EGR 598: Machine Learning and Artificial Intelligence (Fall 2023)

Instructor: Shenghan Guo

Assignment 1

This assignment is worth 200 points. It will be due on Thursday, 9/28/2022, and submitted through Canvas. Code should be written in Python. Other programming language can be used upon instructor's approval. A written report should be submitted as a separate document along with the code through Canvas. If you use Jupyter Notebook, then you may submit a single ".ipynb" file through Canvas.

Data: A dataset, "Raison_Dataset.csv", is provided to you. Data description can be found in "Raison_Dataset.txt". Please read through the document and then use the data to do the following tasks.

Note: You may consider each column a variable. The *input attributes* include: "Area", "MajorAxisLength", "MinorAxisLength", "Electricity", "ConvexArea", "Extent", "Perimeter". The samples are independent and identically distributed (iid).

Part 1 (70 pts)

1. What is the number of *classes* in this dataset? (2 pts)

2. Calculate the *log odds* for the data. Write the *discriminant function* in terms of the log odds. (6 pts)

3. Assume that the input attributes are *multivariate normal*. Further assume that the input attributes in each class follow a different multivariate distribution. Calculate the mean vector and covariance matrix for the input attributes in each class. (Hint: consider your answer in 1. You should obtain this many sets of mean vector and covariance matrix.) (8 pts)

4. Given your answer in 3, generate 10 samples from each of the multivariate distributions. (Hints: the number of samples generated should be 10 times number of classes.) (10 pts)

5. Given the assumption that input attributes are multivariate normal, visualize the joint distribution of "MajorAxisLength" and "MinorAxisLength" for each class. Based on the "multivariate normal" assumption, do you think that "MajorAxisLength" and "MinorAxisLength" are both *univariate normal*, and why? (Hint: use your results from 3 and visualize the parametric form of distribution. Create grids for $[0, 800] \times [0, 800]$ for 3D plots.) (10 pts)

6. Given your answers in 3, write the functional form of the *likelihood ratio*. You may define notations for the mean and covariance of each class. (6 pts)

7. Given your answers in 3 and 6, write the *discriminant function* for each class. Then, calculate the discriminant functions it for each sample point and label each of them with the class name. (Hint: see Eq. (4.20) in textbook. The "label" here is based on your calculated discriminant. You may store the labels in an Excel or .csv file.) (10 pts)

8. Given your answers in 3 and 6, if pooling the covariance of all classes, write the *discriminant function* for each class. Then, calculate the discriminant functions it for each sample point and label each of them with the class name. (Hint: see Eq. (5.21) and (5.22) in textbook. The "label" here is based on your calculated discriminant. You may store the labels in an Excel or .csv file.) (10 pts)

9. Use a *confusion matrix* to show the classification results with the *discriminant function* in 7 and 8, respectively. Calculate the classification accuracy for both and compare the results. Briefly describe your findings. (Hint: you will obtain 2 confusion matrices, one for the result in 7 and the other for 8.) (8 pts)

Part 2 (60 pts)

Do 4-fold cross validation for the dataset and perform classification analysis: (1) randomly shuffle the samples, (2) partition the data into 4 folds, (3) choose 3 out of the 4 folds as *training data* and the rest 1 as *testing data* (you can do this for 4 times by choosing 3 different folds each time).

For each of the 4 replicates, do the following:

1. Assume that the input attributes are *multivariate normal*. Calculate the mean vector and covariance matrix for the input attributes in each class using the training data. (10 pts)

2. Given your answers in 1, calculate the *discriminant function* for the testing data. Then label each testing sample with the class name. Finally, create a *confusion matrix* to show the classification result for testing data. (Hint: You may store the labels in an Excel or .csv file.) (15 pts)

3. Given your answers in 1, if pooling the covariance of all classes, calculate the *discriminant function* for the testing data. Then label each testing sample with the class name. Finally, create a *confusion matrix* to show the classification result for testing data. (Hint: You may store the labels in an Excel or .csv file.) (15 pts)

4. For the *discriminant functions* in 2 and 3, respectively, calculate the average *false positive rate*, *false negative rate*, *true positive rate*, and *true negative rate* for the classification results throughout the 4 replicates that you have completed. (Hint: you will get four rates for each classification method.) (15 pts)

5. Briefly describe the performance of each discrimination method and identify the best one for this dataset based on the average performance across 4-fold cross validation. (5 pts)

Part 3 (70 pts)

For this part, take "Area", "MajorAxisLength", "MinorAxisLength", "Electricity", "Extent", "Perimeter" as independent variables, and "ConvexArea" as dependent variables.

1. Visualize "ConvexArea" against each independent variable and describe the trend and patterns in your plots. (Hints: you will get 6 plots, each with "Area" as the vertical axis and an independent variable as the horizontal axis.) (8 pts)

2. Use the first 600 samples in the dataset as the training data and the rest as the testing data. Calculate the correlation matrix for all dependent and independent variables for the training data. Based on the correlation matrix, identify which independent variables have major impact to the dependent variable. Does the impact imply a causal relationship and why? (Hint: Save the correlations in an Excel or .csv file.) (8 pts)

3. Use Python to fit a linear regression model using the training data. Summarize the model coefficients. Based on the coefficients, which independent variables have more impact to the dependent variable? (10 pts)

4. Use the model fitted in 3 to make predictions for testing data. Calculate the *mean squared error* for the testing samples with respect to the predictions. Do you think the model has a good prediction performance? (Hint: input the testing samples of independent variables into your fitted model and then evaluate the prediction against the true sample values of dependent variable.) (8 pts)

5. Based on result in 3, do you think that the independent variables are mutually linearly independent? What's the influence on the linear regression model with the appearance of linear dependence among independent variables? (5 pts)

6. Perform *Principal Component Analysis* on the training data matrix of independent variables. Show the variance explained by each principal component. (6 pts)

7. Visualize a *Pareto chart* for the variance explained by each principal component. (10 pts)

8. Take the first 3 principal components from 7 and fit a multivariate regression model. Do prediction with the model for testing samples and calculate the mean squared error. (Hint: You need to transform the testing data to principal components as well.) (10 pts)

9. Give a practical scenario when you will use *Principal Component Analysis* to reduce the data dimensionality before fitting a regression model; give a practical scenario when you will NOT use *Principal Component Analysis* to reduce the data dimensionality before fitting a regression model. (Hint: you can name any data source and/or situations, which is not necessarily related to the Raison dataset.) (5 pts)