HEART STROKE PREDICTION

GROUP 13

Nikhil Sannagiri – 0770642
Swetha Bompada – 0774222
Khushwinder kaur – 0754238
Venkat Sai Mandava – 0771590

CONTENTS:

1. INTRODUCTION TO HEART- STROKE:

The human heart is the most important organ in the body. Any heart irregularity might induce discomfort in other parts of the body. In today's society, heart stroke is one of the most common causes of death. Heart stroke can be caused by an unhealthy lifestyle, such as smoking, drinking too much alcohol, or eating too much fat, all of which can lead to hypertension. According to the World Health Organization, more than 10 million people die each year because of a heart attack. The best way to avoid a heart attack is to live a healthy lifestyle and to recognize it early. In today's healthcare, the key difficulty is to provide high-quality services and precise diagnoses. The suggested research aims to detect heart attacks early on to avoid serious effects.

Data mining techniques are methods for obtaining useful and hidden information from enormous amounts of data. Machine Learning (ML), a branch of data mining, excels at handling huge, well-formatted datasets. Machine learning may be used to diagnose, detect, and forecast many disorders in the medical industry. The major purpose of this paper is to give doctors a tool for detecting heart stroke at an early stage. As a result, it will be easier to deliver appropriate treatment to patients while avoiding serious effects.

The importance of machine learning in detecting hidden discrete patterns and analyzing the data is critical. Following data analysis, machine learning approaches aid in the prediction and early detection of heart attacks. This research examines the performance of various machine learning algorithms for predicting cardiac stroke at an early stage, including support vector machines, decision trees, and random forests.

2. MOTIVATION:

The World Health Organization projections acted as a catalyst for tackling this problem. According to the World Health Organization, about 23.6 million people will die from heart stroke between now and 2030. As a result, anticipating a heart attack should be done in order to reduce the risk. The signs, symptoms, and physical examination of a patient are usually used to diagnose

cardiac issues. Finding the proper ailment is the most difficult and complex task in the medical field.

Heart stroke prediction has become one of the most difficult tasks in the medical world in recent years. Heart Stroke claims the lives of about one person per minute in the modern era.

In the realm of healthcare, data science is critical for analyzing massive amounts of data. Because predicting a heart attack is a difficult undertaking, it is necessary to automate the process in order to avoid the risks connected with it and to inform the patient well in advance.

3. DATASET DESCRIPTION:

The research was conducted using the stroke prediction dataset. This dataset has a total of 5110 rows and 12 columns. The output column stroke value is one of two values: one or zero. The result 0 indicates that no stroke risk was found, while 1 indicates that a risk of stroke was discovered. In this dataset, the probability of a 0 in the output column (stroke) is greater than that of a 1. The number 1 is found in 249 rows in the stroke column, while the value 0 is found in 4861 rows. Data preprocessing is used to balance data in order to increase accuracy.

Attribute information:

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 job", "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.

4. METHODS:

In this section we will provide a brief information about the steps or approaches taken, exploratory data analysis, the preprocessing techniques, the machine learning models, and evaluation metrics we are intending to use for our dataset.

4.1 Importing Libraries, Packages, and Dataset:

We imported the required libraries and packages which is essential to run a specific section or all the sections of the code in the python notebook (ipynb file). Then we had imported our dataset and stored it in a data frame for further processing.

4.2 Data Preprocessing:

Data preprocessing is required before building a model to remove unwanted noise and outliers from the dataset that may cause the model to deviate from its intended training. This stage addresses everything that is preventing the model from performing better. The data must be cleaned and prepared for model development after the necessary dataset has been collected.

- There are a total of 12 columns in the dataset.
- This dataset consists of both numerical values and categorical values.
- To begin, the id column is excluded because it has no influence on model construction.
- The column of 'Residence_type' begins with uppercase while others are not. To make a standardize grammar to prevent mistake we changed all column names into lowercase.
- We have removed the significant anomalies by cleaning, all of which have increased the model's efficiency.

- Removed all the null values in the dataset and replaced them with the mean of that column. In this scenario, the null values in the column BMI are filled using the data column's mean.

```
In [15]: def missing (df):
             missing_number = df.isnull().sum().sort_values(ascending=False)
             missing_percent = (df.isnull().sum()/df.isnull().count()).sort_values(ascending=False)
             missing_values = pd.concat([missing_number, missing_percent], axis=1, keys=['Missing_Number', 'Missing_Percent'])
             return missing_values

In [16]: missing(data)
```

Out[16]:

	Missing_Number	Missing_Percent
gender	0	0.0
age	0	0.0
hypertension	0	0.0
heart_disease	0	0.0
ever_married	0	0.0
work_type	0	0.0
residence_type	0	0.0
avg_glucose_level	0	0.0
bmi	0	0.0
smoking_status	0	0.0
stroke	0	0.0

- The above statistics shows that there are no null values, and the data is clean for model development.

4.3 Exploratory Data Analysis:

We did data exploration by looking at the contents of the data frame and figuring out what variables were kept inside. This will help to understand the different types of columns we're dealing with and which ones will be valuable for prediction. For understanding the features within the data, the data type, non-null count, overall count, mean, standard deviation, min, max, and specific quartiles, we employed basic functions such as head, tail, describe, and info.

Exploratory Data Analysis is the crucial process of using summary statistics and graphical representations to undertake preliminary investigations on data in order to discover patterns, spot anomalies, test hypotheses, and validate assumptions.

To visualize the characteristics, we looked at the distribution of Numerical and Categorical Features within the dataset. We generated a heat map for a correlation matrix to summarize the data and decide which features were strongly and weakly connected.

```
In [48]: plt.figure(figsize=(20,15))
         sns.heatmap(data.corr(), annot=True,cmap='coolwarm');
```

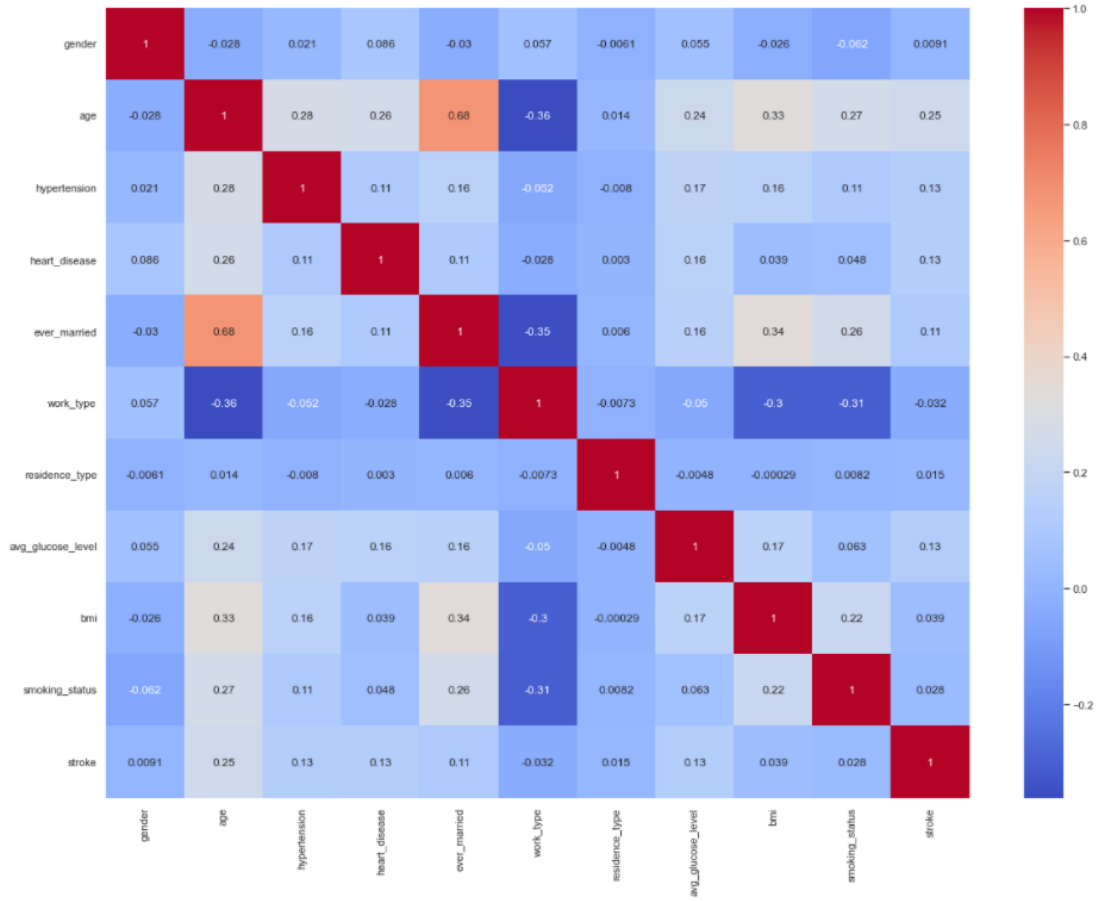
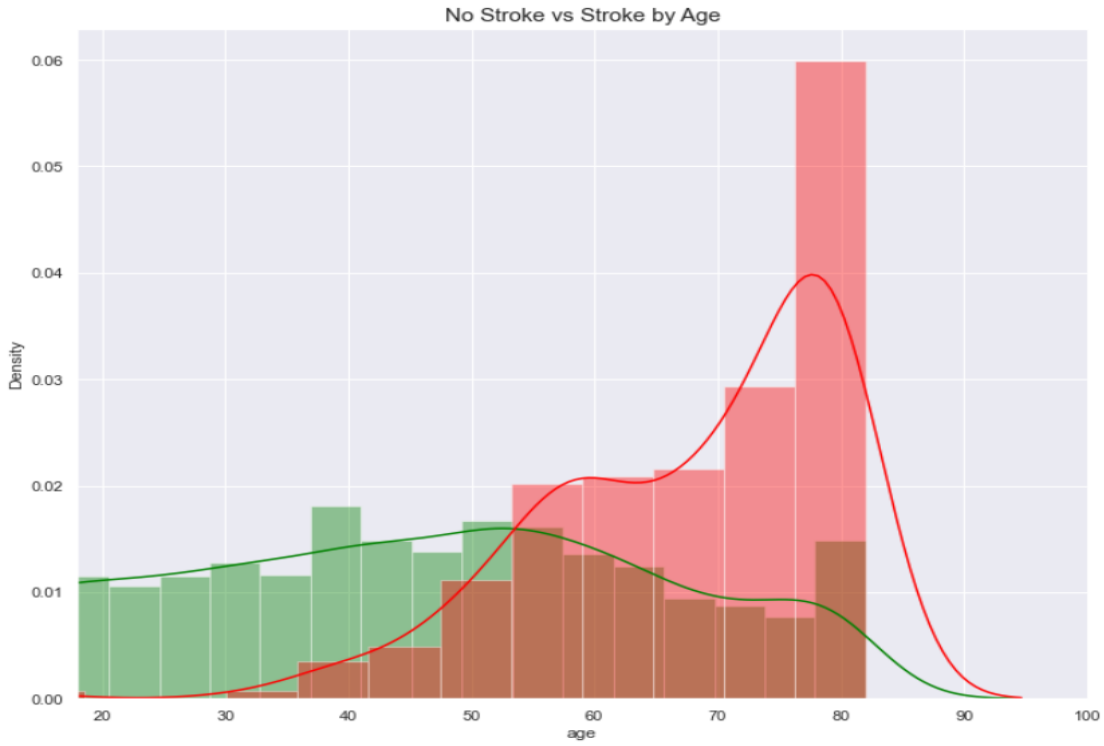


Fig: Correlation matrix

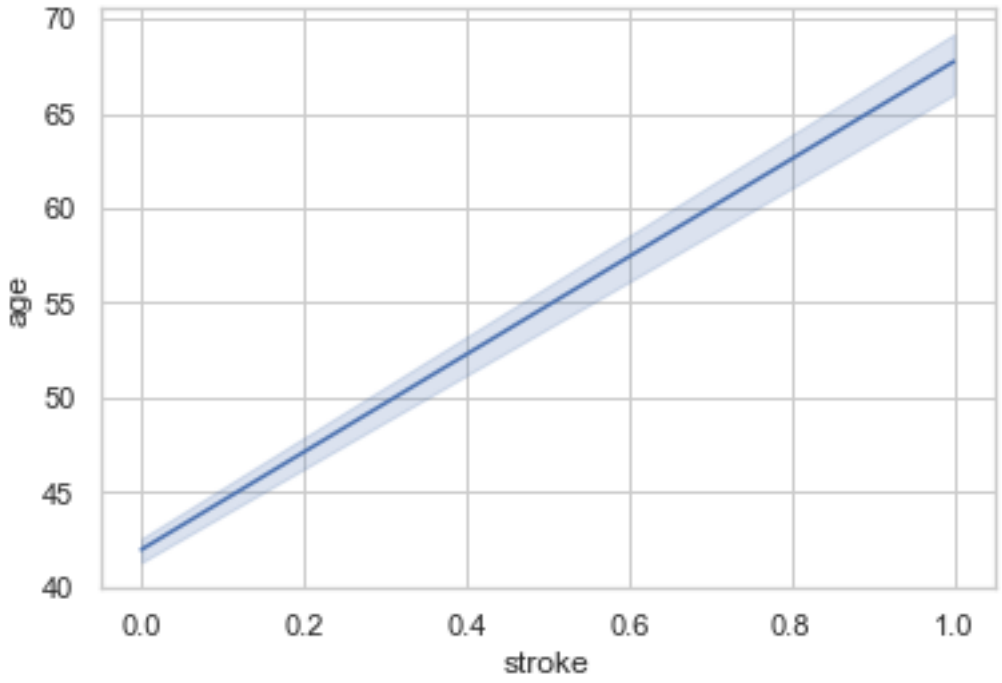
```
In [47]: abs(data.corr()['stroke']).sort_values(ascending=False)

Out[47]: stroke              1.000000
         age                 0.245239
         heart_disease       0.134905
         avg_glucose_level   0.131991
         hypertension        0.127891
         ever_married        0.108299
         bmi                 0.038912
         work_type           0.032323
         smoking_status      0.028108
         residence_type      0.015415
         gender              0.009081
         Name: stroke, dtype: float64
```

- From the above information, age is the primary attribute that is highly correlated to heart stroke occurrence.
- Heart disease, avg_glucose_level and hypertension also the primary reasons that leads to heart stroke.

1. Age:

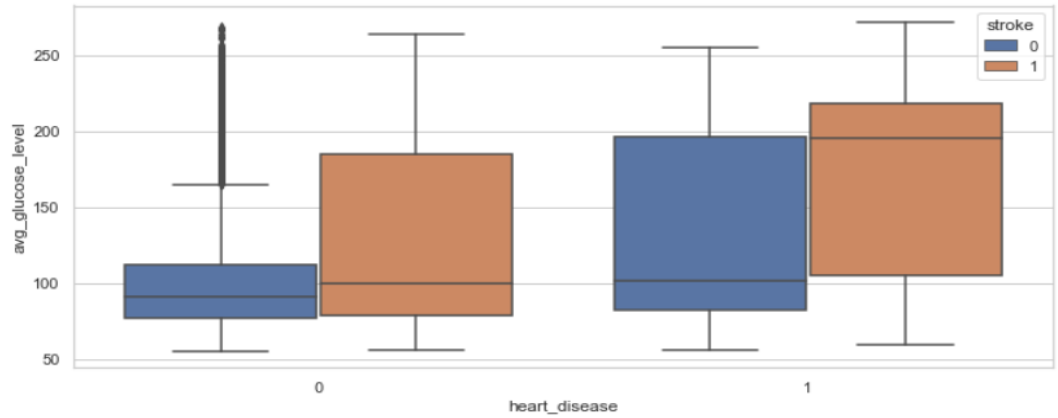


- The age column is a little left skewed with a peak around 60s



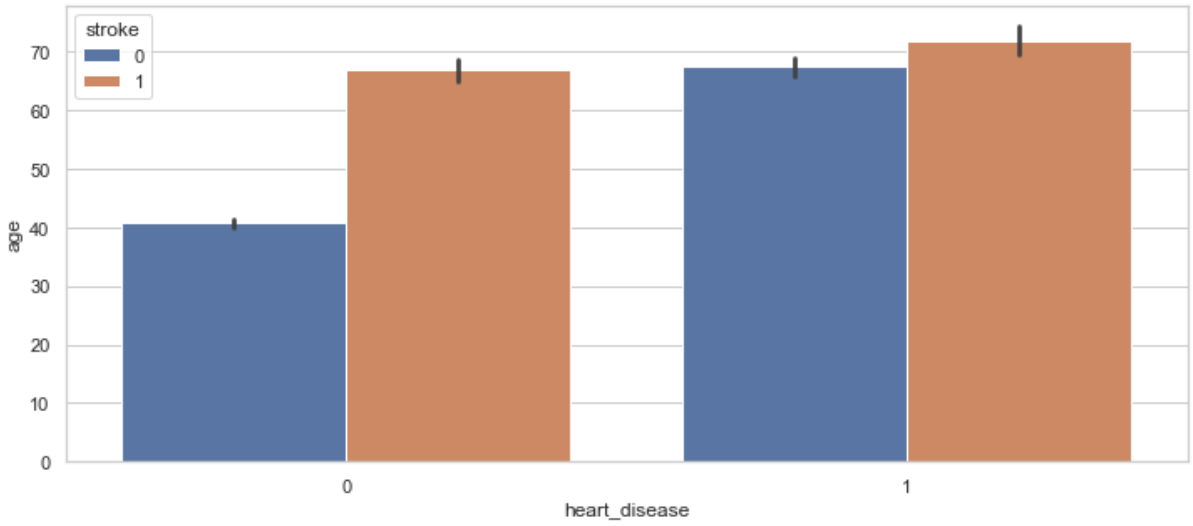
- From the above line graph, we can say that it's very obvious that people get strokes in elder ages.

2. Heart disease and average glucose level:



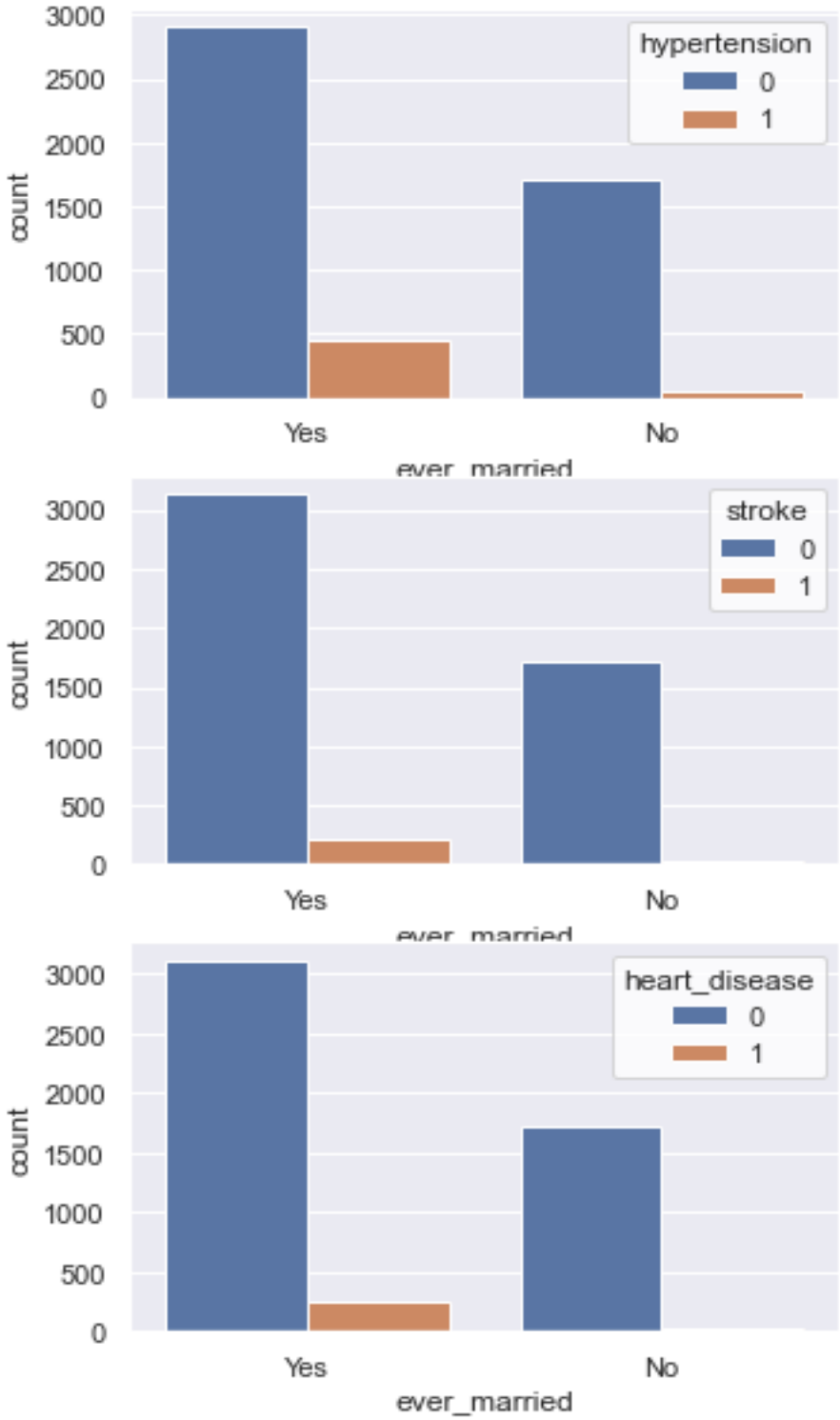
- The four most frequent types of heart disease are coronary artery disease, arrhythmia, heart valve disease, and heart failure.
- From the above box plot, we can say that People who had heart disease and whose average glucose level is above 100 are having more chances of heart stroke.

3. Heart disease and age



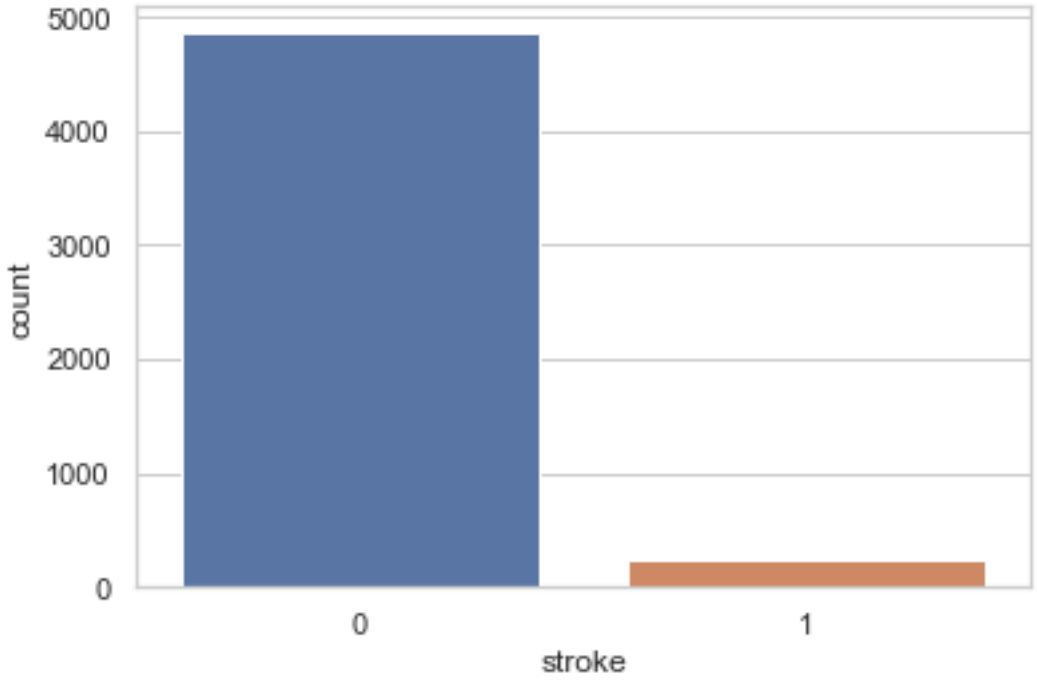
- As stated above, age and heart disease are positively correlated to the occurrence of stroke. From the above bar plot we can summarize that people having heart disease and whose age is high are the one who is affecting with heart stroke.

4. **Hypertension and ever married, stroke and ever married, heart disease and ever married**



- From the above graph, we can say that those who were ever-married have more cases of hypertension, heart-disease and stroke than those who were not.

5. Stroke

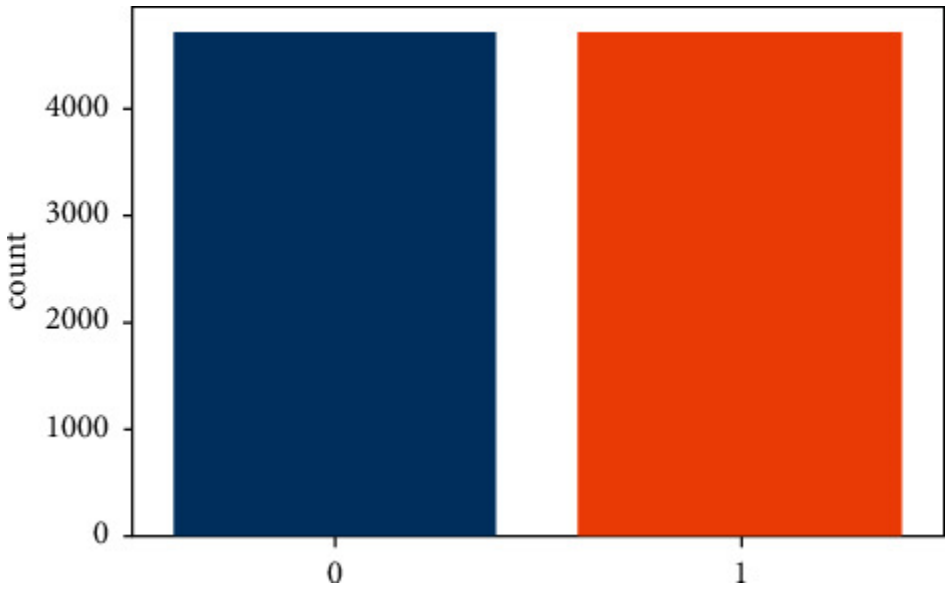


- From the above graph we can say that our target variable stroke is completely imbalanced to improve the accuracy data preprocessing is required on target variable.

Data preprocessing on target variable:

This unbalanced data must be dealt with initially in order to obtain an efficient model. This was accomplished using the SMOTE technique. The balance output column of the dataset is shown in below figure.

```
In [63]: from imblearn.over_sampling import SMOTE
         sm = SMOTE(sampling_strategy='minority')
         X, y = sm.fit_resample(X, y)

         y.value_counts()

Out[63]: 0    4860
         1    4860
         Name: stroke, dtype: int64
```



- From the above bar plot, we can see that our target variable is completely balanced which will result in better accuracy.

4.4 Ordinal Encoding:

Each unique category value is allocated an integer value in ordinal encoding. The Ordinal Encoder class in the scikit-learn Python machine learning toolkit implements this ordinal encoding technique. It will assign integers to labels in the order that they appear in the data by default.

```
In [41]: from sklearn.preprocessing import OrdinalEncoder

         ordinal_encoder = OrdinalEncoder()
         data[object_cols] =ordinal_encoder.fit_transform(data[object_cols])
```

```
In [42]: data.head()
```

Out[42]:

	gender	age	hypertension	heart_disease	ever_married	work_type	residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	1.0	67.0	0	1	1.0	2.0	1.0	228.69	36.600000	1.0	1
1	0.0	61.0	0	0	1.0	3.0	0.0	202.21	28.893237	2.0	1
2	1.0	80.0	0	1	1.0	2.0	0.0	105.92	32.500000	2.0	1
3	0.0	49.0	0	0	1.0	2.0	1.0	171.23	34.400000	3.0	1
4	0.0	79.0	1	0	1.0	3.0	0.0	174.12	24.000000	2.0	1

From the above summary, we can see that all the columns are converted into numerical values using ordinal encoding technique.

We can also use label encoding or one hot encoding techniques to convert categorical columns to numerical columns.

4.5 Machine learning models:

The study of algorithms that can be improved through time is known as machine learning. It is a branch of AI that is used to create a model that can interpret sample data, also known as train data, in order to train the model to make better and more accurate predictions for test data in the future. For our dataset, we employed 3 Machine Learning Models. The algorithms have been developed are:

- Decision tree classifier model
- Support vector machine model
- Logistic Regression model.

4.6 Evaluation metrics:

Evaluation Metrics are used to define a model's capacity. These metrics are used to assess the model's performance. These metrics are critical because they provide a better understanding of how each model is performing at its best and what can be improved in order to improve scores later. We used the following metrics:

- **Classification Report**
- **F1 Score**
- **Precision**
- **Recall**
- **Accuracy**
- **Confusion Matrix**

After all the work done, we printed the accuracy scores and classification report of every model. Then, we plotted confusion matrix of different models to show the accurate guesses and erroneous predictions and we also printed the evaluation metrics scores and compared them together for further analysis. All the results are mentioned in the Results section.

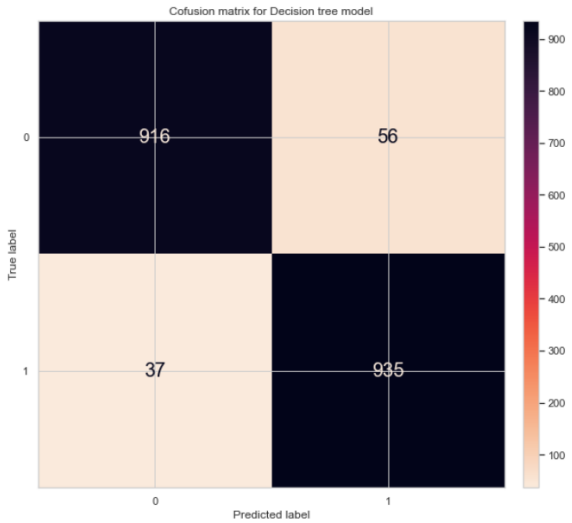
5. RESULTS:

5.1 Decision Tree classifier: Decision Tree algorithm is in the form of a flowchart where the inner node represents the dataset attributes, and the outer branches are the outcome. Decision Tree is chosen because they are fast, reliable, easy to interpret, and very little data preparation is required. In Decision Tree, the prediction of class label originates from root of the tree. The value of the root attribute is compared to record's attribute. On the result of comparison, the corresponding branch is followed to that value and jump is made to the next node.

The classification report for the decision tree classifier is shown below:

```
Decision tree model accuracy : 0.952
F1-score for the decision tree model is : 0.953

 Classification report for Decision tree classifier
              precision    recall  f1-score   support

  Stroke = 0       0.97      0.94      0.95       972
  Stroke = 1       0.94      0.97      0.95       972

    accuracy                           0.95      1944
   macro avg       0.95      0.95      0.95      1944
weighted avg       0.95      0.95      0.95      1944
```

The final F1-score in this case is 95%. An individual's F1-score is 95 percent for healthy individuals and 95 percent for those who had a heart stroke. Also, the precision and recall are shown in the above report. The accuracy is 0.952.

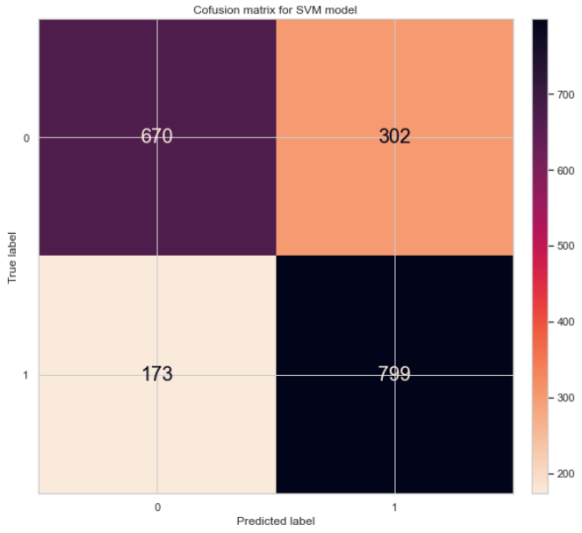


The above confusion matrix shows decision tree classifier prediction. The predicted outcome and the model's calculated performance are shown in the confusion matrix. There are 935 accurate guesses and 56 erroneous predictions.

5.2 Support Vector Machine model: In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.
The classification report for the Support vector machine is shown below:

```
SVM model accuracy : 0.756
F1-score for the support vector machine model : 0.771

 Classification report for SVM classifier
              precision    recall  f1-score   support

  Stroke = 0       0.79      0.69      0.74       972
  Stroke = 1       0.73      0.82      0.77       972

    accuracy                           0.76      1944
   macro avg       0.76      0.76      0.75      1944
weighted avg       0.76      0.76      0.75      1944
```

The final F1-score in this case is 77%. An individual's F1-score is 74 percent for healthy individuals and 77 percent for those who had a heart stroke. Also, the precision and recall are shown in the above report. The accuracy is 0.756.



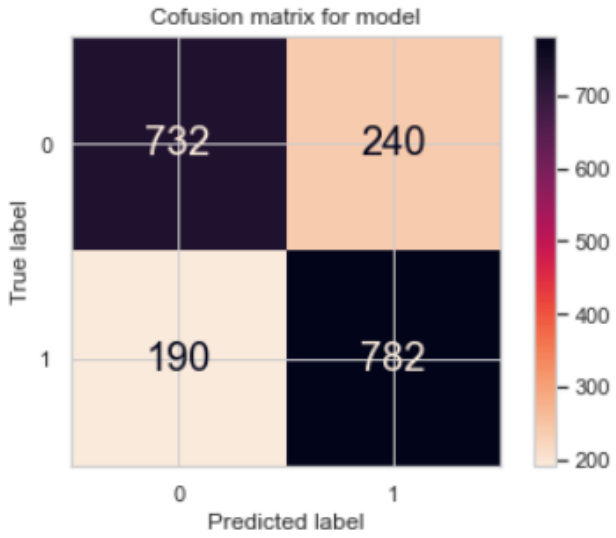
The Support vector machine model prediction is shown in the confusion matrix above. The confusion matrix displays the projected outcome as well as the model's computed performance. There are 799 correct predictions and 302 incorrect ones.

5.3 Logistic Regression: Logistic Regression is one of the most widely used ML algorithms in the supervised learning approach. It is a forecasting strategy that predicts a categorical dependent variable using a set of independent factors.

The classification report for the Logistic Regression model is shown below:

```
Confusion Matrix :

 [[732 190]
 [240 782]]

Classification Report :

              precision    recall  f1-score   support

           0       0.75      0.79      0.77       922
           1       0.80      0.77      0.78      1022

    accuracy                           0.78      1944
   macro avg       0.78      0.78      0.78      1944
weighted avg       0.78      0.78      0.78      1944
```

The final F1-score in this case is 77%. An individual's F1-score is 77 percent for healthy individuals and 78 percent for those who had a heart stroke. Also, the precision and recall are shown in the above report. The accuracy is 0.78(78%).



The Logistic Regression prediction is shown in the confusion matrix above. The confusion matrix displays the projected outcome as well as the model's computed performance. There are 782 correct predictions and 240 incorrect ones.

6. MODEL COMPARISON:

Analysis of different machine learning algorithms

ML model	Accuracy	F1 - score	Precision	Recall
Decision tree Classifier	95%	95%	94%	97%
Support Vector Machine	75%	77%	73%	82%
Logistic Regression	78%	78%	80%	77%

Although all algorithms offer a reasonable level of accuracy, the decision tree algorithm is the better choice due to its higher level of accuracy. Using the Decision tree classifier technique, this paper was able to attain 95 percent accuracy. However, in this paper, Support vector machine model performs poorly. Finally, when selecting a model, recall, Precision and F1 scores provide a more accurate representation of the model's actual performance.

After comparing all three models, based on accuracy, F1 score, precision and recall the best model is “Decision tree Classifier” with an accuracy of 95%.

Values obtained for confusion matrix using different algorithms

ML Model	True Positive	False positive	False Negative	True Negative
Decision tree Classifier	935	56	37	916
Support Vector Machine	799	302	173	670
Logistic Regression	782	240	190	732

- Decision tree classifier has highest number of true positives.

7. CONCLUSION:

Stroke is a potentially fatal medical condition that must be treated as soon as possible to avoid future consequences. The creation of a machine learning model could aid in the early diagnosis of stroke and, as a result, the mitigation of its severe repercussions. This study examines the efficiency of multiple machine learning algorithms in correctly predicting stroke based on a variety of physiological factors. Decision tree classifier model outperforms the other methods tested with a classification accuracy of 95 percent.

In the future, the study might be enhanced by creating a web application based on the Decision tree method and using a larger dataset than the one used in this analysis, which would help to deliver better results and aid health professionals in successfully and efficiently forecasting Heart stroke. In an ideal world, it would help patients obtain early treatment for strokes and rebuild their lives after the event.

- We concluded that Decision tree algorithm, is the best model because of it has the highest test accuracy, F1 score, precision and recall scores.

8. CONTRIBUTIONS:

Name	Contribution
Venkat Sai Mandava	Introduction, Data collection, Initial understanding of data, Motivation.
Khushwinder kaur	Data cleaning, Data processing, Exploratory Data Analysis.
Swetha Bompada	Project Proposal, Visualization & EDA, Conclusion, and presentation.
Nikhil sannagiri	Creating and evaluating models, Encoding, Comparing Models and Final Report.

9. REFERENCES:

- Kaggle dataset:
  - https://www.kaggle.com/fedesoriano/stroke-prediction-dataset
- https://www.cdc.gov/stroke/index.htm
- https://scikit-learn.org/stable/modules/tree.html
- https://www.cdc.gov/stroke/facts.htm
- https://machinelearningmastery.com/types-of-classification-in-machine-learning/
- https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d
- https://www.upgrad.com/blog/data-preprocessing-in-machine-learning/
- https://towardsdatascience.com/model-selection-in-machine-learning-813fe2e63ec6

10. APPENDICES:

- **Group13FinalReportHCA.docx:** Contains the final report
- **Group13ProjectHCA.ipynb**: Contains all the code for our project including importing, preprocessing, exploratory data analysis, machine learning algorithms and evaluation metrics