SOFT8032 - Programming for Data Analytics

Third Assessment

December 17th 2023

1 Read Carefully

This project contributes 50% in your final mark. This is an individual project and has to be all done by yourself. You are not allowed to disclose your code to anyone else.

You may be called for a zoom or Teams meeting to explain different parts of your submission, if needed.

Please use the discussion forum on Canvas if you have questions or you can email me.

A template file has been provided for you, and you are required to answer the questions within this template. Be sure to update the comments at the top of the file with your own details.

It's important to note that any attempt towards a final solution will earn you marks.

Important: The solution must be implemented using functions.

1.1 Dataset Overview

For this project we are going to perform a number analytical tasks on the **weather.csv** files.

1.2 Project Specification

The objective of this project is to provide an insight into the underlying pattern of the dataset such as relationship between features, feature prediction and etc. Please perform the following tasks:

1. Find the number of unique locations present in the dataset. Utilize an appropriate visualization technique to display the five locations with the fewest records or rows. Present the percentage for each section and perform all necessary data cleaning.

2. *Typically, when the air pressure is high, it usually indicates stable weather conditions with less chance of rain. High-pressure systems are associated with descending air, which suppresses cloud formation and precipitation.*

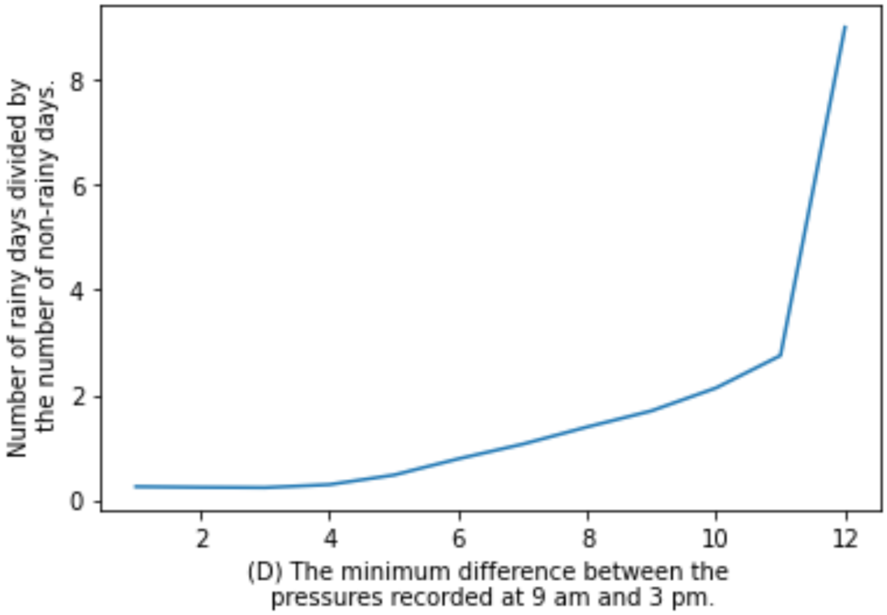

Figure 1: Result of Task 2

Conversely, low-pressure systems often lead to more unstable atmospheric conditions, which can result in cloud formation and precipitation, making rain more likely later on and possibly tomorrow.

The dataset contains data regarding 'Pressure' recorded at two distinct times: 9 am and 3 pm. The objective here is to validate the prior assertion about pressure's effect on subsequent rainfall (tomorrow's rain). The task seeks to determine if a decrease in pressure might lead to increased rainfall chance on the following day. To achieve this, rows with the minimum difference D are extracted, and the number of rainy days is divided by the number of non-rainy days.

This process will be repeated 12 times for D in the range [1, 12] to generate Figure 1. Please use the comment section to discuss the results obtained from this analysis.

3. Create a sub-DataFrame with the following attributes: ['WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Temp9am', 'Temp3pm', 'RainTomorrow']. Consider 'RainTomorrow' as the class attribute and the rest as ordinary attributes.

   For this task, utilize a supervised learning algorithm (decision tree classifier) and experiment with different maximum depths ranging from 1 to 35. Measure the importance level of each attribute for each learning iteration (each maximum depth). Employ an appropriate visualization technique to illustrate the impact of varying maximum depths on the importance levels of features. Please provide comments to explain your findings. Since this question does not focus on the learning accuracy, there is no need to split the data into training and test. Use all data as training.

4. Create a sub-dataset with the attributes: 'WindDir9am', 'WindDir3pm', 'Pressure9am', 'Pressure3pm', and 'RainTomorrow'. Run a classification algorithm twice using the following steps:

   (a) Utilize 'Pressure9am' and 'Pressure3pm' as ordinary attributes, and 'RainTomorrow' as the class attribute. Split the dataset into 33% test data and the remaining as training data. Calculate the accuracy for both the test and training datasets after training the model.

   (b) Use 'WindDir3pm' and 'WindDir9am' as ordinary attributes, and 'RainTomorrow' as the class attribute. Split the dataset into 33% test data and the rest as training data. Calculate the accuracy for both the test and training datasets after training the model.

   Explain your reasoning in the comments below the function to determine which model would be better for predicting 'RainTomorrow'.

5. Create a sub-DataFrame containing the attributes: *RainTomorrow*, *WindDir9am*, *WindGustDir*, and *WindDir3pm*.

   The *WindDir9am*, *WindGustDir*, and *WindDir3pm* attributes denote wind directions, represented by either one letter (e.g., 'W' for West) or two letters (e.g., 'NW' for North West) or three letters (e.g., 'WNW' for West North-West). Exclude rows where at least one of the attributes, *WindDir9am*, *WindGustDir*, or *WindDir3pm*, contains three letters. Perform the following training operations with the remaining data:

   (a) Apply the DecisionTreeClassifier with 10 different depths ranging from 1 to 10. For each depth, conduct cross-validation and store the averages of training and test accuracy obtained after cross-validation in a list.

   (b) Apply the KNeighborsClassifier using 10 different values for neighbors ranging from 1 to 10. For each neighbor value, perform cross-validation and store the averages of training and test accuracy obtained after cross-validation in a list.

   The cross-validation should consist of 5 folds with a 20% allocation for the test data. Consequently, you'll have four lists, each containing 10 values. The first list contains the training accuracy for the Decision Tree Classifier. The second list contains the test accuracy for the Decision Tree Classifier. The third list contains the training accuracy for the KNeighborsClassifier, and the last list contains the test accuracy for the KNeighborsClassifier.

Generate two plots: one for the training and test accuracy of the Decision Tree Classifier and another for the training and test accuracy of the KNeighborsClassifier. Display these two plots on a single canvas, one on top of the other.

Finally, analyze the results comprehensively, interpreting the outcomes. Explain which values for depth and number of neighbors are more optimal for these two machine learning algorithms, and provide reasoning for your findings.

6. Create a dataset with 11 attributes using the following columns: ['MinTemp', 'MaxTemp', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Rainfall', 'Temp9am', 'Temp3pm']. Ensure that there are no non-numerical values present in these columns.

   Apply an unsupervised learning algorithm, specifically K-Means, to this dataset as follows:

   (a) Apply K-Means clustering on the dataset using various numbers of clusters: [2, 3, 4, 5, 6, 7, 8].

   (b) Utilize an appropriate visualization method to determine the optimal number of clusters based on all attributes.

   (c) In another figure, present the entire dataset using a scatter plot. Assign a different color to each cluster. Note that the number of clusters should be determined from step (b).

   Provide an explanation as comment below the function based on the findings.

7. Select and execute an analytical task employing a Machine Learning algorithm of your preference. Ensure that the chosen task aligns with the dataset, offering practical and pertinent insights. Use the comment section to explain upon the task's concept and its implementation and usefulness.

Note that visualization plots need to have proper labels and annotations. Note: Lack comment and interpretation will attract penalty.

1.3 Submission and Deadline

Please use the python file template that is provided for you and complete your project in that file. Each task needs to be implemented as a separated function with interpretation as a comment below the function if needed.

Please write your name and student ID as a comment in the designated area in the provided template python file.

The deadline for this project is 23.59 Sunday 17th Dec 2023. Penalties will be applied to late project submissions as per MTU Marks and Standards.

1.4 Rubric

This rubric is subject to change.

1. Correct task implementation (model training, accuracy reporting, visualization if needed etc). (100%)
2. Relatively correct task implementation (model training, accuracy reporting, visualization if needed etc). (70%)
3. Partly correct task implementation (model training, accuracy reporting, visualization if needed etc). (40%)
4. Wrong task implementation. (0%)