ANALYSIS OF BANKING DATA USING PYSPARK

Group Number:6

Group Members:

Gowtham Reddy Chinthakuntla

Thanveer Shaik

Yeshwanth Buggaveeti

EXECUTIVE SUMMARY

Dataset is related to direct marketing campaigns (phone calls) of Portuguese.

The main goal is to predict if the client will opt in for term deposit.

Used Tools:

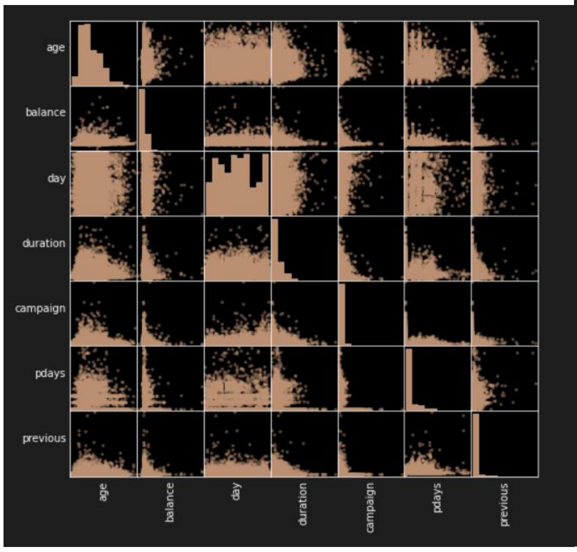
1. PySpark-Python API for Apache Spark.
2. Pandas- Software library written for the Python programming language for data manipulation and analysis
3. OneHotEncoderEstimator- maps a column of category indices to a column of binary vectors, with at most a single one-value per row that indicates the input category index.
4. StringIndexer-converts a single column to an index column.
5. VectorAssembler-combining raw features generated by different transformers into a single feature vector, to train ML models like logistic regression and decision trees.
6. Pipeline- End-to-End construct that orchestrates the flow of data into, and output from a machine learning model.
7. Logistic regression- Process of modelling the probability of a discrete outcome given an input variable.
8. Precision and recall- Precision quantifiers the number of positive class predictions that belong to the positive class. Recall quantifiers the number of positive class predictions made from all positive examples in the dataset.
9. Decision Tree- Decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource cost, and utility.
10. Random Forest- ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.

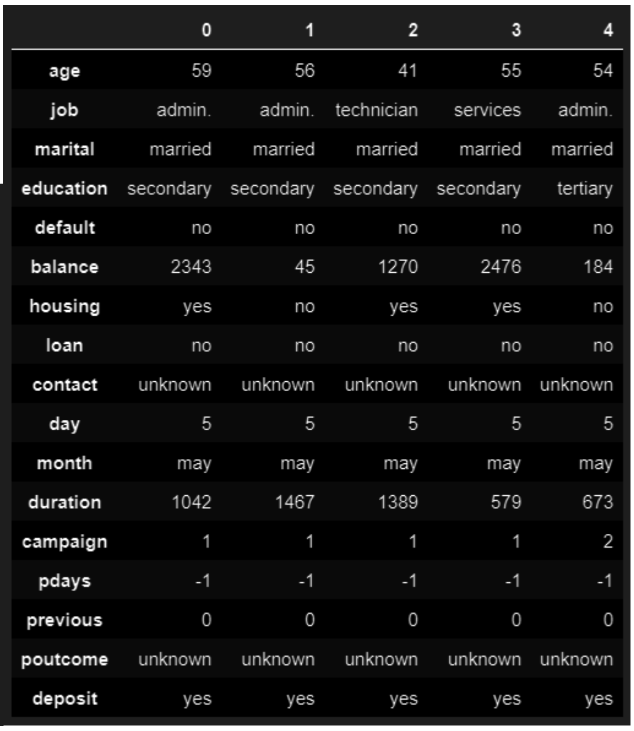
PROJECT MOTIVATION/BACKGROUND

The main reason to choose this project is, the banking system has improved a lot in recent years. Even though banking system has improved, some facilities are not required to every customer. For example, student needs student loan so that he can pursue his education, an IT employee need home loan or Auto loan depending on his requirements. It will be logical to send the relevant notifications and offers to the people who need it most, where it will benefit both bank and their customers. The classification goal is to predict whether the client will subscribe to the respective offer. This project is made in a way the data is taken and depending on the data is there, the decision is taken. That is the main activity for Decision tree. The main components used for this prediction are, Logistic regression, precision and recall, decision tree and random forest. Depending on the requirement of the customer, the notifications will be sent. So, this might be even used in other sectors such as e commerce websites, where they will give offers based on the transactions of the customer and their credit history. This project Blueprint can be used in the sectors where the customers are involved.

DATA DESCRIPTION

The Dataset used in this project is a banking data. The Crucial data that is in this dataset are job, housing, and loan. Depending on the job he's doing, if he has an own house, has any active loans.





	0	1	2	3	4
age	59	56	41	55	54
job	admin.	admin.	technician	services	admin.
marital	married	married	married	married	married
education	secondary	secondary	secondary	secondary	tertiary
default	no	no	no	no	no
balance	2343	45	1270	2476	184
housing	yes	no	yes	yes	no
loan	no	no	no	no	no
contact	unknown	unknown	unknown	unknown	unknown
day	5	5	5	5	5
month	may	may	may	may	may
duration	1042	1467	1389	579	673
campaign	1	1	1	1	2
pdays	-1	-1	-1	-1	-1
previous	0	0	0	0	0
poutcome	unknown	unknown	unknown	unknown	unknown
deposit	yes	yes	yes	yes	yes

We can see the different types of graphs in the outputs above.

DATA PREPARATION ACTIVITIES:

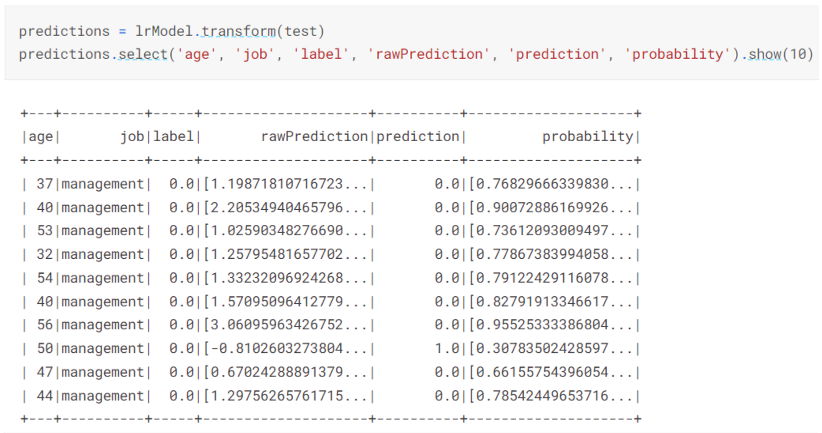
The data is filtered using linear regression, after that, we analysed the data according to the requirements. Then scatter matrix is used to find the data in plots. The next process involves data preparation for machine learning. We indexed each column using the StringIndexer, then convert the indexed categories into one-hot coded variables. We used StringIndexer to encode our labels to label indices. VectorAssembler is used to combine all the feature columns into a single vector column.

Then, Pipeline is used to chain multiple transformers and estimators together to specify our machine learning wo rkflow. Then logistic Regression model is used. By that, we'll get ROC curve. Precision and recall with get the p lot. then decision tree is used to get raw prediction and probability. After that random forest classifier is used to g et test are under ROC, which is 0.8801074851038885 in this case. Finally, the gradient boost is used, where the final value is 0.8981050997838095.

DATA ANALYTICS USED

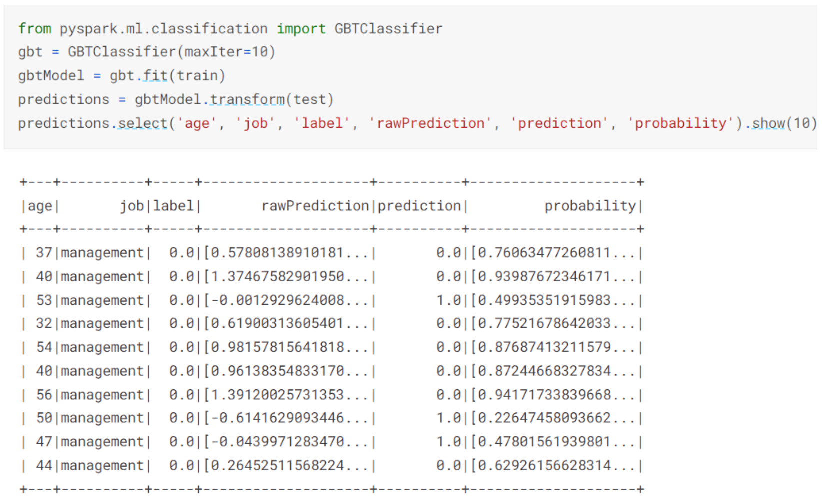
The important techniques used are

- LR model
- Decision Tree classifier
- Random forest Classifier
- Gradient-Boosted Tree classifier.



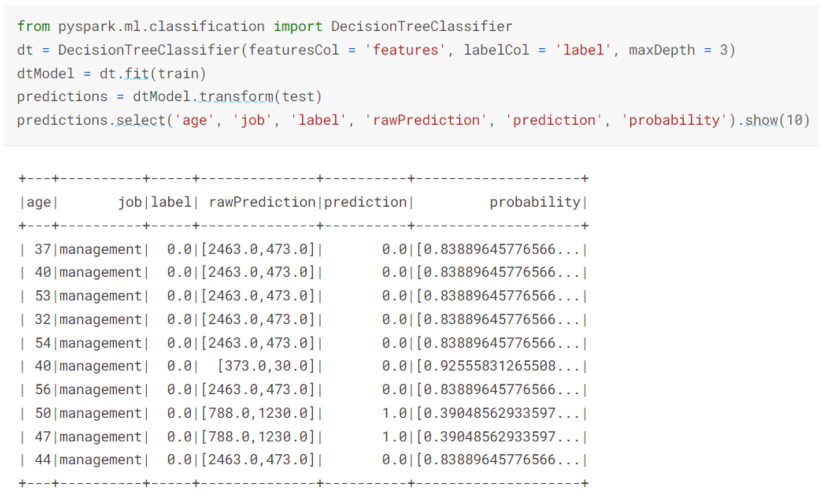
```
predictions = lrModel.transform(test)
predictions.select('age', 'job', 'label', 'rawPrediction', 'prediction', 'probability').show(10)
```

```
+---+----------+-----+--------------------+----------+--------------------+
|age|       job|label|       rawPrediction|prediction|         probability|
+---+----------+-----+--------------------+----------+--------------------+
| 37|management|  0.0|[1.19871810716723...|       0.0|[0.76829666339830...|
| 40|management|  0.0|[2.20534940465796...|       0.0|[0.90072886169926...|
| 53|management|  0.0|[1.02590348276690...|       0.0|[0.73612093009497...|
| 32|management|  0.0|[1.25795481657702...|       0.0|[0.77867383994058...|
| 54|management|  0.0|[1.33232096924268...|       0.0|[0.79122429116078...|
| 40|management|  0.0|[1.57095096412779...|       0.0|[0.82791913346617...|
| 56|management|  0.0|[3.06095963426752...|       0.0|[0.95525333386804...|
| 50|management|  0.0|[-0.8102603273804...|       1.0|[0.30783502428597...|
| 47|management|  0.0|[0.67024288891379...|       0.0|[0.66155754396054...|
| 44|management|  0.0|[1.29756265761715...|       0.0|[0.78542449653716...|
+---+----------+-----+--------------------+----------+--------------------+
```



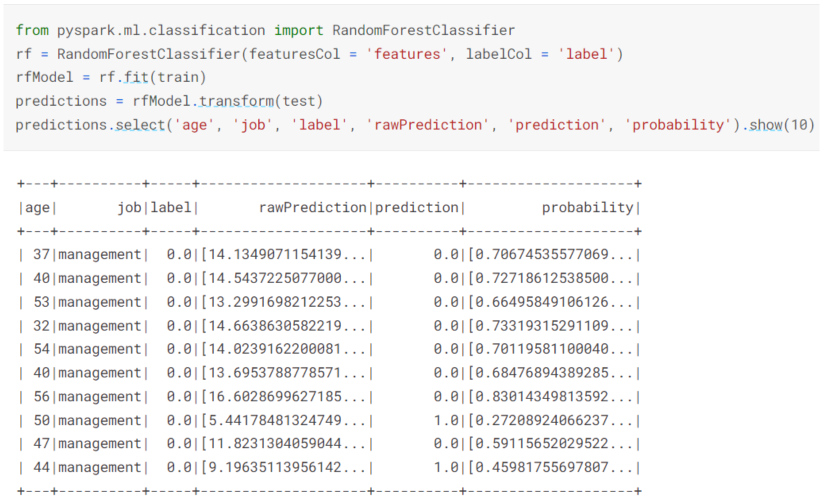
```
from pyspark.ml.classification import GBTClassifier
gbt = GBTClassifier(maxIter=10)
gbtModel = gbt.fit(train)
predictions = gbtModel.transform(test)
predictions.select('age', 'job', 'label', 'rawPrediction', 'prediction', 'probability').show(10)
```

```
+---+----------+-----+--------------------+----------+--------------------+
|age|       job|label|       rawPrediction|prediction|         probability|
+---+----------+-----+--------------------+----------+--------------------+
| 37|management|  0.0|[0.57808138910181...|       0.0|[0.76063477260811...|
| 40|management|  0.0|[1.37467582901950...|       0.0|[0.93987672346171...|
| 53|management|  0.0|[-0.0012929624008...|       1.0|[0.49935351915983...|
| 32|management|  0.0|[0.61900313605401...|       0.0|[0.77521678642033...|
| 54|management|  0.0|[0.98157815641818...|       0.0|[0.87687413211579...|
| 40|management|  0.0|[0.96138354833170...|       0.0|[0.87244668327834...|
| 56|management|  0.0|[1.39120025731353...|       0.0|[0.94171733839668...|
| 50|management|  0.0|[-0.6141629093446...|       1.0|[0.22647458093662...|
| 47|management|  0.0|[-0.0439971283470...|       1.0|[0.47801561939801...|
| 44|management|  0.0|[0.26452511568224...|       0.0|[0.62926156628314...|
+---+----------+-----+--------------------+----------+--------------------+
```



```
from pyspark.ml.classification import DecisionTreeClassifier
dt = DecisionTreeClassifier(featuresCol = 'features', labelCol = 'label', maxDepth = 3)
dtModel = dt.fit(train)
predictions = dtModel.transform(test)
predictions.select('age', 'job', 'label', 'rawPrediction', 'prediction', 'probability').show(10)
```

```
+---+----------+-----+--------------+----------+--------------------+
|age|       job|label| rawPrediction|prediction|         probability|
+---+----------+-----+--------------+----------+--------------------+
| 37|management|  0.0|[2463.0,473.0]|       0.0|[0.83889645776566...|
| 40|management|  0.0|[2463.0,473.0]|       0.0|[0.83889645776566...|
| 53|management|  0.0|[2463.0,473.0]|       0.0|[0.83889645776566...|
| 32|management|  0.0|[2463.0,473.0]|       0.0|[0.83889645776566...|
| 54|management|  0.0|[2463.0,473.0]|       0.0|[0.83889645776566...|
| 40|management|  0.0|   [373.0,30.0]|       0.0|[0.92555831265508...|
| 56|management|  0.0|[2463.0,473.0]|       0.0|[0.83889645776566...|
| 50|management|  0.0|[788.0,1230.0]|       1.0|[0.39048562933597...|
| 47|management|  0.0|[788.0,1230.0]|       1.0|[0.39048562933597...|
| 44|management|  0.0|[2463.0,473.0]|       0.0|[0.83889645776566...|
+---+----------+-----+--------------+----------+--------------------+
```



```
from pyspark.ml.classification import RandomForestClassifier
rf = RandomForestClassifier(featuresCol = 'features', labelCol = 'label')
rfModel = rf.fit(train)
predictions = rfModel.transform(test)
predictions.select('age', 'job', 'label', 'rawPrediction', 'prediction', 'probability').show(10)
```

```
+---+----------+-----+--------------------+----------+--------------------+
|age|       job|label|       rawPrediction|prediction|         probability|
+---+----------+-----+--------------------+----------+--------------------+
| 37|management|  0.0|[14.1349071154139...|       0.0|[0.70674535577069...|
| 40|management|  0.0|[14.5437225077000...|       0.0|[0.72718612538500...|
| 53|management|  0.0|[13.2991698212253...|       0.0|[0.66495849106126...|
| 32|management|  0.0|[14.6638630582219...|       0.0|[0.73319315291109...|
| 54|management|  0.0|[14.0239162200081...|       0.0|[0.70119581100040...|
| 40|management|  0.0|[13.6953788778571...|       0.0|[0.68476894389285...|
| 56|management|  0.0|[16.6028699627185...|       0.0|[0.83014349813592...|
| 50|management|  0.0|[5.44178481324749...|       1.0|[0.27208924066237...|
| 47|management|  0.0|[11.8231304059044...|       0.0|[0.59115652029522...|
| 44|management|  0.0|[9.19635113956142...|       1.0|[0.45981755697807...|
+---+----------+-----+--------------------+----------+--------------------+
```

FINDINGS

In Every technique used, the accuracy of them are nearly 90%, which is significantly higher compared to many models. Raw prediction and probability are important variables what decides many. Dependent variables like raw prediction and probability are based on the job, age, housing etc. these probabilities are given based on features.

BUSINESS IMPLICATIONS/INTELLIGENCE

This can be used in many types of projects, where we can get high accuracy results in predictions. With best dataset, the predictions might even cross 95%. The Steps should be taken for best prediction is:

We should find the correct variables to use for prediction.

Make sure the data is cleaned for the perfect prediction.

CONCLUSIONS

- Logistic Regression is used in this project
- PySpark is mainly used in the whole project.
- Decision Tree, Random Forest Classifier is used for predicting the dataset.
- Accuracy is 90% which is significantly higher than many projects and predictions.

References

Bank Marketing

- a balanced dataset (sample taken from UCI)-_**BARGAVA · UPDATED 5 YEARS AGO**

https://www.kaggle.com/datasets/rouseguy/bankbalanced