Stroke Probability Prediction from Clinical Measurements of Patients using PySpark

Big Data Project
PGDBD03

Shreyasi Chandra

Objective

- Predicting stroke probability and classifying patients based on clinical measurements: such as hypertension, heart disease, age, work type, smoking status, family history of disease
- Using SQL queries to return the result as dataframe
- Using ML models of PySpark for predicting the risk score of stroke, also to predict class labels for each patient id
- Using Different visualization methods to analyse the dataset and risk factors

Introduction

- According to WHO , ischaemic heart disease and stroke are the world's biggest killers
- Analysing the dataset to understand which factors contribute most to the outcome of 'stroke'
- Using clinical measurements to build ML models from PySpark Library
- Using classification algorithms to classify the outcome : 'stroke' or 'no stroke'
- Dataset: Healthcare Dataset Stroke Data (Kaggle)

Why we are using Pyspark

- Apache Spark is an open source framework, concise and easy to use and it allows us to use data in a distributed fashion
- Applications running on PySpark are 100x faster than traditional systems
- PySpark natively has machine learning and graph libraries which makes it very useful while performing prediction tasks on large data set

Dataset

- **Training dataset contains information about 43400 patients, out of which 726 patients had stroke attack.**

```
] train.toPandas().head(5)
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	30669	Male	3.0	0	0	No	children	Rural	95.12	18.0	None	0
1	30468	Male	58.0	1	0	Yes	Private	Urban	87.96	39.2	never smoked	0
2	16523	Female	8.0	0	0	No	Private	Urban	110.89	17.6	None	0
3	56543	Female	70.0	0	0	Yes	Private	Rural	69.04	35.9	formerly smoked	0
4	46136	Male	14.0	0	0	No	Never_worked	Rural	161.28	19.1	None	0

- **Test dataset contains 18601 patients**

```
test.toPandas().head(5)
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
0	36306	Male	80.0	0	0	Yes	Private	Urban	83.84	21.1	formerly smoked
1	61829	Female	74.0	0	1	Yes	Self-employed	Rural	179.50	26.0	formerly smoked
2	14152	Female	14.0	0	0	No	children	Rural	95.16	21.2	None
3	12997	Male	28.0	0	0	No	Private	Urban	94.76	23.4	None
4	40801	Female	63.0	0	0	Yes	Govt_job	Rural	83.57	27.6	never smoked

Data Exploration

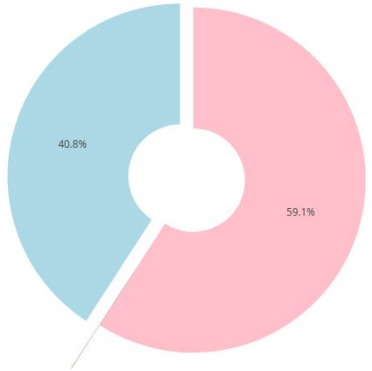
- In the training dataset 783 suffered from stroke, while 42617 didn't
- Explored if a certain 'work type' is more related to stroke probability
- Found people in 'private jobs' are more prone to 'stroke' as compared to people

```
+------+-----+
|stroke|count|
+------+-----+
|     1|  783|
|     0|42617|
+------+-----+
```

```
+-------------+---------------+
|    work_type|work_type_count|
+-------------+---------------+
|      Private|            441|
|Self-employed|            251|
|     Govt_job|             89|
|     children|              2|
+-------------+---------------+
```

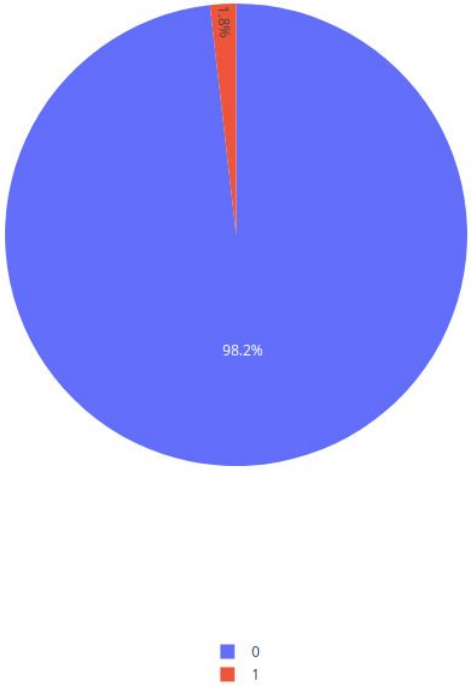
- We found almost 60% people were females while 40% were males

```
+------+------------+------------------+
|gender|count_gender|           percent|
+------+------------+------------------+
|Female|       25665|  59.13594470046083|
| Other|          11|0.02534562211981567|
|  Male|       17724|  40.83870967741935|
+------+------------+------------------+
```

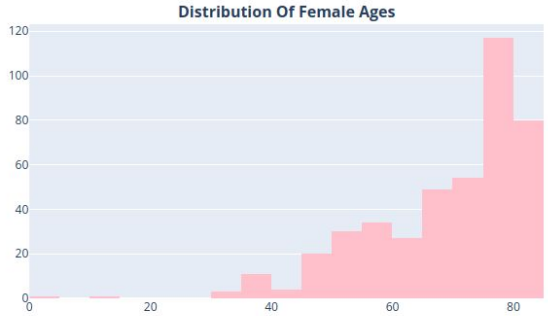


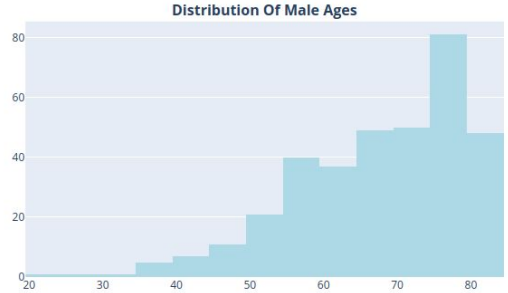
Visual Analysis of the Dataset

Proportion of Stroke Samples



Inference of age on Stroke positive samples

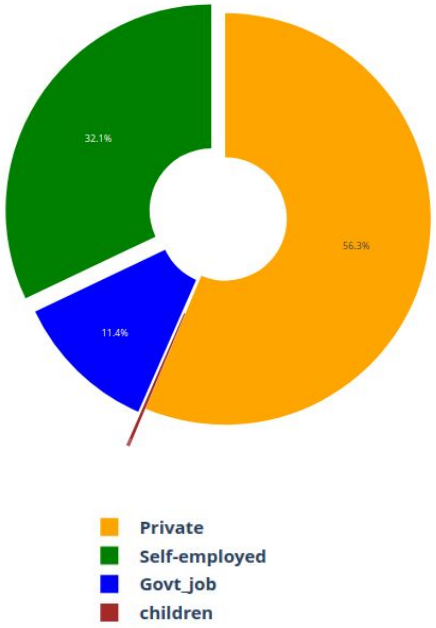




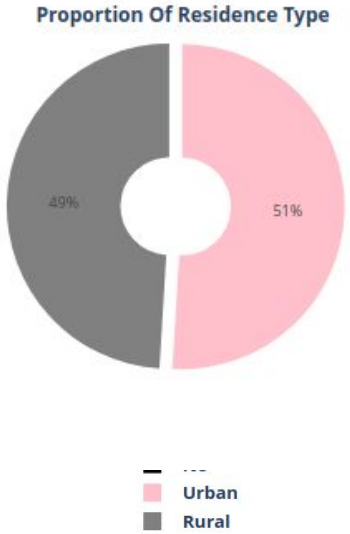
males are more prone to strokes in their early 50/60 where the median of the women stroke age is around 75-79

Visual Representation of Attributions of Stroke Samples

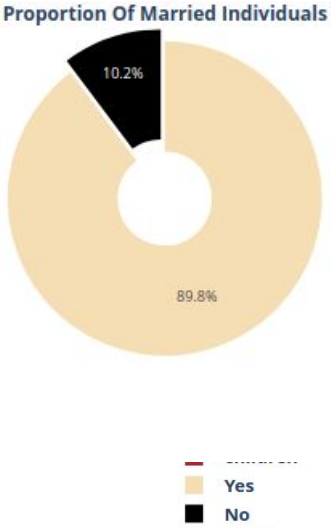
Different work types



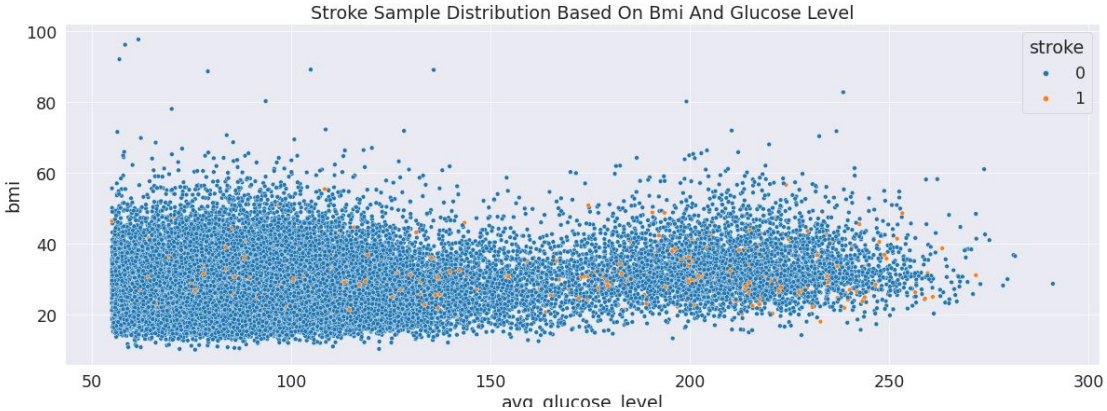
Residency type

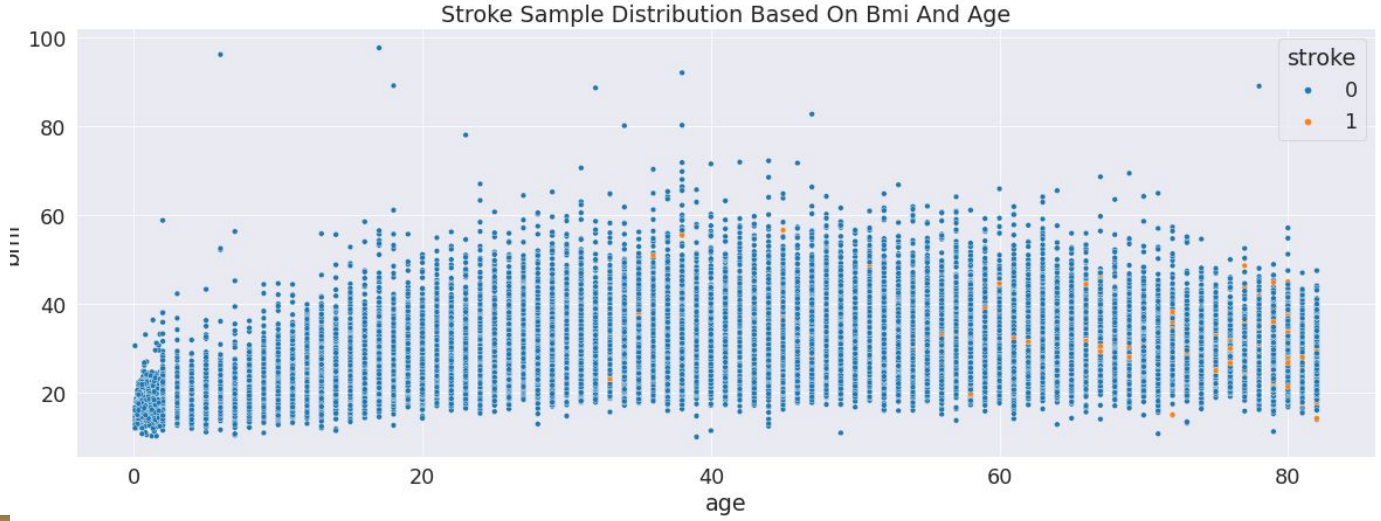


Marriage Status



Marriage status and the residence type are distributed in a way that doesn't tell us anything confounding about stroke-positive individuals, the residence type is close to a uniform distribution and the marriage status is almost completely dominated by one of the values.


Stroke Sample Distribution Based On Bmi And Glucose Level
stroke
0
1
bmi
avg_glucose_level
100
80
60
40
20
50
100
150
200
250
300


Stroke Sample Distribution Based On Bmi And Age
stroke
0
1
age
100
80
60
40
20
0
20
40
60
80

- **Among the males 1.98% had stroke**

```
# 1.98% males had stroke
spark.sql("SELECT gender, count(gender), (COUNT(gender) * 100.0) /(SELECT count(gender) FROM table WHERE gender == 'Male') as perc
```

```
+------+-------------+----------------+
|gender|count(gender)|      percentage|
+------+-------------+----------------+
|  Male|          352|1.98600767321146|
+------+-------------+----------------+
```

- **Among the females 1.67% had stroke**

```
#1.67% of females had stroke
spark.sql("SELECT gender, count(gender), (COUNT(gender) * 100.0) /(SELECT count(gender) FROM table WHERE gender == 'Female') as pe
```

```
+------+-------------+----------------+
|gender|count(gender)|      percentage|
+------+-------------+----------------+
|Female|          431|1.67932982661212|
+------+-------------+----------------+
```

- **Found almost 91.57% people over the age of 50 are suffering from stroke**

```
# 91% patients above age 50 suffered from stroke

spark.sql("SELECT COUNT(age)*100/(SELECT COUNT(age) FROM table WHERE stroke ==1) as percentage FROM table WHERE stroke == 1 AND ag
```

```
+-----------------+
|       percentage|
+-----------------+
|91.57088122605364|
+-----------------+
```

Data Cleaning and Feature Engineering

- There were some ‘missing values’ in smoking status data. This is a categorical data type so filled those values with ‘No info’
- There were some ‘missing values’ in ‘bmi’ column. It is a ‘numerical data type’, so filled these values with ‘mean bmi value’ of that column
- Most ML algorithms can’t work directly with ‘categorical data, so we encoded these features

StringIndexer -> OneHotEncoder -> VectorAssembler

- Indexed all ‘categorical columns’ like ‘gender’, ‘marital status’, ‘work type’, ‘smoking status’ with StringIndexer. StringIndexer is used as it converts the textual data to numeric data keeping the categorical context which is useful for ML algorithms

- **One hot encoded the ‘indexed data’**

```
# Doing one hot encoding of indexed data
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"]
```

- **Used VectorAssembler to merge all the output columns into a vector column**

```
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=['genderVec',
 'age',
 'hypertension',
 'heart_disease',
 'ever_marriedVec',
 'work_typeVec',
 'Residence_typeVec',
 'avg_glucose_level',
 'bmi',
 'smoking_statusVec'],outputCol='features')
```

Model Building

- Created a DecisionTree object using DecisionTreeClassifier

```
from pyspark.ml.classification import DecisionTreeClassifier
dtc = DecisionTreeClassifier(labelCol='stroke',featuresCol='features')
```

- Created a pipeline which consists of a sequence of pipeline stages to be run in specific order

```
from pyspark.ml import Pipeline
pipeline = Pipeline(stages=[indexer1, indexer2, indexer3, indexer4, indexer5, encoder, assembler, dtc])
```

- Splitted the dataset into training and validation data, then fitted the pipeline to the train_data

```
# splitting training and validation data
train_data,val_data = train_f.randomSplit([0.7,0.3])

# training model pipeline with data
model = pipeline.fit(train_data)
```

- **Transformed the validation data and used DecisionTreeClassifier model to predict on the validation set**

```
# making prediction on model with validation data
dtc_predictions = model.transform(val_data)

# Select example rows to display.
dtc_predictions.select("prediction","probability", "stroke", "features").show(5)
```

```
+----------+--------------------+------+--------------------+
|prediction|         probability|stroke|            features|
+----------+--------------------+------+--------------------+
|       0.0|[0.98181280982239...|     0|(16,[0,2,5,6,11,1...|
|       0.0|[0.98181280982239...|     0|(16,[0,2,6,10,11,...|
|       0.0|[0.98181280982239...|     0|(16,[0,2,5,6,10,1...|
|       0.0|[0.98181280982239...|     0|(16,[1,2,8,11,12,...|
|       0.0|[0.98181280982239...|     0|(16,[0,2,4,5,7,11...|
+----------+--------------------+------+--------------------+
only showing top 5 rows
```

- **Decision Tree algorithm had an accuracy of 98.23%**

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
# Select (prediction, true label) and compute test error
acc_evaluator = MulticlassClassificationEvaluator(labelCol="stroke", predictionCol="prediction", metricName="accuracy")
dtc_acc = acc_evaluator.evaluate(dtc_predictions)
print('A Decision Tree algorithm had an accuracy of: {0:2.2f}%'.format(dtc_acc*100))
```

```
A Decision Tree algorithm had an accuracy of: 98.23%
```

Result

- Used the Decision Tree classifier to predict the labels for test data baked on the clinical properties as mentioned
- Also predicted the probabilities for each class

```
# now predicting the labels for test data
test_pred = model.transform(test_f)
test_selected = test_pred.select("id", "features", "prediction","probability")
test_selected.limit(5).toPandas()
```

	id	features	prediction	probability
0	36306	(0.0, 1.0, 80.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0,...	0.0	[0.9818128098223958, 0.018187190177604 15]
1	61829	(1.0, 0.0, 74.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0,...	0.0	[0.9818128098223958, 0.018187190177604 15]
2	14152	(1.0, 0.0, 14.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0,...	0.0	[0.9818128098223958, 0.018187190177604 15]
3	12997	(0.0, 1.0, 28.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0,...	0.0	[0.9818128098223958, 0.018187190177604 15]
4	40801	(1.0, 0.0, 63.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0,...	0.0	[0.9818128098223958, 0.018187190177604 15]

Discussion

- Decision Tree Classifier had a high accuracy of 98.23%
- We saw that age, BMI, and glucose level are the most important features when it comes to predicting stroke-prone individuals, based on the current dataset.
- We observed that women are prone to stroke on average at a much older age (74-79) in comparison to males which experiences strokes on average as soon as their mid 50's and 60, ages which are much rarer for women to have strokes.


Thank
You