CP322A Machine Learning - Assignment 1
Due Date: Oct 7th, 2020 at 11:00 PM

About Submission

When writing and submitting your assignments follow these requirements:

- Name your assignment with your network login, an underscore, 'a' (for 'assignment', then the assignment number: *login_en.zip*. For example, if the user barn4520 submits Assignment 1, the name should be: $barn4520\_a01$. Give your .zip file the same name when exporting your project.
- Late assignment submissions will not be accepted and will be marked with 0.
- Your assignment should be submitted online through the MyLearningSpace website. Email submission is not accepted.
- You are expected to submit a single IPython Notebook file for this assignment. Answers to the conceptual questions can be embedded to the notebook file as markdown cells, and you may use heading cells to further organize your document.
- Please document your program carefully. Source code is be required.

Before You Start

0.1 Setting-up Your Software Environment

This part simply requires you to setup the programming environment that we will be using for the remainder of the course. As stated in class, you may install the required software in your personal computer, in which case we suggest you carefully ensure that everything you install is up-to-date, which will help avoid compatibility issues when your assignments are graded.

Programming assignments will require the use of Python as well as additional Python packages. Most of the relevant software is a part of the SciPy [1]stack, a collection of Python-based open source software for mathematics, science, and engineering (which includes Python, NumPy, the SciPy library, Matplotlib, pandas, IPython, and scikit-learn). The Anaconda Python Distribution[2] is a free distribution for the SciPy stack that supports Linux, Mac, and Windows. Ensure that your machine has the following software installed:

- Python (An interactive, object-oriented, extensible programming language.)
- NumPy (A Python package for scientific computing.)
- SciPy (A Python package for mathematics, science, and engineering.)
- Matplotlib (A Python package for 2D plotting.)
- pandas (A Python package for high-performance, easy-to-use data structures and data analysis tools.)
- IPython (An architecture for interactive computing with Python.)
- scikit-learn (A Python package for machine learning.)

0.2 Create Your First Notebook

Create an IPython Notebook and run the code in Listing 1 with any modifications you desire(e.g., print your name somewhere). Be sure to modify/add at least one line. To create a new IPython Notebook, you simply need to open the terminal *Jupiter Notebook*, and this will bring up the IPython web interface from where you may select *New Notebook*. Once you are finished, rename and save your assignment, and this will generate an .ipynb file.

[1] https://www.scipy.org/

[2] https://www.anaconda.com/distribution/

0.2.1 Sample Python Code for Testing Required Modules

The final notebook should be similar to this one here: .

`https://nbviewer.jupyter.org/github/wlucp640/a1/blob/master/a1_samplecode.ipynb`

Sample Python code for testing required modules:

```
# Testing NumPy
import numpy as np
np.arange(15).reshape(3, 5)
# Testing SciPy
import scipy as sp
sp.linspace(0, 10, 5000)
#Testing matplotlib
import matplotlib.pyplot as plt
x = np.linspace(0, 1)
y = np.sin(4 * np.pi * x) * np.exp(-5 * x)
plt.fill(x, y, 'r')
plt.grid(True)
plt.show()
# Testing pandas
import pandas as pd
ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()
ts.plot()
# Testing Scikit Learn
from sklearn.svm import SVC
from sklearn.datasets import load_digits
from sklearn.feature_selection import RFE
# Load the digits dataset
digits = load_digits()
X = digits.images.reshape((len(digits.images), -1))
y = digits.target
# Create the RFE object and rank each pixel
svc = SVC(kernel="linear", C=1)
rfe = RFE(estimator=svc, n_features_to_select=1, step=1)
rfe.fit(X, y)
ranking = rfe.ranking_.reshape(digits.images[0].shape)
# Plot pixel ranking
plt.matshow(ranking)
plt.colorbar()
plt.title("Ranking of pixels with RFE")
plt.show()
```

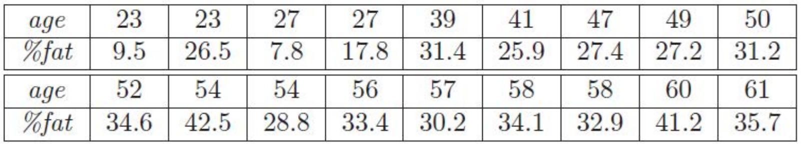
1 Concept Questions

For questions in this section, you do not need to write a Python program; however, you are still expected to demonstrate working steps and results in your *.ipynb* submission. To this end, you can label your solution as *Markdown* instead of *Code*.

1.1 Data Exploration

1. Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45,46, 52, 70. Use smoothing by bin means to smooth the above data, using a bin depth of 3. Illustrate your steps. Comment on the effect of this technique for the given data.

2. Suppose a hospital tested the age and body fat data for 18 randomly selected adults with the following result.



age	23	23	27	27	39	41	47	49	50	52	54	54	56	57	58	58	60	61
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

Calculate the correlation coefficient. Are these two variables positively or negatively correlated?

1.2 Decision Tree

1. It is important to calculate the worst-case computational complexity of the decision tree algorithm. Given data set D, the number of attributes n, and the number of training examples $|D|$, show that the computational cost of growing a tree is at most $n \times |D| \times log(|D|)$.

2. Compare the advantages and disadvantages of eager classification versus lazy classification.

2 Programming Questions:

In this assignment, you will use the *scikit-learn*'s decision tree to predict the risk of lending money. The aim of this question is for you to read the *scikit-learn* API and get comfortable with exploring basic statistics, developing classification models, and handling training/validation splits.

2.1 Know Your Data: 5 points

We will explore a publicly available dataset from *LendingClub*[3], which connects people who need money with people who have money. We attempt to construct a model to analysis the risk of lending money to people given various profile data. In particular, we exploit the historical data to predict whether or not the borrower paid back their loan in full. Here are the meanings of different columns in the data set:

- 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

Please complete the following tasks:

1. Print the first 5 records of your data.
2. Demonstrate the basic statistics of different features, i.e., count, mean, std, min, max, and 25:50:75% percentiles.

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

3. Show the breakup of credit approval status. In our original data, 1 indicates “approved”, 0 means “not approved”.

4. Plot the histogram of installments by “approved” and “not approved”.

5. Illustrate with boxplot Fico score varies between “approved” and “not approved” borrowers.

2.2 Data Preprocessing and Model Construction: 5 points

2.2.1 Data Preprocessing and Splitting

The “purpose” feature in our dataset takes different nominal values, i.e., “credit_card”, “debt_consolidation”, “educational”, “major_purchase”, “small_business”, and “all_other”. In this question, please convert them with dummy variables so *sklearn* can recognize them. More specifically, it can be expanded into 6 different features, with each indicating whether a special purpose is served with a boolean value.

To evaluate the effectiveness of your method, the whole data should be split into two parts. In this question, a ratio of 70%: 30% is set between training and testing data, i.e., you need to randomly select 70% of the data as training, and leave the rest as testing data.

2.2.2 Training a Decision Tree

In this question, you are expected to construct a decision tree for decision making. Luckily, you do not need to develop everything from scratch. You should adopt the *DecisionTreeClassifier* included in *scikit-learn*. Note that figuring out how to use this implementation,its corresponding attributes and methods is a part of the assignment.

Two splitting criteria should be introduced, i.e.,

1. Information Gain,

2. Gini coefficient.

2.3 Performance Evaluation and Analysis: 5 points

To compare the performance of different approaches, we usually examine our model with testing data using various evaluation metrics. In this question, please show **confusion matrix, precision, recall, and f-score** on both Information Gain and Gini based methods.

Finally, to conclude your work, please use concise language to analyze the results based on your observation.