MACHINE LEARNING FOUNDATIONS

PREDICTING CLIENT SUBSCRIPTION TO BANK TERM DEPOSITS - A BINARY CLASSIFICATION

Capstone Project

Chamila Wijayawardhana

Dialog Axiata PLC

Introduction

This is a binary classification problem where the goal is to predict if a client will subscribe to a term deposit or not.

The data is related to 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 assess if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

The source for the data is,

https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

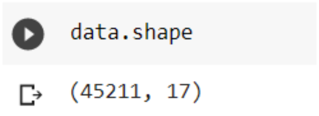
Initial analysis of the data revealed there is a significant class imbalance with a 88 : 12 split of the majority class ('no' to term deposit) and the minority class ('yes' to term deposits). **Hence this problem is identified as a binary classification problem with a class imbalance.**

Github repository for the project is,

https://github.com/chamalj1980/ML_Fundementals/tree/main/capstone%20project

Data

The data set comprises of 16 input variable and 45211 instances



```
data.shape
```

```
(45211, 17)
```

Note: above includes the output variable: y - has the client subscribed a term deposit? (binary: 'yes','no')

Identification of data types

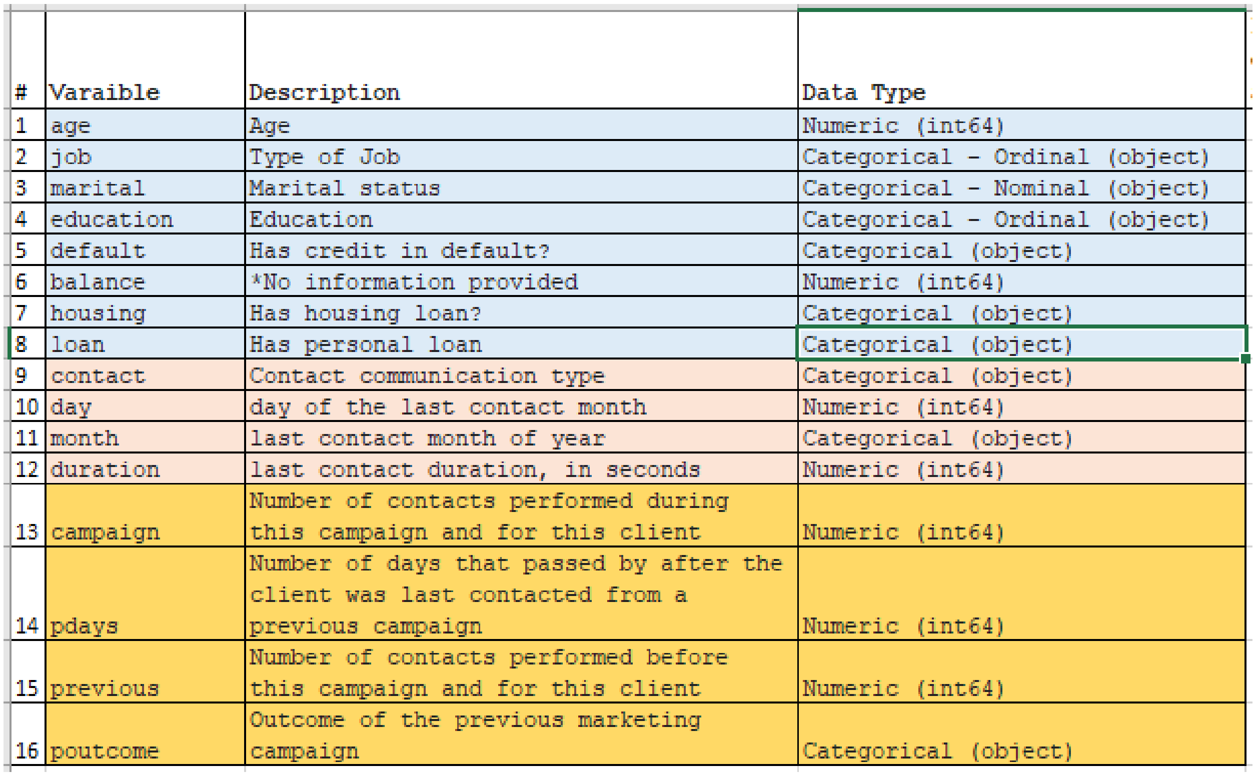
By analysing the details provided with the data set, following information and data types were identified. Data highlighted in "blue" are related bank client data, data highlighted in "red" are related to the last contact of the current campaign and data highlighted in "yellow" are other attributes related to current campaign and the previous campaigns.

Special Notes

1. Duration - It has been stated that this attribute highly affects the output target. 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. **Hence for this model building exercise this input is not considered.**

2. Balance - Information about this attribute is not explicitly provided with the data set. Hence it is assumed that this is related “any type of account balance” with the bank and is considered for analysis.

The table below summarizes the input features being analyzed and data identified data types.

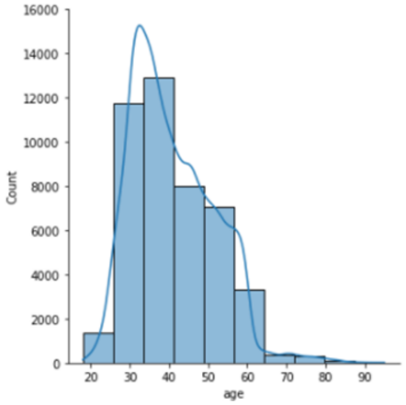


#	Varaible	Description	Data Type
1	age	Age	Numeric (int64)
2	job	Type of Job	Categorical - Ordinal (object)
3	marital	Marital status	Categorical - Nominal (object)
4	education	Education	Categorical - Ordinal (object)
5	default	Has credit in default?	Categorical (object)
6	balance	*No information provided	Numeric (int64)
7	housing	Has housing loan?	Categorical (object)
8	loan	Has personal loan	Categorical (object)
9	contact	Contact communication type	Categorical (object)
10	day	day of the last contact month	Numeric (int64)
11	month	last contact month of year	Categorical (object)
12	duration	last contact duration, in seconds	Numeric (int64)
13	campaign	Number of contacts performed during this campaign and for this client	Numeric (int64)
14	pdays	Number of days that passed by after the client was last contacted from a previous campaign	Numeric (int64)
15	previous	Number of contacts performed before this campaign and for this client	Numeric (int64)
16	poutcome	Outcome of the previous marketing campaign	Categorical (object)

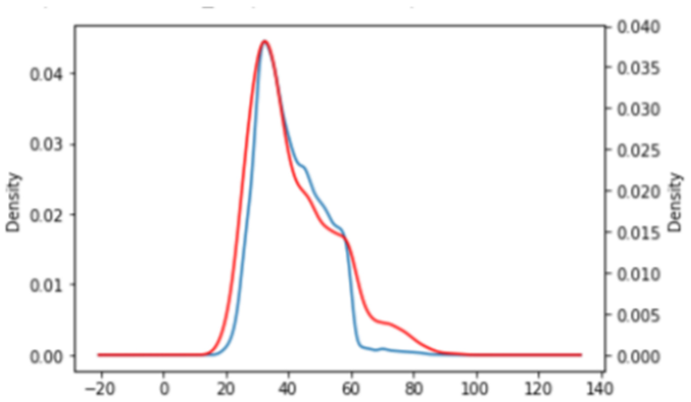
Descriptive Statistics

Following numerical variables are analyzed in this exercise: (a) Age, (b) Balance, (c) Campaign, (d) Pdays and (e) Previous.

Descriptive statistics measures (mean, standard deviation, mode etc.) for these parameters are calculated and included with the notebook provided. Analyses of the **‘age’ variable** for the population and for the clients who have subscribed for (y = ‘yes’) and not subscribed y=’no’) revealed that they have similar probability distributions . Analyses of other numerical variables **revealed a similar pattern** where there is no discernible difference in probability distribution with regards to the target variable.

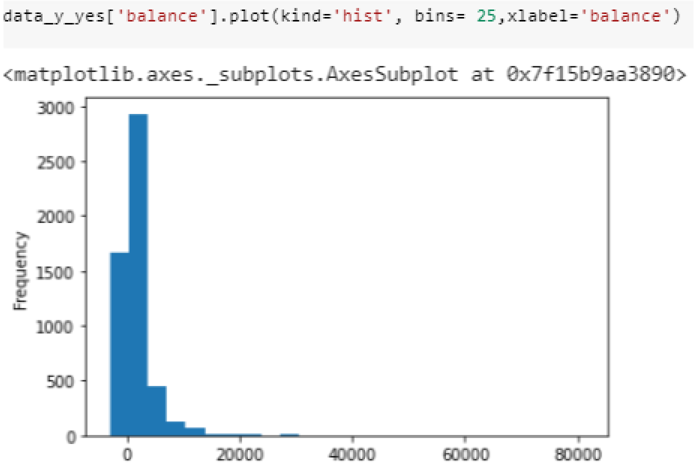


Distribution pf ‘Age’ attribute for the total data set



Distribution of ‘Age’ for subscription (y=’yes’) and those who have not opted for subscription (y = ‘no’)

Analyses of the **‘balance’ attribute** revealed that clients who subscribed to the term deposit (y = ‘yes’) either **had minus or lower account balances**.



```
data_y_yes['balance'].plot(kind='hist', bins= 25,xlabel='balance')
```

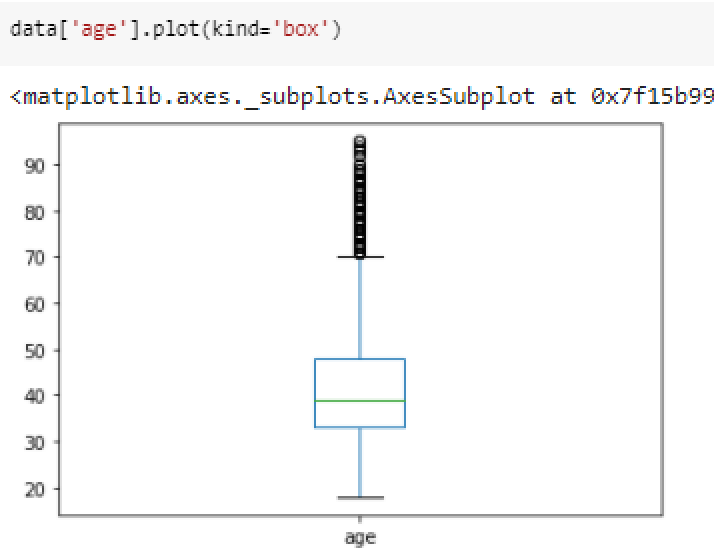
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f15b9aa3890>
```

Analyses of the **‘campaign’ variable** revealed that 90% of the **data are related to 1 to 5 campaigns** (i.e 1 to 5 contact performed to this client for this period). Analyses of the ‘**pdays’ and ‘previous’ attributes** revealed that **majority of the clients (82%) have not been contacted in the previous contact** and 64% of those subscribed to the term deposit have not been previously contacted. Also the fact that more times the clients are contacted they are unlikely to subscribe to the term deposit.

Identifying null values and outliers - numerical variables

The **numerical attributes** that were analyzed **do not have ‘null’ values.** However the ‘pdays’ attribute has 82% of (-1) values , indicative of the percentage of people that have not been contacted in the previous campaigns. This conveys similar information to that of the ‘previous’ attribute and may possibly be discarded (pdays attribute) for consideration for the machine learning model.

Analysis of the **‘box plot’** for the numerical variables revealed that **all of the variables** have a significant **number of ‘outliers’** and will need to undergo ‘feature scaling’ in preparation for the machine learning model.



```
data['age'].plot(kind='box')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f15b99
```

Exploring Categorical Variables

Following categorical variables were identified from the data set : (a) Job, (b) Marital, (c) Education, (d) Default, (e) Housing, (f) Loan (g) Contact, (h) Month (i) poutcome

Identifying null values in categorical variables

Analysis of these attributes revealed there are a significant **number of “unknown” values** associated. These values are treated as null values. For the **‘job’** and ‘**education’** attributes, these null values are **replaced with the highest frequency category** of each of the attributes.

The '**poutcome'** attribute (Outcome of the previous marketing campaign) has **82% of unknown values** and hence will **not be considered** for building the machine learning model**.**

Methodology (Solutions approach, Tools used)

As there is significant **class imbalance** in this classification problem, with with class 0 (y = no) and class 1 (y = yes) having 88 : 12 percent split, **'Random Forest' algorithm** is primarily selected for building the machine learning model as its ensemble properties may assist in negating the the imbalance. However for the comparison purposes other classification models are also considered in this analysis.Additionally an attempt has been made to **undersample the majority class** to treat the aforementioned class imbalance.

Model Training Pipeline

Data Pre-processing

Column selection

Based on exploratory data analytics and taking into consideration correlation of attributes with the target variable, following numerical and categorical attributes were selected for the initial model building : `'age','balance','campaign','previous', 'job_group', 'education', 'default', 'housing', 'loan'`

Treatment of numerical variables

All the **numerical variables** were **standardized on Z-score normalization** as the probability distributions of all these variables are skewed.

Treatment of categorical variables

(a) Replacement of null values

Null values in **'job'** and **'education'** are **imputed** by the **most frequently occurring category** for each attribute.

(b) Grouping

There are **11 different job categories** as listed in the data set and they are **grouped into 3 categories** as follows, Job_category1 - {unemployed, students, housemaids} , Job_category 2 - {bluecollar, technician, services, admin, retired}, Job_category 3 - {management, self employed,entrepreneur}

(c) Encoding

All of the selected **categorical variables are one-hot encoded** for building the machine learning model

Model training

Model was trained on a 70 : 30 split of training and testing data. As the problem is that of a class imbalance the training data set was resampled using the Random Undersampling technique for the majority class.

However this resampling did not improve the model performance for both the classes and for further processing in the pipeline the origins train test split was retained.

Fitting multiple models with different hyper parameters

For the purpose of selecting the best model (parameters) , multiple Random Forest classifiers with the below listed hyper parameters were tested.

- N_estimators: The number of trees in the forest
- The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

The combination of hyperparameters selected are as follows.

Model Name	**N_estimators**	**Maximum depth**
rf1	100	None
rf2	100	10
rf3	500	5
rf4	500	10
rf5	500	15

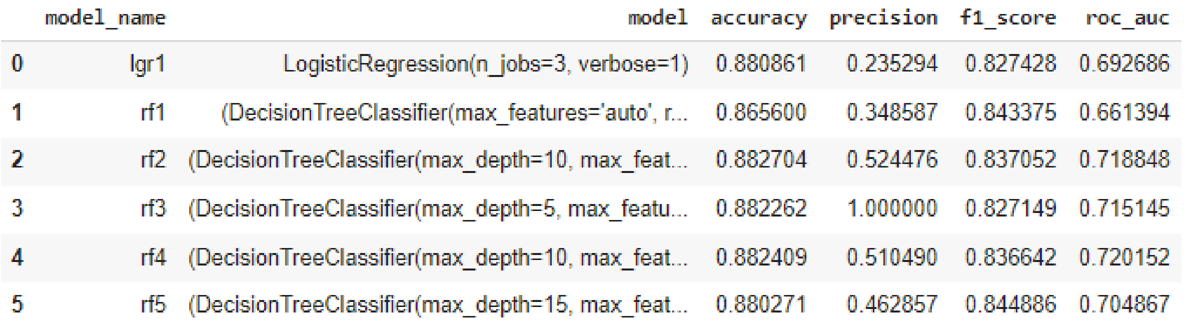
Alternatively a grid search based cross validation was also performed on the model to select/validate the optimal parameters.

Results

As this is a **binary classification problem** with a **class imbalance**, model **accuracy will not be considered** as an evaluation metric. Only the (a) model precision, (b) f1 score and the (c) ROC AUC (Area under the ROC Curve) metrics are considered for evaluating the best model.

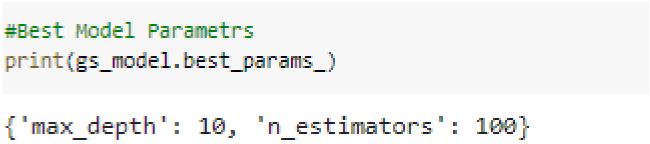
Evaluation

From manual hyperparameter tuning, the evaluation table was obtained.



	model_name	model	accuracy	precision	f1_score	roc_auc
0	lgr1	LogisticRegression(n_jobs=3, verbose=1)	0.880861	0.235294	0.827428	0.692686
1	rf1	(DecisionTreeClassifier(max_features='auto', r...	0.865600	0.348587	0.843375	0.661394
2	rf2	(DecisionTreeClassifier(max_depth=10, max_feat...	0.882704	0.524476	0.837052	0.718848
3	rf3	(DecisionTreeClassifier(max_depth=5, max_featu...	0.882262	1.000000	0.827149	0.715145
4	rf4	(DecisionTreeClassifier(max_depth=10, max_feat...	0.882409	0.510490	0.836642	0.720152
5	rf5	(DecisionTreeClassifier(max_depth=15, max_feat...	0.880271	0.462857	0.844886	0.704867

By cross validation with grid search it was revealed that following to be the optimal parameters for the model



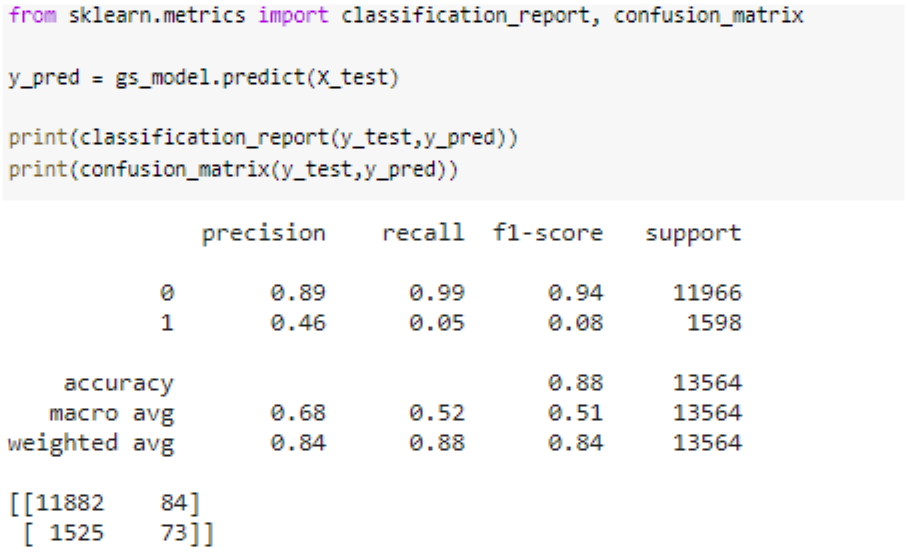
```
#Best Model Parametrs
print(gs_model.best_params_)
```

```
{'max_depth': 10, 'n_estimators': 100}
```

From above analysis the **Random Forest Classifier - 'rf2' is selected as the best model.**

Confusion Matrix

Confusion matrix for the selected model is listed below. An analysis of the confusion matrix reveals that even though the model performs well on the majority class, it performs rather poorly on the minority class, which is the important class ("yes" to a term deposit subscription) with regards to this classification problem.



```python
from sklearn.metrics import classification_report, confusion_matrix

y_pred = gs_model.predict(X_test)

print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.99      0.94     11966
           1       0.46      0.05      0.08      1598

    accuracy                           0.88     13564
   macro avg       0.68      0.52      0.51     13564
weighted avg       0.84      0.88      0.84     13564

[[11882    84]
 [ 1525    73]]
```

Note : Due to the lack of responses captured for the minority class, it was not possible to create a calibration table and a ROC curve.

Explaining the machine learning model

An attempt has been made to explain a sample out of the model with LIME (Local Interpretable Model-Agnostic Explanations and SHAP (SHapley Additive exPlanations), the details of which are included with the notebook.

Model Deployment

As part of this exercise following components were developed which can be used to create **an inference pipeline** for the purpose of model deployment

1. Pre-processing function
2. Machine Learning model (which is saved using "Joblib" and "pickle")
3. Predict / Scoring function
4. Post processing function

Additionally the following component can be further developed to create an end to end machine learning solution.

1. A user interface to get input data
2. API that takes in user inputs and invoke the inference pipeline (listed above)

3. An output user interface that displays the output from the inference pipeline. The output can also be further improved by analysing the output with LIME or SHAP and giving end users the reasons / factors that contributed to the predicted output.

The solution can be deployed either locally or on cloud infrastructure, either on virtualized or containerized hardware depending on the particular client requirements.

Conclusion

This problem of predicting subscriptions to bank term deposits based on data from marketing campaigns, is a binary classification problem with a class imbalance. An attempt was made with exploratory data analytics to select the most appropriate features for the model, identify null values and outliers in data, normalize / standard data where necessary, as needed for the selected machine learning classifier. This being a class imbalance problem, Random Forest classifier was chosen as the most appropriate, and the model was trained on pre-processed data for the optimal model parameters (hyper parameters). An attempt was also made to re-sample the data to treat the class imbalance. As part of this exercise a model training pipeline and an inferencing pipeline were also developed, and sample outputs were explained /interpreted using LIME and SHAP libraries.

Discussion

An analysis of the model revealed that it performs well on the majority class, but rather poorly on the minority class, which is the most important class with regards to this classification problem. A further attempt at feature engineering and treating the data for its imbalances may be further improvements that can be done to improve the model performance.