Report

Task 1:

PCA, LDA, and Kernel PCA are all included in the folder and implemented in the main.py.

Task 2:

Three dimensionality reduction methods are compared based on the iris dataset (see Table 1). Three dimensionality reduction methods are compared based on the MNIST dataset (see Table 2).

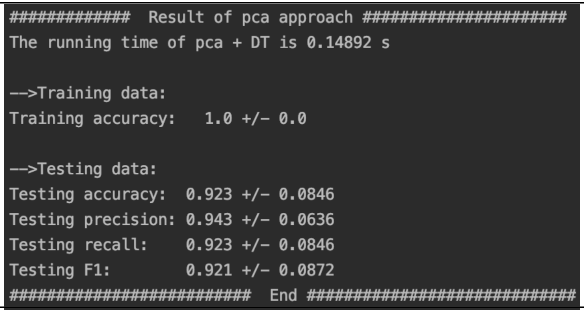
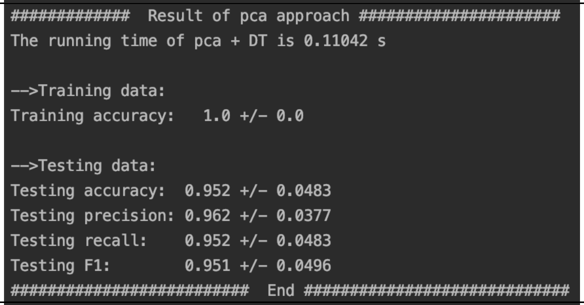
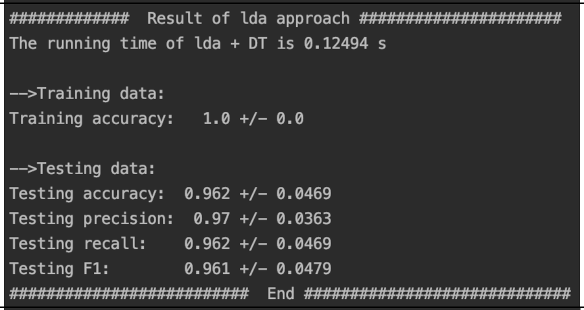
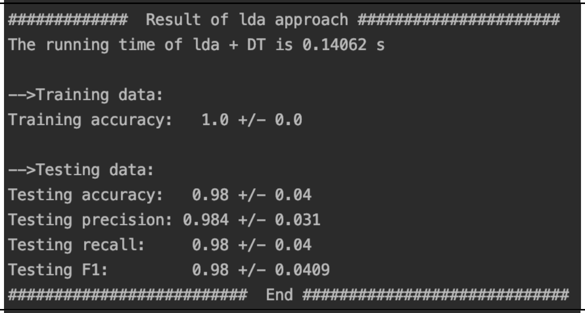
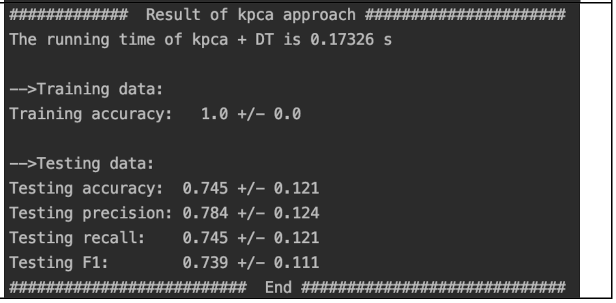
Output results:

- Training accuracy
- Testing accuracy
- Precision
- Recall
- F1

The above results are inserted in the last column in each table. Each row represents one testing case.

Cross-validation is considered before model training (see in main.py).

Table 1 Performance of different dimensionality reduction methods [Iris dataset]

Dimensionality Reduction Method	Parameters			Run Time and Model Evaluation Parameters
	n_components	kernel	gamma	
PCA	1	-	-	 ############ Result of pca approach ##################### The running time of pca + DT is 0.14892 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.923 +/- 0.0846 Testing precision: 0.943 +/- 0.0636 Testing recall: 0.923 +/- 0.0846 Testing F1: 0.921 +/- 0.0872 ######################### End ############################
	3	-	-	 ############ Result of pca approach ##################### The running time of pca + DT is 0.11042 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.952 +/- 0.0483 Testing precision: 0.962 +/- 0.0377 Testing recall: 0.952 +/- 0.0483 Testing F1: 0.951 +/- 0.0496 ######################### End ############################
LDA	1	-	-	 ############ Result of lda approach ##################### The running time of lda + DT is 0.12494 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.962 +/- 0.0469 Testing precision: 0.97 +/- 0.0363 Testing recall: 0.962 +/- 0.0469 Testing F1: 0.961 +/- 0.0479 ######################### End ############################
	3	-	-	 ############ Result of lda approach ##################### The running time of lda + DT is 0.14062 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.98 +/- 0.04 Testing precision: 0.984 +/- 0.031 Testing recall: 0.98 +/- 0.04 Testing F1: 0.98 +/- 0.0409 ######################### End ############################
Kernel PCA	1	rbf	5	 ############ Result of kpca approach ##################### The running time of kpca + DT is 0.17326 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.745 +/- 0.121 Testing precision: 0.784 +/- 0.124 Testing recall: 0.745 +/- 0.121 Testing F1: 0.739 +/- 0.111 ######################### End ############################

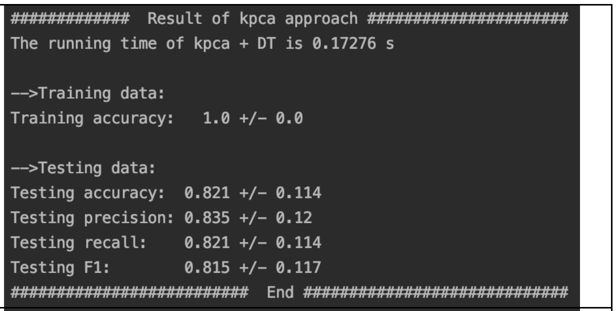
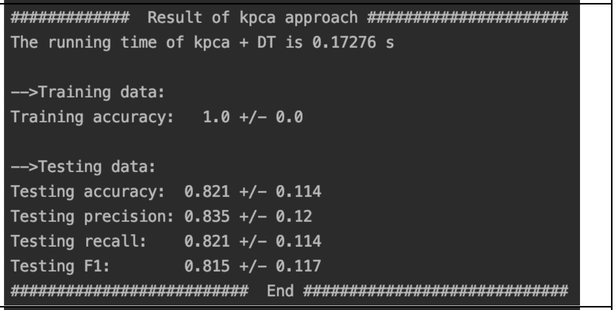
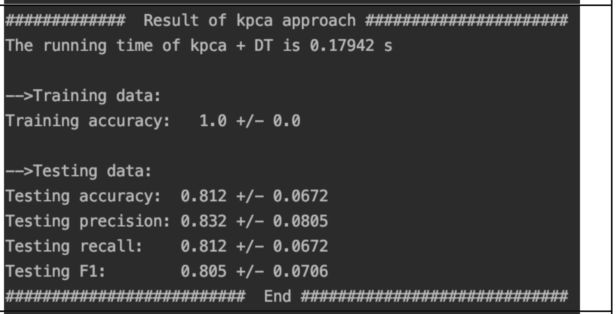
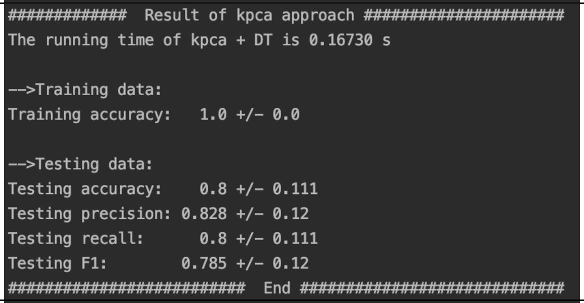
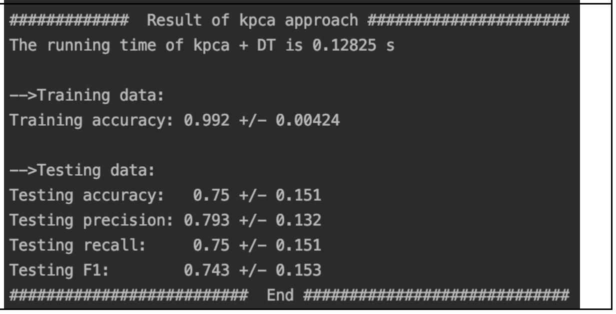
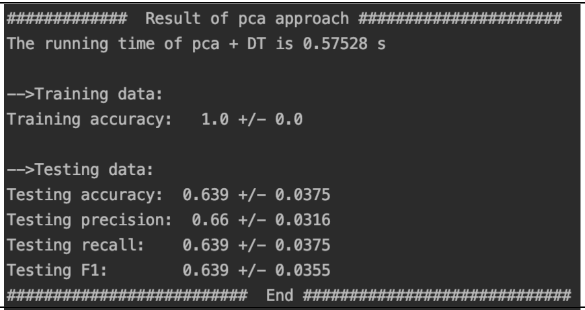
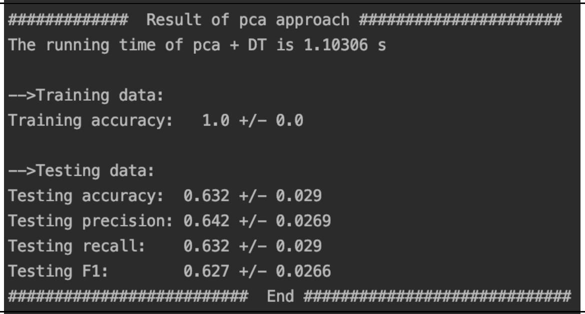
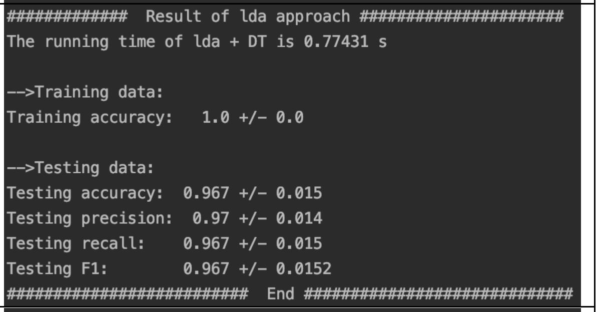
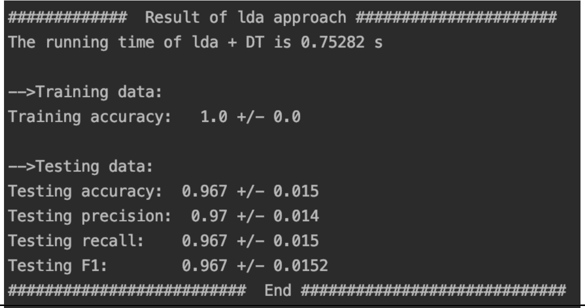
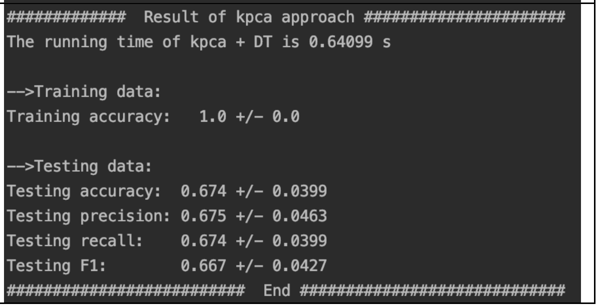
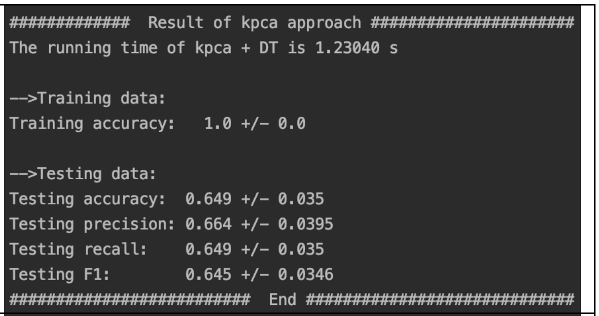
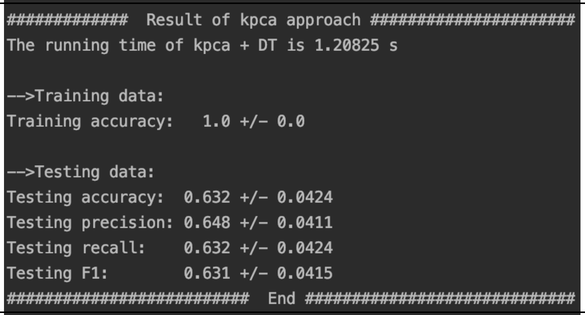
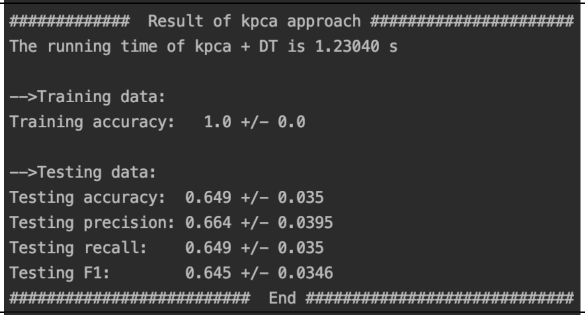
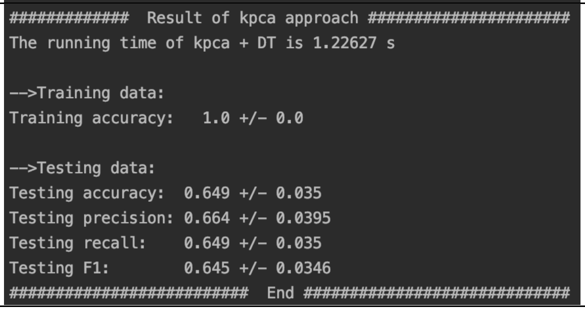
	3	rbf	5	 ############ Result of kpca approach ##################### The running time of kpca + DT is 0.17276 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.821 +/- 0.114 Testing precision: 0.835 +/- 0.12 Testing recall: 0.821 +/- 0.114 Testing F1: 0.815 +/- 0.117 ######################### End ############################
	3	rbf	5	 ############ Result of kpca approach ##################### The running time of kpca + DT is 0.17276 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.821 +/- 0.114 Testing precision: 0.835 +/- 0.12 Testing recall: 0.821 +/- 0.114 Testing F1: 0.815 +/- 0.117 ######################### End ############################
	3	sigmoid	5	 ############ Result of kpca approach ##################### The running time of kpca + DT is 0.17942 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.812 +/- 0.0672 Testing precision: 0.832 +/- 0.0805 Testing recall: 0.812 +/- 0.0672 Testing F1: 0.805 +/- 0.0706 ######################### End ############################
	2	rbf	5	 ############ Result of kpca approach ##################### The running time of kpca + DT is 0.16730 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.8 +/- 0.111 Testing precision: 0.828 +/- 0.12 Testing recall: 0.8 +/- 0.111 Testing F1: 0.785 +/- 0.12 ######################### End ############################
	2	rbf	15	 ############ Result of kpca approach ##################### The running time of kpca + DT is 0.12825 s -->Training data: Training accuracy: 0.992 +/- 0.00424 -->Testing data: Testing accuracy: 0.75 +/- 0.151 Testing precision: 0.793 +/- 0.132 Testing recall: 0.75 +/- 0.151 Testing F1: 0.743 +/- 0.153 ######################### End ############################

Table 2 Performance of different dimensionality reduction methods [MNIST dataset]

Dimensionality Reduction Method	Parameters			Run Time and Model Evaluation Parameters
	n_components	kernel	gamma	
PCA	10	-	-	 ############# Result of pca approach ###################### The running time of pca + DT is 0.57528 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.639 +/- 0.0375 Testing precision: 0.66 +/- 0.0316 Testing recall: 0.639 +/- 0.0375 Testing F1: 0.639 +/- 0.0355 ########################## End #############################
	50	-	-	 ############# Result of pca approach ###################### The running time of pca + DT is 1.10306 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.632 +/- 0.029 Testing precision: 0.642 +/- 0.0269 Testing recall: 0.632 +/- 0.029 Testing F1: 0.627 +/- 0.0266 ########################## End #############################
LDA	10	-	-	 ############# Result of lda approach ###################### The running time of lda + DT is 0.77431 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.967 +/- 0.015 Testing precision: 0.97 +/- 0.014 Testing recall: 0.967 +/- 0.015 Testing F1: 0.967 +/- 0.0152 ########################## End #############################
	50	-	-	 ############# Result of lda approach ###################### The running time of lda + DT is 0.75282 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.967 +/- 0.015 Testing precision: 0.97 +/- 0.014 Testing recall: 0.967 +/- 0.015 Testing F1: 0.967 +/- 0.0152 ########################## End #############################
Kernel PCA	10	cosine	50	 ############# Result of kpca approach ###################### The running time of kpca + DT is 0.64099 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.674 +/- 0.0399 Testing precision: 0.675 +/- 0.0463 Testing recall: 0.674 +/- 0.0399 Testing F1: 0.667 +/- 0.0427 ########################## End #############################

	50	cosine	50		 ############ Result of kpca approach #################### The running time of kpca + DT is 1.23040 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.649 +/- 0.035 Testing precision: 0.664 +/- 0.0395 Testing recall: 0.649 +/- 0.035 Testing F1: 0.645 +/- 0.0346 ####################### End ###########################
	50	sigmoid	50		 ############ Result of kpca approach #################### The running time of kpca + DT is 1.20825 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.632 +/- 0.0424 Testing precision: 0.648 +/- 0.0411 Testing recall: 0.632 +/- 0.0424 Testing F1: 0.631 +/- 0.0415 ####################### End ###########################
	100	cosine	50		 ############ Result of kpca approach #################### The running time of kpca + DT is 1.23040 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.649 +/- 0.035 Testing precision: 0.664 +/- 0.0395 Testing recall: 0.649 +/- 0.035 Testing F1: 0.645 +/- 0.0346 ####################### End ###########################
	50	cosine	150		 ############ Result of kpca approach #################### The running time of kpca + DT is 1.22627 s -->Training data: Training accuracy: 1.0 +/- 0.0 -->Testing data: Testing accuracy: 0.649 +/- 0.035 Testing precision: 0.664 +/- 0.0395 Testing recall: 0.649 +/- 0.035 Testing F1: 0.645 +/- 0.0346 ####################### End ###########################

Task 3

The Iris (Table 1) and MNIST (Table 2) datasets are both processed by three dimensionality reduction methods.

- Iris data

  Read the data from an input file, the file name is not used during inputting process.
- MNIST

  The original MNIST data contains 70000 instances, which is too many to run in laptop. 2% of the dataset is selected as a sub dataset and save as 'MNIST.csv' in the same folder. In the main code, only this sub dataset is used for training and testing, including 1400 instances. A data preprocess is used to achieve the above process, and the corresponding code is placed outside the main code.

Task 4

In Tables 1 and 2, three dimensionality reduction methods are tested by different parameters. The model performances are evaluated with parameters, including training accuracy, testing accuracy, precision, recall, and F1.

<u>For iris dataset:</u>

- For PCA, the parameter of n_components is chosen for 1 and 3. The increase of n_components leads to overall accuracy increase for testing data from 0.92 to 0.95. The training accuracy keeps 1.
- For LDA, the parameter of n_components is also chosen for 1 and 3. The accuracies is obviously higher than that of PCA. The highest accuracy for testing data is higher than 0.98.
- For Kernel PCA, no matter what combinations of parameters are set, the overall accuracy is only close to 0.8. Using the kernel PCA doesn't increase the accuracy and decrease the predicting performance instead. The potential reason may ascribe to the simplicity of the iris dataset.
- Conclusion: the accuracy of training data is always 1, which means the decision tree model can totally separate the instances and lead to an overfitting issue in prediction

For MNIST dataset:

- For PCA, the parameter of n_components is chosen for 10 and 30. The increase of n_components doesn’t increase the accuracy, which is nearly 0.63 for both cases.
- For LDA, the parameter of n_components is also chosen for 10 and 50. With the help of LDA, the accuracy increase to 0.96, which is a relatively high improvement as compared with PCA. The selection of n_components doesn’t influence the predict.
- For Kernel PCA, no matter what combinations of parameters are set, the overall accuracy return to about 0.65, which is almost the same accuracy with PCA.
- Conclusion:
  - the accuracy of training data is always 1, which means the decision tree model can totally separate the instances and lead to an overfitting issue in prediction.
  - The LDA method is proved to be the most powerful approach to increase the model accuracy for MNIST dataset.

Task 5

A readme.txt file is attached.