Case Study on Banking Data Set using Decision Tree

**Data Set information**

**Source:**

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

**Description:**

The data is from 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 'no' for subscription.

The data set contains the bank-additional-full.csv with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010), very close to the data analyzed in [Moro et al., 2014]. You can download the data set from the following link:

<https://s3.amazonaws.com/acadgildsite/wordpress_images/datasets/bank/bank-additional-full.csv>

**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')

**Problem Statement:** The data is from direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit.

## Read the data and import all necessary libraries

1. list all columns (for reference)
2. convert the response to numeric values and store as a new column
3. **Comment on Features**

*## 1. age*

*## 2 .job*

*## 3. Default*

*## 4. contact*

*## 5. month*

*## 6. duration*

*## 7.1. previous*

*## 7.2. poutcome*

*## 8. euribor3m*

## Model building

1. create X dataframe having 'default', 'contact', 'previous', 'euribor3m' and including 13 dummy columns
2. evaluate the decision tree model by splitting into train and test sets
3. calculate cross-validated AUC
4. Store the predicted data in 'predicted' array
5. generate evaluation metrics-
6. Print out the confusion matrix
7. Print out the classification report, and check the f1 score
8. Find out the mean cross validation score/accuracy of the fitted model, use 5 cv steps
9. **Model Visualisation**
10. import numpy as np, pandas as pd, matplotlib.pyplot as plt, pydotplus
11. from sklearn import tree, metrics, model\_selection, preprocessing