TÉCNICO LISBOA

MEIC 2019/20

Data Science Project

Deadline – November 22nd (23:59)

1. Project Goal

Critical application of data science techniques to discover information in two distinct problems (datasets). Students are asked to explore the datasets and, in accordance with their findings, adequately select and learn models for the available data, as well as assess and relate those models.

Additionally, students should be able to criticize the results achieved, hypothesize causes for the limited performance of the learned models, and identify opportunities to improve the mining process.

2. Data

The datasets for the two target problems in this project:

- **Parkinson Disease** (pd_speech_features.csv). Source data and description in: https://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification
- **Covertype** (covtype.info + covtype.data). Source data and description in: https://archive.ics.uci.edu/ml/datasets/Covertype
  (please undersample all classes with more than 10k instances)

3. Methodology

Information discovery on both datasets must be done using essential *exploratory data analysis*, *preprocessing techniques* (data transformations to facilitate the learning and or handle specific data aspects such as semi-structured data, high-dimensionality, missings, non-i.i.d. attributes or imbalanced classes), ***unsupervised techniques*** (pattern mining and clustering), and ***classification techniques***, including naïve Bayes, kNN, decision trees, random forests and XGBoost.

Students may choose the mining tool to apply, between **python** (using *scikit-learn*) and **R**.

Guidelines

Description

The students should perform a statistical analysis of the datasets in advance and summarize relevant implications in the report, such as the underlying distributions and hypothesized forms feature dependency.

Preprocessing

In accordance with the properties of the input dataset and the behavior of the target learning algorithm, the students are allowed to apply preprocessing techniques when needed or under a solid conjecture of its potential impact on learning.

Unsupervised Learning

Unsupervised exploration must be done through clustering and association rule mining. Class attributes should not be used to explore the data, unless there is a well substantiated interest for mining class-conditional data or discovering association rules with classes in the antecedent/consequent. Nevertheless, class attributes may be used to objectively assess clustering results and evaluate the discriminative power of certain association rules. Besides this, statistical evaluation must be performed using the studied indexes.

Classification

Supervised exploration must be done via the application of *kNN, Naïve Bayes, Decision Trees, Random Forests* and *XGBoost*. For this purpose, the use of class attributes is mandatory. Evaluation of the obtained models should be done as usual, through accuracy measures and evaluation charts, as studied in the classes. A thorough comparison of the adequacy of the models should be present taking into consideration the adequacy of their behavior against the properties of each dataset and their observed performance.

Excellence

A project that applies the suggested data mining techniques over the given datasets and provides a clear and *sound analysis of the collected results is not necessarily an excelling project*.

Excelling projects have four major characteristics.

First, they show an acute understanding of the data characteristics and their impact on the learning. Excelling projects formulate hypothesis behind differences in performance.

Second, they have precise and succinct language: no redundancies, unnecessary or subjective statements.

Third, excelling projects are often a result of a creative thinking on ways of improving the learning. Illustrating, the justified use of a specific preprocessing technique (whether the inclusion of new features, space transformations, handling of outliers or specific forms of noise) can make a difference.

Fourth, robust assessments go beyond simple performance indicators. Excelling projects draw (parameter-varying) plots, test hypotheses, and establish ratios to understand less-trivial performance views such as robustness to noise, domain adequacy or overfitting propensity.

4. Delivery

Students should register their groups and deliver the project through **Fénix system**.

Students have to submit a report and the code, according to the following specifications.

Report (90%)

The report should be named **`report_X.pdf`** (replacing X by the group number) and submitted through Fénix. It should follow the template and structure given, without any cover page, **8 pages** including any appendix. Each additional page won't be considered. The report may be written in Portuguese or English. The report can be written using Latex or Word, and submitted in .PDF format.

The report should describe in a succinct/structured form the placed choices, pre-processing performed, applied parameterizations, found results, their interpretation and critical analysis for each dataset. Additionally, it should include a comparison of the results achieved in both problems, and the relation among the information discovered through the different techniques.

Functional project (10%)

Students should develop a *Python project* or *R project* that automatically returns relevant results in the presence of new data instances from one of the two given datasets. The code will be submitted through Mooshak system (details will be posted on Fénix – section Project).

Your project should be able to read from the input console. The input console will provide:

- the number of lines to read
- the name of the dataset: `PD|CT` where `PD` and `CT` respectively define whether the sourced data instances belong to the first dataset (Parkinson disease) or second dataset (covertype)
- the task: `preprocessing|unsupervised|classification` where `preprocessing` option should return in the output console up to 100 lines of statistics comparing the adequacy of different preprocessing techniques, the `unsupervised` option should produce up to 100 lines of statistics on the found clusters and patterns; and `classification` option should return up to 200 lines of statistics comparing the classifiers' performance (including a confusion matrix per classifier) using a 10-fold cross validation
- the dataset (header followed by data instances)

Example:

INPUT CONSOLE

```
100 CT classification
Elevation,Aspect,Slope,Horizontal_Distance_To_Hydrology, [...], Cover_Type
2596,51,3,258,0,510,221,232,148,6279,1,0,0,0,0,0,0,0,0,0, [...], 5
[...]
```

OUTPUT CONSOLE

```
1. Applied preprocessing: <...>
2. Classifiers:
2.1 NB
a) Suggested parameterization: <...>
```

```
b) Confusion matrix: <...>
# performance improvements against default preprocessing and parameterization can be included
2.2 <...>
3. Comparative performance: NB | kNN | DT | RF
3.1 Accuracy: 0.76 | 0.81 | 0.56 | 0.90
3.2 Sensitivity: <...>
<...>
```

Students should guarantee the interpretability of the outputs and that each command can be executed in useful time. External libraries (other than scikit-learn, pandas, scipy, numpy, matplotlib) should be explicitly included in the project within a folder `imports`.

FAQ and additional details will be provided in the webpage of the course in the upcoming weeks.

Plagiarism

Plagiarism is an act of fraud. We will apply state-of-the-art software to detect plagiarism. Students involved in projects with evidence of plagiarism (on code or written report) will be reported to the IST pedagogical council in accordance with IST regulations.

5. Evaluation Criteria

The project will be evaluated as a *whole*. Nevertheless, we provide below a decomposition of the total project score for the purpose of guidance and prioritization. Reference scores:

1. **Statistical description** (**5%**)
2. **Preprocessing** (**20%**)
2. **Unsupervised**
   a. Association Rules (**7.5%**)
   b. Clustering (**7.5%**)
3. **Classification**
   a. Naïve Bayes (**2%**)
   b. Instance-based Learning (**3%**)
   c. Decision Trees (**5%**)
   d. Random Forests (**5%**)
   e. XGBoost (**5%**)
6. **Evaluation and critical analysis (30%)**
7. **Creativity (10%)**