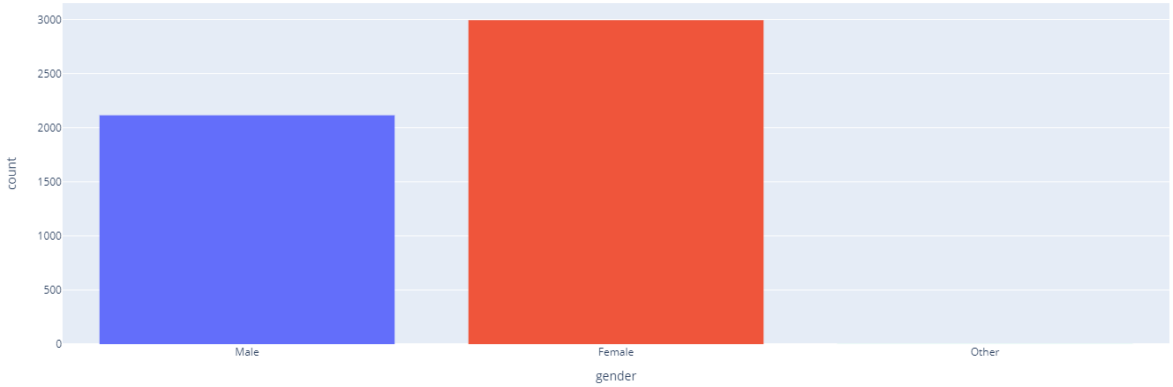
Stroke Prediction Model

Problem definition and motivation:

The problem is to predict the probability of a person having a stroke based on data like age, body mass index (bmi), heart disease, average glucose level … etc. The data used is a stroke prediction dataset that specifies 11 clinical features for predicting stroke events. The significance of such a problem is that it can be used to predict stroke events in a timely manner, so as to prevent the disease from occurring only by measuring some clinical features for a patient.

The dataset:

We are using a popular dataset from Kaggle that lists 11 clinical features for a sample of 5110 examples with 249 having a stroke and 4861 healthy. Analysis was done on each feature to see its value distribution initially. The first of which being the gender distribution of the samples.



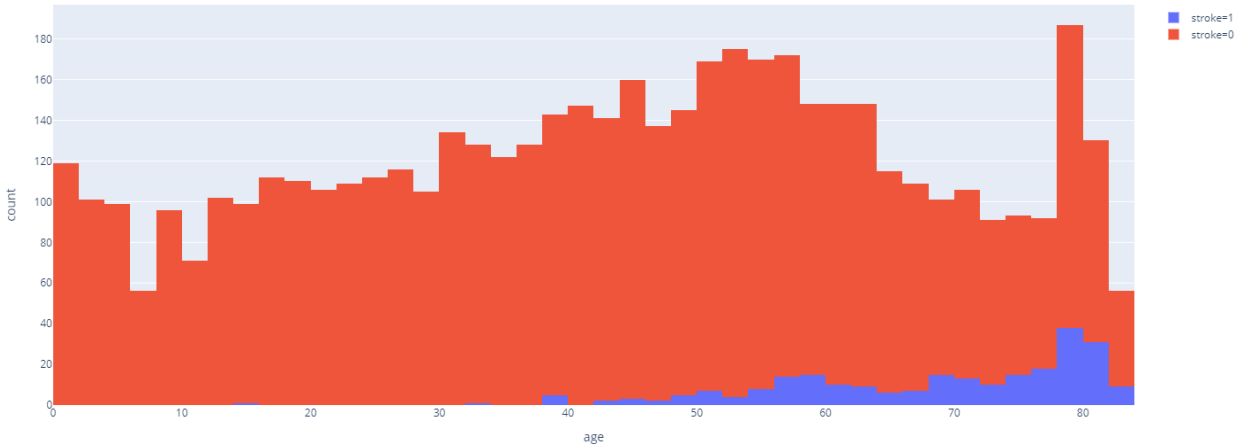
As evident by the graph, more females than males are represented in the data. There also exists one entry of a person that identifies as “Other”. This entry; however, was removed from the dataset to avoid misclassification in further stages.

This was the output after removing the entry

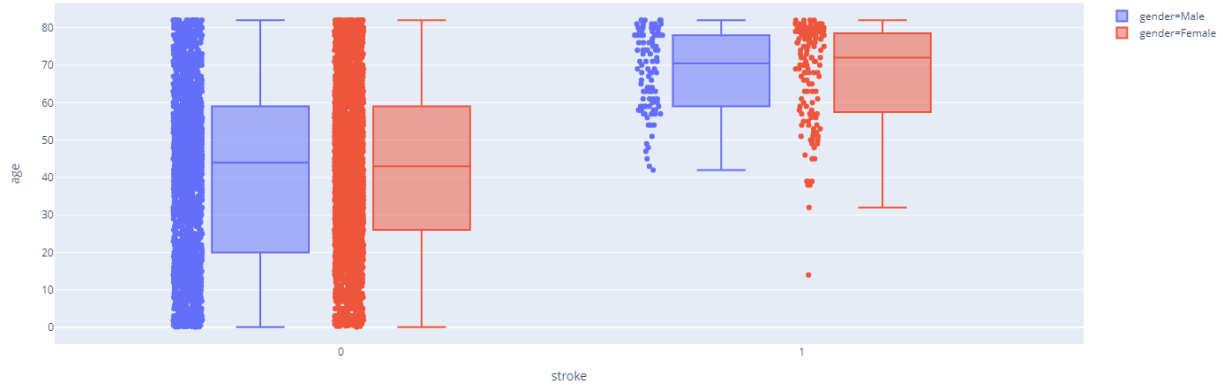
For the features Age, Average glucose levels and BMI, a distribution for both stroke patients and healthy individuals were plotted, along with box plots to show the interquartile range, median, mean and help us detect outliers.

Age:

Distribution for stroke patients and healthy individuals



Boxplot of Stroke and Non-stroke patient samples divided by gender



As shown in the box plot, there are two outliers in female stroke patients. Upon further exploration we find them out to be

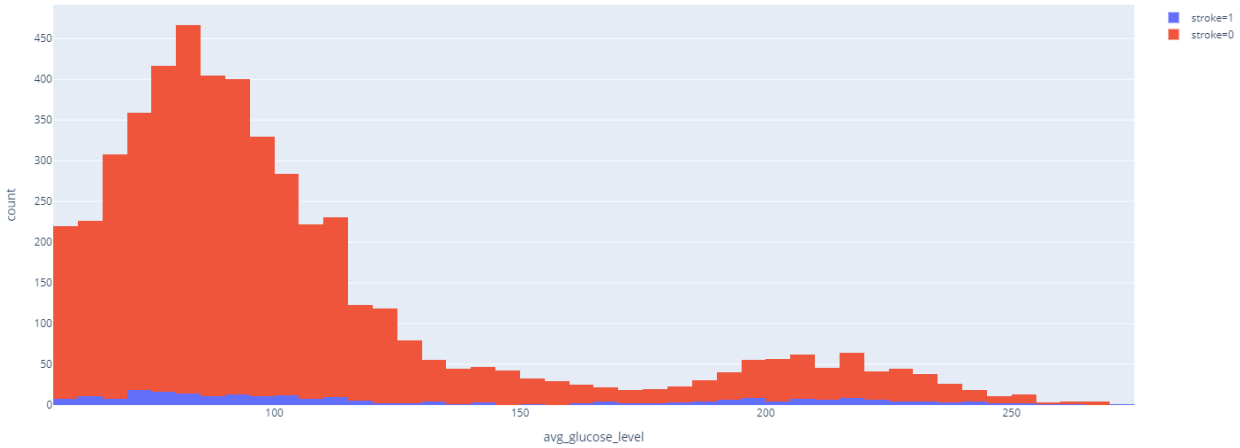
	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
162	69768	Female	1.32	0	0	No	children	Urban	70.37	NaN	Unknown	1
245	49669	Female	14.00	0	0	No	children	Rural	57.93	30.9	Unknown	1

As for the first sample we found data about an individual 1.32 years old, with no BMI data, this did not seem to be beneficial to further analysis so it was removed.

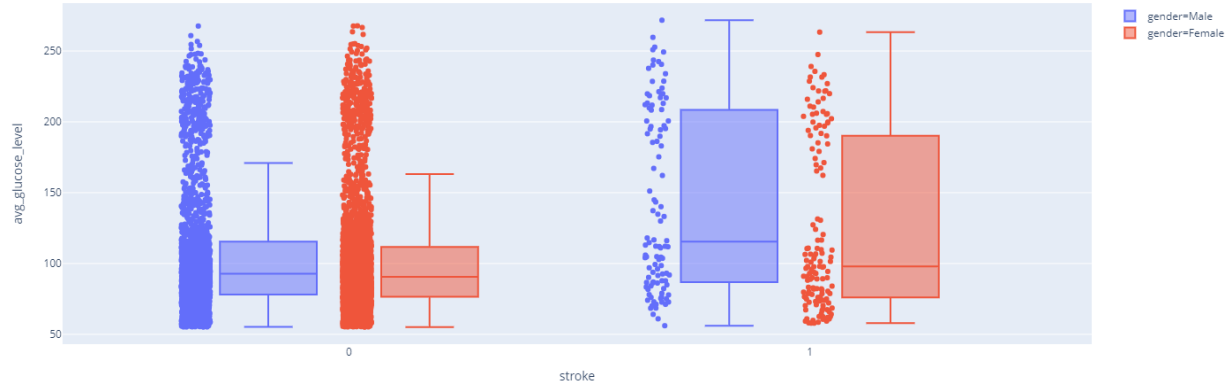
As for the second sample it seems to be of a teenager with 30.9 BMI; therefore, the female described is an obese 14 year old. The above-mentioned sample was beneficial to further analysis and modelling so it was not removed.

Average glucose level:

Distribution for stroke patients and healthy individuals

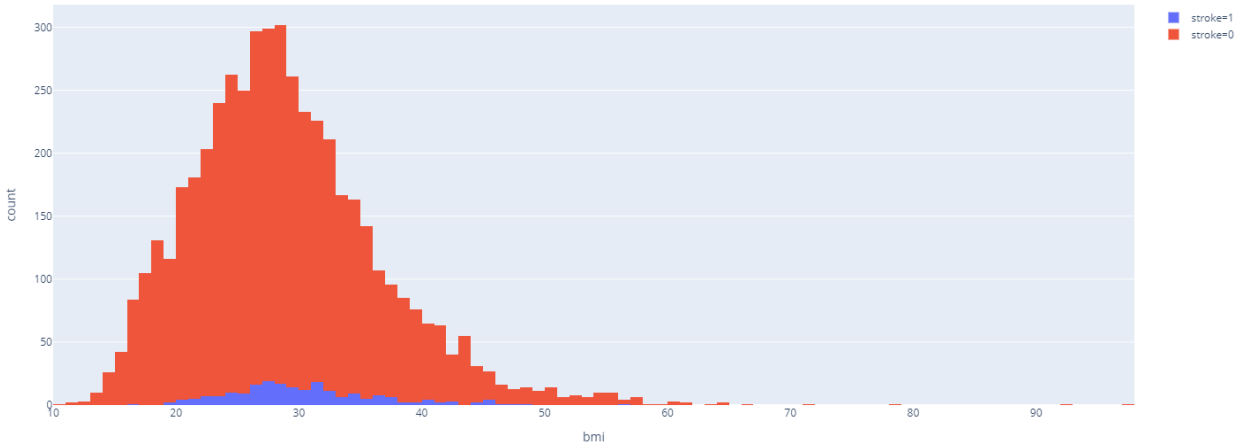


Boxplot of Stroke and Non-stroke patient samples divided by gender

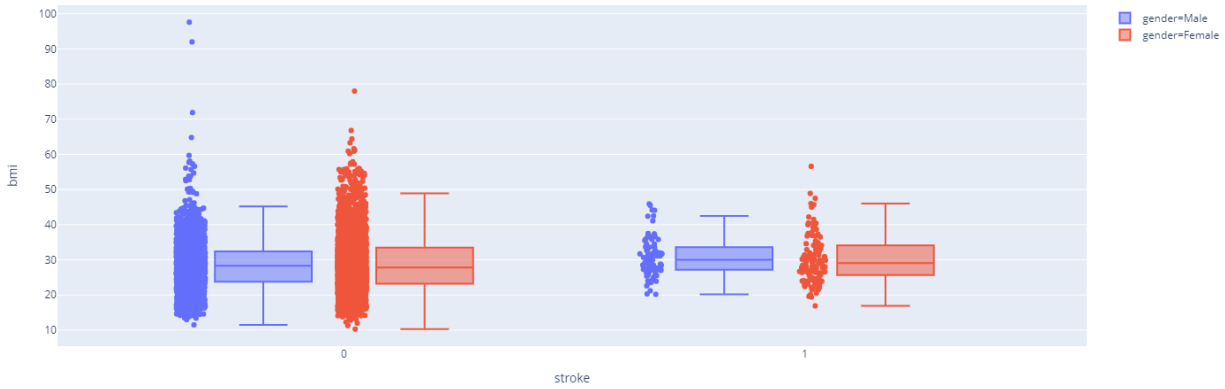


BMI:

Distribution for stroke patients and healthy individuals

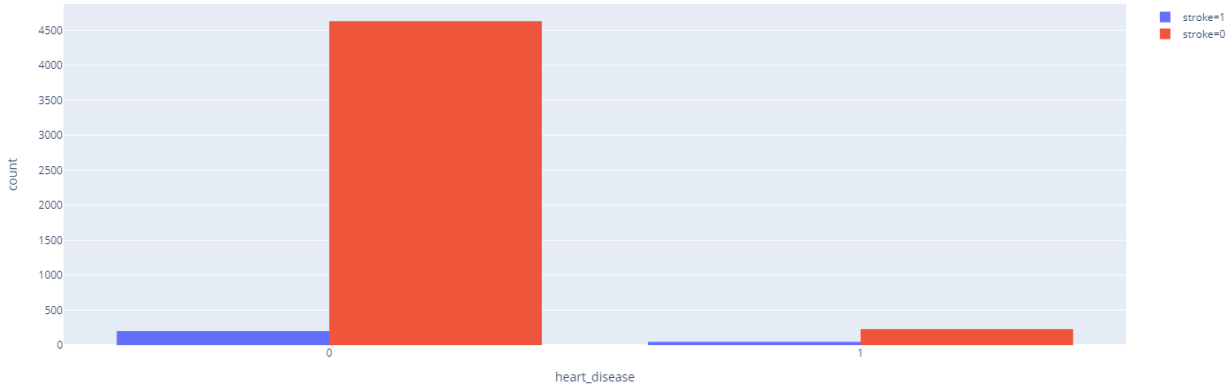


Boxplot of Stroke and Non-stroke patient samples divided by gender



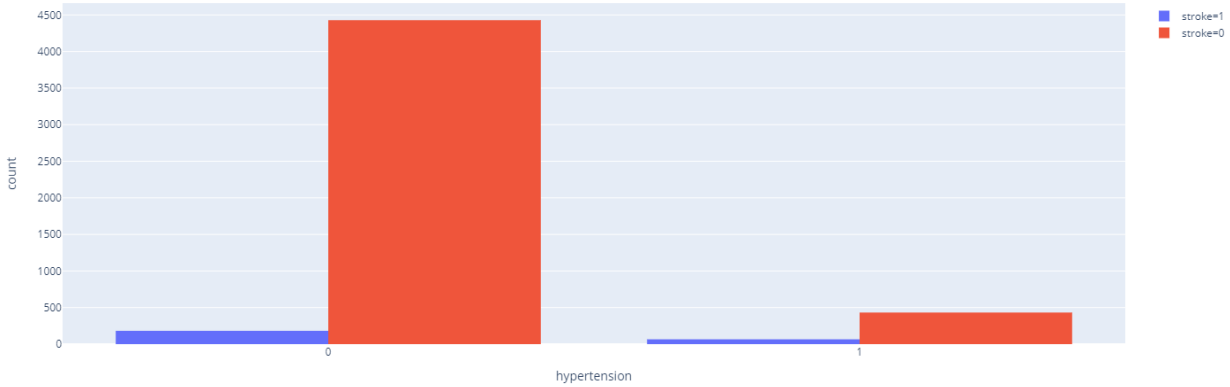
Heart Disease:

Distribution for stroke patients and healthy individuals



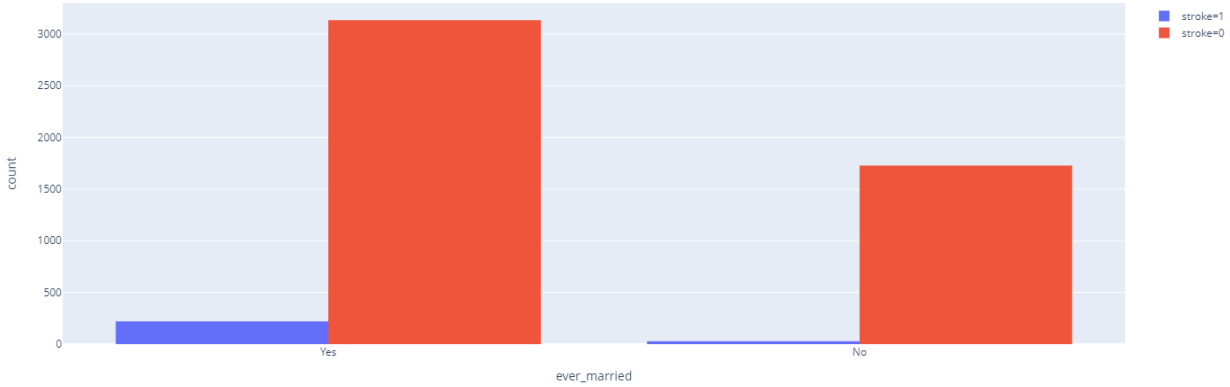
Hypertension:

Distribution for stroke patients and healthy individuals



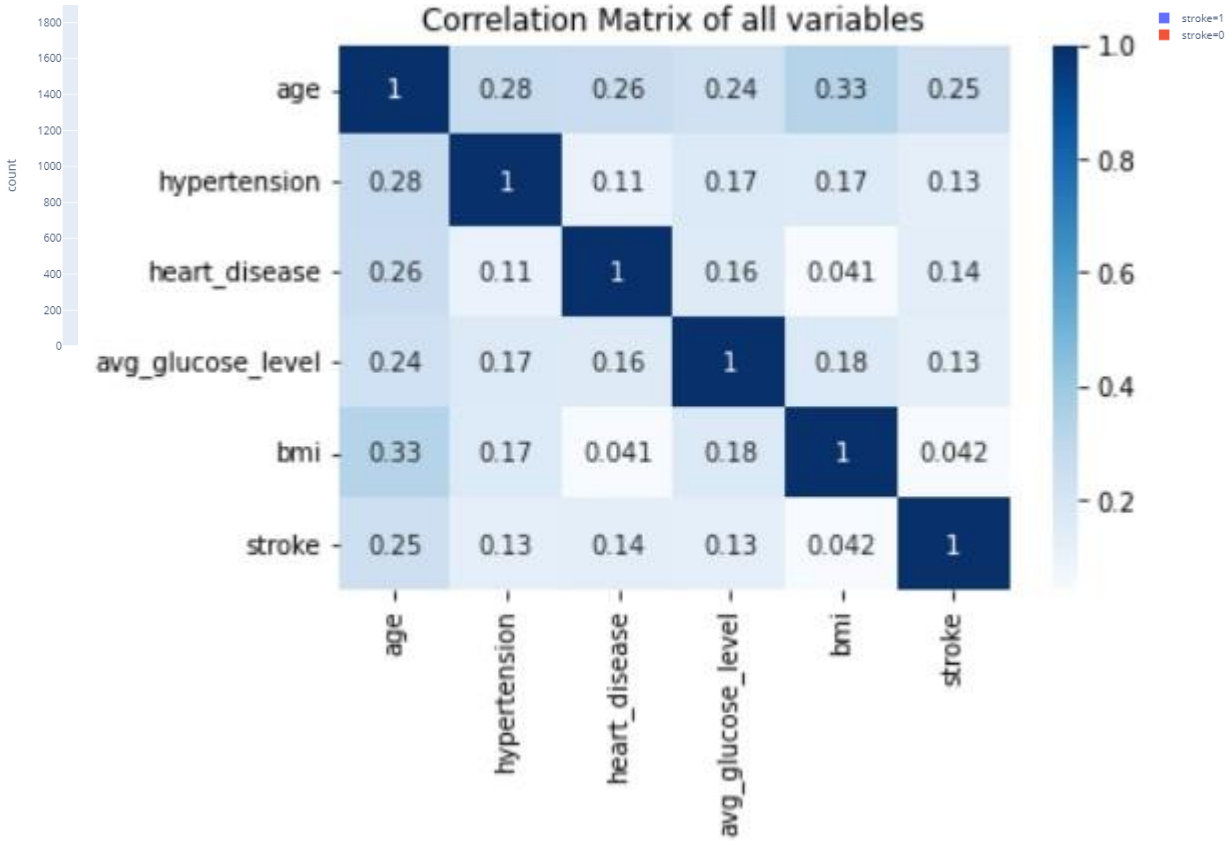
Ever Married:

Distribution for stroke patients and healthy individuals



Smoking Status:

Distribution for stroke patients and healthy individuals



Correlation Matrix of all features:

As evident by the matrix, Age has the highest correlation with stroke of all other features. BMI, on the other hand, has the lowest correlation with the features.

Methodology and Pre-processing

- **Data Imbalance**

  The data set we used is highly imbalanced with an imbalance ratio ~ 5:100. To deal with this imbalance in the dataset, we used the famous techniques of oversampling and undersampling. Specifically, we manually combined Synthetic Minority Oversampling Technique (SMOTE) and random undersampling in a pipeline that combines both together at each step. The SMOTE over-samples the minority class (no_stroke) and the random undersampling under-samples the majority class and then the data is fit together by the pipeline. We have also repeated the model but employed a different method to deal with the data imbalance; we only used oversampling (SMOTE) to compare the accuracy.

- **Data imputation and pre-processing**

  There were 200 missing bmi values in the dataset. To deal with the missing bmi values, we used KNN Imputor to replace the missing values. For bmi, we found it best to use KnnImputer because the meaning and the scale of bmi is dependent on age. The same bmi for a child and for an elderly person has different meanings. So, to replace the missing bmi values, we will take into account the neighbors to choose a meaningful replacement. The dataset was separated into two main categories: numeric features and categorical features. Finally, we designed a pipeline that used Standard Scaler for numeric features and One Hot Encoding for categorical data.

- **Data Splitting and modeling**

  We split the data into a training set (70%) and a testing set (30%). Different models were trained and tested to find the best accuracy. The models used were:

  1. Logistic regression
  2. K-Nearest Classifier
  3. Decision Trees
  4. Gradient Boosting Classifier
  5. Random Forest Classifier

Cross Validation:

We used a stratified k-fold for cross-validating our data. With oversampling the method should only be done on the training data and then get the models' metrics after running on the non-transformed stratified test data.

Results and discussion:

For dealing with the data imbalance, we found that the oversampling approach gave best results and the model that performed best was the gradient boosting classifier:

1. The oversampling and undersampling combination results:

	model	test_acc	test_precision	test_recall	test_f1	test_roc_auc
0	L_regressor	0.740313	0.134864	0.799102	0.230731	0.836144
1	K Nearest Classifier	0.789759	0.115255	0.496762	0.186916	0.738569
2	Decison Tree	0.807371	0.108816	0.410803	0.171980	0.619239
3	Gradient Boosting Classifier	0.844423	0.163446	0.531456	0.249717	0.822542
4	Random Forest Classifier	0.864188	0.152940	0.396136	0.220323	0.803113

2. The oversampling results:

	model	test_acc	test_precision	test_recall	test_f1	test_roc_auc
0	L_regressor	0.741161	0.132402	0.776218	0.226174	0.835173
1	K Nearest Classifier	0.925766	0.154018	0.116435	0.132245	0.623906
2	Decison Tree	0.908937	0.139887	0.168762	0.152171	0.557809
3	Gradient Boosting Classifier	0.948924	0.111111	0.008054	0.014396	0.826141
4	Random Forest Classifier	0.944684	0.125907	0.022776	0.038170	0.786312

Hyperparameter Tuning:

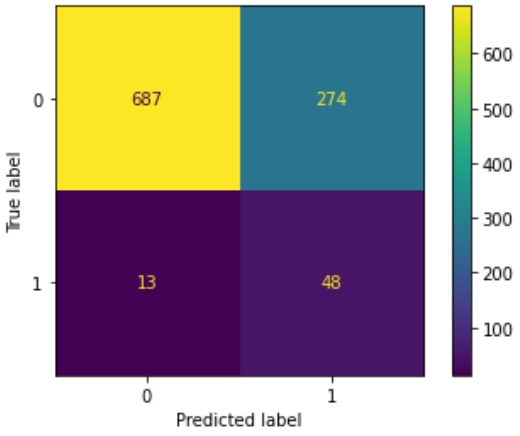
After running cross validation on all of our models we had a decent view on all the relevant metrics to our data set and accordingly we chose the best two models for hyperparameter tuning. The two models are logistic regression and stochastic gradient descent. The best parameters found for each model are listed below:

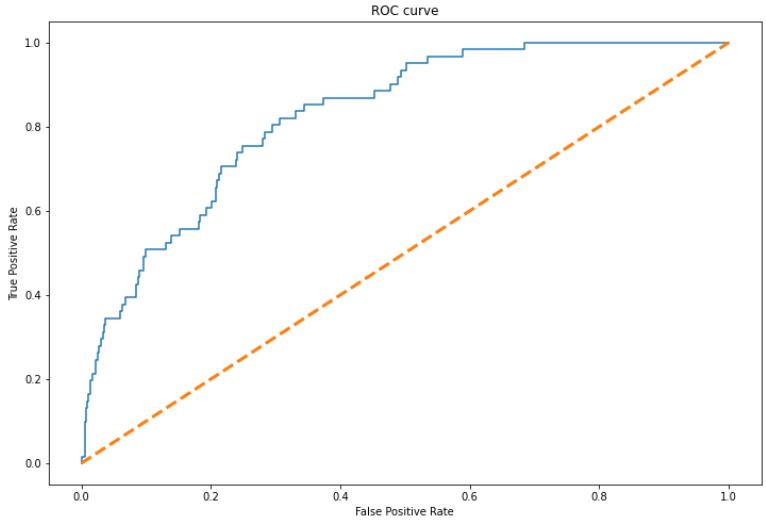
```
{'logisticregression__C': 0.01,
 'logisticregression__penalty': 'l2',
 'logisticregression__solver': 'liblinear'}
```

```
{'sgdclassifier__alpha': 0.1,
 'sgdclassifier__loss': 'hinge',
 'sgdclassifier__penalty': 'l1'}
```

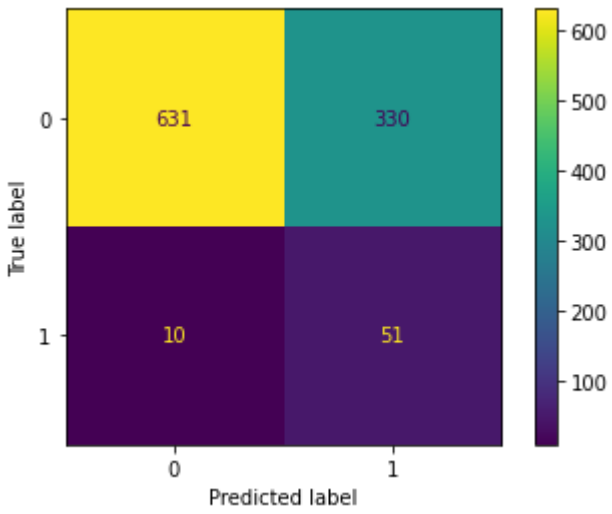
Then, after tuning our models using the best produced parameters, we fit the two models on the training data set and calculated predictions for the input test data and represented these predictions in a Confusion Matrix and Roc Curve

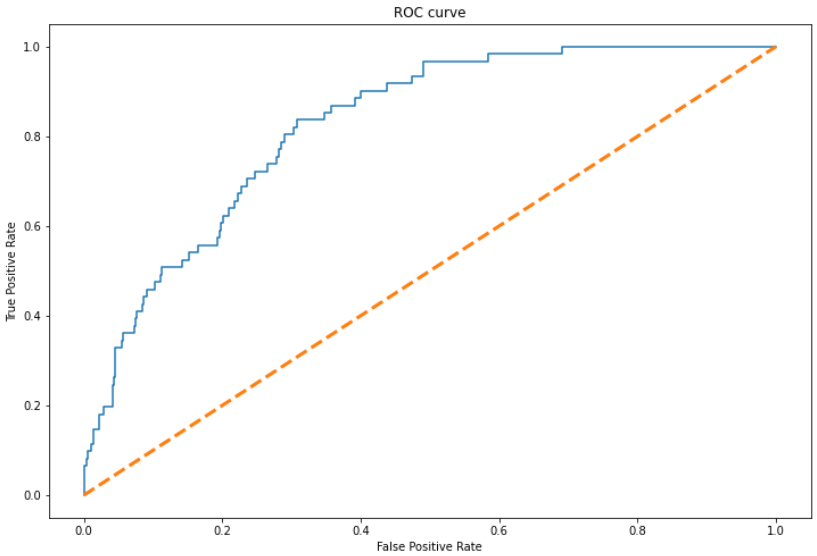
I. Logistic Regression





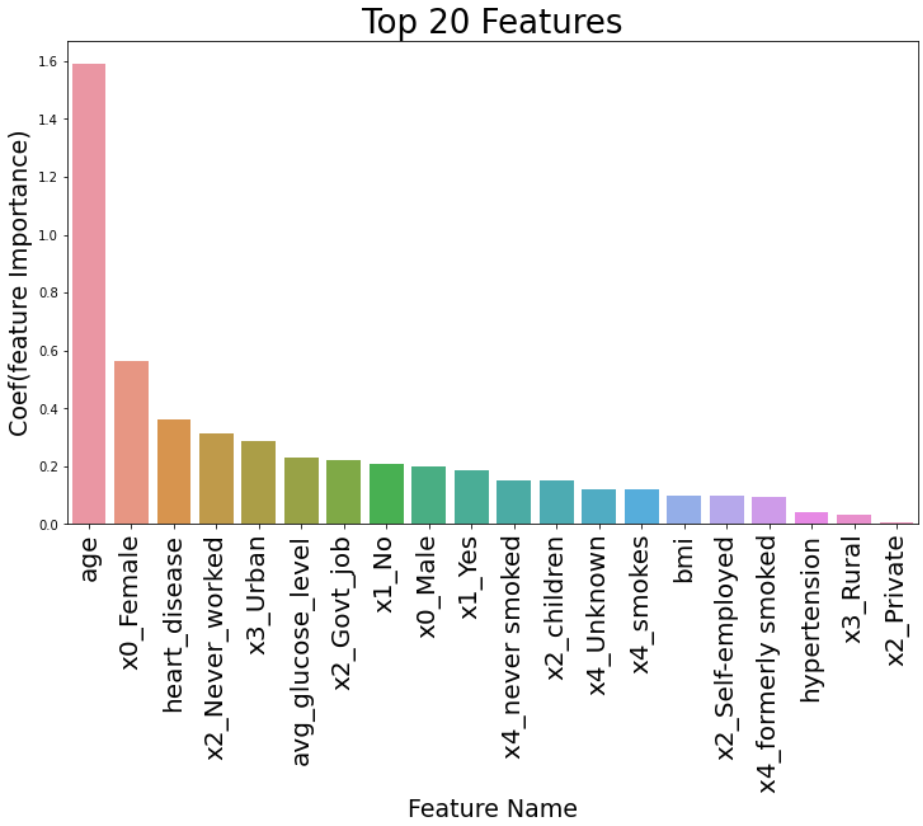
II. Stochastic Gradient Descent





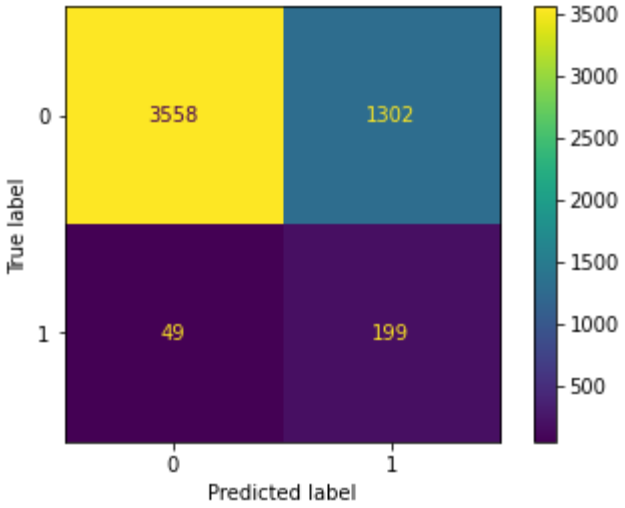
Feature Importance:

We calculated the importance of each feature as a function of the coefficients of the logistic regression formula and plotted the results in a histogram sorted descendingly.



Final Model

We built our final model using the logistic regression classifier (as it performed better across all tests on average). We fitted it on the whole dataset using our tuned parameters and the results of the model's predictions are shown in the Confusion Matrix below.



As can be seen from the matrix, the model is trying to minimize the number of false negative predictions which could be at the cost of more false positives but in real life the consequences of a single false negative can be drastic!

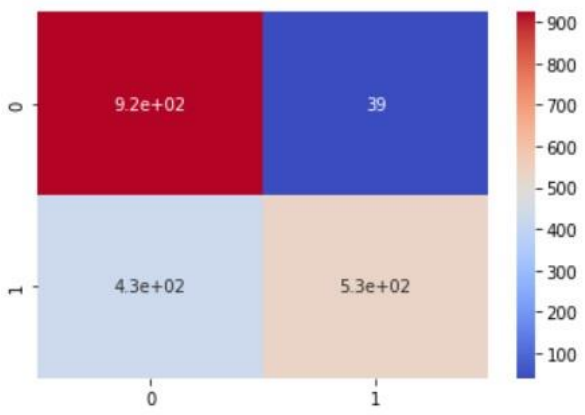
Stroke Prediction

Finally, after choosing the model and optimizing it to achieve our target we wrote a function that takes the patient's data and calculates the probability (risk) of them getting a stroke. We tested our model on a 49-year-old male who has a history of heart disease, never smoked, married, has relatively high average glucose level and normal BMI. The model predicted the probability of the patient having a stroke about 28% which is not too high but also definitely not low!

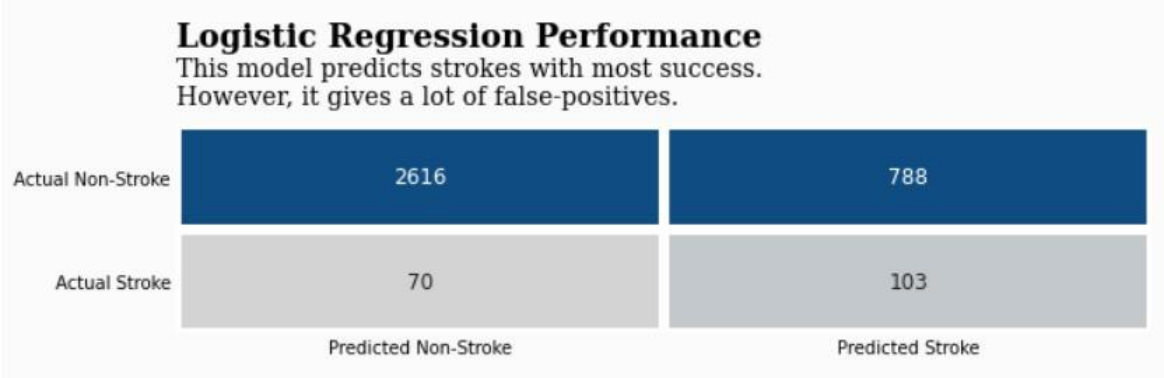
Comparisons to other models:

The following confusion matrices are of other Kaggle models that either use different modelling techniques, or logistic regression with different feature engineering and preprocessing. As per the matrices and our intent behind our modeling choice technique our model seems to be of comparable quality to other logistic regression models, and better performing in recall than other models.

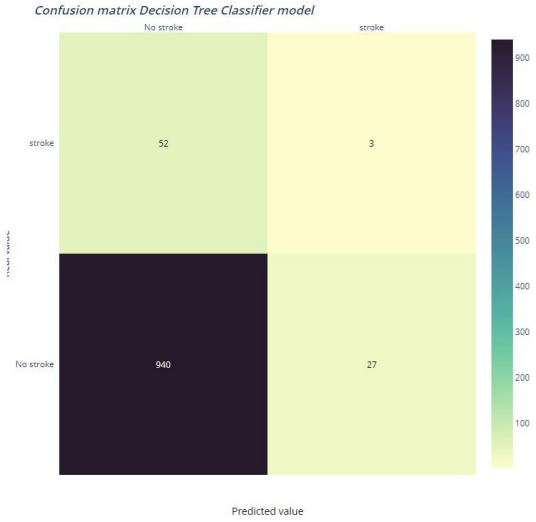
1-Random-Forest Model by MITESH SONI (The highest on the hottest ranking)



2-Logistic Regression Model by JOSH(one of the most voted models)



[3-Decision Tree Model by RACHIDYZ] (one of the most voted models)



Tips and Improvements

Our model achieved fairly good results on the given dataset. However these results could be improved by:

- Using a larger dataset with diverse patients from different cities and countries, this will allow our model to learn from hidden patterns and anomalies that research data couldn't find yet.
- Collecting more data about the patients and trying to find the inner relationship between the data and how to apply feature transformation based said relationships