Predicting Probability of Stroke

Using Machine Learning techniques with PySpark

Debadrito Saha
Roll-A20007
DS'July-Kol-2021

Index

Abstract

A stroke is a sudden interruption in the blood supply of the brain. Most strokes are caused by an abrupt blockage of arteries leading to the brain (ischemic stroke). Other strokes are caused by bleeding into brain tissue when a blood vessel bursts (hemorrhagic stroke). Because stroke occurs rapidly and requires immediate treatment, stroke is also called a brain attack. According to the World Health Organization, ischemic heart disease and stroke are the world's biggest killers.

In this project we will try to predict the probability of an observation belonging to a category (in our case probability of having a stroke i.e. 1 and not having a stroke i.e. 0.

Project Goal

- We will consider Stroke as our target variable
- We have a historic dataset containing Stroke and non-Stroke patients with their individual health,lifestyle and demographic parameters . We will build classification Machine Learning models, using the data set.
- Finding the best model among the test models, will help us to predict whether a person is prone to incur a stroke or not by considering above mentioned factors so that the person can take proper measures to avoid future risk of strokes.

Data Set Information

Independent Variables

1. **Age**
2. **Gender**
3. **Hypertension**
4. **Heart Disease**
5. **Marital Status**
6. **Work Type**
7. **Residence**
8. **Average Glucose Level**
9. **BMI**
10. **Smoking Status**

Target Variable

- **Stroke :** ‘1’ for yes ‘0’ for no

[redacted]

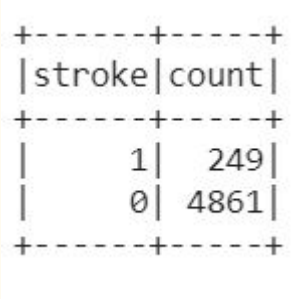
```
root
 |-- id: integer (nullable = true)
 |-- gender: string (nullable = true)
 |-- age: double (nullable = true)
 |-- hypertension: integer (nullable = true)
 |-- heart_disease: integer (nullable = true)
 |-- ever_married: string (nullable = true)
 |-- work_type: string (nullable = true)
 |-- Residence_type: string (nullable = true)
 |-- avg_glucose_level: double (nullable = true)
 |-- bmi: string (nullable = true)
 |-- smoking_status: string (nullable = true)
 |-- stroke: integer (nullable = true)
 |-- _c12: string (nullable = true)
```

[redacted]

Exploratory Data Analysis (spark.sql)

Rows:- 5110
Columns or Features - 13

Number Stroke and non-Stroke



```
+------+-----+
|stroke|count|
+------+-----+
|     1|  249|
|     0| 4861|
+------+-----+
```

So number of non-Stroke patients ('0') is more compared to Stroke patient ('1')

Observation:-

As can be seen from this observation. This is an **Imbalanced dataset,** where the number of observations belonging to one class is significantly lower than those belonging to the other class. In this case, the predictive model could be biased and inaccurate

Exploratory Data Analysis (spark.sql)

```
+-------------+---------------+
|    work_type|work_type_count|
+-------------+---------------+
|      Private|            149|
|Self-employed|             65|
|     Govt_job|             33|
|     children|              2|
+-------------+---------------+
```

OBSERVATION : Private occupation is the most dangerous work type in this dataset.

```
+------+------------+--------------------+
|gender|count_gender|             percent|
+------+------------+--------------------+
|Female|        2994|  58.590998043052835|
| Other|           1|0.019569471624266144|
|  Male|        2115|   41.3894324853229|
+------+------------+--------------------+
```

OBSERVATION : 59% of all people are Female and only 41% are Male that participated in stroke research.

Exploratory Data Analysis (spark.sql)

```
+------+-------------+----------------+
|gender|count(gender)|      percentage|
+------+-------------+----------------+
|  Male|          108|5.10638297872340|
+------+-------------+----------------+
```

OBSERVATION : 5% Male have had a stroke.

```
+------+-------------+----------------+
|gender|count(gender)|      percentage|
+------+-------------+----------------+
|Female|          141|4.70941883767535|
+------+-------------+----------------+
```

OBSERVATION : 4.7% Female have had a stroke.

Exploratory Data Analysis (spark.sql)

age	age_count
78.0	21
79.0	17
80.0	17
81.0	14
57.0	11
76.0	10
68.0	9
74.0	9
63.0	9
82.0	9
59.0	8
77.0	8
71.0	7
58.0	7
69.0	6
70.0	6
72.0	6
61.0	6
54.0	6
75.0	6

only showing top 20 rows

From age vs number of patients who had stroke, we can see that as the age is decreasing the number of stroke patients is also decreasing , age has positive relationship with stroke.
This indicate that elder people are more prone to incur strokes

```
[153] train.filter((train['stroke'] == 1) & (train['age'] > '50')).count()
```

```
226
```

OBSERVATION: using filter operation to calculate the number of stroke cases for people after 50 years.

As we can see Age is an important risk factor for developing a stroke.

Data Pre-Processing (Null Operation)

Checking Null values

id	0
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
bmi	201
smoking_status	0
stroke	0

Only BMI has null values

- For BMI we will compute the mean value and replace the null values with the mean

```
# fill in miss values with mean for feature "bmi"

from pyspark.sql.functions import mean
mean = train.select(mean(train['bmi'])).collect()
mean_bmi = mean[0][0]
train_f = train.na.fill(mean_bmi,['bmi'])
```

One Hot Encoding (for categorical variable)

One hot encoding is a process by which categorical variables are converted into a binary form that could be provided to ML algorithms to do a better job in prediction.

Categorical variables have to be encoded , we have used VectorAssembler,OneHotEncoder and StringIndexer module from 'pyspark.ml.feature' library .

Categorical Variables:

1. Gender
2. Ever_married
3. Work_type
4. Residence
5. Smoking Status

Code:

```
from pyspark.ml.feature import OneHotEncoder
encoder = OneHotEncoder(inputCols=["genderIndex","ever_marriedIndex","work_typeIndex","Residence_typeIndex","smoking_statusIndex"],
                        outputCols=["genderVec","ever_marriedVec","work_typeVec","Residence_typeVec","smoking_statusVec"])
```

Train-Test Split

The **train-test split** is a technique for evaluating the performance of a machine learning algorithm.The procedure involves taking a dataset and dividing it into two subsets.

- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.

We have split the data into 7:3 ration so 70 % of the data will be used to train the model (train_data) and 30% of the data will be used to test the model (val_data). We **seed** the split to a pseudo-random number =100 so that we get same outcome everytime we run the ML models.

```
[162] # splitting training and validation data

      train_data,val_data = train_f.randomSplit([0.7,0.3],seed=100)
```

Building Machine Learning Model

We have created **pipelines** for different models :
A machine learning **pipeline** is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a **model** that can be tested and evaluated to achieve an outcome, whether positive or negative.

We have used below machine learning techniques :

- Decision Tree (Non-Linear Model)
- Logistic regression (Linear Model)
- Random Forest (Ensemble Model)

Ensemble modeling is a process where multiple diverse models are created to predict an outcome, either by using many different modeling algorithms or using different training data sets. The ensemble model then aggregates the prediction of each base model and results in once final prediction for the unseen data.

Evaluation Metrics

We will use two evaluation methods to measure the performance of our model

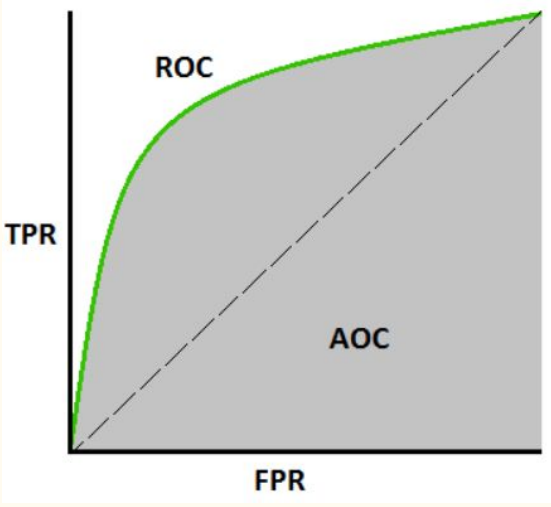
Accuracy Score:

Model accuracy is defined as the number of classifications a model correctly predicts divided by the total number of predictions made.

AUC :

AUC stands for "Area under the ROC Curve." it measures the entire two-dimensional area underneath the entire ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds. In the picture a area under the blue curve is AUC score . True Positive Rate vs False Positive Rate Curve.

As our data set is imbalanced and this is a classification problem so we will more focus on AUC score.



Model Evaluation

Decision Tree
Accuracy : 95.09%
AUC Score : 0.3578

Logistic Regression
Accuracy : 95.16%
AUC Score : 0.8009

Random Forest
Accuracy : 95.23%
AUC Score : 0.7826

If we look at the Accuracy scores of different Models, there is no significant changes happening. But if we focus on the AUC Scores we will observe Decision Tree has the least AUC which says that it is inefficient in classifying observation to Positive Class, i.e low bias high variance model.

Again if we focus on AUC of Random forest model it's quite higher than Decision tree, there by low variance.

Logistic Regression has the best AUC among the three models.Thereby its capability of classifying observations to positive class is higher and is not affected by the amount of positive observations present in the data set. As we are are focussing more on the Stroke positive classification as because misclassifying it may be more riskier thereby we are focussing more on True Positive Rate or sensitivity of the model.

Conclusion

The accuracy of Logistic model is 95.16 % so for 100 cases it can correctly predict 95.16 instances also the AUC score is highest for this , so this model will be our final model

Our project goal was to build a classification model to predict future chance of stroke depending upon individual health,lifestyle and demographic factors and it looks like we have successfully created one with a satisfactory performance score.

Conclusion

Post deploying the model if a person provide his/her health details the model will suggest whether he/she is stroke prone or not ,this way we can take necessary measures to avoid strokes which will eventually increase life expectancy and will save a lot of money for treatment.

This project will help to increase awareness on strokes among individuals and help them to adopt healthy lifestyles.