Artificial Intelligence 2022/2023

Portuguese Banking Information Campaign

Assignment No. 2 Supervised Learning

L.EIC – BSc/First Degree in Informatics and Computing

Work Done by Group 11:

- Luis Filipe Pinto Cunha, up20170375@up.pt (33.3%)

- Inês de Magalhães Garcia, up202004810@up.pt (33.3%)

- Henrique Correia Vicente, up202005321@up.pt (33.3%)

1. Problem Definition

The goal of this project is to build a machine learning model that can predict whether a customer will subscribe to a bank term deposit based on a variety of features related to direct marketing campaigns conducted by a Portuguese banking institution. The dataset contains both numerical and categorical features, including information on the customer's demographic profile, bank balance, previous interactions with the bank, and details about the marketing campaign itself, such as the mode of communication and the outcome of previous promotional contacts. The output variable to be predicted is whether the customer subscribed to a term deposit or not. The model will be trained on a labeled dataset containing information on whether the customer subscribed to the term deposit or not and will be tested on a separate set of calls for which the subscription status needs to be predicted. The goal is to develop a model that can help the bank optimize its marketing strategies and increase the likelihood of acquiring new customers for term deposits.

2. Dataset

Two datasets with true information about a campaign were provided, both similar.

For this work we decided to stick with just the Excel file, named “Train.xlxs”.

The Dataset contains:

- 31647 Rows (Number of Entries)
- 18 Columns:
  - Numeric:
    1. **ID**: Identifier of the costumer.
    2. **Age**: Age of the costumer.
    3. **Balance**: The bank balance of the costumer.
    4. **Day_of_Moth**: The day of the month on which the communication take place.

5. **Duration**: States the duration of the call (in sec) regarding the campaign. This is an important feature as this attribute highly affects the target label (e.g., if duration=0 then y='no'). Thus, this input will be included for benchmark purposes and will be discarded if the intention is to have a realistic predictive model.
6. **Num_contacts_in_campaign**: Number of contacts performed during the campaign with the client for subscription.
7. **Days_since_prev_campaign_contact**: Gap (in number of days) between two contacts.
8. **Num_contacts_prev_campaign**: Number of contacts performed for promoting the campaign beforehand.

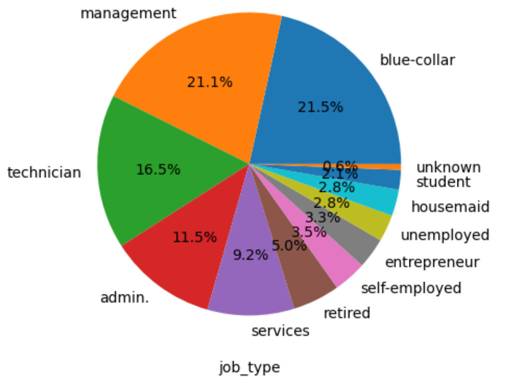
- Categorical:

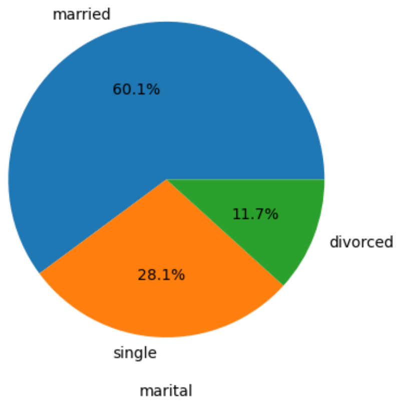
1. **Job_type**: What job the costumer has.
2. **Marital**: States the marital status of the customer.
3. **Education**: What education did the costumer achieved.
4. **Default:** Is the customer a defaulter or not?
5. **Housing_Loan**: Has a housing loan or not?
6. **Personal_Loan:** Has a personal loan or not?
7. **Communication_type**: Mode of communication during the campaign.
8. **Month**: Month in which contact with the customer took place for the campaign.
9. **Prev_campaign_outcome**: The outcome of promotional contact with the client beforehand for attending/interest in the campaign.

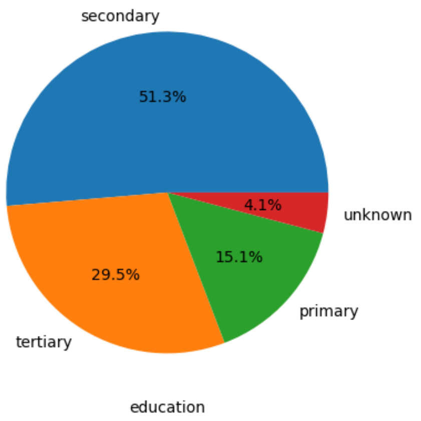
3. Approach

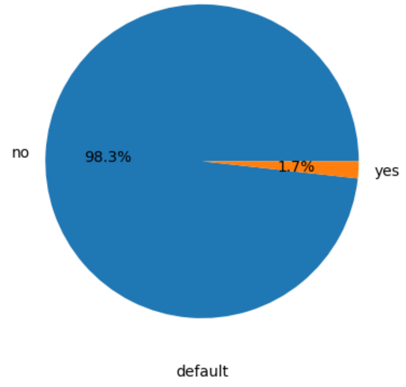
3.1 Data Analysis

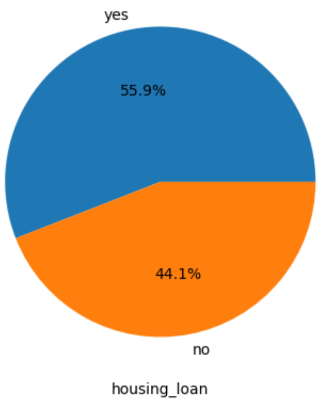
First, we made the graphics for the information that was made available without any changes:

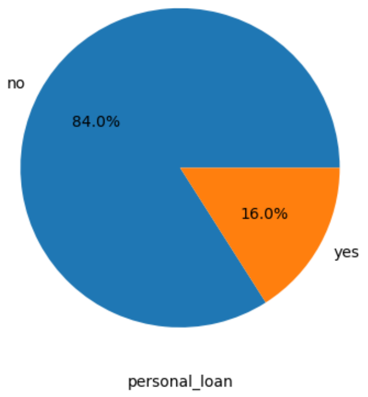


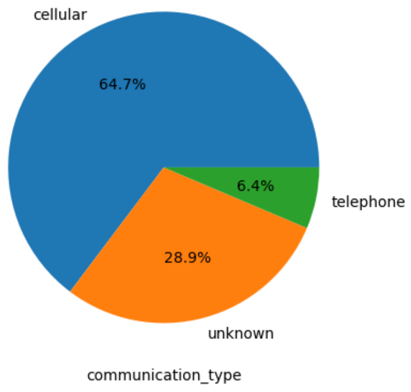


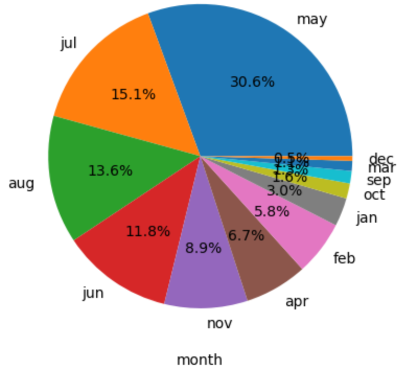


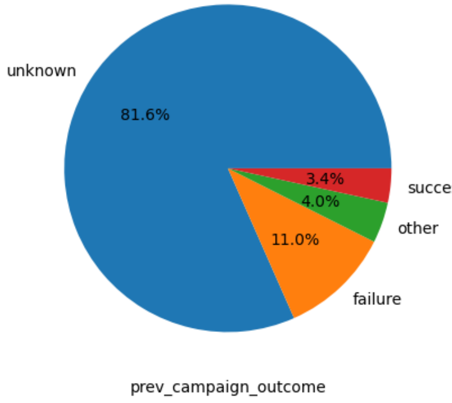




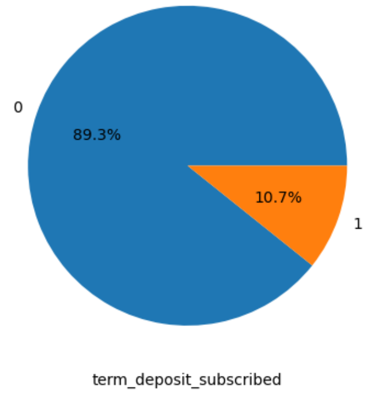








And the Target column, which is the purpose of the function:



Step1: We handle the missing values in the dataset by filling empty columns with default values chosen by us. The default values are defined for each column: customer_age_default, marital_default, balance_default, personal_default, last_contact_duration_default, num_contacts_in_campaign_default, and days_since_prev_campaign_contact_default.

The default values we decided on were:

marital_default = "widowed" - For obvious reasons, you are either single, or divorced, or married;

balance_default = data["balance"].median() - In order not to change the dynamics of the column too much and not have to delete entries;

last_contact_duration_default = round(data["last_contact_duration"].mean(), 0) - In order not to change the dynamics of the column too much and not have to delete entries;

num_contacts_in_campaign_default = data["num_contacts_in_campaign"].median() - In order not to change the dynamics of the column too much and not have to delete entries;

days_since_prev_campaign_contact_default = 0 - As entries were never subject to previous campagins, we put it 0 , since the last contact never existed

Step2: Rows with specific conditions are removed from the dataset using boolean indexing. Rows with 'job_type' as "unknown", 'personal_loan' as "unknown", 'customer_age' as "unknown", and 'prev_campaign_outcome' as "other" are dropped from the dataset.

At this stage, we thought it would be best to remove the rows that had:

-The job_type as unknown, as they were not mentioned in all known jobs.

-The personal_loan, should be easy answer, yes or no, if it were unknown, it wouldn't help at all.

-The costumer_age as unknown, because it is a very simple question that would have an even simpler answer and so we don't have fictitious information.

-And finally, the prev_campagain outcome because as mentioned above, it would be a easy answer, success or failure. We kept “unknown” for entries that were never the target of previous campaigns.

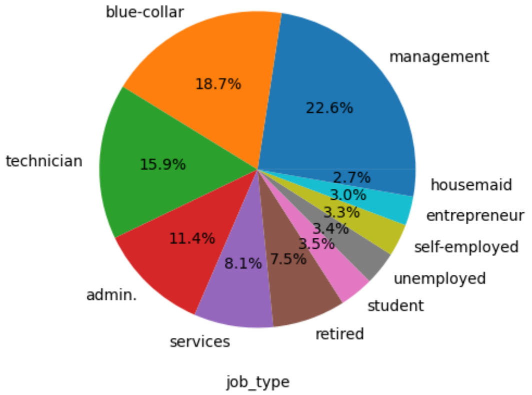
Step3: At the end , we balanced the number of entries, from the target column, "term_deposit_subscribed", we made the number of entries of 0s and 1s equal.

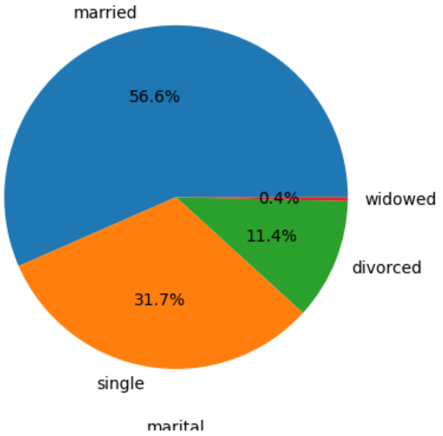
After our analysis, we are left with a total of:

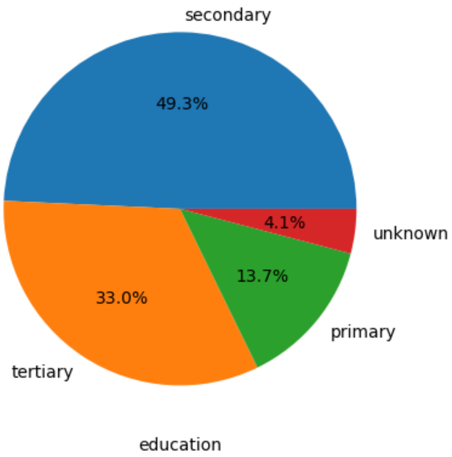
- 6226 rows

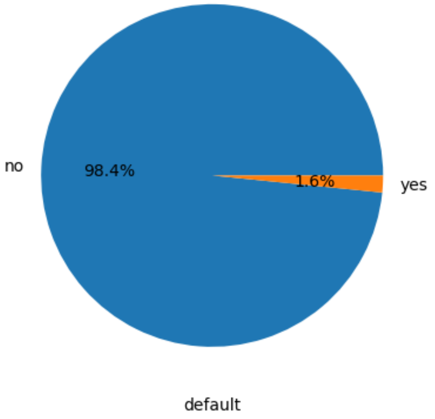
- 18 columns

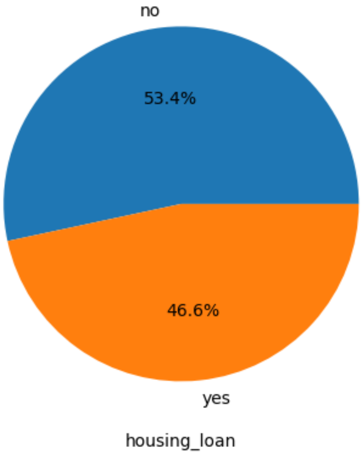
And the resulted graphs were:

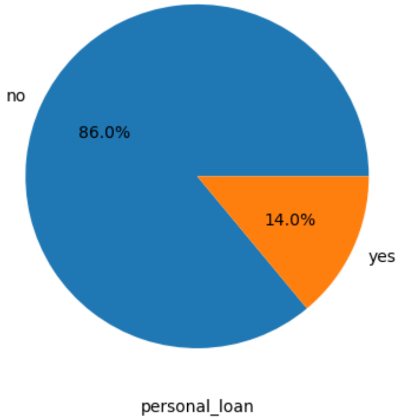


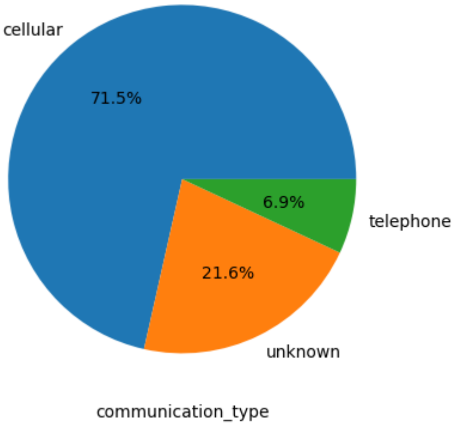


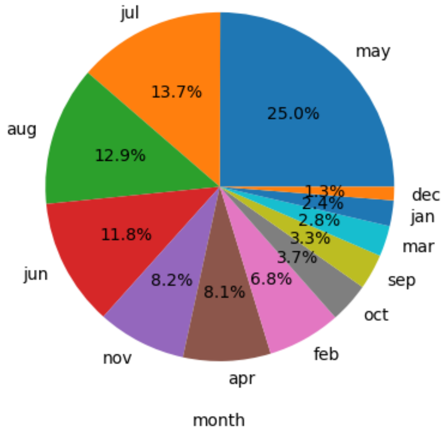


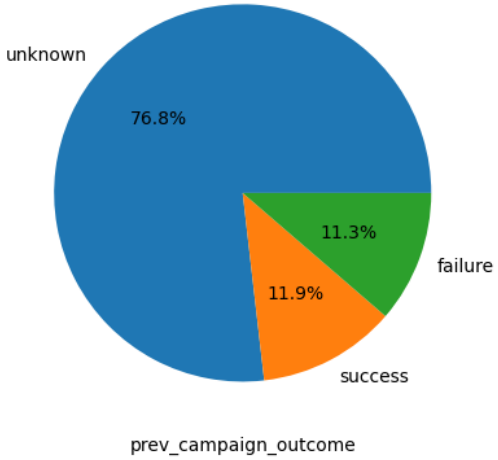


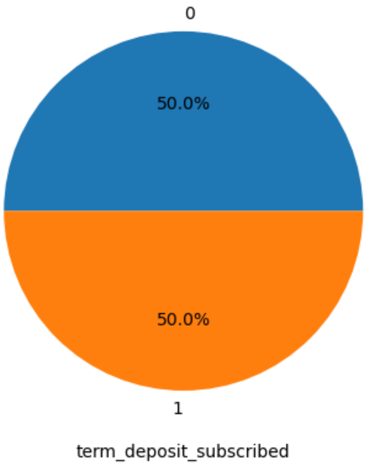












3.2 Algorithm Analysis

The algorithms used in this assignment were:

- **Support Vector Machines (SVM):**

SVM finds an optimal hyperplane that separates different classes by maximizing the margin between them.

It transforms the data into a higher-dimensional space using kernel functions to make non-linear separation possible.

SVM aims to find the hyperplane that has the largest distance to the nearest data points of each class.

The algorithm determines the support vectors, which are the data points closest to the hyperplane, to define the decision boundary.

SVM is based on the idea of finding the best balance between maximizing the margin and minimizing the misclassification.

-Decision Trees:

Decision Trees create a hierarchical structure of decisions based on features in the data.

The algorithm selects the most informative feature at each internal node and splits the data based on its values.

The splitting process continues recursively until a stopping criterion is met, such as reaching a maximum depth or achieving homogeneous subsets.

The resulting tree represents a series of if-else conditions leading to different outcomes or class labels at the leaf nodes.

Decision Trees can handle both numerical and categorical features and are easy to Interpret due to their intuitive structure.

-**Neural Networks**:

Neural Networks consist of interconnected layers of artificial neurons (perceptrons).

Each neuron applies a weighted sum of inputs, passes it through an activation function, and produces an output.

The network has an input layer that receives the features, one or more hidden layers for intermediate processing, and an output layer for the final prediction.

During training, the network adjusts the weights to minimize the difference between predicted and actual outputs using optimization techniques like backpropagation.

Neural Networks are capable of learning complex patterns and relationships in the data by iteratively updating the weights through the layers.

- **Random**:

Random was only used to compare performance, choose random between 0 and 1 in the target column, "term_deposit_subscribed".

The performance metrics used are:

-Accuracy: Measures the overall correctness of the model's predictions.

-Precision: Measures the proportion of correctly predicted positive instances out of the total instances predicted as positive.

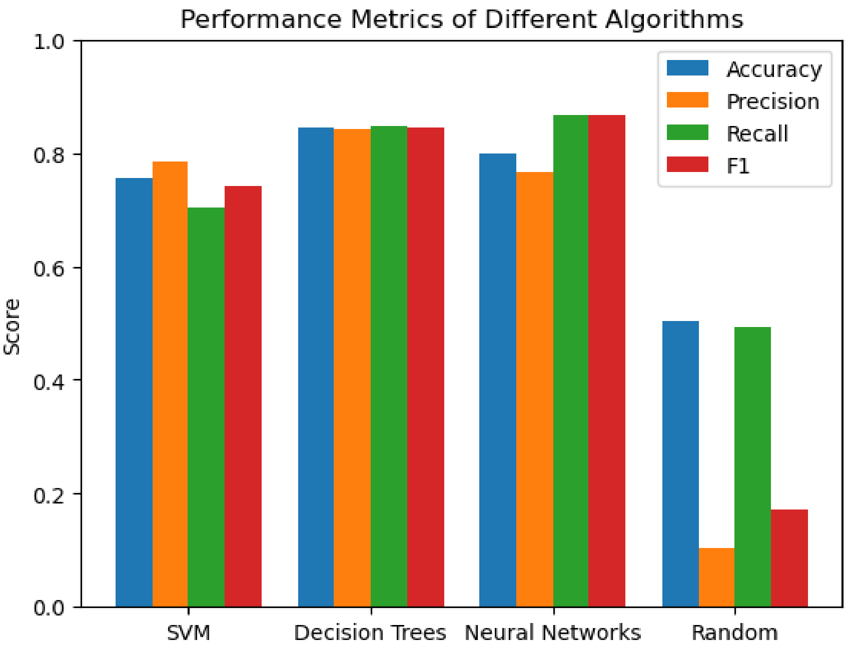
-Recall: Measures the proportion of correctly predicted positive instances out of the total actual positive instances.

-**F1:** The F1 score is the harmonic mean of precision and recall, providing a balanced measure of model performance.

4. Results

The results obtained for the performance metrics:

Metrics range from 0 to 1



Accuracy	0.755	0.844	0.8	0.505
Precision	0.785	0.842	0.767	0.104
Recall	0.703	0.848	0.866	0.492
F1	0.742	0.845	0.866	0.172
	SVM	**Decision Trees**	**Neural Networks**	**Random**

5. Conclusion

In this project, we have successfully applied machine learning models and algorithms to solve a supervised learning problem. The objective was to classify examples based on a predefined concept. We began by conducting an exploratory data analysis to gain insights into the dataset, including class distribution and attribute values.

We employed multiple supervised learning algorithms, including SVM, Decision Trees, Neural Networks, and a Random algorithm. Each algorithm was trained and tested using appropriate evaluation metrics such as accuracy, precision, recall, and F1 score. We also considered the time taken to train and test the models.

The results obtained from the evaluation metrics and performance analysis allowed us to compare the effectiveness of different algorithms. We observed that Decision Trees performed well in terms of accuracy, precision, recall, and F1 score. Neural Networks also showed promising results, although it required more computational resources for training.