PySpark

Pyspark is a Python module that allows you to communicate using Apache Spark. It varies to Python within this Spark data frameworks are mirrored, despite the fact that it lets you to construct Spark applications utilizing Python code. Spark includes a number of functionalities, such as Spark SQL as well as the Machine Learning Library. A Spark application consists of a driver software that performs numerous concurrent operations on a cluster while the user's primary goal is accomplished.

In this project, I used PySpark to explore, clean, evaluate and showcase fairly dataset. This project follows supervised machine learning algorithms, which involves acquiring data, analysing it, pre-processing it, selecting the best model, training it, and evaluating it.The necessary python libraries are imported in order to run the code. I tried to make a Binary Classification application with PySpark and the Machine Learning Library's Pipelines API. The Logistic Regression algorithm beat the Decision Tree, Random Forest, and Gradient Boosted Tree methods on the data set.

Column Information:

- age (numeric)
- job : type of job (categorical:admin.','bluecollar','entrepreneur','housemaid','management','retired','selfemployed',
- marital : marital status (categorical:'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
- education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unk
- default: has credit in default? (categorical: 'no','yes','unknown')
- balance: average yearly balance, in euros (numeric)
- housing: has housing loan? (categorical: 'no','yes','unknown')
- loan: has personal loan? (categorical: 'no','yes','unknown')
- contact: contact communication type (categorical:'cellular','telephone')
- day: last contact day of the month (numeric 1 -31)
- month: last contact month of year (categorical: 'jan', 'feb','mar', …, 'nov', 'dec')
- duration: last contact duration, in seconds (numeric).
- Important note: this attribute highly affects the output target (e.g., ifduration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known.Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
- campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
- previous: number of contacts performed before this campaign and for this client (numeric)
- poutcome: outcome of the previous marketing campaign(categorical: 'failure','nonexistent','success')

- target: has the client subscribed a term deposit? (binary:"yes","no")

In [1]:
```
import pyspark
```

In [2]:
```
from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark import SparkConf
from pyspark.sql.functions import desc
```

In [3]:
```
from pyspark.sql.functions import *
from pyspark.sql.functions import max as sparkMax
import pyspark.sql.functions as F
```

SparkSession
.builder
.master("local[*]")
.appName("Pyspark")
.config("spark.memory.fraction", 0.8)
.config("spark.executor.memory", "16g")
.config("spark.driver.memory", "16g")
.config("spark.sql.shuffle.partitions" , "800")
.config("spark.memory.offHeap.enabled",'true')
.config("spark.memory.offHeap.size","16g")
.getOrCreate()

#Creating a Spark session mySpark = SparkSession.builder.getOrCreate() spark = SparkSession(myOwnSpark)

In [4]:
```
spark = SparkSession.builder.appName('mySparkProject').getOrCreate()
```

In [6]:
```
#importing the required libraries
import pandas as pd
import numpy as np
import seaborn as sns

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt

from scipy.sparse import csr_matrix

import warnings; warnings.filterwarnings(action='ignore')
```

In [7]:
```
#loading the dataset
data = spark.read.csv("bank_data.csv",inferSchema=True, header=True)
```

Data exploration using using some of spark analysis techniques

Column names and data types

In [8]:
```
data.printSchema()
```

```
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)
 |-- Target: string (nullable = true)
```

In [9]:
```
data.count()
```

Out[9]:
```
45211
```

Our dataset has 45,211 rows.

In [10]:
```
data.show(10)
```

```
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
|age|        job| marital|education|default|balance|housing|loan|contact|day|m
onth|duration|campaign|pdays|previous|poutcome|Target|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
| 58|  management| married| tertiary|     no|   2143|    yes|  no|unknown|  5|
may|     261|       1|   -1|       0| unknown|    no|
| 44|  technician|  single|secondary|     no|     29|    yes|  no|unknown|  5|
may|     151|       1|   -1|       0| unknown|    no|
| 33|entrepreneur| married|secondary|     no|      2|    yes| yes|unknown|  5|
may|      76|       1|   -1|       0| unknown|    no|
| 47| blue-collar| married|  unknown|     no|   1506|    yes|  no|unknown|  5|
may|      92|       1|   -1|       0| unknown|    no|
| 33|     unknown|  single|  unknown|     no|      1|     no|  no|unknown|  5|
may|     198|       1|   -1|       0| unknown|    no|
| 35|  management| married| tertiary|     no|    231|    yes|  no|unknown|  5|
may|     139|       1|   -1|       0| unknown|    no|
| 28|  management|  single| tertiary|     no|    447|    yes| yes|unknown|  5|
may|     217|       1|   -1|       0| unknown|    no|
| 42|entrepreneur|divorced| tertiary|    yes|      2|    yes|  no|unknown|  5|
may|     380|       1|   -1|       0| unknown|    no|
| 58|     retired| married|  primary|     no|    121|    yes|  no|unknown|  5|
may|      50|       1|   -1|       0| unknown|    no|
| 43|  technician|  single|secondary|     no|    593|    yes|  no|unknown|  5|
may|      55|       1|   -1|       0| unknown|    no|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
only showing top 10 rows
```

Removal of rows & columns with Null values

In [11]:
```
data.dropna()
```

Out[11]:
```
DataFrame[age: int, job: string, marital: string, education: string, default: s
tring, balance: int, housing: string, loan: string, contact: string, day: int,
month: string, duration: int, campaign: int, pdays: int, previous: int, poutcom
e: string, Target: string]
```

In [12]:
```
data.show(10)
```

```
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
|age|        job| marital|education|default|balance|housing|loan|contact|day|m
onth|duration|campaign|pdays|previous|poutcome|Target|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
| 58|  management| married| tertiary|     no|   2143|    yes|  no|unknown|  5|
may|     261|       1|   -1|       0| unknown|    no|
| 44|  technician|  single|secondary|     no|     29|    yes|  no|unknown|  5|
may|     151|       1|   -1|       0| unknown|    no|
| 33|entrepreneur| married|secondary|     no|      2|    yes| yes|unknown|  5|
may|      76|       1|   -1|       0| unknown|    no|
| 47| blue-collar| married|  unknown|     no|   1506|    yes|  no|unknown|  5|
may|      92|       1|   -1|       0| unknown|    no|
| 33|     unknown|  single|  unknown|     no|      1|     no|  no|unknown|  5|
may|     198|       1|   -1|       0| unknown|    no|
| 35|  management| married| tertiary|     no|    231|    yes|  no|unknown|  5|
may|     139|       1|   -1|       0| unknown|    no|
| 28|  management|  single| tertiary|     no|    447|    yes| yes|unknown|  5|
may|     217|       1|   -1|       0| unknown|    no|
| 42|entrepreneur|divorced| tertiary|    yes|      2|    yes|  no|unknown|  5|
may|     380|       1|   -1|       0| unknown|    no|
| 58|     retired| married|  primary|     no|    121|    yes|  no|unknown|  5|
may|      50|       1|   -1|       0| unknown|    no|
| 43|  technician|  single|secondary|     no|    593|    yes|  no|unknown|  5|
may|      55|       1|   -1|       0| unknown|    no|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
only showing top 10 rows
```

In [13]:
```
data.count()
```

Out[13]:
```
45211
```

In [14]:
```
#Removal of duplicate values
data.dropDuplicates()
```

Out[14]:
```
DataFrame[age: int, job: string, marital: string, education: string, default: s
tring, balance: int, housing: string, loan: string, contact: string, day: int,
month: string, duration: int, campaign: int, pdays: int, previous: int, poutcom
e: string, Target: string]
```

In [16]:
```
data.count()
```

Out[16]:
```
45211
```

In [17]:
```
#check the null column count using sql
from pyspark.sql.functions import col,isnan, when, count
data.select([count(when(isnan(a) | col(a).isNull(), a)).alias(a) for a in data.co
```

```
+---+---+-------+---------+-------+-------+-------+----+-------+---+-----+-----
---+--------+-----+--------+--------+------+
|age|job|marital|education|default|balance|housing|loan|contact|day|month|durat
ion|campaign|pdays|previous|poutcome|Target|
+---+---+-------+---------+-------+-------+-------+----+-------+---+-----+-----
---+--------+-----+--------+--------+------+
|  0|  0|      0|        0|      0|      0|      0|   0|      0|  0|    0|
0|       0|    0|       0|       0|     0|
+---+---+-------+---------+-------+-------+-------+----+-------+---+-----+-----
---+--------+-----+--------+--------+------+
```

Conversion of values to integer:

We will convert the columns with yes/no values to 1 and 0 for a proper data analysis.

In [18]:
```
df = data
```

In [19]:
```
df = df.replace('yes','1')
df = df.replace('no','0')
```

In [20]:
```
df = df.withColumn("default",df.default.cast('integer'))
df = df.withColumn("loan",df.loan.cast('integer'))
df = df.withColumn("housing",df.housing.cast('integer'))
df = df.withColumn("Target",df.Target.cast('integer'))
```

In [21]:
```
#Display of column name and data type
df.printSchema()
```

```
root
 |-- age: integer (nullable = true)
 |-- job: string (nullable = true)
 |-- marital: string (nullable = true)
 |-- education: string (nullable = true)
 |-- default: integer (nullable = true)
 |-- balance: integer (nullable = true)
 |-- housing: integer (nullable = true)
 |-- loan: integer (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)
 |-- Target: integer (nullable = true)
```

We're now doing a number of analysis to learn more about the dataset characteristics.

The findings of univariant and multivariant analysis are shown below.

Group By

In [22]:
```
data.groupBy("month").count().show()
```

```
+-----+-----+
|month|count|
+-----+-----+
|  jun| 5341|
|  aug| 6247|
|  may|13766|
|  feb| 2649|
|  sep|  579|
|  mar|  477|
|  oct|  738|
|  jul| 6895|
|  nov| 3970|
|  apr| 2932|
|  dec|  214|
|  jan| 1403|
+-----+-----+
```

In [23]:
```
data.groupBy("job").count().show()
```

```
+-------------+-----+
|          job|count|
+-------------+-----+
|   management| 9458|
|      retired| 2264|
|      unknown|  288|
|self-employed| 1579|
|      student|  938|
|  blue-collar| 9732|
| entrepreneur| 1487|
|       admin.| 5171|
|   technician| 7597|
|     services| 4154|
|    housemaid| 1240|
|   unemployed| 1303|
+-------------+-----+
```

In [24]:
```
data.groupBy("marital").count().show()
```

```
+--------+-----+
| marital|count|
+--------+-----+
|divorced| 5207|
| married|27214|
|  single|12790|
+--------+-----+
```

In [25]:
```
data.groupBy("education").count().show()
```

```
+---------+-----+
|education|count|
+---------+-----+
|  unknown| 1857|
| tertiary|13301|
|secondary|23202|
|  primary| 6851|
+---------+-----+
```

In [26]:
```
data.groupBy("housing").count().show()
```

```
+-------+-----+
|housing|count|
+-------+-----+
|     no|20081|
|    yes|25130|
+-------+-----+
```

In [27]:
```
data.groupBy("loan").count().show()
```

```
+----+-----+
|loan|count|
+----+-----+
|  no|37967|
| yes| 7244|
+----+-----+
```

In [28]:
```
data.groupBy("contact").count().show()
```

```
+---------+-----+
|  contact|count|
+---------+-----+
|  unknown|13020|
| cellular|29285|
|telephone| 2906|
+---------+-----+
```

In [29]:
```
data.groupBy("campaign").count().show()
```

```
+--------+-----+
|campaign|count|
+--------+-----+
|      31|   12|
|      34|    5|
|      28|   16|
|      26|   13|
|      27|   10|
|      44|    1|
|      12|  155|
|      22|   23|
|       1|17544|
|      13|  133|
|       6| 1291|
|      16|   79|
|       3| 5521|
|      20|   43|
|       5| 1764|
|      19|   44|
|      41|    2|
|      15|   84|
|      43|    3|
|      37|    2|
+--------+-----+
only showing top 20 rows
```

In [30]:
```
data.groupBy("poutcome").count().show()
```

```
+--------+-----+
|poutcome|count|
+--------+-----+
| success| 1511|
| unknown|36959|
|   other| 1840|
| failure| 4901|
+--------+-----+
```

In [31]:
```
data.groupBy("target").count().show()
```

```
+------+-----+
|target|count|
+------+-----+
|    no|39922|
|   yes| 5289|
+------+-----+
```

In [32]:
```
data.filter((F.col('poutcome')=='unknown'))\
    .filter(
        (F.col('target') == 'yes')
    )\
    .show()
```

```
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+--
---+--------+--------+-----+--------+--------+------+
|age|        job| marital|education|default|balance|housing|loan|contact|day|mo
nth|duration|campaign|pdays|previous|poutcome|Target|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+--
---+--------+--------+-----+--------+--------+------+
| 59|     admin.| married|secondary|     no|   2343|    yes|  no|unknown|  5|
may|    1042|       1|   -1|       0| unknown|   yes|
| 56|     admin.| married|secondary|     no|     45|     no|  no|unknown|  5|
may|    1467|       1|   -1|       0| unknown|   yes|
| 41| technician| married|secondary|     no|   1270|    yes|  no|unknown|  5|
may|    1389|       1|   -1|       0| unknown|   yes|
| 55|   services| married|secondary|     no|   2476|    yes|  no|unknown|  5|
may|     579|       1|   -1|       0| unknown|   yes|
| 54|     admin.| married| tertiary|     no|    184|     no|  no|unknown|  5|
may|     673|       2|   -1|       0| unknown|   yes|
| 42| management|  single| tertiary|     no|      0|    yes| yes|unknown|  5|
may|     562|       2|   -1|       0| unknown|   yes|
| 56| management| married| tertiary|     no|    830|    yes| yes|unknown|  6|
may|    1201|       1|   -1|       0| unknown|   yes|
| 60|    retired|divorced|secondary|     no|    545|    yes|  no|unknown|  6|
may|    1030|       1|   -1|       0| unknown|   yes|
| 39| technician|  single|  unknown|     no|  45248|    yes|  no|unknown|  6|
may|    1623|       1|   -1|       0| unknown|   yes|
| 37| technician| married|secondary|     no|      1|    yes|  no|unknown|  6|
may|     608|       1|   -1|       0| unknown|   yes|
| 34|     admin.| married|secondary|     no|    869|     no|  no|unknown|  6|
may|    1677|       1|   -1|       0| unknown|   yes|
| 55| unemployed|divorced|secondary|     no|    387|    yes|  no|unknown|  6|
may|     918|       1|   -1|       0| unknown|   yes|
| 28|   services|  single|secondary|     no|   5090|    yes|  no|unknown|  6|
may|    1297|       3|   -1|       0| unknown|   yes|
| 30| technician| married|secondary|     no|    484|    yes|  no|unknown|  6|
may|     703|       1|   -1|       0| unknown|   yes|
| 36| technician| married|secondary|     no|    368|    yes| yes|unknown|  6|
may|    1597|       2|   -1|       0| unknown|   yes|
| 37|     admin.|  single|secondary|     no|    245|    yes| yes|unknown|  7|
may|     732|       2|   -1|       0| unknown|   yes|
| 45|blue-collar| married|secondary|     no|    154|    yes|  no|unknown|  7|
may|    1138|       1|   -1|       0| unknown|   yes|
| 53|   services|divorced|  primary|     no|   -291|    yes| yes|unknown|  7|
may|     591|       1|   -1|       0| unknown|   yes|
| 38|     admin.|  single|secondary|     no|    100|    yes|  no|unknown|  7|
may|     786|       1|   -1|       0| unknown|   yes|
| 30|blue-collar| married|secondary|     no|    309|    yes|  no|unknown|  7|
may|    1574|       2|   -1|       0| unknown|   yes|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+--
---+--------+--------+-----+--------+--------+------+
only showing top 20 rows
```

In [33]:
```
data.filter((F.col('poutcome')=='success'))\
    .filter(
        (F.col('target') == 'yes')
    )\
    .show()
```

```
+---+-----------+--------+---------+-------+-------+-------+----+---------+---
+-----+--------+--------+-----+--------+--------+------+
|age|        job| marital|education|default|balance|housing|loan|  contact|day
|month|duration|campaign|pdays|previous|poutcome|Target|
+---+-----------+--------+---------+-------+-------+-------+----+---------+---
+-----+--------+--------+-----+--------+--------+------+
| 56|  technician| married|secondary|     no|    589|    yes|  no|  unknown| 23
|  oct|     518|       1|  147|       2| success|   yes|
| 53|     retired| married| tertiary|     no|   2269|     no|  no| cellular| 17
|  nov|    1091|       2|  150|       1| success|   yes|
| 45|  management|divorced|secondary|     no|    644|    yes|  no| cellular| 19
|  nov|     418|       1|  168|       1| success|   yes|
| 46|  unemployed|divorced|secondary|     no|   3354|    yes|  no| cellular| 19
|  nov|     522|       1|  174|       1| success|   yes|
| 40|  management| married| tertiary|     no|   3352|    yes|  no| cellular| 19
|  nov|     639|       2|   27|       1| success|   yes|
| 40|    services|divorced|secondary|     no|    687|    yes|  no| cellular|  2
|  feb|     531|       1|  208|       1| success|   yes|
| 31|  management| married| tertiary|     no|   1331|     no|  no| cellular|  3
|  feb|     182|       2|   90|       1| success|   yes|
| 40| blue-collar| married|  unknown|     no|   1181|    yes|  no| cellular|  4
|  feb|     718|       2|  189|       2| success|   yes|
| 38|      admin.|divorced|secondary|     no|     19|    yes|  no| cellular|  5
|  feb|    1130|       3|  251|       2| success|   yes|
| 31|  management|  single| tertiary|     no|  12857|    yes|  no| cellular|  6
|  feb|     158|       1|   92|       1| success|   yes|
| 33| blue-collar|  single| tertiary|     no|    700|     no|  no| cellular|  9
|  feb|     126|       1|   88|       1| success|   yes|
| 40| blue-collar| married|secondary|     no|   5060|     no|  no| cellular| 10
|  feb|     154|       2|   93|       1| success|   yes|
| 30|  management|  single| tertiary|     no|   5561|    yes|  no| cellular| 27
|  feb|     195|       1|  100|       1| success|   yes|
| 38|  technician| married|secondary|     no|   5115|    yes|  no| cellular| 27
|  feb|      99|       1|  102|       6| success|   yes|
| 44|  management|  single| tertiary|     no|    483|     no|  no| cellular|  6
|  mar|     207|       2|  199|       6| success|   yes|
| 37|      admin.| married|secondary|     no|   1207|    yes|  no|telephone|  6
|  apr|    1353|       2|  138|       2| success|   yes|
| 52|entrepreneur|  single| tertiary|     no|   3469|    yes|  no| cellular| 16
|  apr|     583|       1|  147|       1| success|   yes|
| 32|  management|divorced| tertiary|     no|    294|    yes|  no| cellular| 17
|  apr|    1095|       1|  274|       5| success|   yes|
| 53|  technician| married|secondary|     no|   5303|    yes|  no|telephone| 17
|  apr|     320|       3|  180|       1| success|   yes|
| 50|     unknown| married|  primary|     no|    341|    yes| yes| cellular| 20
|  apr|     670|       4|  340|       2| success|   yes|
+---+-----------+--------+---------+-------+-------+-------+----+---------+---
+-----+--------+--------+-----+--------+--------+------+
only showing top 20 rows
```

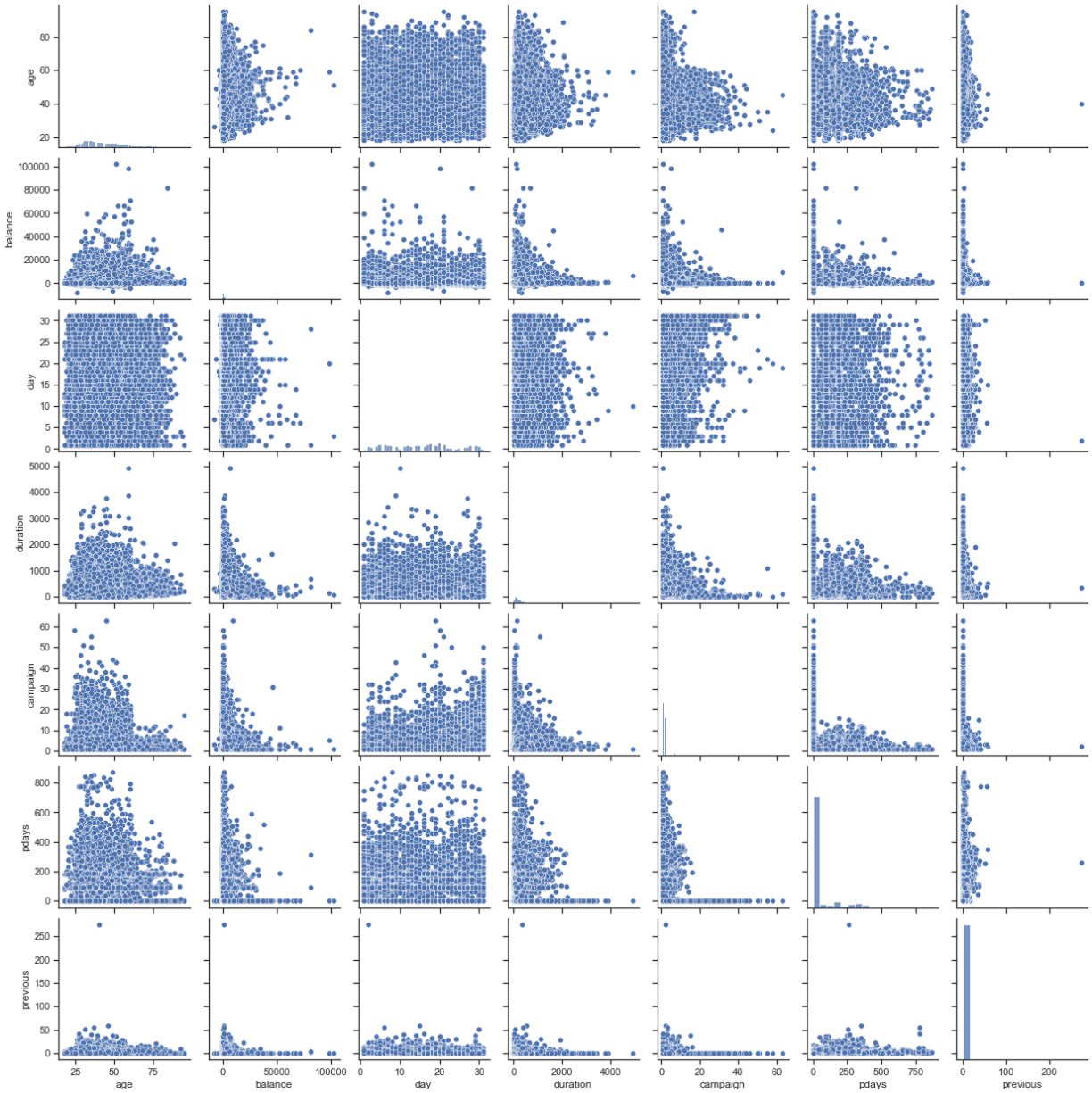
In [34]:
```
data.filter((F.col('poutcome')=='failure'))\
    .filter(
        (F.col('target') == 'yes')
    )\
    .show()
```

```
+---+-------------+--------+---------+-------+-------+-------+----+---------+--
-+-----+--------+--------+-----+--------+--------+------+
|age|          job| marital|education|default|balance|housing|loan|  contact|da
y|month|duration|campaign|pdays|previous|poutcome|Target|
+---+-------------+--------+---------+-------+-------+-------+----+---------+--
-+-----+--------+--------+-----+--------+--------+------+
| 33|     services| married|secondary|     no|   3444|    yes|  no|telephone| 2
1|  oct|     144|       1|   91|       4| failure|   yes|
| 36|   management| married| tertiary|     no|      0|    yes|  no|telephone| 2
3|  oct|     140|       1|  143|       3| failure|   yes|
| 34|       admin.| married| tertiary|     no|    899|    yes|  no|  unknown| 1
2|  nov|     114|       1|  170|       3| failure|   yes|
| 49|     services| married|secondary|     no|    202|    yes|  no| cellular| 1
7|  nov|     651|       2|  104|       1| failure|   yes|
| 37|   technician| married|secondary|     no|   5115|    yes|  no| cellular| 1
7|  nov|    1210|       2|  171|       4| failure|   yes|
| 45| entrepreneur| married|secondary|     no|    781|     no| yes| cellular| 1
7|  nov|     652|       2|  126|       2| failure|   yes|
| 46|self-employed| married| tertiary|     no|   2421|     no|  no| cellular| 1
9|  nov|    1084|       1|  100|       4| failure|   yes|
| 58|      retired|divorced|  primary|     no|   2538|    yes|  no| cellular| 1
9|  nov|     680|       2|  111|       6| failure|   yes|
| 32|   technician| married| tertiary|     no|   4654|    yes| yes| cellular| 2
0|  nov|     276|       1|  128|       2| failure|   yes|
| 30|  blue-collar| married|secondary|     no|    501|    yes| yes| cellular| 2
0|  nov|     994|       1|  177|       1| failure|   yes|
| 46|   technician| married| tertiary|     no|      0|     no|  no| cellular| 2
0|  nov|     531|       1|  167|       1| failure|   yes|
| 38| entrepreneur| married| tertiary|     no|   1110|    yes|  no| cellular| 2
0|  nov|     888|       2|  183|       2| failure|   yes|
| 32|     services| married|secondary|     no|    983|    yes|  no| cellular| 2
0|  nov|     500|       2|  133|       1| failure|   yes|
| 31|   unemployed| married|secondary|     no|    314|    yes|  no| cellular| 2
0|  nov|    1341|       3|  178|       7| failure|   yes|
| 50|  blue-collar| married|  primary|     no|  12519|    yes|  no| cellular| 2
1|  nov|     615|       3|   34|       1| failure|   yes|
| 47|   technician| married|secondary|     no|      0|     no|  no| cellular| 2
1|  nov|     591|       1|   10|       1| failure|   yes|
| 59|   management| married| tertiary|     no|   7049|     no|  no| cellular| 2
1|  nov|     530|       1|  163|       2| failure|   yes|
| 31|self-employed| married| tertiary|     no|      5|    yes| yes| cellular| 2
1|  nov|     635|       1|  135|       2| failure|   yes|
| 31|   management| married|secondary|     no|   8629|    yes|  no| cellular| 2
1|  nov|     957|       1|  184|       2| failure|   yes|
| 53|  blue-collar| married|secondary|     no|   1777|    yes|  no| cellular| 2
1|  nov|     796|       5|  154|       1| failure|   yes|
+---+-------------+--------+---------+-------+-------+-------+----+---------+--
-+-----+--------+--------+-----+--------+--------+------+
only showing top 20 rows
```

Let's carry out some visualizations on the data

In [35]:
```
sns.set(style="ticks")

sns.pairplot(data.toPandas())
plt.show()
```

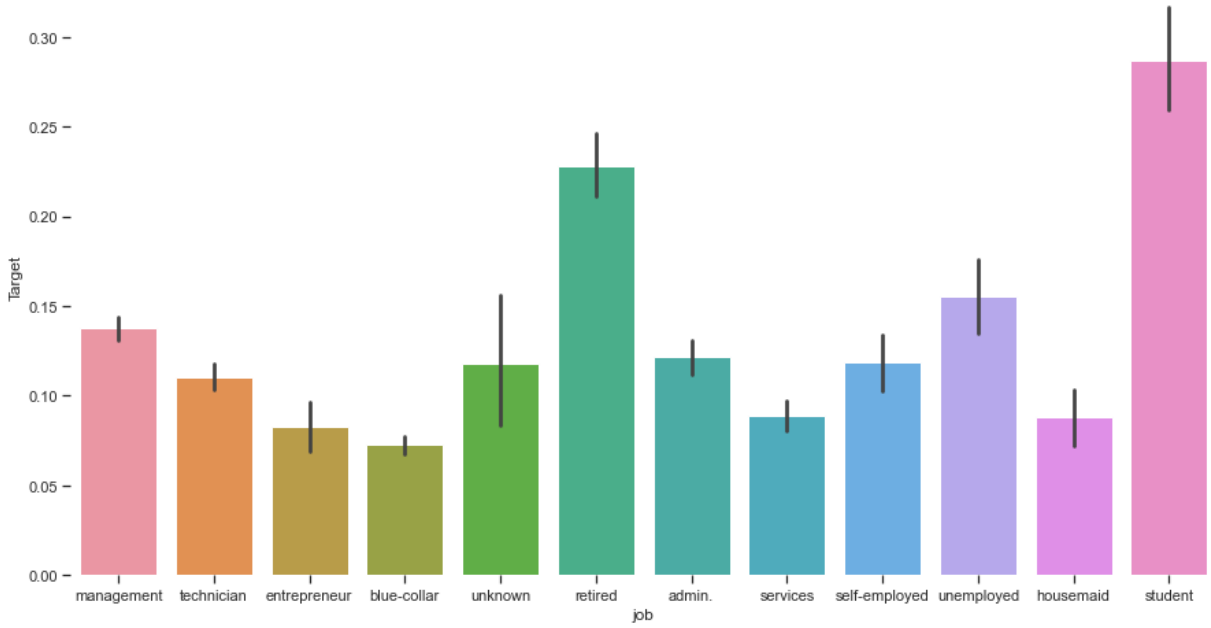


In [36]:
```
df.show()
```

```
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
|age|        job| marital|education|default|balance|housing|loan|contact|day|m
onth|duration|campaign|pdays|previous|poutcome|Target|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
| 58|  management| married| tertiary|      0|   2143|      1|   0|unknown|  5|
may|     261|       1|   -1|       0| unknown|     0|
| 44|  technician|  single|secondary|      0|     29|      1|   0|unknown|  5|
may|     151|       1|   -1|       0| unknown|     0|
| 33|entrepreneur| married|secondary|      0|      2|      1|   1|unknown|  5|
may|      76|       1|   -1|       0| unknown|     0|
| 47| blue-collar| married|  unknown|      0|   1506|      1|   0|unknown|  5|
may|      92|       1|   -1|       0| unknown|     0|
| 33|     unknown|  single|  unknown|      0|      1|      0|   0|unknown|  5|
may|     198|       1|   -1|       0| unknown|     0|
| 35|  management| married| tertiary|      0|    231|      1|   0|unknown|  5|
may|     139|       1|   -1|       0| unknown|     0|
| 28|  management|  single| tertiary|      0|    447|      1|   1|unknown|  5|
may|     217|       1|   -1|       0| unknown|     0|
| 42|entrepreneur|divorced| tertiary|      1|      2|      1|   0|unknown|  5|
may|     380|       1|   -1|       0| unknown|     0|
| 58|     retired| married|  primary|      0|    121|      1|   0|unknown|  5|
may|      50|       1|   -1|       0| unknown|     0|
| 43|  technician|  single|secondary|      0|    593|      1|   0|unknown|  5|
may|      55|       1|   -1|       0| unknown|     0|
| 41|      admin.|divorced|secondary|      0|    270|      1|   0|unknown|  5|
may|     222|       1|   -1|       0| unknown|     0|
| 29|      admin.|  single|secondary|      0|    390|      1|   0|unknown|  5|
may|     137|       1|   -1|       0| unknown|     0|
| 53|  technician| married|secondary|      0|      6|      1|   0|unknown|  5|
may|     517|       1|   -1|       0| unknown|     0|
| 58|  technician| married|  unknown|      0|     71|      1|   0|unknown|  5|
may|      71|       1|   -1|       0| unknown|     0|
| 57|    services| married|secondary|      0|    162|      1|   0|unknown|  5|
may|     174|       1|   -1|       0| unknown|     0|
| 51|     retired| married|  primary|      0|    229|      1|   0|unknown|  5|
may|     353|       1|   -1|       0| unknown|     0|
| 45|      admin.|  single|  unknown|      0|     13|      1|   0|unknown|  5|
may|      98|       1|   -1|       0| unknown|     0|
| 57| blue-collar| married|  primary|      0|     52|      1|   0|unknown|  5|
may|      38|       1|   -1|       0| unknown|     0|
| 60|     retired| married|  primary|      0|     60|      1|   0|unknown|  5|
may|     219|       1|   -1|       0| unknown|     0|
| 33|    services| married|secondary|      0|      0|      1|   0|unknown|  5|
may|      54|       1|   -1|       0| unknown|     0|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
only showing top 20 rows
```

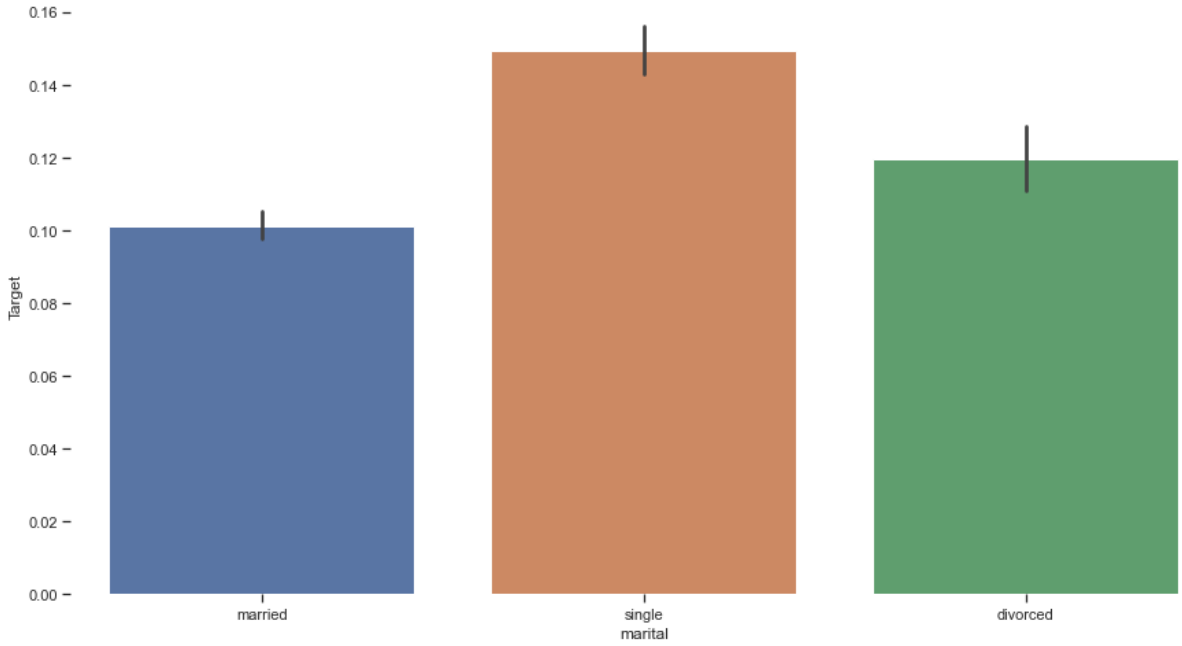
In [37]:
```
f, ax = plt.subplots(figsize = (15,8))
sns.barplot(x="job", y = "Target", data = df.toPandas())
sns.despine(left = True, bottom = True)
ax.set(xlabel='job', ylabel='Target')
```

Out[37]:
```
[Text(0.5, 0, 'job'), Text(0, 0.5, 'Target')]
```



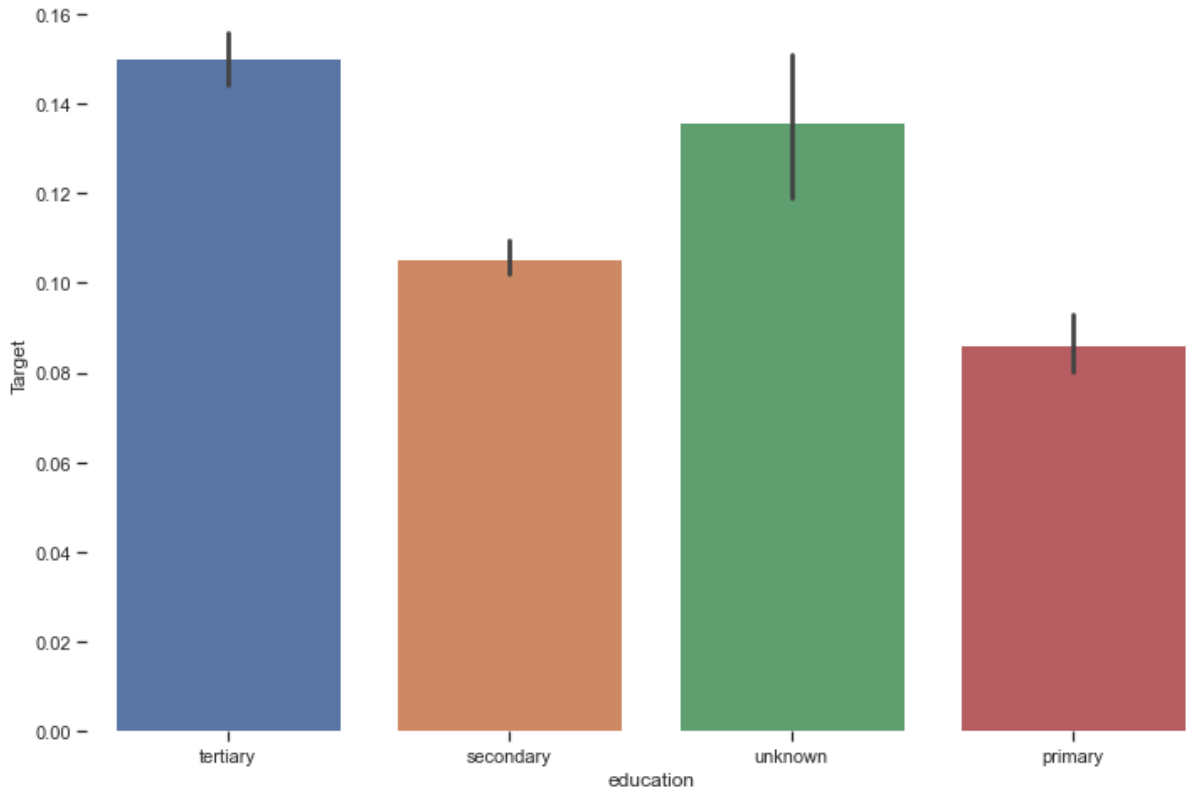
In [38]:
```
f, ax = plt.subplots(figsize = (15,8))
sns.barplot(x="marital", y = "Target", data = df.toPandas())
sns.despine(left = True, bottom = True)
ax.set(xlabel='marital', ylabel='Target')
```

Out[38]:
```
[Text(0.5, 0, 'marital'), Text(0, 0.5, 'Target')]
```



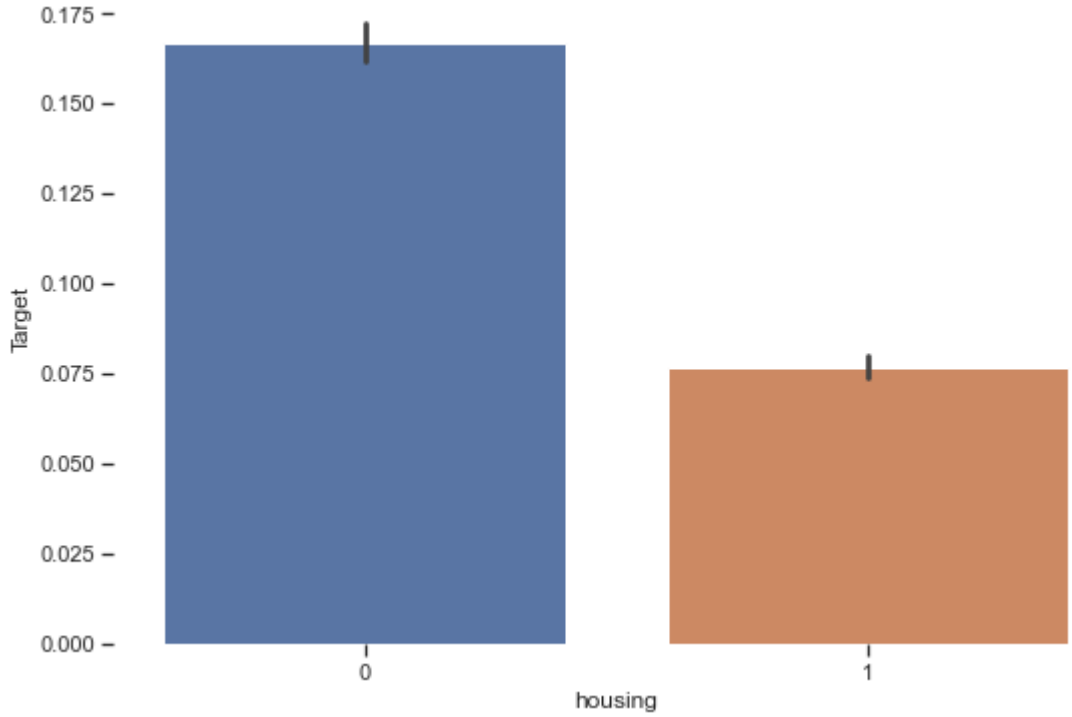
In [39]:
```
f, ax = plt.subplots(figsize = (12,8))
sns.barplot(x="education", y = "Target", data = df.toPandas())
sns.despine(left = True, bottom = True)
ax.set(xlabel='education', ylabel='Target')
```

Out[39]:
```
[Text(0.5, 0, 'education'), Text(0, 0.5, 'Target')]
```



In [40]:
```
f, ax = plt.subplots(figsize = (9,6))
sns.barplot(x="housing", y = "Target", data = df.toPandas())
sns.despine(left = True, bottom = True)
ax.set(xlabel='housing', ylabel='Target')
```

Out[40]:
```
[Text(0.5, 0, 'housing'), Text(0, 0.5, 'Target')]
```



In [41]:
```
df.show()
```

```
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
|age|        job| marital|education|default|balance|housing|loan|contact|day|m
onth|duration|campaign|pdays|previous|poutcome|Target|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
| 58|  management| married| tertiary|      0|   2143|      1|   0|unknown|  5|
may|     261|       1|   -1|       0| unknown|     0|
| 44|  technician|  single|secondary|      0|     29|      1|   0|unknown|  5|
may|     151|       1|   -1|       0| unknown|     0|
| 33|entrepreneur| married|secondary|      0|      2|      1|   1|unknown|  5|
may|      76|       1|   -1|       0| unknown|     0|
| 47| blue-collar| married|  unknown|      0|   1506|      1|   0|unknown|  5|
may|      92|       1|   -1|       0| unknown|     0|
| 33|     unknown|  single|  unknown|      0|      1|      0|   0|unknown|  5|
may|     198|       1|   -1|       0| unknown|     0|
| 35|  management| married| tertiary|      0|    231|      1|   0|unknown|  5|
may|     139|       1|   -1|       0| unknown|     0|
| 28|  management|  single| tertiary|      0|    447|      1|   1|unknown|  5|
may|     217|       1|   -1|       0| unknown|     0|
| 42|entrepreneur|divorced| tertiary|      1|      2|      1|   0|unknown|  5|
may|     380|       1|   -1|       0| unknown|     0|
| 58|     retired| married|  primary|      0|    121|      1|   0|unknown|  5|
may|      50|       1|   -1|       0| unknown|     0|
| 43|  technician|  single|secondary|      0|    593|      1|   0|unknown|  5|
may|      55|       1|   -1|       0| unknown|     0|
| 41|      admin.|divorced|secondary|      0|    270|      1|   0|unknown|  5|
may|     222|       1|   -1|       0| unknown|     0|
| 29|      admin.|  single|secondary|      0|    390|      1|   0|unknown|  5|
may|     137|       1|   -1|       0| unknown|     0|
| 53|  technician| married|secondary|      0|      6|      1|   0|unknown|  5|
may|     517|       1|   -1|       0| unknown|     0|
| 58|  technician| married|  unknown|      0|     71|      1|   0|unknown|  5|
may|      71|       1|   -1|       0| unknown|     0|
| 57|    services| married|secondary|      0|    162|      1|   0|unknown|  5|
may|     174|       1|   -1|       0| unknown|     0|
| 51|     retired| married|  primary|      0|    229|      1|   0|unknown|  5|
may|     353|       1|   -1|       0| unknown|     0|
| 45|      admin.|  single|  unknown|      0|     13|      1|   0|unknown|  5|
may|      98|       1|   -1|       0| unknown|     0|
| 57| blue-collar| married|  primary|      0|     52|      1|   0|unknown|  5|
may|      38|       1|   -1|       0| unknown|     0|
| 60|     retired| married|  primary|      0|     60|      1|   0|unknown|  5|
may|     219|       1|   -1|       0| unknown|     0|
| 33|    services| married|secondary|      0|      0|      1|   0|unknown|  5|
may|      54|       1|   -1|       0| unknown|     0|
+---+-----------+--------+---------+-------+-------+-------+----+-------+---+-
----+--------+--------+-----+--------+--------+------+
only showing top 20 rows
```

In [42]:
```
cols = df.columns
```

Machine Learning

We will make use of One-Hot Encoder, String Indexer, and VectorAssembler, a distinctive converter that merges many columns into a vector column. The code below, taken directly from databrick's website, utilises the StringIndexer to index each category column, then combines the indexed groups into a single encrypted parameter. The binary trajectories are appended to the end of each row in the final result. We will then use the StringIndexer once more to encode our labels to label indices. After that, we use the VectorAssembler to combine all of the feature columns into a single supervector column.

In [43]:
```
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorAssembler
categoricalColumns = ['job', 'marital', 'education', 'contact', 'month', 'poutcom
stages = []
for categoricalCol in categoricalColumns:
    stringIndexer = StringIndexer(inputCol = categoricalCol, outputCol = categori
    encoder = OneHotEncoder(inputCols=[stringIndexer.getOutputCol()], outputCols=
    stages += [stringIndexer, encoder]
label_stringIdx = StringIndexer(inputCol = 'Target', outputCol = 'label')
stages += [label_stringIdx]
numericCols = ['age', 'default', 'balance', 'housing', 'loan', 'day', 'duration',
assemblerInputs = [c + "classVec" for c in categoricalColumns] + numericCols
assembler = VectorAssembler(inputCols=assemblerInputs, outputCol="Subscribed")
stages += [assembler]
```

Pipeline: We use the term "pipeline" to characterise our machine learning technique called "Pipeline", which involves chaining together a number of Generators and Estimation methods.

In [44]:
```
from pyspark.ml import Pipeline
pipeline = Pipeline(stages = stages)
pipelineModel = pipeline.fit(df)
df = pipelineModel.transform(df)
selectedCols = ['label', 'Subscribed'] + cols
df = df.select(selectedCols)
df.printSchema()
```

```
root
 |-- label: double (nullable = false)
 |-- Subscribed: vector (nullable = true)
 |-- age: integer (nullable = true)
 |-- job: string (nullable = true)
 |-- marital: string (nullable = true)
 |-- education: string (nullable = true)
 |-- default: integer (nullable = true)
 |-- balance: integer (nullable = true)
 |-- housing: integer (nullable = true)
 |-- loan: integer (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)
 |-- Target: integer (nullable = true)
```

In [45]:
```
pd.DataFrame(df.take(5), columns=df.columns).transpose()
```

Out[45]:

	0	1	2	3	4
label	0.0	0.0	0.0	0.0	0.0
Subscribed	(0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, ...	(1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
age	58	44	33	47	33
job	management	technician	entrepreneur	blue-collar	unknown
marital	married	single	married	married	single
education	tertiary	secondary	secondary	unknown	unknown
default	0	0	0	0	0
balance	2143	29	2	1506	1
housing	1	1	1	1	0
loan	0	0	1	0	0
contact	unknown	unknown	unknown	unknown	unknown
day	5	5	5	5	5
month	may	may	may	may	may
duration	261	151	76	92	198
campaign	1	1	1	1	1
pdays	-1	-1	-1	-1	-1
previous	0	0	0	0	0
poutcome	unknown	unknown	unknown	unknown	unknown
Target	0	0	0	0	0

Train and Test Split by a ratio of 80% to 20%

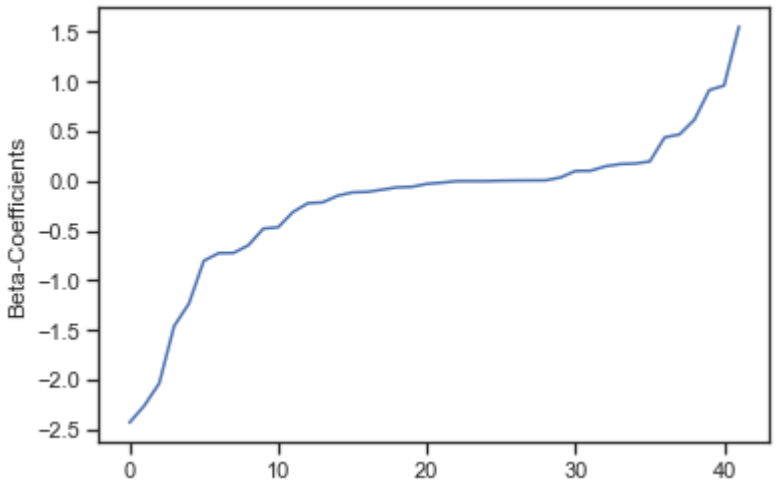
In [46]:
```
train, test = df.randomSplit([0.8, 0.2], seed = 2022)
print("Training Dataset Count: " + str(train.count()))
print("Test Dataset Count: " + str(test.count()))
```

```
Training Dataset Count: 36061
Test Dataset Count: 9150
```

Logistic Regression

In [47]:
```
from pyspark.ml.classification import LogisticRegression
lr = LogisticRegression(featuresCol = 'Subscribed', labelCol = 'label', maxIter=1
lrModel = lr.fit(train)
```

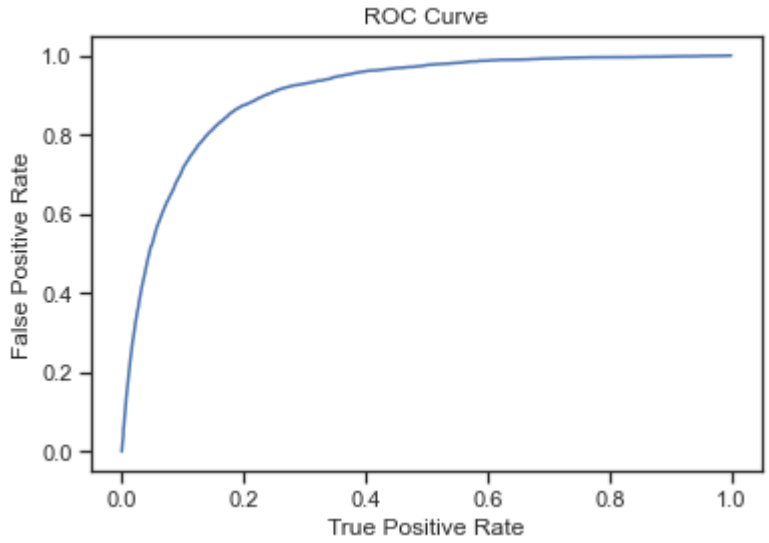
In [48]:
```
beta = np.sort(lrModel.coefficients)
plt.plot(beta)
plt.ylabel('Beta-Coefficients')
plt.show()
```



ROC (Receiver Operating Characteristics) curve

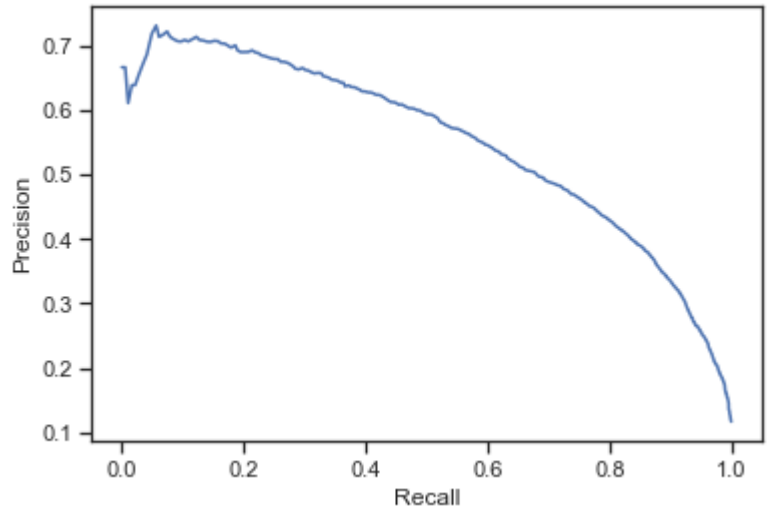
The ROC curve will help us evaluate the prediction power of our model.

In [53]:
```
trainingSummary = lrModel.summary
roc = trainingSummary.roc.toPandas()
plt.plot(roc['FPR'],roc['TPR'])
plt.ylabel('False Positive Rate')
plt.xlabel('True Positive Rate')
plt.title('ROC Curve')
plt.show()
print('Training set area Under ROC: ' + str(trainingSummary.areaUnderROC))
```




```
Training set area Under ROC: 0.9065897716294987
```

In [54]:
```
pr = trainingSummary.pr.toPandas()
plt.plot(pr['recall'],pr['precision'])
plt.ylabel('Precision')
plt.xlabel('Recall')
plt.show()
```



In [55]:
```
predictions = lrModel.transform(test)
predictions.filter((F.col('prediction') == 1.0))\
    .show()
```

```
+-----+--------------------+---+----------+-------+---------+-------+-------+-
------+----+---------+---+-----+--------+--------+-----+--------+--------+-----
-+--------------------+--------------------+----------+
|label|          Subscribed|age|       job|marital|education|default|balance|h
ousing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|Targe
t|       rawPrediction|         probability|prediction|
+-----+--------------------+---+----------+-------+---------+-------+-------+-
------+----+---------+---+-----+--------+--------+-----+--------+--------+-----
-+--------------------+--------------------+----------+
|  0.0|(42,[0,11,13,16,1...| 36|blue-collar|married|secondary|      0|    195|
1|   1| cellular|  7|  may|    1297|       2|   -1|       0| unknown|     0|[-
0.9348452366197...|[0.28194275576627...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 27|blue-collar|married|secondary|      0|   1295|
1|   0| cellular| 14|  may|    1106|       1|   -1|       0| unknown|     0|[-
0.7373382714164...|[0.32358646451319...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 40|blue-collar|married|secondary|      0|    356|
1|   1| cellular|  7|  may|    1254|       3|  365|       1| failure|     0|[-
0.7280340572398...|[0.32562628819413...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 32|blue-collar|married|secondary|      0|   9714|
1|   0| cellular| 15|  may|    1237|       2|  361|       2| failure|     0|[-
1.3548003976345...|[0.20508667357502...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 48|blue-collar|married|secondary|      0|   1513|
0|   1| cellular| 17|  jul|    1171|       1|   -1|       0| unknown|     0|[-
0.8768608251502...|[0.29382871514779...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 53|blue-collar|married|secondary|      0|   3079|
0|   0| cellular|  5|  apr|     759|       1|   -1|       0| unknown|     0|[-
0.3531504387401...|[0.41261865694525...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 35|blue-collar|married|secondary|      0|   1207|
1|   0| cellular| 17|  apr|     694|       2|  331|       1| success|     0|[-
1.6691636032686...|[0.15853572410195...|       1.0|
|  0.0|(42,[0,11,13,17,1...| 38|blue-collar|married|secondary|      0|    376|
1|   0|  unknown|  7|  may|    1521|       1|   -1|       0| unknown|     0|[-
0.8721291498641...|[0.29481146341910...|       1.0|
|  0.0|(42,[0,11,13,17,1...| 38|blue-collar|married|secondary|      0|    384|
1|   0|  unknown|  6|  may|    1906|       3|   -1|       0| unknown|     0|[-
2.3229277366275...|[0.08924181325646...|       1.0|
|  0.0|(42,[0,11,13,18,2...| 32|blue-collar|married|secondary|      0|    485|
1|   0|telephone| 12|  may|    1100|       2|   -1|       0| unknown|     0|[-
0.5002320803868...|[0.37748613059593...|       1.0|
|  0.0|(42,[0,11,15,16,1...| 56|blue-collar|married|  primary|      0|   -116|
1|   0| cellular|  8|  may|    1014|       2|   -1|       0| unknown|     0|[-
0.0469196907492...|[0.48827222874946...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 39|blue-collar|married|  primary|      0|    394|
1|   0| cellular| 17|  apr|     981|       2|   -1|       0| unknown|     0|[-
0.3905306233159...|[0.40358957047113...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 44|blue-collar|married|  primary|      0|    612|
1|   1| cellular|  3|  apr|    1091|       1|  135|       1| failure|     0|[-
0.5146314267162...|[0.37410844014730...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 62|blue-collar|married|  primary|      0|   1381|
0|   0| cellular| 19|  oct|    1020|       1|   -1|       0| unknown|     0|[-
2.2439300587634...|[0.09587433312548...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 30|blue-collar|married|  primary|      0|    201|
```

```
1|   0| cellular|  5|  mar|     116|       7|  186|      13| success|     0|[-
0.1157458173902...|[0.47109580779837...|       1.0|
|  0.0|(42,[0,11,15,17,1...| 34|blue-collar|married|  primary|      0|    183|
1|   0|  unknown| 19|  may|    3078|       4|   -1|       0| unknown|     0|[-
7.1326817325896...|[7.97937735819626...|       1.0|
|  0.0|(42,[0,11,15,17,1...| 41|blue-collar|married|  primary|      0|    406|
1|   0|  unknown|  9|  may|    2462|       1|   -1|       0| unknown|     0|[-
4.7135176144216...|[0.00889335640130...|       1.0|
|  0.0|(42,[0,11,15,24,2...| 31|blue-collar|married|  primary|      0|   1738|
0|   0|telephone|  5|  feb|     895|       3|   -1|       0| unknown|     0|[-
0.2388225418529...|[0.44057653715319...|       1.0|
|  0.0|(42,[0,12,13,17,1...| 35|blue-collar| single|secondary|      0|   1253|
0|   0|  unknown| 29|  may|    2260|       2|   -1|       0| unknown|     0|[-
5.0347180138819...|[0.00646595240120...|       1.0|
|  0.0|(42,[0,12,15,16,2...| 30|blue-collar| single|  primary|      0|      0|
1|   0| cellular| 20|  nov|    1329|       2|   -1|       0| unknown|     0|[-
1.2129493167824...|[0.22917961939210...|       1.0|
+-----+--------------------+---+-----------+-------+---------+-------+-------+-
------+----+---------+---+-----+--------+--------+-----+--------+--------+-----
-+--------------------+--------------------+----------+
only showing top 20 rows
```

In [56]:
```
predictions.filter((F.col('prediction') == 1.0))\
    .count()
```

Out[56]: 573

In [57]:
```
from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator = BinaryClassificationEvaluator()
print('Test Area Under ROC', evaluator.evaluate(predictions))
```

```
Test Area Under ROC 0.9122801993241221
```

Decision Tree Classifier.

In [61]:
```
from pyspark.ml.classification import DecisionTreeClassifier
tree = DecisionTreeClassifier(featuresCol = 'Subscribed', labelCol = 'label', max
treemodel = tree.fit(train)
predictions = treemodel.transform(test)
predictions.filter((F.col('prediction') == 1.0))\
    .show()
```

```
+-----+--------------------+---+-----------+-------+---------+-------+-------
+-------+----+---------+---+-----+--------+--------+-----+--------+--------+-
-----+------------+--------------------+----------+
|label|          Subscribed|age|        job|marital|education|default|balance
|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|T
arget|rawPrediction|         probability|prediction|
+-----+--------------------+---+-----------+-------+---------+-------+-------
+-------+----+---------+---+-----+--------+--------+-----+--------+--------+-
-----+------------+--------------------+----------+
|  0.0|(42,[0,11,13,16,1...| 36|blue-collar|married|secondary|      0|    195
|      1|   1| cellular|  7|  may|    1297|       2|   -1|       0| unknown|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 27|blue-collar|married|secondary|      0|   1295
|      1|   0| cellular| 14|  may|    1106|       1|   -1|       0| unknown|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 40|blue-collar|married|secondary|      0|    356
|      1|   1| cellular|  7|  may|    1254|       3|  365|       1| failure|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 32|blue-collar|married|secondary|      0|   9714
|      1|   0| cellular| 15|  may|    1237|       2|  361|       2| failure|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 47|blue-collar|married|secondary|      0|   2548
|      1|   0| cellular| 11|  may|     577|       1|  368|       1|   other|
0| [87.0,133.0]|[0.39545454545454...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 48|blue-collar|married|secondary|      0|   1513
|      0|   1| cellular| 17|  jul|    1171|       1|   -1|       0| unknown|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 53|blue-collar|married|secondary|      0|   3079
|      0|   0| cellular|  5|  apr|     759|       1|   -1|       0| unknown|
0|[177.0,241.0]|[0.42344497607655...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 24|blue-collar|married|secondary|      0|   -220
|      1|   0| cellular| 17|  apr|     401|       1|   15|       1| failure|
0|  [67.0,68.0]|[0.49629629629629...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 31|blue-collar|married|secondary|      0|   1716
|      1|   0| cellular| 13|  apr|     542|       2|  340|       2|   other|
0| [87.0,133.0]|[0.39545454545454...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 53|blue-collar|married|secondary|      0|    -76
|      0|   0| cellular| 29|  jan|     173|       1|  164|       4|   other|
0|[280.0,506.0]|[0.35623409669211...|       1.0|
|  0.0|(42,[0,11,13,17,1...| 38|blue-collar|married|secondary|      0|    376
|      1|   0|  unknown|  7|  may|    1521|       1|   -1|       0| unknown|
0|[152.0,164.0]|[0.48101265822784...|       1.0|
|  0.0|(42,[0,11,13,17,1...| 38|blue-collar|married|secondary|      0|    384
|      1|   0|  unknown|  6|  may|    1906|       3|   -1|       0| unknown|
0|[152.0,164.0]|[0.48101265822784...|       1.0|
|  0.0|(42,[0,11,13,17,1...| 51|blue-collar|married|secondary|      0|    701
|      1|   0|  unknown| 15|  may|    1051|       3|   -1|       0| unknown|
0|[152.0,164.0]|[0.48101265822784...|       1.0|
|  0.0|(42,[0,11,13,17,2...| 59|blue-collar|married|secondary|      0|    448
```

```
|      0|   0|  unknown|  8|  jun|     198|       1|  651|       7|   other|
0|[280.0,506.0]|[0.35623409669211...|       1.0|
|  0.0|(42,[0,11,13,18,2...| 32|blue-collar|married|secondary|      0|    485
|      1|   0|telephone| 12|  may|    1100|       2|   -1|       0| unknown|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,15,16,1...| 56|blue-collar|married|  primary|      0|   -116
|      1|   0| cellular|  8|  may|    1014|       2|   -1|       0| unknown|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,15,16,1...| 39|blue-collar|married|  primary|      0|    481
|      0|   0| cellular| 13|  may|     197|       3|  357|       1|   other|
0|[280.0,506.0]|[0.35623409669211...|       1.0|
|  0.0|(42,[0,11,15,16,1...| 57|blue-collar|married|  primary|      0|   3498
|      0|   0| cellular| 15|  jul|     210|       2|  456|       2| success|
0|[280.0,506.0]|[0.35623409669211...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 45|blue-collar|married|  primary|      0|    292
|      1|   0| cellular| 29|  aug|     961|       4|   -1|       0| unknown|
0|[303.0,495.0]|[0.37969924812030...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 37|blue-collar|married|  primary|      0|      0
|      0|   0| cellular| 18|  nov|     850|       2|   -1|       0| unknown|
0|[177.0,241.0]|[0.42344497607655...|       1.0|
+-----+--------------------+---+-----------+-------+---------+-------+-------
+-------+----+---------+---+-----+--------+--------+-----+--------+--------+-
-----+-------------+--------------------+----------+
only showing top 20 rows
```

In [62]:
```
predictions.filter((F.col('prediction') == 1.0))\
    .count()
```

Out[62]: 714

In [63]:
```
evaluator = BinaryClassificationEvaluator()
print('Test Area Under ROC', evaluator.evaluate(predictions))
```

```
Test Area Under ROC 0.2649764141294298
```

Random Forest Classifier

In [64]:
```
from pyspark.ml.classification import RandomForestClassifier
rf = RandomForestClassifier(featuresCol = 'Subscribed', labelCol = 'label')
rfmodel = rf.fit(train)
predictions = rfmodel.transform(test)
predictions.filter((F.col('prediction') == 1.0))\
    .show()
```

```
+-----+--------------------+---+-----------+--------+---------+-------+-----
--+-------+----+---------+---+-----+--------+--------+-----+--------+--------
+------+--------------------+--------------------+----------+
|label|          Subscribed|age|        job| marital|education|default|balan
ce|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome
|Target|       rawPrediction|         probability|prediction|
+-----+--------------------+---+-----------+--------+---------+-------+-----
--+-------+----+---------+---+-----+--------+--------+-----+--------+--------
+------+--------------------+--------------------+----------+
|  0.0|(42,[5,11,14,16,1...| 67|    retired| married| tertiary|      0|   19
48|      0|   0| cellular|  2|  jul|     844|       2|  155|       5| success
|     0|[8.6595567233258,...|[0.43297783616629...|       1.0|
|  0.0|(42,[7,11,14,16,2...| 71|entrepreneur| married| tertiary|      0|  152
65|      0|   0| cellular| 25|  feb|     865|       1|  192|       2| failure
|     0|[9.67870721080854...|[0.48393536054042...|       1.0|
|  1.0|(42,[1,11,14,16,1...| 36|  management| married| tertiary|      0|    2
55|      0|   0| cellular| 22|  may|     970|       2|   92|       2|   other
|     1|[9.86294068709076...|[0.49314703435453...|       1.0|
|  1.0|(42,[1,11,14,16,2...| 30|  management| married| tertiary|      0|   19
96|      0|   0| cellular|  8|  feb|    1133|       3|  101|       1|   other
|     1|[9.83811379987605...|[0.49190568999380...|       1.0|
|  1.0|(42,[1,12,14,16,2...| 33|  management|  single| tertiary|      0|   68
07|      0|   0| cellular| 21|  oct|     512|       2|  184|       1| success
|     1|[9.44886100409145...|[0.47244305020457...|       1.0|
|  1.0|(42,[1,14,16,20,3...| 55|  management|divorced| tertiary|      0|   23
83|      0|   0| cellular|  5|  aug|    1019|       2|   63|       2| success
|     1|[8.88789081708229...|[0.44439454085411...|       1.0|
|  1.0|(42,[1,14,16,21,3...| 57|  management|divorced| tertiary|      0|   32
87|      0|   0| cellular| 22|  jun|     867|       1|   84|       3| success
|     1|[9.44534534030586...|[0.47226726701529...|       1.0|
|  1.0|(42,[2,12,14,16,2...| 28|  technician|  single| tertiary|      0|   49
87|      0|   0| cellular|  2|  jun|     924|       2|  113|      21| success
|     1|[9.64305447254550...|[0.48215272362727...|       1.0|
|  1.0|(42,[5,11,13,16,1...| 69|    retired| married|secondary|      0|
0|      0|   0| cellular| 27|  jul|     666|       1|   90|       4| success|
1|[9.47104926112176...|[0.47355246305608...|       1.0|
|  1.0|(42,[5,11,13,19,3...| 77|    retired| married|secondary|      0|   41
12|      0|   0|telephone| 29|  jul|     426|       1|  184|       3| success
|     1|[9.55208673812343...|[0.47760433690617...|       1.0|
|  1.0|(42,[5,11,13,25,3...| 77|    retired| married|secondary|      0|   41
12|      0|   0|telephone| 26|  jan|    1616|       1|   95|       2| success
|     1|[8.77991572402009...|[0.43899578620100...|       1.0|
|  1.0|(42,[5,11,14,27,3...| 84|    retired| married| tertiary|      0|   47
61|      0|   0|telephone|  9|  sep|    1405|       1|   92|       3| failure
|     1|[9.87189622303920...|[0.49359481115196...|       1.0|
|  1.0|(42,[5,11,15,16,1...| 70|    retired| married|  primary|      0|   27
95|      0|   0| cellular|  8|  jul|     480|       1|  181|       2| success
|     1|[9.99874660271292...|[0.49993733013564...|       1.0|
|  1.0|(42,[5,15,19,32,3...| 70|    retired|divorced|  primary|      0|    4
```

```
82|      0|   0|telephone| 14|  jul|     413|       1|  181|       3| success
|     1|[9.58677730662026...|[0.47933886533101...|       1.0|
+-----+--------------------+---+-----------+--------+---------+-------+-----
--+-------+----+---------+---+-----+--------+--------+-----+--------+--------
+------+--------------------+--------------------+----------+
```

In [65]:
```
evaluator = BinaryClassificationEvaluator()
print("Test Area Under ROC: " + str(evaluator.evaluate(predictions)))
```

```
Test Area Under ROC: 0.8902834259312294
```

Gradient-Boosted Tree Classifier

In [66]:
```
from pyspark.ml.classification import GBTClassifier
gbc = GBTClassifier(featuresCol = 'Subscribed', labelCol = 'label', maxIter=10)
gbcmodel = gbc.fit(train)
predictions = gbcmodel.transform(test)
predictions.filter((F.col('prediction') == 1.0))\
    .show()
```

```
+-----+--------------------+---+----------+-------+---------+-------+-------
+-------+----+---------+---+-----+--------+--------+-----+--------+--------+-
-----+--------------------+--------------------+----------+
|label|          Subscribed|age|       job|marital|education|default|balance
|housing|loan|  contact|day|month|duration|campaign|pdays|previous|poutcome|T
arget|       rawPrediction|         probability|prediction|
+-----+--------------------+---+----------+-------+---------+-------+-------
+-------+----+---------+---+-----+--------+--------+-----+--------+--------+-
-----+--------------------+--------------------+----------+
|  0.0|(42,[0,11,13,16,1...| 36|blue-collar|married|secondary|      0|    195
|      1|   1| cellular|  7|  may|    1297|       2|   -1|       0| unknown|
0|[-0.1645754548094...|[0.41844723425828...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 27|blue-collar|married|secondary|      0|   1295
|      1|   0| cellular| 14|  may|    1106|       1|   -1|       0| unknown|
0|[-0.2327567344727...|[0.38567869355876...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 40|blue-collar|married|secondary|      0|    356
|      1|   1| cellular|  7|  may|    1254|       3|  365|       1| failure|
0|[-0.2294437827938...|[0.38724975737922...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 32|blue-collar|married|secondary|      0|   9714
|      1|   0| cellular| 15|  may|    1237|       2|  361|       2| failure|
0|[-0.2544698531545...|[0.37544211599718...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 47|blue-collar|married|secondary|      0|   2548
|      1|   0| cellular| 11|  may|     577|       1|  368|       1|   other|
0|[-0.0702459667219...|[0.46493467424232...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 48|blue-collar|married|secondary|      0|   1513
|      0|   1| cellular| 17|  jul|    1171|       1|   -1|       0| unknown|
0|[-0.2270166207181...|[0.38840225578173...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 53|blue-collar|married|secondary|      0|   3079
|      0|   0| cellular|  5|  apr|     759|       1|   -1|       0| unknown|
0|[-0.3785617430038...|[0.31927111406327...|       1.0|
|  0.0|(42,[0,11,13,16,2...| 31|blue-collar|married|secondary|      0|   1716
|      1|   0| cellular| 13|  apr|     542|       2|  340|       2|   other|
0|[-0.0367589668171...|[0.48162879037146...|       1.0|
|  0.0|(42,[0,11,13,21,2...| 38|blue-collar|married|secondary|      0|   1391
|      1|   0|telephone|  5|  jun|     614|       2|   -1|       0| unknown|
0|[-0.1230821302411...|[0.43876782940022...|       1.0|
|  0.0|(42,[0,11,15,16,1...| 57|blue-collar|married|  primary|      0|   3498
|      0|   0| cellular| 15|  jul|     210|       2|  456|       2| success|
0|[-0.1244864030609...|[0.43807634266336...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 45|blue-collar|married|  primary|      0|    292
|      1|   0| cellular| 29|  aug|     961|       4|   -1|       0| unknown|
0|[-0.1896015251702...|[0.40631912600415...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 37|blue-collar|married|  primary|      0|      0
|      0|   0| cellular| 18|  nov|     850|       2|   -1|       0| unknown|
0|[-0.1126367158274...|[0.44391861048300...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 39|blue-collar|married|  primary|      0|    394
|      1|   0| cellular| 17|  apr|     981|       2|   -1|       0| unknown|
0|[-0.1896015251702...|[0.40631912600415...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 44|blue-collar|married|  primary|      0|    612
```

```
|      1|   1| cellular|  3|  apr|    1091|       1|  135|       1| failure|
0|[-0.2239080733378...|[0.38988012202606...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 34|blue-collar|married|  primary|      0|   6718
|      0|   0| cellular| 13|  jan|     278|       4|   97|       1|   other|
0|[-0.1160690723496...|[0.44222468136904...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 62|blue-collar|married|  primary|      0|   1381
|      0|   0| cellular| 19|  oct|    1020|       1|   -1|       0| unknown|
0|[-0.0795078580509...|[0.46032962784636...|       1.0|
|  0.0|(42,[0,11,15,16,2...| 37|blue-collar|married|  primary|      0|    623
|      0|   0| cellular| 26|  mar|     267|       2|   -1|       0| unknown|
0|[-0.4606479599696...|[0.28469391528080...|       1.0|
|  0.0|(42,[0,11,15,24,2...| 31|blue-collar|married|  primary|      0|   1738
|      0|   0|telephone|  5|  feb|     895|       3|   -1|       0| unknown|
0|[-0.1316227406187...|[0.43456606504850...|       1.0|
|  0.0|(42,[0,12,13,16,2...| 33|blue-collar| single|secondary|      0|   3975
|      1|   1| cellular| 17|  apr|     515|       1|  150|       1|   other|
0|[-0.0404126486701...|[0.47980466868643...|       1.0|
|  0.0|(42,[0,12,13,16,2...| 34|blue-collar| single|secondary|      0|   -317
|      1|   0| cellular| 17|  apr|     518|       3|  346|       1|   other|
0|[-0.0563698860349...|[0.47184487224104...|       1.0|
+-----+--------------------+---+-----------+-------+---------+-------+-------
+-------+----+---------+---+-----+--------+--------+-----+--------+--------+-
-----+--------------------+--------------------+----------+
only showing top 20 rows
```

In [67]:
```
predictions.filter((F.col('prediction') == 1.0))\
    .count()
```

Out[67]: 637

In [68]:
```
evaluator = BinaryClassificationEvaluator()
print("Test Area Under ROC: " + str(evaluator.evaluate(predictions)))
```

```
Test Area Under ROC: 0.9126390900404298
```

Modifying this model with the ParamGridBuilder as well as the CrossValidator because the Logistic Regression algorithm produced the best results.

In [69]:
```
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
paramGrid = (ParamGridBuilder().addGrid(lr.maxIter, [500]) \
                               .addGrid(lr.regParam, [0]) \
                               .addGrid(lr.elasticNetParam, [1]) \
                               .build())
cv = CrossValidator(estimator=lr, estimatorParamMaps=paramGrid, evaluator=evaluat
cvModel = cv.fit(train)
predictions = cvModel.transform(test)
evaluator.evaluate(predictions)
```

Out[69]: 0.912823308485996

In [70]:
```
predictions.show()
```

```
+-----+--------------------+---+----------+-------+---------+-------+-------+-
------+----+--------+---+-----+--------+--------+-----+--------+--------+------
+--------------------+--------------------+----------+
|label|          Subscribed|age|       job|marital|education|default|balance|h
ousing|loan| contact|day|month|duration|campaign|pdays|previous|poutcome|Target
|       rawPrediction|         probability|prediction|
+-----+--------------------+---+----------+-------+---------+-------+-------+-
------+----+--------+---+-----+--------+--------+-----+--------+--------+------
+--------------------+--------------------+----------+
|  0.0|(42,[0,11,13,16,1...| 48|blue-collar|married|secondary|      1|      0|
1|   1|cellular|  7|  may|     275|       1|   -1|       0| unknown|     0|[3.1
3785523542612...|[0.95842750819995...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 30|blue-collar|married|secondary|      0|   -109|
1|   1|cellular|  7|  may|     160|       4|   -1|       0| unknown|     0|[4.0
3860822226261...|[0.98268317551867...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 31|blue-collar|married|secondary|      0|    154|
1|   1|cellular| 15|  may|     454|       1|   -1|       0| unknown|     0|[2.4
6808750009277...|[0.92187413347783...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 31|blue-collar|married|secondary|      0|    864|
1|   1|cellular| 13|  may|     127|       1|   -1|       0| unknown|     0|[3.8
5560928097114...|[0.97927779003858...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 36|blue-collar|married|secondary|      0|    195|
1|   1|cellular|  7|  may|    1297|       2|   -1|       0| unknown|     0|[-0.
9361501387853...|[0.28167865215444...|       1.0|
|  0.0|(42,[0,11,13,16,1...| 36|blue-collar|married|secondary|      0|    418|
1|   1|cellular| 13|  may|     132|       2|   -1|       0| unknown|     0|[3.9
2543458983428...|[0.98064831854426...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 37|blue-collar|married|secondary|      0|   1114|
1|   1|cellular|  6|  may|     133|       1|   -1|       0| unknown|     0|[3.8
8041012427981...|[0.97977513145376...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 37|blue-collar|married|secondary|      0|   3269|
1|   1|cellular| 14|  may|     121|       2|   -1|       0| unknown|     0|[3.9
3828577570745...|[0.98089069750357...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 38|blue-collar|married|secondary|      0|   -367|
1|   1|cellular| 12|  may|     354|       1|   -1|       0| unknown|     0|[2.9
1499303224934...|[0.94858263893948...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 40|blue-collar|married|secondary|      0|    503|
1|   1|cellular| 18|  may|      24|       2|   -1|       0| unknown|     0|[4.3
3946090229122...|[0.98712438576124...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 44|blue-collar|married|secondary|      0|   -182|
1|   1|cellular| 14|  may|      10|      10|   -1|       0| unknown|     0|[5.1
4445937612811...|[0.99420218446484...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 44|blue-collar|married|secondary|      0|    130|
1|   1|cellular| 11|  may|     528|       1|   -1|       0| unknown|     0|[2.1
8276558755207...|[0.89869114382684...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 25|blue-collar|married|secondary|      0|    148|
1|   0|cellular| 18|  may|     119|       1|   -1|       0| unknown|     0|[3.4
1658145327322...|[0.96821874702659...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 26|blue-collar|married|secondary|      0|    160|
1|   0|cellular| 18|  may|     252|       4|   -1|       0| unknown|     0|[3.1
2186157284621...|[0.95778556015664...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 27|blue-collar|married|secondary|      0|   1295|
1|   0|cellular| 14|  may|    1106|       1|   -1|       0| unknown|     0|[-0.
7232591158116...|[0.32667570754645...|       1.0|
```

```
|  0.0|(42,[0,11,13,16,1...| 28|blue-collar|married|secondary|      0|   -470|
1|   0|cellular|  7|  may|     275|       2|   -1|       0| unknown|     0|[2.9
3791428325493...|[0.94968916540137...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 28|blue-collar|married|secondary|      0|      4|
1|   0|cellular| 15|  may|     310|       1|   -1|       0| unknown|     0|[2.6
3508696917240...|[0.93308586323893...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 28|blue-collar|married|secondary|      0|     83|
1|   0|cellular| 15|  may|     224|       1|   -1|       0| unknown|     0|[2.9
9686181135929...|[0.95243215241756...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 28|blue-collar|married|secondary|      0|   1325|
1|   0|cellular| 15|  may|     242|       1|   -1|       0| unknown|     0|[2.9
0994929218307...|[0.94833608017311...|       0.0|
|  0.0|(42,[0,11,13,16,1...| 29|blue-collar|married|secondary|      0|     25|
1|   0|cellular| 15|  may|      50|       5|   -1|       0| unknown|     0|[4.0
8550146747896...|[0.98346335365917...|       0.0|
+-----+--------------------+---+-----------+-------+---------+-------+-------+-
------+----+--------+---+-----+--------+--------+-----+--------+--------+------
+--------------------+--------------------+----------+
only showing top 20 rows
```

In [71]:
```
predictions.count()
```

Out[71]:
```
9150
```

In [ ]:
```

```