Case study report

Introduction

This report is a study of a dataset about a marketing campaign of a Portuguese bank and to suggest future strategies for an effective marketing that targets to improve the number of customers subscribing the term deposits.

Using Spark for data analysis

Spark is chosen as the platform for analysis of data given that with its ***Resilient Distributed Dataset (RDD)***, which distributes the huge data into various logical partitions called ***clusters*** and the computations on these partitions are done independently on different nodes of the cluster, makes it extremely fast and its ***lazy loading*** technique helps in successfully managing the cache memory.

Data distribution

The bank.csv file which contains the dataset is downloaded as a csv file using *'wget'*. The data is read to a '***Pyspark Dataframe'***. Using the functions like ***printschema(), describe(), crosstab()*** various columns have been analysed.

Observations

- All the columns are nullable and categorical columns have values as *'unknown'*.
- The numerical columns are spread on a wide range starting at 0.8 for *'previous'* column to values in 3000 for *'balance'* column.
- There are no lengthy sentences and so no tokenization is required.
- Categories in various categorical column is displayed below

```
+-------+ +-------+ +----+ +---------+ +--------+ +-------+ +--------+
|default| |housing| |loan| |  contact| | marital| |deposit| |poutcome|
+-------+ +-------+ +----+ +---------+ +--------+ +-------+ +--------+
|     no| |     no| |  no| |  unknown| |divorced| |     no| | success|
|    yes| |    yes| | yes| | cellular| | married| |    yes| | unknown|
+-------+ +-------+ +----+ |telephone| |  single| +-------+ |   other|
                           +---------+ +--------+           | failure|
                                                            +--------+
```

job	month
management	jun
retired	aug
unknown	may
self-employed	feb
student	sep
blue-collar	mar
entrepreneur	oct
admin.	jul
technician	nov
services	apr
housemaid	dec
unemployed	jan

Data wrangling and pre-processing

Data Cleaning

- Rows with null values are removed using the '***dropna()'*** function.
- All rows with value *'unknown'* in categorical columns are removed using ***'Spark SQL'***.
- Only the rows with values *'failure'* and *'success'* for *'poutcome'* are retained.

Data Pre-processing

Now that we have cleaned data, we need to prepare it for the Machine learning algorithms which mostly requires a single vectorised 'features' column.

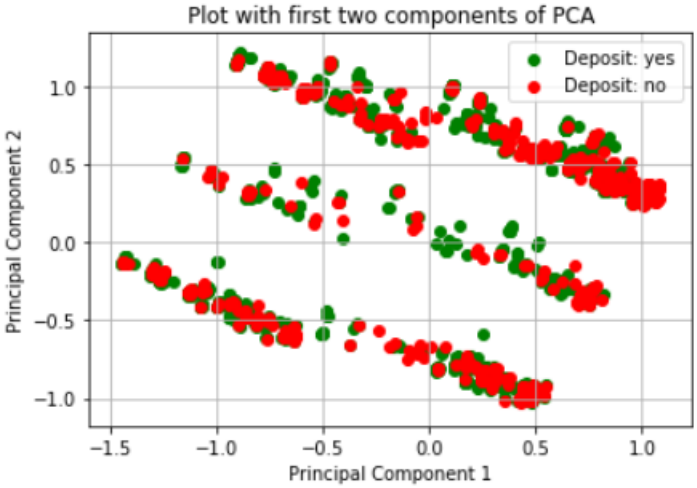
- The categorical columns are first indexed using the ***StringIndexer*** that indexes the categories starting from 0 as per repetition of each distinct category in the column.
- The indexed columns are further converted to sparse vectors using the ***OneHotEncoderEstimator*** which forms the sparse vectors of length equal to number of categories in the column.
- ***VectorAssembler*** vectorises the non-vector columns which are numerical columns in our case and assembles all input columns given to a single vectorised column.
- Further, we go for a ***Min-Max normalisation*** to normalise the values between 0 and 1 and this output acts as input to ML algorithms.

PCA

Since we have 32 features when all numerical, categorical columns along with various categories are considered, it is quite impossible for a data visualisation. So, we go for a ***dimensionality reduction using Principal Component Analysis*** which uses orthogonal transformation to squeeze in possibly correlated data into uncorrelated values called ***'Principal components'*** *(Jolliffe, 2002).*

Observations

- We have reduced the dimensionality to 2, that acquired a variance of *0.24, 0.17* and plotted the two PCA components thus obtained on a scatter plot.
- By increasing the dimensionality, the variance is decreasing.



Unsupervised learning

In unsupervised learning we *don't give any training data* but go for giving the complete dataset and expect the algorithm to form clusters by grouping the closely related data that helps us to understand the data distribution more clearly.

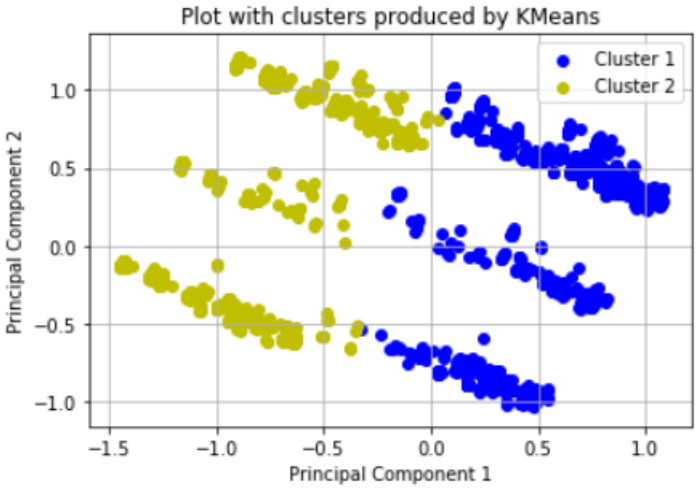
KMeans

KMeans algorithm tries to group together similar data which are termed as ***clusters*** by forming ***centroids*** and by optimizing the squared mean distance between this groups *(Hartigan & Wang, 1979).*

Observations:

- The ***Within Set Sum of Squared Error*** (WSSSE) which is calculated as the aggregate of sum of squares of distance of a point from its nearest centre in a partition has been recorded as *65.05284535441899* with two clusters and is reduced by increasing clusters.
- The ***Silhouette score*** which is the difference between a point and points within the cluster and to the neighbouring cluster is about *0.34* with 2 clusters taken and is better compared to taking more clusters.
- Using the deposit as label we tried to apply the ***Multi class classification evaluator*** which gave us an accuracy score of *0.52* and *0.56* AUC as per ***Binary Evaluator.***
- Also, we tried to plot the clusters using PCA components which gave us a clear view on segregated clusters.

```
Cluster Centers:
[0.57541899 0.13184358 0.0603352  0.01340782 0.04804469 0.00782123
 0.03575419 0.03575419 0.0603352  0.01675978 0.47486034 0.45027933
 0.         0.97094972 0.99776536 0.62569832 0.92625698 0.48826816
 0.31088587 0.03688247 0.07516506 0.21330303 0.03718187]
[0.0311042  0.18351477 0.19906687 0.20139969 0.11897356 0.11508554
 0.04821151 0.04276827 0.0155521  0.02177294 0.62908243 0.26360809
 0.84836703 0.         0.99611198 0.49533437 0.88569207 0.57465008
 0.36131971 0.02968395 0.07429662 0.25288935 0.03981625]
```



Supervised Learning

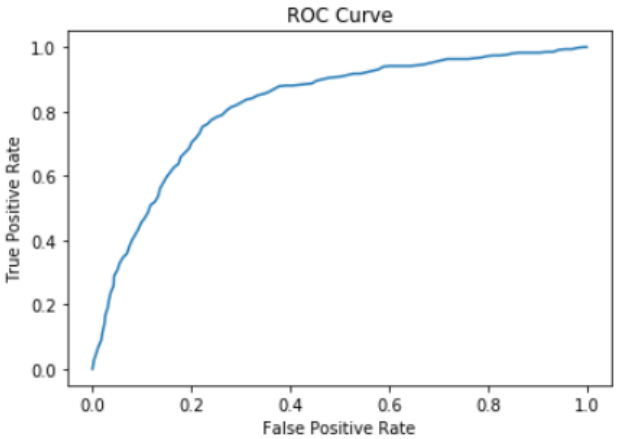
In this learning process we give the training data to create the model and then give the test data to get the predictions which are then evaluated using various evaluators. We divided the available data into *70* and *30* percent for *training* and *test* data respectively.

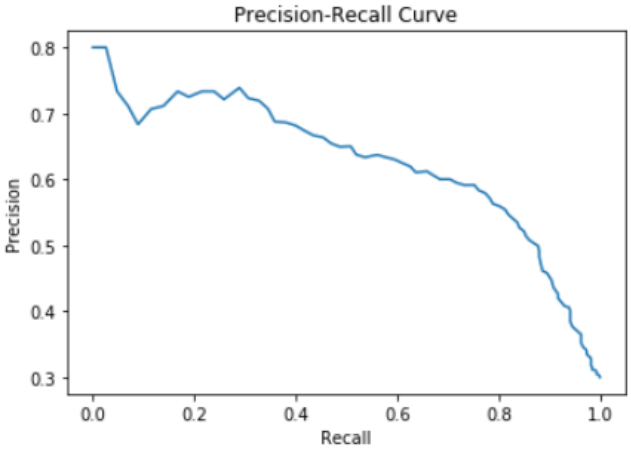
Logistic Regression

The logistic regression involves forming a logarithmic model by considering all the features and building the model that has the least logistic loss given by ***1 / (1 + e^-value).***

Observations

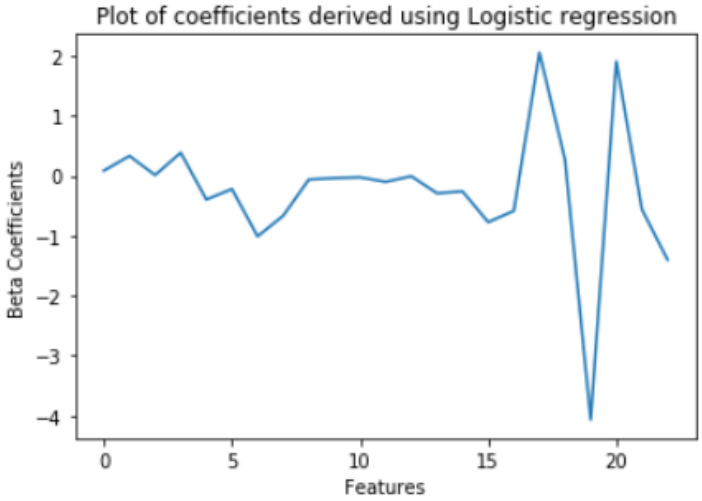
- The ***Receiver Operating Characteristics (ROC)*** has been plotted using ***False Positive Rate (FPR)*** and ***True Positive Rate (TPR)*** derived from the model formed and ***Area Under Curve (AUC)*** of *0.81* is observed. As it is close to value 'one' which states that the model formed is quite similar to the training data, it is a good number.
- ***Multi class classification evaluator*** has given an accuracy score of *0.74* and ***Binomial classification evaluator*** has given AUC of *0.68* when evaluated using the test data.





Training set areaUnderROC: 0.8181597573306368

- On analysing the coefficients, the features *balance, poutcome, campaign, previous* and *job* are the most influential towards the deposit as they have high absolute values adding more weight to the final outcome.



feature	coefficient
job_management	0.0
job_technician	0.08309589128005467
job_admin.	0.32757334062461385
job_blue-collar	0.0096105141436745
job_retired	0.38190941125150324
job_services	-0.39774944792469447
job_student	-0.22315221504037752
job_unemployed	-1.0115944989211938
job_self-employed	-0.6673825968709629
job_entrepreneur	-0.06422556314687217
job_housemaid	-0.04052176974477358
marital_married	0.0
marital_single	-0.02874554906578332
marital_divorced	-0.1028090951688996
education_secondary	0.0
education_tertiary	-0.01020230416420...
education_primary	-0.29307701174854
default_no	0.0
default_yes	-0.26309379068036487
housing_no	0.0
housing_yes	-0.7754311127062343
loan_no	0.0
loan_yes	-0.5864069648855801
poutcome_failure	0.0
poutcome_success	2.050485245963458
age	0.2604392385346028
balance	-4.06600492148482
campaign	1.9019832166550692
pdays	-0.5628888207448501
previous	-1.400697868170869

Decision Tree

Decision tree based on the entropy of various features selects a feature as a root node and starts forming a tree with this root in a descending value of entropy until the max depth mentioned. The test data is further evaluated based on this tree *(Rokach L, 2008).*

Observations

- Feature *'loan'* was taken as root node when we took the max depth as 4 and accuracy score came at around *0.75*.

- The features *campaign, pdays, loan, housing, default, job* are the important features as per the model developed with max depth as 4.
- The accuracy increased as max depth was increasing starting from 2 until max depth 5 where it reached *0.76* and then it started decreasing.

```
Learned classification tree model:
DecisionTreeClassificationModel (uid=DecisionTreeClassifier_f05a8cf046a4) of depth 4 with 21 nodes
  If (feature 17 <= 0.5)
   If (feature 22 <= 0.25)
    If (feature 21 <= 0.2045720984759672)
     Predict: 0.0
    Else (feature 21 > 0.2045720984759672)
     If (feature 22 <= 0.19444444444444442)
      Predict: 0.0
     Else (feature 22 > 0.19444444444444442)
      Predict: 1.0
   Else (feature 22 > 0.25)
    Predict: 1.0
  Else (feature 17 > 0.5)
   If (feature 15 <= 0.5)
    If (feature 21 <= 0.4396248534583822)
     If (feature 21 <= 0.11547479484173506)
      Predict: 0.0
     Else (feature 21 > 0.11547479484173506)
      Predict: 1.0
    Else (feature 21 > 0.4396248534583822)
     Predict: 0.0
   Else (feature 15 > 0.5)
    If (feature 16 <= 0.5)
     If (feature 4 <= 0.5)
      Predict: 1.0
     Else (feature 4 > 0.5)
      Predict: 0.0
    Else (feature 16 > 0.5)
     If (feature 3 <= 0.5)
      Predict: 0.0
     Else (feature 3 > 0.5)
      Predict: 1.0
```

Naïve Bayes

Naïve Bayes presumes independency between the features present in the data and generates probability of happening of each feature and populates them all to get probability of happening of the given label.

Observations

- An accuracy score of *0.72* is achieved by applying the model on test data.
- Increasing smoothness which provides 'fail-safe' probability, increases the accuracy score

Adding Up

Considering all the above results the *features* ***campaign, balance, job, loan*** seems to be most important features to be considered. In the next survey bank should target customers who are unemployed/students who has more balance and should increase number of campaigns to achieve more subscriptions in term deposits. Though we got an accuracy of 0.75, increasing the iterations in logistic regression, max depth in Decision tree and smoothing in Naïve Bayes can result in better results.

References

Jolliffe, I., 2011. Principal component analysis (pp. 1094-1096). Springer Berlin Heidelberg.

Hartigan, J.A. and Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), pp.100-108.

Rokach, L. and Maimon, O.Z., 2008. Data mining with decision trees: theory and applications (Vol. 69). World scientific.

Group contribution

We have made sure that all the group members involve in all the tasks as an equal understanding and exposure of all the concepts among all the group members is what we were aiming at.

We held regular meetups at the university, discussed, reflected and worked on the tasks collectively. So, it is quite complex to segregate the contribution. However, a brief splitting is given below for a generic idea.

Rajesh Kallumari (SID 218721162): Data cleaning, Data Distribution, OHE, logistic regression code and report.

Sai Midhil Chowdary Kari (SID 219054636): Unsupervised learning Kmeans, Normalisation, PCA code and report.

Jay Vimalbhai Trivedi (SID 218449725): Supervised learning Naïve Bayes, Decision tree code and report.