Course Project

Due: 5/6

Now you are all experts in Data Mining, the time has come to explore an interesting project of your choice. You can work in a group of no more than 4 people. Please finalize your teammates and sign up your team following the link given below.

Each team has to create your own deliverables. It is not allowed to copy another team's code or text and modify it. If you use publicly available code or text, you must **cite the source** in your report!

Project key events:

- Team sign ups are due on **3/18**.
  https://docs.google.com/spreadsheets/d/1M3klIMqPSoIQKw9-chzd9hgsbupxkTw1gT9aq7bbvnE/edit?usp=sharing
- Final code and project report are due on **5/6**.
- Each team present their project on **5/6**.

The project is graded in a scale of 100, which breaks down as follows:

- Program (code and execution results match the report) 50%
- Report 50%

Possible Course Project: Join a Kaggle Competition

https://www.kaggle.com/competitions

Default Project

You probably will be more motivated if you choose a project on your own. Here is the default project to use if you don't have other preferences.

1 Introduction

This project requires you to explore classification algorithms on a real world dataset, and write a report explaining your experimental results. The language of implementation is up to you — the only requirement is that your program be able to interpret the data format specified below, and be able to classify instances and produce interesting statistics such as accuracy, false positive rate, false negative rate, etc. You are free to construct whatever user interface for your program, but you must *fully document* your interface.

2 Algorithm

- Your algorithm should be based on the classification algorithms learned during the course. Usually a straight forward implementation of one method will not lead to satisfactory performance. Your algorithm can be a combination of methods and should incorporate one or more data mining techniques when the situation arises. These techniques include (and certainly not limited to):
  - Handling imbalanced dataset
  - Proper imputation methods for missing values
  - Different treatment of various type of features: continuous, discrete, categorical, etc.

3 Data

You'll be examining the behavior of your classification algorithm on a dataset from the UCI machine learning lab. The dataset is represented in a standard format, consisting of 3 files. The first file, `census-income.names`, describes the categories and features of the dataset. It also has some empirical results for your reference. The other two files are `census-income.data` and `census-income.test`, containing the actual data instances, formatted at one instance per line, as follows:

$F_1^1, F_1^2, \ldots, F_1^k,\ \text{label}_1$

$F_2^1, F_2^2, \ldots, F_2^k,\ \text{label}_2$

$\vdots$

$F_n^1, F_n^2, \ldots, F_n^k,\ \text{label}_n$

where F_i^j, label_i $(i = 1, \ldots, n, j = 1, \ldots, k)$ represent the value of the j^{th} feature and class category for the i^{th} instance respectively.

The data you will be examining was extracted from the census bureau database. Each instance contains an individual's educational, demographic and family information. Prediction task is to determine whether a person makes over 50K a year. You should use `census-income.data` to train your classifier and use `census-income.test` to evaluate the performance of your learning algorithm.

4 Your Mission...

Deliverables for this project are:

- Code to implement the classification algorithm for the data file formats given above.
- **A README file, with simple, clear instructions on how to compile and run your code**.
- Testing statistics for the application of your learning algorithm. At a minimum you should provide training set accuracy, test set accuracy.

- A discussion of data mining techniques employed in your algorithm.
- A report analyzing the behavior of your algorithm on the dataset, including any unusual or anomalous (in your opinion) behavior.

5 How to turn in your code

- **Your program must run on erdos.dsm.fordham.edu**
- **Zip all your files (code, README, written report, etc.) in a zip file named** $\{firstname\}\_\{lastname\}\_CS5790\_project.zip$ **and upload it to Blackboard**
- **Only one person in your group needs to turn in the code and the report. Make sure every team member's name is listed on the cover of the report**