NAME: Venkata Sai Siva Pruthvi. Ustepalle S-ID: 800957126

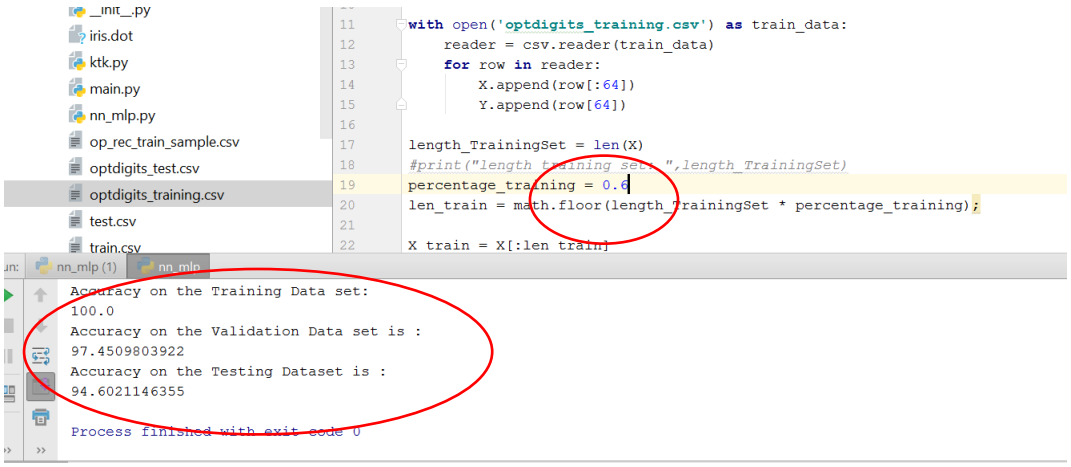
(Worked together with Chaitanya Sri Krishna Lolla)

Digit Recognition dataset

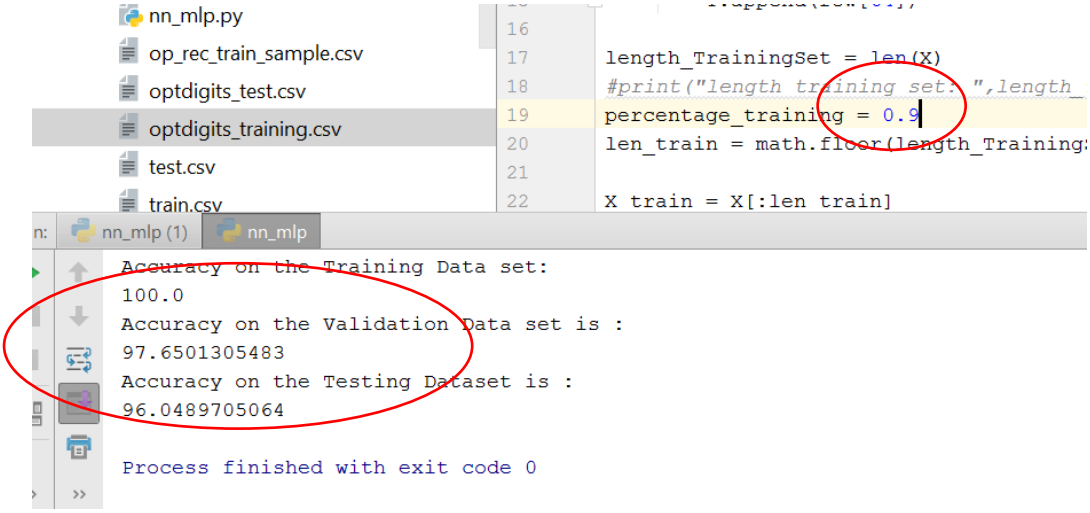
- RESULTS:

Splitting the given data into 2 parts to perform cross validation (as a step for pruning the data).

Training on anything below 60% for cross validation which is resulting in a 100% accuracy on training set and 97% accuracy on the validation data.



But on cross validating it with a 90-10 split of training data it is resulting in a better fit as it can be observed in the accuracy on testing data:



Above accuracy is a resultant of the classifier with all the default values provided by the sklearn classifier,

hidden layers 100

Activation function – rectified linear unit function

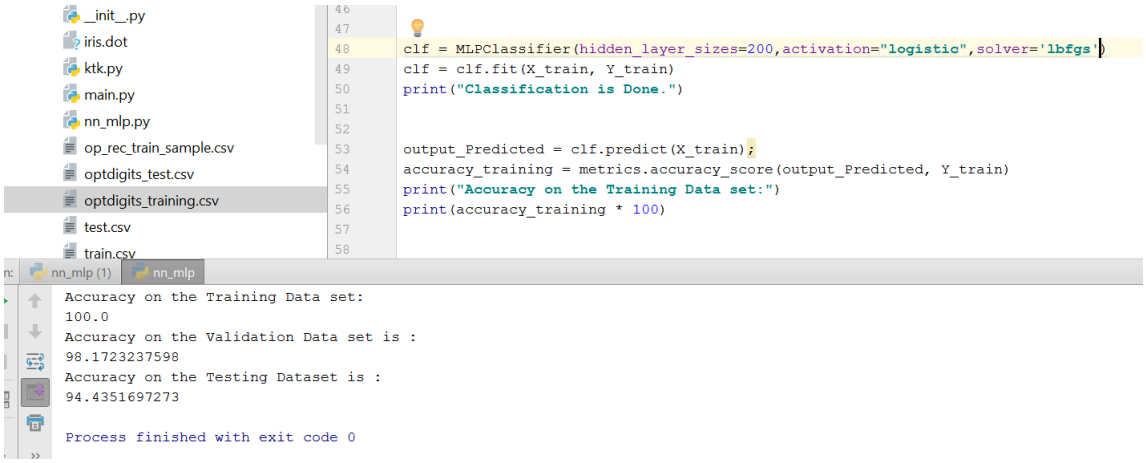
Solver – adam (stochastic gradient-based optimizer proposed by Kingma, Diederik, and Jimmy Ba)

Then applying various attributes for the MLP classifier provided by sklearn package.

hidden layers 200(any more # is resulting in the same result)

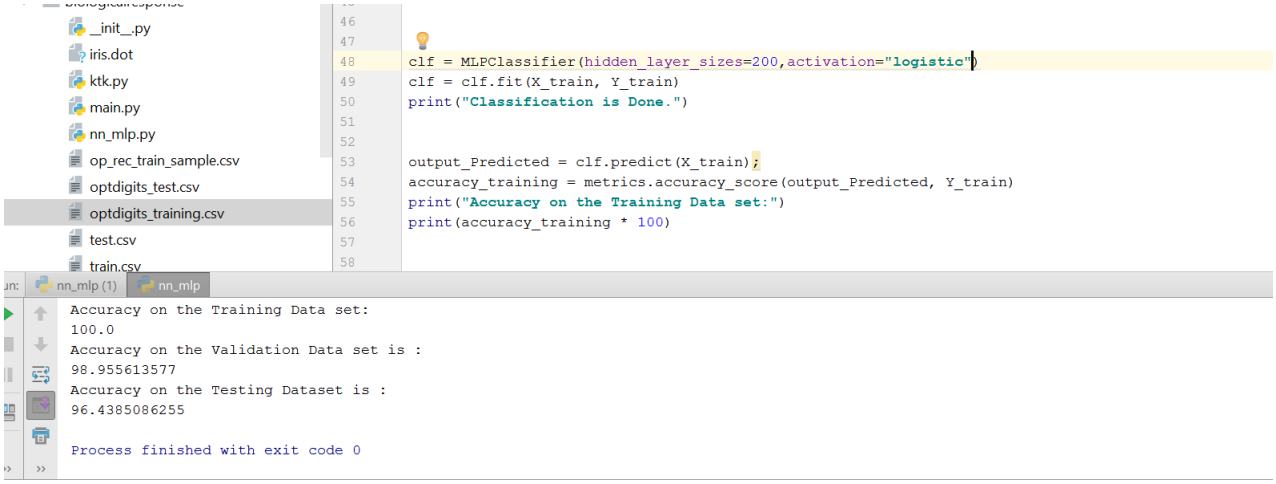
Activation function – logistic sigmoid function.

Solver – lbfgs(which is proposed to work better on smaller datasets)

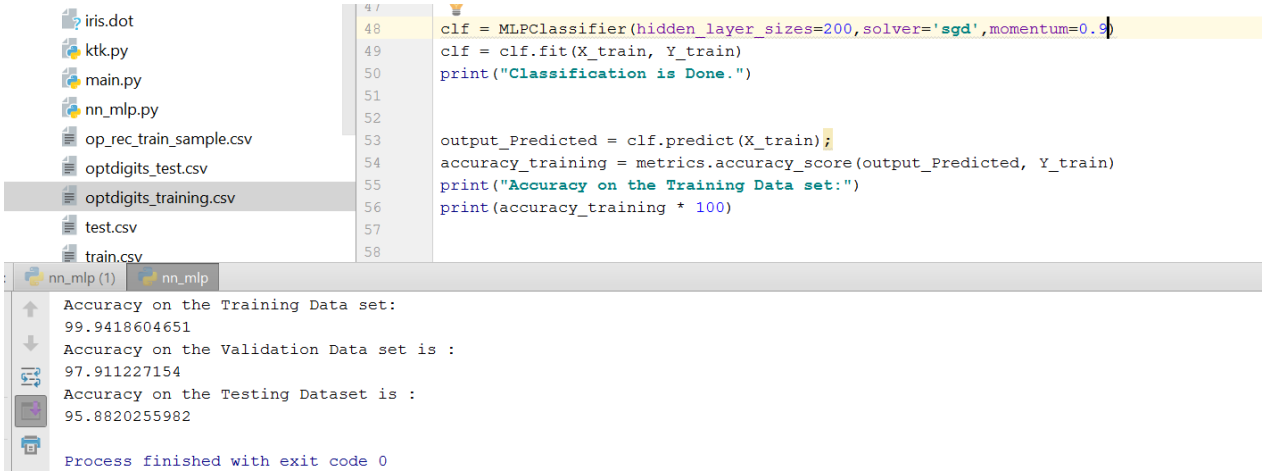


Though this could increase the accuracy on the validation data set it did not perform well on test dataset.

Depicting the problem of using solver='lfbgs' on removing it the classifier can perform better.



On using the gradient descent as solver and momentum value 0.9 the accuracy is little bit decreased because the model is finding an early local minimum and halting the learning criteria.



Amazon review dataset

- Result:

Using nltk library services to lemmatize, tokenize the reviews along with removing the punctuation from the review texts as a part of preprocessing steps as in done in the previous assignment.

Pre-processing the text and making it into two sets of positive and negative reviews.

Picking top 5000 of the most commonly occurring words and assigning them numerical values to pass as parameters for the decision tree classifier method.

Used TF-IDF to build a matrix of numerical representation of the words in each sentence.

And the same procedure is repeated on the test dataset and the accuracy is measured.

- Used a normal MLPClassifer() method and checked for the accuracy of model (which has default values as mentioned above) but tweaked the number of hidden layers to be lesser than default model gave # to be 50 which is 50% less than the default value. Observed that it gave a very less accuracy as a snapshot below:

```
C:\Users\pruth\Anaconda3\python.exe D:/pruthvi/python_ws/amazon_review_sentiment_analysis/nn_mlp.py
Mean of review attribute is :  4.120430078052725
reviews grouped by their count: rating
0     34398
1    111529
Name: review, dtype: int64
Done
55558
[('not', 80912), ('baby', 70749), ('one', 66194), ('love', 52997), ('great', 47666), ('like', 45664), ('woul
Train data:  99.9910914361
Done
27828
[('not', 20502), ('baby', 17687), ('one', 16201), ('love', 13132), ('great', 11756), ('would', 11417), ('lik
Accuracy of test data :  57.3854129522
```

Used a combination of different sets of hidden layers and activation techniques and observed that it has not taken a significant rise in defining the accuracy and finally settled with the following classifier;

- Now I 've changed the values for hidden layers = 200, activation = logistic as I've observed that this is giving a better fit.

```
C:\Users\pruth\Anaconda3\python.exe D:/pruthvi/python_ws/amazon_review_sentiment_analysis/nn_mlp.py
The Mean of the Review Attribute is :
4.120430078052725
Done
55558
[('not', 80912), ('baby', 70749), ('one', 66194), ('love', 52997), ('great', 47666), ('like', 45664), ('would', 45661), ('use', 42480), ('seat', 39416), ('get', 3830
99.9910914361
Done
27828
[('not', 20502), ('baby', 17687), ('one', 16201), ('love', 13132), ('great', 11756), ('would', 11417), ('like', 11267), ('seat', 10442), ('use', 10437), ('get', 9549
67.7592780536
```