[Project Code: PPNB]

Polydipsia Prediction using Gaussian Naive Bayes Classifier Learning Model

Project Duration : 22-Jan-2023 ~~ 11-Feb-2023
Submission Information : (via) CSE-Moodle

Objective:

Dr. Strange's Lab has decided to build a machine learning model for predicting polydipsia, a medical disease of excessive thirst, for women. The model will take the various diagnostic results (like hormone level, blood pressure etc.) and patient's conditions (like age, BMI, lineage factor) as input features and predict whether she has polydipsia. Moreover, they have decided to use a probabilistic machine learning model as they want the model to output a prediction probability (showing the model's confidence on its prediction) rather than just 0 or 1. In particular, they have chosen Naive Bayes classifiers as they can add new features later without re-training the model for the existing features. Your task is to help Dr. Strange's Lab to build the Naive Bayes classifier.

More precisely, your tasks are the following:

1. You will write a class to implement a Gaussian Naive Bayes classifier. The class should implement the following method:
    a. Train: the method will take the train data as input and train the classifier on the data.
    b. Predict: the method will take the test data as input and return the prediction probabilities on the data.
2. You will implement and try several feature transformation methods on the input features to improve the performance of the classifier.
3. Finally, you should generate results on the given data and compare those with the results obtained from the Gaussian Naive Bayes classifier of the scikit-learn package.

Note: *The program can be written in C++ / Java / Python programming language from scratch. No machine learning /data science /statistics package / library should be used for model creation.*

Relevant information:

Dataset Filename: `polydipsia.csv`
Number of Output Classes: `2`
Data Description:
Number of Instances: 768
Number of Attributes: 7 (all numeric)
Attribute Information:

1. Pregnancies,
2. BloodPressure
3. HormoneLevel
4. HumulinLevel
5. BMI

6. Age
7. LineageFactor

Tasks to be done:

1. **Five Fold Cross Validation:** The dataset is not divided into train and validation sets. The first task is to randomly partition the complete dataset into 5 parts: assign the first part as validation set and the rest for training the model. Repeat the process 5 times, assigning the validation sets in a round robin manner. *(five fold cross-validation)*. A sample python code for five fold cross-validation with the Gaussian Naive Bayes classifier of the scikit-learn package has been given.

2. **Implementation of the Classifier:**
   a. Implement the Gaussian Naive Bayes classifier as stated in the Objective Statement.
   b. Train and test your implementation on the data provided in the dataset.
   c. Compare the results with the results obtained from the Gaussian Naive Bayes classifier of scikit-learn package.

3. **Experiments on Feature Transformation:** The Gaussian Naive Bayes assume that each feature of the data follows a Gaussian distribution. When some feature does not closely follow a Gaussian distribution, the classifier's performance deteriorates. Therefore, sometimes, the features are transformed by some simple transformation function to make their distributions more close to Gaussian. Skewness is a measure to test whether a distribution is symmetric around its mean value. If the distribution is almost symmetric, its skewness will be close to zero. On the other hand, asymmetric distributions have large (in absolute value) skewness. Since, Gaussian distribution is a symmetric distribution, for a perfectly normally distributed feature, the skewness is 0 (note that the converse is not true). Thus, skewness can be used to measure whether a feature highly deviates from the Gaussian distribution. If some feature has very large (in absolute value) skewness, the feature is definitely not closely following Gaussian distribution. Thus, in this part you will perform the following:
   a. Calculate the skewness of all the features. The code to calculate the skewness of the features using python pandas package can be found in the sample code.
   b. If some feature has large skewness, say outside of the range [-2, 2], apply some simple transformation of the feature to reduce its skewness. Here are some examples of simple transformation:

      $f(x) = log(x)$, $f(x) = x^2$, $f(x) = \sqrt{x}$.
   c. Train and test your classifier on the dataset with transformed features.
   d. Compare the results with the results obtained from the Gaussian Naive Bayes classifier of scikit-learn package.

4. **Outcomes and Reporting:** Prepare and submit a report with the following –
   a. Results on the original dataset.
   b. Results on the dataset with the transformed features. Also provide the skewness of the original features and transformed features. Clearly mention the set of transformations you tried.
   c. You need to calculate precision, recall, f1-score and accuracy for all the experiments.

d. Report the average score for the 5 folds.

Submission Details: (to be submitted under the specified entry in CSE-Moodle)

1. ZIPPED Code Distribution in CSE-Moodle
2. A brief (2-3 page) report/manual of your work
   (with your hyperparameter tuning results also presented in that report)

Submission Guidelines:

1. You may use one of the following languages: C/C++/Java/Python.
2. Your Programs should run on a Linux Environment.
3. You are **not** allowed to use any library apart from these (Also explore all these libraries if doing in Python, or equivalent of these):

```
import numpy # linear algebra
import csv # data processing, CSV file I/O
import pandas # data processing, CSV file I/O
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.model_selection import KFold
from sklearn import naive_bayes # sklearn Naive Bayes
from sklearn.naive_bayes import GaussianNB # sklearn Gaussian Naive Bayes
import operator
from math import log
from collections import Counter
```

   Your program should be standalone and should **not** use any *special purpose* library for Machine Learning for the Naive Bayes classifier algorithm. Numpy and Pandas may be used. And, you can use libraries for other purposes, such as generation and formatting of data.
4. You should submit the program file and README file and **not** the output/input file.
5. You should name your file as <GroupNo_ProjectCode.extension>.
   (e.g., *Group99_PPNB.zip* for code-distribution and *Group99_PPNB.pdf* for report)
6. The submitted program file *should* have the following header comments:
   # Group Number
   # Roll Numbers : Names of members (listed line wise)
   # Project Number
   # Project Title
7. Submit through CSE-MOODLE only.
   Link to our Course page: https://moodlecse.iitkgp.ac.in/moodle/course/view.php?id=508

You should not use any code available on the Web. Submissions found to be plagiarized or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.

For any questions about the assignment, contact the following TA:
Suvadeep Hajra (Email: suvadeep.hajra@gmail.com)