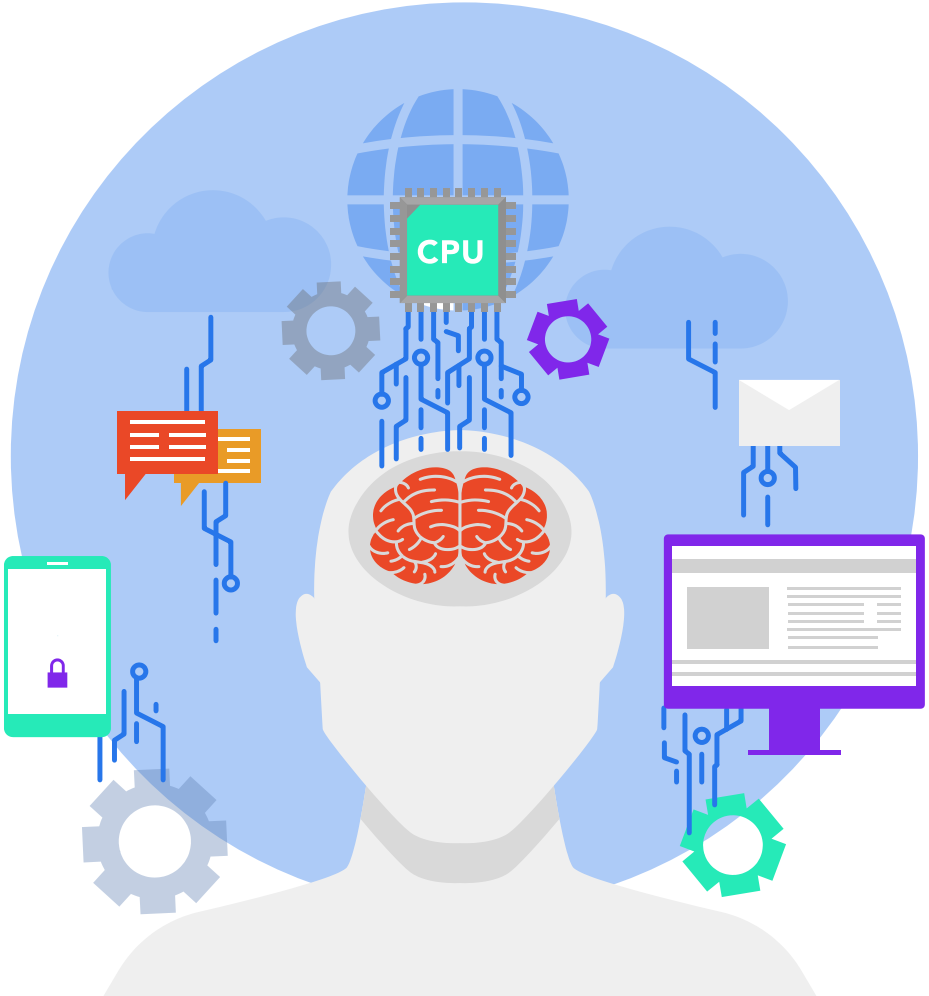


ARTIFICIAL INTELLIGENCE

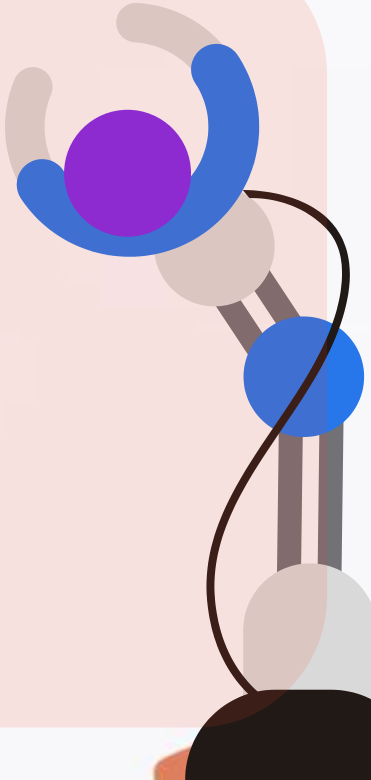
G32 2022/2023

Diogo Babo - up202004950@up.pt
Gustavo Costa - up202004187@up.pt
João Oliveira - up202004407@up.pt

Related Work

ML Supervised Learning

- Repository that has specific information about our dataset, such as its origin and the description of each attribute.
- Also different works and approaches on the same dataset are available on its Kaggle.
- Online Shoppers Intention Dataset Origin
- Online Shoppers Intention Kaggle



Problem Specification

ML Supervised Learning

Dataset:

- Online Shoppers Purchasing Intention - 12,330 user sessions;
- ***Classification***.

Features:

- 10 ***Numerical*** & 8 ***Categorical***;
- ***Visited Pages Information*** (type and duration) - Administrative, Informational & Product Related;
- BounceRate, ExitRate, PageValues & SpecialDay;
- ***Website Access Information*** - Month, OS, Browser, Region & Weekend;
- VisitorType & TrafficType.

Target Variable:

- Revenue (Whether or not the user made a purchase).

Work Carried Out

ML Supervised Learning

Programming Language:

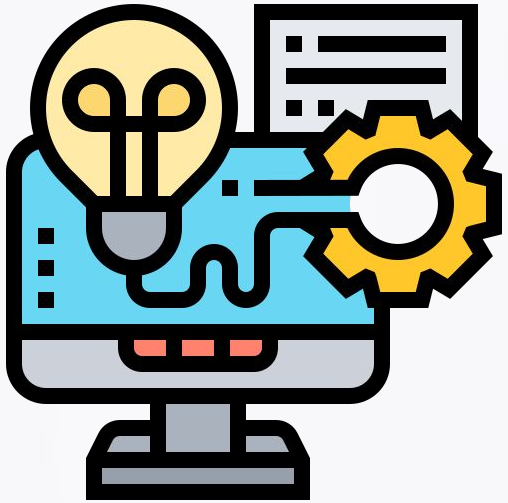
- Python;
- Numpy, Seaborn, SkLearn, MatPlot & Imblearn.

Work Developed:

- Data Pre-processing;
- Exploratory Data Analysis;
- Algorithm Comparison

Development Environment:

- Jupyter notebook.



Data Pre-Processing

ML Supervised Learning

- Checking for null values;
- Checking for **outliers:**
  - Although there were many outliers on the features, we thought it would better not to deal with most of them because we could be damaging the relationship between the attributes;
  - So we only removed impossible values, like negative time durations or percentages above 1.0 and below 0.0.

- We could conclude that our data doesn't follow a Gaussian/Normal distribution, as our data is very skewed and also that the target is highly imbalanced.

- **Data Encoding:**
  - Only 4 variables were not numerical. Two of them were binary ("Revenue" & "Weekend") and just mapped to 0's or 1's depending on its value;
  - "VisitorType" was encoded using one-hot-encoding. (Creating a new binary column for each type of visitor);
  - "Month" was encoded using circular encoding, so we could make sure that distances between the months were preserved. (e.g january is as close to february as it is to december).

Data Pre-Processing

ML Supervised Learning

- Measured the correlation between continuous attributes using a **correlation** matrix;
- Measured the correlation between continuous attributes and the target using **Point-Biserial Correlation**.
- Checked for a relationship between categorical/discrete attributes and the target using **Chi Square Test**.
  - **H0:** The attribute and the target have no relationship.
  - **H1:** The target depends on the attribute.
  - If the p-value is lower than 0.05, **H0** is rejected meaning there is a relationship between both attributes.
- This tests allowed to do some **feature selection**, because we managed to remove some attributes that had no/a weak relationship with the target. Resulting in a better performance for the algorithms/models tested.

Data Pre-Processing

ML Supervised Learning

- We did **Data Standardization** by using StandardScaler, which is a preprocessing technique used to standardize the features of a dataset. It transforms the data so that it has zero mean and unit variance.

- Since our dataset was highly imbalanced:
  - 84.5% - False
  - 15.5% - True

- We had two different approaches to overcome this problem, since most algorithms don't work well with this type of datasets:
  - **Oversampling -** Using SMOTE, we generate synthetic samples of the minority class in order to have the same proportion on both.
  - **Undersampling -** Using RandomUnderSampler, we achieved a balanced distribution between the majority and minority classes by randomly selecting a subset of instances from the majority class until we match the quantity of the minority class.

- We also used **GridSearch** to find the best parameters for our algorithms.

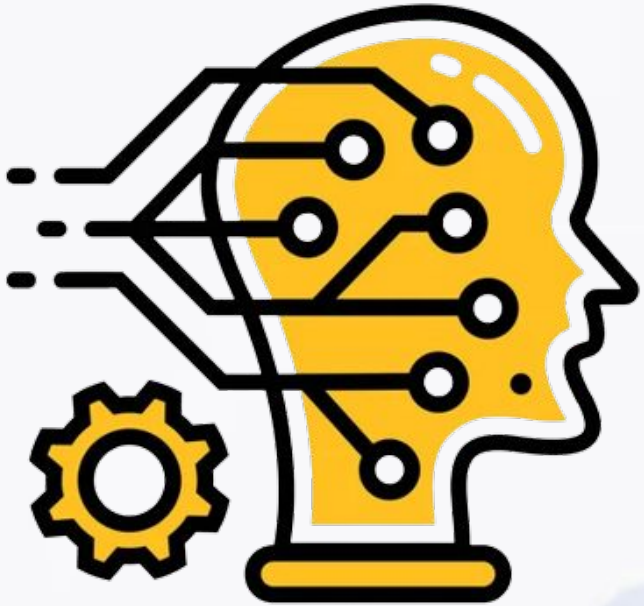
- We used 80% of the dataset for training and the remaining 20% for testing.

Tools & Algorithms

ML Supervised Learning

Classification Algorithms Used:

- Naive Bayes;
- K-Nearest Neighbors;
- Logistic Regression;
- Support Vector Machine;
- Random Forest;
- Stratified Cross Validation.



Algorithm Evaluation:

- Accuracy;
- Precision;
- Recall;
- F1-Score;
- AUC.

Algorithm Comparison

ML Supervised Learning