UNIVERSITY COLLEGE LONDON
Faculty of Engineering Sciences

Department of Biochemical Engineering

BENG0095: Group Assignment

PREDICTING CREDIT CARD DEFAULT

Dr. Dariush Hosseini (dariush.hosseini@ucl.ac.uk)

Overview

- Assignment Release Date: Friday 6th November 2020
- **Assignment Hand-in Date: Tuesday 11th January 2021 at 4pm**
- Weighting: 50% of module total
- Submission Format: A zip file containing a PDF file of the written report and a Juptyer Notebook .ipynb file of the code

Please note that this is a group assignment. As a first step, please arrange yourself into groups of 5 or 6 (up to the specified group maximum capacity) and then register your group on the Moodle module page under the 'BENG0095 (2020/21) Group Assignment: Group Choice' section under the 'Assessment ' tab.

Assignment Description

This assignment prompts you to examine one of the oldest use cases of machine learning in finance: credit card default prediction.

You are provided with data taken from the 'Default of Credit Card Clients' Data Set taken from the UCI Machine Learning Repository (Lichman, M. (2013). UCI Machine Learning Repository `http://archive.ics.uci.edu/ml`. Irvine, CA: University of California, School of Information and Computer Science). The data is split into a training and a test data set.

Your task is to use the training data to build a machine learning model that can predict the outcome of credit card default on the test data.
You will need to build a model that predicts the value of the `default_payment_next_month` variable which can take the values $\{0, 1\}$ indicating no default or default.
You can use the training data along with a suitable evaluation method (e.g. splitting the training data into training and validation sets) to train and validate your model.

You will be assessed primarily not so much on the final predicitve accuracy which you obtain, but rather on your approach when attempting this task, and on the level of understanding which you display. Be creative and ask questions of this data, perhaps engineer your own features as an input to your classifier, and think hard about what constitutes success. There are many existing approaches to credit card default prediction so do a search of the literature to find inspiration. Most of all, please reflect upon what you have learned during the term, and seek to apply machine learning in a way that is appropriate for this task.

The assignment submission will take the form of:

1. A PDF file containing a report of the task.
2. A Jupyter notebook containing the **Python** source code of your approach as well as (brief) in-line documentation.

Data Description

The data is available online via the module's Moodle page under the 'Group Assessment' link, and comprises two files: `CreditCard_train.csv` and `CreditCard_test.csv`, described below:

`CreditCard_train.csv`: This file contains the data that you are to train and evaluate your model on. It consists of input and output data for the task. The features are as follows:

- `ID`: The client id
- `X1`: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit
- `X2`: Gender (1 = male; 2 = female).
- `X3`: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- `X4`: Marital status (1 = married; 2 = single; 3 = others).
- `X5`: Age (year).
- `X6 - X11`: History of past payment: (The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above)
  - `X6` = the repayment status in September, 2005
  - `X7` = the repayment status in August, 2005
  - `X8` = the repayment status in July, 2005
  - `X9` = the repayment status in June, 2005
  - `X10` = the repayment status in May, 2005
  - `X11` = the repayment status in April, 2005.
- `X12 - X17`: Amount of bill statement: (NT dollar)
  - `X12` = amount of bill statement in September, 2005
  - `X13` = amount of bill statement in August, 2005
  - `X14` = amount of bill statement in July, 2005
  - `X15` = amount of bill statement in June, 2005
  - `X16` = amount of bill statement in May, 2005
  - `X17` = amount of bill statement in April, 2005.
- `X18 - X23`: Amount of previous payment: (NT dollar)
  - `X18` = amount paid in September, 2005
  - `X19` = amount paid in August, 2005
  - `X20` = amount paid in July, 2005
  - `X21` = amount paid in June, 2005
  - `X22` = amount paid in May, 2005
  - `X23` = amount paid in April, 2005
- `Y`: The default outcome (0: No default, 1: Default)

`CreditCard_test.csv`: This file contains the data that will be used to perform the final predictions which form part of your submission analysis.

Getting Started

Some points to help you get started:

- There are lots of papers available online that detail different approaches to this problem. It is worth spending some time at the start of the project doing background research and getting a feel for the data but **do** reference any work that you have taken inspiration from.
- Also, please note that while this is a well-explored data set in the literature, you will receive a portion of your marks for the novelty with which you approach the problem. If you merely seek to re-implement an existing solution you should not expect high marks.
- You **can** use existing libraries such as scikit-learn to provide implementations of key algorithms. I do not expect you to write your own versions of individual algorithms.
- All source code should be written in Python.

Submission Format & Structure

The assignment submission will take the form of a zip file containing:

1. A **PDF file** containing a **report** of the task (this should be at most 10 A4 sides in length, including references). This PDF should be prepared using the LaTeX files included on the module Moodle page under the 'Group Assessment' link, which provide a format similar in style to "preprint" publications such as arXiv.

2. A **Jupyter notebook .ipynb file** containing the **Python source code** of your approach as well as (brief) in-line documentation. The notebook should include an analysis of the performance of your classifier on the data from the test set file.

The PDF report should adopt the following structure:

1. **Introduction**

   A brief description of your approach to the problem and the results that you have obtained on the training data.

2. **Data Transformation & Exploration**

   Any transformations that you apply to the data prior to training. Also, any exploration of the data that you performed such as visualization, feature selection, etc.

3. **Methodology Overview**

   Start by describing in broad terms your methodology. Include any background reading you may have done and a step by step description of how you have trained and evaluated your model. Describe any feature engineering that you have applied. If you had attempted different approaches prior to landing on your final methodology, then describe those approaches here.

4. **Model Training & Validation**

   This contains a breakdown of how your model was trained and evaluated.

5. **Results**

   Here you show the results that you obtain using your model on the training data. If you have multiple variations or approaches, this is where you compare them.

6. **Final Predictions on Test Set**

   This is the section where you perform your final predictions on the test set using the model that you have trained in the previous section.

7. **Conclusion**

   This is the section where you consider your findings and suggest avenues for future reasearch.

The Notebook should adopt the following structure:

1. **Introduction**

   A brief précis of the equivalent section in your report.

2. **Data Import**

   This section is how you import the data into the notebook. It should be written in such a way that I can modify it to run on my own machine by simply changing the location of the training data and any additional data sources that you have used.

3. **Data Transformation & Exploration**

   Code for the equivalent section in your report, together with in-line documentation of that code.

4. **Methodology Overview**

   Code for the equivalent section in your report, together with in-line documentation of that code.

5. **Model Training & Validation**

   Code for the equivalent section in your report, together with in-line documentation of that code.

6. **Results**

   Code for the equivalent section in your report, together with in-line documentation of that code.

7. **Final Predictions on Test Set**

   Code for the equivalent section in your report, together with in-line documentation of that code.

Note:

- Your notebook need only contain brief in-line documentation, while the PDF should contain a more detailed description.
- You will be assessed primarily on the contents of your PDF report. The notebook is required so that we can check that your results are replicable.
- Keep in mind that your notebook should be written in such a way that we can modify the location of the data and then step through your notebook to obtain the same results as you have submitted.

Marking Guidelines

All reports will be marked against the marking rubric, which is available online via the module's Moodle page under the 'Group Assessment' link.
The mark weighting for each section is as follows:

- **Methodology (15%)**

  How well is the methodology described? How appropriate is it to the task at hand? Have you done more than just apply a classifier to the training data?

- **Evaluation Strategy (15%)**

  Has a suitable evaluation strategy been used so as to avoid any possible bias? If your methodology contains multiple parameters, how have the final parameter values been chosen? Have you used any form of cross validation?

- **Presentation of Results (15%)**

  Have you presented results on the training data? Are the results presented appropriate and displayed in an easy to interpret manner? Do they reveal any extra insights about how your model performs?

- **Interest of Approach (40%)**

  How interesting and novel is your approach (regardless of predictive accuracy)? Have you transformed the training data in an interesting way? Have you done something that is beyond simply using a standard classifier on the training data?

- **Format, structure, referencing, and clarity of writing/code (15%)**

  Is your final notebook well laid out and does the write-up follow a clear structure? Have you included any references to show background research/reading? Is your writing free from spelling, punctuation, and grammatical errors and is your code well commented?

For a more detailed breakdown of what constitutes a good (and bad) mark for each of these sections please refer to the marking rubric.