[Project Code: FPNB]

Fault Prediction using Naive Bayes Classifier based Learning Model

Project Duration : 22-April-2024 ~~ 07-June-2024
Submission Information : (via) Google Classroom

Objective:

The transmission line is the most important component of the power system. The need for power and its allegiance has expanded dramatically in the modern period, and a transmission line's primary function is to carry electric power from the source area to the distribution network. The electrical power system is made up of numerous complex dynamic and interacting parts that are always susceptible to disruption or an electrical malfunction.

A power system normally functions under balanced conditions. A short-circuit or fault in the line occurs when the system becomes imbalanced owing to insulation failures at any point or due to the contact of live wires. Faults in the power system can develop for a variety of reasons, including natural disturbances (lightning, high-speed winds, earthquakes), insulation degradation, tree fall, bird shorting, and so on.

The dataset consists of six input features and one output label:

- Inputs:
  - Ia: Phase A line current
  - Ib: Phase B line current
  - Ic: Phase C line current
  - Va: Phase A line voltage
  - Vb: Phase B line voltage
  - Vc: Phase C line voltage
- Output:
  - S: Binary label (0 or 1) indicating the presence or absence of a fault:
    - 1: Fault is present
    - 0: No fault detected

1. Based on the dataset (described earlier), you will write a program to learn a **Naive Bayes Classifier.**
2. Compare the results with the results generated by the Naive Bayes classifier learning algorithm from a pre-created package such as scikit-learn.

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

DataSets:

Filename: fault.csv

Tasks to be done:

1. In the dataset given the features are continuous,hence use Gaussian Naive Bayes for solving this problem.
2. Randomly split the dataset as 80/20 split i.e., 80% for training and 20% for testing.
3. Naive Bayes Classifier Model:
   a. Implement naive bayes algorithm in your code and mention the same in the report. Do NOT use scikit-learn for this part.
   b. Test the implementation of the Classifier from scikit-learn package.
4. Classification Report:
   a. Create a classification report in tabular form.
   b. You need to calculate precision, recall, f1-score and accuracy of the model.

Submission Details: (to be submitted under the specified entry in Google classroom)

1. ZIPPED Code Distribution in Google Classroom
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.tree import DecisionTreeClassifier # sklearn Decision Tree
from sklearn.naive_bayes import MultinomialNB # sklearn Naive Bayes
from sklearn.naive_bayes import GaussianNB
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 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.
5. You should name your file as <GroupNo_ProjectCode.extension>.

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.
