DEPARTMENT *of* COMPUTING
College of Business & Technology
EAST TENNESSEE STATE UNIVERSITY

CSCI 5260 – Artificial Intelligence

Project 4 – Guess What?

Description

Background

You now work for a prominent winery that has hired you to predict the quality of the wine they produce, based on already-collected data. The winery collects two main sets of data: one on the white wines they produce (winequality-white.csv, n=4898), and one on the red wines they produce (winequality-red.csv, n=1599).

Data Description

Data are in two files: winequality-white.csv (4898 rows x 12 columns) and winequality-red.csv (1599 rows x 12 columns).

Input Variables

These input variables are based on physiochemical tests that occur regularly.

1. fixed acidity — Range: 3.8 to 15.9
2. volatile acidity — Range: 0.08 to 1.58
3. citric acid — Range: 0 to 1.66
4. residual sugar — Range: 0.9 to 65.8
5. chlorides — Range: 0.009 to 0.611
6. free sulfur dioxide — Range: 1 to 289
7. total sulfur dioxide — Range: 6 to 440
8. density — Range: 0.98711 to 1.03898
9. pH — Range: 2.72 to 4.01
10. sulphates — Range: 0.22 to 2.0
11. alcohol — Range: 8 to 14.9

Output Variable

12. quality — Range: 0 to 10

Part 1 – Unsupervised Learning

Coding and Analysis Requirements

Create a file called **project4_clustering.py**. Write a program that does the following:

1. Read winequality-white.csv and winequality-red.csv into two separate Pandas data frames.
   a. Reference: https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html#pandas.read_csv
2. Create a `target_white` data frame and a `target_red` data frame by selecting the data's last column (the 'quality' column) and storing it there. For example: `target_red = data_red['quality']`. Be sure to use the `drop` function after you have copied it to remove it from the original data.

    a. See https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html?highlight=drop#pandas.DataFrame.drop
3. **Using sklearn.cluster.KMeans**, run the k-means clustering algorithm on the white wines and the red wines. **You should use 11 clusters** because we know there are 11 quality metrics (labeled 0-10).
    a. See https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.
    b. Note that the result of the fit function returns a data structure containing the following:
        i. **cluster_centers_*ndarray of shape (n_clusters, n_features)***
            1. Coordinates of cluster centers. If the algorithm stops before fully converging (see tol and max_iter), these will not be consistent with labels_.
        ii. **labels_*ndarray of shape (n_samples,)***
            1. Labels of each point
        iii. **inertia_*float***
            1. Sum of squared distances of samples to their closest cluster center.
        iv. **n_iter_*int***
            1. Number of iterations run.
4. **Analyze the results for the white wine and the red wine examples**. Add a discussion to the Project4.docx writeup document. Remember that the cluster labels ARE NOT predictions of quality. The label is simply the grouping to which an example belongs. To analyze this you should:
    a. Write a procedure that determines the **quality for each cluster** by averaging the qualities of all items in that cluster.
    b. This is OPEN-ENDED but you should **use this information to plot the quality values for each cluster**. Include these plots in your **Project4.docx writeup**.
    c. Does the data indicate 11 clearly-defined quality metrics? Explain why it does or does not.

Part 2 – Supervised Learning

Coding and Analysis Requirements

Create a file called **project4_ml.py**. Using the same data set as above, do the following.

1. **Combine the data sets into a single data set**.
    a. To do this, add a column called "type" to each data frame.
    b. Set red wine as type 0 and white wine as type 1.
2. **Split the data into train and test sets**.
3. **Train and Test two** of the following learning algorithms from the scikit-learn library. Be sure to use the same train and test data for each.
    a. Decision Tree Classifier - https://scikit-learn.org/stable/modules/tree.html#classification
    b. Linear Regression Classifier - https://scikit-learn.org/stable/modules/linear_model.html#generalized-linear-regression
    c. Gaussian Naïve Bayes Classifier - https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes
    d. Nearest Neighbor Classifier - https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification
    e. Support Vector Machine - https://scikit-learn.org/stable/modules/svm.html#classification
4. **Analyze the results** by showing the following (add your analysis to Project4.docx):
    a. Which classification method performed better?
        i. You should measure the number of true negatives, true positives, false negatives, and false positives. If you want to drill down, it might be helpfult to track this by class.
    b. Based on the results, what could you do to improve performance?
        i. Keep in mind the ideas of feature engineering and feature scaling as you respond to this.

Part 3 – Deep Learning

Coding and Analysis Requirements

Create a file called **project4_nn.py**.

1. Use the combined data set from Part 2, and the same train and test sets.
2. Train and test a Multilayer Perceptron Neural Network Classifier (MLPClassifer).
    a. https://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification
3. **Analyze the results** (recording the analysis in **Project4.docx**) by:
    a. Showing true negatives, true positives, false negatives, and false positives. If you want to drill down, you might track this by class to better analyze results.
    b. Comparing the results to the models trained above.

Submission and Due Date

Interim Submission

An interim submission is required for this assignment to the **Project 4 Interim** dropbox. You should briefly detail the progress you've made and note any problems or questions you have. Submit a screenshot of one portion of the code that you have completed.

Failure to submit the interim submission ON TIME will result in the loss of 10% from the final Project 4 grade. Until you have feedback on this dropbox, you will not be allowed to submit the final solution. This is to encourage you to work on this early!

Project 4 interim submission is due to the D2L dropbox at or before Monday, April 12, 2021 at 11:59 PM

Final Submission

Submit all code and documentation, zipped into an archive: **Surname_Project4.zip**. The folder should be self-contained in a way that allows the code to run. You can assume that I have all necessary libraries installed.

Your archive should contain the following files:

1. project4_clustering.py
2. project4_ml.py
3. project4_nn.py
4. winequality-red.csv
5. winequality-white.csv
6. The Project4.docx Word Document containing your analysis.

Project 4 is due to the D2L dropbox at or before Monday, April 19, 2021 at 11:59 PM

The rubric appears on the following page.

Rubric

A letter grade will be assigned to each of the following, and will translate to a numeric grade based on the scale in the syllabus, and averaged into an overall percentage. As a reminder, anything below a C translates to an F by University Graduate School policy. It is provided here to appropriately reflect each level.

For source code, please add comments so I can understand what is going on. Believe it or not, some student code is difficult to read. :D

	A		B		C		D		F		Zero	
	A	A-	B+	B	B-	C+	C	C-	D+	D	F	0
Part 1 – Unsupervised Learning												
Q1/2 – Data Frames and Targets												
Q2 – sklearn.cluster.KMeans												
Q4a – Cluster Quality												
Q4b – Plots of each cluster's quality values												
Q4c – Analysis of quality												
Part 2 – Supervised Learning												
Q1 – Combined Data Sets												
Q2 – Test and Train Split												
Q3-1 – First Classifier												
Q3-2 – Second Classifier												
Q4a – Performance Analysis												
Q4b – Improvement Suggestions												
Part 3 – Deep Learning												
Q2 – MLP Classifier												
Q3a – Analysis of test set												
Q3b – Comparison to classifers from Part 2.												