2022

Portfolio 5

MACHINE LEARNING

NITHIN MATHEW JOSEPH

ASTON UNIVERSITY

Contents

Abstract:

The key question of this task is to build a model that can detect stroke in a patient based on the feature set provided. So, the first approach in this given scenario will be to identify the main ingredients for the user-friendly machine learning model. As the raw data provided is supervised, there is no need for applying any unsupervised or reinforcement model. Whereas dimensionality reduction can be applied to improve the performance of the supervised machine learning model.

Code Justification:

The first step in the algorithm is to read the raw data given and converting it into data frame for ease of operation. It is followed by identifying the type of info contained in the data frame. Since the data types are mainly integer and float, there is no further action required.

Next step is to check if there are any missing datapoints. When they are identified, it was removed using pandas function dropna.

Based on the research data provided by the client, there are some facts which states what kind of categorical data the data frame needs to be converted into. For example, the total cholesterol column in the data frame can be divided into desirable, borderline high, and high depending on milligrams (mg) of cholesterol per decilitre (dL) of blood.

	RANDID	TOTCHOL	AGE	SYSBP	DIABP	TIMEMI	CIGPDAY	TIME
1	6238	Borderline	58.00000	Normal	Normal	8766.00000	0.00000	4344
2	11252	Borderline	58.00000	Normal	Normal	8766.00000	0.00000	4285
3	11263	Borderline	55.00000	Normal	Normal	8766.00000	0.00000	4351
4	12806	Borderline	57.00000	Normal	Normal	8766.00000	0.00000	4289
5	14367	Desirable	64.00000	Borderline	Normal	8766.00000	18.00000	4438
6	16365	Borderline	55.00000	Normal	Normal	8766.00000	0.00000	4368
7	16799	Borderline	62.00000	Hypertensive	Borderline	8766.00000	30.00000	4431
8	23727	Borderline	53.00000	Normal	Normal	5592.00000	0.00000	4503
9	24721	Desirable	51.00000	Borderline	Normal	6411.00000	20.00000	4408
10	33077	Borderline	60.00000	Normal	Normal	8766.00000	0.00000	4383
11	34689	Borderline	49.00000	Normal	Normal	8766.00000	0.00000	4289
12	36459	Borderline	53.00000	Normal	Normal	8766.00000	0.00000	4411
13	40435	Desirable	54.00000	Normal	Normal	8766.00000	0.00000	4372
14	43522	Borderline	55.00000	Normal	Normal	8766.00000	0.00000	4403
15	43770	Borderline	64.00000	Normal	Normal	6384.00000	0.00000	4375
16	45464	Borderline	64.00000	Hypertensive	Normal	8766.00000	0.00000	4368
17	47561	Borderline	56.00000	Normal	Normal	8766.00000	0.00000	4071
18	55965	Borderline	72.00000	Hypertensive	Normal	8766.00000	0.00000	4438
19	63156	Borderline	47.00000	Borderline	Normal	8766.00000	0.00000	4248
20	66472	High	72.00000	Normal	Normal	8423.00000	0.00000	4347

Figure 1

This conversion is implemented using pandas function called cut. Similar concept is applied for diastolic and systolic blood pressure.

Total Cholesterol Level	**Category**
Less than 200mg/dL	Desirable
200-239 mg/dL	Borderline high
240mg/dL and above	High

Figure 2

Then the categorical data is converted to integer format. This is done with the help of the sklearn.preprocessing module called LabelEncoder.

As mentioned in the task, segregating the dataset into well and at-risk dataframes. This is made possible using the pandas function loc and by applying conditional operation on the stroke dataset.

A person with stroke data as 2 has a high chance of having another stroke. Whereas that is not the case for an individual with data as 1. Using this methodology data points are classified into healthy and at-risk datasets.

Creating individual function for each classification algorithm. The first two are based on Naïve Bayes model, that is Multinomial and Gaussian.

The other three are decision tree, multi-layer perceptron (neural network) and random forest classifier. Turning back to the pre-processed data, the requirement suggests that the data be separated into observation and response dataset. If we use RANDID it might give better performance for one of the models but it won't be based on relevant factors that affect a patient having stroke. Therefore, dropping RANDID from the observation dataset.

Each model performance is evaluated and the one with best score is observed for random forest model. The partitioned dataset is also run on the same model which shows best performance for both healthy and at-risk datasets.

To improve the model performance, recursive feature elimination is applied. This step is implemented with the help of Principal Component Analysis (PCA). The dataset is first standardised using sklearn.preprocessing module called standard scaler. Once it was standardised, PCA is applied by removing the least significant feature.

The least significant data is the age, in Article 1 it is mentioned that some risk factors like age cannot be controlled.

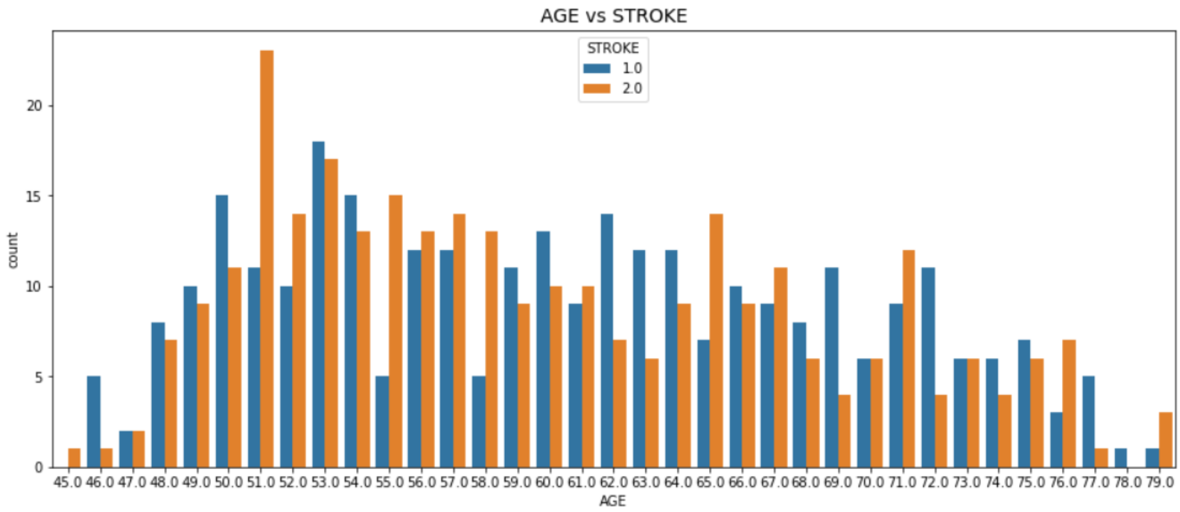

Figure 3

From the plot we can deduce that the set of data for each age group is diverse. An explanation might be that the dataset taken for stroke of one set of age group might be due to pure coincidence, where most of the individuals might have had a stroke, which will not be a proper datapoint for this analysis. Hence, PCA is applied after dropping age dataset. This resulted in better performance for all models except for random forest model. Random forest accuracy remains the same.

Furthermore, for the next least significant feature, I took time of first angina/spasm which is basically time data, that will have no correlation with the other set of features. So, when it was dropped the performance of the models increased, except for random forest and decision tree model.

When removing other features, the performance started decreasing. So, I plotted graphs of each feature against stroke to understand the reason as to why. Initially, I plotted total cholesterol level against stroke. From that I identified the borderline category of individuals have stroke, and this set of information felt informative. Similarly, when plotting for cigarettes per day or Time from baseline to first hypertension or body mass index against stroke. This convinced me that these features are significant which aligned with the result.

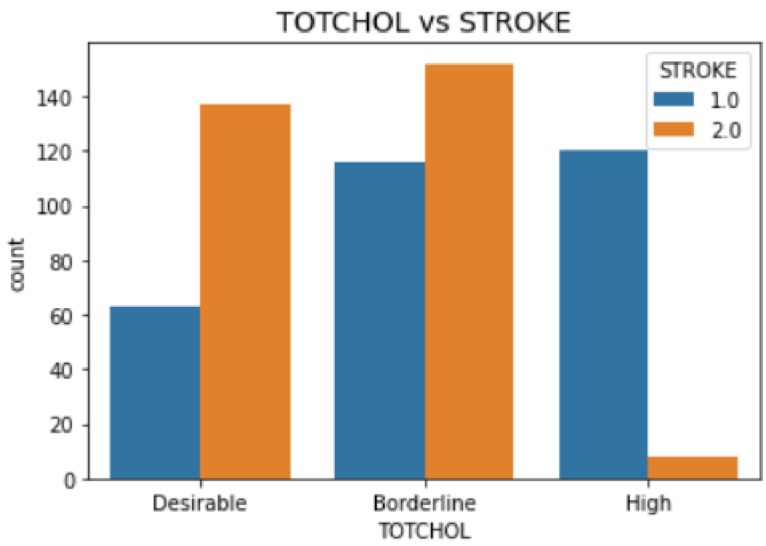

Figure 4

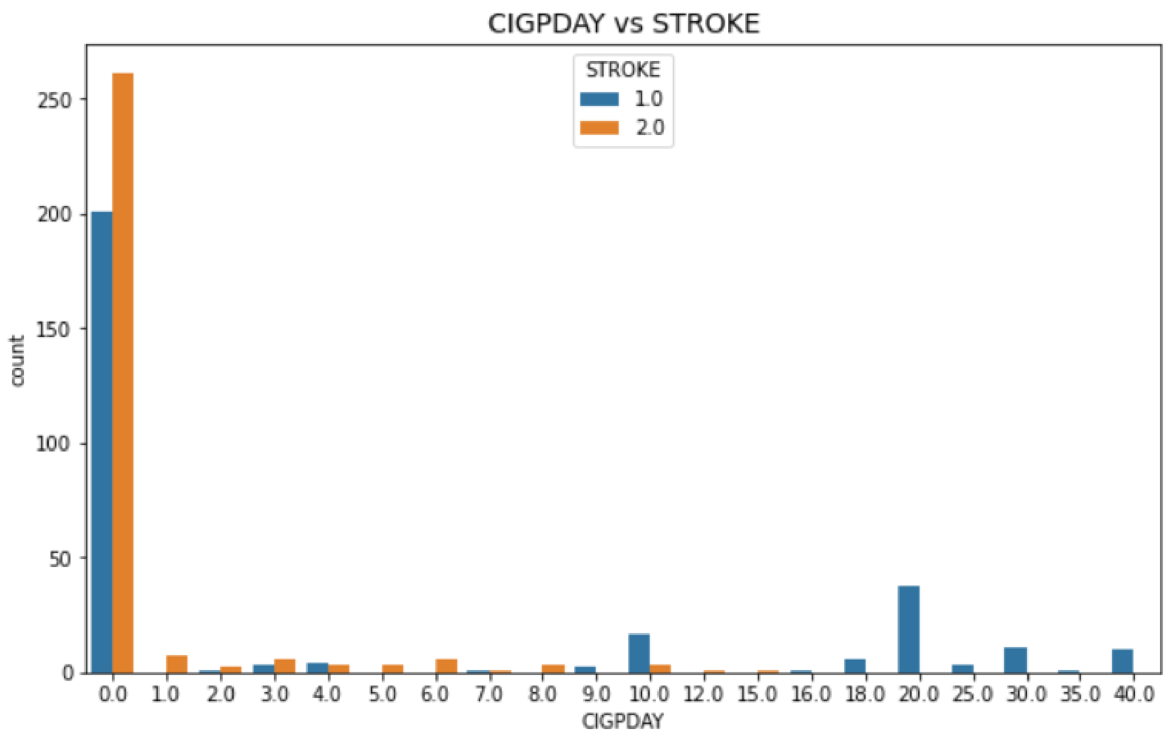

Figure 5

Hence the dataset has 6 significant features and based on recursive feature elimination, the performance showed significant improvement in neural network whereas best performance was observed for random forest model. But when I had plotted diastolic BP or Systolic BP against stroke it did not make much sense, cause the individuals with normal diastolic and systolic blood pressure had more cases of stroke.

As it made no sense, I decided to try another approach, where I merged the data given for systolic and diastolic blood pressure into one. That is blood pressure based on the research data provided in one of the articles. Wherein after label encoding, conditional operation was applied on systolic and diastolic datapoints. Meaning if either one had a higher threshold than the later, the higher value was initialised and if equal either value was taken.

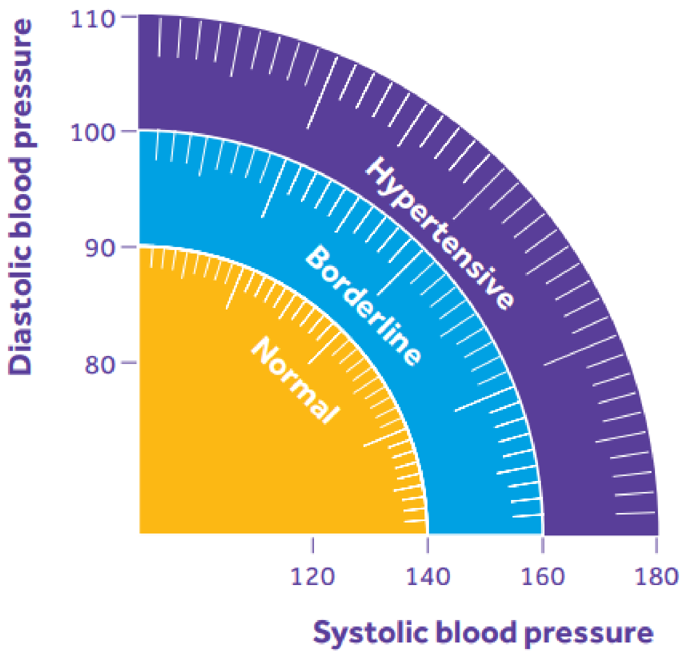

Figure 6

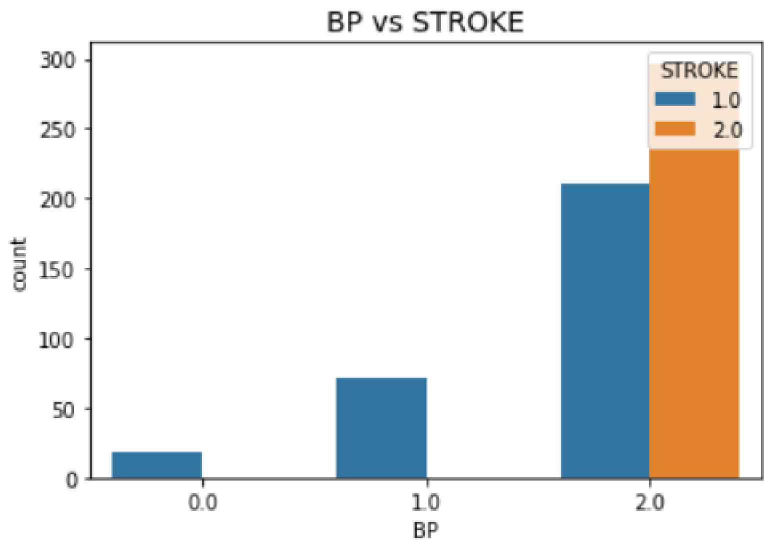

Figure 7

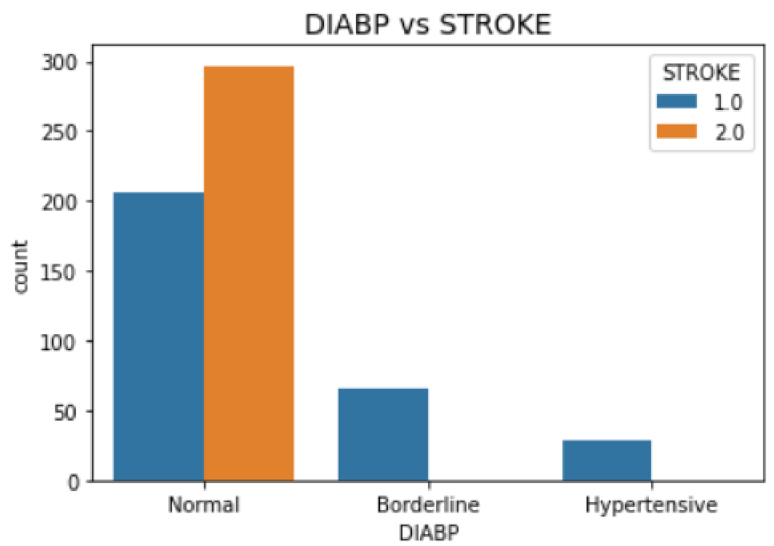

Figure 8

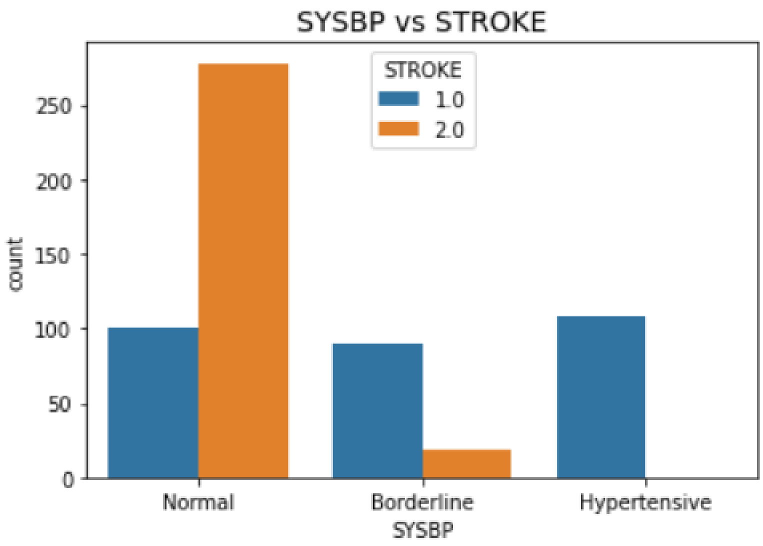

Figure 9

Now if we compare the plot of blood pressure against stroke versus diastolic or systolic blood pressure against stroke. An observation would be that the individuals with high blood pressure or hypertension have more chances of stroke which is opposite to what is being observed for diastolic or systolic blood pressure. This makes much more sense and its useful in prediction.

Once pre-processed, classification algorithms were run on it, and I observed 100 percent accuracy for decision tree model and random forest even without applying dimensionality reduction.

Conclusion:

From the above case study, the first focus was removing the missing data and pre-process the raw dataset to better fit the model. Then applied the train_test_split module, in order to predict (with reasonable accuracy) whether or not a new patient (i.e., one that is not in the dataset) was at high risk of stroke. Secondly, checked whether the solution performs well for both well and at-risk patients. By applying recursive feature elimination, the performance of some models increased. I had tried logistic regression since it gave a mediocre result did not proceed forward with it. Tried cross validation, since it represents the same conclusion as accuracy, confusion matrix and classification report did not feel the need to use it. I also merged the systolic and diastolic blood pressure information to get a better performance out of the models and it works.