Supervised Learning

Assignment 2
https://git.fe.up.pt/up202008133/ia_t03g34_assignment2

João Pinheiro (202008133)
José Araújo (202007921)
Ricardo Cavalheiro (202005103)

1. Problem Specification

(definition of the machine learning problem to address: dataset, features, target variable)

- Use of machine learning models and algorithms related to supervised learning
- The goal of this project is to detect fraudulent occurrences
- Given some dataset, where in each entry there is a column labeling the occurrence as being fraudulent or not (classification), we are supposed to predict, on a new dataset, whether or not a bank account should or shouldn’t be tagged as a fraud
- Prediction accuracy is as good as the quality of the training data

2. Description of the tools to use

Plot and table libraries: seaborn, matplotlib, tabulate

Data analysis libraries: pandas, imblearn (oversampling and undersampling)

Math libraries: numpy (numpy arrays are very efficient)

Machine learning libraries: sklearn (all machine learning algorithms and tools are included), scipy

Evaluating

Log loss (good for binary problem), confusion matrix (a better number of true positive may be better), Recall, ROC curve

3. Work implemented

- **Data Understanding and Exploration** (visualize dataset, address outliers, target variable distribution, division of variables, etc)
- **Exploratory Data Analysis (EDA)** (plots and statistics to understand relations between variables and target variable and also between variables)
- **Feature Engineering and Selection** (data encoding, selection of relevant features, analysing dataset imbalance recurring to ROC curves)
- **Datasets preparation for the models** (splitting processed dataset into training and testing sets)
- **Statistical Modeling** (applying modeling techniques)
- **Models Evaluation and Validation** (analyse models performance on testing data, fine-tuning and applying cross validation techniques)
- **Interpretation and Conclusions**

4. Data Preprocessing

- Visualized data to observe the data we were working with
- Displayed and analyzed box-plots
- Removed column ‘device_fraud_count’ from dataset as it only contained 1 value (does not provide any meaningful variation)
- Produced histograms of numerical columns to see each feature distribution (identify features that need transformation to achieve normal distribution)
- Count plot to check target variable distribution (to identify imbalance)
- Division of variables into 4 groups: target variable, continuous variables, categorical variables and binary variables
- Identify and analyse outliers (not necessary as we found they would not influence analysis accuracy)
- Correlation Matrix to understand relation between continuous features
- Plots to understand correlation between continuous and target variables
- Tests to understand correlation between categorical and target variables

4. Data Preprocessing

- T-test and Mann-Whitney U Test (equal variance vs assumption of not equal variance in 2 groups - group of 1’s and group of 0’s from target variable)
- Cohen U3 to check difference between fraud and not fraud categories for each variable
- Point-Biserial Correlation (relation between continuous and target variables)
- Chi-square test (correlation between categorical and target variables)
- Data encoding (Ordinal Encoding - assigns unique integers to each category)
- Selection of most relevant features based on plot and statistical results analysis
- Creation of different datasets (resorting to oversampling, undersampling, SMOTE) as our original dataset is imbalanced (many more 0’s than 1’s in target variable) and that leads to poor performance regarding identifying fraudulent cases
- Usage of ROC curves to find which technique used previously was best (undersampling had the best performance, higher AUC (Area Under Curve))

5. Developed models

- **Naive Bayes:** a simple and efficient classification algorithm based on Bayes' theorem and the assumption of feature independence
- **Decision Tree:** a tree-based classification algorithm that splits data based on feature values to create decision rules for predicting class labels
- **k-Nearest Neighbors (KNN):** a classification algorithm that predicts the class of an instance based on its nearest neighbors in the feature space
- **Logistic Regression:** a classification algorithm that predicts the probability of an instance belonging to a class using a sigmoid function and learns optimal coefficients from the data
- **Support Vectors:** a classification algorithm that finds an optimal hyperplane to separate classes by maximizing the margin and uses support vectors as critical data points
- **Random Forest:** ensemble classification algorithm that combines multiple decision trees to make predictions by aggregating their results, providing improved accuracy and handling of complex datasets

6. Models evaluation and comparison