Lab5: Data classification using Bayes Classifier with Gaussian Mixture Model (GMM); Regression using Linear Regression and Polynomial Regression

Deadline for submission: 29th September 2022, 11:55 PM

PART-A:

You are given the Steel Plates Faults Data Set as a csv file (`SteelPlateFaults-2class.csv`) in Assignment4 (Lab4). The dataset used for this assignment contains features extracted from the steel plates of types A300 and A400 to predict whether the image of the surface of the steel plate contains two types of faults such as Z_Scratch and K-Scratch. It consists of 1119 tuples each having 27 attributes which are indicators representing the geometric shape of the fault. The last attribute (28th attribute) for every tuple signifies the class label (0 for K_Scratch fault and 1 for Z_Scratch fault). It is a two-class problem. Use the same train data file (**`SteelPlateFaults-train.csv`**) and test data file (**`SteelPlateFaults-test.csv`**), used in Assignment4, in this assignment also.

1. Build a Bayes classifier with multi-modal Gaussian distribution (GMM) with *Q* Gaussian components (modes) as class conditional density for each class on the training data **`SteelPlateFaults-train.csv`**. Build a GMM with *Q* components for class1 and build a GMM with *Q* components for class2. Classify every test tuple using the **Bayes classifier with GMM** for the different values of *Q*=2, 4, 8, and 16. Perform the following analysis:
   a. Find **confusion matrix** for each *Q*
   b. Find the **classification accuracy** for each *Q*. Note the value of *Q* for which the accuracy is high.

   Note: Remove the attributes **X_Minimum**, **Y_Minimum**, **TypeOfSteel_A300** and **TypeOfSteel_A400** from both training and test data set. Because correlation of **X_Minimum** & **X_Maximum** is 1 and correlation of **Y_Minimum** & **Y_Maximum** is also 1. Also, correlation between **TypeOfSteel_A300** and **TypeOfSteel_A400** is -1. This indicates the variance and covariance of these pair of attributes are the same. This leads to the covariance matrix singular. Due to this, the inverse of the covariance matrix cannot be computed, hence, the likelihood cannot be computed. To avoid this issue, we need to remove the above-mentioned attributes. Bayes classifier is now built using the data with only 23 attributes.

2. Tabulate and compare the best result of the KNN classifier, the best result of the KNN classifier on normalized data, the result of the Bayes classifier using unimodal Gaussian density (all from Assignment-4), and Bayes classifier using GMM.

Note:
Use the function "`mixture.GaussianMixture`" from scikit-learn to build GMM.

```
GMM = mixture.GaussianMixture(n_components=Q, covariance_type='full')
GMM.fit(x)
```

Compute the weighted log probabilities for each sample using `GMM.score_samples(x)`.

Compute accuracy using `metrics.accuracy_score`.

PART B:

You are given a data file `abalone.csv`. Abalones are marine snails. The dataset has been prepared with the aim of making age predictions easier. Customarily, the age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope. But it is a tedious and time-consuming task. Therefore, other measurements, which are easier to obtain, are used to predict age.

Attribute information:

Given is the attribute name, attribute type, the measurement unit, and a brief description. The number of rings is the value to predict: either as a continuous value or as a classification problem.

Name / Data Type / Measurement Unit / Description

1. **Length** / continuous / mm / Longest shell measurement
2. **Diameter** / continuous / mm / Diameter of the shell calculated as perpendicular to length
3. **Height** / continuous / mm / Height of the shell with meat in shell
4. **Whole weight** / continuous / grams / Weight of whole abalone
5. **Shucked weight** / continuous / grams / Weight of meat
6. **Viscera weight** / continuous / grams / Gut-weight (after bleeding)
7. **Shell weight** / continuous / grams / Weight of the shell after being dried
8. **Rings** / integer / -- / Number of rings in a shell. (Adding 1.5 to the number of rings gives the age of abalone in years)

Write a python program to split the data from `abalone.csv` into train data and test data. Train data contain 70% of tuples and test data contain the remaining 30% of tuples. Save the train data as `abalone-train.csv` and save the test data as `abalone-test.csv`.

Note: Use the command `train_test_split` from scikit-learn given below to split the data (keep `random_state=42` to get the same random values for every student).

1. Use the attribute which has the highest Pearson correlation coefficient with the target attribute **Rings** as an input variable and build a simple linear (straight-line) regression model to predict rings. (Prerequisite: calculate the Pearson correlation coefficient of every attribute with the target attribute rings.)
   - ***a.*** Plot the best fit line on the training data where the x-axis represents ***the chosen attribute*** value and the y-axis represents **Rings**.
   - b. Find the prediction accuracy on the training data using root mean squared error.
   - c. Find the prediction accuracy on the test data using root mean squared error.
   - d. Plot the scatter plot of actual **Rings** (x-axis) vs predicted **Rings** (y-axis) on the test data. Draw inferences from the scatter plot.

2. Build a multivariate (multiple) linear regression model to predict **Rings**. All the attributes other than the target attribute should be used as input to the model.
   - a. Find the prediction accuracy on the training data using root mean squared error.
   - b. Find the prediction accuracy on the test data using root mean squared error.
   - c. Plot the scatter plot of actual **Rings** (x-axis) vs predicted **Rings** (y-axis) on the test data. Draw inferences from the scatter plot.

3. Use the attribute which has the highest Pearson correlation coefficient with the target attribute **Rings** as input and build a simple nonlinear regression model using polynomial curve fitting to predict **Rings**.
    a. Find the prediction accuracy on the training data for the different values of degree of the polynomial (*p* = 2, 3, 4, 5) using root mean squared error (RMSE). Plot the bar graph of RMSE (y-axis) vs different values of degree of the polynomial (x-axis).
    b. Find the prediction accuracy on the test data for the different values of degree of the polynomial (*p* = 2, 3, 4, 5) using root mean squared error (RMSE). Plot the bar graph of RMSE (y-axis) vs different values of degree of the polynomial (x-axis).
    c. Plot the best fit curve using the best fit model on the training data where the x-axis represents ***the chosen attribute*** value and the y-axis is **Rings**.
    (**Note**: The best fit model is chosen based on the *p*-value for which the test RMSE is minimum.)
    d. Plot the scatter plot of the actual number of **Rings** (x-axis) vs the predicted number of **Rings** (y-axis) on the test data for the best degree of the polynomial (*p*). Comment on the scatter plot and compare it with that of in 1(d).

4. Build a multivariate nonlinear regression model using polynomial regression to predict **Rings**. All the attributes other than the target attribute should be used as input to the model.
    a. Find the prediction accuracy on the training data for the different values of degree of the polynomial (*p* = 2, 3, 4, 5) using root mean squared error (RMSE). Plot the bar graph of RMSE (y-axis) vs different values of degree of the polynomial (x-axis).
    b. Find the prediction accuracy on the test data for the different values of degree of the polynomial (*p* = 2, 3, 4, 5) using root mean squared error (RMSE). Plot the bar graph of RMSE (y-axis) vs different values of degree of the polynomial (x-axis).
    (**Note**: The best fit model is chosen based on the *p*-value for which the test RMSE is minimum.)
    c. Plot the scatter plot of the actual number of **Rings** (x-axis) vs the predicted number of **Rings** (y-axis) on the test data for the best degree of the polynomial (*p*). Comment on the scatter plot and compare it with that of in 1(d).

Hints and code snippets:

- For linear regression use:

```
from sklearn.linear_model import LinearRegression

reg = LinearRegression().fit(X, y)

        #Input arguments: X: Input variable(s) of training data
        and y: target values

y_pred = predict(X) #Predict using the linear model.
```

- For polynomial curve fitting and regression use:

```
from sklearn.preprocessing import PolynomialFeatures

poly_features = PolynomialFeatures(p) #p is the degree

x_poly = poly_features.fit_transform(X)

regressor = LinearRegression()

regressor.fit(x_poly, y)

        #Input arguments: x_poly: Polynomial expansion of input
        variable(s) of training data and y: target values

y_pred = regressor.predict(X)
```

Instructions:

- **Your python program(s) should be well commented on.**

In case the program is found to be copied from others, both the person who copied and who helped to copy will get zero as a penalty.