ML PROJECT

BANK MARKETING

Student:

Valerio Botta

FINAL WRITEUP

Abstract

- **Problem**: Predict if the client will subscribe (yes/no) a term deposit
- **Dataset**: The data is related with direct marketing campaigns of a Portuguese banking institution

Introduction

The dataset was updated in 2014 but the issue is still relevan: the sell of bank/financial product via telephone. It is a common way to sell products. The analysis of this dataset has the goal to create a program that could predict a sale and validate the results.
The daset ha a multivariate characteristics using 41188 instances of real type data. It has twenty attributes plus the target one.

Dataset and Features

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, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.
Dataset url: https://archive-beta.ics.uci.edu/ml/datasets/bank+marketing
My dataset has named “bank-additional-full.csv” with 41188 examples and 20 features (plus the target one).

Dataset structure

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")

Columns that I could use and relevat to the analysis are: age, job, marital, default, housing, loan, campaign, previous, poutcome, emp-var.rate, cons.price.idx, cons.conf.idx.

The last column ("y") is the target to predict. It's a simple binary choice: yes or no based on possibile subscribtion of the term deposit.

Methods

ML Algorithms to perform

Algorithms to perform:

- Logistic regression
- Neural network for classification
- Decision tree classifier

Logistic regression

The first operation split the dataset into test and train: 70% for train and 30% for test. Then, after the settings of *Kfold* (10 splits) I execute the *LogisticRegression* method using the *sklearn* library. After a few tests I decided to leave the default value of the algorithm except the *max_iter* parameter (1000 instead of 100). After I tried the cross-validation score using that model and the *Kfold* split, the fit method and the predict to obtain the value of the target column.

At this point I perform the accuracy of cross-validation, the accuracy of the predicted column, the confusion matrix and the roc curve to validate the results.

Neural network for classification

The process is the same of the Logistic Regression. I used the library *MLPClassifier* of *sklearn* library.
Parameters:

- *activation='logistic'*: activation function for the hidden layer, logistic sigmoid function f(x)=1/(1+exp(-x));
- *alpha=1e-3*: the L2 regularization terms;
- *max_iter=1000*: max number of the iteration until converge;
- *random_state=1*: random number generation for weights and bias initialization.

Decision Tree Classifier

The process is the same of the Logistic Regression. I used the library *DecisionTreeClassifier* of *sklearn* library.
Parameters:

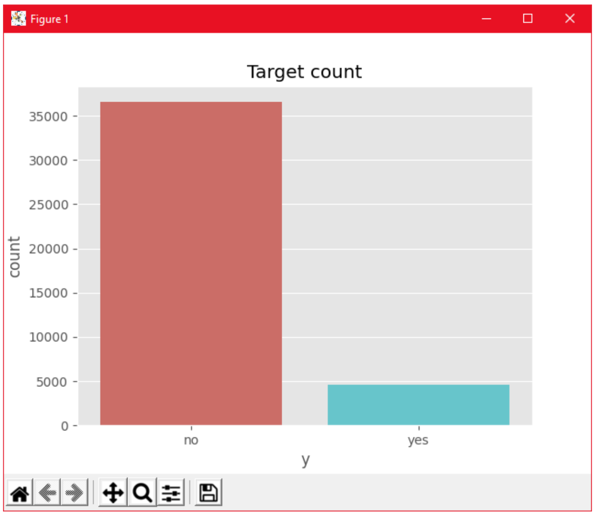
- *criterion='entropy'*: the function to measure the quality of a split
- *max_depth=8*: max depth of the tree;
- *min_samples_leaf=2*: minimum number of sample required to be at a leaf node;
- *random_state=42*: random number generation for weights and bias initialization;
- *max_leaf_nodes=52*: max number of leaf nodes.

Experiments and Results

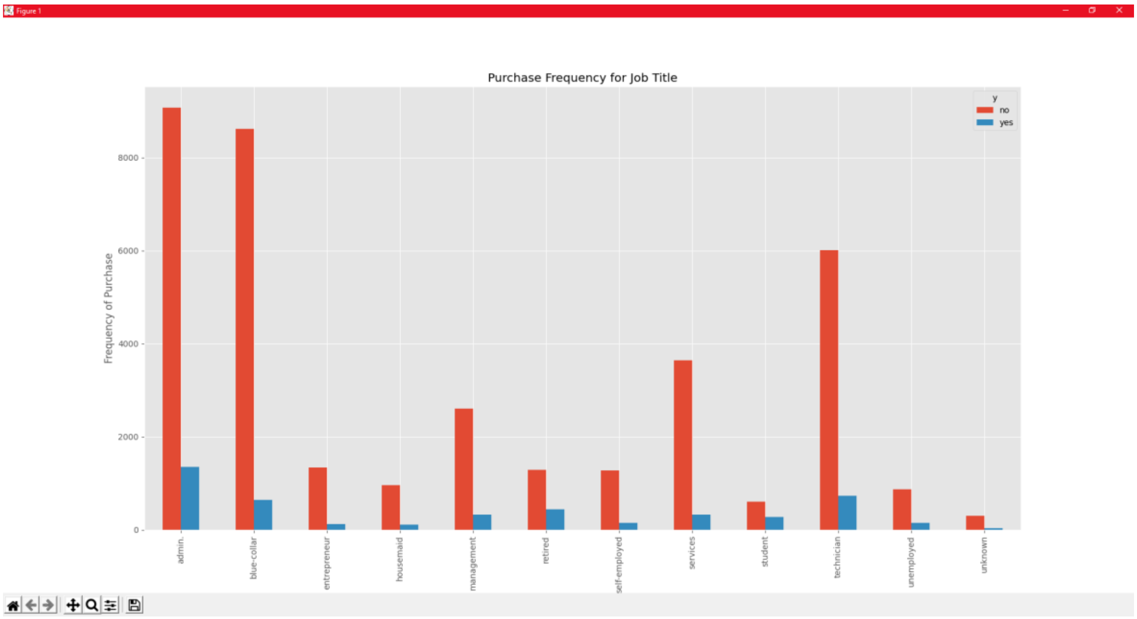
Preliminary experiments

I create the cleaning_dataser.py to check and clean the dataset. I found some problems that need to be fixed. The first function replace the symbol “ with nothing because every cell had that char, this create a problem during the read of csv file. Than, for the same reason, I replaced the ; with ,. Another replace involve the name of the index because the comma create problem, than I replace it with _ (for example: emp.var.rate -> emp_var_rate). The second function check the missing value and null value.
The last two things are encoder and drop. With encoder I replace the string with an encoder label, with drop I remove all the columns useless. At the end I saved the dataset into the “dataset.csv”.
Now I started some analysis:

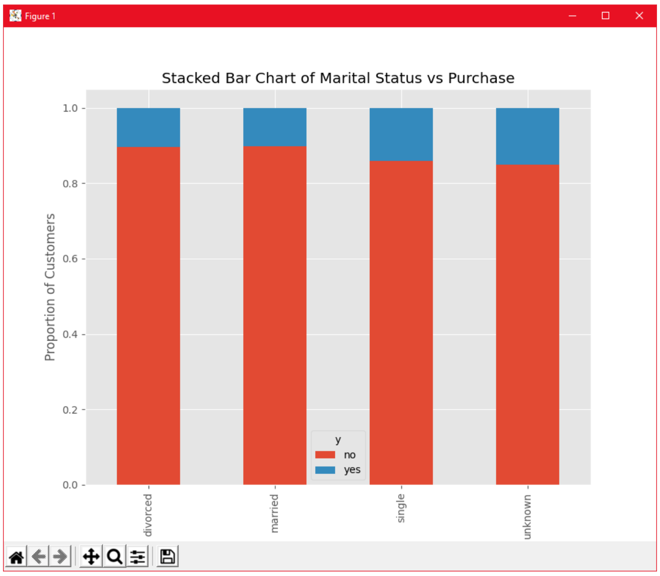
- Target count:
  no 36548 Percentage of no subscription is 88.73458288821988
  yes 4640 Percentage of subscription 11.265417111780131



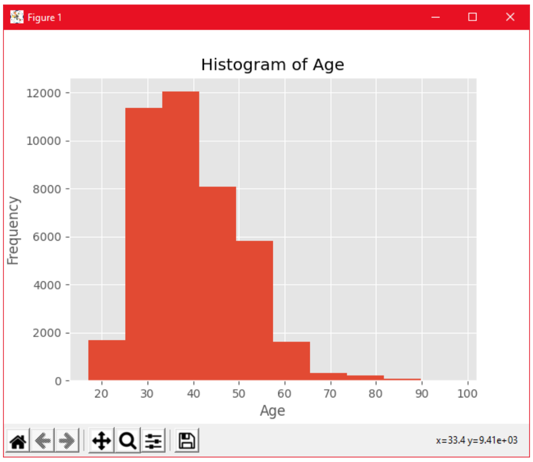
- Purchase frequency for job title, the frequency depends a great deal on the job title.



- Marital status: does not seem a strong predictor for the outcome variable.



- Most of the customers of the bank in this dataset are in the age range of 30–40.



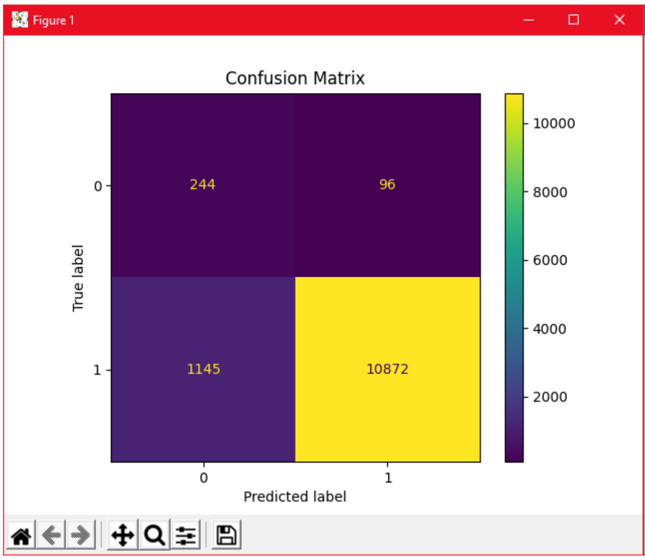
Results

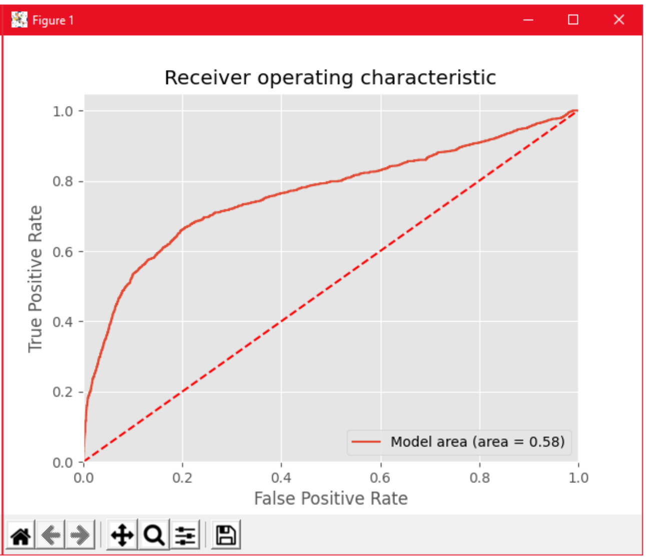
Logistic regression

Accuracy of K-fold validation: 89.88298165987328

Accuracy: 89.9571093307437

Execution time (second): 0:00:16.133801



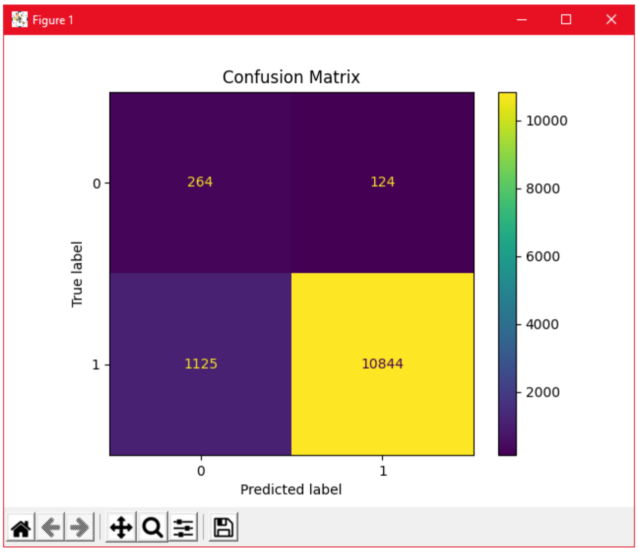


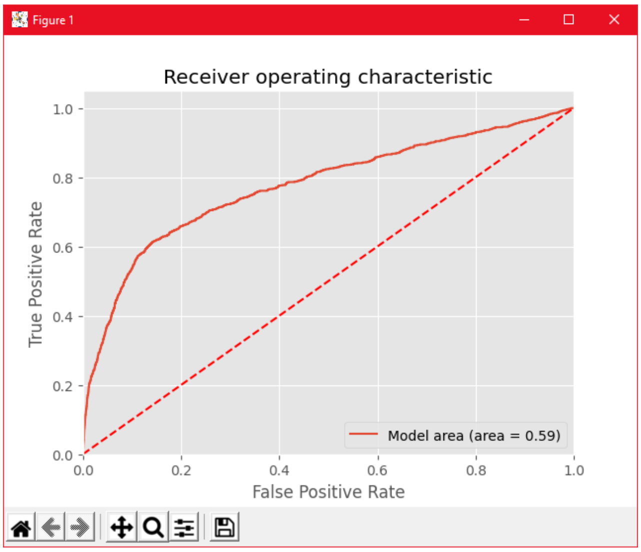
Neural network for classification

Accuracy of K-fold validation: 89.85142531777718

Accuracy: 89.89236869790402

Execution time (second): 0:01:46.224695



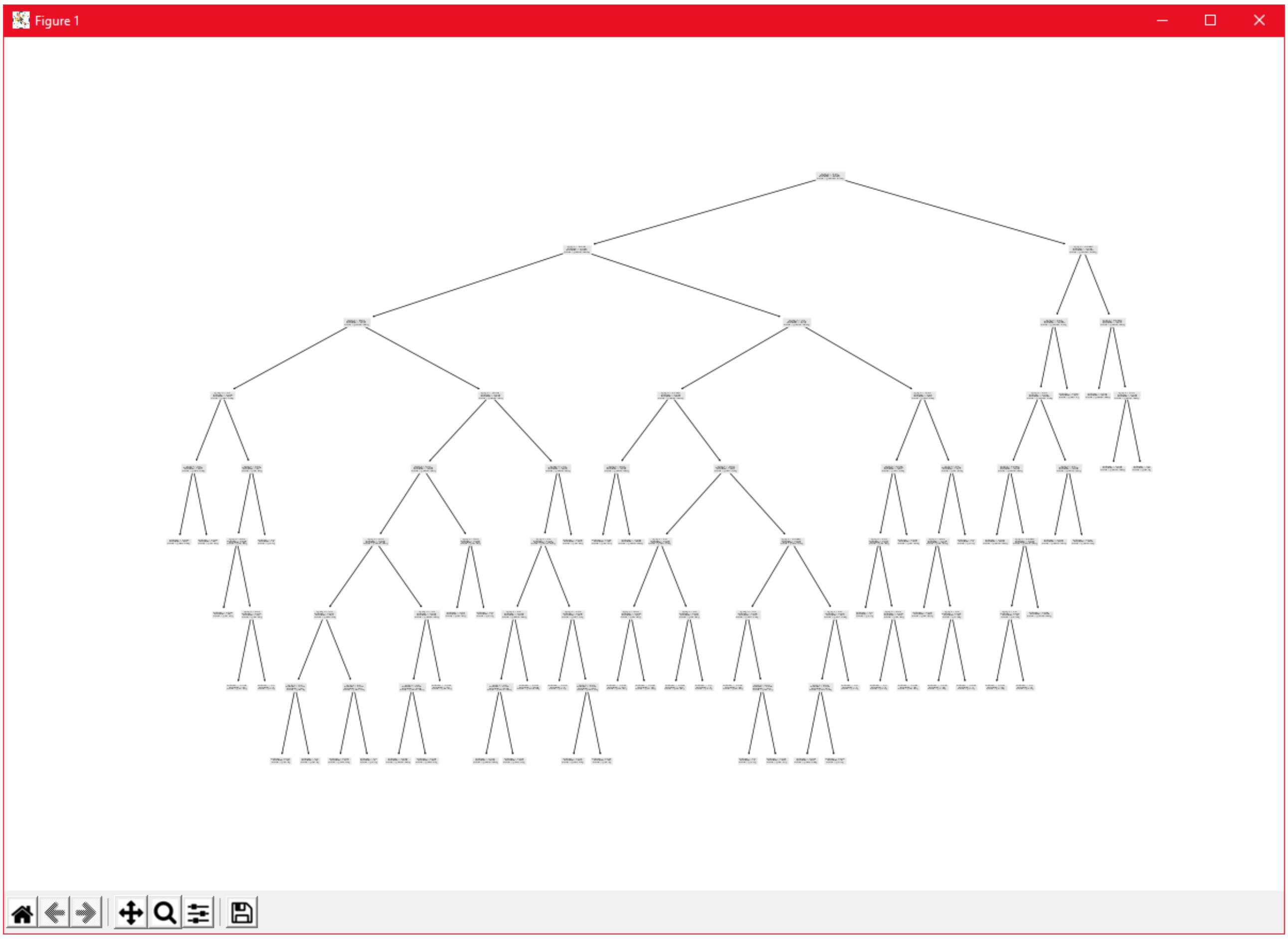


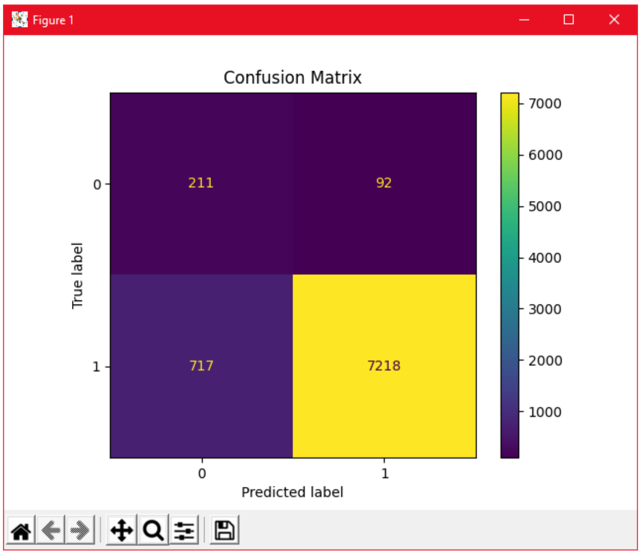
Decision Tree Classifier

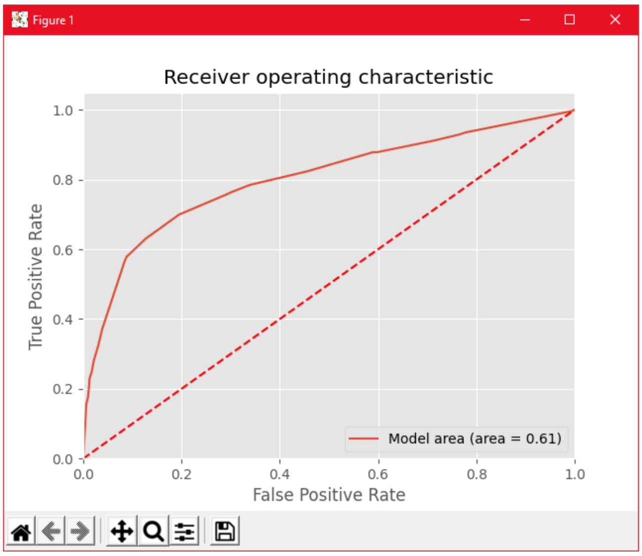
Accuracy of K-fold validation: 89.88540943360475

Accuracy: 90.17965525613013

Execution time (second): 0:00:09.367204







Conclusion

The results show that the three algorithms are similar.

The K-fold accuracy is pretty the same, equal to 89.9% (rounding up).

The accuracy of the fit methods changes a bit. The worst is the neural network algorithm with 89.89% of accuracy, the better is the decision tree algorithm with 90.19%.

The same is the ROC curve area: a bit better in decision tree, a little worst in logistic regression.

An important difference is showed thanks to the confusion matrix. The decision tree algorithm is the worst because recognize less true positive and true negative values than the other two. The logistic regression and the neural network algorithms are similar.

The last important parameter is the execution time: the decision tree is the faster but gives not the best results, the logistic regression and the neural network gives the best results but the second one is very slow than the first one. The logistic regression requires, in this case and executed on my machine, a bit more of 16 seconds to finish, instead the neural network algorithm requires more than 1 minute and half.

At the end the best algorithm in this case is the logistic regression.