Lab4: Data Classification using K-Nearest Neighbor Classifier and Bayes Classifier with Unimodal Gaussian Density

Deadline for submission: 26 October 2020, 10:00 PM

You are given the **Seismic-Bumps Data Set** as a csv file (`seismic-bumps.csv`). The data describe the problem of high energy (higher than 10^4 J) seismic bumps forecasting in a coal mine. This data is collected from two of longwalls located in a Polish coal mine. Mining activity was and is always connected with the occurrence of dangers which are commonly called mining hazards. A special case of such threat is a seismic hazard which frequently occurs in many underground mines. Seismic hazard is the hardest detectable and predictable of natural hazards and in this respect it is comparable to an earthquake. More and more advanced seismic and seismoacoustic monitoring systems allow a better understanding rock mass processes and definition of seismic hazard prediction methods. Accuracy of so far created methods is however far from perfect. Complexity of seismic processes and big disproportion between the number of low-energy seismic events and the number of high-energy phenomena (e.g. > 10^4 J) causes the statistical techniques to be insufficient to predict seismic hazard.

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. 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. Other attributes are input features. For more information refer [1].

Attribute Information:

1. `seismic`: result of shift seismic hazard assessment in the mine working obtained by the seismic method (1 - lack of hazard, 2 - low hazard, 3 - high hazard, 4 - danger state);
2. `seismoacoustic`: result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method (1 - lack of hazard, 2 - low hazard, 3 - high hazard, 4 - danger state);
3. `shift`: information about type of a shift (W - coal-getting, N -preparation shift);
4. `genergy`: seismic energy recorded within previous shift by the most active geophone (GMax) out of geophones monitoring the longwall;
5. `gpuls`: a number of pulses recorded within previous shift by Gmax;
6. `gdenergy`: a deviation of energy recorded within previous shift by GMax from average energy recorded during eight previous shifts;
7. `gdpuls`: a deviation of a number of pulses recorded within previous shift by GMax from average number of pulses recorded during eight previous shifts;
8. `ghazard`: result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method based on registration coming from GMax only;
9. `nbumps`: the number of seismic bumps recorded within previous shift;
10. `nbumps2`: the number of seismic bumps (in energy range [10^2, 10^3)) registered within previous shift;
11. `nbumps3`: the number of seismic bumps (in energy range [10^3, 10^4)) registered within previous shift;
12. `nbumps4`: the number of seismic bumps (in energy range [10^4, 10^5)) registered within previous shift;

13. `nbumps5`: the number of seismic bumps (in energy range [10^5, 10^6)) registered within the last shift;
14. `nbumps6`: the number of seismic bumps (in energy range [10^6, 10^7)) registered within previous shift;
15. `nbumps7`: the number of seismic bumps (in energy range [10^7, 10^8)) registered within previous shift;
16. `nbumps89`: the number of seismic bumps (in energy range [10^8, 10^10)) registered within previous shift;
17. `energy`: total energy of seismic bumps registered within previous shift;
18. `maxenergy`: the maximum energy of the seismic bumps registered within previous shift;
19. `class`: the decision attribute - '1' means that high energy seismic bump occurred in the next shift ('hazardous state'), '0' means that no high energy seismic bumps occurred in the next shift ('non-hazardous state').

---

1. Write a python program to split the data of each class from `seismic-bumps.csv` into train data and test data. Train data contain 70% of tuples from each of the class and test data contain remaining 30% of tuples from each class. Save the train data as **`seismic-bumps-train.csv`** and save the test data as **`seismic-bumps-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).

   ```
   [X_train, X_test, X_label_train, X_label_test] = train_test_split(X, X_label, test_size=0.3, random_state=42, shuffle=True)
   ```

   Classify every test tuple using **K-nearest neighbor (KNN)** method for the different values of *K*=1, 3, and 5. Perform the following analysis:

   a. Find **confusion matrix** (use '`confusion_matrix`' function from `scikit-learn`) for each *K*.

   b. Find the **classification accuracy** (You can use '`accuracy_score`' function) for each *K*. Note the value of *K* for which the accuracy is high.

2. Normalize all the attributes (except class attribute) of **`seismic-bumps-train.csv`** using Min-Max normalization to transform the data in the range [0-1]. Save the file as **`seismic-bumps-train-Normalised.csv.`** Normalize the test dataset using the **minimum and maximum values of train dataset** and save the test data as **`seismic-bumps-test-normalised.csv`**.

   Classify every test tuple using **K-nearest neighbor (KNN)** method for the different values of *K*=1, 3, and 5. Perform the following analysis:

   a. Find confusion matrix for each *K*.

   b. Find the classification accuracy for each *K*. Note the value of *K* for which the accuracy is high.

3. Build a **Bayes classifier** (with unimodal Gaussian density used to model the distribution of the data) on the training data **`seismic-bumps-train.csv`.** Test the performance on **`seismic-bumps-test.csv`** and give confusion matrix and accuracy.

   Note: Compute mean vector and covariance matrix from the training data of each classes separately. Use them to compute likelihood for a class. For computing likelihood use the expression of multivariate Gaussian density. (Do not use Gaussian Naïve Bayes function from sklearn).

4. Tabulate and compare the best result of KNN classifier, best result of KNN classifier on normalised data, and result of Bayes classifier using unimodal Gaussian density.

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.