Project Part 2: Experimenting with SVM

Data:
The dataset contains 50 categories/classes. The training data contains 4786 samples in the file and the testing data contains the 1833 samples in the file. In this each sample is being described with 3 rows of different features i.e X1, X2,X3 in the corresponding file and the category vector is Y.

Step – 0
Part 1:
First of all import the training data and check the keys of the data in order to know about the labels of the data then import the test data and check it. Import the libsvm and then flatten the Y label of both the train and the test data test as it is in the form of 2d array. After that i uses svm_train function to train the label using the linear kernel and then used svm_predict to find its accuracy with the test data for each of the 3 features X1,X2,X2.

For X1 feature

```
p_label_x1, p_acc_x1,p_val_x1 = svm_predict(Y_test,X_test_X1,train_X1)
```

```
Accuracy = 11.3648% (214/1883) (classification)
```

For X2 feature

```
p_label_x2, p_acc_x2,p_val_x2 = svm_predict(Y_test,X_test_X2,train_X2)
```

```
Accuracy = 17.5252% (330/1883) (classification)
```

For X3 feature

```
p_label_x3, p_acc_x3,p_val_x3 = svm_predict(Y_test,X_test_X3,train_X3)
```

```
Accuracy = 8.60329% (162/1883) (classification)
```

Part -2
In this I obtained the posterior probability of 3 features by using parameter -b 1 in svm_train function and used svm_predict function to find its accuracy with the test data.

For X1 feature

```
train_X1_pp = svm_train(Y_train,X_train_X1,"-c 10 -t 0 -b 1")
```

```
p_label_x1_pp, p_acc_x1_pp,p_val_x1_pp = svm_predict(Y_test,X_test_X1,train_X1_pp,"-b 1")
```

```
Accuracy = 28.0404% (528/1883) (classification)
```

For X2 feature

```
train_X2_pp = svm_train(Y_train,X_train_X2,"-c 10 -t 0 -b 1")
```

```
p_label_x2_pp, p_acc_x2_pp,p_val_x2_pp = svm_predict(Y_test,X_test_X2,train_X2_pp,"-b 1")
```

```
Accuracy = 28.0935% (529/1883) (classification)
```

For X3 feature

```
train_X3_pp = svm_train(Y_train,X_train_X3,"-c 10 -t 0 -b 1")
```

```
p_label_x3_pp, p_acc_x3_pp,p_val_x3_pp = svm_predict(Y_test,X_test_X3,train_X3_pp,"-b 1")
```

```
Accuracy = 27.7217% (522/1883) (classification)
```

Step 1:
In this I have to find the sum of three features of posterior probability and then divide it by 3 and then find its argmax. So for it first I declared two empty lists then added all the three features posterior probability which we did in step 0 part 2 after that divided by 3 which is equivalent to mean then find its argmax. After that found its accuracy in terms of test data.

```
pp=[]
W_i=[]
for i in range(len(p_val_x3_pp)):
    temp=[]
    for j in range(len(p_val_x3_pp[0])):
        sum_pp=p_val_x3_pp[i][j]+p_val_x2_pp[i][j]+p_val_x1_pp[i][j]
        mean_pp=(sum_pp/3)
        temp.append(mean_pp)
    pp.append(temp)
    W_i.append(np.argmax(temp)+1)
```

```
# W_i
```

```
count=0
for i in range(len(Y_test)):
    if Y_test[i]==W_i[i]:
        count+=1
```

```
print("Accuracy in Testing Set =",count/len(Y_test)*100)
```

```
Accuracy in Testing Set = 44.66277217206585
```

Step 2:
In this I concatenated all the 3 features training data in order to form a single feature and the same with the test data. Then trained the SVM classifier by using svm_train function and passed -c 10 -t 0 -b 1and in order to check its accuracy tested with the test data using svm_predict by passing -b 1 parameter.

```
conct_X_features_train= np.zeros(shape=(len(X_train_X2), 3000))
for i in range(len(X_train_X2)):
    conct_X_features_train[i]=np.concatenate((X_train_X1[i],X_train_X2[i],X_train_X3[i]),axis=None)
```

```
conct_X_features_test= np.zeros(shape=(len(X_test_X2), 3000))
for i in range(len(X_test_X2)):
    conct_X_features_test[i]=np.concatenate((X_test_X1[i],X_test_X2[i],X_test_X3[i]),axis=None)
```

```
conct_X_features_train1=svm_train(Y_train, conct_X_features_train, '-c 10 -t 0 -b 1')
```

```
p_label_conct, p_acc_conct, p_val_conct = svm_predict(Y_test, conct_X_features_test, conct_X_features_train1, '-b 1'
```

```
Accuracy = 48.4334% (912/1883) (classification)
```