*CSE 569: Fundamentals of Statistical Learning and Pattern Recognition*

*Project 2 Report*

*Mallela Saidubabu*  
1225284411

**Introduction**

As part of the project2 , the idea is to understand the use the common SVM through some classification tasks on given training and testing data set. While understanding the data we can see that there are 50 categories in it.

The training data has 4786 Samples, and the testing data has 1883 samples. And Each sample is described by three feature rows namely X1,X2,X3 and given that all the three features are normalized histograms.

Have used python language for coding and have done using jupyter notebook from google colab. As part of the code, we have installed some of the packages that are required and have listed the dependencies section below. The SVM classifier that’s used here as given in the problem statement is libSVM.

Here as part of the problem solving, we have first created the label using svm\_problem which has been passed to the svm\_train with relevant parameters to create the training model by giving proper naming conventions and screenshots to understand clearly.

The SVM abbreviated as Support vector machines are machine learning algorithms which are used for classification or regression. Basically, SVM finds a hyper plane that created a boundary between two different types of data. In SVM we basically plot every data item of the dataset into a N dimensional space and then find the optimal boundary or hyperplane to divide the data.

In the given project, it is divided into 3 Steps(step0,step1,step2) which we will talk in detail in following pages

**##Step 0 – Classification by Individual Features**

Here Step 0 has two parts in it. The main idea is to train the svm classifier for every individual feature X1,X2,X3 which will result in three corresponding models for each of them.

In part1 we will be training the model normally and in the part2 we train the model to retrieving the probabilities

**Part-1**

As part of part1 we will training the individual data features using SVM as stated in below screenshot

Text

Description automatically generated

From the above we are training the model normally for each individual feature by passing the relevant parameters as stated in parameters input.

Here model1\_Step0\_Part1is trained with X1 data to obtain h1(x) model

Here model2\_Step0\_Part1is trained with X2 data to obtain h2(x) model

Here model3\_Step0\_Part1is trained with X3 data to obtain h3(x) model

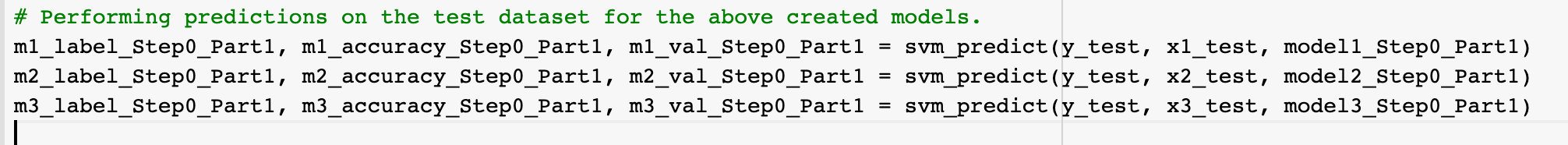
Here in the parameters:

-c 10: this represents cost parameter or penalty parameter for error term in SVM.

-t 0: This is used to set the kernel type. 0 here indicates the corresponding values for linear

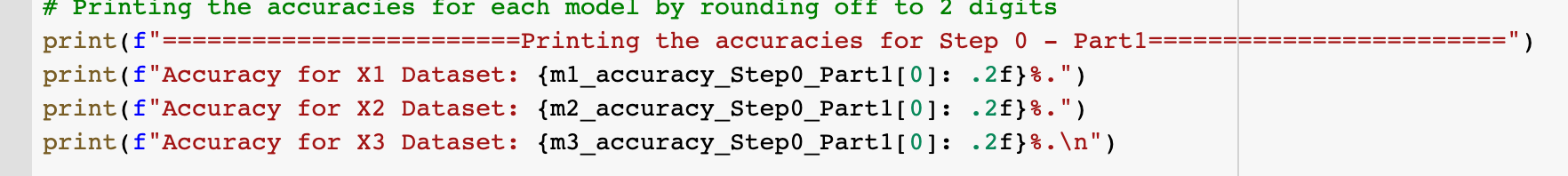
kernel.

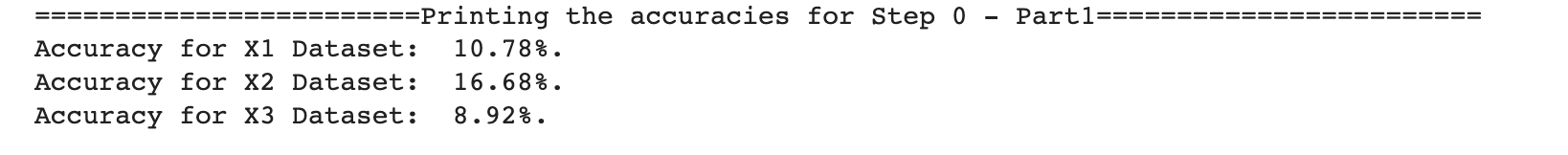
Once the models are trained, we predicted the test data set on every feature and generated the accuracies. This operation is done by svm\_predict function as shown below.



Here the prediction is done for the models generated above(h1(x),h2(x),h3(x)) for using test data of corresponding features.

The accuracies obtained for the same is show below:





**Part-2**

In the second part we are asked to retrieve the posterior for which we modify the svm parameters as stated Adding “-b 1” to ensures to calculate the probabilities

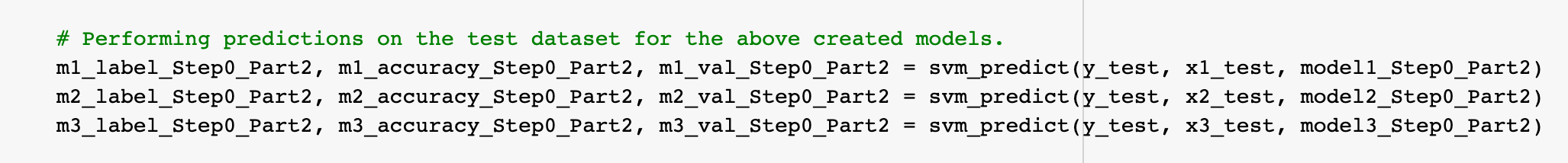
As part of part2 we will training the individual data features using SVM as stated in below screenshot

Text

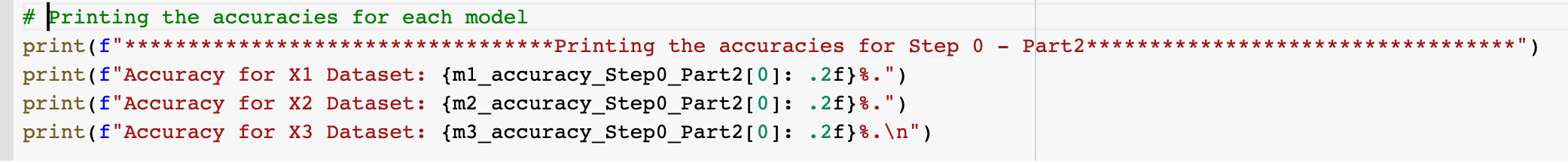
Description automatically generated

We can see that “-b 1”has been passed to generate the posterior.

Here the generated model is used to predict the test data for each individual feature and generate the accuracies as show below



The accuracies obtained for the same is show below



Text

Description automatically generated

Observations:

(1) The first observation is for larger datasets the train time of the model is very high. So

it takes significant time to train the model.

(2) When the model is trained with -b 1 parameter then the train time increases

significantly.

**##Step 1 – Feature combination by fusion of classifiers**

In this Step by using the probabilities obtained during the prediction of test data in

Step 0 part 2 and we will combine them and get new labels and find the accuracy.

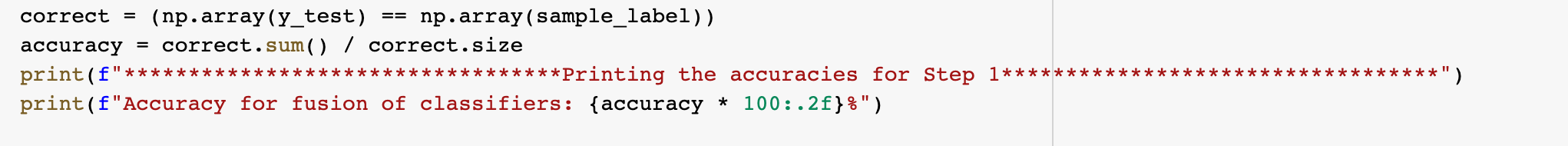
As stated in the problem statement The final probability fusion of 3 classifiers will be 𝑝(𝑤𝑖|𝐱) = Σ𝑘 𝑝𝑘(𝑤𝑖|𝐱)⁄3 and the final recognition result is 𝑤𝑖∗ = argmax𝑖 𝑝(𝑤𝑖|𝐱).

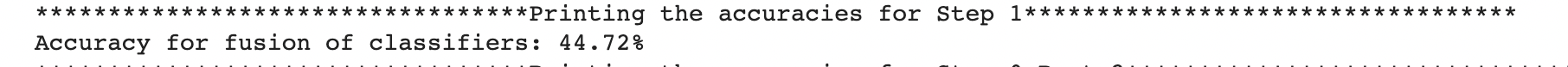
Text

Description automatically generated

The sample\_prob has the probabilities of every data point for all the 50 categories and the

final\_sample\_label has the predicted labels after combining the probabilities

**

**

**The accuracy obtained is around 44.72%** which is almost 1.6 times the accuracy obtained when

it is done individually.

Below data shows the comparison between the accuracy in the testing test of step1 to that of part2 in step 0

Text

Description automatically generated with low confidence

Graphical user interface, text

Description automatically generated

**Observations**:

From the data we see that the accuracy obtained after the fusion of features is 1.6 more

than the accuracy when calculated individually.

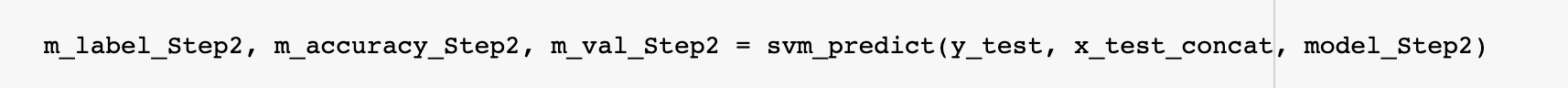
**##Step 2 – Feature combination by simple concatenation**

In this Step, we concatenate the data X1\_train, X2\_train, X3\_train to combine them into

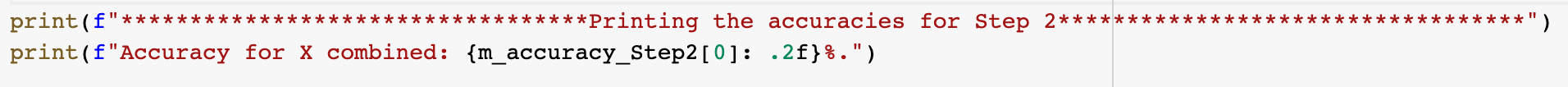
a single feature X and use the concatenated data to train and then use that model to classify

the test data.

Here the test data is also concatenated, and we can see the svm\_predict function for the same

****

The accuracy obtained for the same is shown below

****

***The accuracy in this step for concatenation is :* 37.0685% (698/1883) (classification)**

Below data shows the comparison between the accuracy in the testing test of step2 to that of part1 in step 0

**Calendar

Description automatically generated**

Graphical user interface, text, application

Description automatically generated

**Observation:**

The accuracy obtained after using all the features is between 2 to 3 times the accuracy obtained without combining. This tells us that using all features to train the model will have higher chance of getting the prediction right.

**Conclusion:**

Here we can see the accuracies with different types of classification by using the SVM classifier and successfully able to compare the results across different training models.

**Dependencies:**

To run the code the below dependencies are required. Given name and command to install

1. Libsvm – “ pip install -U libsvm-official “
2. Scipy – “pip install scipy “
3. Numpy - “ pip install numpy”