Bank Marketing Campaign-
Data Cleansing and Transformation

Group Name : Data team

Contributors names:

- Fawzi El Khatib, fawzi_khatib@hotmail.com, Lebanon, Data Glacier, Data Science.
- Nigel Chitere, tawanda.nigel@gmail.com, Zimbabwe, Rocapply, Data Analysis.

GitHub repo link: *https://github.com/Fawzikh/DG-BankMarketing-Campaign*

Date: 15-May-2021

Problem description

The goal of the project is to develop a model that will aid our client ABC Bank identify which clients are likely to be interested in their term deposit product before actually introducing this product into their customer base without prior research and client targeting.

In order to make clear analysis and exploration of the data in hand, we will clean our data, make some filtering, and analyze the outliers.

Moreover, before building our machine learning models we must preprocess our data in order to make it ready to be fitted in machine learning models. Also, for each modeling technique we should create its corresponding preprocessed data, which means the preprocessing steps of the Logistic Regression model will be different of the Random Forest Classifier model.

So the preprocessing steps are the transformations done on the data to transform all the features to have numerical values. These transformations depend on the type of the features.

Data Cleansing:

In order to clean our data we:

- Removed observations where “*duration*” is null, by filtering.
- Replaced “unknown” and “non_existent” values by “nan” nan values, using “*df.replace*” method or by specifying the values in “*na_values*” while reading the dataset.
- Dropping missing values using “*pd.dropna*” method.
- Dropped the “*poutcome*” feature because after dropping the unknown values including the ones in this feature we had left with 7842 observations from 45210

This small set of data where “*poutcome*” is known has led us to infer a good insight. Hence, from any part of data mining process we could get an additional insight toward who we should target of new customers that have similar characteristics

Data Preprocessing:

In this part we are going to transform our data to be ready to fitted in machine learning models. We will divide it into two parts, we will remove the outliers and transform the object features to numerical values.

I. Removing outliers: (This is a common step for both types of models we are going to use.) for Numerical features
   1. get the quantile (‘balance’, ‘duration’, ‘campaign’). Using “*.quantile(0.99)*” method, so we deal with the skewness to the right (problem with the high-end values). Or “*.quantile(0.01)*” if the problem with low-end values.
   2. Filtering the data on observations that are lower or higher than the respective quantile.

II. Transformations of the object features: (In this step we used two techniques)
   1. for Binary *(‘yes’ or ‘no’)* features: we mapped the data using “*.map*” method. Where we mapped yes→1 and no→0. This is what we did for *“housing”, “default”, and “loan”* features

   2. for Categorical features: the values of each feature here represent categories that are equally meaningful. For the purpose of making quantitative analysis, to add numeric meaning to our categorical nominal values, we used *dummy variables “pd.get_dummies” method*. This is how we transformed *'job’, 'marital’, and 'contact’* features.

Final notes:

As we mentioned before that there are differences between the preprocessing data depending on the type of the model. So let's clarify the different approaches I used in our project.

For Regression models we know that we should prevent multicollinearity. To do so we dropped the first category of each dummy variable when creating the dummies. We specify the "*drop_first*" argument to "*True*". Take into consideration that dummies are only compared to their respective benchmark, which is the dropped category.

Moreover, for our logistic regression model we created two models, one with standardized numerical features and the other without scaling the features. The purpose of the standardization is to fix the order of magnitude issue. Thus, prevent our model from learning patterns because of the magnitude of numbers.

On the other hand, concerning our Random Forest model we neither dropped the first category of the dummy nor scaled our numerical features.

Ps: we must not standardize dummy variables because we will loose the whole interpretability of the dummy.