Aman Bhansali
B20ME010

KAGGLE COMPETITION

Data

We were given four sets of data that were Train features, Train labels and Test features and Sample Submission. The train data set contains 10,000 data points with 200 features, while the Test data set consists of 400 data points. This was a multi-class problem which contained a total of 100 classes. The datasets provided were in txt format this time while we needed to predict our output in csv format. The output we needed to predict consisted of 5 classes with the highest probabilities in descending form and for this were provided with the Sample Submission to understand the format of the predicted output.

Steps

Since the dataset was given in a .txt format we needed to do certain conversions to use the data sets directly during classifiers application. First, used open function and then read lines and the file contained all the data inside strings. Then, made an empty list and appended the same inside that and used split and got a 2D list which now contained the individual values inside string. The problem now was that the last digit of all the inside lists contained "\n" so, we needed to remove this also. For this I used the indexing and removed "\n" and then converted all the values inside string to float. Then same did for test and train labels. Then converted the lists to Data Frame and now we had all the train and test data set. Now, we are set to use different classifiers.

Now, say we finalized the model after this I used the function classifier.predict_proba(Test_data) which gives the probability of all the classes. After, this I again used a sorting code and got the 5 classes corresponding to probabilities Observing this I found that the prediction made contained labels that were being taken from zero. So, increased the value of each class by 1. Thus, we get the final data frame output.

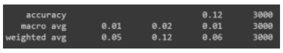
Models Applied and Validation

Since, this was a multi-class problem and we know many classification problems and I followed the following order for evaluation:

1. Logistic Regression
2. MLP Classifier
3. SVM

For Logistic regression I divided the train data provided into 70:30 ratio with random-state = 42 and used this 30% for validation. For, this question the evaluation was based on map@5 metric which is based on precision values. Say, we needed to predict the 5 best labels, the evaluation was based on precision, like say if we have actual label 4 and the one we predicted say is 4 3 5 6 7, then the model evaluates it to 1 and say 5 6 4 7 8, then the model evaluates to 0.25.

The observed accuracy for random_state = 42 I got



```
    accuracy                          0.12      3000
   macro avg       0.01      0.02      0.01      3000
weighted avg       0.05      0.12      0.06      3000
```

Aman Bhansali
B20ME010

KAGGLE COMPETITION

For MLP Classifier:

Since, neural networks are said to work pretty well in most of the cases so I tried with different values of parameters like random-states, no. of nodes and no. of hidden layers. I started with: no. of nodes = 100, no. of layers = 1, max. iterations = 400 and activation function = logistic. The observed accuracy was: `0.14066666666666666` on the 30% data. Then I trained the classifier with the same parameters on the original complete train data and then used the same steps as explained above to get the final output and the final accuracy on prediction on Kaggle was 0.25963.

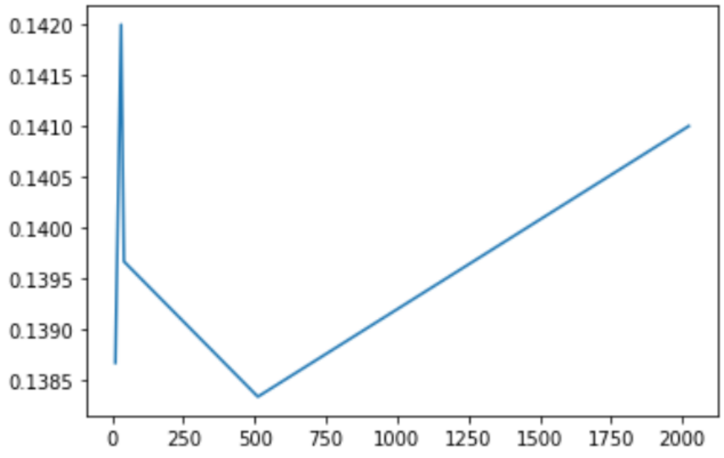
Then I used the random-state value = 21, with the rest parameters same as above and got an accuracy of `0.137` on the 30% data and on Kaggle it gave 0.25807.

Then I increased the number of layers to two.

Now the parameters were no. of nodes = (200, 100) no. of layers = 2, max. iterations = 400 and activation function = logistic. I evaluated the model as per accuracy on the 30% data for different values of random_state and the best I got were for 42 and 2023 that were `0.13766666666666666` and `0.13633333333333333`. But on kaggle the one with more accuracy gave approximately 0.25 accuracy.

Also, then I increased the number of nodes of first layer to 150 and second being the same and the rest parameters being the same. Then I again used loop to iterate over the best one and found that the best I got was for 31, 41, 2023 that were `0.142`, `0.13966666666666666`, `0.141`. Then among these I submitted the one with 2023 and 31 on Kaggle and got an accuracy of 0.26302 and 0.27 since the one with 31 was giving accuracies same as one of my previous models on kaggle.

This is the curve for accuracy vs random-state for the above case.



Also, I tried with the same above parameters just changing the activation to relu but the results were low. The max was `0.11266666666666666` for a random state value = 21.

Also, I tried increasing the number of layers to 3 but the results were not much convincing. The nodes I took were 100, 100, 50, but couldn’t get accuracy above 12.77 percent. As, increasing the number of layers now more also decreased the accuracy.

For now, I got my highest for two layers 150, 100 for 31 as random state and got 0.27 accuracy.

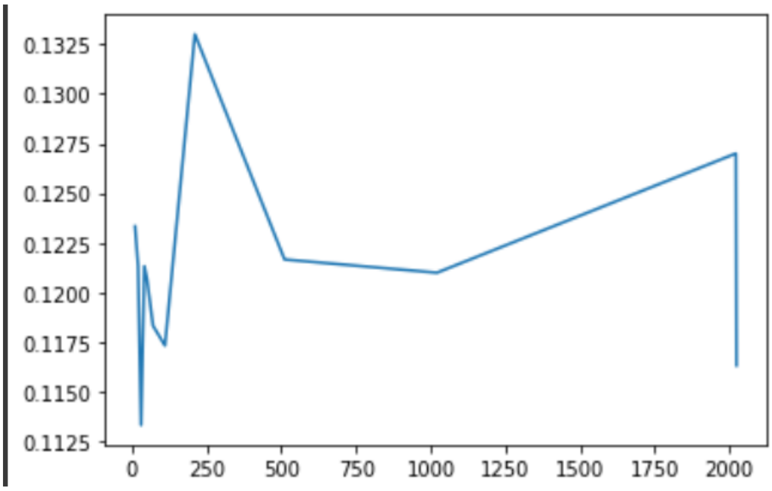
Aman Bhansali
B20ME010

KAGGLE COMPETITION

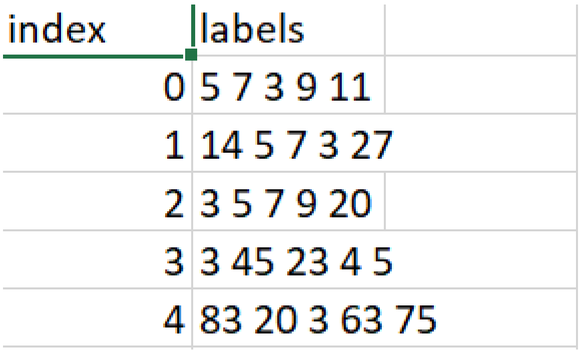
Then I also applied SVM on the dataset.

As, I observed that increasing the value of C the accuracy was slowly decreasing so I fixed my C=1 and kernel=rbf which is used for multi-class and which gave the best accuracy and varied the random-states.

This is the curve for random-state and accuracies. From this I got that at 211 the accuracy was the most and so I trained my model on this and the accuracy observed on kaggle was



Thus, doing all this also my accuracy remained the same and hence I used the same above as my final model whose accuracy was also relatively better than many others I trained. Its first five values were:



index	labels
0	5 7 3 9 11
1	14 5 7 3 27
2	3 5 7 9 20
3	3 45 23 4 5
4	83 20 3 63 75

References:

1. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
2. http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
3. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
4. https://stackoverflow.com/questions/16858652/how-to-find-the-corresponding-class-in-clf-predict-proba

Aman Bhansali
B20ME010

KAGGLE COMPETITION

5. https://www.geeksforgeeks.org/python-program-to-find-n-largest-elements-from-a-list/
6. https://stackoverflow.com/questions/6910641/how-do-i-get-indices-of-n-maximum-values-in-a-numpy-array
7. https://stackoverflow.com/questions/3277503/how-to-read-a-file-line-by-line-into-a-list