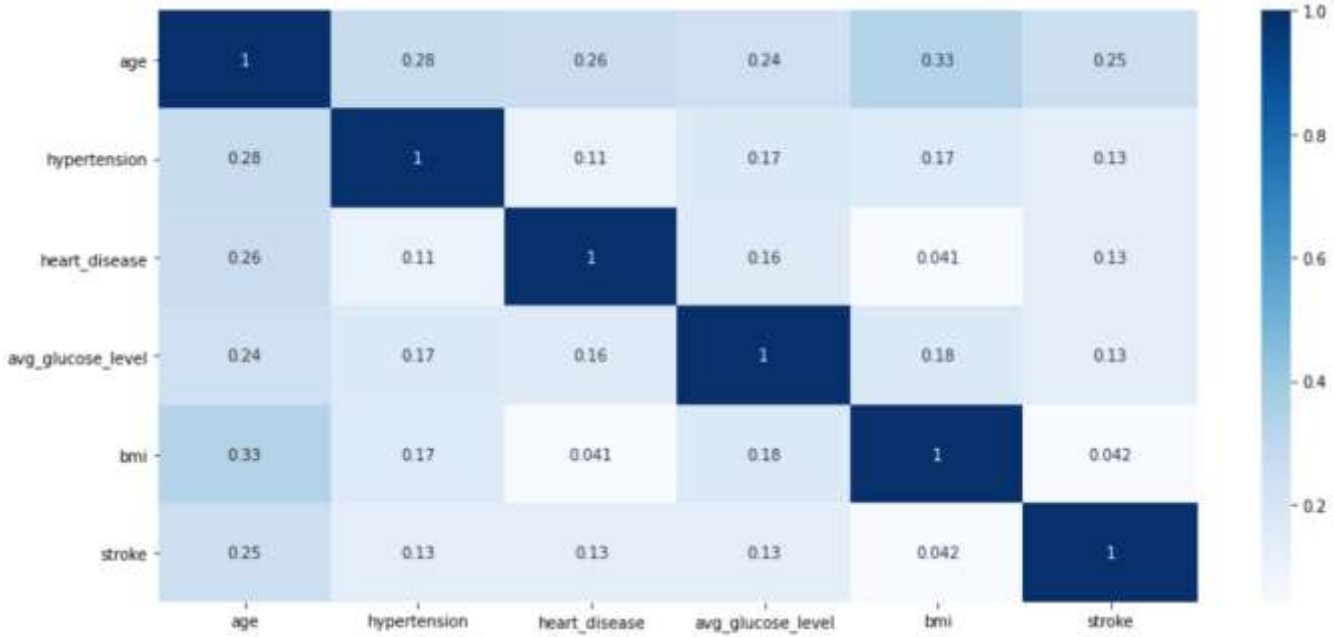




Data Preparation

Correlation Matrix

- A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables
- From the correlation heatmap below, we cannot see any strong correlation between any two feature. In conclusion, based on the correlation map we cannot eliminate any feature.



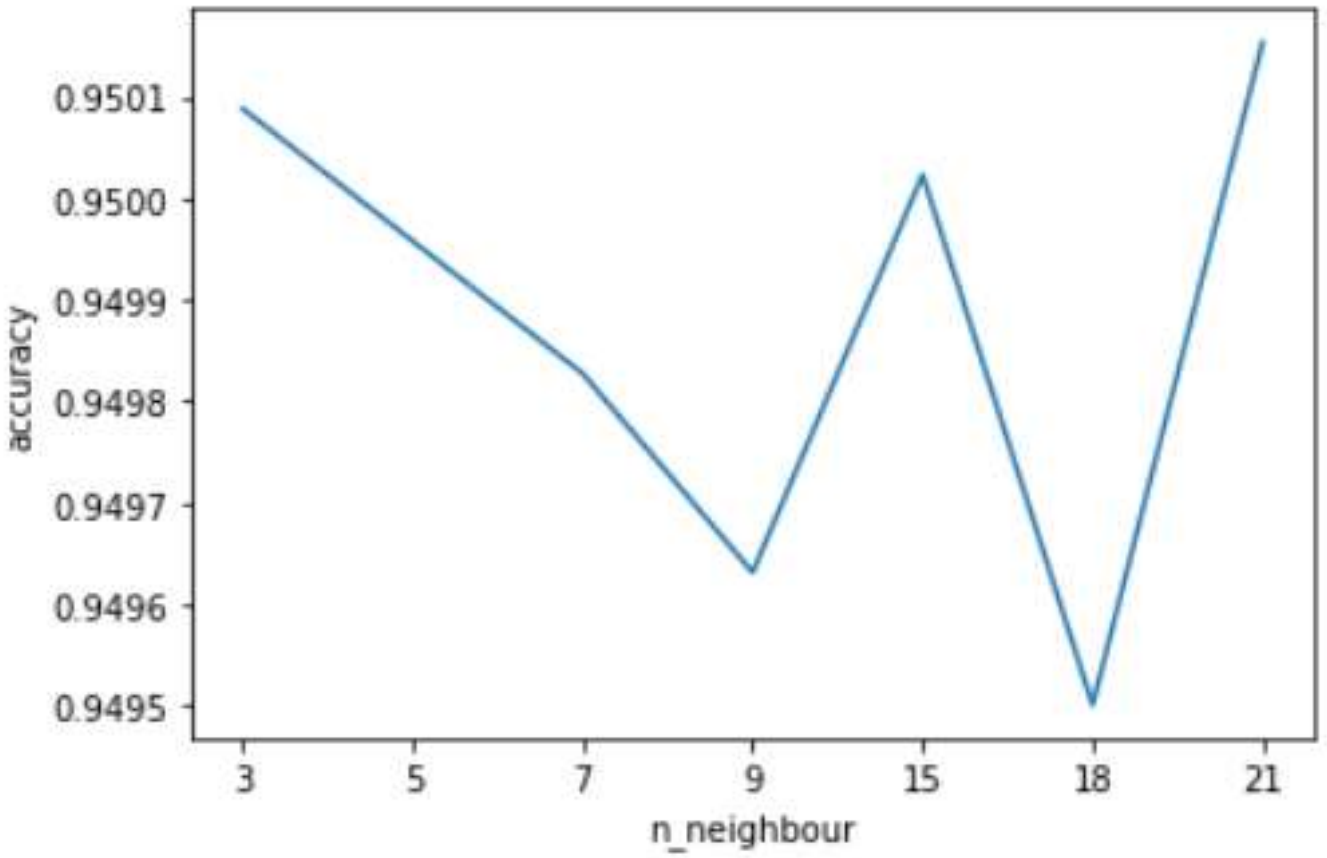
Conversion of Data Types

Machine learning algorithms cannot process categorical variables and can give poor results. In this case, we need to turn our column of labels into separate columns of 0s and 1s. For that reason, we changed all columns to numerical types. If the column have only two options(such as gender) we mapped it directly into two options 0 and 1. In other case if the column has more than two options(such as smoking_status), we use dummy variables.

```
gender               object
age                 float64
hypertension          int64
heart_disease         int64
ever_married         object
work_type            object
Residence_type       object
avg_glucose_level   float64
bmi                 float64
smoking_status       object
stroke                int64
```

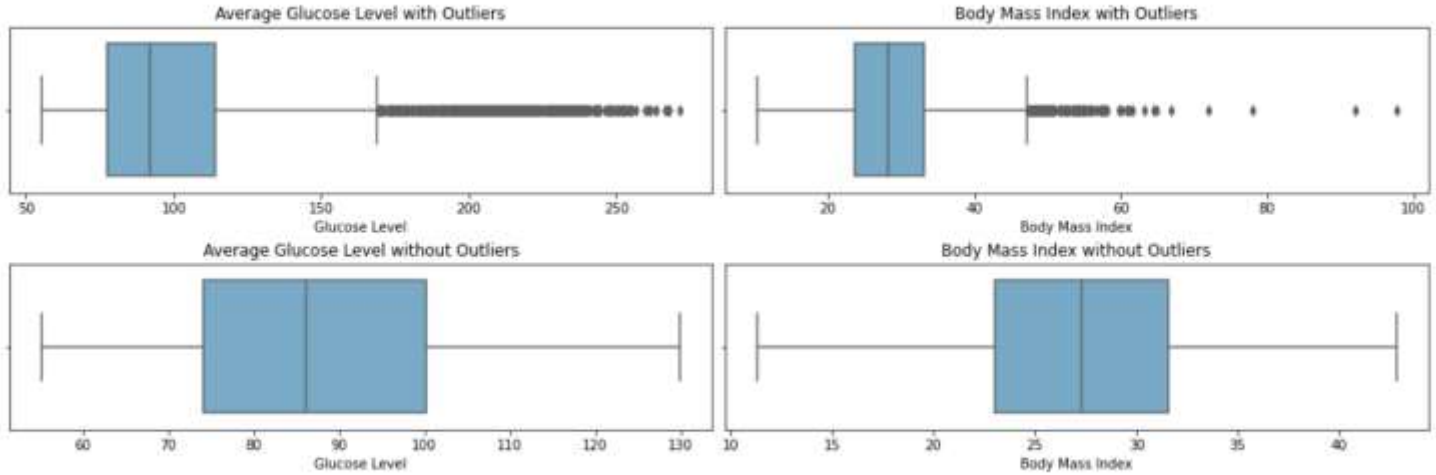
```
gender                           int64
age                            float64
hypertension                     int64
heart_disease                    int64
ever_married                     int64
Residence_type                   int64
avg_glucose_level              float64
bmi                            float64
stroke                           int64
work_type_Never_worked           uint8
work_type_Private                uint8
work_type_Self-employed          uint8
work_type_children               uint8
smoking_status_formerly smoked   uint8
smoking_status_never smoked      uint8
smoking_status_smokes            uint8
```

Filling Null and Empty Fields

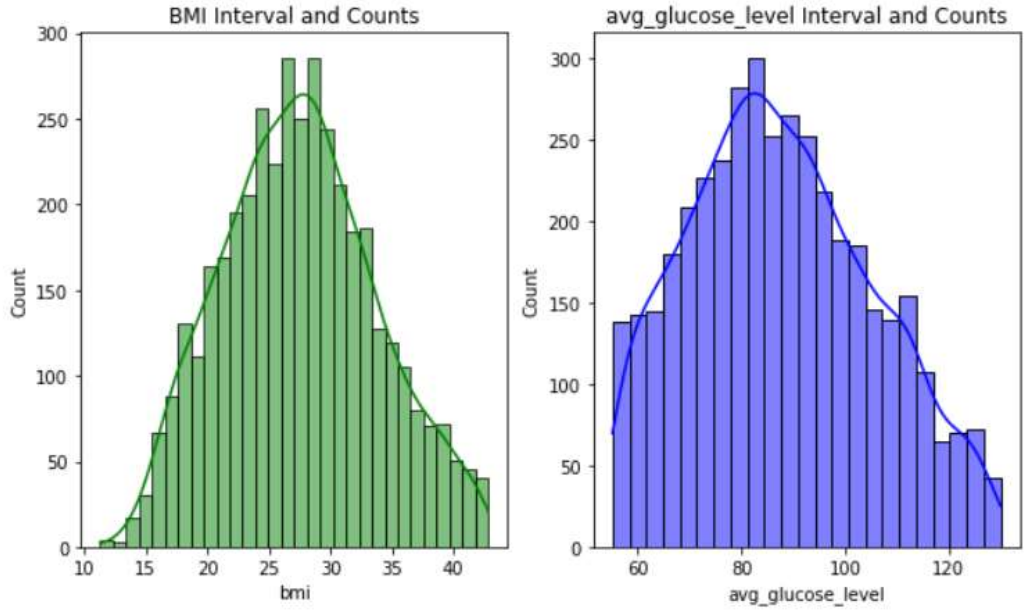


- We use knn imputer for filling null and empty values. What basically knn imputer does is, it imputates for completing missing values using k-Nearest Neighbors. Each sample's missing values are imputed using the mean value from n_neighbors nearest neighbors found in the training set. Two samples are close if the features that neither is missing are close.
- By looking at the graph below we can find best n_neighbor's parameter. We will use this value for knn imputer algorithm.

Dealing with outliers and anomalies

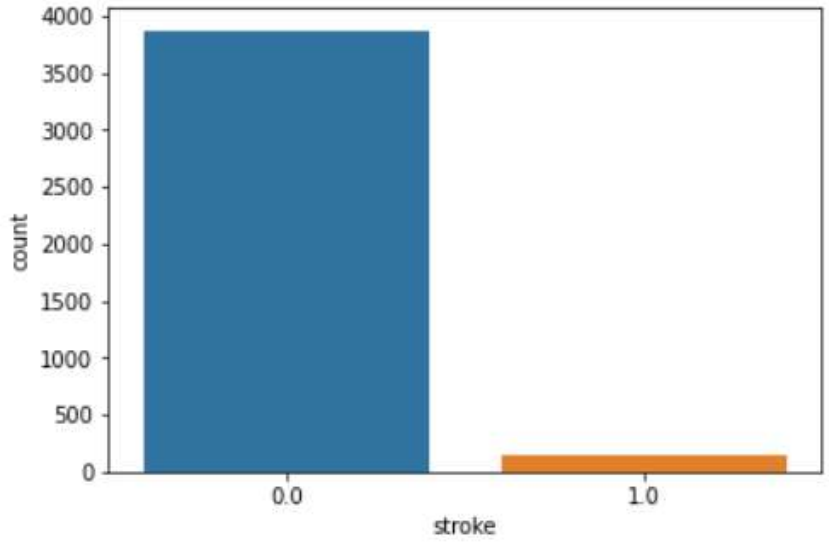


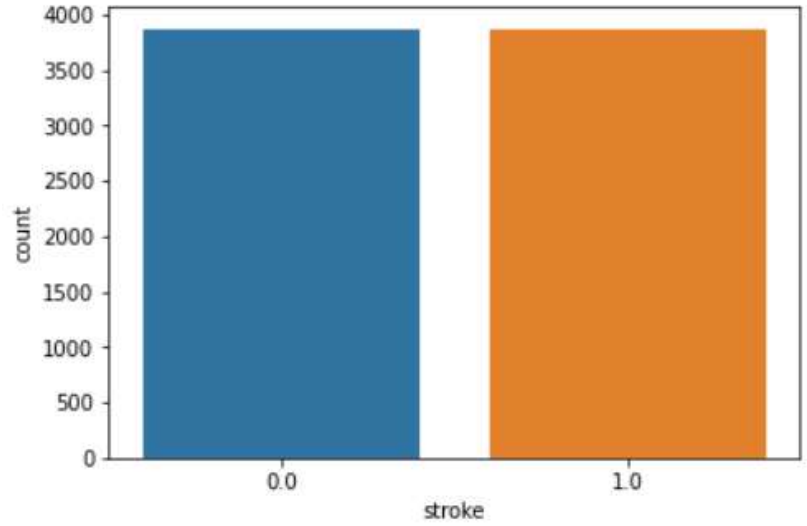
- We mentioned before that dataset bmi and avg_glucose_level features have outliers. Having outliers in the dataset guides the algorithm onto making wrong predictions, that seem just right to the predictive model. Therefore, we have eliminated the outliers using a graphical method.
- The outliers of bmi and avg_glucose_level index have been removed. Let's plot the distribution to see if they are still skewed.



Balance the dataset which is highly imbalanced

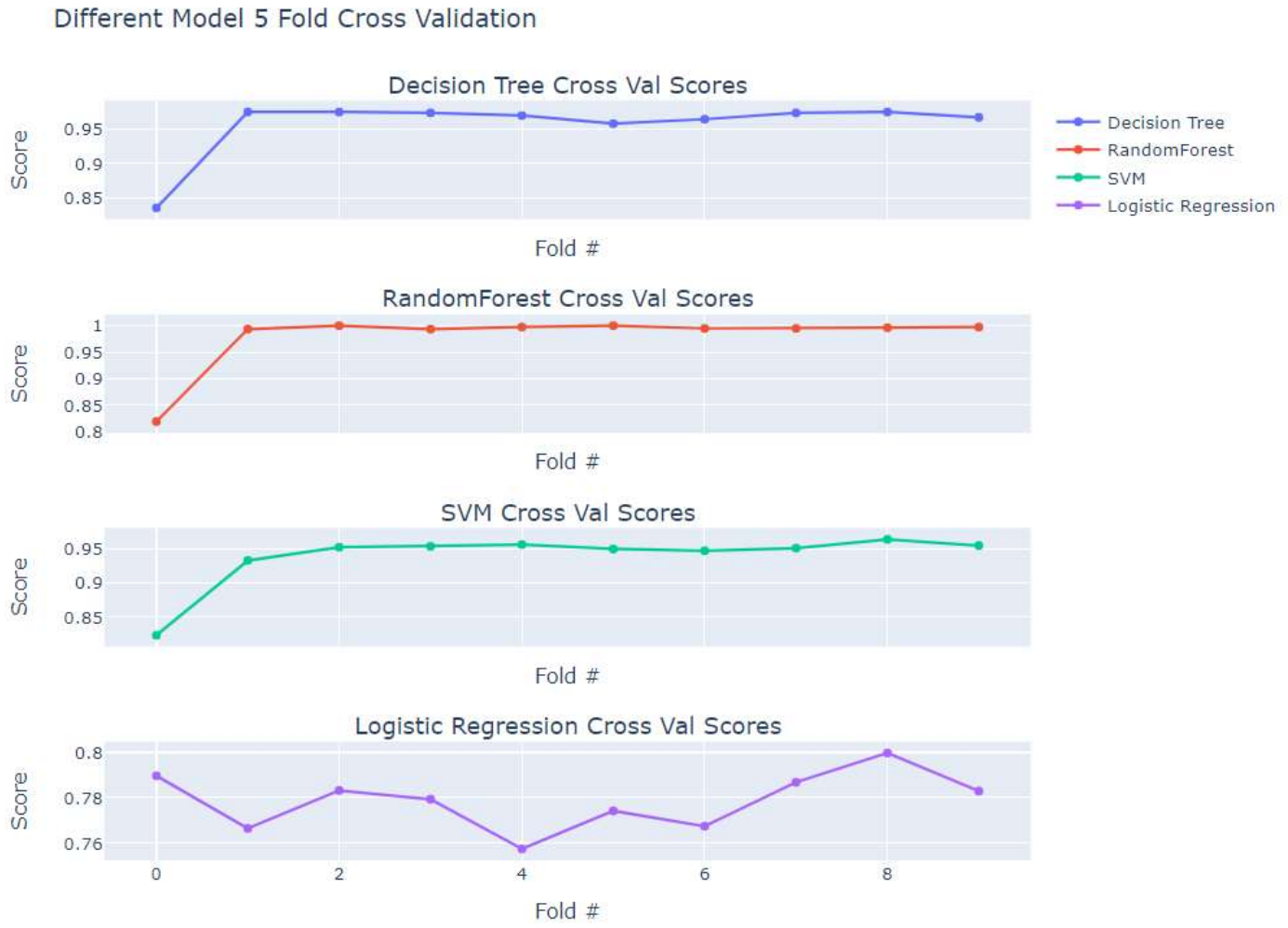
- Before applying any classifier algorithm, we should balance over dataset because imbalanced classifications pose a challenge for predictive modeling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class.
- Since our data is highly imbalanced, there are two ways to deal with it. We can either undersample the majority class or we could oversample the minority class. We will be using oversampling technique for this project.
- The simplest approach involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling technique or **SMOTE** for short.
- After appyling smote we can see that our data is balanced.





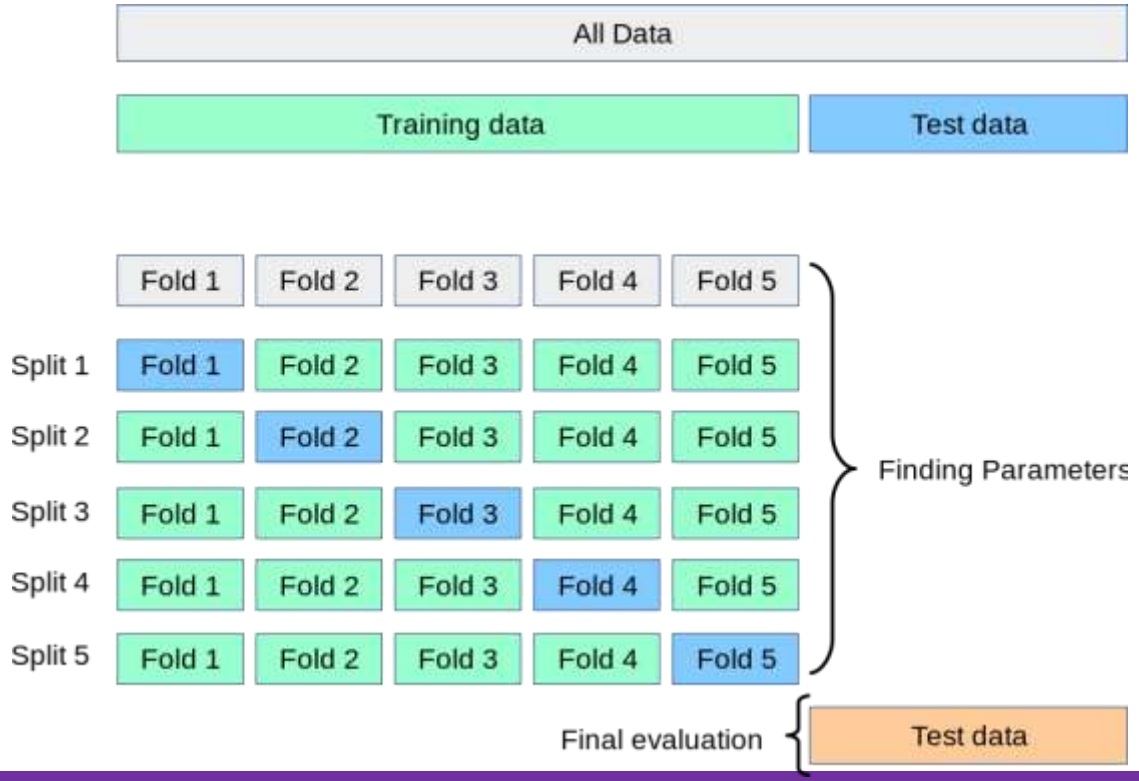
Modeling

- Cross-validation is a statistical method used to estimate the skill of machine learning models.
- It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement.
- We tried four different algorithms to see which one will give better results. Based on on the cross-validation scores, we decided to use Random Forest for classification because it is more stable and more accurate than other algorithms.
- Moreover, cross validation score allows us to prevent overfitting.



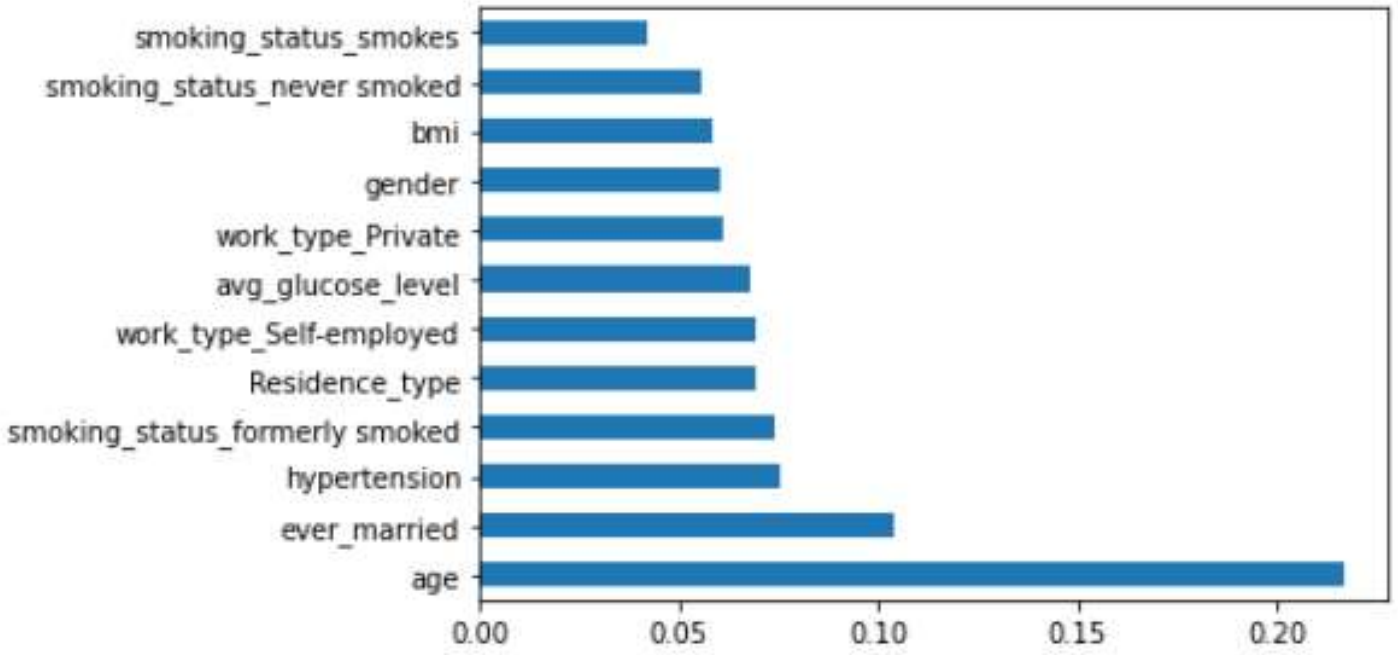
Split the dataset and normalization

- We used strafied train-test split so that split the dataset into train and test sets in a way that preserves the same proportions of examples in each class as observed in the original dataset.
- After that we use **Min-Max Normalization** technique before evaluating the machine learning model. Min-max normalization basically transforms features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g., between zero and one.
- Finally, we used grid search to find best parameters and used that parameters in random forest classification.



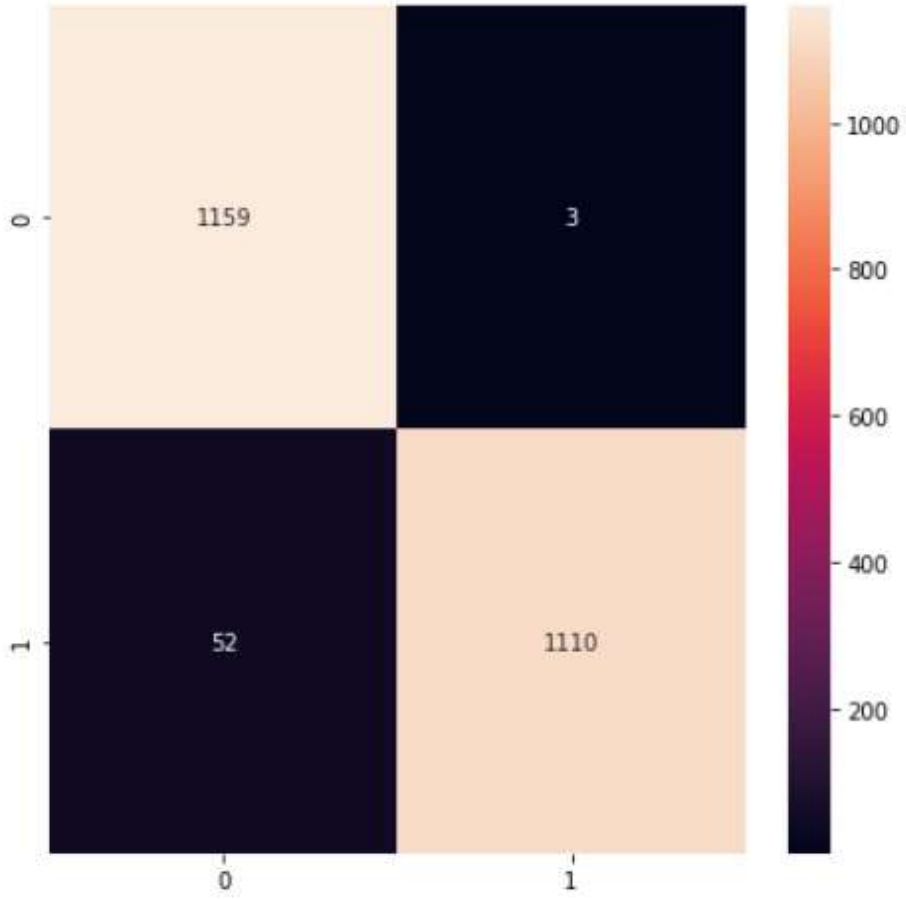
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

By looking feature importance graph below, we can say that **age** feature has significantly more effect than the other features for having stroke or not.



Confusion Matrix

- Confusion matrix evaluates the accuracy of a classification. Rows are predicted values columns are actual values so that we can interpret our model using these informations.
- By looking our model's confusion matrix, we can say that our model's prediction for not having stroke better than having stroke. Our model predict 3 not having stroke as having stroke and 52 having stroke as not having stroke.



Classification Report

	precision	recall	f1-score	support
Stroke Negative	0.96	1.00	0.98	1162
Stroke Positive	1.00	0.96	0.98	1162
accuracy			0.98	2324
macro avg	0.98	0.98	0.98	2324
weighted avg	0.98	0.98	0.98	2324

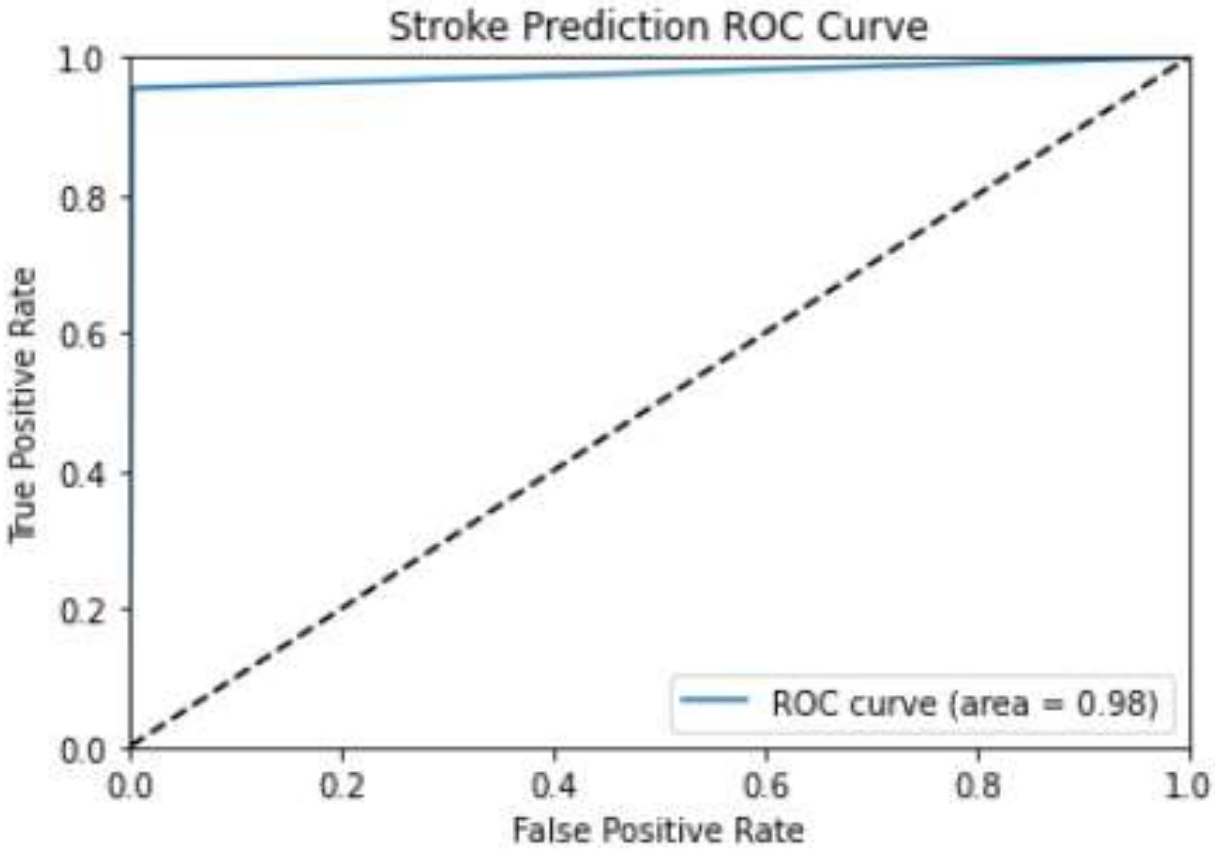
- The classification report visualizer displays the precision, recall, F1, and support scores for the model.
- **True Positives (TP):** These are the correctly predicted positive values which means that the value of actual class is yes, and the value of predicted class is also yes.
- **True Negatives (TN):** These are the correctly predicted negative values which means that the value of actual class is no, and value of predicted class is also no.
- **False Positives (FP):** When actual class is no and predicted class is yes.
- **False Negatives (FN):** When actual class is yes but predicted class in no.

- The **recall** means "What percent of our predictions were correct" Recall is the ability of a classifier to find all positive instances Recall = TP/(TP+FN)
- The **precision** will be "What percent of the positive cases did we catch
- The **f1-score** means "What percent of our positive predictions were correct".
- The **support** is the number of occurence of the given class in your dataset.
- The **macro-average** precision and recall score is calculated as arithmetic mean of individual classes' precision and recall scores.
- The **macro-average F1-score** is calculated as arithmetic mean of individual classes' F1-score.

By looking our classification report, we can say that our model's precision,recall and f1-score very similar between having stroke or not.

Roc-Curve

- **AUC-ROC** curve is the model selection metric for bi–multi class classification problem. ROC is a probability curve for different classes. ROC tells us how good the model is for distinguishing the given classes, in terms of the predicted probability.
- The higher the AUC(Area under curve), the better the performance of the model at distinguishing between the positive and negative classes.
- When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.
- As we can see from below graph , we have a clear distinction between the two classes as a result, we have the AUC of 1. The maximum area between ROC curve and base line is achieved here.
- Our result shows us, our classifer almost excellent.



References

- [1] https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html
- [2] https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/
- [3] https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html
- [4] https://medium.com/@venkatasujit272/overview-of-cross-validation-3785d5414ece
- [5] https://scikit-learn.org/0.17/modules/generated/sklearn.metrics.r2_score.html
- [6] https://vitalflux.com/micro-average-macro-average-scoring-metrics-multi-class-classification-python/
- [7] https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397
- [8] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- [9] https://towardsdatascience.com/clustering-why-to-use-it-16d8e2fbafe#:~:text=Inertia%20is%20the%20sum%20of,how%20dense%20the%20clusters%20are
- [10] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- [11] https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74
- [12] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
- [13] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
- [14] https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/
- [15] https://www.kaggle.com/aditimulye/stroke-prediction-visualization-prediction
- [16] https://www.kaggle.com/bariscal/stroke-entirely-ml-project-and-eda
- [17] https://www.kaggle.com/ivangavrilove88/stroke-fe-smote-technique-17-models
- [18] https://www.kaggle.com/jabeen12-stroke-prediction-data-visualization
- [19] https://www.kaggle.com/ginelledsouza/stroke-analysis
- [20] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html
- [21] https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/
- [22] https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/
- [23] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html