Evaluation of Term Deposit Prediction with Supervised Machine Learning

Pol Monné Parera
Polmonne7@gmail.com

Table of Contents

Data Information

Source

The dataset was extracted from the UC Irvine Machine Learning Repository (Moro, Rita, & Cortez, 2012).

The data was donated to the website on the 13th of February of 2012. The website states that the data was extracted from direct marketing campaigns of a Portuguese banking institution between May 2008 and November 2010.

Description

The data was supposedly collected from a direct marketing campaign (phone calls) that a Portuguese banking institution conducted between 2008 and 2010. It contains a variety of characteristics of the consumer contacted and whether the client subscribed to a term deposit after the approach.

The repository contains a total of 4 datasets:

1. **bank-additional-full.csv** – This one has been used for the development of the project as it contains the largest amount of data, is the newest version and it is the closest one to the one the authors used to conduct a research paper.
2. bank-additional.csv – It contains a 10% of the inputs from the previous dataset.
3. bank-full.csv – Older version with less inputs of the first dataset.
4. bank.csv – It contains 10% of the inputs of the ‘bank-full.csv’ dataset

Data’s original usage

The original usage of the dataset was to predict weather a client would subscribe to a term deposit (binary classification: yes/no). This predictive capability would be of great relevance to the banking institution as it would provide them with the tools to personalize and direct their marketing campaigns to consumers more inclined to purchase their products.

Size

The raw data contains a total of 21 columns and a total of 41188 instances. This outputs a data schema of a 21x41188 matrix.

Target

The models implemented aim to predict weather a costumer is going to subscribe to a term deposit. This result is represented in the last column of the dataset **y**, which has a binary result of yes or no.

Data Attributes

Column Name	Data Type	Example Value
y	Binary	no
age	Integer	56
job	Categorical	housemaid
marital	Categorical	married
education	Categorical	basic.4y
default	Binary	no
housing	Binary	no
loan	Binary	no
contact	Categorical	telephone
month	Date	may
day_of_week	Date	mon
duration	Integer	261
campaign	Integer	1
pdays	Integer	999
previous	Integer	0
poutcome	Categorical	nonexistent
emp.var.rate	Float	1.1
cons.price.idx	Float	93.994
cons.conf.idx	Float	-36.4
euribor3m	Float	4.857
nr.employed	Integer	5191

Exploratory Analysis

Missing values

The dataset did not contain any missing values in none of its features. Nevertheless, it is important to note that it did contain quite a few 'unknown' values in multiple categories. Instead of removing these instances it was opted to keep them as they could share valuable information on the costumer being analyzed (e.g. some clients might be less willing to share financial info). Moreover, the fact that a value is unknown might correlate with the target variable y (e.g. clients who avoid disclosing loan status might be less likely to subscribe to a term deposit).

The chosen approach was to treat unknown as a separate category instead of dropping it. When applying one-hot encoding for the categorical data each possible category became a separate binary column, including 'unknown'. This would allow the model to learn a relationship between 'unknown' and the outcome 'y'.

Outliers

To detect outliers, the IQR and z-score methods were applied, and each numerical feature was analyzed individually. The variable *duration* was removed due to data leakage, while legitimate rare cases in *campaign* and *previous* were retained but capped or transformed to reduce skew. Furthermore, the special code 999 in *pdays* represented a 96.32% of the total values in it and greatly skewed the results so it was also dropped entirely.

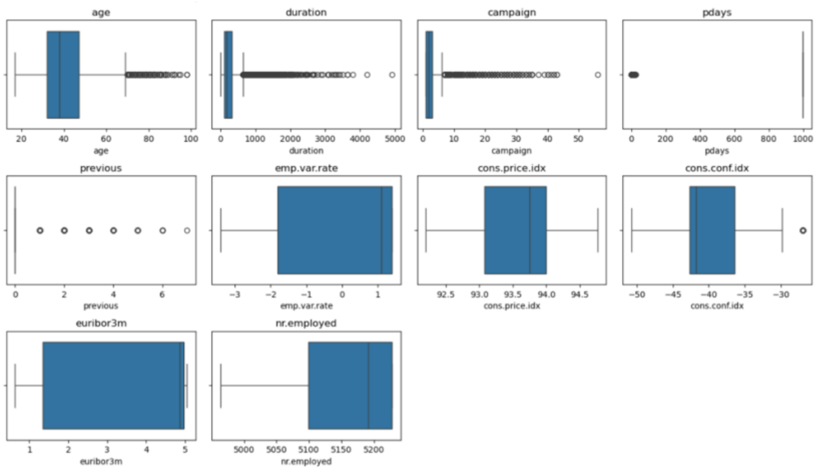
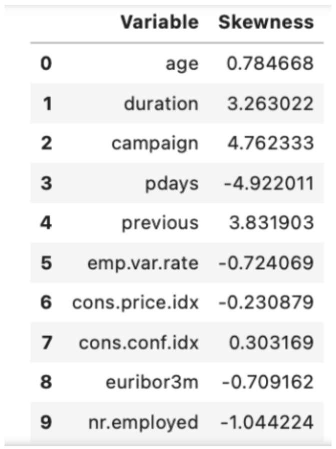


Figure 1 Data Distribution of numerical variables

Distributions

During the exploratory analysis of the data, it was detected that a few features had a skewed distribution. To be precise, the detected skewed distributions were *campaign*, *previous* and *nr.employed*. To avoid further results being affected by this skewness of data a normalization of these variables has been applied in the preprocessing steps of the project.



	Variable	Skewness
0	age	0.784668
1	duration	3.263022
2	campaign	4.762333
3	pdays	-4.922011
4	previous	3.831903
5	emp.var.rate	-0.724069
6	cons.price.idx	-0.230879
7	cons.conf.idx	0.303169
8	euribor3m	-0.709162
9	nr.employed	-1.044224

Figure 2 Skewness of numerical features

It must be considered that from the previous table the features *duration* and *pdays* must be ignored as they were dropped in previous preprocessing steps.

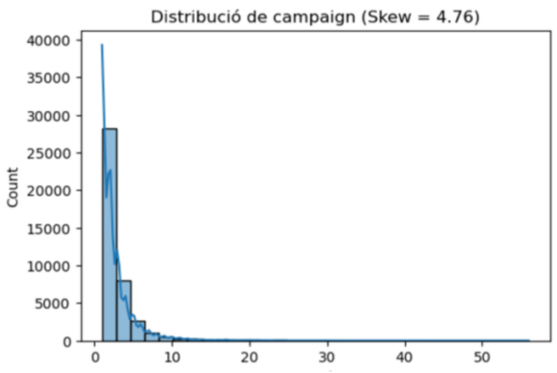


Figure 3 Example of the campaign feature distribution

Correlations between features

To discover correlations between variables a correlation heatmap containing all numerical features has been conducted. This heatmap also contains the transformed variable (converted to binary) 'y', as it is relevant and interesting to understand how the target value correlates to the general features.

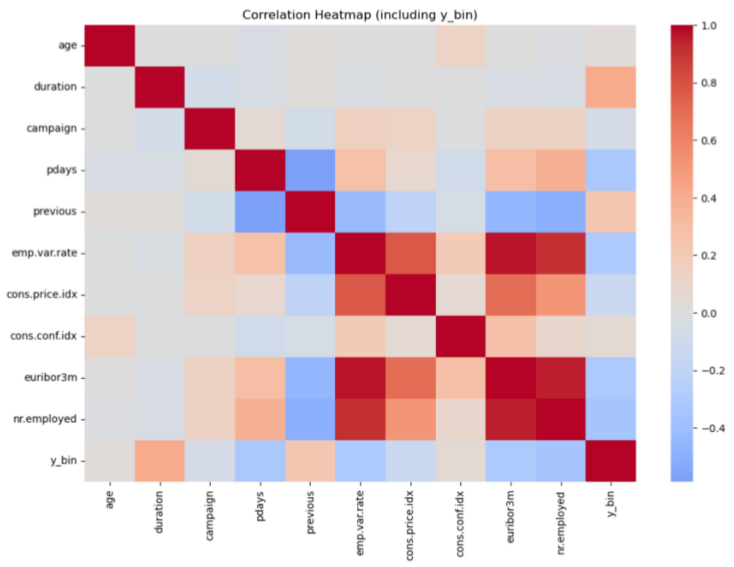


Figure 4 Correlation Heatmap of variables

As it was previously discovered, the duration feature can be seen to closely correlate to the chance of success of the call, this has been considered to be a "leakage" feature as it is something that would not normally be known before the contact and is giving away the target value.

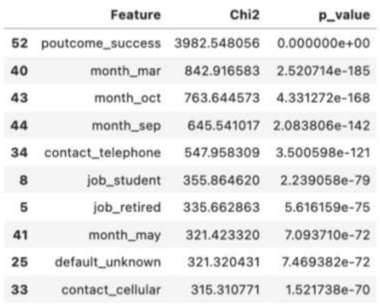
It is also interesting to see how *age, cons.conf.idx* and *campaign* don't seem to have much of a correlation and therefore probably are independent from the result and unnecessary to study.

Important features

To determine which were the most relevant numerical features to study the previous correlation matrix has been examined in detail to extract those variables that kept a closer relation with the target feature, as these were the ones with the highest predictive power. After a close inspection it was determined that the following features were the most relevant:

- previous
- emp.var.rate
- cons.price.idx
- euribor3m
- nr.employed

Considering that the dataset also consists of categorical data, the chi-square test method has been applied to detect the relevance of these. With this method it has been determined that the top 10 most relevant categorical features are the ones in the following table:



	Feature	Chi2	p_value
52	poutcome_success	3982.548056	0.000000e+00
40	month_mar	842.916583	2.520714e-185
43	month_oct	763.644573	4.331272e-168
44	month_sep	645.541017	2.083806e-142
34	contact_telephone	547.958309	3.500598e-121
8	job_student	355.864620	2.239058e-79
5	job_retired	335.662863	5.616159e-75
41	month_may	321.423320	7.093710e-72
25	default_unknown	321.320431	7.469382e-72
33	contact_cellular	315.310771	1.521738e-70

Figure 5 Top 10 most relevant categorical features by Chi-square method

With the aid of a bar plot it was possible to observe that up until the top 14 the features still held relevant information, and it was concluded that those were the ones to be used.

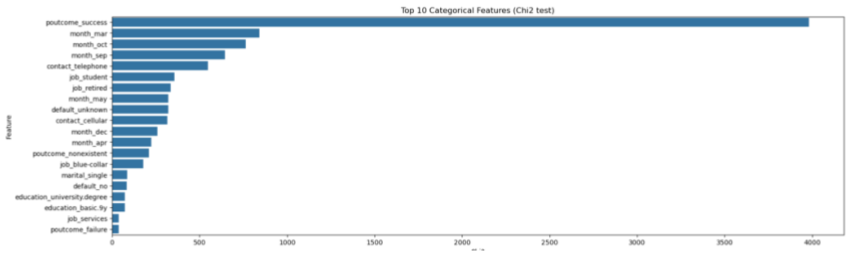


Figure 6 Bar plot of the top 20 most relevant categorical features

It can be observed that the most relevant one is *poutcome_success*, this is a binary feature characterizing if in a previous marketing approach the client accepted the offer and is a clear correlation on how probable it is to accept now. The other features represent a variety of instance characteristics that must be considered in the following steps.

Classification Methodology

Preprocessing steps

After checking all features, it was detected that there were no missing values in the data and that instead, there were 'unknown' values. These were identified and since they gathered potentially relevant information were not deleted and rather converted to a column during the One-Hot Encoding process latter applied.

No outliers were removed, but their analysis allowed the detection of a highly skewed variable (96% of its data had the same value) that could potentially later affect the results and therefore was removed (*pdays*).

On further exploratory analysis it was detected that some numerical data presented some skewness and to correct this finding the features *campaign*, *previous* and *nr.employed* were transformed to follow a more normal distribution.

Finally, correlation analysis and chi-square test were applied to identify redundant or less informative features to avoid using them in further analysis.

Classification Methods used

For the analysis of the data four classification methods were implemented with different but similar accuracy results. These methods were the following:

- Decision Tree – This method was used for its simplicity and comprehensive results and after examining various possible combinations the following hyperparameters were adjusted:
    - Max depth = 5
    - Min samples split = 10
    - Min samples leaf = 5
    - Criterion = gini

- Weighted Decision Tree – Following the results from the previously implemented base decision tree, it was observed that the model performed better at predicting negative outcomes (non-subscriptions) than positive ones. However, for the business objective recall of the positive class was prioritized.

  The final configuration was:
    - Max depth = 5
    - Min samples split = 2
    - Min samples lift = 5
    - Criterion = gini
    - Class weight = {0:1, 1:3} (errors in predicting positive outcomes 3x more penalizing than negatives)

This adjustment improved the models recall for the positive class, even at the cost of some additional false positives.

- Random Forest – This algorithm was implemented aiming to improve recall and for its robustness against overfitting. After a grid search with 10-fold cross-validation the optimal hyperparameters were found to be:
    - Max depth = 10
    - Min sample split = 2
    - Min sample leaf = 2
    - Number of estimators = 200
    - Max features = log2

- K-Nearest Neighbors – The KNN algorithm was implemented to evaluate a non-tree like model and compare its results. After tuning the hyperparameters via grid search, the best-performing configuration was found to be:
    - N_neighbors = 15
    - Weights = uniform
    - Metric = euclidean

Results

Base Decision Tree

This model had a good performance overall, with a mean cross-validation accuracy of 0.900516 and a precision of 0.682857. Even with those high values it lacked recall and therefore the f score was not great either, the recall value was 0.257543 and the f score was 0.187011.

The model was great at predicting false values but not so good at predicting positive ones and the resulting performance was not as good as desired/expected. If the bank contacted all its consumers, it would have a 11.27% of success and with this model, by contacting just the predicted positives, it would have an increased success rate of a 68% (68% of the contacted people would have subscribed). On paper this is a great increase as the money spent is better targeted and the profit of the campaign will be certainly better, but it must be considered that it is missing out on 74.25% of the costumer's that would contract their services.

Therefore, even though the model represents an increase in performance and a great economical approach in comparison to contacting all its customers, it still can be improved in an intent to obtain a higher amount of true positives detected (even at the cost of contacting more false positives).

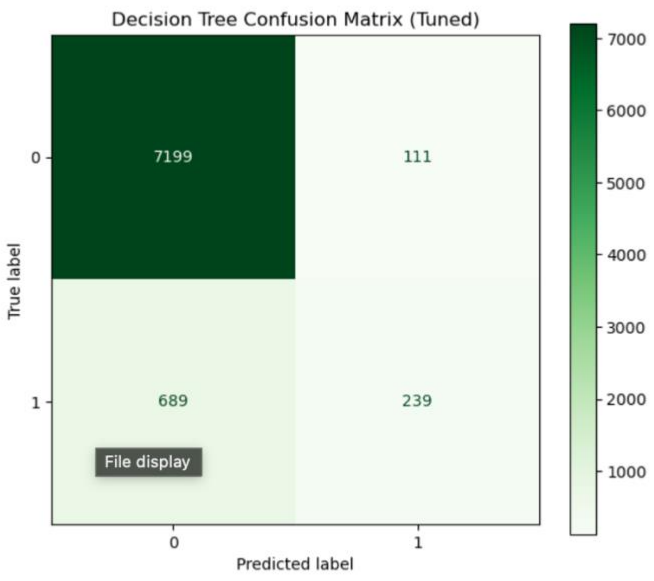


Figure 7 Confusion matrix of the decision tree model

Weighted Decision Tree

Seeing the improvements the previous model offered and, even more, the holes in its implementation and results. The approach followed intended to try and get a higher ratio of true positives at the cost of some more false negatives by applying a higher weight to the positive outcome's relevance.

With this approach, and applying a weight of 3 to the positives, a mean cross-validation accuracy of 0.8806 and a precision of 0.474729 was obtained. The final recall was an increased 0.56681 and the f-score also increased to 0.25835.

With this model the results were objectively better for the purpose at hands, it had a success rate of 47.47% predicting positive outcomes, while missing out on 43.32% of the costumer's that would contract their services. Even though the precision decreased it is important to detect that the costumer's that contracted the bank's services more than doubled in size.

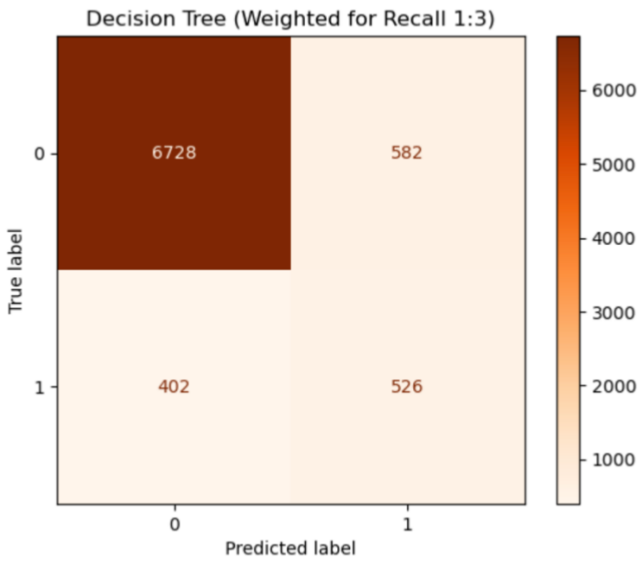


Figure 8 Confusion matrix of the Weighted Decision Tree (Weight = 3)

Another model tested consisted of giving even more weight to the positive outcomes, to be precise a weight of 1 to 5 instead of 1 to 3. This model was finally not selected because there was no available data on the costs of operations of the bank for contacting each costumer. In other words, it was impossible to know at which point the bank stopped gaining a benefit from contacting a higher percentage of true positive outcomes at the cost of also contacting more false positive outcomes.

Nevertheless, with the modification of a higher weight the results obtained were a mean cross-validation accuracy of 0.8776 and a precision of 0.465928. The final recall was an increased 0.589439 and an f-score of 0.260228 in comparison to the other weighted tree.

This model also performed objectively better than a base decision tree for the purpose at hands, but it cannot be said with certainty that it was better than a less weighted decision tree for the lack of information previously stated. It had a success rate of 46.59% predicting positive outcomes, while missing out on just 41.06% of the costumer's that would contract their services.

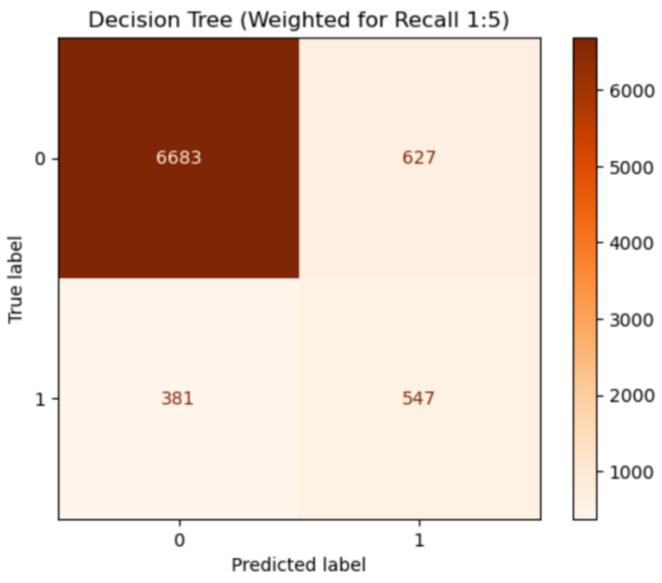


Figure 9 Confusion matrix of the Weighted Decision Tree (Weight = 5)

Random Forest

This model had a slightly worse performance to the weighted decision tree (weight = 3), its precision was a 0.496932. The recall value was 0.523707 and the f-score 0.254984. The obtained results are a clear improvement in comparison to a base decision tree and slightly worse than a weighted one, which still has a slightly better overall performance.

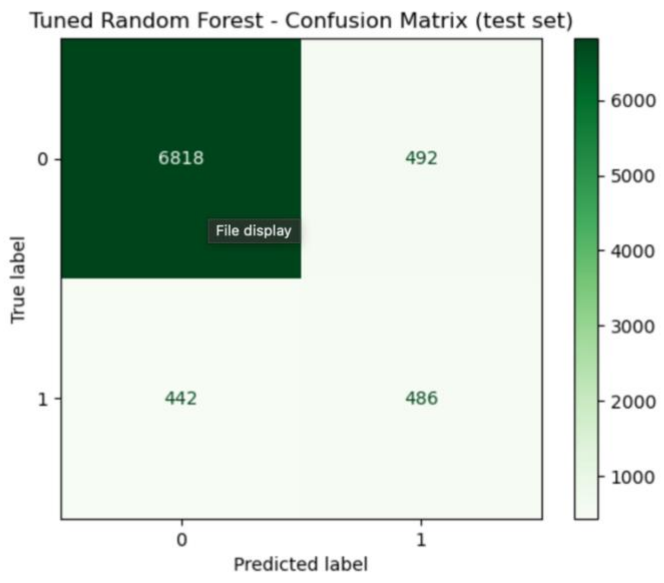


Figure 10 Confusion matrix of the Random Forest model

K-Nearest Neighbors

This algorithm was implemented to look at other non-tree-based models. The precision was 0.243288, the recall had a value of 0.634698 and the f-measure a 0.175873.

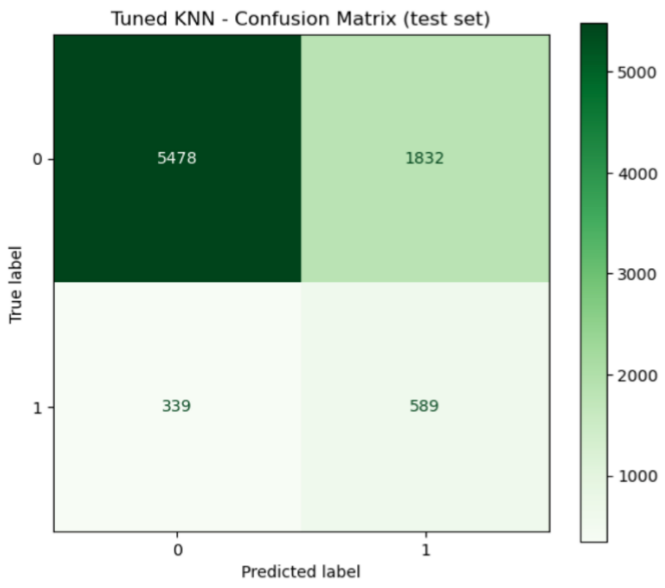


Figure 11 Confusion matrix of the KNN model

This model shows the highest recall for the positive class out of all the ones studied, providing with a predictive approach that with a precision of 25.33% only misses out on 36% of the positive instances. This model allows to contact approximately two thirds of the target audience, while avoiding contact with 70.61% of the total audience. This represents a clear improvement in comparison with the base marketing model, but without further information of costs cannot be classified as the best algorithm.

Bibliography

Moro, S., Rita, P., & Cortez, P. (2012, February 13). *Bank Marketing*. Retrieved from UC Irvine Machine Learning Repository: https://archive.ics.uci.edu/dataset/222/bank+marketing