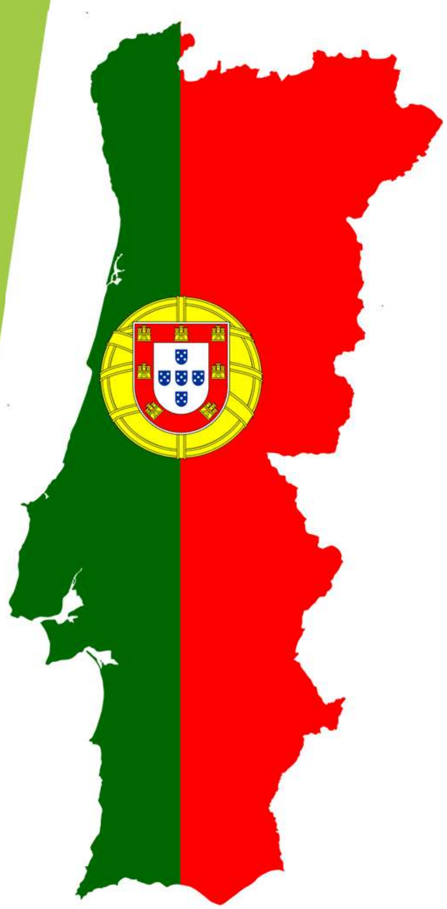


# Predict if client will subscribe to term deposit or not

Aishwarya Pai
Sparsh Tekriwal
Siddhesh Karanjkar
Naeem Sunesara
Deep Doshi

# Problem Statement


- Decline in revenues for a Portuguese bank
- Root cause is clients not depositing as frequently as before
- Term deposit allow banks to hold onto investments for a specific amount of time
- These deposits --> invest in higher gain financial products == Profit
- Or, customers can be persuaded to buy other products such as funds or insurance == further increased revenues
- Hence, goal is to identify likely customers and focus marketing budgets on them.

# About the Dataset

- There are 41,188 observations and 21 Variables in the Data Set.
- There are 10 continuous and 10 categorical variables.
- The target response (y) is a binary response indicating whether the client subscribed to a term deposit or not.
- Our first objective was to determine which variables have the highest influence on whether a client purchases a term deposit or not.
- The second objective is to determine patterns in variables that produce the most term deposit purchases.

# Train Data Description- 41188 rows & 20 features

20 features categorized into 3 categories

**Category 1: Personal Attributes**

| Column | Renamed Column | Description | Type | Unique Value Count |
|---|---|---|---|---|
| age | age | Age of a person | Numeric | 78 |
| job | job | Job status | Categorical | 12 |
| marital | marital | Marital status | Categorical | 4 |
| education | education | Education Status | Categorical | 8 |
| default | credit_default | Has credit in default? | Categorical | 3 |
| housing | housing_loan | Has housing loan? | Categorical | 3 |
| Loan | personal_loan | Has personal loan? | Categorical | 3 |

# Data Description

**Category 2: Contact Related Details**

| Column | Renamed Column | Description | Type | Unique Value Count |
|---|---|---|---|---|
| contact | contact_type | Contact communication type | categorical | 2 |
| month | last_contact_month | Last contact month of year | Categorical | 10 |
| day_of_week | last_contact_day_of_week | Last contact day of week | Categorical | 5 |
| duration | last_contact_duration | Last contact duration | Numerical | 1544 |
| campaign | no_of_contacts | Number of contacts performed in a campaign | Numerical | 42 |
| pdays | time_between_contacts | Number of days passed from previous contact | Numerical | 27 |
| previous | previous_no_of_contacts | Number of contacts before this campaign | Numerical | 8 |
| poutcome | prev_outcome | Outcome of previous campaign | Categorical | 3 |

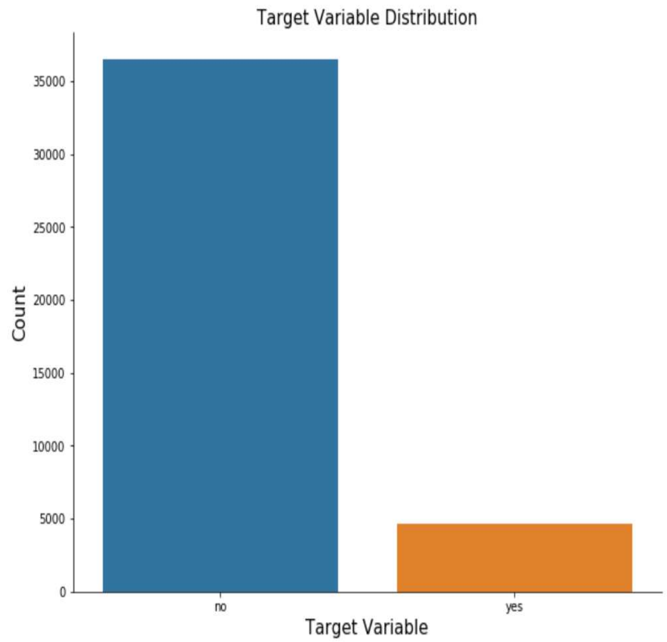
# Data Description

### Category 3 Various Indexes

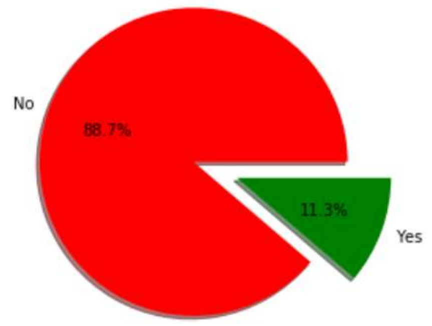
| Column | Renamed Column | Description | Type | Unique Value Count |
|---|---|---|---|---|
| emp.var.rate | emp_var_rate | Employment variation rate(quarterly indicator) | Numerical | 10 |
| cons.price.idx | consumer_price_index | Consumer price index (monthly indicator) | Numerical | 26 |
| cons.conf.idx | consumer_conf_index | Consumer confidence index (monthly indicator) | Numerical | 26 |
| euribor3m | euribor_3month_rate | Euribor 3 month rate (daily indicator) | Numerical | 316 |
| nr.employed | num_of_employed | Number of employees (quarterly indicator) | Numerical | 11 |

# Target : Will the client Subscribe?
# No Probably !!

```
fig, ax = plt.subplots()
fig.set_size_inches(10, 8)
sns.countplot(x = 'target', data = df)
ax.set_xlabel('Target Variable', fontsize=15)
ax.set_ylabel('Count', fontsize=15)
ax.set_title('Target Variable Distribution', fontsize=15)
sns.despine()
# there is a class imbalance that needs to be handled
```




```
labels = 'No', 'Yes'
sizes = [36548, 4640]
colors = ['red', 'green']
explode = (0.3, 0)  # explode 1st slice

# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=0)

plt.axis('equal')
plt.show()
```



# Missing Data !!!

```
In [14]: df.isnull().sum()

Out[14]: age                          0
         job                          0
         marital                      0
         education                    0
         credit_default               0
         housing_loan                 0
         personal_loan                0
         contact_type                 0
         last_contact_month           0
         last_contact_day_of_week     0
         last_contact_duration        0
         no_of_contacts               0
         time_between_contacts        0
         previous_no_of_contacts      0
         prev_outcome                 0
         emp_var_rate                 0
         consumer_price_index         0
         consumer_conf_index          0
         euribor_3month_rate          0
         num_of_employed              0
         target                       0
         dtype: int64
```

- There are no missing values in any column
- But there are some abnormal values like **Unknown & 999**
- Here we have considered Unknown in all the features as a new class
- 999 is also kept as it is because it makes more sense according to business

# Outliers, There are outliers detected by IQR method, but none are Outliers actually!!

```python
numeric_cols = ['age',
 'last_contact_duration',
 'no_of_contacts',
 'time_between_contacts',
 'previous_no_of_contacts',
 'emp_var_rate',
 'consumer_price_index',
 'consumer_conf_index',
 'euribor_3month_rate',
 'num_of_employed']
for col in numeric_cols:

    Q1=df[col].quantile(q = 0.25)
    Q2=df[col].quantile(q = 0.50)
    Q3=df[col].quantile(q = 0.75)
    Q4=df[col].quantile(q = 1.00)

    IQR= Q3-Q1
    print('The Range for',col ,'is', Q1 - 1.5*(IQR), 'to', Q3 + 1.5*(IQR))
    print('There are',sum((df[col]>(Q3 + 1.5*(IQR))) | (df[col] < (Q1 - 1.5*(IQR)))),"outliers in ", col)
```

```
The Range for age is 9.5 to 69.5
There are 469 outliers in  age
The Range for last_contact_duration is -223.5 to 644.5
There are 2963 outliers in  last_contact_duration
The Range for no_of_contacts is -2.0 to 6.0
There are 2406 outliers in  no_of_contacts
The Range for time_between_contacts is 999.0 to 999.0
There are 1515 outliers in  time_between_contacts
The Range for previous_no_of_contacts is 0.0 to 0.0
There are 5625 outliers in  previous_no_of_contacts
The Range for emp_var_rate is -6.6000000000000005 to 6.200000000000001
There are 0 outliers in  emp_var_rate
The Range for consumer_price_index is 91.69650000000001 to 95.3725
There are 0 outliers in  consumer_price_index
The Range for consumer_conf_index is -52.150000000000006 to -26.949999999999992
There are 447 outliers in  consumer_conf_index
The Range for euribor_3month_rate is -4.081499999999999 to 10.3865
There are 0 outliers in  euribor_3month_rate
The Range for num_of_employed is 4905.6 to 5421.6
There are 0 outliers in  num_of_employed
```

**IQR Outlier Detection Method**

Anything above Q3+1.5*IQR or below Q3 – 1.5*IQR is an outlier

# Age & Job

```python
temp = pd.crosstab(pd.cut(df['age'],bins=[17,20,25,55,60,100]),df['target'])
total = (temp['no']+temp['yes'])
temp['ratio'] = temp['yes']/total
print(temp)
```

| target / age | no | yes | ratio |
|---|---|---|---|
| (17, 20] | 80 | 55 | 0.407407 |
| (20, 25] | 1234 | 292 | 0.191350 |
| (25, 55] | 32390 | 3550 | 0.098776 |
| (55, 60] | 2345 | 327 | 0.122380 |
| (60, 100] | 496 | 414 | 0.454945 |

```python
temp = pd.crosstab(df['job'],df['target'])
total = (temp['no']+temp['yes'])
temp['ratio'] = temp['yes']/total
print(temp)
```

| target / job | no | yes | ratio |
|---|---|---|---|
| admin. | 9070 | 1352 | 0.129726 |
| blue-collar | 8616 | 638 | 0.068943 |
| entrepreneur | 1332 | 124 | 0.085165 |
| housemaid | 954 | 106 | 0.100000 |
| management | 2596 | 328 | 0.112175 |
| retired | 1286 | 434 | 0.252326 |
| self-employed | 1272 | 149 | 0.104856 |
| services | 3646 | 323 | 0.081381 |
| student | 600 | 275 | 0.314286 |
| technician | 6013 | 730 | 0.108260 |
| unemployed | 870 | 144 | 0.142012 |
| unknown | 293 | 37 | 0.112121 |

While Age is just a number, it matters a lot here, clients before the age of 20 and who are students and those above 60 and retired are highly likely to subscribe as compared to middle aged clients

# Education : Illiterates very low in numbers but high in subscription

```
temp = pd.crosstab(df['education'],df['target'])
total = (temp['no']+temp['yes'])
temp['ratio'] = temp['yes']/total
print(temp)
```

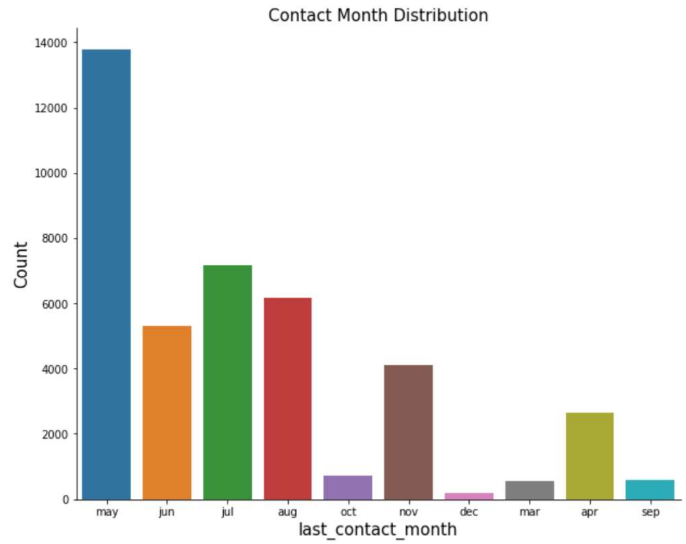
```
target                  no   yes     ratio
education
basic.4y              3748   428  0.102490
basic.6y              2104   188  0.082024
basic.9y              5572   473  0.078246
high.school           8484  1031  0.108355
illiterate              14     4  0.222222
professional.course   4648   595  0.113485
university.degree    10498  1670  0.137245
unknown               1480   251  0.145003
```

# Contacts : High in month of May-Aug, but the subscriptions are lower

```
temp = pd.crosstab(df['last_contact_month'],df['target'])
total = (temp['no']+temp['yes'])
temp['ratio'] = temp['yes']/total
print(temp)
```

```
target                no  yes     ratio
last_contact_month
apr                 2093  539  0.204787
aug                 5523  655  0.106021
dec                   93   89  0.489011
jul                 6525  649  0.090466
jun                 4759  559  0.105115
mar                  270  276  0.505495
may                12883  886  0.064347
nov                 3685  416  0.101439
oct                  403  315  0.438719
sep                  314  256  0.449123
```

```
fig, ax = plt.subplots()
fig.set_size_inches(10, 8)
sns.countplot(x = 'last_contact_month', data = df)
ax.set_xlabel('last_contact_month', fontsize=15)
ax.set_ylabel('Count', fontsize=15)
ax.set_title('Contact Month Distribution', fontsize=15)
sns.despine()
```

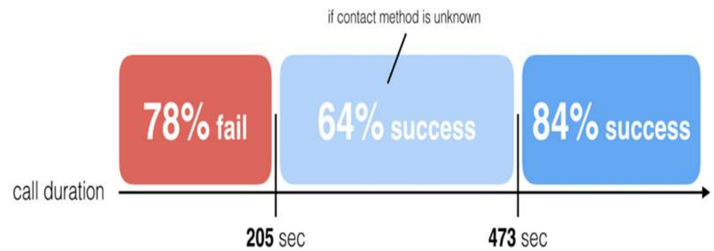


# Number of  Contacts & Call Duration

```python
temp = pd.crosstab(df['no_of_contacts'],df['target'])
total = (temp['no']+temp['yes'])
temp['ratio'] = temp['yes']/total
print(temp)
```

| target<br>no_of_contacts | no | yes | ratio |
|---|---|---|---|
| 1 | 15342 | 2300 | 0.130371 |
| 2 | 9359 | 1211 | 0.114570 |
| 3 | 4767 | 574 | 0.107471 |
| 4 | 2402 | 249 | 0.093927 |
| 5 | 1479 | 120 | 0.075047 |
| 6 | 904 | 75 | 0.076609 |
| 7 | 591 | 38 | 0.060413 |
| 8 | 383 | 17 | 0.042500 |
| 9 | 266 | 17 | 0.060071 |
| 10 | 213 | 12 | 0.053333 |
| 11 | 165 | 12 | 0.067797 |
| 12 | 122 | 3 | 0.024000 |
| 13 | 88 | 4 | 0.043478 |
| 14 | 68 | 1 | 0.014493 |
| 15 | 49 | 2 | 0.039216 |

- As the number of contacts increases probability of subscription reduces, above 10 there is extremely low probability of subscription




- One of the most important features is Call Duration, there is a clear indication that high durations lead to conversions

# Previous Contacts

```
temp = pd.crosstab(df['prev_outcome'],df['target'])
total = (temp['no']+temp['yes'])
temp['ratio'] = temp['yes']/total
print(temp)
```

```
target          no   yes     ratio
prev_outcome
failure       3647   605  0.142286
nonexistent  32422  3141  0.088322
success        479   894  0.651129
```

```
temp = pd.crosstab(df['previous_no_of_contacts'],df['target'])
total = (temp['no']+temp['yes'])
temp['ratio'] = temp['yes']/total
print(temp)
```

```
target                      no   yes     ratio
previous_no_of_contacts
0                        32422  3141  0.088322
1                         3594   967  0.212015
2                          404   350  0.464191
3                           88   128  0.592593
4                           32    38  0.542857
5                            5    13  0.722222
6                            2     3  0.600000
7                            1     0  0.000000
```

- There is a very low chance for first timers to convert

# Heatmap of Numerical Variables

# High Correlation Among Indexes

- num_of_employed & euribor_3month_rate : 0.95
- num_of_employed & emp_var_rate : 0.91
- num_of_employed  & consumer_price_index : 0.52
- euribor_3month_rate & consumer_price_index : 0.69
- consumer_price_index & emp_var_rate : 0.78
- previous_no_of_contacts & time_between_contacts: -0.59

## Modeling

- XGBoost
- Random Forest
- Decision Trees
- KNN
- Logistic Regression

## Handling Class Imbalance

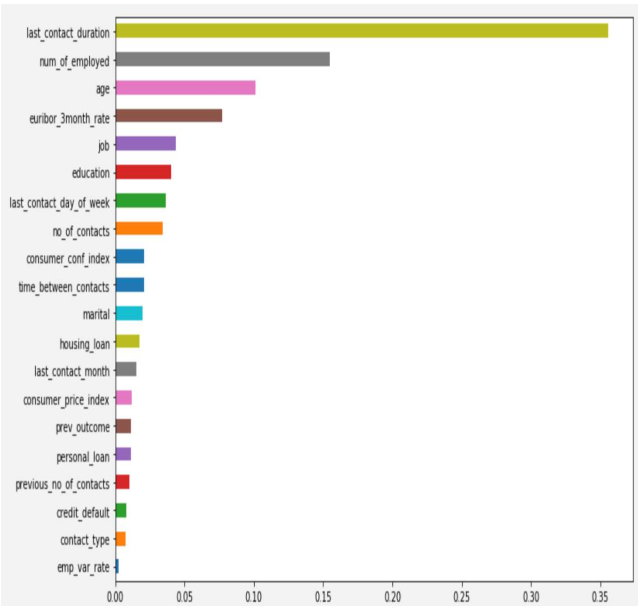
Actual Data: 36548:4640 → Down sampling & SMOTE(up sampling) → Model Data: 7000:7000

# First Model, Random Forest with all variables

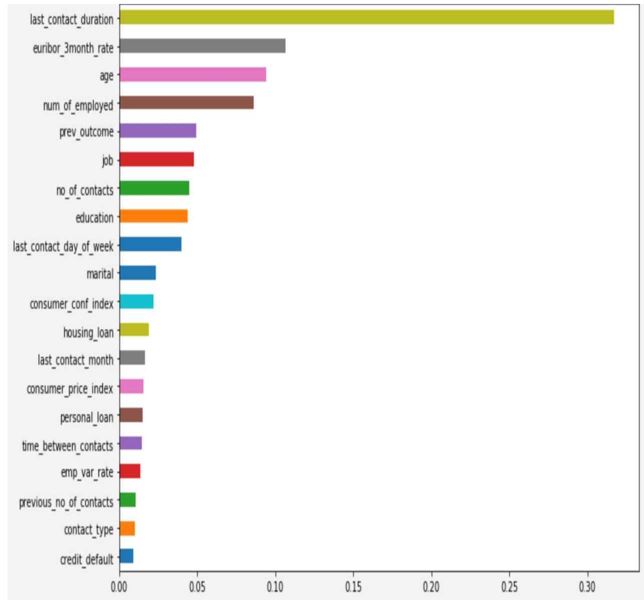
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.97 | 0.95 | 7303 |
| 1 | 0.66 | 0.47 | 0.55 | 935 |
| micro avg | 0.91 | 0.91 | 0.91 | 8238 |
| macro avg | 0.80 | 0.72 | 0.75 | 8238 |
| weighted avg | 0.90 | 0.91 | 0.91 | 8238 |

# Feature Selection: Pearson Correlation

# Feature Selection




Decision Tree Classifier




Random Forest Classifier

```
X_rfe.columns
```

```
Index(['age', 'job', 'education', 'last_contact_day_of_week',
       'last_contact_duration', 'no_of_contacts', 'time_between_contacts',
       'consumer_conf_index', 'euribor_3month_rate', 'num_of_employed'],
      dtype='object')
```

Recursive Feature Elimination

# KNN Results

## Selected Variables

- 'last_contact_duration'
- 'euribor_3month_rate'
- 'age'
- 'prev_outcome'
- 'job'
- 'no_of_contacts'
- 'education'
- 'last_contact_day_of_week'

## Results

### Confusion Matrix

```
print("Confusion Metrix:\n",confusion_matrix(y_test,knn1.predict(X_test)))

Confusion Metrix:
 [[1381   34]
 [  44 1341]]
```

### Classification Report

```
print (classification_report(y_test,pred1))

              precision    recall  f1-score   support

           0       0.97      0.98      0.97      1415
           1       0.98      0.97      0.97      1385

   micro avg       0.97      0.97      0.97      2800
   macro avg       0.97      0.97      0.97      2800
weighted avg       0.97      0.97      0.97      2800
```

# Best Model: Tuned Random Forest

## Selected Variables

- 'last_contact_duration’
- 'euribor_3month_rate’
- 'age’
- 'prev_outcome’
- 'job’
- 'no_of_contacts’
- 'education’
- 'last_contact_day_of_week’

## Results

### Confusion Matrix

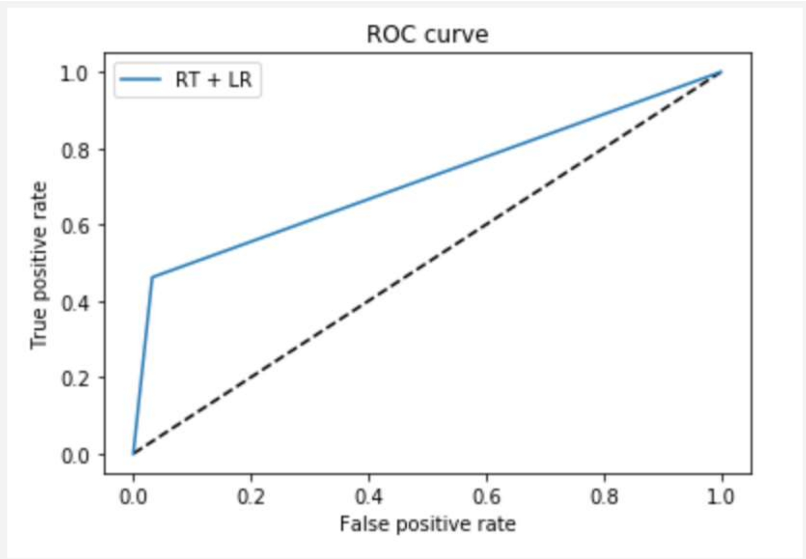
```
from sklearn.metrics import confusion_matrix
print("Confusion Metrix:\n",confusion_matrix(y_test1,rfc.predict(X_test1)))

Confusion Metrix:
 [[1412    3]
 [  34 1351]]
```

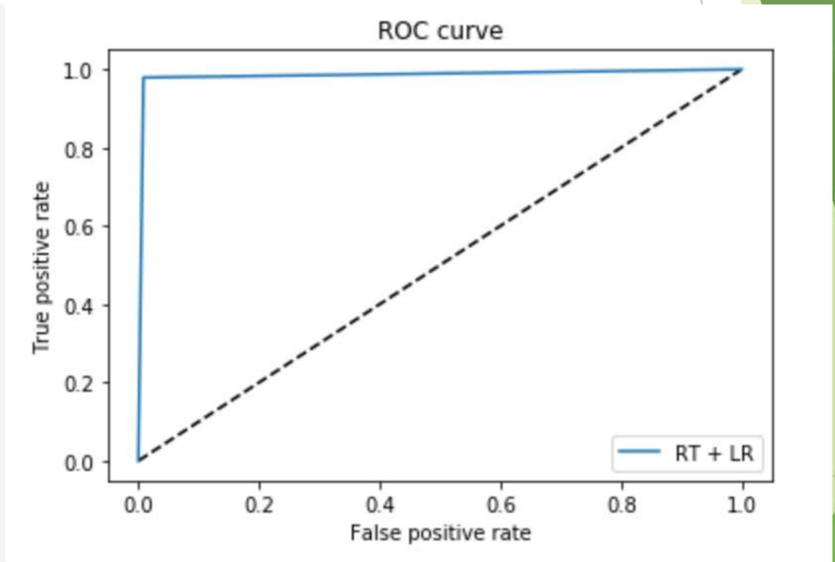
### Classification Report

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 1.00 | 0.99 | 1415 |
| 1 | 1.00 | 0.98 | 0.99 | 1385 |
| micro avg | 0.99 | 0.99 | 0.99 | 2800 |
| macro avg | 0.99 | 0.99 | 0.99 | 2800 |
| weighted avg | 0.99 | 0.99 | 0.99 | 2800 |

# Pre and Post Feature Selection ROC Curves




AUC = 0.71

AUC = 0.98

# Next Models in Pipeline

- Trying out Random Forest with a split of 9000(neg):7000(pos)
- Trying out Random Forest with a split of 12000(neg):7000(pos)
- XGBoost with hyper parameter tuning

# Final Thoughts

- Need more information about successful calls such as:
- The sales representatives who conducted the calls
- And create strategies to make the calls last longer
- Find out why the contact method has been recorded as unknown for some of the clients, rather than telephone or cellphone
- Correlations are not always causations, and there might be other hidden reasons for a client to subscribe:
- Longer calls could equate to interested clients asking questions or they could be setting up deposits over the phone
- It would be a good idea to set up a small A/B test to check if call duration is significantly impacting the subscription rate

# Thank You