**Assessment Task 2: Individual Problem Solving task**

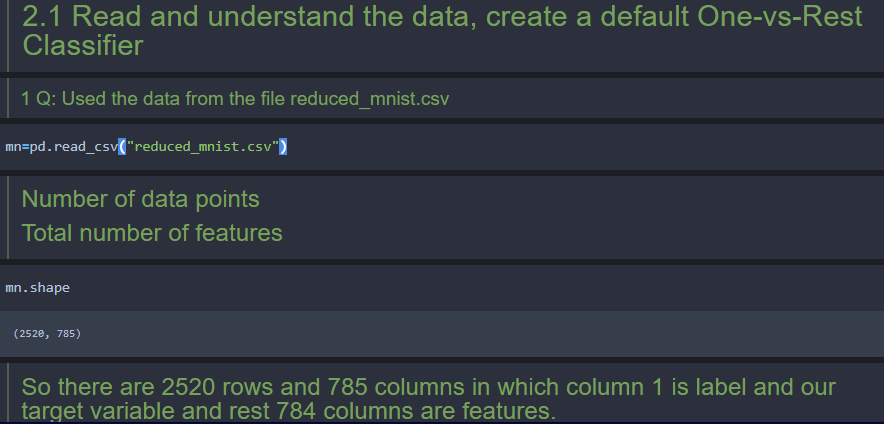
**Part-2 (Multiclass Classification):**

**2.1 Read and understand the data, create a default One-vs-Rest Classifier**

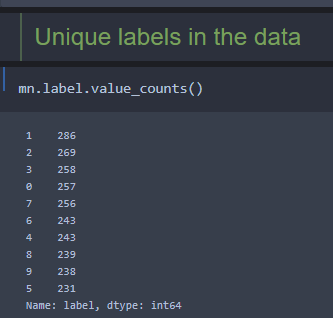
**1- Use the data from the file reduced\_mnist.csv in the data directory. Begin by reading the data. Print the following information:**

* Number of data points

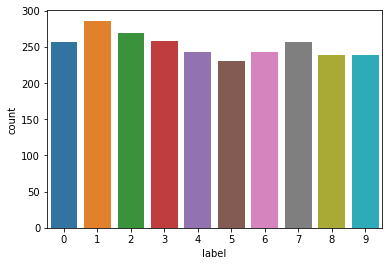
• Total number of features



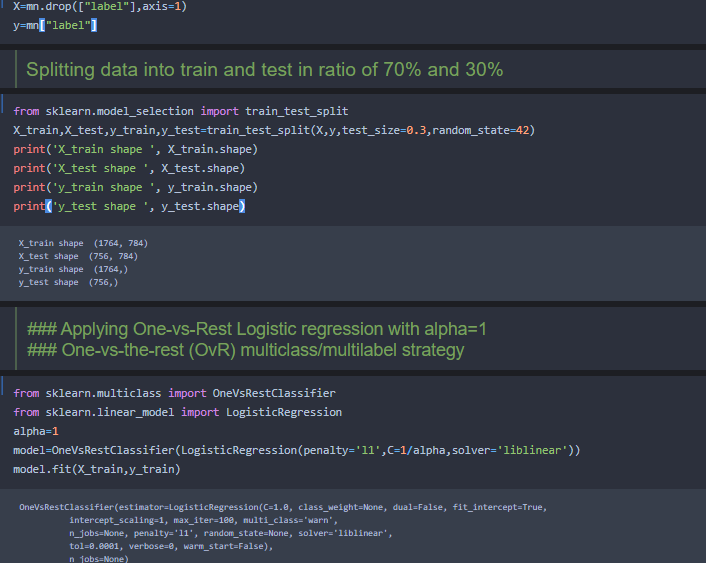
**• Unique labels in the data:**

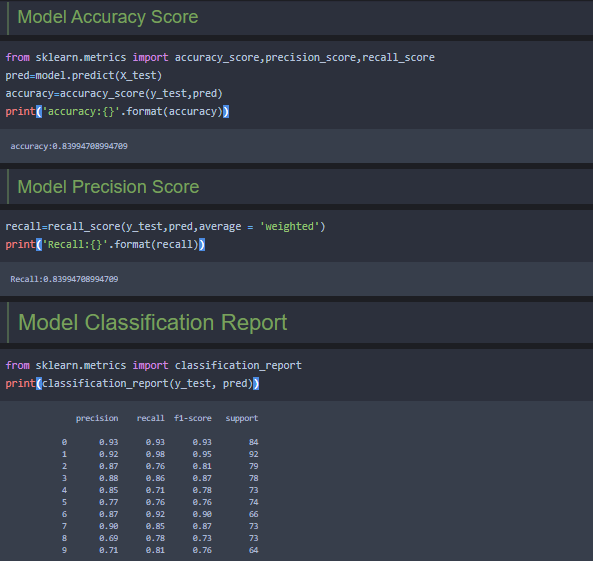


**Countplot for Labels:**

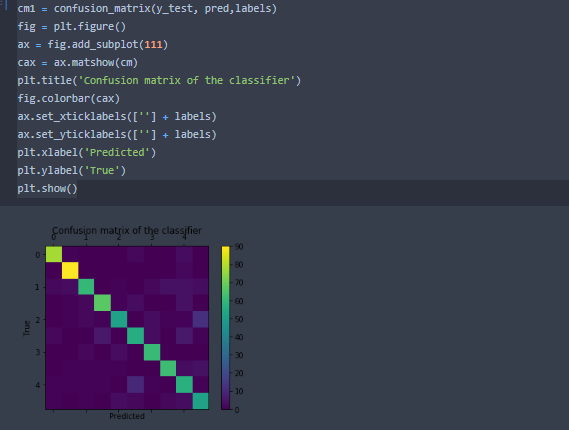
****

**2- Split the data into 70% training data and 30% test data. Fit a One-vs-Rest Classifier (which uses Logistic regression classifier with alpha=1) on training data, and report accuracy, precision, recall on testing data.**





**Confusion Matrix:**



**2.2 Choosing the best hyper-parameter**

**1- Choose the best value of alpha from the set a={0.1, 1, 3, 10, 33, 100, 333, 1000, 3333, 10000, 33333} by observing average training and validation performance P. On a graph, plot both the average training performance (in red) and average validation performacne (in blue) w.r.t. each hyperparameter value. Comment on this graph by identifying regions of overfitting and underfitting. Print the best value of alpha hyperparameter.**

**Output:**

C: 10.0

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 1.0

Test accuracy: 0.8306878306878307

C: 1.0

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 1.0

Test accuracy: 0.8452380952380952

C: 0.3333333333333333

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 1.0

Test accuracy: 0.8452380952380952

C: 0.1

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 1.0

Test accuracy: 0.843915343915344

C: 0.030303030303030304

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 1.0

Test accuracy: 0.8544973544973545

C: 0.01

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 0.9971655328798186

Test accuracy: 0.8677248677248677

C: 0.003003003003003003

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 0.9603174603174603

Test accuracy: 0.8743386243386243

C: 0.001

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 0.9246031746031746

Test accuracy: 0.8597883597883598

C: 0.00030003000300030005

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 0.86281179138322

Test accuracy: 0.8346560846560847

C: 0.0001

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 0.764172335600907

Test accuracy: 0.7473544973544973

C: 3.000030000300003e-05

Coefficient of each feature: [[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

...

[0. 0. 0. ... 0. 0. 0.]

[0. 0. 0. ... 0. 0. 0.]

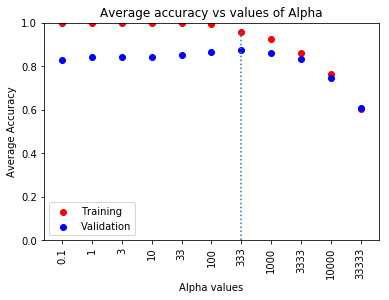
[0. 0. 0. ... 0. 0. 0.]]

Training accuracy: 0.6037414965986394

Test accuracy: 0.6084656084656085

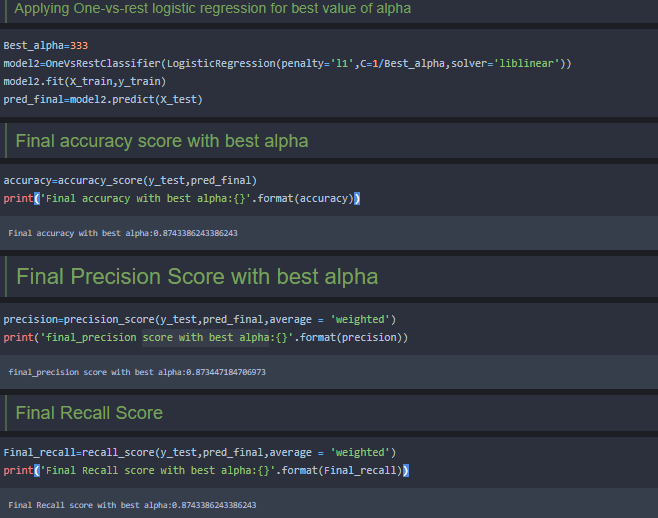
#### So we can see that our model accuracy is decreasing for training data when alpha is increasing. So here is some sign of under fitting in the model

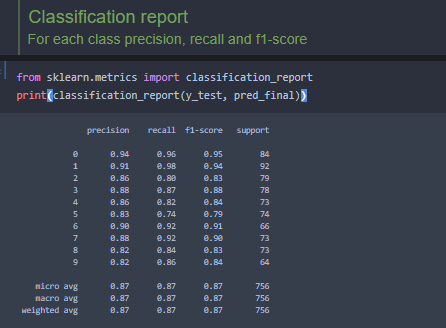
**Plot between training accuracy and test accuracy for choosing best alpha:**

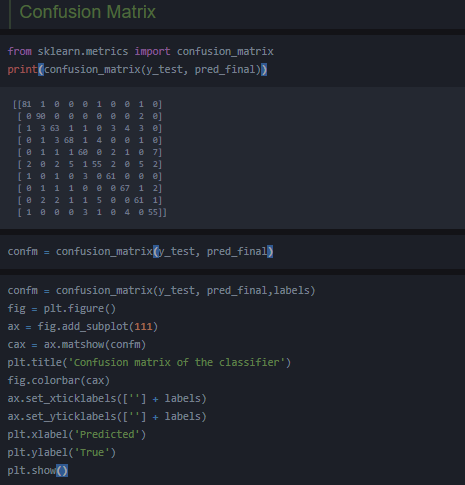


**Best Alpha: 333**

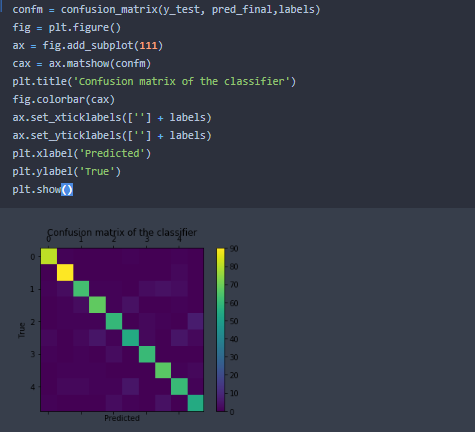
**2- Use the best alpha and all training data to build the final model and then evaluate the prediction performance on test data and report the following:**







**Confusion matrix plot:**



**Q3:Discuss if there is any sign of underfitting or overfitting with appropriate reasoning**

**Answer:** *Over fitting means that our trained model is not generalizing enough. Our model can give us excellent predictions for our training data, but can give us poor predictions for any data never seen before during training.* Under fitting is the opposite. Our model is generalizing too much to the point that it gives poor predictions even if you used our training data. The trick is to balance between them. We need to generate a model that gives good results when we use our training data and also gives good predictions for any data never seen before during training. As our model accuracy is good for test data so there is no sign of under fitting and overfitting.

**Reference:**

1. Data Science Journal. (2012, April). Available Volumes. Retrieved from Japan Science and Technology Information Aggregator, Electronic: http://www.jstage.jst.go.jp/browse/dsj/\_vols [Archived](https://web.archive.org/web/20120403153707/http:/www.jstage.jst.go.jp/browse/dsj/_vols) 3 April 2012 at the Wayback Machine
2. R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2–3, pp. 271–274, 1998
3. Jesse; Martino, Luca; Olmos, Pablo M.; Luengo, David (2015-06-01). "Scalable multi-output label prediction: From classifier chains to classifier trellises". *Pattern Recognition*. **48** (6): 2096–2109
4. Riemenschneider, M; Senge, R; Neumann, U; Hüllermeier, E; Heider, D (2016). ["Exploiting HIV-1 protease and reverse transcriptase cross-resistance information for improved drug resistance prediction by means of multi-label classification"](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4772363)