Oklahoma State University

MS, Business Analytics and Data Science

BAN 5753 Mini Project 2

Group Members

Akansha Mehta - A20341730

Karanveer Singh - A20164474

Sucharita Vallabhaneni - A20340678

Timothy Walsh - A20327822

Table of Contents

Click on an item to jump to its location in the document.

1. Exploratory Data Analysis

1. **Renamed** the columns to better reflect their actual meaning according to the data dictionary and avoid any confusion when using the fields in analysis.

Old name	New Name
emp.var.rate	employment_variation_rate
Cons.price.idx	consumer_price_index
Cons.conf.idx	consumer_confidence_index
euribor3m	euribor_3_monthrate
nr.employed	number_of_employees
y	outcome
default	credit_default

2. Checked the column **Data Types** to make sure each field is in the correct data type format for analysis.

```
|-- age: integer (nullable = true)
|-- job: string (nullable = true)
|-- marital: string (nullable = true)
|-- education: string (nullable = true)
|-- credit_default: string (nullable = true)
|-- housing: string (nullable = true)
|-- loan: string (nullable = true)
|-- contact: string (nullable = true)
|-- month: string (nullable = true)
|-- day_of_week: string (nullable = true)
|-- duration: integer (nullable = true)
|-- campaign: integer (nullable = true)
|-- pdays: integer (nullable = true)
|-- previous: integer (nullable = true)
|-- poutcome: string (nullable = true)
|-- employment_variation_rate: double (nullable = true)
|-- consumer_price_index: double (nullable = true)
|-- consumer_confidence_index: double (nullable = true)
|-- euribor_3_monthrate: double (nullable = true)
|-- number_of_employees: double (nullable = true)
|-- outcome: string (nullable = true)
```

3. **Counted datatypes** for each column.

```
Counter({'string': 11, 'int': 5, 'double': 5})
```

4. Counted the **null values** in each column, to understand the completeness of the data.

```
{'age': 0,
 'job': 0,
 'marital': 0,
 'education': 0,
 'credit_default': 0,
 'housing': 0,
 'loan': 0,
 'contact': 0,
 'month': 0,
 'day_of_week': 0,
 'duration': 0,
 'campaign': 0,
 'pdays': 0,
 'previous': 0,
 'poutcome': 0,
 'employment_variation_rate': 0,
 'consumer_price_index': 0,
 'consumer_confidence_index': 0,
 'euribor_3_monthrate': 0,
 'number_of_employees': 0,
 'outcome': 0}
```

The result was that **no null values** were found in any of the columns.

5. Checking the **cardinality** to see the number of unique values for each of the variables

```
age                            78
job                            12
marital                         4
education                       8
credit_default                  3
housing                         3
loan                            3
contact                         2
month                          10
day_of_week                     5
duration                     1544
campaign                       42
pdays                          27
previous                        8
poutcome                        3
employment_variation_rate      10
consumer_price_index           26
consumer_confidence_index      26
euribor_3_monthrate           316
number_of_employees            11
outcome                         2
dtype: int64
```

6. The **categorical variable** columns and their unique values and distributions are shown below:

Weeks

```
[Row(day_of_week='fri'),
 Row(day_of_week='thu'),
 Row(day_of_week='tue'),
 Row(day_of_week='wed'),
 Row(day_of_week='mon')]
```

```
+-----------+-----+
|day_of_week|count|
+-----------+-----+
|        fri| 7827|
|        thu| 8623|
|        tue| 8090|
|        wed| 8134|
|        mon| 8514|
+-----------+-----+
```

There is no data for Saturday and Sunday and data points are spread relatively equally over the week.

Months

```
[Row(month='jun'),
 Row(month='aug'),
 Row(month='may'),
 Row(month='mar'),
 Row(month='oct'),
 Row(month='jul'),
 Row(month='nov'),
 Row(month='apr'),
 Row(month='dec'),
 Row(month='sep')]
```

```
+-----+-----+
|month|count|
+-----+-----+
|  jun| 5318|
|  aug| 6178|
|  may|13769|
|  mar|  546|
|  oct|  718|
|  jul| 7174|
|  nov| 4101|
|  apr| 2632|
|  dec|  182|
|  sep|  570|
+-----+-----+
```

There is **no data** for January and February. Additionally some months like September, December, October and March have **far fewer data points** compared to the other months. It is possible that the bank's data collection system was down in those months or that for some reason the bank decided not to retain data for them. The bank's data collection and curation policies should be reviewed to determine whether this is something the bank was aware that it was doing or whether this was an error with their data collection and retention system and standard operating procedures.

Marital

```
[Row(marital='unknown'),
 Row(marital='divorced'),
 Row(marital='married'),
 Row(marital='single')]
```

```
+--------+-----+
| marital|count|
+--------+-----+
| unknown|   80|
|divorced| 4612|
| married|24928|
|  single|11568|
+--------+-----+
```

There are far fewer data points for the value "**divorced**" and only 80 unknowns. The unknown values indicate that this might not have always been a piece of information required by the bank or simply a case of human error in not entering 80 of these data points.

Education

```
[Row(education='high.school'),
 Row(education='unknown'),
 Row(education='basic.6y'),
 Row(education='professional.course'),
 Row(education='university.degree'),
 Row(education='illiterate'),
 Row(education='basic.4y'),
 Row(education='basic.9y')]
```

```
+--------------------+-----+
|           education|count|
+--------------------+-----+
|         high.school| 9515|
|             unknown| 1731|
|           basic.6y| 2292|
| professional.course| 5243|
|   university.degree|12168|
|          illiterate|   18|
|           basic.4y| 4176|
|           basic.9y| 6045|
+--------------------+-----+
```

The highest number of data points are associated with the **university degree** level of education and **highschool** level of education. Only a few are present for **illiterates**.

Housing

```
[Row(housing='unknown'), Row(housing='no'), Row(housing='yes')]
```

```
+-------+-----+
|housing|count|
+-------+-----+
|unknown|  990|
|     no|18622|
|    yes|21576|
+-------+-----+
```

Yes and no values for housing constitute most of the values for housing with about 15% more “yes” values than “no” values. There are only a small number (around 900) of unknowns.

Loan

```
[Row(loan='unknown'), Row(loan='no'), Row(loan='yes')]
```

```
+-------+-----+
|   loan|count|
+-------+-----+
|unknown|  990|
|     no|33950|
|    yes| 6248|
+-------+-----+
```

“No” values make up most of the Loan values. It has the same number of unknowns as Housing.

Poutcome **(outcome of previous marketing campaign)**

```
[Row(poutcome='success'), Row(poutcome='failure'), Row(poutcome='nonexistent')]
```

```
+-----------+-----+
|   poutcome|count|
+-----------+-----+
|    success| 1373|
|    failure| 4252|
|nonexistent|35563|
+-----------+-----+
```

Success and failures along with mostly nonexistent outcomes

Contact

```
[Row(contact='cellular'), Row(contact='telephone')]
```

```
+---------+-----+
|  contact|count|
+---------+-----+
| cellular|26144|
|telephone|15044|
+---------+-----+
```

Two contacts are available, cellular and telephone and data is unevenly distributed with around 2/3rd of values being cellular.

Credit Default

```
[Row(credit_default='unknown'),
 Row(credit_default='no'),
 Row(credit_default='yes')]
```

```
+--------------+-----+
|credit_default|count|
+--------------+-----+
|       unknown| 8597|
|            no|32588|
|           yes|    3|
+--------------+-----+
```

About 80% of Credit Default values are “no” while about 20% are “yes.” There are only 3 unknowns, which is a small amount given the volume of data.

Job

```
[Row(job='management'),
 Row(job='retired'),
 Row(job='unknown'),
 Row(job='self-employed'),
 Row(job='student'),
 Row(job='blue-collar'),
 Row(job='entrepreneur'),
 Row(job='admin.'),
 Row(job='technician'),
 Row(job='services'),
 Row(job='housemaid'),
 Row(job='unemployed')]
```

```
+-------------+-----+
|          job|count|
+-------------+-----+
|   management| 2924|
|      retired| 1720|
|      unknown|  330|
|self-employed| 1421|
|      student|  875|
|  blue-collar| 9254|
| entrepreneur| 1456|
|       admin.|10422|
|   technician| 6743|
|     services| 3969|
|    housemaid| 1060|
|   unemployed| 1014|
+-------------+-----+
```

A major chunk of the people for which data is collected are either working in **admin**, **blue-collar** jobs or are **technicians**.

Outcome

```
[Row(outcome='no'), Row(outcome='yes')]
```

```
+-------+-----+
|outcome|count|
+-------+-----+
|     no|36548|
|    yes| 4640|
+-------+-----+
```

The outcome was “yes” (successful) around 11% of the time.

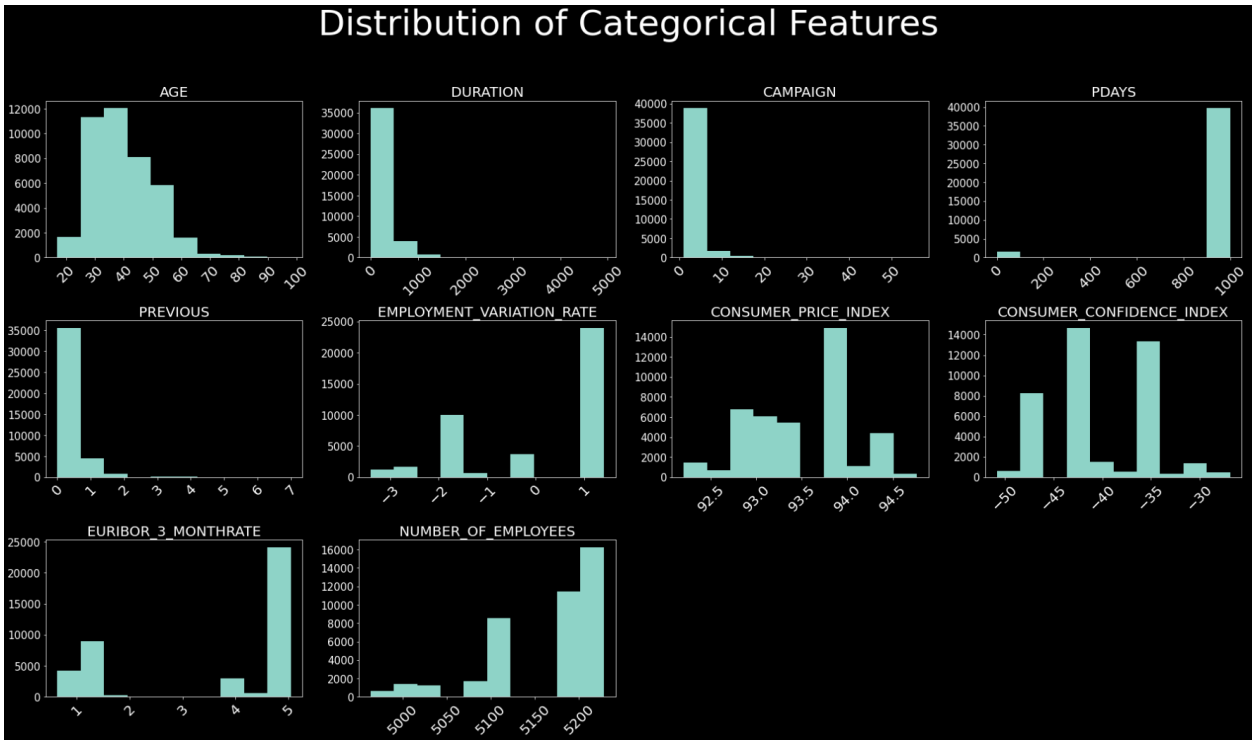
7. Below, is a table of the **numerical columns** and their respective distributions.

	0	1	2	3	4
summary	count	mean	stddev	min	max
age	41188	40.02406040594348	10.421249980934043	17	98
duration	41188	258.2850101971448	259.27924883646455	0	4918
campaign	41188	2.567592502670681	2.770013542902331	1	56
pdays	41188	962.4754540157328	186.910907344741	0	999
previous	41188	0.17296299893172767	0.49490107983928927	0	7

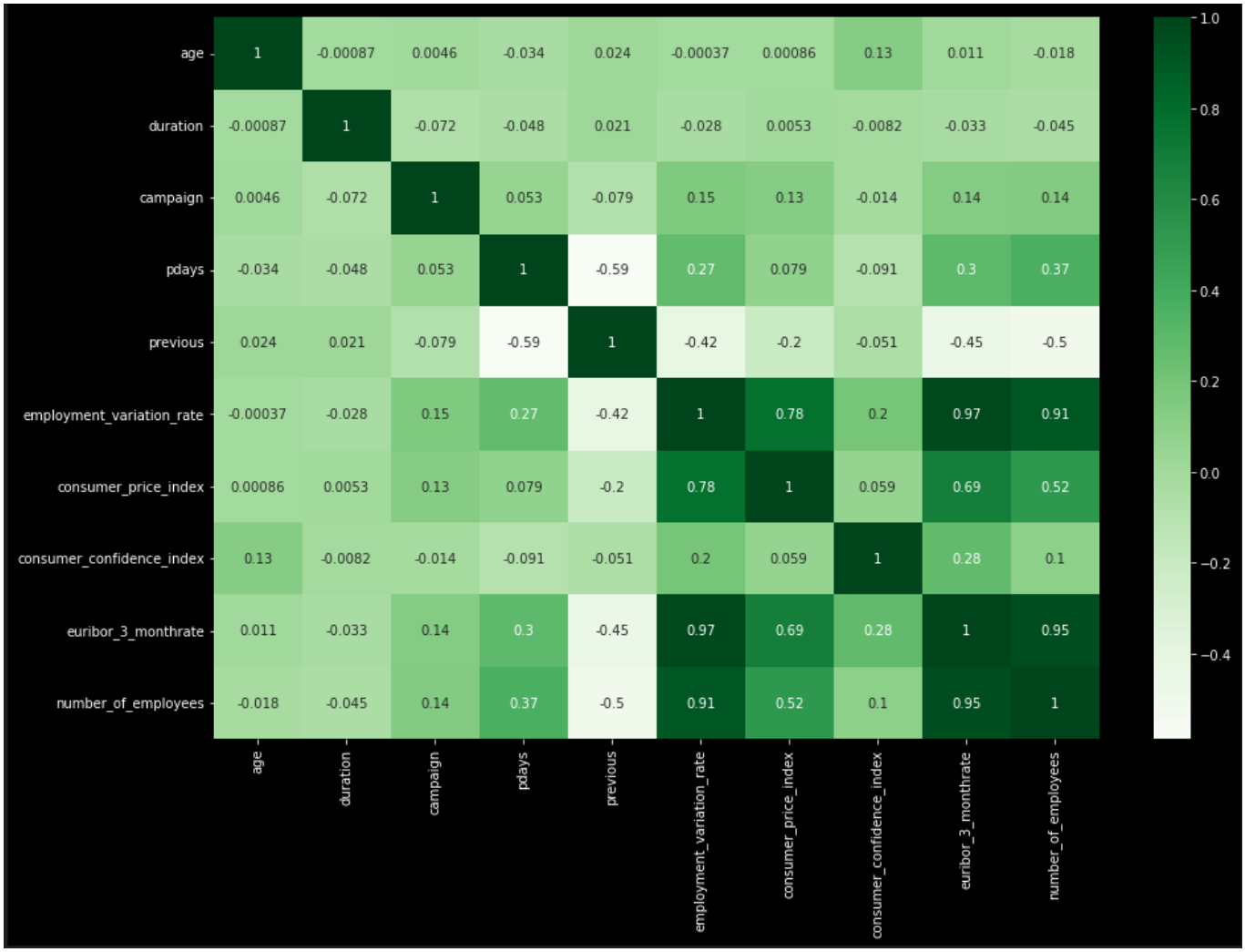
	0	1	2	3	4
summary	count	mean	stddev	min	max
employment_variation_rate	41188	0.08188550063178966	1.57095974051703	-3.4	1.4
consumer_price_index	41188	93.5756643682899	0.5788400489540823	92.201	94.767
consumer_confidence_index	41188	-40.502600271918276	4.628197856174573	-50.8	-26.9
euribor_3_monthrate	41188	3.621290812858533	1.7344474048512595	0.634	5.045
number_of_employees	41188	5167.035910943957	72.25152766826338	4963.6	5228.1

8. The plots below show the distribution of the features. Many of the features have sizable skew.

Age, Duration, Campaign, and Previous are all **right-skewed**. Employment Variation Rate, Euribor 3 Month Rate, and Number of Employees are all **left-skewed**.



9. Built a **correlation matrix** in the style of a heat map to check the correlation for variables



In the correlation matrix, stronger correlations/associations are indicated by darker colored squares. We can see that some of the strongest correlations are with **number_of_employees** correlation to both **employment_variation_rate** and **euribor_3_months**.

The strength of the associations are evaluated as follows:

Strength of Association	Positive	Negative
Small	0.1 to 0.3	-0.1 to - 0.3
Medium	0.3 to 0.5	- 0.3 to -0.5
Large	0.5 to 1.0	- 0.5 to -1.0

2. Predictive Models

1. Logistic Regression

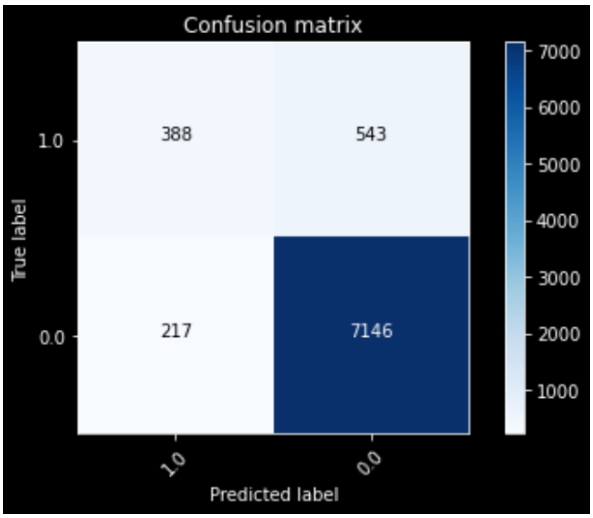
Logistic Regression is a classification technique used in machine learning. It uses a logistic function to model the dependent variable. The dependent variable is dichotomous in nature, i.e. there could only be two possible classes such as in this problem - whether a person will subscribe for the term deposit or not.

The model has been implemented using the code below -

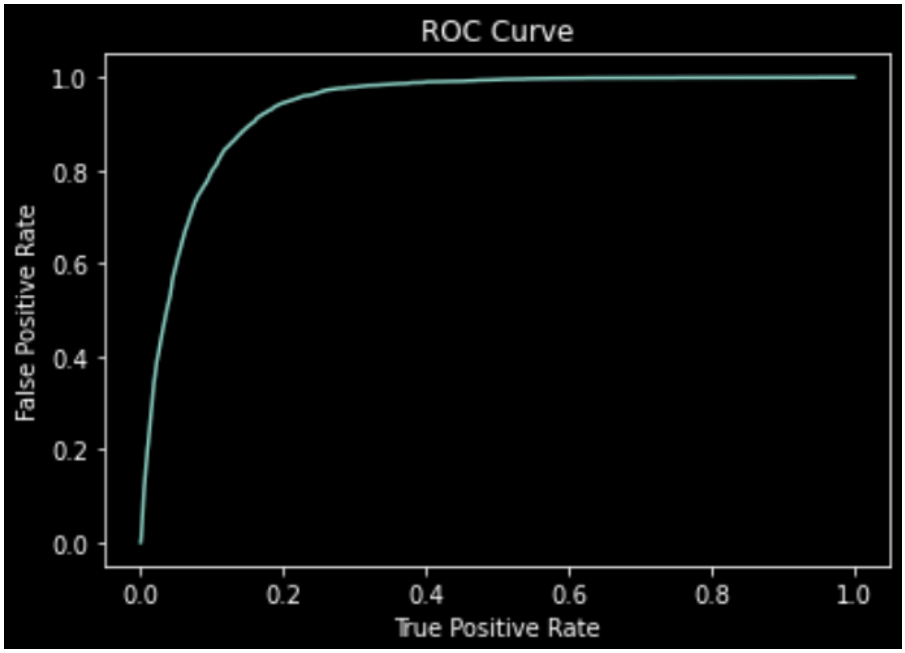
```
from pyspark.ml.classification import LogisticRegression
lr = LogisticRegression(featuresCol = 'features', labelCol = 'label', maxIter=5)
lrModel = lr.fit(train)
predictions = lrModel.transform(test)
#predictions_train = lrModel.transform(train)
predictions.select('label', 'features',  'rawPrediction', 'prediction', 'probability').toPandas().head(5)
```

Model Accuracy: 90.8%

Logistic Regression Confusion Matrix



Logistic Regression ROC Curve



2. **Decision Tree**

A **decision tree** is a flowchart-like tree structure where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in a recursive manner called recursive partitioning.

The decision tree model was implemented using the code below:

```
dtc = DecisionTreeClassifier(featuresCol="features", labelCol="label")
dtc = dtc.fit(train)

pred = dtc.transform(test)
pred.toPandas().head(3)
```

Model Accuracy: **91.02%**

Decision Tree Confusion Matrix

```
Confusion Matrix:
[[7038  325]
 [ 405  526]]
```

3. **Random Forest**

A **random forest** is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

The model was implemented using the code below:

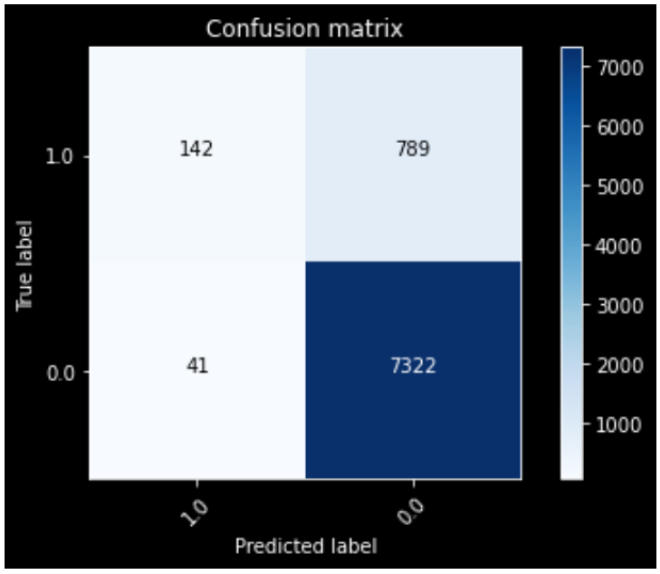
```
from pyspark.ml.classification import RandomForestClassifier

rf = RandomForestClassifier(featuresCol = 'features', labelCol = 'label')
rfModel = rf.fit(train)
predictions = rfModel.transform(test)
predictions.select('*', 'probability').show(5)
```

Model Accuracy: 86.87%

Random Forest Confusion Matrix

```
Confusion matrix, without normalization
[[ 142  789]
 [  41 7322]]
```



4. We also tried running XG Boost Regressor and but we were not able to run it on Spark, hence we ended up comparing 3 Machine Learning algorithms

3. Model Selection

Model	Accuracy
Logistic Regression	**90.83**
Decision Tree	90.95
Random Forest	87.41

We picked **Logistic Regression** as our main classification model over the other models. It wins over Decision Tree even though the accuracy for Decision Tree is higher. This is because:

- We already know that Decision Tree always **overfits** the model
- Random Forest (average of 100 Decision Trees) which is a better and more enhanced version of Decision Tree is giving a lower accuracy than the Decision Tree and Logistic Regression.

4. K-Means Clustering

K-means is an unsupervised classification algorithm, also called clusterization, that groups objects into k groups based on their characteristics. The grouping is done by minimizing the sum of the distances between each object and the group or cluster centroid.

It was implemented using the code below:

```
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator
silhouette_score=[]
evaluator = ClusteringEvaluator(predictionCol='prediction', featuresCol='standardized', \
                                metricName='silhouette', distanceMeasure='squaredEuclidean')
for i in range(2,10):

    KMeans_algo=KMeans(featuresCol='standardized', k=i)

    KMeans_fit=KMeans_algo.fit(data_scale_output)

    output=KMeans_fit.transform(data_scale_output)



    score=evaluator.evaluate(output)

    silhouette_score.append(score)

    print("Silhouette Score:",score)
```

Silhouette Scores are as follows (As per the range defined in the above code)

```
Silhouette Score: 0.20502898666544395
Silhouette Score: 0.08988258426821231
Silhouette Score: 0.1144094038992598
Silhouette Score: 0.08714920146095612
Silhouette Score: 0.0817298467015279
Silhouette Score: 0.10501205378873585
Silhouette Score: 0.10736601452606331
Silhouette Score: 0.09890540478269687
```

Range	Silhouette Score
2	.205
3	.090
4	.114
5	.087
6	.082
7	.105
8	.107
9	.099

We see that the silhouette score is maximized at **k = 2** - the optimum number of clusters. As per the business problem, we need to predict whether a customer subscribes to the term deposit or not (i.e dichotomous in nature). Hence, grouping data into 2 clusters would be the best choice if K-means has to be applied.

5. Prescriptive and Descriptive Recommendations

These recommendations are based on the variable importance received on the basis of feature weights

1. Most of the values for **poutcomes** (previous outcome for the marketing campaign) are not recorded. A good analysis requires good data collection pipelines. We would recommend that the bank deploy better data collection pipelines and only then relevant marketing suggestions can be given.

2. Although the data mentions the outcome of the previous campaign for a user, it would be better if the **cost of term deposit** were mentioned too. This would help to prioritize specific users who could be more profitable for the business.

3. We are only able to see **11.27%** conversions. More ways to reach out to the customers and engagement programs can benefit the business based on demographic data. We can see that the average customer age is 40 - the bank can use this by researching which methods of marketing best work on their **average age demographic**. Additionally, when combined with the previous recommendation of tracking the cost of term deposit, the bank could see its **most profitable age/generation segment** and apply marketing methods specifically to target that **most profitable group**.

4. Based on the **variable importance** (top10) for the model, here are some insights that the business can involve in their decision making process -
    - Duration (i.e the last contact duration) is the most significant variable for determining the target variable. Thus, the duration to when this user was last contacted plays the most important role in determining whether they would subscribe to the term deposit or not. Based on this, we recommend that the bank train its client-facing employees to increase the possibility of extended contact durations with the client by 1) asking if there is anything in addition to the initial contact reason that they can help the customer with and 2) Mentioning other useful services including the benefits of subscribing to a term deposit. This **extends contact duration** with the client while **directly marketing the term deposit subscription**.

    - The other top 4 significant variables from most to least importance were number_of_employees, poutcome (case : when it is success), employment_variation_rate and euribor_3_monthrate. It is noteworthy to see the success outcome of the previous contact makes a difference to whether the customer would subscribe to the term deposit now. Based on this, the bank can **target their marketing on clients that had successful previous outcomes** as they are more likely to result in success again.

    - The remaining 5 significant variables from most to least importance were consumer_confidence_index, contact (case : cellular), pdays, previous and consumer_price_index. Similar to the previous recommendations, these can be used to target marketing toward client segments with the greatest chance of responding favorably. For example, since **contact (case: cellular)** is fairly important,the bank could utilize a **text marketing** campaign.