Data Glacier

Your Deep Learning Partner

Team Name: Intern_Project					
Sl.No	Name	Email	Country	Company	Specialization
1	Vijayarajan Vijaya Jothi	vijayajothi23s@gmail.com	United Kingdom	DataGlacier	DataScience

Project Report

Table Of Contents

Problem Description:

ABC Bank wants to sell it's term deposit product to customers and before launching the product they want to develop a model which help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

Business Understanding:

Bank wants to use ML model to shortlist customer whose chances of buying the product is more so that their marketing channel (tele marketing, SMS/email marketing etc) can focus only to those customers whose chances of buying the product is more.This will save resource and their time (which is directly involved in the cost (resource billing)).Develop model with Duration and without duration feature and report the performance of the model.Duration feature is not recommended as this will be difficult to explain the result to business and also it will be difficult for business to campaign based on duration.

Dataset

Data Set Information :

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

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

Attribute Information:
Input variables:
bank client data:
1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no','yes','unknown')
6 - housing: has housing loan? (categorical: 'no','yes','unknown')
7 - loan: has personal loan? (categorical: 'no','yes','unknown')
related with the last contact of the current campaign:
8 - contact: contact communication type (categorical: 'cellular','telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=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.
other attributes:
12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - 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)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
social and economic context attributes
16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):
21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Project Lifecycle

Project Lifecycle:

Tasks	Week7	Week 8	Week 9	Week 10	Week 11	Week 12
	19-Oct	26-Oct	02-Nov	09-Nov	16-Nov	23-Nov
Business Understanding						
Data understanding						
Exploratory data Analysis						
Data Preparation						
Model Building (Logistic Regression, ensemble, Boosting etc)						
Model Selection						
Performance reporting						
Deploy the model						
Converting ML metrics into Business metric and explaining result to business						
Prepare presentation for non technical persons.						

Data Intake Report

Data Intake Report:	
Name: Bank Marketing – Data Science	
Name: Bank Marketing – Data Science **Report date: 19th October 2024** **Internship Batch: LISUM37** **Version: 1.0** **Data intake by: Vijayarajan Vijaya Jothi** **Data intake reviewer: Vijayarajan Vijaya Jothi**	
Data Storage Location:	https://github.com/VijayaJothi24/dataGlacier/tree/main/Week7_Project

Tabular data details:

Total number of observations	3424
Total number of files	3
Total number of features	17
Base format of the file	.csv and .txt
Size of the data	567 KB

Github Repository:

Github Repository-https://github.com/VijayaJothi24/dataGlacier/tree/main/Week7_Project(LISUM37: 30 August - 30 Nov 24)

Submitted by: Vijayarajan Vijaya Jothi

Submitted to: Data Glacier

Date: 19th October 2024

Data Types

In this project, with reference from the Data intake Report, We have dataset with the following datatypes, "object types" means categorical columns:

bank client data:
1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no','yes','unknown')
6 - housing: has housing loan? (categorical: 'no','yes','unknown')
7 - loan: has personal loan? (categorical: 'no','yes','unknown')
related with the last contact of the current campaign:
8 - contact: contact communication type (categorical: 'cellular','telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=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.
other attributes:
12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - 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)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
social and economic context attributes
16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):
21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Data Problems

Null Values: This Dataset has NO NULL VALUES
Outliers: Also,have only two numerical columns and both of them have few outliers.
Skewness and Kurtosis Also,got some skewness and kurtosis in two numerical columns.

Transformation

As there is hardly no any null values, to perform in transformation is almost nil. We have some skewness and kurtosis in our two numerical features, scale the values by StandardScaler() is performed. Outliers are removed by calcultaing IQR and remove data smaller/greater than two whiskers.

Machine Learning

Machine learning is complex, but it's really about teaching computers to learn from experience and improve over time, just like we do in our everday lives, such as sorting emails,or any other classification problems , where an alogorithm is createds in place to solve issues.

Ridge Regression

It is used when there is a high correlation between the independent variables. This is because, in case of multi collinear data , the least square estimates gives unbiased values. But, in case the collinearity is very high, there can be some bias value. Therefore, a bias matrix is introduced in the equation of Ridge Regression. This is a powerful regression method where the model is less susceptible to overfitting.

λ

Below is the equation used to denote the Ridge Regression, where the introduction of

(Lambda) solves the problem of multicollinearity.

The matrix X^TX+λI has full rank and it is **invertible**. As a consequence:

$$\beta = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$$

Logistic Regression

Logistic regression is a statistical method used for binary classification problems, where the goal is to predict the probablity that an instance belongs to one of two classes.Logistic Regression is actua... ion algorithm.

$$y = \frac{e^{(b_0 + b_1 X)}}{1 + e^{(b_0 + b_1 X)}}$$

Logistic Regression – Sigmoid Function

here,

- x = input value
- y = predicted output
- b0 = bias or intercept term
- b1 = coefficient for input (x)

Random Forest Classifier

It is an ensemble learning method for classification and regression tasks, thart operate by constructing multiple decision trees (each trained on a subset of samples using a subset of features) at training time and outputing the class that is mode of the classes (classification) or mean prediction (regresssion) of the individual trees.

Because they are extremely robust, easy to get started with, good at heterogeneous data types, and have very few hyperparameters, random forests are often a data scientist's first port of call when developing a new machine learning system, as they allow data scientists to get a quick overview of what kind of accuracy can reasonably be achieved on a problem, even if the final solution may not involve a ranom forest.

Outlier Detection and Handling

```
#Outlier Detection and Handling:
#Identify and remove outliers in the 'balance' column:
Q1 = new_bank_data['balance'].quantile(0.25)
Q3 = new_bank_data['balance'].quantile(0.75)
IQR= Q3-Q1

print(Q1)
print(Q3)
print(IQR)
```

```
#Outlier Detection and Handling:
#Identify and remove outliers in the 'age' column:
Q1 = new_bank_data['age'].quantile(0.25)
Q3 = new_bank_data['age'].quantile(0.75)
IQR= Q3-Q1

print(Q1)
print(Q3)
print(IQR)
```

```
Bank_data1=Bank_data[ ~((Bank_data['balance']<(Q1-1.5*IQR))|(Bank_data['balance']>(Q3+1.5*IQR)))]
print(Bank_data1)
```

```
Bank_data1=Bank_data[ ~((Bank_data['age']<(Q1-1.5*IQR))|(Bank_data['age']>(Q3+1.5*IQR)))]
print(Bank_data1)
```

```python
import pandas as pd
Bank_data4 = Bank_data['age'].interpolate()
print(Bank_data4)
```

```
0        58
1        44
2        33
3        47
4        33
         ..
45206    51
45207    71
45208    72
45209    57
45210    37
Name: age, Length: 45211, dtype: int64
```

```python
import pandas as pd
Bank_data4 = Bank_data['balance'].interpolate()
print(Bank_data4)
```

```
0        2143
1          29
2           2
3        1506
4           1
         ...
45206     825
45207    1729
45208    5715
45209     668
45210    2971
Name: balance, Length: 45211, dtype: int64
```

+ Code + Markdown

```python
import numpy as np

def whisker(col):
    q1, q3 = np.percentile(col, [25, 75])
    iqr = q3 - q1
    lw = q1 - 1.5 * iqr
    up = q3 + 1.5 * iqr
    return lw, up
```

```python
lw, up = whisker(Bank_data['duration'])
print(f'Lower whisker: {lw}')
print(f'Upper whisker: {up}')
```

```
Lower whisker: -221.0
Upper whisker: 643.0
```

```python
lw, up = whisker(Bank_data['campaign'])
print(f'Lower whisker: {lw}')
print(f'Upper whisker: {up}')
```

```
Lower whisker: -2.0
Upper whisker: 6.0
```

Statistical Summary Exploratory Data Analysis

```
Bank_data.describe()
```

	sl. no	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	22606.000000	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	13051.435847	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	1.000000	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	11303.500000	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	22606.000000	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	33908.500000	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	45211.000000	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

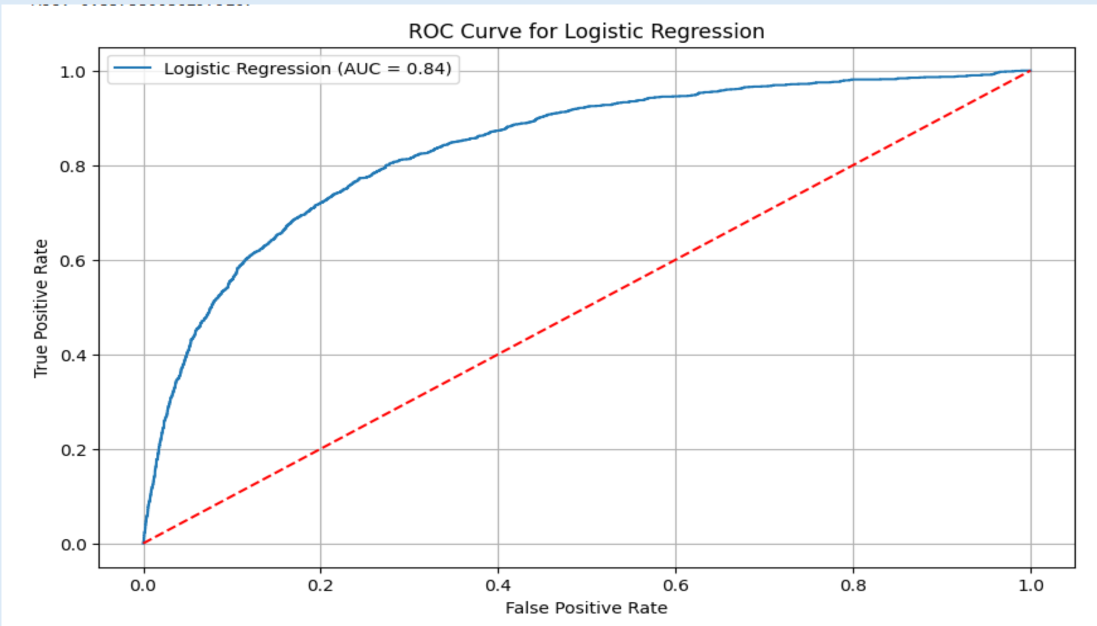
Data

The one step performed is "one hot encoding" , For using classifiers we need numerical values, to do this I used One Hot Encoding that implemented by "get_dummies()" function from Pandas Library.

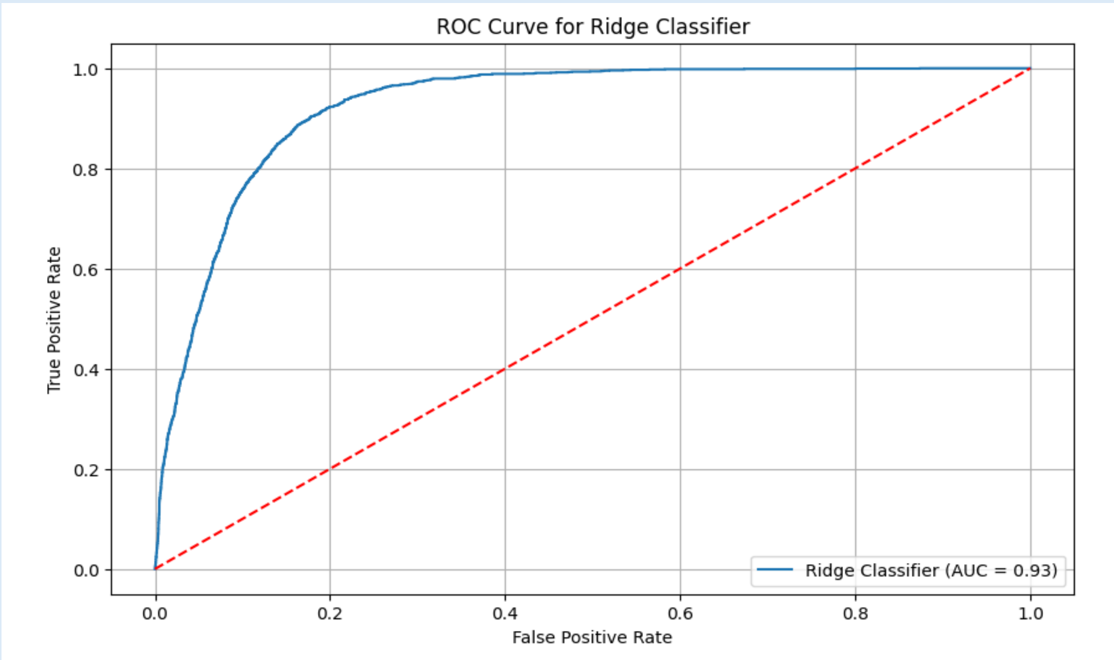
Model Training

Now , Prepared data suits to perform classifiers models on the train set which is derived by splitting whole dataset to train and test sets in the way 70% fro train set and 30% for test set.

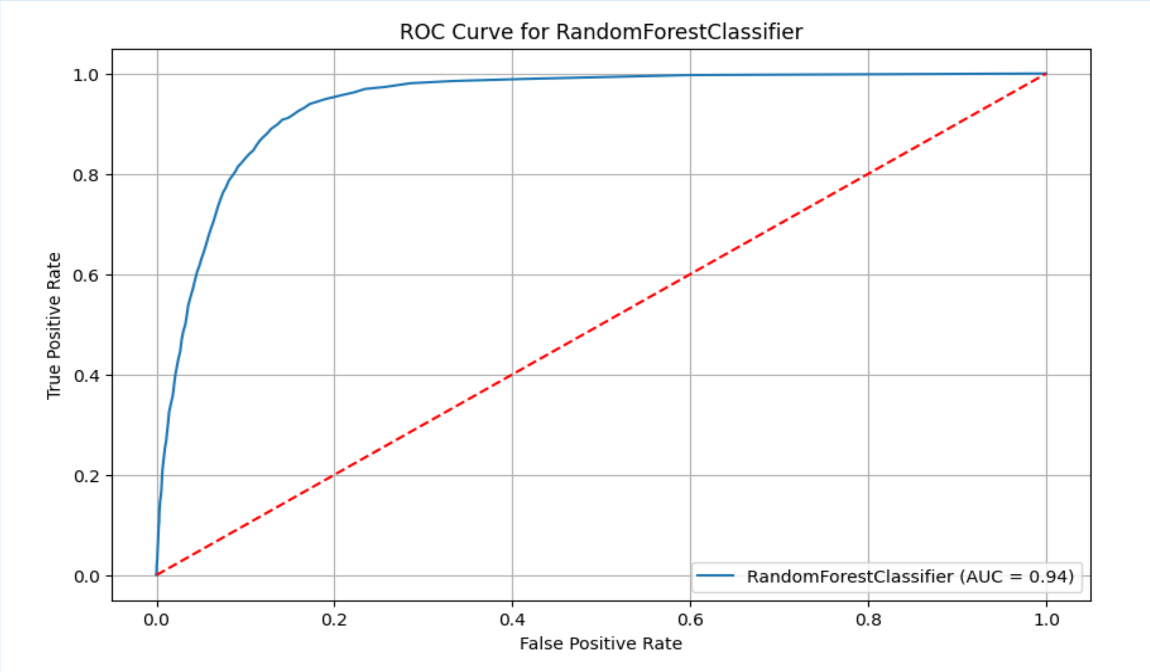
Model Creation - Logistic Regression



Model Creation - Ridge classifier



Model Creation - Random forest Classifier



	precision	recall	f1-score	support
No Deposited	0.93	0.97	0.95	11966
Deposited	0.68	0.49	0.57	1598
accuracy			0.91	13564
macro avg	0.81	0.73	0.76	13564
weighted avg	0.90	0.91	0.91	13564

Accuracy: 0.9128575641403716
AUC: 0.9421902942776749

Conclusion

Approximately all the classifiers have same result, but Random Forest was the best one.
The model has around 89% Accuracy.
Random Forest has 93% Precision, 97% Recall, & 95% F1 Score.
