**Project document**

**5014502 – Savoldi Chiara**

**22/02/2022**

## **introduction**

I have decided to consider two different datasets for my project: the first, Bank\_Personal\_Loan\_Modelling.csv, contains files from a bank whose management wants to explore ways to convert their passive customers into personal loan customers (while keeping them as depositors). Last year, the bank launched a campaign for liability customers that showed a conversion rate of over 9% success. This encouraged the retail marketing department to design campaigns with better-targeted marketing to increase the success rate with a minimal budget.

The goal of my project is a predictive analysis to examine historical facts (campaign) and provide predictions: will people become or not personal loan customers?

I focused on analysing two of the most serious problems a dataset can present:

- Overfitting

- Classing Balance.

Both concepts are based on the 'Bias' notion. The bias is the difference between the average forecast of the model and the corrected value I am trying to predict. Overfitting occurs when the model captures the noise along with the underlying model in the data. It happens when we train the model excessively on noisy data sets.

Imbalanced data typically refers to a problem with classification problems where the classes are not represented equally. This will introduce a bias to select more samples from one class than from another.

The second dataset, GBvideos.csv, is a daily record with daily 7 statistics for trending Youtube videos. Data includes video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count. I choose to analize the trendind Youtube videos for Great Britain. My analysis turns out to be a regression problem because I would like to predict the number of views based on the other characteristics. I could turn it into a classification problem, creating classes with the number of views, for example from 0 to 3000000 views (of a video) is the least viral class and so on.

*I decided to keep both datasets because I had done a good clean up for both. At the oral I will present only the Bank Personal Loan dataset. I tried to clean up the datasets using both R studio and Python.*

## **background theory and scientific foundations**

My project required a background in Data analysis and Machine learning: In fact, in the first part of my project I performed an exploratory analysis of the data: this allowed me to realize that I have very unbalanced classes. In particular:

**0 4520**

**1 480**

**Name: Personal Loan, dtype: int64.**

**As can be seen, only 480 observations out of 5000 belong to class 1.**

Data analysis is a process of inspecting, cleansing, transforming, and modelling data to highlight information that suggests conclusions and supports decisions for the Bank.

Data analysis is also a process to highlight information that suggests the features a video should have to become viral.

Logistic regression is a statistical tool that aims to model a binomial result with one or more explanatory variables. It is generally used for binary problems, where there are only two classes, for example, 0,1, as in my case.

The result is determined thanks to the use of a logistic function, or logit, which estimates a probability and then defines the closest class (positive or negative) to the probability value obtained.

The K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithms. [KNN](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm) is extremely easy to implement in its most basic form, and yet performs quite complex classification tasks.

Statsmodels provides various functions for estimating different statistical models and performing statistical tests.

Neural networks predict cases using a mathematical equation, where the output is the weighted sum of the inputs, after passing through the hidden layers. In each hidden unit the net input is passed through an activation function.

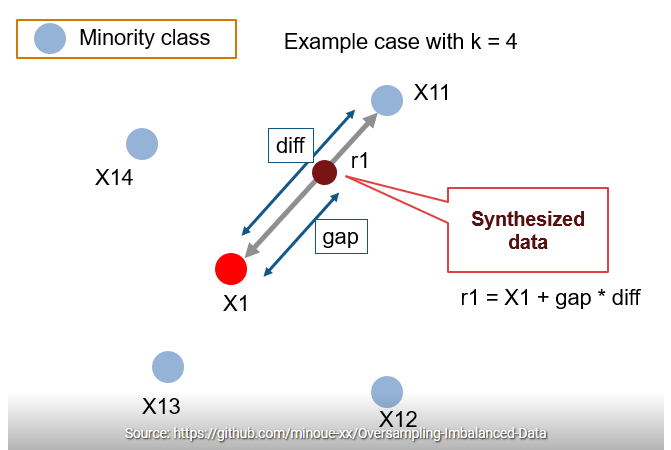
It was the first time I was working on an unbalanced dataset: I used tutorials from people who had already been in the situation focusing on the methods that exist to deal with the problem.

To be a good data scientist, I think I need to know how to use machine learning libraries such as Scikit-learn or Pandas to solve whatever problem I have at hand.

I need to know how those libraries and algorithms work under the hood: this is where my Mathematical Foundations come in. Linear algebra and calculus underly machine learning algorithms and data science models.

## One of the techniques I have used to tackle the unbalanced dataset problem is SMOTE.

## At first the total number of oversampling observations, N is set up. Generally, it is selected such that the binary class distribution is 1: 1. But that could be tuned down based on need. Then the iteration starts by first selecting a positive class instance at random. Next, the KNN's (by default 5) for that instance is obtained. At last, N of these K instances is chosen to interpolate new instances. To do that, using any distance metric the difference in distance between the feature vector and its neighbours is calculated. Now, this difference is multiplied by any random value between 0 and 1 and is added to the previous feature vector.



Another problem I encountered was having a dataset too large for the memory supported by my pc. A good Data scientist should know how to deal with these problems: my idea was to delete the rows that possessed unrelated (negatively correlated) characteristics, with the target variable. I'm not sure this is the best approach for future classifications / regressions.

## **technical analysis**

### **Python libraries and specific functions used**

Here is the list of libraries used:

- Import pandas as pd

- Import numpy as np

- Import matplotlib as mpl

- Import seaborn as sns

- Sklearn

- Statsmodels

- From IPython.display import Image

-Imblearn

-Keras

-TensorFlow

Pandas is a library for manipulating data in sequential or tabular format. I have used Pandas for loading formats for tabular data in a CSV (Comma-separated Values) format. The library was mainly used for exploratory analysis. Pandas can quickly provide us information regarding the shape of a dataframe, the summary of the dataframe, statistical information about the variables, access the columns, and draw graphs with easy and few lines of code. It offers [data structures](https://en.wikipedia.org/wiki/Data_structure) and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series).

Matplotlib is a charting library for the NumPy library. The numpy library allows working with vectors and matrices more efficiently and quickly.

Numpy works on vectors by exploiting the vector calculation optimizations of the machine's processor. The library has another library inside which contains the functions to generate random numbers (random, seed) according to different distribution functions.

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with Pandas data structures. I used it to set the display features of my confusion matrix.

Scikit-learn contains classification, regression, and clustering algorithms and support vector machines, logistic regression, Bayesian classifier, k-mean, and DBSCAN, and is designed to work with the NumPy and SciPy libraries. I used it to work on the two models: Decision tree and Logistic regression. For example, Scikit-Learn contains the tree library, which contains built-in classes/methods for the decision tree algorithms. Since I have performed a classification I have used the Logistic Regression. The fit method of this class is called to train the algorithm on the training data, which is passed as a parameter to the fit method.

I have used the Statsmodels library which provides a Logit() function for performing logistic regression. The Logit() function accepts y and X as parameters and returns the Logit object. The model is then fitted to the data.

I used the Ipython library to insert an image into my code. I have applied some of the resampling techniques, using the Python library [imbalanced-learn](http://contrib.scikit-learn.org/imbalanced-learn/stable/).

Keras is a deep learning API written in Python, running on top of the machine learning platform [TensorFlow](https://github.com/tensorflow/tensorflow). It was used to implement my Neural Network.

### **Python code**

The Bank.xls file contains data on 5000 customers. My objective is to predict whether a liability customer will buy a personal loan or not. Personal loan will be my target variable.

I did an exploratory analysis of the data: for example, I have done the treatment of the outliers. I have plotted on the x-axis each variable from the dataset, because it is interesting to understand how much the parameters are related to our variable y. One of the three levels may be more likely to answer Yes rather than No. After that, I decided to analyze some predictive models.

A model can make two kinds of wrong predictions: wrongly identify customers as loan borrowers but they are not (False Positive), wrongly identifying customers as not borrowers but they actually buy loans (False Negative).

It is interesting to see the confusion matrix, which is a table with 4 different combinations of predicted and actual values. It is extremely useful for measuring Recall, Precision, Specificity, Accuracy.

I will build the logistic regression algorithm, with two different approaches. I have also implemented a neural network.

The last analysis conducted concerned the treatment of the unbalanced dataset.

A widely adopted technique for dealing with highly unbalanced datasets is called resampling. It consists of removing samples from the majority class (under-sampling) and/or adding more examples from the minority class (over-sampling).

Finally, I concluded by analyzing the differences in the various models used, comparing the main measures used to identify the accuracy of the model.

The GBvideos.csv file contains 30000 videos. My objective is to predict how many views a video should have to become viral. The number of views becomes my target variable.

The number of Views does not represent a categorical or binary variable: to deal with it I decided to create a new Views\_1 column: in particular it will be a categorical variable. The first class will contain videos with a number of views between 851 and 3000000, the second class will contain videos with a number of views included between 3000000 and 6000000, finally there will be the class with views included between 6000000 and 920000. I did an exploratory analysis of the data: the dataset was very dirty and has, for example, some missing values. I want to classify something that has more than 2 categories: this is where multi-class classification comes in. Multiclass classification can be defined as the classifying instances into one of three or more classes. I have done multi-class classification using K Nearest Neighbours. KNN is a super simple algorithm, which assumes that similar things are in proximity of each other. So, if a datapoint is near to another datapoint, it assumes that they both belong to similar classes.

## **conclusions**

### I think that I might go more in-depth with a thesis work on this topic, to analyse well the existing libraries for the implementation of a model, and to improve knowledge at Machine Learning level: one of the challenges common to almost all companies concern the improvement of the decision-making process through data analysis. Today, companies try to extract value from data to predict the future.

### The surprise in my analysis was that I found an unbalanced dataset. Classing balance is a very frequent problem in cases where risks, fraud, etc. are measured.

The project could be deepened by entering the merits and analyzing more in-depth the results obtained by each ML algorithms.

Machine Learning is the basis of a new way of exploiting algorithms to extract data, learn from it and then extract useful information from it.

## This is also a good article to read: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d0176>.

Since the work on the Youtube video dataset is not thorough and perfect, I wanted to ask what you thought and if in case a better interpretation of the results could be made.

## **BIBLIOGRAPHY AND MORE**

NB: In each cell, I left comments on the instructions for installing packages that I didn't have installed on my pc.

-         <https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets->

-         <https://en.wikipedia.org/wiki/Main_Page>

-         <https://pulplearning.altervista.org/tecniche-undersampling-oversampling/>

-         <https://www.kaggle.com/>

- <https://keras.io/about/>

- <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

- https://www.freecodecamp.org/news/how-to-build-your-first-neural-network-to-predict-house-prices-with-keras-f8db83049159/