**READ ME File - Nano Degree Program-Data Scientist- Project 4-Capstone Project**

Contents

[1. Installation 2](#_Toc23761404)

[2. Project Motivation 2](#_Toc23761405)

[3. File Descriptions 3](#_Toc23761406)

[4. Instructions: 3](#_Toc23761407)

[5. RESULTS 3](#_Toc23761408)

[6. Licensing, Authors, Acknowledgements 6](#_Toc23761409)

# Installation

1. Install **mixtend** using one of the following approaches:

* Conda: conda install mixtend
* from PyPI- pip install mixtend
* **Dev version: pip install git+git://github.com/rasbt/mlxtend.git** – I have used this method

# Project Motivation

Data Analytics is a fast-growing field in competitive business world.  Businesses are aiming to transform Big Data into actionable intelligence by leveraging AI (Artificial Intelligence) and ML (Machine Learning).

The data driven decision making is becoming an integral part of Company’s core strategy to derive the customer insights for making better business decisions.  Businesses are using the ML models to develop the prediction models in areas such as Operational improvement, i.e. Customer Retention, Fraud Prevention, Price Modeling etc.

The goal of this capstone project is to design and deploy a Supervised ML “Binary Classification” model, such as Logistic regression to predict if a Bank customer will subscribe(yes/no) a term deposit campaign (Response variable). This will help to execute some prescriptive measures (i.e. revised campaign strategy to offer better interest rate/promo etc.) to increase the term deposit acceptance rate.

I have used the data is which is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe (yes/no) a term deposit (Response variable).

This dataset is sourced from the **IBM Watson Platform**- UCI Machine Learning Repository. This data set contains 10% of the examples and 17 inputs, randomly selected from the full data set, bank-full.csv. The full data set is available on the Watson Studio Community as well as at <https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>.

The classification goal is to predict if the client will subscribe (yes/no) a term deposit (Response variable).

# File Descriptions

The project work is housed across three folders. All the data /scripts/models can be accessed in GitHub using following URL;

GitHub URL: <https://github.com/paridad/Udacity-Nano-DS-Capstone>

* + data :bank\_marketing\_data.csv: Banks’ campaign Data set
  + models :DS-Capstone-Project.ipynb: Jupyter notebook to create, improve and train a model and test the model

# Instructions:

Pls. make sure to load the data to appropriate directory

* 1. Since I have used the Local Jupyter Notebook, I have uploaded the data into same folder as notebook.
  2. But you may need to modify the code, if you need to store data in a different place

# read in the csv file

bank\_data = 'bank\_marketing\_data.csv'

# print out some data

bank\_df = pd.read\_csv(bank\_data)

print('Data shape (rows, cols): ', bank\_df.shape)

print()

bank\_df.head()

* 1. Run the Notebook and analyze the result.

# ANALYSIS

**Data Exploration:**

Started with 45,211 Data points and 17 features/columns

Data shape (rows, cols): (45211, 17)

**Key Predictor/feature variables**, but not limited to:

Continuous/Numeric Variables:

* Age
* Balance
* Campaign(# of contacts performed during this campaign)
* Duration (Last Contact duration)

Categorical Variables:

* Job
* Marital Status
* Education
* Default to a loan Payment
* Own House
* Loan taken

Target (Response variable):

* Term Deposit (Yes/No)

**Sample Data** (Source: bank\_marketing\_data.csv(details in github repository) file)



**Exploratory Visualization**

**For Continuous/Numeric variables** (age, balance, duration etc.):

Histogram for Numeric Variable with respect to term deposit Flag = "No"

A screen shot of a building

Description automatically generated

Histogram for Numeric Variable with respect to term deposit Flag = "Yes"

A picture containing crossword puzzle, shoji, text

Description automatically generated

**Observation:**

* Middle age Customers have accepted the Term deposit offer
* Lower account Balance led to higher acceptance of term deposit offer
* Lower Contact call duration led to higher acceptance of term deposit offer
* Lower Campaign(# of contacts) call resulted higher acceptance of term deposit offer
* Low Previous # of contacts led to acceptance of term deposit offers

**For Categorical Variables**:

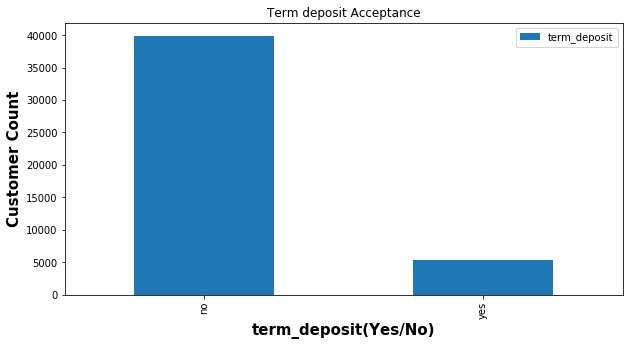
* The Customers with following attributes have accepted the term deposit offers
* Job (Management, Technician, Admin, Blue- Collar)
* Married
* Default-No Defaulted Customers
* Education-Customers with tertiary/secondary education
* Housing-Customers with (no Housing)
* Loan-Customers with no loan
* Contact-Customer contacted by Cellular

A screenshot of a computer

Description automatically generated

1. **For Target (Response) variable-**

* Term deposit (Yes/No)
  + Observation: **Imbalanced** Data set. As you can see, we have more term deposit (No) data points than term deposit(Yes) Data points

****

# RESULTS

**Model Evaluation and Validation**

I have used the following Strategy for Model selection process

* Training Data set to fit model
* Validation Data set to choose best model (Model’s **effectiveness)**
* Test Data Set to estimate **performance/quality** /**Evaluate** of Chosen Model

1. GridSearch Cross validation- I have used the K-fold Cross validation to measure the effectiveness of the model. This step randomly divides the set of observations into k groups. The 1st set is treated as Validation set and the method is fit (Training) on remaining k-1 fold

Fitting 10 folds for each of 40 candidates, totalling 400 fits

[Parallel(n\_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.

[Parallel(n\_jobs=-1)]: Done 34 tasks | elapsed: 10.4s

[Parallel(n\_jobs=-1)]: Done 184 tasks | elapsed: 2.7min

[Parallel(n\_jobs=-1)]: Done 400 out of 400 | elapsed: 3.5min finished

best\_score(Mean cross\_validated score) : 0.8712323773325226

Best Penalty: l1

Best C: 1.0

1. Test the Model using Test Data

grid\_mod = GridSearchCV(logistic,param\_grid=param\_grid,cv=10,n\_jobs=-1, verbose =1,scoring='roc\_auc')

precision recall f1-score support

0.0 0.98 0.81 0.89 4477

1.0 0.25 0.79 0.38 364

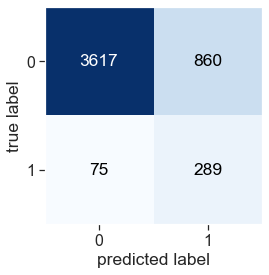
accuracy 0.81 4841

macro avg 0.62 0.80 0.63 4841

weighted avg 0.92 0.81 0.85 4841

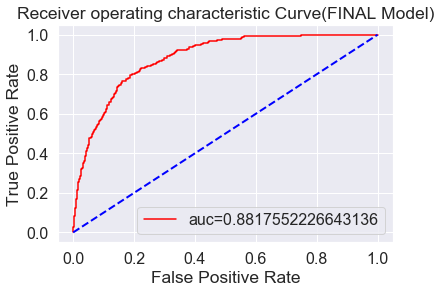
Accuracy-FINAL Model: 0.8068580871720719

1. **Confusion Matrix**
   1. Model has correctly predicted 3617 customers to not accept the Term deposit offer(True Negative)
   2. Model has correctly predicted 289 customers to accept the offer(True Positive)
   3. Model has in correctly predicted 75 customers to accept the offer(False Positive)
   4. Model has correctly predicted 860 customers NOT to accept the offer( False negative)

****

1. Model -**ROC Curve**

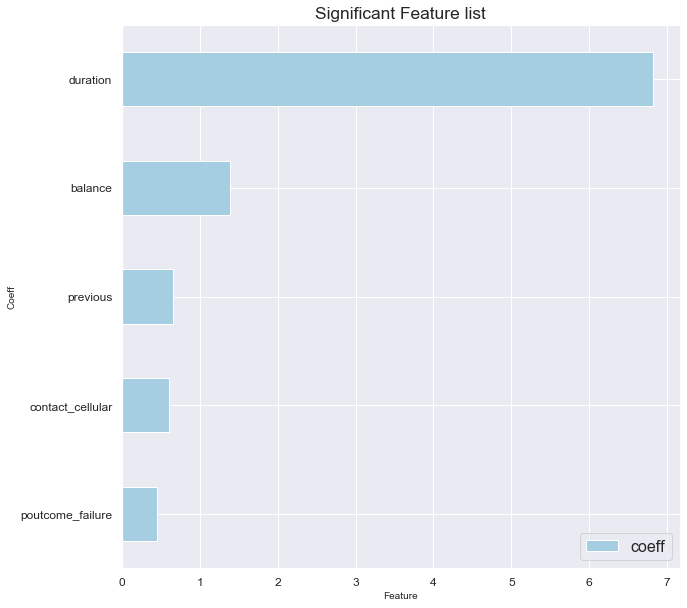
Receiver Operating Characteristic(ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity. The AUC score was **88%**



Based on Model effectiveness of 87**%** and Recall of 81**%**, I would believe our model is predicting the customer acceptance of term deposit offers correctly. However, the Model needs to get updated/trained with new Ground Truth Labels (i.e. new Term deposit accepted data) to sustain the prediction accuracy.

1. **Significant Features from the Model**

The following features/predictor variables are significant to drive higher response rate for term deposit offers



# CONCLUSION

**Reflection**

* I believe I have followed the Machine learning flows in this capstone Project, by leveraging Udacity excellent learning materials and 6 credit hours course work from Georgia Tech-Online MS in Analytics program.
* Still I think Modeling is an art. We get better at it as we use it more.
* Based on prediction recall (=81%), I think this model should it be used in a general setting to predict term deposit offers for Bank Customers.

**Improvement**

* **Model Tuning**
  + Need to retrain the model with ground truth (i.e. new Bank customers data with accepted term deposit offers) to boost model prediction
  + Need to leverage Amazon Sage Maker or any other Cloud Machine learning Platform to avail Model tuning options.
  + We can use sklearn ‘s RandomizedSearchCV to perform random search on Hyper Parameters
  + We could have used different classification models such as:
    - * Gaussian Naive Bayes
      * Decision Tree
      * Random Forest
      * XGBoost
* **Training Set Quality** 
  + Need to resample data in order to get a “balanced” data set
  + Need to ensure all the required predictor variable data were populated
  + Instead of removing rows with Missing data, need to assess the data imputation methods (such as Mean/Mode/Median, Prediction model etc.)

**References**

1. Choosing the right metric [Choosing the Right Metric for Evaluating Machine Learning Models](https://www.kdnuggets.com/2018/04/right-metric-evaluating-machine-learning-models-1.html).
2. Article about [What metrics should be used for evaluating a model on an imbalanced data set](https://towardsdatascience.com/what-metrics-should-we-use-on-imbalanced-data-set-precision-recall-roc-e2e79252aeba)
3. Article on [various techniques of the data exploration process](https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/).
4. Data Mining for Business Analytics- *Galit Shmueli, pertr C Bruce ,Wiley Publications*
5. DataCamp: Logistic regression :<https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python>