RESEARCH

Introduction to the profile areas of data sciences: project 4

Rohan Chitte] Natalja Amiridze] Michael Onyebuchi Ohaya

Abstract

Goal of the project:
The first goal of this project is to perform data loading and preprocessing with Spark/Spark SQL and collecting statistical information about the data.
The second goal is to develop two predictors that estimate the probability of stroke in a patient and evaluate the predictors using a balanced and unbalanced dataset.

Main results of the project: We determined which features have most impact in causing stroke.

Personal key learnings: We learned how to implement data imputation, statistical analysis as well as make predictions on dataset using Spark.

Estimated working hours: 12

Project evaluation: 1

Number of words: 1653 Words

1 Scientific Background

Stroke is a disease that affects the arteries leading to and within the brain. A stroke occurs when a blood vessel that carries oxygen and nutrients to the brain is either blocked by a clot or bursts (or ruptures). When that happens, part of the brain cannot get the blood (and oxygen) it needs, so the brain and its cells die. Mckinsey's healthcare has captured several health, demographic and lifestyle details about its patients. The goal is to find which features of the patients are classified as high risk factors of stroke and to make predictions of stroke occurring on future patients.

2 Goal

Major goal of the project was to get oneself acquainted with the Spark/Spark-SQL framework, gather some statistical information from the dataset, perform data manipulation using the Spark-SQL method, establish related features in the dataset. Finally, three predictions were performed using the unbalanced dataset and thereafter the balanced dataset. The prediction algorithms used are the Spark Logistics regression, Random Forest and Support Vector Machine.

3 Data and Preprocessing

The training and testing dataset was read in comprising of 42,400 and 18601 respectively. Data taken from two major groups(stroke and non-stroke) and age ranging

from 01.8 - 82 years. Different features were observed and recorded.Data preprocessing done includes filling null values and One-hot-encoding on categorical data using .na.fill() and .OneHotEncoder(). In order to read and manipulate data we imported pyspark. Firstly, test and train csv files were loaded using spark.read() and previewed using show(). To find null values in the datasets, we used isNull() on each column. We then performed data imputation. For columns that had string data type null values, na.fill() was used to replace them to a specific string value. Where as for columns that had integer values, a mean value was first calculated using mean() and was filled in place of null values. We converted the spark dataset into pandas dataframe using toPandas() and imported missingno and matplotlib package. This converted pandas dataframe and imported packages were then used to execute mno.matrix(), which plots a visual image showing column wise null values of the entire dataset. To make probabilistic predictions using the unbalanced data first and then the balanced data, some steps were employed (string indexing, one hot encoding and generating a pipeline) The .StringIndexer() indexes all categorical column, .OneHotEncoding() convert the generated StringIndex to binary values

4 Methods

4.1 Task 1

To obtain the summary statistics we used .describe() which creates an output table containing count, mean, stddev, min and max value of each column. Next, to perform feature analysis, we did SQL Querying on individual features to test their influence on stroke using spark.sql(). Although, Before we executed SQL queries, we transformed the dataframe into a table as a temporary view using createOrReplaceTempView('table') to perform SQL queries. To get results in terms of visuals, we plotted bar graphs using plt.bar(). Finally, to compute correlation between two attributes we used spark's dataframe method stat.corr(). Similarly, to compute covariance between two classes we used stat.cov().

4.2 Task 2

Having collected some statistical information from the dataset,the next step was to make probabilistic predictions using the unbalanced data first and then the balanced data. To carry out predictions, some steps were employed (string indexing, one hot encoding and generating a pipeline. We used the .Pipeline() to wrap the complex processes that occurs during predictions (see notebook for specific information). Using the pipeline, three predictions were made and compared, namely .LogisticsRegression(), .RandomForest() and the .LinearSVC(). They were all used to predict the relationship between stroke and other features. Using .withColumn(), .unionAll(), .drop() and .filter() as well .col(), .explode(), .array(), .lit(), from pySpark the data were balanced by oversampling and undersampling. The already balanced data was then split using the .randomSplit() in the proportion of 0.7 for training and 0.3 for validation. Training and validation was done using the .fit() .transform() method respectively, see [figure 9] and [figure 12]. Accuracy of the model was obtained using the .MulticlassClassificationEvaluator() and evaluated using the .BinaryClassificationMetrics() on the probability and stroke columns. Finally, each of the model was used on the balanced test data to know its efficacy on data never seen before [figure 10] and [figure 13].

5 Results

5.1 Task1

The summary staististics for all the attributes is acquired using the describe() method [figure 1].

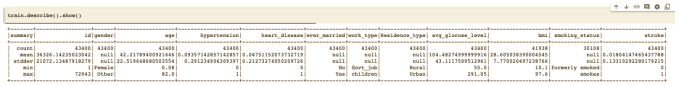

Figure 1 Summary Statistics

As a part of data exploration, we observed that the target distribution "stroke" is an imbalanced dataset, where the number of observations belonging to one class is significantly lower than those belonging to the other classes [figure 7]. Upon analysing different feature's distribution with target distribution, we concluded that maximum stroke occurred among people who were self employed. About 2% Males had stroke, when only 1.6% Females had stroke. So, Males had the higher tendency of getting stroke than females. While analysing the risk factor of getting stroke associated with age, we could spot that 91.5% stroke had occurred for person who are more than 50 years old. The occurrence of stroke was high among the people who had bmi count between 20 and 40. The influence of heart disease on triggering stroke was relatively much higher than that of hypertension. Married couples were also at a higher risk than those who were single. Further, smoking reduced the chances of getting stroke. Interestingly, the area of residence had less significance on influencing stroke. Data imputation was necessary for this dataset as there were 1462 and 13292 null values in attributes bmi and smoking status respectively [figures 2].

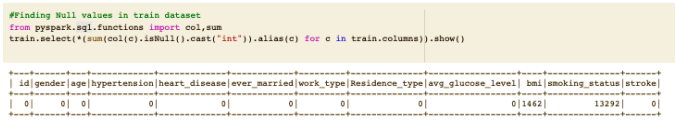

Figure 2 Null Value Count

In order to fill the null values in bmi, we calculated the mean value of bmi column [figure 3].

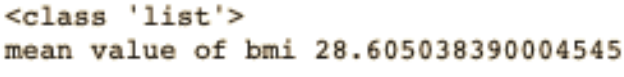

Figure 3 Mean value of bmi

However, for smoking status column, we filled the null values with string value "No

Info". and then replaced the null value with the calculated null value. These null values were filled/replaced using na.fill() method and the visualization of the result on the dataset can be seen in [figure 4].

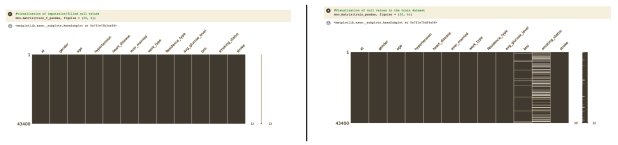

Figure 4 Visualization of data imputation on null values

Additionally we computed correlation and covariance to analyse further into the features. Evidently, Correlation and covariance was strongest between Age and Stroke. In Contrast, Correlation and covariance was relatively much weaker between Bmi and Stroke [Table 1 and 2].

Features	Correlation between Features and Stroke
Bmi	0.017
Age	0.155

Table 1 Correlation between Features and Stroke

Features	Covariance between Features and Stroke
Bmi	0.018
Age	0.457

Table 2 Covariance between features and Stroke

Figure 5 Caption

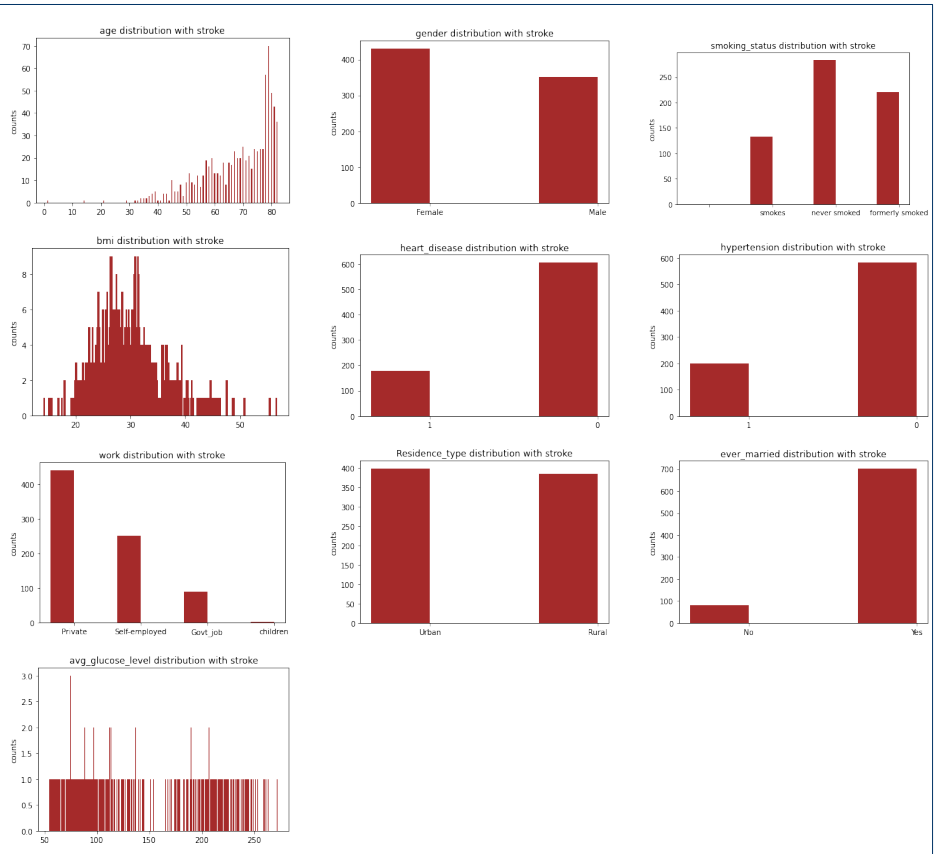

Figure 6 Barplots for distribution of features as results from SQL queries
1st row: age, gender, smoking_status
2nd row: bmi, heart_disease, hypertension
3rd row: work_type, residence_type, ever_married
4th row: average_glucose_level by patients with stroke

5.2 Task 2

Three classifiers Logistic Regression (LR), Random Forest (RF) and Support Vector Machine (SVC) were developed, evaluated and compared.The classifier should be able to predict risk of stroke for future patients based on clinical measurements such as hypertension, heart disease, age, smoking, gender, married status, work type, residence type, average glucose level, bmi for a number of patients as well as information about whether each patient has had a stroke or not. All classifiers show a high accuracy of 98.3%. This result may be incorrect as unbalanced data was used (see Figure 7) with very low stroke patient number compared with no. of patients without stroke.

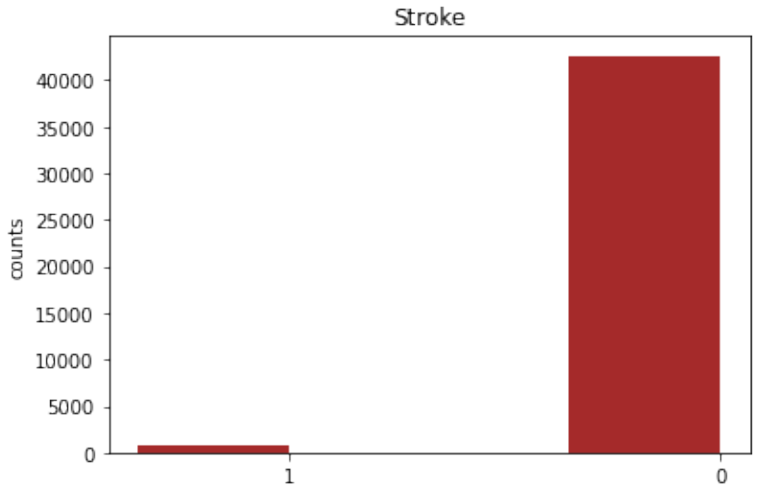

Figure 7 with stroke vs w/o stroke patient distribution

So accuracy can be misleading. Also, the three classifiers had high AUC of 86%, 81% and 82% respectively (see Figure 8).

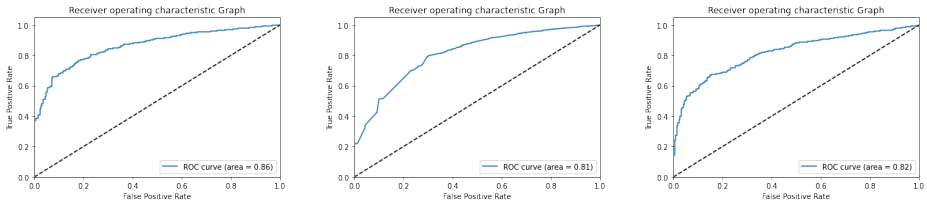

Figure 8 Roc Curve Plot of three developed models (left: LR ; right: RF Model svc) with unbalanced data

The results of validation after training the model show the table with predictions, probabilities, stroke as label and features list with gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi and smoking_status for given validation data (see Table 3).

Classifier	Logistic Regression	Random Forest	Support Vector Machine
Accuracy	98.27%	98.24%	98.30%
AUC	86.3%	80.9%	81.8%

Table 3 Important Classification Figures: accuracy, AUC for with unbalanced data

The excerpt of validation results are shown in the Figure 9 The test results show a table with possible predictions of stroke based on test data without a stroke label (see Figure 10).

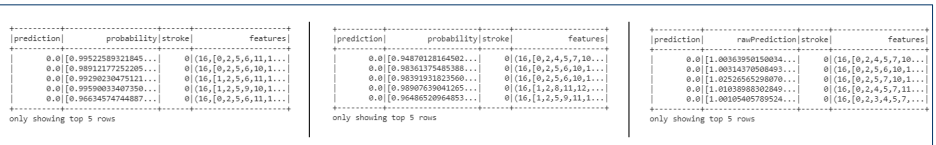

Figure 9 validation Results of three developed models (left: LR ; right: RF Model svc) with unbalanced data

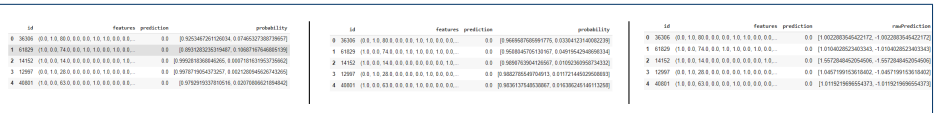

Figure 10 Test Results of three developed models (left: LR ; right: RF Model svc) with unbalanced data

All the results of classifiers with unbalanced data are good, but it is not the reality, because we have many more samples of one class than of another class, so the classifier trains much more with the samples of only one class. Hence, the results must be considered with caution. To get more reliable results, we trained the models of the classifiers with balanced data that we could generate by resampling. We generated balanced data with both oversampling and undersampling. The results of the classification with both data were very similar, so for better clarity only the results of the under sampling are shown here. The accuracy values were lower than with unbalanced data (see Table 4)

Classifier	Logistic Regression	Random Forest	Support Vector Machine
Accuracy	77.73%	77.32%	78.14%
AUC	85.45%	83.3%	85.5%

Table 4 Important Classification Figures: accuracy, AUC for with balanced data

and were roughly of 78% for all three classifiers. AUC values were slightly higher for unbalanced data and were 85.5% for LR and SVC and 83.3% for RF (see Figure 11).

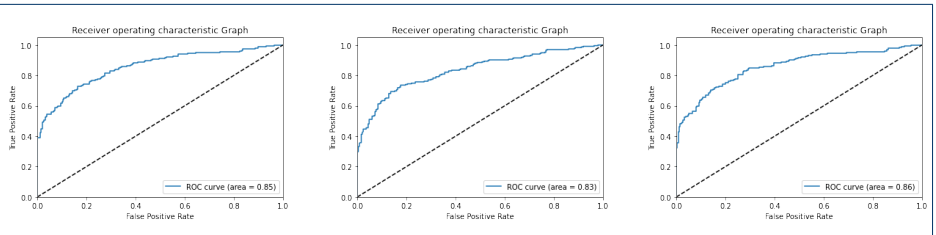

Figure 11 Roc Curve Plot of three developed models (left: LR ; right: RF Model SVC) with balanced data

The validation and test result extracts are shown in the figures 12 and 13.

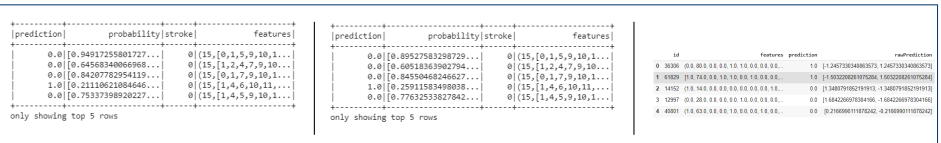

Figure 12 validation Results of three developed models (left: LR ; right: RF Model svc) with balanced data

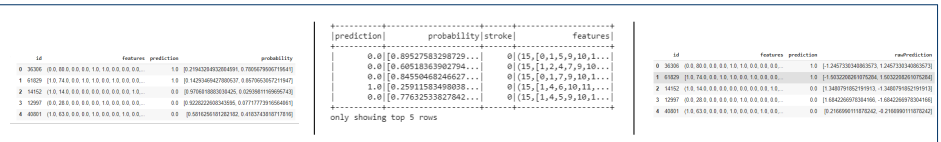

Figure 13 Test Results of three developed models (left: LR ; right: RF Model svc) with balanced data

These results are more reliable because the same number of samples with and without stroke are used.

6 Discussion

6.1 Results

Using Pyspark, we performed statistical feature analyses, cleaning of data and data imputation. We identified the impact of features such as age, bmi, work type and marital status on stroke. Other characteristics didn't appear to be risk factors. Due to data being unbalanced, the accuracy values were unreliable despite of being high for all classifiers. Since accuracy = correctly predicted instances/All instances, it is possible to get a decent accuracy while having mostly incorrect predictions for the minority class. One possible solution in this case is to resample the data to get more balanced data or to use classifiers that can be made aware of the imbalanced data by incorporating the weights of the classes into the cost function. LR with Cross entropy as a cost function for binary classification, also support vector machine and random forest may be adjusted for sensitive learning by setting the class_weight parameter by sklearn. Unfortunately, there is no parameter for LR, RF and SVC from pyspark.ml to set class_weight.

6.2 Importance of the project for datascience

This project is very useful for learning how to handle and process large data more efficiently by working with the database using Spark, SparkSQL that support the processing and analysis of the data. Since the amount of data is constantly growing, it is of great benefit for datascientists to have tools to handle and process big data efficiently.

7 Appendix

Name	Work Description
Michael	Goals, Data and preprocessing, Method of Tasks 2
Rohan	Methods and Results of Task 1, Code for Task 1, Scientific Background
Natalja	Results of Task2, Discussion, Code for Task 2 and some parts (barplots and queries) for Task1

Table 5 Task responsibilities

References