INFS770

Assignment 1.

Due 2/14/2022 + one-week grace period

**Please develop an ipython notebook titled 770\_hw1\_*yourlastname* to finish the following tasks (you will lose 1 point if you do not use the specified filename)**. You are required to create **an ipython notebook cell for each of the following questions**, where (C) indicates that you need to write code for the task, (O) indicates that you need write print function to show output, and (A) that you need to type your answers using Markdown text.

At the beginning of each cell, you need to indicate which task the cell is about. For example, in the cell related to task 2.1, you should first type “# Task 2.1: Import data”. **If you do not clearly label the cells, you will lose 1-2 points (out of 18 points).**

Task 1. Basic python programming:

1. Create an ipython notebook cell and develop the function ist\_ele\_idx(li). The argument of the function is a list, li. The return value of the function should be a list of tuples. Each tuple includes an element in the list li and the index of the element. For example, if you have a list of two string ["DSU","MSA"], this function should return a list of tuples [("DSU",0), ("MSA",1)]. **(C)**
2. Test the list\_ele\_idx function you just developed. Create an ipython notebook cell and copy the following two lines of code to it. **(C)(O)**

print('---------------- task 1.1 test case----------------')

print(list\_ele\_idx([5,3,2,6])) # you code should output: [(5, 0), (3, 1), (2, 2), (6, 3)]

1. Create a ipython notebook cell and write the function reverse\_key\_value(dict1) **(C)**

This function takes a dictionary as its argument. An example of such dictionary can be:

{“John”: “A”, “Sarah”: “A”, “Karen”: “B”, “Ken”: “C”}

This function needs to reverse each key/value pair in dict1 and returns a dictionary dict2 that includes the reversed key/value pairs:

{“A”:[“John”, “Sarah”], “B”:[“Karen”], “C”:[“Ken”]}

In Python, a dictionary includes key-value pairs. The keys must be unique, but we can have duplicate values. In the example above, two students “John” and “Sarah” both got “A”. Hence, when you reverse the key/value pairs, the keys in the new dictinary dict2 include the grades ("A", "B", and "C"), and the values need to be lists. For example, the key “A” corresponds to a list that includes two students “John” and “Sarah” because both John and Sarah got an A, while the key "B" corrsponds to a list with one element "Karen". You can assume that the values in the dictionary are hashable, i.e., they can be used as keys in the reversed dictionary dict2. As we discussed in the python tutoral, numbers, strings and tuples are hashable, while lists, dictionaries, and objects are un-hashable since they are mutable.

1. Test the reverse\_key\_value function you just developed. Create an ipython notebook cell and copy the following two lines of code to it. **(C)(O)**

print('---------------- task 1.3 test case----------------')

print(reverse\_key\_value({"John": "A", "Sarah": "A", "Karen": "B", "Ken": "C"}))# should print {'A': ['Sarah', 'John'], 'C': ['Ken'], 'B': ['Karen']}; As long as your program output these three items (the order of them may be different), you will get the points.

Task 2. Classification using scikit-learn: You probably want to finish the tasks by modifying the German credit notebook I used in week 3 lecture

[**Dataset**: magic04.csv]

<https://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope>

For this problem you will experiment with various classifiers provided as part of the **scikit-learn (sklearn)** machine learning module, as well as with some of its preprocessing and model evaluation capabilities.  The data is provided in a CSV formatted file with the first row containing the attribute names. Click “Data Folder”, and you can download the dataset to your PC by right-clicking and then selecting “save link as” the magic04.data link. The **description of the different fields** in the data is provided at <http://archive.ics.uci.edu/ml/machine-learning-databases/magic/magic04.names> . Please try to read the document and understand the case and the dataset.

In this assignment, you need to use the scikit-learn package, the main machine learning package in python to develop an ipython notebook. Please take a look at the scikit-learn home page (<http://scikit-learn.org/stable/index.html>) to get an overview of the package.

You want to make sure the scikit-learn package you are using is v20 or later versions.

1. You need to import data. **(C)**
2. In this dataset, the dependent variable is **class**. It includes two categories: g and h. g represents gamma (signal), and h hadron (background). Please insert a cell and print the value count of each category. **(C)(O)**
3. All the other variables are independent variables. Please insert a cell and print the histograms of the independent variables **(C)(O).**
4. Insert a cell and print the basic stats of each independent variable using the describe() method **(C)(O).**
5. Insert a cell and write code to split the dataset into training and validation sets (Please use 70%-30% split) **(C)**.
6. Insert a cell and describe the uses of validation (at least 3 uses). **(A)**.
7. Insert a cell. In this cell, you need to use scikit-learn’s logistic regression classifier ([http://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html#sklearn.linear\_model.LogisticRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression.score)) and fit a model using the training dataset **(C)**. Then you run the classifier on the validation set **(C)**. Print the validation dataset classification report and Area Under the Receiver Operating Characteristic Curve (ROC AUC)for the validation set. **(please google to find out how to get AUC using scikit-learn) (C)(O).**
8. Insert a cell and use your own language to describe the SVM algorithm (with at most 8 sentences) **(A).**
9. Insert a new cell. In this cell, you use the same training and validation dataset you obtained in task 2.5 to fit SVM classifiers (Please use the SVC function in scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>). You need to tune the SVM hyperparameter, **C (default = 1.0),** the Regularization parameter.You need to try each C in the list [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] – you must use a FOR loop. In each iteration, please print the validation set classification report and AUC. **(C)(O).**
10. Insert a new cell. In this cell, please first tell me which C in the list [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] gives you the optimal SVM classifier with respect to AUC **(A)**. Then, please use your own language (with at most 4 sentences) to discuss what this hyperparameter C means **(A)**.

**Please click Cell-> Run All to run the code and submit an iPython notebook that includes required code, outputs and markdown cells to “Assessment” -> “Dropbox” -> “Assignment 1”. If you do not do “Run All”, you will lose 1 out of 18 points.**