



---

# MODERN DATA SCIENCE

---

## Assignment 2



GROUP NO-126  
SAI THARUN PALAGIRI-217636438  
FAROOQ SARFRAZ-218436204  
SAI PRASANTH THIRUMANI-218008216

## Table of Contents

Case Study Report.....	2
1:-Exploratory Analysis .....	2
2.0 The Method of Data wrangling: .....	2
2.1 Descriptive Statistics .....	2
2.2 Check for Missing Values: .....	3
2.3 Selecting the features: .....	3
2.4 One Hot Encoding: .....	4
2.5 Normalization:.....	4
3.0 Unsupervised Learning: .....	5
3.1 Clustering using K means: .....	5
Principal Component Analysis: .....	5
3.2 Supervised Learning:.....	6
3.2.1 Logistic Regression:.....	6
3.2.2 Decision Tree: .....	7
3.2.3 Navie Bayes:.....	7
4.0: Features which affect the label.....	7
5.0 Analysis and Improvement: .....	8
6.0 Team Work:.....	8

## Case Study Report

The bank data has been imported on ipython notebook to gain valuable information by performing machine learning algorithms on Spark.

### 1:-Exploratory Analysis

The name of the attributes and class of the attributes present in the bank data can be explored by performing **df.printSchema()** on the dataset and the number of observations present in the data can be read by running **df.count()**.The following are the results obtained.

```
root
|-- age: integer (nullable = true)
|-- job: string (nullable = true)
|-- marital: string (nullable = true)
|-- education: string (nullable = true)
|-- default: string (nullable = true)
|-- balance: integer (nullable = true)
|-- housing: string (nullable = true)
|-- loan: string (nullable = true)
|-- contact: string (nullable = true)
|-- day: integer (nullable = true)
|-- month: string (nullable = true)
|-- duration: integer (nullable = true)
|-- campaign: integer (nullable = true)
|-- pdays: integer (nullable = true)
|-- previous: integer (nullable = true)
|-- poutcome: string (nullable = true)
|-- deposit: string (nullable = true)
```

The number of observations present in bank data set are 11162.

From the above results it is clear that age balance, day, duration, campaign, pdays, previous are numeric variables and the rest of the variables are categorical variables. The number of records present in the dataset are 11162.

## 2.0 The Method of Data wrangling:

### 2.1 Descriptive Statistics

**df.describe().show()** code has been run to know the descriptive summary of all variables present in the data.

```
# Summary statistics of all attributes present in the data
df.describe().show()
```

summary	age	job	marital	education	default	balance	housing	loan	contact
count	11162	11162	11162	11162	11162	11162	11162	11162	11162
mean	41.231947679627304	null	null	null	null	1528.5385235620856	null	null	null
stddev	11.913369192215518	null	null	null	null	3225.413325946149	null	null	null
min	18	admin.	divorced	primary	no	-6847	no	no	cellular
max	95	unknown	single	unknown	yes	81204	yes	yes	unknown

Fig 1: Screenshot of summary statistics of data variables.

## 2.2 Check for Missing Values:

**for col in df.columns:**

```
print (col, "\t", "with null values: ", df.filter(df[col].isNull()).count())
```

The above can be used to check any missing values present in each variable in the data set. The results shows that there are no missing values present in the bank data.

```
age          with null values:  0
job          with null values:  0
marital      with null values:  0
education    with null values:  0
default      with null values:  0
balance      with null values:  0
housing      with null values:  0
loan         with null values:  0
contact      with null values:  0
day          with null values:  0
month        with null values:  0
duration     with null values:  0
campaign     with null values:  0
pdays       with null values:  0
previous     with null values:  0
poutcome     with null values:  0
deposit      with null values:  0
```

---

Fig 2: Missing Values in the data.

In some cases, the entries in some of the rows can be "?". So to check whether any entries are "?" we have used the code

**for col in df.columns:**

```
print(col, "\t", "with '?' values:", df.filter(df[col]=="?").count())
```

The results shows that there are no entries with "?" in the data set.

## 2.3 Selecting the features:

The list of attributes which are needed to be considered to build predictive models have been selected as specified in the Assignment instructions.

```
df.select("poutcome").distinct().rdd.map(lambda r: r[0]).collect().
```

```
['success', 'unknown', 'other', 'failure']
```

---

It has been identified that the unique values present in the poutcome are “success”, “failure”, “other” and “unknown”. As only “success” and “failure” are termed to be valued entries the other two entries have been removed by using SQL operations. The and now poutcome has only “success” and “failure” as its entries. The number of records after filtering the poutcome are 2299.

```

print("The Number of records after filteing poutcome {0}".format(dftemp.count()))

```

The NUumber of records after filteing poutcome 2299

## 2.4 One Hot Encoding:

From the schema and descriptive statistics, it is clear that some of the variables are categorical. In order to perform predictive models such as clustering and logistic regression the categorical variables are converted to numerical variables using one hot encoding technique. In the one Hot encoding technique all the categorical variables which are in the format string are first converted to arrays and using **OneHotEncoderEstimator** which has been imported from **pyspark.ml** to convert all categorical arrays to numeric variables and encoded using pipeline. The vectors which have been created by encoding categorical variables can be assembled into one attribute by using **VectorAssembler** and the attribute is named as **feature**.

feature
[33.0,4.0,0.0,0.0...]
[56.0,2.0,0.0,0.0...]
[34.0,3.0,0.0,1.0...]
[53.0,5.0,0.0,1.0...]
[37.0,2.0,0.0,0.0...]
[45.0,9.0,0.0,0.0...]

Fig 4: Screenshot of feature attribute.

**2.5 Normalization:** As clustering using k means works efficiently when all the variables present in the dataset are in the same range. So it often necessary to normalize the variable on a scale of 0 to 1. Therefore, Min-Max Normalization Technique has been used to normalize the values of each variable and assembled all values into a new variable called “features”.

feature	features
[33.0,4.0,0.0,0.0...]	[0.2,0.3636363636...]
[56.0,2.0,0.0,0.0...]	[0.50666666666666...]
[34.0,3.0,0.0,1.0...]	[0.21333333333333...]
[53.0,5.0,0.0,1.0...]	[0.46666666666666...]
[37.0,2.0,0.0,0.0...]	[0.25333333333333...]

Figure 5: After applying **MinMaxScaler()** on feature.

### 3.0 Unsupervised Learning:

#### 3.1 Clustering using K means:

K means algorithms has been applied on scaled data using a random k value at first then the optimum value of K has been determined using elbow method. The performance of the model has been evaluated using Silhouette index.

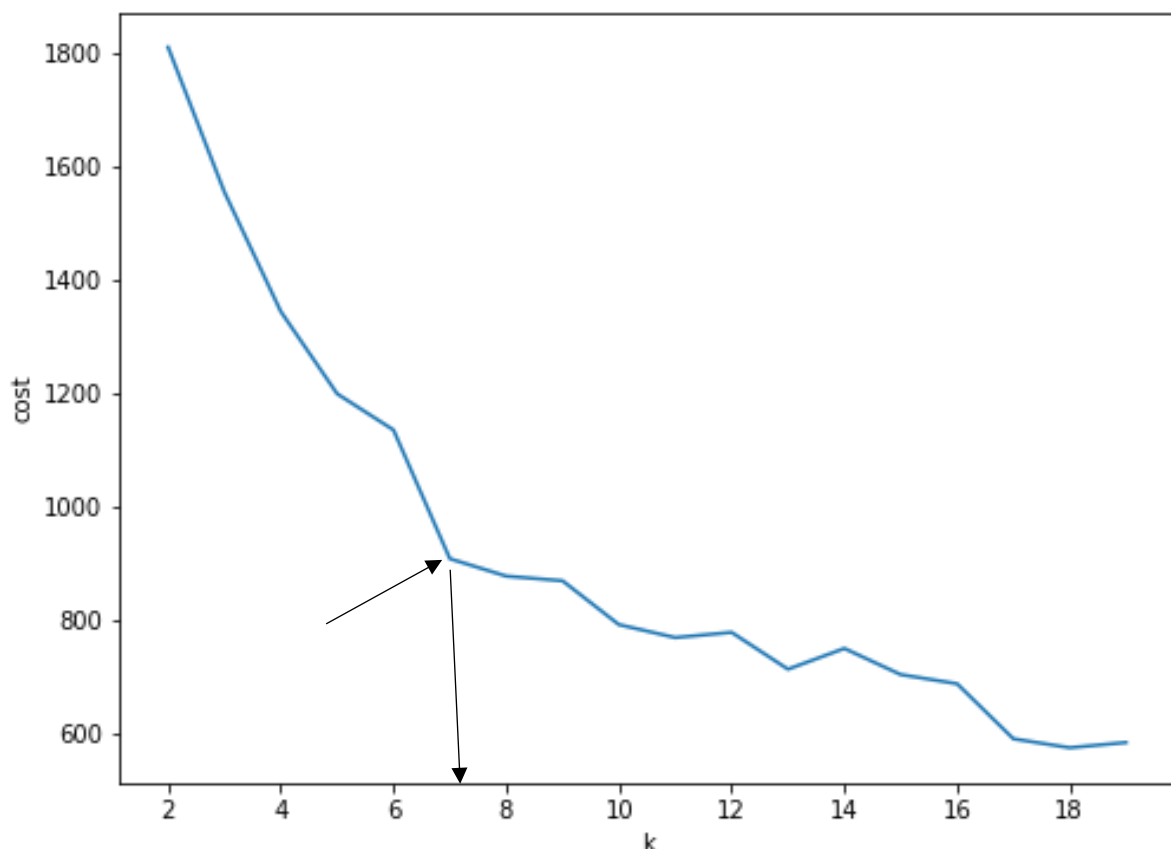


Figure 6: Elbow method

From figure 6 it is clear that the first elbow has occurred at  $k=7$ . So by fitting the k means model for  $k=7$  the value of Silhouette index is 0.479. As silhouette index lies in the interval -1 and 1. The value near to +1 indicates that the record which has been assigned to a cluster is having high similarity with the other records within intra cluster and maintains dissimilarity within inter clusters. The value 0.479 is close to 1 which indicates that the model has good clustering accuracy at  $k=7$ .

**Principal Component Analysis:** PCA is a dimensional reduction algorithm which is used to when the model has high number of independent variables. The dimensions are reduced to 2 and the variance explained by the two PCA's are calculated.


 The variance explained by pc1 and pc2 are [37.7313 15.6113]

Fig 7: Variance of PC1 and PC2

From the above result it is clear that the principal component 1 explains 37.73% variance in the output which is “deposit” and 15.611% of the variance in deposit can be explained by principal component 2.

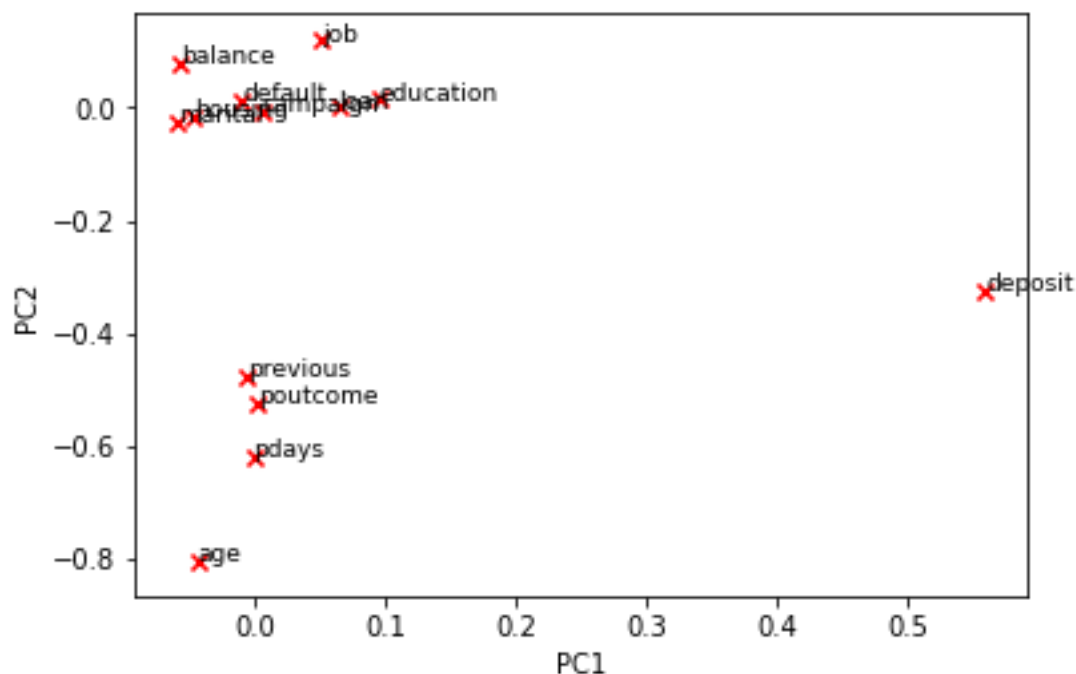


Fig 8: The 2D plot of PC1 vs PC2

### 3.2 Supervised Learning:

To fit model on bank data, first the data has been split into training and testing with percentage of 70 and 30 respectively. The models have been built on training data set and the performance of the model has been estimated using testing data.

#### 3.2.1 Logistic Regression:

Logistic regression estimates the output (Deposit) which is a binary variable so accurately even if the residuals doesn't follow normal distribution.

The Roc is nearly equal to 1 which shows that the model is accurate in predicting the output.

### 3.2.2 Decision Tree:

70% of the data is used to train the decision tree model and 30% of the data is used to test the model performance. By fitting decision tree on training data and testing the model the accuracy of the model is estimated to be 1 which shows that the model performance is good and can be used to predict the outcomes of unseen data.

Test Accuracy 1.0

### 3.2.3 Navie Bayes:

70% of the data is used to train the Navie Bayes model and 30% of the data is used to test the model performance. By fitting Navie Bayes on training data and testing the model the accuracy of the model is estimated to be 0.8918128654 which shows that the model performance is good but not better than decision tree and logistic regression and can be used to predict the outcomes of unseen data.

## 4.0: Features which affect the label

Attribute	Coefficients	exp(Coefficients)	Odds
'age',	-5.83491529	0.002923671	-99.708%
'job_index',	-4.7030214	0.009067838	-99.093%
'marital_index',	-4.13887764	0.015940733	-98.406%
'education_index',	-1.91093382	0.147942171	-85.206%
'default_index',	-1.38065349	0.251414203	-74.859%
'balance',	-1.28927482	0.275470476	-72.453%
'housing_index',	-1.21337854	0.297191508	-70.281%
'loan_index',	-0.83258525	0.434923446	-56.508%
'campaign',	-0.74008648	0.477072656	-52.293%
'pdays',	0.20288795	1.224935207	22.494%
'previous',	1.02935382	2.799256427	179.926%
'poutcome_index',	16.65021614	17025386.98	1702538598.265%

Figure 9: Logistic Regression Coefficients:

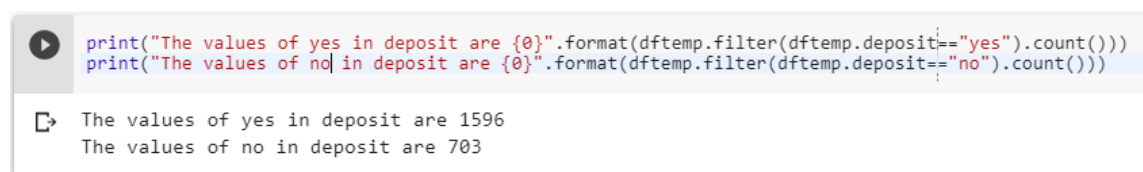
- For on unit increase in “age”, decreases the likelihood of subscribing a term deposit by 99.708% on holding other variables constant.
- For on unit increase in “job Index”, decreases the likelihood of subscribing a term deposit by 99.093% on holding other variables constant.
- For on unit increase in “marital\_index”, decreases the likelihood of subscribing a term deposit by 98.406% on holding other variables constant.
- For on unit increase in “education\_index”, decreases the likelihood of subscribing a term deposit by 85.206% on holding other variables constant.
- For on unit increase in “default\_index”, decreases the likelihood of subscribing a term deposit by 74.859% on holding other variables constant.



- For on unit increase in “balance”, decreases the likelihood of subscribing a term deposit by 72.453% on holding other variables constant.
- For on unit increase in “housing\_index”, decreases the likelihood of subscribing a term deposit by 70.281% on holding other variables constant.
- For on unit increase in “loan\_index”, decreases the likelihood of subscribing a term deposit by 56.508% on holding other variables constant.
- For on unit increase in “campaign”, decreases the likelihood of subscribing a term deposit by 52.293% on holding other variables constant.
- For on unit increase in “pday”, increases the likelihood of subscribing a term deposit by 22.494% on holding other variables constant.
- For on unit increase in “previous”, increases the likelihood of subscribing a term deposit by 179.926% on holding other variables constant.

## 5.0 Analysis and Improvement:

- The model has utilized same data for training and testing the performance of model which makes the model overfit the data.
- Bootstrapping could help in building a more effective models in predicting the output as we can see that the accuracy for all the models is nearly equal to one and it may indicates the models have memorised the results.



```
print("The values of yes in deposit are {}".format(dftemp.filter(dftemp.deposit=="yes").count()))
print("The values of no in deposit are {}".format(dftemp.filter(dftemp.deposit=="no").count()))
```

```
The values of yes in deposit are 1596
The values of no in deposit are 703
```

The label is unbalanced, so there is a need for balancing the label by up sampling the label.

## 6.0 Team Work:

From the day 1 after forming the group for assignment 2 we are so determined, passionate to complete the assignment. All our group members are equally done their part to accomplishing the results. We used to share our ideas and knowledge in doing the assignment. Our knowledge in the area of modern data science has been excelled by doing this assignment. We learned so many new things such as team work, time management, leadership qualities etc. and made new friends.