Stroke Prediction

A project report

Ashika Mattu

PROBLEM STATEMENT

Application of Machine Learning to the health care domain has made a significant impact on people. Most of the issues can be resolved through early diagnosis of diseases and necessary steps can be taken to avoid severe conditions. Nowadays, heart strokes are being witnessed largely due to unhealthy lifestyle and also certain external factors. By assessing the environment of an individual, we can detect whether the person will face any risk of getting strokes in the near future. They can be notified earlier and be advised to follow the recommendations provided by health care professionals.

In this project, the main aim is to predict whether a person is likely to have a heart stroke or not, by assessing various parameters that affects the health of a person. With relevant data in hand, we can extensively use various Machine Learning algorithms to obtain insights and unlock hidden patterns that can assist in making accurate predictions. Some of the features available in this dataset are smoking status, work type, blood glucose level and so on that have direct impact on the probability of getting a stroke.

Similar prediction models can also be incorporated in smart watches and cellphones that continuously collect user data such as pulse rate, daily activity to monitor their health. This can be a large-scale project if real time data is utilized to build the machine learning models as opposed to relying on synthetic data.

KEYWORDS

KNN, Prediction, Accuracy, Stroke, Lifestyle

DATA EXPLORATION

This dataset is a supervised binary classification problem where the main task is to predict the column 'stroke' as 0 or 1 where 0 means the person is not likely to get a stroke and 1 being the person more susceptible to having a stroke. We can get great insights and visualize some the raw patterns in data by plotting graphs. From Figure 1, it can be very easily determined that the number of instances where stroke = 0 is more than that of stroke = 1. This indicates that the dataset is imbalanced with a greater number of occurrences of no stroke which will lead the machine learning model to learn patterns of only this particular class and not pick the patterns behind the scenario of having a stroke.

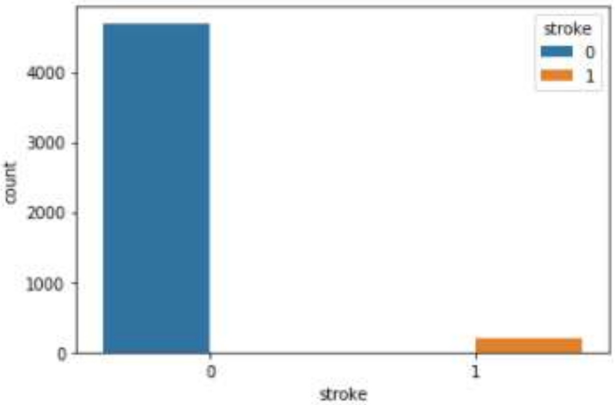

Figure 1: Graph depicts count of stroke =1 and stroke = 0

In Figure 2, we can note that the majority of the values of the columns 'bmi' which is basically Body Mass Index, are between 20 and 30 as the graph has its peak around this range of values.

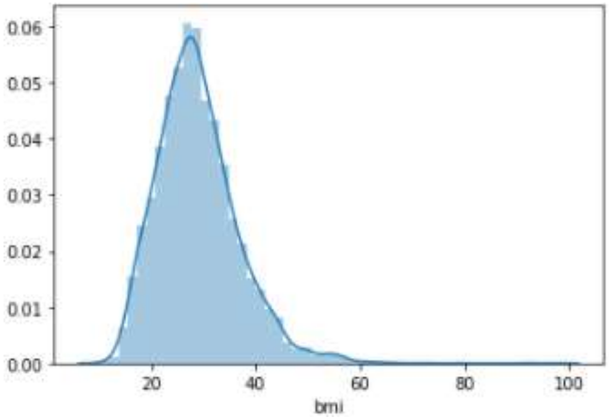

Figure 2: Distributed plot of feature 'bmi'

The next graph in Figure 3 shows that the features 'bmi', 'age' and avg_glucose_level' shows a linear relationship with stroke. In other words, we can state that the probability of getting a stroke increases as the values of BMI, age and average glucose level increase.

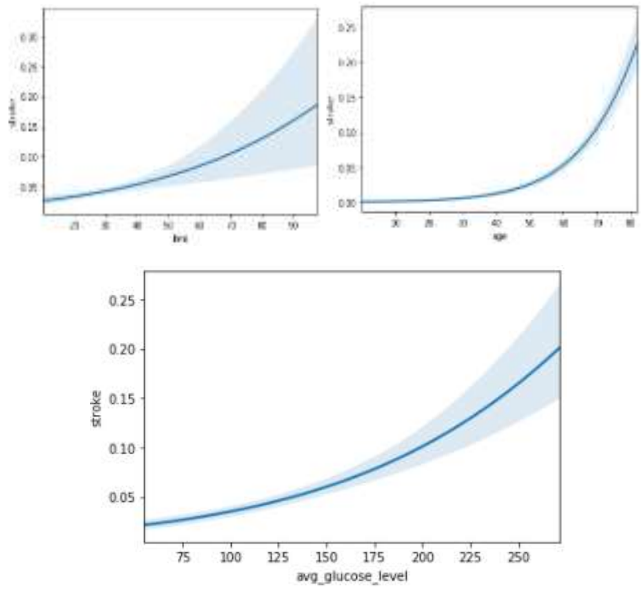

Figure 3: Regression plot of BMI, Age and Glucose level v/s stroke

To gain some more insights, plotting a bar graph of 'ever married' v/s 'stroke' as shown in Figure 4 depicts that people who were married at some point have less chances of getting a stroke. But this conclusion is not reliable as the dataset is imbalanced with less instances of people having a stroke.

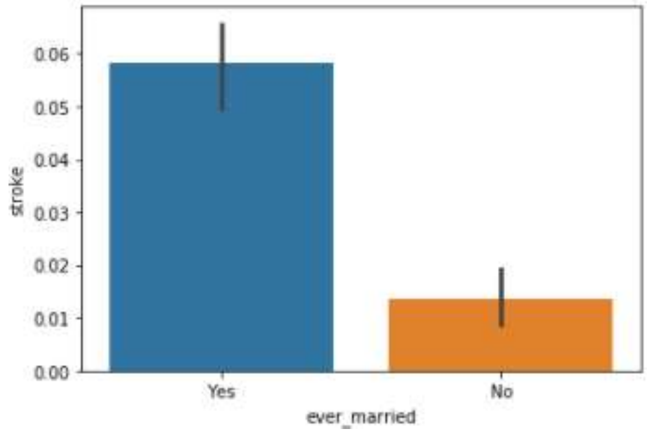

Figure 4: Regression plot of BMI, Age and Glucose level v/s stroke

DATA PREPARATION

The quality of the data plays an important role in the performance of the Machine Learning models. Therefore, before creating the models it is essential to prepare them so that it is relevant to the underlying problem. There are a total of 11 features available in the raw dataset. They are: 'id', 'gender', 'age', 'hypertension', 'heart_disease', 'ever_married', 'work_type', 'residence_type', 'avg_glucose_level', 'bmi', 'smoking_status'. The main goal here is to predict the value of the column 'stroke' as either 0 or 1 as accurately as possible by assessing the understanding various other features.

The column 'id' contains serial number assigned to each sample and thus will not have any positive impact on the prediction if it is incorporated in the model. Thus, it can be dropped to reduce complexity and. Similarly, we can drop the only sample with 'gender' = 'other' as there is no enough data to consider this value.

The next task is to identify if any null values are present in any of the columns. It can be noted that 'bmi' has 201 missing values. There are many ways to fill these missing values such as replacing with mean or median and so on. This might not be accurate at all times so in this project, the rows containing these null values are dropped as they are less in number and will not cause any information loss.

The columns 'gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status' have categorical values and they need to label encoded/one hot encoded, based on the number of unique values in each column. Additionally, it can be noted that the feature 'work_type' has variety of values such as government job, private job or self-employed, children and not working which can complicate the model. Hence, it would be a good approach to group the values government job, private job or self-employed as 'working' and children and not working as the 'not working' class.

Similarly, 'smoking_status' column can also be grouped as 'smoking' and 'not smoking' for simplicity. Once this is achieved, they can be label encoded as 0s and 1s. Additionally, this column has some unknown values in it, which can be replaced with 'numpy.nan'. Then, a KNNImputer can be employed to assign values to these unknowns as either smoking or not smoking. This imputation will assign decimal values to the column which should then be rounded off.

As discussed, the dataset is imbalanced with only 249 instances of having a stroke but with 4861 cases of people without the likelihood of getting a stroke. This will cause the model to bias towards instances with no stroke and to avoid this, over sampling is used here in this project which will be discussed in detail in the following sections.

ALGORITHMS

This section discussed about three prediction models developed using well-known Machine Learning algorithms and Deep Learning principles.

Before diving into the specifics., it is worth noting some key methods that are followed before proceeding with model creation. Firstly, since the dataset is imbalanced Synthetic Minority Oversampling technique (SMOTE) is used to create synthetic data of the class that is less in number by closely analyzing the existing samples. This is a better approach as opposed to under sampling and duplicating data. Next, a pipeline is created with feature scaling and model initialization so as to create model after it is being scaled.

The very first model is developed using K Nearest Neighbor (KNN), a basic yet powerful classification algorithm. It works by plotting the new data in the data space and identifies 'K' nearest training data point(s). Then it assigns the majority class to which K nearest points belong to and assigns it to the new data point.

The model is trained with K=3 as it is considered as one of the best-practices to be followed. It performed pretty well with an accuracy of 87% with test dataset of size 1227. From 4908 samples in the training dataset, it can be observed that there were 3378 True Positives and 160 True Negatives. Although, there are some incorrect predictions with 141 False Positives and 2 False Negatives. On the other hand, the confusion matrix had only 113 False positives and 37 False negatives. The True Positive value of 1067 and True negative of 10 are not bad either but it is bet by the next model.

The next model is designed based on Support Vector Machines (SVM) which is considered to work best for small to medium size datasets for solving classification problem. The main ideology behind SVMs is that it tries to segregate the data points in the linear space by learning the best hyper plane that provides the best partition.

The model is trained using default parameters and it showed an accuracy of 91% with test dataset of size 1227. The confusion matrix also was very impressive as it had only 76 False positives and 35 False negatives. The graphs for the same are provided in the last section of the report and it gives better picture of the overall performance of the model.

The last model used here is based on Neural Network. Neural networks are effective when the dataset is huge and it mimics human brain to understand the underlying patterns. There are numerous hyperparameters that affect the performance of the neural network model and can be fine-tuned to arrive at optimal performance, given a good amount of training data.

In this project, Tensorflow keras is being used to build, train and test the model. A simple 4 hidden layer with 30-40 neuron architecture is built by using 'relu' activation and for the output layer, '**softmax**' is used as we have two classes – 0 and 1. The model was optimized using Adam optimizer with a learning rate of 0.01. The model did not perform as good as the other two models and achieved an accuracy of 80%. This is not a drawback of the neural network as the dataset being used is not huge enough to fully utilize the capabilities of the neural network.

Algorithm	**Precision**		**Recall**		**Accuracy**
	0	**1**	**0**	**1**	
KNN	0.97	0.08	0.9	0.21	87%
SVM	0.97	0.14	0.94	0.26	91%
Neural Network	0.97	0.11	0.81	0.48	80%

Table 1: Performance comparison of the algorithms

To better understand the performance, Table 1 can be referred to. Precision is the ratio of True Positive to the sum of True Positive and False Positive values. This metric depicts out of all the 1s predicted by the model; how many were in fact, 1. Recall is the ratio of True Positive to the sum of True Positive and False negative. This takes into consideration the samples that were 1 but wrongly predicted as 0 by the model.

Finally, we can conclude that SVM performs the best and it is further explained in the next section with relevant graphs.

BEST MODEL GRAPHS

As mentioned above, SVM performs the best while predicting stroke in people based on their habits and lifestyle. To better understand the model graphs are plotted as seen below.

Figure 5 depicts the Receiver Operating Characteristics (ROC) which is a plot of True Positive Rate (TPR) v/s False Positive Rate (FPR). TPR is the ration of True Positive to the sum of True Positive and False Negative. FPR is the ratio of False Positive to sum of True Negative and False Positive. Since the ROC curve (blue line) is not linear, it can be understood that the model is able to distinguish the two classes 0 and 1 with higher accuracy.

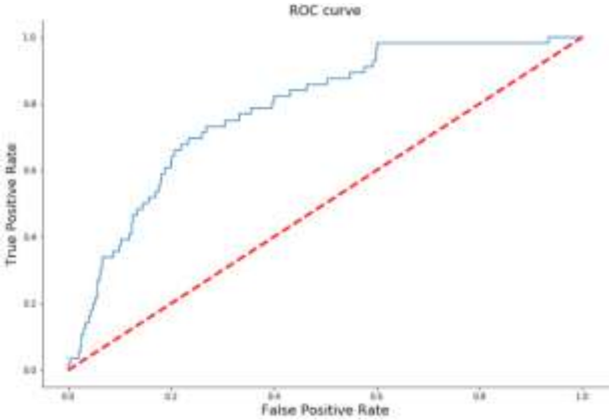

Figure 5: ROC of SVM model

The confusion matrix for the SVM model discussed in the previous section can be visualized as shown in Figure 6.

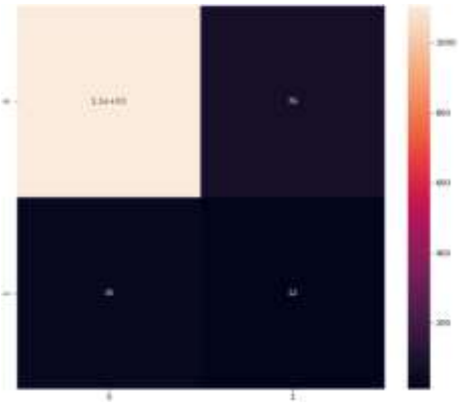

Figure 6: Confusion Matrix of SVM Model

For various thresholds, the precision-recall curve depicts the tradeoff between precision and recall. A large area under the curve indicates good recall and precision, with high precision indicating a low false positive rate and high recall indicating a low false negative rate. High scores for both indicate that the classifier is producing accurate (high precision) results as well as a majority of all positive outcomes (high recall).

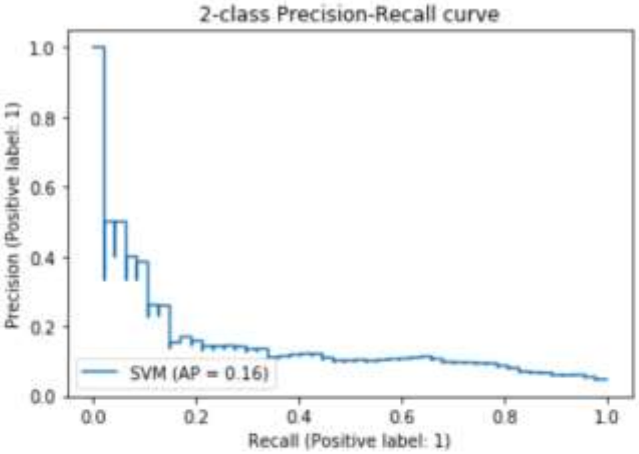

Figure 7: Precision v/s Recall curve.

The training and validation scores of an SVM for various values of the kernel parameter gamma are shown in this plot. If the gamma values are very low, both the training and validation scores are low. This is referred to as underfitting. High values for both scores will arise from medium gamma values, indicating that the classifier is working rather well. If gamma is set too high, the classifier will overfit, resulting in an excellent training score but a low validation score.

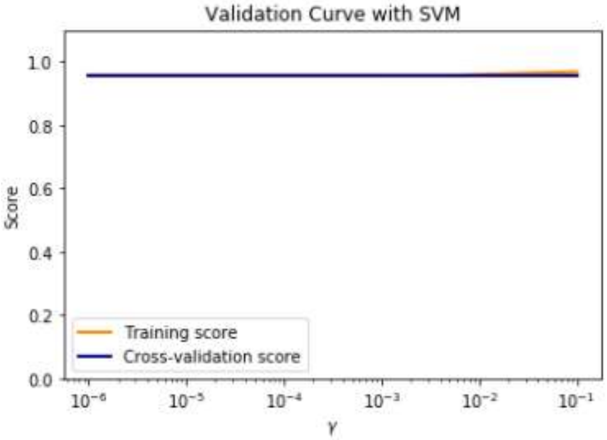

Figure 8: Comparison of Training score and cross-validation score

From Figure 9 it can be seen that age has the highest correlation with stroke.

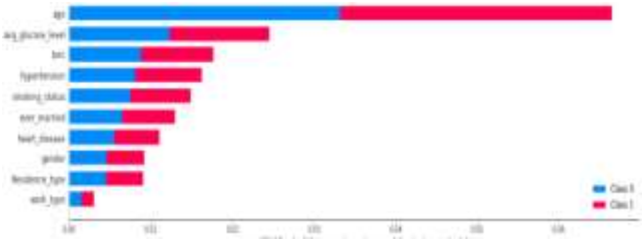

Figure 9: Correlation of features with output variable stroke.