DATA AND INFORMATION QUALITY PROJECT REPORT

PROJECT ID: 8

PROJECT NUMBER: 1

ASSIGNED DATASET: adult.csv

STUDENT: Flavio La Manna 10620549

ASSIGNED TASK: Classification

I started the project analyzing the dataset with Pandas profiling. The dataset is composed by 15 features with information about people like age, work, education, relationships, sex and native country. I didn't know the meaning of the attribute "fnlwgt" so searching on internet I found the original dataset from the UCI Machine Learning Repository. I discovered that it comes from the 1994 US Census database and "fnlwgt" represents the number of people that are probably represented by an entry. The label is the "income" and it can be greater or lower than 50k$ per year. The goal of my analysis, but also of the original dataset, is to predict whether a person makes over 50k$ per year.
I chose as ML algorithms: Logistic Regression, KNN and AdaBoost Classifier. I decided to use Logistic Regression and KNN as simple algorithms to have a baseline performance and then I used a boosting method to try improving the results. The AdaBoost Classifier is composed by 50 decision trees with maximum depth 3.
To evaluate the performance of the algorithms I computed accuracy, precision, recall and F1 measure.
I chose for the standard imputation techniques different strategies depending on the attribute:

- for attributes "age", "education-num", "hours per week" I imputed the mean given that they are numerical variables.
- for attributes "workclass", "education", "maritial status", "occupation", "relationship", "race", "sex" I used backward fill so propagating valid values in the previous missing cells. This because they are categorical variables and I didn't want to impute one single value for all missing ones (for example the mode).
- for attributes "capital gain" and "capital loss" I imputed 0 because more than 90% of their values are 0.
- for attribute "native country" I imputed "United States" because it is the most frequent category.
- for attribute "fnlwgt" I imputed a random number between the maximum and the minimum value.

I chose as advanced imputation technique MICE using a KNeighbors Regressor.

Pipeline for the standard imputation techniques:
I imported the dataset, injected the missing values and imputed using the standard techniques described above. I computed the accuracy of the imputation as the ratio between the correct imputed values (equal to the original dataset before the injection) and all the imputed values, obtaining 0.4 (for both 10% and 50% injection). I converted the categorical variables into numerical variables using pandas.get_dummies because the ML algorithms

need numerical variables. I split the dataset intro training and test set, fit the models on the training set and tested on the test set. For the dataset injected with 10% missing values the main results are:

- Logistic Regression accuracy: 0.77
- KNN accuracy: 0.74
- AdaBoost classifier accuracy: 0.8

For the dataset injected with 50% missing values the main results are:

- Logistic Regression accuracy: 0.77
- KNN accuracy: 0.74
- AdaBoost classifier accuracy: 0.76

As expected, initially the boosting ensemble has the best performance. In the second case it has a reduction of the performance while the other two remain the same probably because there are too many missing values and the ensemble of models increases the error.

Pipeline for the advanced imputation technique:
Same as before, but in this case I need to convert all the categorical variables to numerical in order to apply the MICE. The difficulty was converting the variables while keeping the NaN values so I couldn't use the OneHotEncoder because I would have lost them. So, I decided to convert the variables using the LabelEncoder knowing that it isn't the ideal solution because it introduces an implicit order in the feature. MICE uses a KNeighbors Regressor for the imputation so in order to not impute some meaningless averages, I used only one nearest neighbor. I applied the same label encoding to the original dataset to compare them and obtained an imputation accuracy of 0.44 (for the 10% injection) and 0.39 (for the 50% injection). I split the dataset intro training and test set, fit the models on the training set and tested on the test set. For the dataset injected with 10% missing values the main results are:

- Logistic Regression accuracy: 0.76
- KNN accuracy: 0.73
- AdaBoost classifier accuracy: 0.79

For the dataset injected with 50% missing values the main results are:

- Logistic Regression accuracy: 0.74
- KNN accuracy: 0.75
- AdaBoost classifier accuracy: 0.76

As before, AdaBoost has always the best performance and there is a reduction when incrementing the number of missing values.
I didn't expect that the standard imputation techniques in both cases and for all the models could perform better than the ML techniques. Probably the poor results of MICE are due to the Label Encoding of the categorical variables and the KNeighbors technique with only one neighbor.