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

Deadline for submission: 5 November 2020, 10:00 PM

PART-A: You are given the **Seismic-Bumps Data Set** as a csv file (`seismic-bumps.csv`). This dataset contains recorded features from the seismic activity in the rock mass and seismoacoustic activity with the possibility of rockburst occurrence to predict the hazardous and non-hazardous state. It consists 2584 tuples each having 19 attributes (ignore the attributes `nbumps` to `nbumps89` as most of their values are 0). The last attribute for every tuple signifies the class label (0 for hazardous state and 1 for non-hazardous state). It is a two class problem. Consider the **`seismic-bumps-train.csv`** and **`seismic-bumps-test.csv`** obtained from the **Assignment-4** for training and testing.

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 **`seismic-bumps-train.csv`**. (Build a GMM with Q components for class1 and build a GMM with Q components for class2) Classify every test tuple using **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.

2. Tabulate and compare the best result of KNN classifier, best result of KNN classifier on normalised data, result of 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 with data file **`atmosphere_data.csv`** that contains the readings from various sensors installed at 10 locations around Mandi district. These sensors measure the different atmospheric factors like *temperature*, *humidity*, *atmospheric pressure*, *amount of rain*, *average light*, *maximum light* and *moisture content*. The goal of this dataset is to model the atmospheric temperature.

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

1. Build the simple linear regression (straight-line regression) model to predict *temperature* given *pressure*.

   a. Plot the best fit line on the training data where x-axis is *pressure* value and y-axis is *temperature*.

   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 *temperature* (x- axis) vs predicted *temperature* (y-axis) on the test data. Comment on the scatter plot.

2. Build the simple nonlinear regression model using polynomial curve fitting to predict *temperature* given *pressure*.

   a. Find the prediction accuracy on the training data for the different values of degree of 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 polynomial (x-axis).

   b. Find the prediction accuracy on the test data for the different values of degree of 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 polynomial (x-axis).

   c. Plot the best fit curve using best fit model on the training data where x-axis is pressure value and y-axis is temperature.

      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 actual temperature (x-axis) vs predicted temperature (y-axis) on the test data for the best degree of polynomial (p). Comment on the scatter plot and compare with that of in 1(d).

Note:

Simple and Multiple Linear Regression:

Import the **`LinearRegression`** from **`sklearn.linear_model`**

Code snippet for prediction using linear regression:

```
regressor = LinearRegression()
```

`regressor.fit(x, y)`: `x` is set of univariate or multivariate training data used for building simple of multiple linear regression. `y` is corresponding dependent variable.

```
y_pred = regressor.predict(x)
```

Polynomial Curve Fitting and Polynomial Regression*:*

Import the **`PolynomialFeatures`** from **`sklearn.preprocessing`**

Code snippet for prediction using linear regression:

```
polynomial_features = PolynomialFeatures degree=p)
```

`x_poly = polynomial_features.fit_transform(x)`: `x` is set of univariate or multivariate training data used for building simple of multiple polynomial regression.

```
regressor = LinearRegression()
```

`regressor.fit(x_poly, y)` : `x_poly` is set of polynomial expansions (monomials of polynomial up to degree p) training data used for building simple of multiple linear regression. `y` is corresponding dependent variable.

```
y_pred = regressor.predict(x)
```

Instructions:

- **Your python program(s) should be well commented. Comment section at the beginning of the program(s) should include your name, registration number and mobile number.**
- **The python program(s) should be in the file extension .py**
- **Report should be strictly in PDF form. Write the report in word or latex form and then convert to PDF form. Template for the report (in word and latex) is uploaded.**
- **First page of your report must include your name, registration number and mobile number. Use the template of the report given in the assignment.**
- **Upload your program(s) and report in a single zip file. Give the name as <roll_*number*>_Assignment4.zip. Example: *b19001*_Assignment4.zip**
- **Upload the zip file in the link corresponding to your group only.**

In case the program found to be copied from others, both the person who copied and who help for copying will get zero as a penalty.