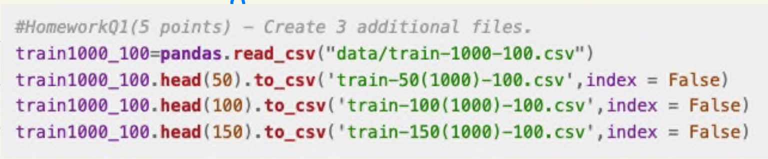
Due: 02/28/2022
Devika Chandnani
A13405666

Homework 2

Question 1

: Asks to create 3 additional files from train-1000-100.csv

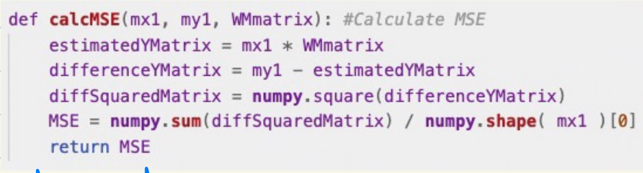
→ Files created using:



```
#HomeworkQ1(5 points) - Create 3 additional files.
train1000_100=pandas.read_csv("data/train-1000-100.csv")
train1000_100.head(50).to_csv('train-50(1000)-100.csv',index = False)
train1000_100.head(100).to_csv('train-100(1000)-100.csv',index = False)
train1000_100.head(150).to_csv('train-150(1000)-100.csv',index = False)
```

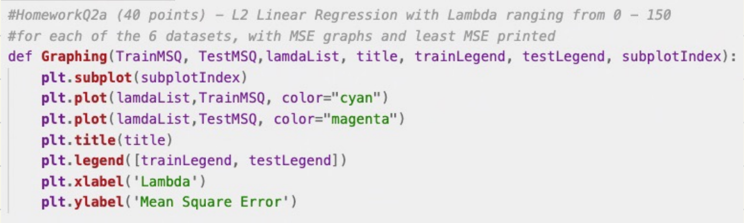
Question 2

· Asks to plot MSE, it is calculated using:



```
def calcMSE(mx1, my1, WMmatrix): #Calculate MSE
    estimatedYMatrix = mx1 * WMmatrix
    differenceYMatrix = my1 - estimatedYMatrix
    diffSquaredMatrix = numpy.square(differenceYMatrix)
    MSE = numpy.sum(diffSquaredMatrix) / numpy.shape( mx1 )[0]
    return MSE
```

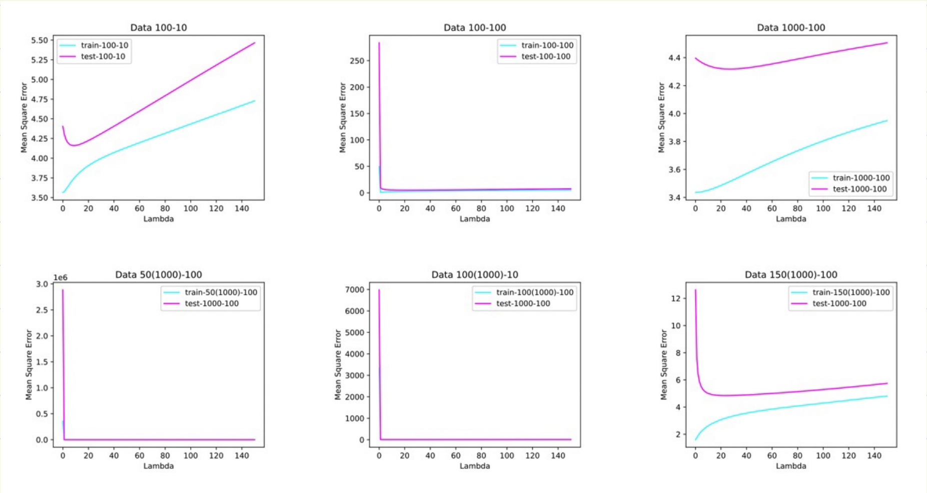
: and graphed using:



```
#HomeworkQ2a (40 points) - L2 Linear Regression with Lambda ranging from 0 - 150
#for each of the 6 datasets, with MSE graphs and least MSE printed
def Graphing(TrainMSQ, TestMSQ,lamdaList, title, trainLegend, testLegend, subplotIndex):
    plt.subplot(subplotIndex)
    plt.plot(lamdaList,TrainMSQ, color="cyan")
    plt.plot(lamdaList,TestMSQ, color="magenta")
    plt.title(title)
    plt.legend([trainLegend, testLegend])
    plt.xlabel('Lambda')
    plt.ylabel('Mean Square Error')
```

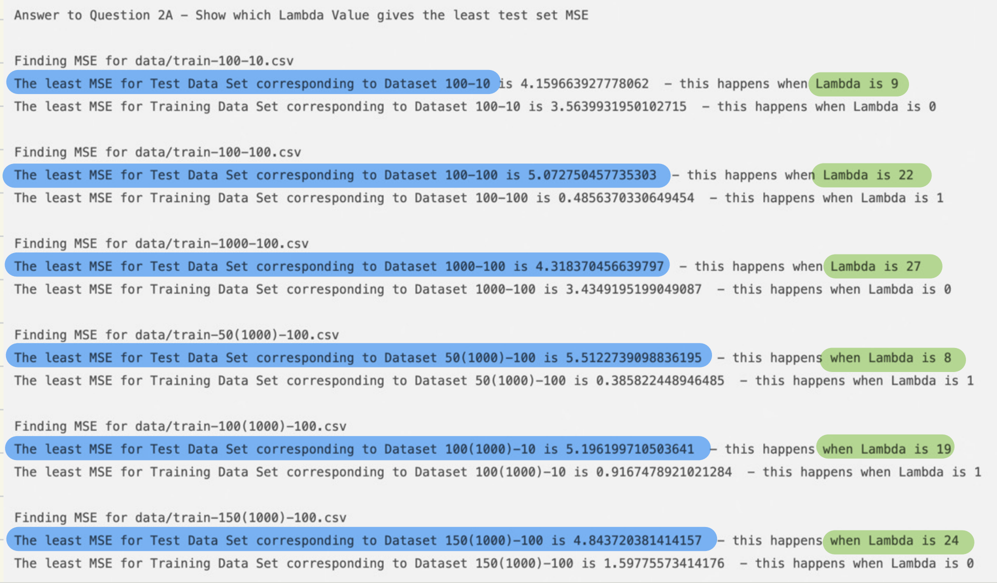
Below are the plots displaying the Mean Square error for both the training set MSE and the test set MSE.

They are plotted against Lambda ranging from 0 - 150.



Question 2A

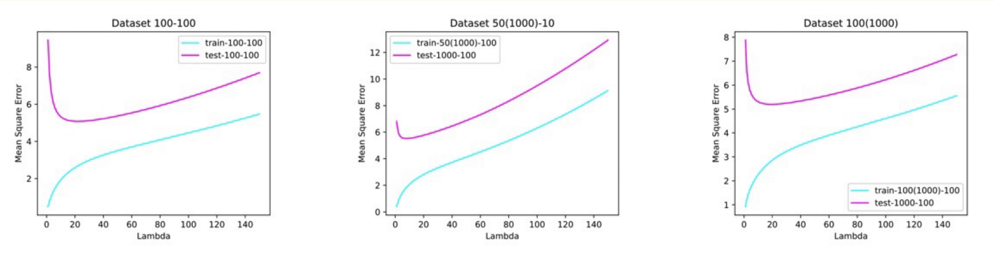
: For each of the datasets, the least MSE (test) is given at the following lambdas:



```
Answer to Question 2A - Show which Lambda Value gives the least test set MSE

Finding MSE for data/train-100-10.csv
The least MSE for Test Data Set corresponding to Dataset 100-10 is 4.159663927778062  - this happens when Lambda is 9
The least MSE for Training Data Set corresponding to Dataset 100-10 is 3.5639931950102715  - this happens when Lambda is 0

Finding MSE for data/train-100-100.csv
The least MSE for Test Data Set corresponding to Dataset 100-100 is 5.072750457735303 - this happens when Lambda is 22
The least MSE for Training Data Set corresponding to Dataset 100-100 is 0.4856370330649454  - this happens when Lambda is 1

Finding MSE for data/train-1000-100.csv
The least MSE for Test Data Set corresponding to Dataset 1000-100 is 4.318370456639797  - this happens when Lambda is 27
The least MSE for Training Data Set corresponding to Dataset 1000-100 is 3.4349195199049087  - this happens when Lambda is 0

Finding MSE for data/train-50(1000)-100.csv
The least MSE for Test Data Set corresponding to Dataset 50(1000)-100 is 5.5122739098836195  - this happens when Lambda is 8
The least MSE for Training Data Set corresponding to Dataset 50(1000)-100 is 0.385822448946485  - this happens when Lambda is 1

Finding MSE for data/train-100(1000)-100.csv
The least MSE for Test Data Set corresponding to Dataset 100(1000)-10 is 5.196199710503641 - this happens when Lambda is 19
The least MSE for Training Data Set corresponding to Dataset 100(1000)-10 is 0.9167478921021284  - this happens when Lambda is 1

Finding MSE for data/train-150(1000)-100.csv
The least MSE for Test Data Set corresponding to Dataset 150(1000)-100 is 4.843720381414157 - this happens when Lambda is 24
The least MSE for Training Data Set corresponding to Dataset 150(1000)-100 is 1.59775573414176  - this happens when Lambda is 0
```

Question 2B

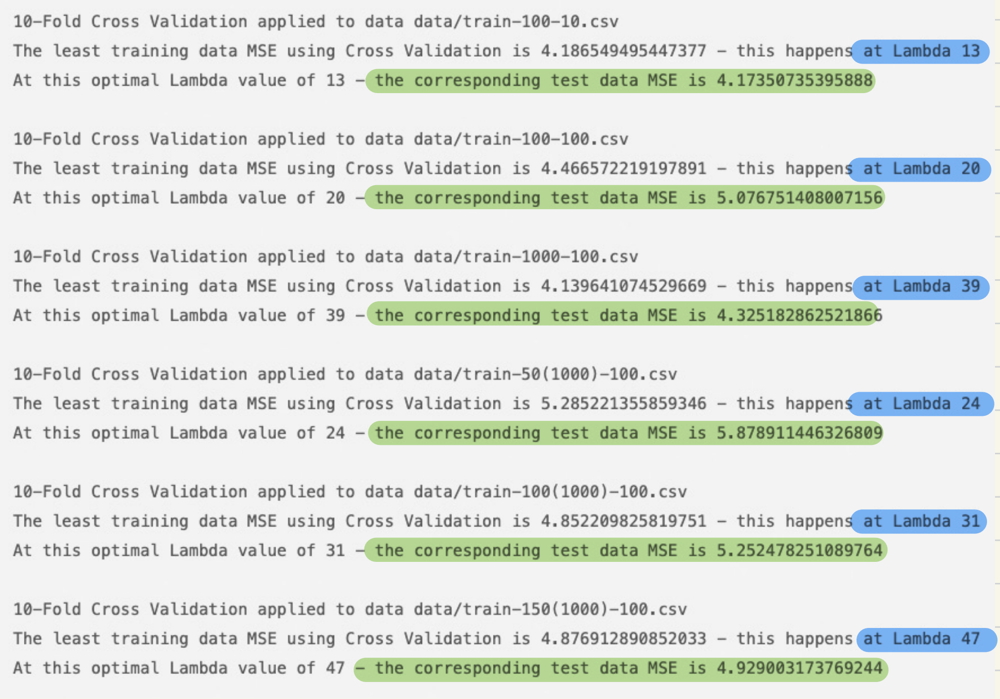
: Asks to provide an additional graph for datasets train-100-100.csv, train-50(1000)-100.csv and train-100(1000)-100.csv. With lamba starting at 1 instead of 0.



Question 2C

In L2 Ridge Regression, Lambda is a hyper parameter which when high adds high penalty error term making the learned hyper plain almost linear. when Lambda is close to zero it has almost no effect on the error term causing no regularization. The 3 datasets have a high MSE because of overfitting at Lambda 0. The model has too many parameters and these datasets have fewer instances or observations compared to the others. When the model is applied to the test data it results in a high MSE since it has been over-fitted.

Question 3A



```
10-Fold Cross Validation applied to data data/train-100-10.csv
The least training data MSE using Cross Validation is 4.186549495447377 - this happens at Lambda 13
At this optimal Lambda value of 13 - the corresponding test data MSE is 4.17350735395888

10-Fold Cross Validation applied to data data/train-100-100.csv
The least training data MSE using Cross Validation is 4.466572219197891 - this happens at Lambda 20
At this optimal Lambda value of 20 - the corresponding test data MSE is 5.076751408007156

10-Fold Cross Validation applied to data data/train-1000-100.csv
The least training data MSE using Cross Validation is 4.139641074529669 - this happens at Lambda 39
At this optimal Lambda value of 39 - the corresponding test data MSE is 4.325182862521866

10-Fold Cross Validation applied to data data/train-50(1000)-100.csv
The least training data MSE using Cross Validation is 5.285221355859346 - this happens at Lambda 24
At this optimal Lambda value of 24 - the corresponding test data MSE is 5.878911446326809

10-Fold Cross Validation applied to data data/train-100(1000)-100.csv
The least training data MSE using Cross Validation is 4.852209825819751 - this happens at Lambda 31
At this optimal Lambda value of 31 - the corresponding test data MSE is 5.252478251089764

10-Fold Cross Validation applied to data data/train-150(1000)-100.csv
The least training data MSE using Cross Validation is 4.876912890852033 - this happens at Lambda 47
At this optimal Lambda value of 47 - the corresponding test data MSE is 4.929003173769244
```

Question 3B

: In part 2A, the returned MSE values are the least ones found amongst Lambda ranging from 0-150. It returns the least ones found at a specific Lambda valle for each test data set.

In part 3A, the returned MSE values for the test set are dependent on the optimal Lambda derived from the least MSE in the training data.

The key difference is that in the latter case we only use the training data to determine the results. Unfortunately, it is rarely the case that the traing data and test data are available in real world applications. The data has to be spilt into folds. In question 3A we ran a 10 fold CV.

Question 3C

The drawbacks of cross validation:

→ Increased Training Time : Earlier we had to train our model only once but in cross validation you train your model on multiple sets

→ Needs more computation : CV is computationally expensive & requires a lot of processing. The traing algorithm runs from scratch k times

: Additional challenges with CV (unrelated to this case) is that it doesn't work very well with sequential data.

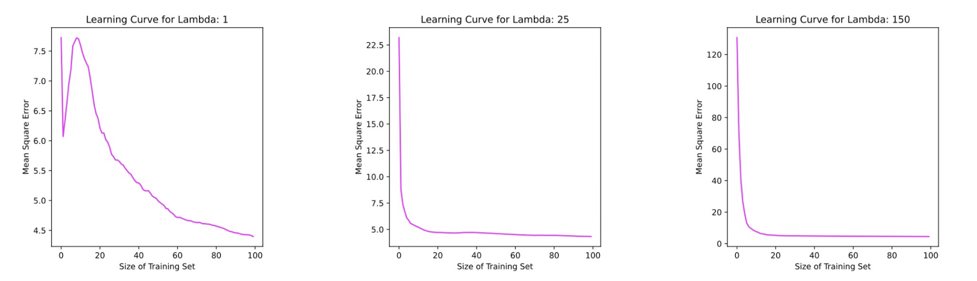
Question 3D

: The main factor affecting CV performance is repetition and also output correlation. You can find the right bias-variance tradeoff by performing multiple rounds of cross-validation. Correlation between values affects the bias aspect.

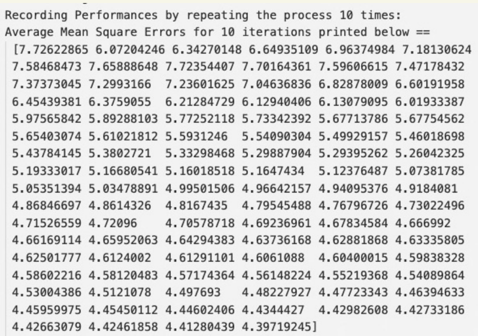
: The CV model is also affected by size of your data.

: LOOCV versus K-Fold and model / computation type will affect the efficiency of cross validation.

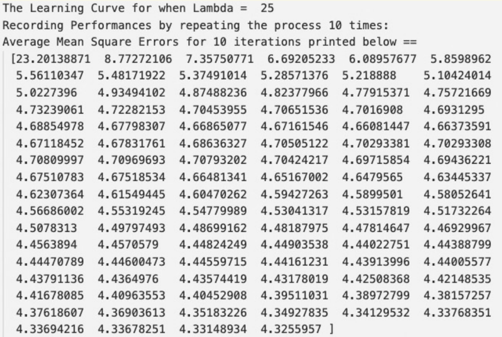
Question 4



: Above are the 3 learning curves at Lambda values 1, 25 and 50. We do so by generating the MSE at increasing subset sizes which are generated randomly.

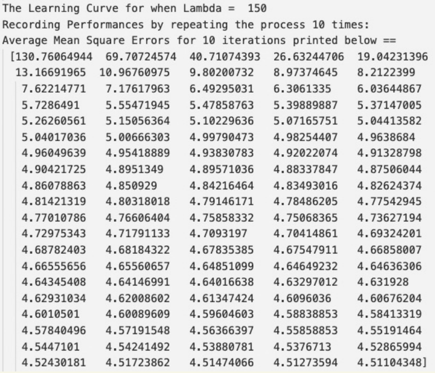


```
Recording Performances by repeating the process 10 times:
Average Mean Square Errors for 10 iterations printed below ==
 [7.72622865 6.07204246 6.34270148 6.64935109 6.96374984 7.18130624
 7.58468473 7.65888648 7.72354407 7.70164361 7.59606615 7.47178432
 7.37373045 7.2993166  7.23601625 7.04636836 6.82878009 6.60191958
 6.45439381 6.3759055  6.21284729 6.12940406 6.13079095 6.01933387
 5.97565842 5.89288103 5.77252118 5.73342392 5.67713786 5.67754562
 5.65403074 5.61021812 5.5931246  5.54090304 5.49929157 5.46018698
 5.43784145 5.3802721  5.33298468 5.29887904 5.29395262 5.26042325
 5.19333017 5.16680541 5.16018518 5.1647434  5.12376487 5.07381785
 5.05351394 5.03478891 4.99501506 4.96642157 4.94095376 4.9184081
 4.86846697 4.8614326  4.8167435  4.79545488 4.76796726 4.73022496
 4.71526559 4.72096    4.70578718 4.69236961 4.67834584 4.666992
 4.66169114 4.65952063 4.64294383 4.63736168 4.62881868 4.63335805
 4.62501777 4.6124002  4.61291101 4.6061088  4.60400015 4.59838328
 4.58602216 4.58120483 4.57174364 4.56148224 4.55219368 4.54089864
 4.53004386 4.5121078  4.497693   4.48227927 4.47723343 4.46394633
 4.45959975 4.45450112 4.44602406 4.4344427  4.42982608 4.42733186
 4.42663079 4.42461858 4.41280439 4.39719245]
```



```
The Learning Curve for when Lambda =  25
Recording Performances by repeating the process 10 times:
Average Mean Square Errors for 10 iterations printed below ==
 [23.20138871  8.77272106  7.35750771  6.69205233  6.08957677  5.8598962
  5.56110347  5.48171922  5.37491014  5.28571376  5.218888    5.10424014
  5.0227396   4.93494102  4.87488236  4.82377966  4.77915371  4.75721669
  4.73239061  4.72282153  4.70453955  4.70651536  4.7016908   4.6931295
  4.68854978  4.67798307  4.66865077  4.67161546  4.66081447  4.66373591
  4.67118452  4.67831761  4.68636327  4.70505122  4.70293381  4.70293308
  4.70809997  4.70969693  4.70793202  4.70424217  4.69715854  4.69436221
  4.67510783  4.67518534  4.66481341  4.65167002  4.6479565   4.63445337
  4.62307364  4.61549445  4.60470262  4.59427263  4.5899501   4.58052641
  4.56686002  4.55319245  4.54779989  4.53041317  4.53157819  4.51732264
  4.5078313   4.49797493  4.48699162  4.48187975  4.47814647  4.46929967
  4.4563894   4.4570579   4.44824249  4.44903538  4.44022751  4.44388799
  4.44470789  4.44600473  4.44559715  4.44161231  4.43913996  4.44005577
  4.43791136  4.4364976   4.43574419  4.43178019  4.42508368  4.42148535
  4.41678085  4.40963553  4.40452908  4.39511031  4.38972799  4.38157257
  4.37618607  4.36903613  4.35183226  4.34927835  4.34129532  4.33768351
  4.33694216  4.33678251  4.33148934  4.3255957 ]
```

This is the generated data for Lambda values 1, 25 & 50, each run 10 times. The data is recorded in the graphs above.



```
The Learning Curve for when Lambda =  150
Recording Performances by repeating the process 10 times:
Average Mean Square Errors for 10 iterations printed below ==
 [130.76064944  69.70724574  40.71074393  26.63244706  19.04231396
  13.16691965  10.96760975   9.80200732   8.97374645   8.2122399
   7.62214771   7.17617963   6.49295031   6.3061335    6.03644867
   5.7286491    5.55471945   5.47858763   5.39889887   5.37147005
   5.26260561   5.15056364   5.10229636   5.07165751   5.04413582
   5.04017036   5.00666303   4.99790473   4.98254407   4.9638684
   4.96049639   4.95418889   4.93830783   4.92022074   4.91328798
   4.90421725   4.8951349    4.89571036   4.88337847   4.87506044
   4.86078863   4.850929     4.84216464   4.83493016   4.82624374
   4.81421319   4.80318018   4.79146171   4.78486205   4.77542945
   4.77010786   4.76606404   4.75858332   4.75068365   4.73627194
   4.72975343   4.71791133   4.7093197    4.70414861   4.69324201
   4.68782403   4.68184322   4.67835385   4.67547911   4.66858007
   4.66555656   4.65560657   4.64851099   4.64649232   4.64636306
   4.64345408   4.64146991   4.64016638   4.63297012   4.631928
   4.62931034   4.62008602   4.61347424   4.6096036    4.60676204
   4.6010501    4.60089609   4.59604603   4.58838853   4.58413319
   4.57840496   4.57191548   4.56366397   4.55858853   4.55191464
   4.5447101    4.54241492   4.53880781   4.5376713    4.52865994
   4.52430181   4.51723862   4.51474066   4.51273594   4.51104348]
```