CP640 Machine Learning - Assignment 1
Due Date: Oct 11, 2023 at 11:59 PM

Assignment Submission Guidelines

1. **File Naming**: Ensure your assignment file is named in the following format: your network login followed by the assignment number. For instance, if the username is "barn4520" and you're submitting Assignment 1, the file should be named "barn4520_a01.ipynb".
2. **Assignment Format**: All assignments should be completed using Jupyter Notebook. Once you're done, ensure you run all the cells to verify their functionality, then save your work as a ".ipynb" file. For theoretical or conceptual queries, provide your responses within the notebook using markdown cells. Coding segments must be well-documented for clarity.
3. **Submission Platform**: All submissions must be made via the MyLearningSpace website. We do not accept assignments through email.
4. **Late Submissions**: If you submit your assignment within 24 hours post the deadline, your grade will be reduced by 50%. Unfortunately, we cannot accept submissions made beyond 24 hours from the deadline, and such submissions will receive a grade of 0.
5. **Plagiarism Policy**: All submitted code will undergo a plagiarism check. Engaging in plagiarism can have significant academic consequences. Using generative AI to aid in or fully complete your coursework will be considered academic misconduct.

PART 0: Preliminary Steps

0.1 Configuring Your Software Environment

Begin by establishing the software ecosystem that we'll be utilizing throughout this course. As discussed during our sessions, you're free to set up the necessary software on your personal device. If you opt for this route, it's crucial to verify that all installed components are up-to-date.

For our programming tasks, we'll predominantly be using Python, supplemented by a few Python libraries. Most of these tools are encompassed within the SciPy stack, an open-source collection tailored for applications in mathematics, science, and engineering. For ease of installation, we recommend the Anaconda Python Distribution, which conveniently bundles the SciPy stack and is compatible across Linux, Mac, and Windows platforms.

Ensure your system has the following components installed:

- **Python**: A versatile, object-oriented programming language.
- **NumPy**: Essential for scientific computations in Python.
- **SciPy**: Designed for advanced math, science, and engineering applications.
- **Matplotlib**: Ideal for creating 2D visualizations in Python.
- **pandas**: Offers powerful data structures and tools for efficient data analysis.
- **IPython**: Provides an enhanced interactive Python computing experience.
- **scikit-learn**: A comprehensive library for machine learning in Python.

0.2 Initiate Your First Notebook

Kickstart by crafting an IPython Notebook. Incorporate the sample code provided below and feel free to introduce any personal touches (for instance, you can display your name). Ensure you tweak or append at least one line of code. To set up a new IPython Notebook, launch the Jupiter Notebook from your terminal. This action will usher you into the IPython web interface, from where you can opt for 'New Notebook'. Once you've finalized your edits, rename your assignment, and save it, which will produce an '.ipynb' file.

0.3 Testing Sample Python Code with Essential Modules

Your finalized notebook should bear resemblance to this reference:
`https://nbviewer.jupyter.org/github/wlucp640/a1/blob/master/a1_samplecode.ipynb`.

Problem Definition

In this assignment, you'll delve into the intricacies of real-world data, employing classifiers such as decision trees and KNN from scikit-learn to gauge the lending risk. The objective here is to familiarize you with the scikit-learn API. By the end, you'll be adept at exploring fundamental statistics, constructing classification models, managing train/validation data divisions, and assessing the outcomes.

You'll dive into an accessible dataset from *LendingClub*[1], a platform bridging individuals in need of funds with potential lenders. Your goal is to build a model that assesses the risk associated with lending money based on diverse profile attributes. Specifically, we'll leverage historical records to predict **if a borrower has fully repaid their loan**. Below are the descriptions for each column in the dataset:

- credit.policy: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.
- purpose: The purpose of the loan (takes values "credit_card", "debt_consolidation", "educational", "major_purchase", "small_business", and "all_other").
- int.rate: The interest rate of the loan, as a proportion (a rate of 11% would be stored as 0.11). Borrowers judged by LendingClub.com to be more risky are assigned higher interest rates.
- installment: The monthly installments owed by the borrower if the loan is funded.
- log.annual.inc: The natural log of the self-reported annual income of the borrower.
- dti: The debt-to-income ratio of the borrower (amount of debt divided by annual income).
- fico: The FICO credit score of the borrower.
- days.with.cr.line: The number of days the borrower has had a credit line.
- revol.bal: The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).
- revol.util: The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).
- inq.last.6mths: The borrower's number of inquiries by creditors in the last 6 months.
- delinq.2yrs: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.
- pub.rec: The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments).
- not.fully.paid: The quantity of interest for classification - whether the borrower paid back the money in full or not

You can download the complete dataset, 'loan_data.csv', from MyLearningSpace. For your submission, please include only the .ipynb file; there is no need to attach the original dataset.

[1] https://www.lendingclub.com/

1 Questions:

Please answer the following questions with Python program:

1. Explore data statistics (6 points).
   (a) Calculate the average FICO credit score for customers who meet (credit.policy = 1) and don't meet (credit.policy = 0) the credit underwriting criteria.
   (b) Visualize the distribution of FICO scores for those who have and haven't fully paid their loans with two separate histograms in a single plot.
   (c) Calculate the correlation between interest rate and FICO score and explain what it implies.
   (d) Visualize the relationship between FICO score and interest rate using a scatter plot and compare it with your conclusion drawn from the last question.
   (e) What's the average interest rate based on the purpose of the loan?
   (f) Is there a significant difference in the distribution of FICO scores between customers who fully paid their loans and those who didn't? (Hint: Visualize using a boxplot)
2. Prepare the dataset for model training (4 points). 1) convert categorical variables (e.g., purpose) into dummy variables; 2) drop the category reference; 3) show the first 5 rows after preprocessing; 4) Split the data into training and test sets (70% training, 30% test), and show the sizes of training and testing data.
3. Construct a decision tree (4 points). 1) Train a decision tree classification model using the Gini criterion and show its accuracy. 2) Train a decision tree classification model using the Entropy criterion and show its accuracy. 3) Which decision tree model (Gini or Entropy) performs better on the test set?
4. Build a KNN (3 points). 1) Train a K-Nearest Neighbors (KNN) classifier with $k = 5$ and show its accuracy. 2) Train a K-Nearest Neighbors (KNN) classifier with $k = 3$ and show its accuracy. 3) Which K-Nearest Neighbors model ($k = 3$ or $k = 5$) performs better on the test set?
5. Perform evaluation (3 points). 1) Evaluate the performance of the decision tree (using Entropy) and KNN ($k = 5$) models using the F1-score. 2) Calculate the precision and recall for the decision tree (using Entropy) model. 3) Calculate the ROC-AUC score for both the decision tree (using Entropy) and KNN ($k = 5$) models.