Exploring Classification using Naive Bayes

1 Introduction

In this assignment, you will implement Naive Bayes on a toy dataset for classification task.Your key goal in this assignment is to correctly implement this model and analyse the results you obtain.

Starter code and dataset are available here: `https://github.com/ashutoshbsathe/cs725-hw`

NOTE: You are not allowed to use any python package other than python standard packages and the others mentioned in requirements.txt.

2 Data

The `data/` directory from our repository contains our toy dataset. It contains two files, train_dataset.csv and validation_dataset.csv. Both of them contain $(X_1, ..., X_{10})$ coordinates of points and class label in the last column. The code to load and extract the data has already been given to you and you need not make any modifications as such.

3 Starter Code

The starter code contains following files:

1. *model.py*: Contains boilerplate implementation of Naive Bayes classifier. You must go through comments of each function to understand their expected behavior. We will be using accuracy and f1-score metrics to test your models. You need to implement `fit`, `getParams`, `predict`, `f1score`, `recall` and `precision` functions.

4 Resources

Each datapoint consists of 10 dimension, which we label as $(X_1, X_2, .., X_{10})$ and class label from $(0, 1, 2)$.

1. (X_1, X_2) are drawn independently from two different univariate Gaussian distributions.
2. (X_3, X_4) are random variables drawn independently from two different Bernoulli Distributions
3. (X_5, X_6) are random variables drawn independently from two different Laplace Distributions
4. (X_7, X_8) are random variables drawn independently from two different Exponential Distribution
5. (X_9, X_10) are random variables drawn independently from two different Multinomial Distributions

Your goal is to calculate Maximum Likelihood estimators for each of these distributions and create a naive Bayes classifier with an appropriate prior to classify these points. Along with that you will be analysing your results using F1-score. You are free to code up MLE's for all these distributions yourself. You can find resources for MLE's for Gaussian,Bernoulli,Laplace,Exponential and Multinomial Distributions.

You will need to return the following parameters for each of these classes:

1. Gaussian : (μ, σ^2)
2. Bernoulli : p
3. Laplace : (μ, b)
4. Exponential : λ
5. Multinomial : $[p_1, p_2, .., p_k]$

All of these notations are consistent with the Wikipedia links provided. More details are provided in `getParams` function.

Note: F1-score is a metric often used while testing binary classifiers since accuracy alone is sometimes not enough. Read more about f1-score here. For our case of multi-class classifiers, we can find F1-score of each class by taking it as the positive class and all others as negative class. You can have a read here.

5 Report

We expect you to make a simple and short report explaining your approach to solving this problem and reporting all your estimated parameters and results. Be sure to report your F1 score here.

6 Submission Structure

Once you are done with the assignment, submit (1) your best model for the train dataset (2) your code for reproducing the best results i.e. your *model.py* (3) the report with the results you obtain. Since the assignment will be autograded, it is important to maintain the submission structure as mentioned below:

```
submission/
    model.pkl
    model.py
    report.pdf
```

Pack everything under `submission/` directory under a `.tar.gz` archive named as `rollno1_rollno2.tar.gz` and upload that to Moodle.