SPARK MINI PROJECT

BAN 5753

GitHub:-

https://github.com/saswataraautray/pyspark-deposit-opening-classification

NOREEN CHIHORA

TEJASWI MARUTHI

SASWATA RAUTRAY

PRATHAMESH KULKARNI

BUSINESS PROBLEM

In this project, the goal is to identify clients who will subscribe for a term deposit.

DATA EXPLORATION

Data Summary

Step 1:- Loading Data

We load the data using the PySpark function and into the spark schema

Data: The given dataset had 41,188 rows and 21 columns

From this dataset, 10 columns are categorical, 10 are numeric and 1 is the target variable. Below are the different categories in the categorical columns.

job	marital	education	default	housing
management|	unknown	high.school	unknown	unknown
self-employed	divorced	unknown	yes	yes
retired	married	basic.6y	no	no
unknown	single	professional.course		
student		university.degree		
blue-collar		illiterate		
entrepreneur		basic.4y		
admin		basic.9y		
technician				
services				
housemaid				
unemployed				

loan	contact	month	day_of_week	poutcome
unknown	cellular	mar	monday	success
yes	telephone	apr	tuesday	failure
no		may	wednesday	nonexistent
		jun	thursday	
		jul	friday	
		aug		
		sep		
		oct		
		nov		
		dec		

We further try to check the schema of the dataframe

```
root
 |-- age: integer (nullable = true)
 |-- job: string (nullable = true)
 |-- marital: string (nullable = true)
 |-- education: string (nullable = true)
 |-- 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)
 |-- emp_var_rate: double (nullable = true)
 |-- cons_price_idx: double (nullable = true)
 |-- cons_conf_idx: double (nullable = true)
 |-- euribor3m: double (nullable = true)
 |-- nr_employed: double (nullable = true)
 |-- y: string (nullable = true)
```

The target variable has two possible outcomes, yes and no.

Null Values

age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp_var_rate	cons_price_idx	cons_conf_idx	euribor3m	nr_employed	y
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The dataset does not have null values, but some categorical features do contain some of 'unknown' values. These unknown values can be considered as the feature in the classification model we are using. These missing values can be treated as a possible class label or using deletion or imputation techniques.

Step 2:- EDA

Numeric Features Description

	0	1	2	3	4	5	6	7
summary	count	mean	stddev	min	25%	50%	75%	max
age	41188	40.02406040594348	10.421249980934043	17	32	38	47	98
duration	41188	258.2850101971448	259.27924883646455	0	102	180	319	4918
campaign	41188	2.567592502670681	2.770013542902331	1	1	2	3	56
pdays	41188	962.4754540157328	186.910907344741	0	999	999	999	999
previous	41188	0.17296299893172767	0.49490107983928927	0	0	0	0	7
empl_variation_rate	41188	0.08188550063178966	1.57095974051703	-3.4	-1.8	1.1	1.4	1.4
consumer_price_idx	41188	93.5756643682899	0.5788400489540823	92.201	93.075	93.749	93.994	94.767
consumer_confidence_idx	41188	-40.502600271918276	4.628197856174573	-50.8	-42.7	-41.8	-36.4	-26.9
euribor3m	41188	3.621290812858533	1.7344474048512595	0.634	1.344	4.857	4.961	5.045
num_employees	41188	5167.035910943957	72.25152766826338	4963.6	5099.1	5191.0	5228.1	5228.1

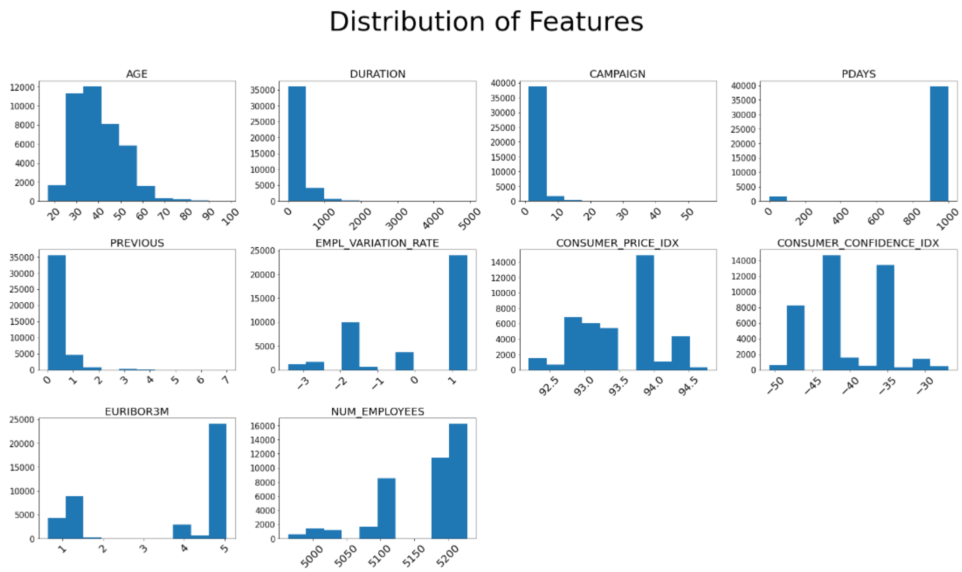
There is an equal number of records for each column in the dataframe because we do not have any null values.

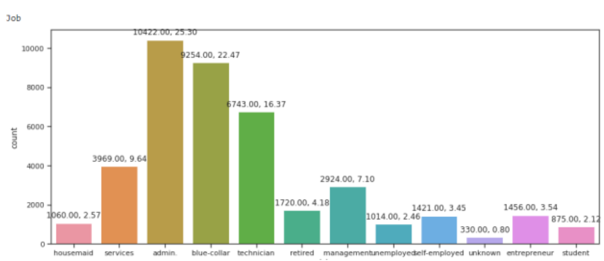
Correlation

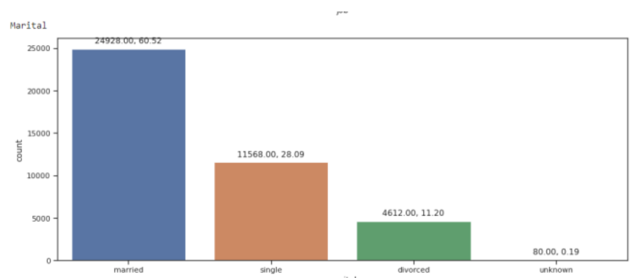
	age	duration	campaign	pdays	previous	emp_var_rate	cons_price_idx	cons_conf_idx	euribor3m	nr_employed
age	1.00	-0.00	0.00	-0.03	0.02	-0.00	0.00	0.13	0.01	-0.02
duration	-0.00	1.00	-0.07	-0.05	0.02	-0.03	0.01	-0.01	-0.03	-0.04
campaign	0.00	-0.07	1.00	0.05	-0.08	0.15	0.13	-0.01	0.14	0.14
pdays	-0.03	-0.05	0.05	1.00	-0.59	0.27	0.08	-0.09	0.30	0.37
previous	0.02	0.02	-0.08	-0.59	1.00	-0.42	-0.20	-0.05	-0.45	-0.50
emp_var_rate	-0.00	-0.03	0.15	0.27	-0.42	1.00	0.78	0.20	0.97	0.91
cons_price_idx	0.00	0.01	0.13	0.08	-0.20	0.78	1.00	0.06	0.69	0.52
cons_conf_idx	0.13	-0.01	-0.01	-0.09	-0.05	0.20	0.06	1.00	0.28	0.10
euribor3m	0.01	-0.03	0.14	0.30	-0.45	0.97	0.69	0.28	1.00	0.95
nr_employed	-0.02	-0.04	0.14	0.37	-0.50	0.91	0.52	0.10	0.95	1.00

As we can see that all variables have collinearity less than 0.85 but there is a high correlation between employment variation rate and euribor 3 month rate, number of employees and euribor 3 month rate and number of employees and employment variation rate.

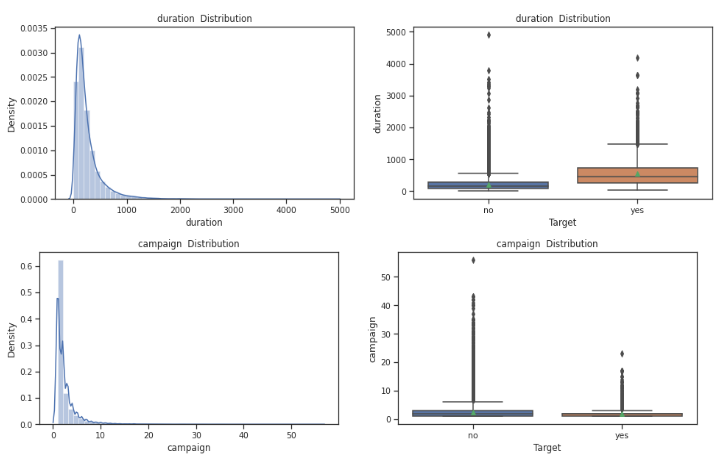
Distribution of Features

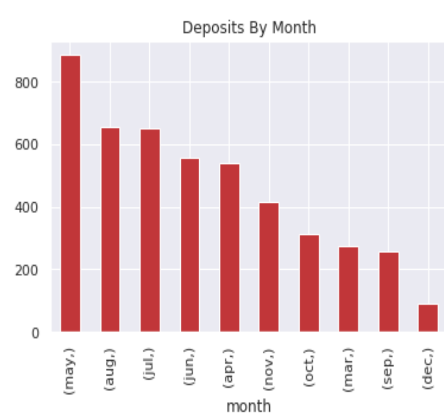






We further try to check the outliers using the distribution graph and the box plots.





Step 3:- Making the pipeline (Feature Engineering)

We try to make the ML pipeline that can be used to run model.

First, we convert the categorical feature to nominal features using one hot encoding. We convert the variables such as :-

```
# Selecting categorical columns only
categoricalColumns = ['job', 'default', 'housing', 'loan', 'marital', 'education', 'contact', 'month', 'day_of_week', 'poutcome']
```

Then we try to do scaling of the data using the numerical and one hot encoded data. Scaling will help the model to give equal importance to the variables and help from preventing more importance to anyone variable :-

```
# Vectorizing to create a new features column with indexed and encoded values
assembler = VectorAssembler(inputCols=assemblerInputs, outputCol="vectorized_features", handleInvalid="skip")
stages += [assembler]

# Standard Scaling
scaler = StandardScaler(inputCol="vectorized_features", outputCol="features")
stages += [scaler]
```

We then check the data :-

label	features	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp_var_rate
0.0	(53,[0,1,2,3,5,6,...	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	261	1	999	0	nonexistent	1.1
0.0	(53,[0,1,2,3,5,6,...	57	services	married	high.school	unknown	no	no	telephone	may	mon	149	1	999	0	nonexistent	1.1
0.0	(53,[0,1,2,3,5,6,...	37	services	married	high.school	no	yes	no	telephone	may	mon	226	1	999	0	nonexistent	1.1
0.0	(53,[0,1,2,3,5,6,...	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	151	1	999	0	nonexistent	1.1
0.0	(53,[0,1,2,3,5,6,...	56	services	married	high.school	no	no	yes	telephone	may	mon	307	1	999	0	nonexistent	1.1

Step 4:- Doing the supervised learning

We further want to check the class distribution as imbalances target variable may cause a lot of issue to the classification model. If we have imbalance, then we try to correct the imbalance using the appropriate weights.

```
{1: 4.438362068965517, 0: 0.5634781656999015}
```

We assign the above weights as an additional feature to the dataset before training the will correct the imbalance issue of the dataset. We then further try to split the dataset.

```
Training Dataset Count: 28975
Test Dataset Count: 12213
```

1)Running the Machine learning models:-

1) Logistic Regression Model:-
   We try to run a logistic regression model and try to get the variable importance for the predicted logits.

features	label	rawPrediction	prediction	probability
(53,[0,1,2,3,4,5,...	0.0	[2.59094136041744...	0.0	[0.93027630066708...
(53,[0,1,2,3,4,5,...	0.0	[1.07830280903530...	0.0	[0.74617267136112...
(53,[0,1,2,3,4,5,...	0.0	[2.22867246999743...	0.0	[0.90279492223633...
(53,[0,1,2,3,4,5,...	0.0	[0.41684314607445...	0.0	[0.60272759621648...
(53,[0,1,2,3,4,5,...	0.0	[-0.0277122401558...	1.0	[0.49307238330450...

only showing top 5 rows

	Features	Weights
1	duration	1.848358
6	cons_price_idx	1.324406
8	euribor3m	1.243814
40	month_encoded_aug	0.397314
46	month_encoded_mar	0.239781
37	contact_encoded_cellular	0.218799
9	nr_employed	0.148130
21	default_encoded_no	0.087047
49	day_of_week_encoded_wed	0.053425
44	month_encoded_oct	0.043443

	Features	Weights
5	emp_var_rate	-3.539902
31	education_encoded_high.school	-0.547826
32	education_encoded_basic.9y	-0.461804
30	education_encoded_university.degree	-0.460457
34	education_encoded_basic.4y	-0.444564
38	month_encoded_may	-0.403155
33	education_encoded_professional.course	-0.402110
52	poutcome_encoded_failure	-0.337632
41	month_encoded_jun	-0.310711
35	education_encoded_basic.6y	-0.275750

2) Linear_SVC Model:-

We try to run a Linear _SVC model and try to get the variable importance for the predicted logits.

label	rawPrediction	prediction
0.0	[1.62528898825679...	0.0
0.0	[0.59511994817338...	0.0
0.0	[1.38879846661605...	0.0
0.0	[0.17714548866918...	0.0
0.0	[-0.1716408307432...	1.0

only showing top 5 rows

	Features	Weights
1	duration	1.239538
8	euribor3m	0.579999
6	cons_price_idx	0.425688
40	month_encoded_aug	0.175282
46	month_encoded_mar	0.141259
37	contact_encoded_cellular	0.080372
50	day_of_week_encoded_tue	0.053066
49	day_of_week_encoded_wed	0.041348
25	loan_encoded_no	0.039707
21	default_encoded_no	0.039575

3) Decision Tree Model:-

We try to run a Decision Tree model and try to get the variable importance for the predicted logits.

label	rawPrediction	prediction	probability
0.0	[7715.14304476370...	0.0	[0.98694136055648...
0.0	[673.356408011384...	0.0	[0.63821891709968...
0.0	[7715.14304476370...	0.0	[0.98694136055648...
0.0	[673.356408011384...	0.0	[0.63821891709968...
0.0	[673.356408011384...	0.0	[0.63821891709968...

only showing top 5 rows

	Feature	Importance
0	duration	0.573994
7	nr_employed	0.337176
5	cons_conf_idx	0.045359
6	euribor3m	0.018390
8	month_encoded_oct	0.012342
2	pdays	0.011379
4	cons_price_idx	0.001121
3	previous	0.000120
1	campaign	0.000120

4) Random Forest Model:-

We try to run a Random Forest ecision Tree model and try to get the variable importance for the predicted logits.

label	rawPrediction	prediction	probability
0.0	[13.5083060160837...	0.0	[0.67541530080418...
0.0	[10.7677719907926...	0.0	[0.53838859953963...
0.0	[13.0334040341431...	0.0	[0.65167020170715...
0.0	[10.8661628562083...	0.0	[0.54330814281041...
0.0	[8.58946213224159...	1.0	[0.42947310661207...

only showing top 5 rows

	Feature	Importance
1	duration	0.280093
9	nr_employed	0.220024
8	euribor3m	0.131527
5	emp_var_rate	0.116503
6	cons_price_idx	0.039885
3	pdays	0.036674
43	poutcome_encoded_nonexistent	0.034186
31	contact_encoded_cellular	0.030339
32	month_encoded_may	0.026836
7	cons_conf_idx	0.019318

5) Gradient Boosting Model:-

We try to run a Random Forest ecision Tree model and try to get the variable importance for the predicted logits.

```
+-----+--------------------+----------+--------------------+
|label|       rawPrediction|prediction|         probability|
+-----+--------------------+----------+--------------------+
|  0.0|[1.54150600027751...|       0.0|[0.95618654239317...|
|  0.0|[0.67889104194661...|       0.0|[0.79539899177199...|
|  0.0|[1.54150600027751...|       0.0|[0.95618654239317...|
|  0.0|[0.52878070916151...|       0.0|[0.74222425344585...|
|  0.0|[-0.4853872888681...|       1.0|[0.27472613936155...|
+-----+--------------------+----------+--------------------+
only showing top 5 rows
```

	Feature	Importance
1	duration	0.418595
9	nr_employed	0.163776
8	euribor3m	0.104315
7	cons_conf_idx	0.040156
5	emp_var_rate	0.034292
0	age	0.031693
37	month_encoded_oct	0.029717
6	cons_price_idx	0.019217
2	campaign	0.018320
3	pdays	0.017716

6) We further try to check performance metric of the models and try to check which models performs better. As we can see that Gradient Boosting Algorithm. As we can see that F-1 Score is 0.58, Recall is 0.914 and precision is 0.431.

model	accuracy	precision	recall	f1score	auc
LogisticRegression_10146ae4e77f	0.859658	0.429043	0.881688	0.577208	0.936069
LinearSVC_5830fbc0e036	0.845411	0.405587	0.908063	0.560726	0.935425
DecisionTreeClassifier_e2933ff7b468	0.851142	0.415432	0.908817	0.570213	0.878508
RandomForestClassifier_affe14a68409	0.836568	0.381003	0.807084	0.517641	0.912642
GBTClassifier_90ba6940d533	0.859903	0.431673	0.914092	0.586415	0.944473

7) Champion model:-

As the Gradient Boosting algorithm is the best and try to use cross validation method in grid search methods.

```
{'model': 'GBTClassifier_0000cc5bae4d',
 'accuracy': 0.8499140260378285,
 'precision': 0.41464237516869096,
 'recall': 0.9261492087415222,
 'f1score': 0.5728268468888371,
 'auc': 0.9425893008324536}
```

Step 5:- Check by model performance

	Feature	Importance
1	duration	0.449452
9	nr_employed	0.222170
8	euribor3m	0.078273
7	cons_conf_idx	0.037278
25	month_encoded_oct	0.035836
0	age	0.026460
5	emp_var_rate	0.020804
6	cons_price_idx	0.017382
11	job_encoded_blue-collar	0.014950
2	campaign	0.013952

We can use the above table to check the variable importance of the best model. As we can see it is clear The feature "duration" is the most important variable in the customer conversion rate, followed by the features "nr_employed", "euribor3m" in the decreasing order. We can see that the feature that is least important for the customer coversion rate would be type of "campaign".

Conclusion & Result:-

The main objective of this project is to increase the effectiveness of the bank's telemarketing campaign, which was successfully met through data analysis, visualization and analytical model building. A target customer profile was established while classification and regression models were built to predict customers' response to the term deposit campaign.

By applying Gradient Boosting Algorithm, classification and estimation model were successfully built. With these two models, the bank will be able to predict a customer's response to its telemarketing campaign before calling this customer. In this way, the bank can allocate more marketing efforts to the clients who are classified as highly likely to convert and not.

Recommendation:-

1) **Need to Frequently contact the customer**
   We can see that most important feature is the duration between the conatct made to customer. Therefore if the bank can reduce the duration of the contact by more targeted campaign, we can seee more conversion happening.

2) **Need to regulaet the number of employees at Bank**
   As number of employee is one of the most crucial factor in conversion of customer. We can say from the box plot that if we keep the number of employee between the 5000 to 5100 the customer conversion rate is moslty yes while increasing the employee beyond the 5100 mark can lead to less conversion rate.

3) **Need to control euribor 3 rate**
   We can say that as we control euribor 3 rate there would be significant change in customer conversion rate for bank. Again takin inference from the box plot we can say that if the the euribor 3 rate is kept between 0.5 to 1.3 then there would be maximum conversion rate of customer rather than increasing it above 1.5.