SMAI ASSIGNMENT-2 REPORT

Sai Krishna Charan Dara

20171140

November 3, 2019

Questions

Question_1

1(a). What are eigen faces?

- Eigenfaces are an ***orthogonal basis set*** (because covariance matrix is PSD) from which most of all faces can be constructed . They are basically eigen vectors of the image after smoothening or reshaping images to a 2-D vector.
- They are blurry depictions of faces that each highlight a certain type of feature.
- Eigenfaces are constructed by training on a set of real faces.
- The goal is to create the minimum number of eigenfaces that can adequately represent the entire training set. It is a type of ***Principal Component Analysis.***

1(b). Number of Eigenvectors required to satisfactorily represent a face?

- We know from PCA that maximum variance of the data lies along the eigen vector that has highest eigen value and second highest variance along the eigen vector with second highest eigen value and so on. Our purpose is to choose sufficient eigen vectors from which most of all faces can be reconstructed. This is often decided by looking at how much information is lost in doing the dimensionality reduction. The estimate of this

$$\frac{\sum_{i=k+1}^{d} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

Often k is picked such that the above ratio is less than 5% or 10%

- This can also be determined by looking at the cumulative *explained variance ratio* as a function of the number of components. We take number of components corresponding to 0.96% cumulative variance.

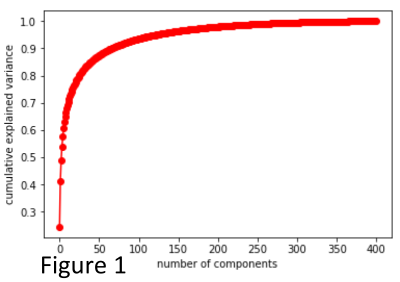


Figure 1

IMFDB DATA SET
Cumulative Variance chose here : 0.9601550671573126
Number of components to choose: 142

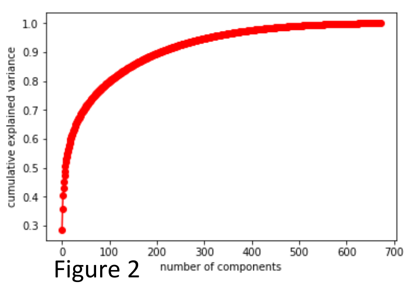


Figure 2

IIT-CFW DATA SET
Cumulative Variance chose here : 0.9601586908166865
Number of components to choose: 338

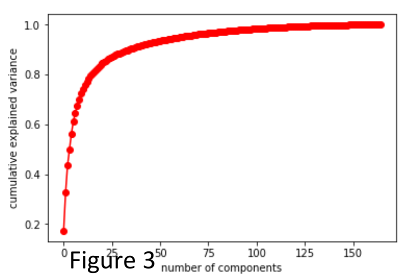


Figure 3

YALE DATA SET
Cumulative Variance chose here : 0.9601750671573126
Number of components to choose: 69

1(c). Which person/identity is difficult to represent compactly with fewer eigen vectors?

- We can say that the class with most reconstruction error is difficult to represent with fewer eigen vectors. Reconstruction error high implies its cannot be sufficiently represented by these eigen vectors.

<u>*For IMFDB Data Set*</u>

- Intuitively we can say that the person with features/face structure different from rest cannot be compactly represented with fewer eigen vectors because while doing PCA we project data on maximum variance side. The person who has odd features will not have most his features along the maximum variance side. Hence difficult. From my observation Sharukhan has odd features hence difficult to represent with few features.
- From Empirical Observations

  Class: 2 (Sharukhan) has most reconstruction loss. Loss: **0.03633969710344384**

For IIT-CFW Data Set

- From my observation Aishwarya Rai has odd features, hence difficult to represent with few features. She has cartoon images with different emotions from rest of others. Hence capturing different emotions requires more eigen vectors.
- From Empirical Observations

  Class: 3(Aishwarya Rai) has most reconstruction loss. Loss: **0.06345402349393735**

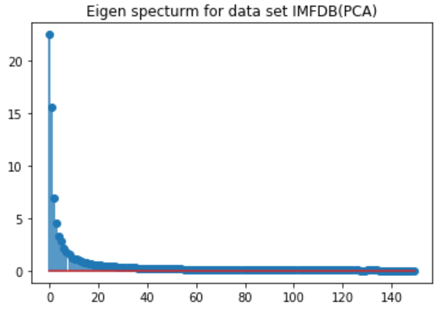
For YALE Data Set

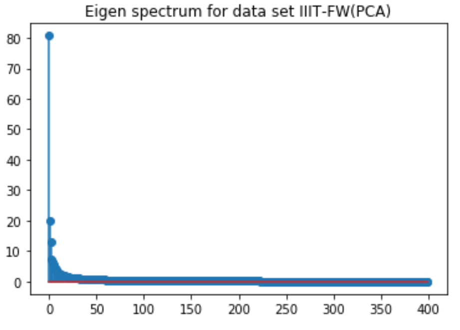
- Faces of the people in the Yale dataset to have very much similar facial features. Intuitively all classes requires almost same eigen faces to represent. Empirical loss calculated also have very near or similar values. But among all

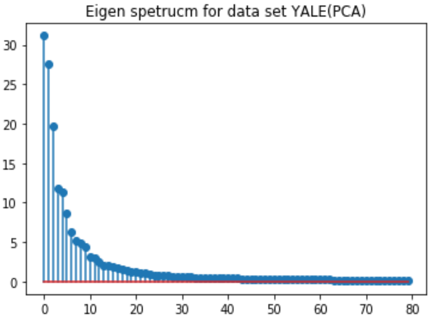
  Class: 0 has most reconstruction loss. Loss: **0.054167505236136**

1(d) Which dataset is difficult to represent compactly with fewer eigen vectors?

- From figure1,2,3 we saw that number of Eigen vectors required to have 0.96% cumulative variance are 142(4.6%),338(11%),69(2.2%) respectively for IMFDB,IIIT-CFW,YALE data sets.
- Empirically Yale data set can be represented with least number of eigen vectors.
- Intuitively also we can say that Yale Data Set has people whose features are very similar to each other (similar facial structures) and this data contains 95% of men . And, IMFDB AND IIIT-CFW has both gender people in equal proportion and hence more number of eigen vectors compared to Yale Data set. IIIT-CFW has cartoon features which vary in a significant amount from one person to other person and emotions of these people in images are widely varied. Hence more number of eigen vectors for IIIT-CFW. That is true empirically also.
- So Yale data set has least number of eigen vectors (69,2.2%) and IIIT-CFW has most number of eigen vectors(338,11%)







QUESTION_2

2(a) Comparative study of MLP classifier with different features.

- *Multi Layer Perceptron(MLP)*
  - Contains 2 hidden layers each with number of neurons 200.
  - Hidden Layers have activation function ReLu.
  - Output Layer have activation function SoftMax.
- Using too few neurons in the hidden layers will result in underfitting. Using too many neurons in the hidden layers can result in overfitting (too much information processing capacity and limited amount of information) and also increases computational time.
- Since this is a classification problem, at the output layer SoftMax is chosen. It maps outputs through probabilities.
- Hence choosing the hidden neurons in the range of input layers and output layers can give better performance. So for 3 data sets 200 is in the range of input and output layers and hence choosing this gave me good performance.

MLP on IMFDB Data Set

	Method	Reduced Space	Classification error	Accuracy	f1-score	precision	recall
0	PCA+MLP	142	23.75	76.25	0.768799	0.767000	0.780355
1	K_PCA+MLP	142	20.00	80.00	0.816575	0.816142	0.847863
2	LDA+MLP	7	2.50	97.50	0.970779	0.968750	0.979167
3	K_LDA+MLP	7	3.75	96.25	0.962051	0.957386	0.971814
4	VGG+MLP	4096	12.50	87.50	0.869233	0.872443	0.876966
5	RESNET+MLP	2048	8.75	91.25	0.905551	0.910795	0.913822

MLP on IIT-CFW Data Set

	Method	Reduced Space	Classification error	Accuracy	f1-score	precision	recall
0	PCA+MLP	338	48.888889	51.111111	0.497909	0.527204	0.495767
1	K_PCA+MLP	338	50.370370	49.629630	0.497590	0.511633	0.494148
2	LDA+MLP	7	0.740741	99.259259	0.990643	0.994565	0.987500
3	K_LDA+MLP	7	7.407407	92.592593	0.904292	0.909218	0.904600
4	VGG+MLP	4096	34.814815	65.185185	0.615395	0.683507	0.636994
5	RESNET+MLP	2048	3.703704	96.296296	0.956644	0.946726	0.971100

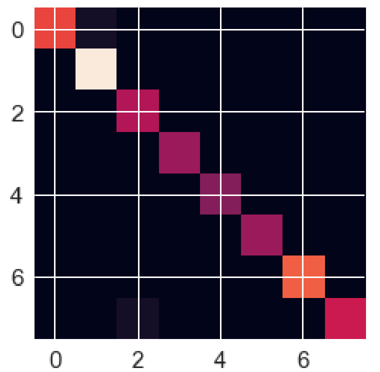
MLP on YALE Data Set

	Method	Reduced Space	Classification error	Accuracy	f1-score	precision	recall
0	PCA+MLP	69	9.090909	90.909091	0.865934	0.862821	0.882051
1	K_PCA+MLP	69	12.121212	87.878788	0.819048	0.845238	0.845238
2	LDA+MLP	7	0.000000	100.000000	1.000000	1.000000	1.000000
3	K_LDA+MLP	7	0.000000	100.000000	1.000000	1.000000	1.000000
4	VGG+MLP	4096	39.393939	60.606061	0.552698	0.602222	0.622222
5	RESNET+MLP	2048	0.000000	100.000000	1.000000	1.000000	1.000000

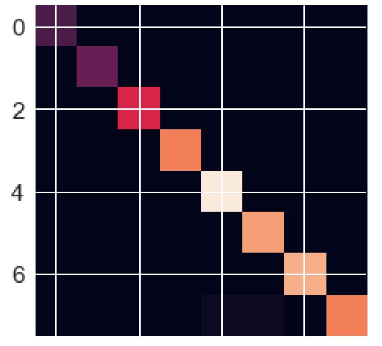
- LDA and KLDA (rbf kernel) works well here than others because LDA is a supervised learning. It projects the data in a manner such that distance between classes is maximized and variance within class is minimized. So doing feature LDA at first step itself completes most of the task and passing through MLP classifier does remaining task. Hence it has more accuracy.
- VGG and RESNET uses CNN and maps into higher dimensional space and works well than PCA.
- PCA has least accuracy here because PCA is unsupervised learning, it just projects the data on maximum variance side and has no role to play with which class it belongs, hence least accuracy.
- We can see that LDA for YALE produces accuracy 100% because the data is linearly separable and hence LDA achieves maximum accuracy. F1-score and accuracy are in a parallel fashion.
- Here I tried mixing various features and did MLP on this concatenated feature set but I didn't get improved accuracy . Just reduced space increases, no increase in performance.

BEST MODELS FOR DIFFERENT DATA SETS **CONFUSION MATRIX**

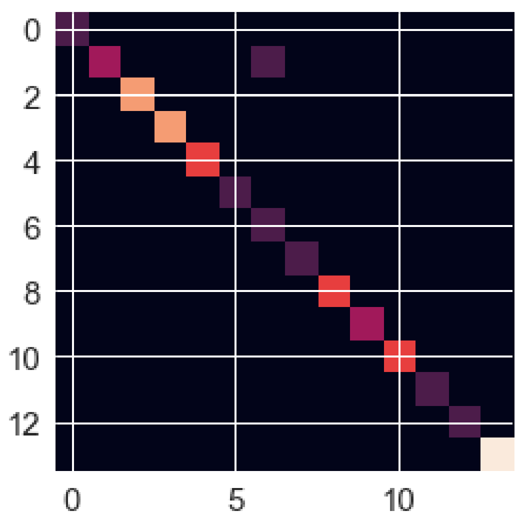
```
LDA+MLP for IMFDB
[[11  1  0  0  0  0  0  0]
 [ 0 18  0  0  0  0  0  0]
 [ 0  0  8  0  0  0  0  0]
 [ 0  0  0  7  0  0  0  0]
 [ 0  0  0  0  6  0  0  0]
 [ 0  0  0  0  0  7  0  0]
 [ 0  0  0  0  0  0 12  0]
 [ 0  0  1  0  0  0  0  9]]
```




```
LDA+MLP for CFW
[[ 5  0  0  0  0  0  0  0]
 [ 0  7  0  0  0  0  0  0]
 [ 0  0 14  0  0  0  0  0]
 [ 0  0  0 19  0  0  0  0]
 [ 0  0  0  0 26  0  0  0]
 [ 0  0  0  0  0 21  0  0]
 [ 0  0  0  0  0  0 22  0]
 [ 0  0  0  0  1  1  0 19]]
```




```
K_PCA(POLY)+MLP for Yale
[[1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 2 0 0 0 0 1 0 0 0 0 0 0 0]
 [0 0 4 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 4 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 3 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 3 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 2 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 3 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 5]]
```

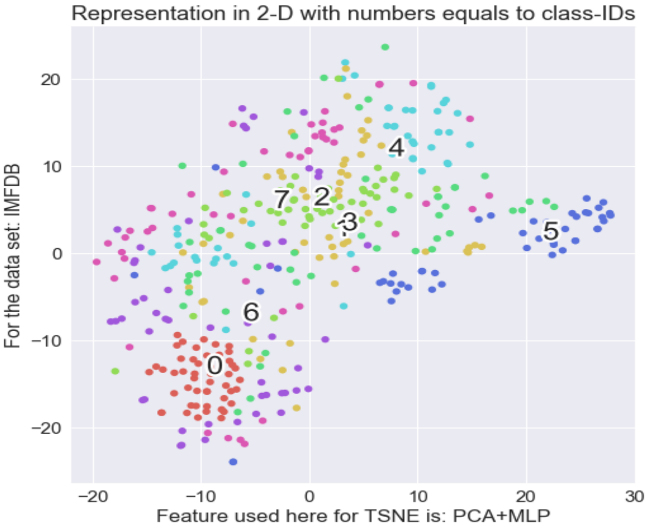


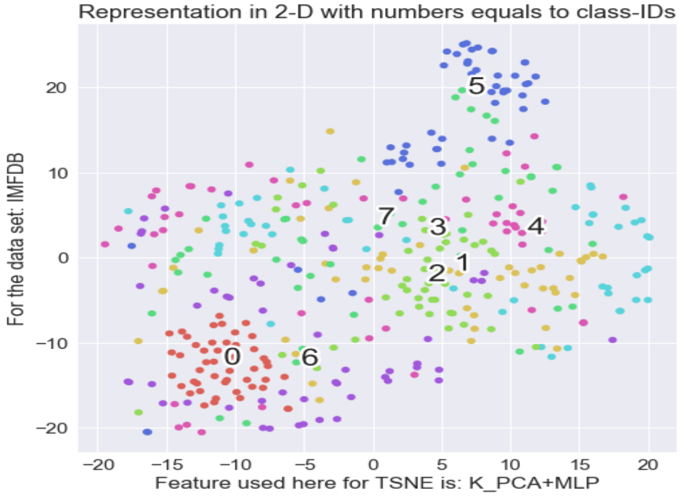
QUESTION_3

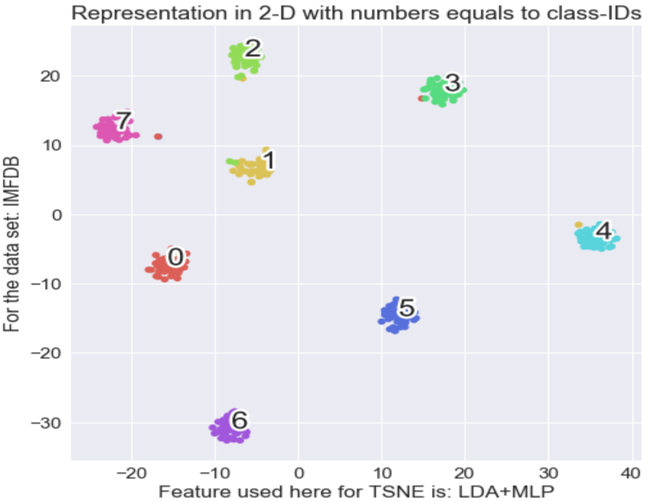
3(a) t-SNE based visualization

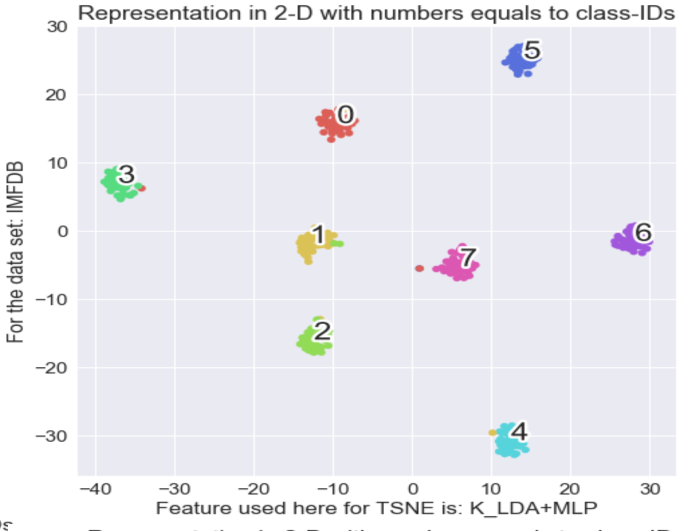
- t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.
- t-Distributed stochastic neighbor embedding (t-SNE) minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.
- t-SNE maps the multi-dimensional data to a lower dimensional space and attempts to find patterns in the data by identifying observed clusters based on similarity of data points with multiple features. However, after this process, the input features are no longer identifiable, and you cannot make any inference based only on the output of t-SNE. Hence it is mainly a data exploration and visualization technique.
- Here it is useful to find clusters of data . How data is separable by using different features mentioned.
- So, this makes sense with respect to visualization and find patterns in the data.

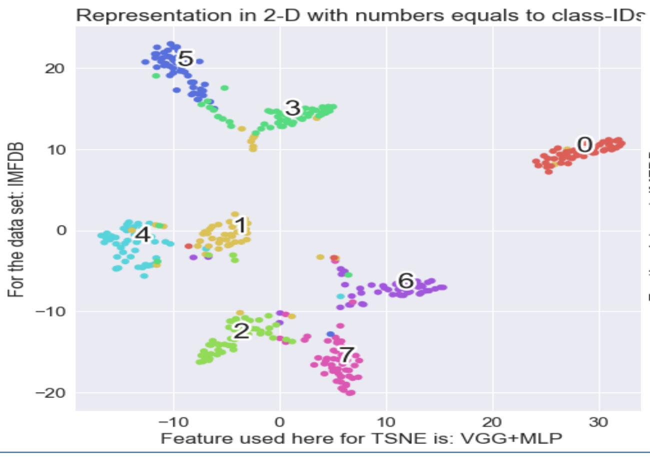
3(a) Do you see similar people coming together? something else?(For individual data sets)

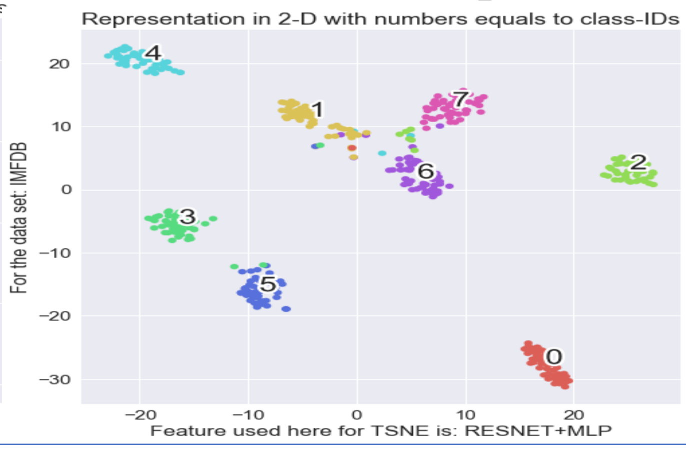


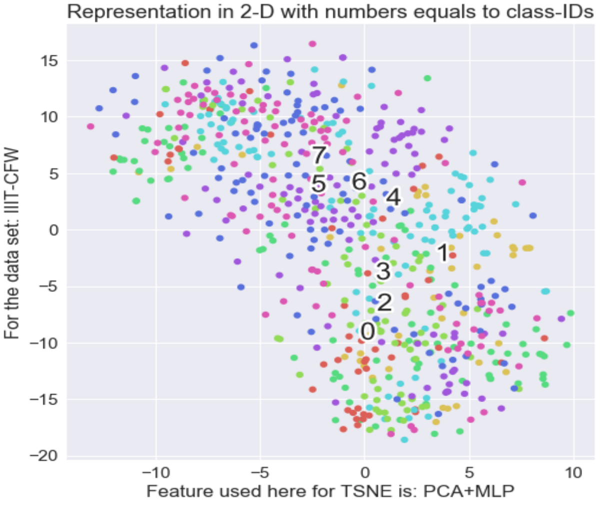


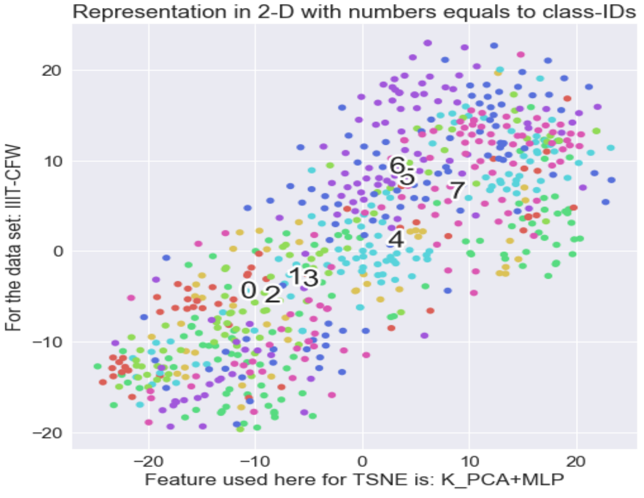


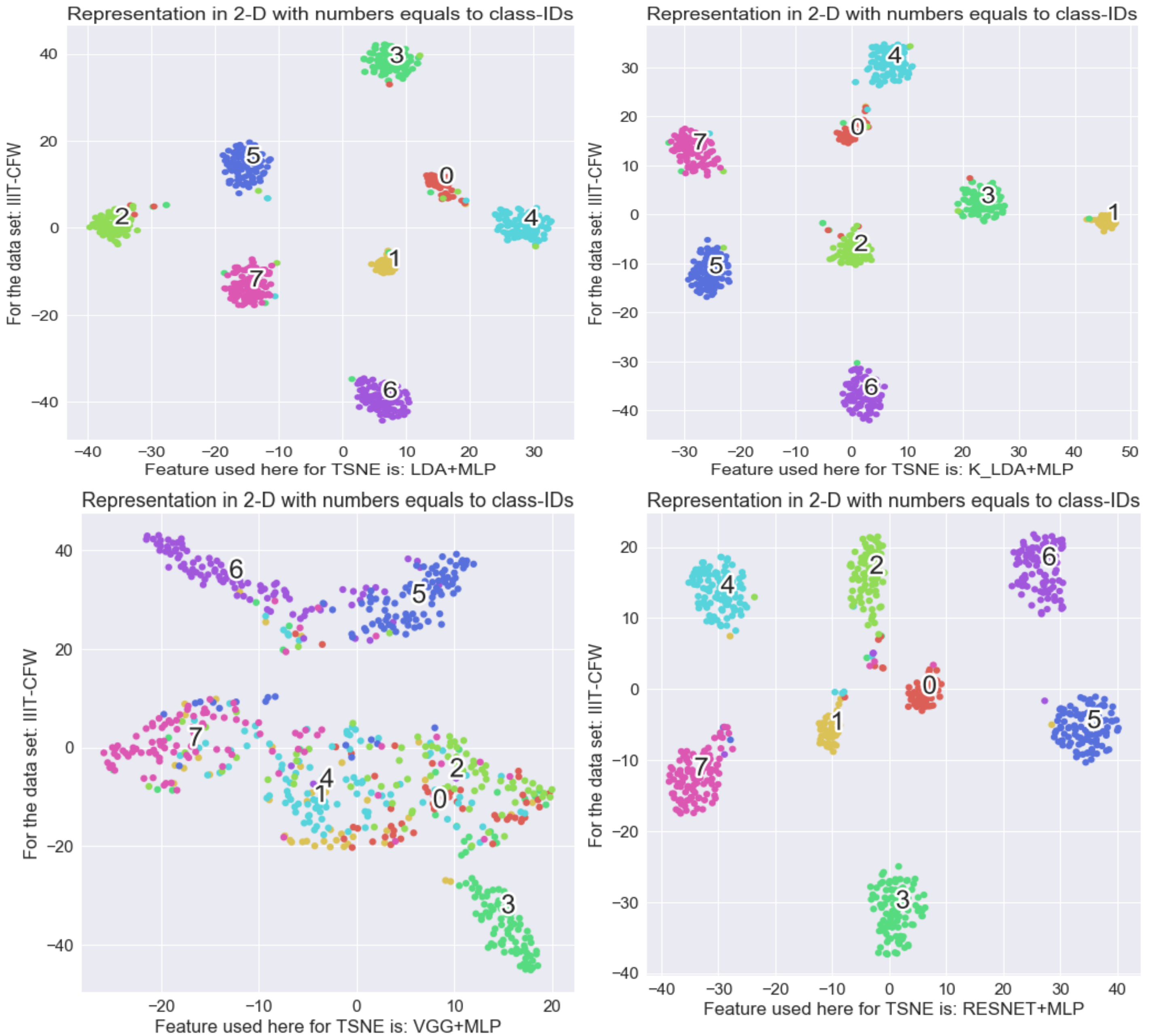


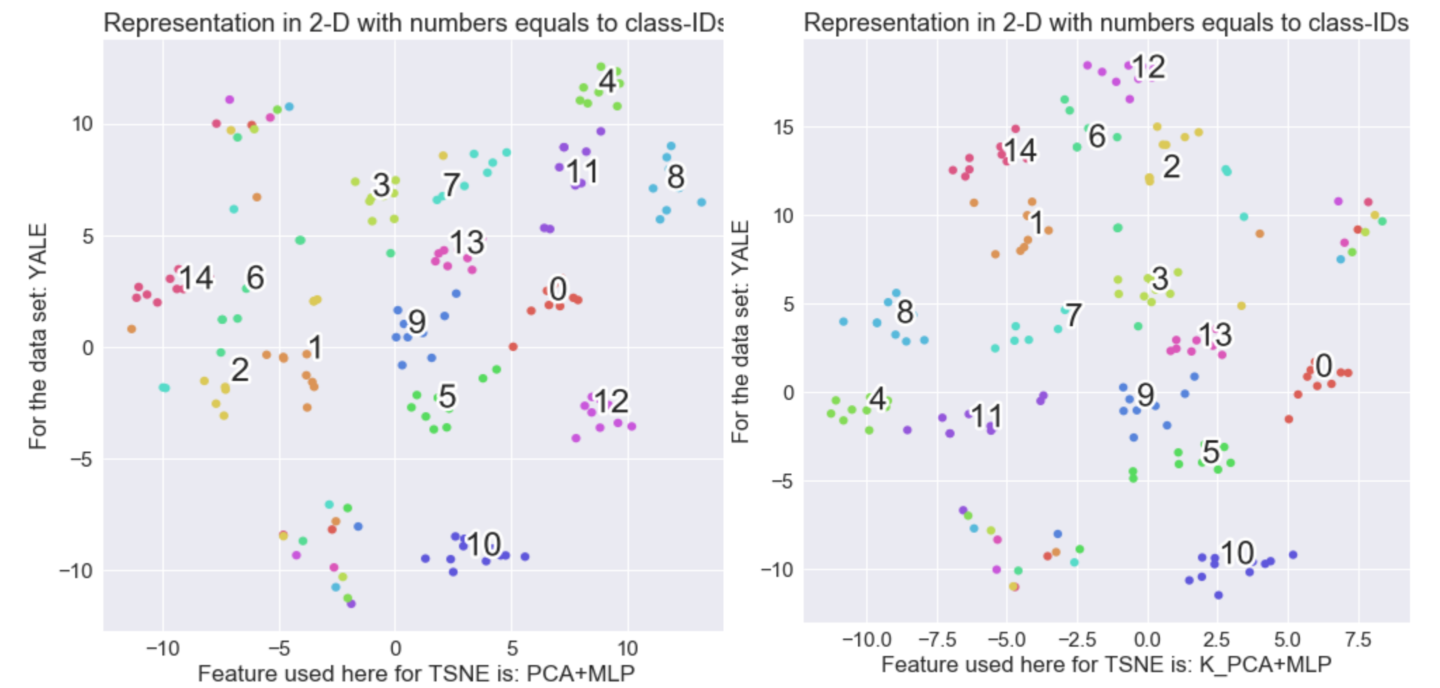


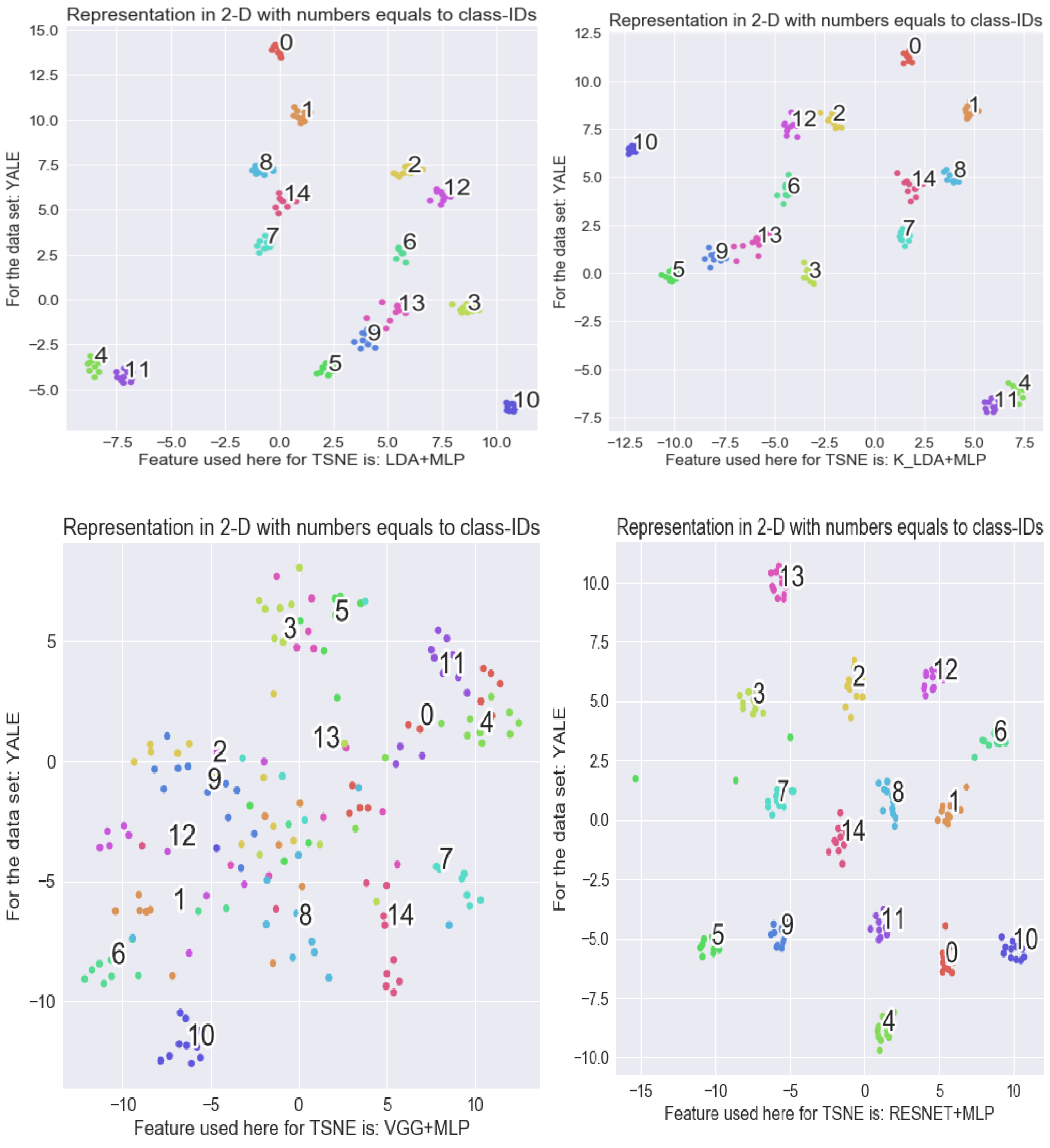






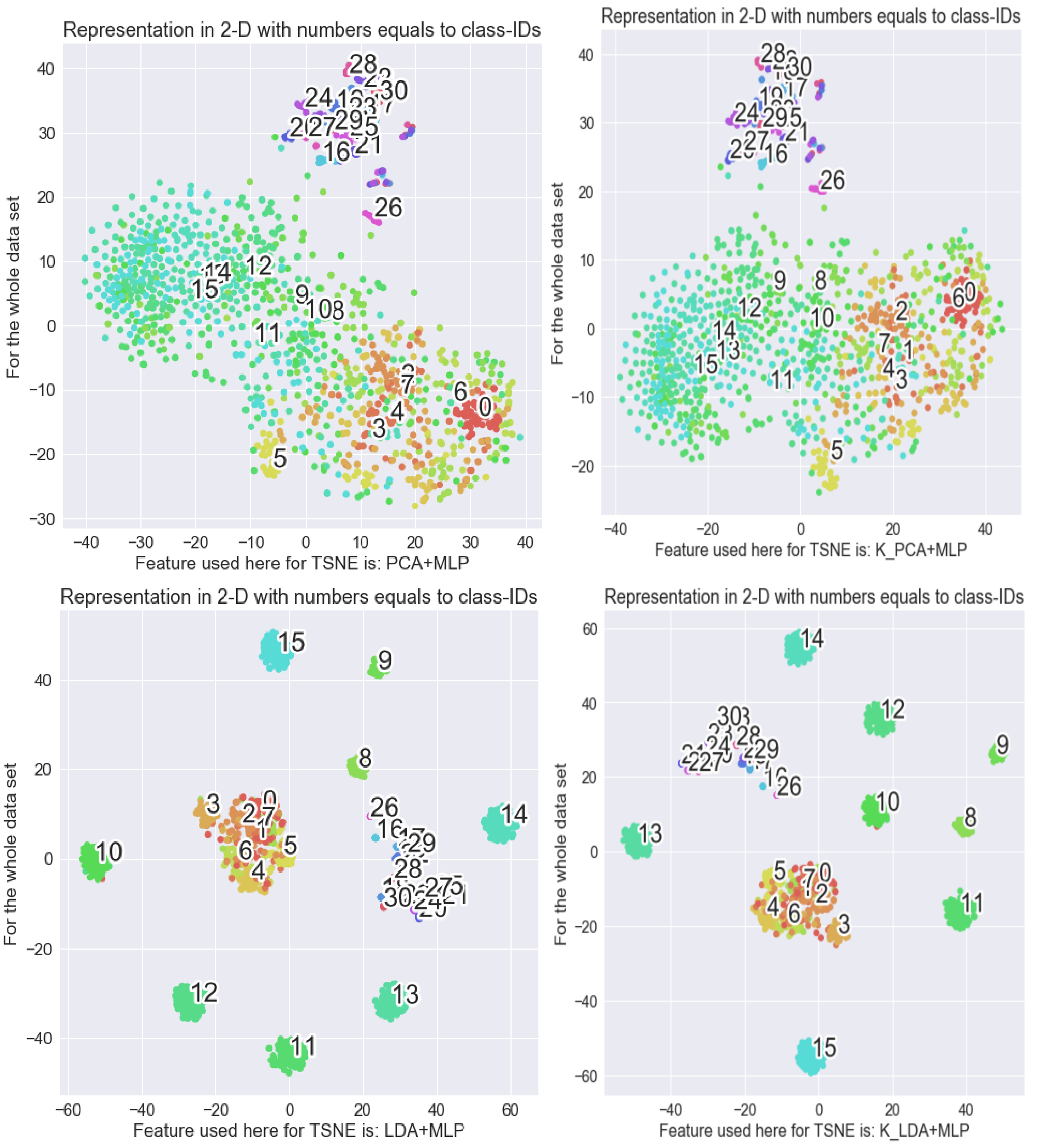

Representation in 2-D with numbers equals to class-IDs
For the data set: IIIT-CFW
Feature used here for TSNE is: LDA+MLP
Representation in 2-D with numbers equals to class-IDs
For the data set: IIIT-CFW
Feature used here for TSNE is: K_LDA+MLP
Representation in 2-D with numbers equals to class-IDs
For the data set: IIIT-CFW
Feature used here for TSNE is: VGG+MLP
Representation in 2-D with numbers equals to class-IDs
For the data set: IIIT-CFW
Feature used here for TSNE is: RESNET+MLP


Representation in 2-D with numbers equals to class-IDs
For the data set: YALE
Feature used here for TSNE is: PCA+MLP
Representation in 2-D with numbers equals to class-IDs
For the data set: YALE
Feature used here for TSNE is: K_PCA+MLP



- Here we can see that classes are closely clustered in LDA,KLDA,VGG,RESNET but in PCA there is no close clustering of the groups. This is because LDA,KLDA feature itself have property to minimize the mean between labels while maximizing variance and when this data pass through t-SNE there is further clear cut division of data. In a similar way VGG and RESNET due to their own properties the data is divided into different groups.
- As PCA is unsupervised learning, it does not take labels(yi's) into account. As t-SNE visualizes into different clusters. Hence they are separated but not effective as LDA's
- Therefore by looking above graphs, similar people came together.

3(a) Do you see similar people coming together? something else? (For Whole data set)



- 0-7 Labels of IMFDB | 8-15 Labels of IIIT-CFW | 16-30 Labels of Yale
- We see that the some of labels are clearly separated and some labels have some similarities because the distance between means of those is very low, implies features are closely related.
- Yes, we can see similar people coming together.

QUESTION_4

4 a) How do we formulate the problem using KNN?

- KNN is not suitable for the large dimensional data. In such cases to improve the performance. Hence we do feature extraction by various methods and reduce the dimension. Then perform KNN to classify the labels.
- For a new image we find the K nearest neighbors and assign the class which most neighbors have to the new image. The distance metric may be Euclidean/ Manhattan depending upon type of data set.
- There is no need to train a model for generalization and KNN can be used for non-linear data as well.

4 b) How do we analyze the performance ?

- We want more number of classifications to be correct and very less to be misclassified. Hence we can use ***accuracy*** metric to judge this. And out of all classifications that are labelled correct how many actually are correct? This is also a valid metric here. Hence we use ***precision***. Higher precision relates to low false positive rate.
- Accuracy and precision are correct metrics to analyze the performance here.

4 c) Empirical results with all the representations and variations in K.(For 3 data sets)

k= 5	Method	Reduced Space	verification error	Accuracy	precision
0	PCA+MLP	142	42.50	57.50	0.643601
1	K_PCA+MLP	142	38.75	61.25	0.645085
2	LDA+MLP	7	3.75	96.25	0.946726
3	K_LDA+MLP	7	3.75	96.25	0.960871
4	VGG+MLP	4096	11.25	88.75	0.888850
5	RESNET+MLP	2048	2.50	97.50	0.977778
0	PCA+MLP	338	67.407407	32.592593	0.540931
1	K_PCA+MLP	338	67.407407	32.592593	0.537977
2	LDA+MLP	7	2.962963	97.037037	0.973214
3	K_LDA+MLP	7	2.222222	97.777778	0.980655
4	VGG+MLP	4096	36.296296	63.703704	0.583482
5	RESNET+MLP	2048	2.222222	97.777778	0.982230
0	PCA+MLP	69	18.181818	81.818182	0.720513
1	K_PCA+MLP	69	18.181818	81.818182	0.834184
2	LDA+MLP	7	0.000000	100.000000	1.000000
3	K_LDA+MLP	7	0.000000	100.000000	1.000000
4	VGG+MLP	4096	39.393939	60.606061	0.535714
5	RESNET+MLP	2048	0.000000	100.000000	1.000000

For k= 7

	Method	Reduced Space	verification error	Accuracy	precision
0	PCA+MLP	142	41.25	58.75	0.679267
1	K_PCA+MLP	142	45.00	55.00	0.642878
2	LDA+MLP	7	3.75	96.25	0.962607
3	K_LDA+MLP	7	1.25	98.75	0.979167
4	VGG+MLP	4096	12.50	87.50	0.890422
5	RESNET+MLP	2048	3.75	96.25	0.956101
0	PCA+MLP	338	68.888889	31.111111	0.471230
1	K_PCA+MLP	338	78.518519	21.481481	0.407065
2	LDA+MLP	7	2.962963	97.037037	0.967071
3	K_LDA+MLP	7	2.222222	97.777778	0.981477
4	VGG+MLP	4096	33.333333	66.666667	0.713457
5	RESNET+MLP	2048	2.222222	97.777778	0.978692
0	PCA+MLP	69	18.181818	81.818182	0.828889
1	K_PCA+MLP	69	27.272727	72.727273	0.716667
2	LDA+MLP	7	0.000000	100.000000	1.000000
3	K_LDA+MLP	7	0.000000	100.000000	1.000000
4	VGG+MLP	4096	57.575758	42.424242	0.388889
5	RESNET+MLP	2048	0.000000	100.000000	1.000000

QUESTION_5 CARTOON VS REAL IMAGES

5 a)Briefly explain the problem. Why the problem is not trivial ?

- The problem is to distinguish between cartoon and real images. Cartoon Images(IIIT-CFW data set) and Real Images(IMFDB and YALE).
- This is not trivial problem as there are many different features that differentiate cartoon and real images.
- The cartoon have many varied facial structure compared to real images. The depiction of emotions for cartoon and real images are highly varied. Hence it is worth classifying them.

5 b) Why a solution to this may be of some use. Suggest good applications. Suggest good reasons why solving your problem is ?

- There are many use cases to this solution. World is full of fake people. They morph the real images to something similar to cartoon type and put it in social media like facebook , google and many other sites. Hence it is necessary to classify them what is real and what is not.
- The morphing also happens in government issued cards like aadhar card, ration card, driving license. Hence it is necessary for government to classify these images.
- The problem is interesting as there are many features to analyze for both type of images and also I like the use cases of this. Hence I am solving this problem.

5 c) Explain your experimental pipeline, splits, evaluation metrics, quantitative results, qualitative results.

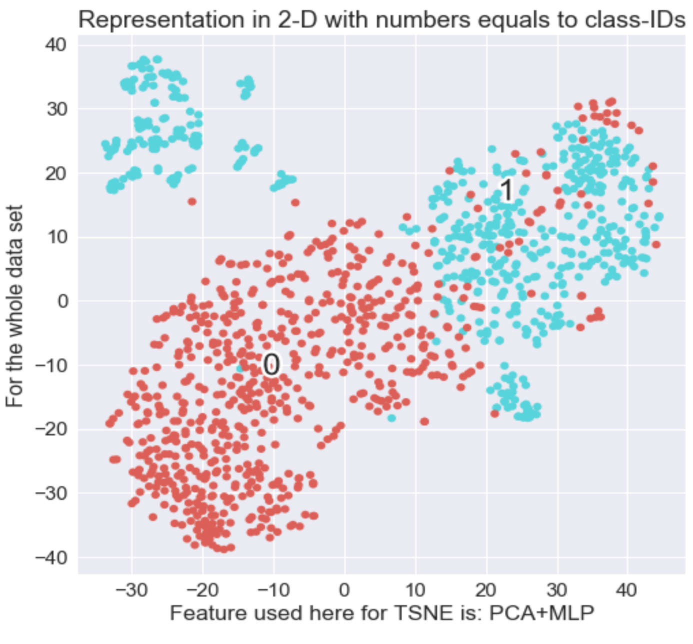
- I used various features like PCA,LDA,KLDA,KPCA for this data set and sent into the MLP classifier.
- *Multi Layer Perceptron(MLP)*
  - Contains 2 hidden layers each with number of neurons 200.
  - Hidden Layers have activation function ReLu.
  - Output Layer have activation function SoftMax

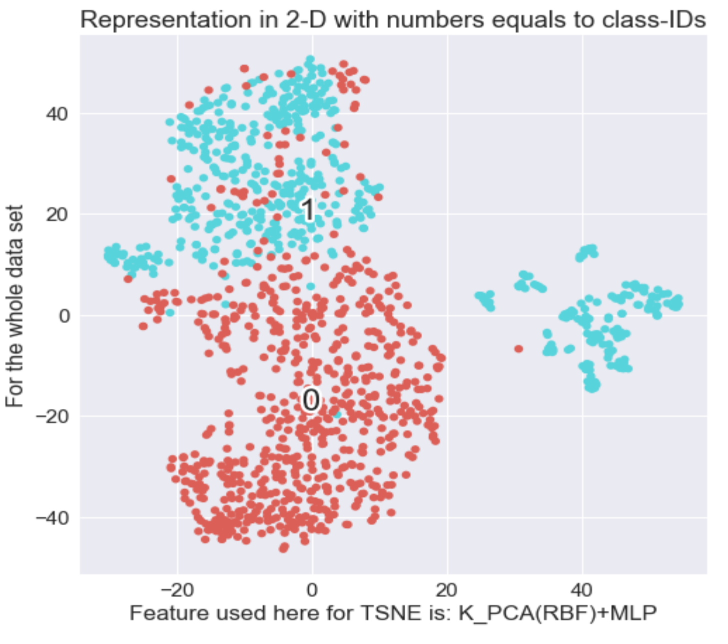
- Train-test split 20%

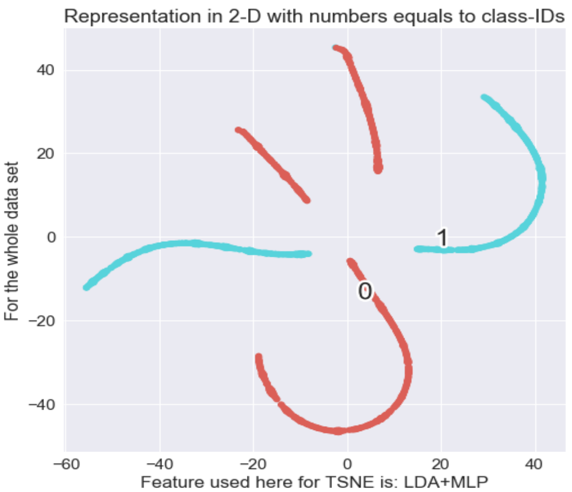
QUALITATIVE RESULTS

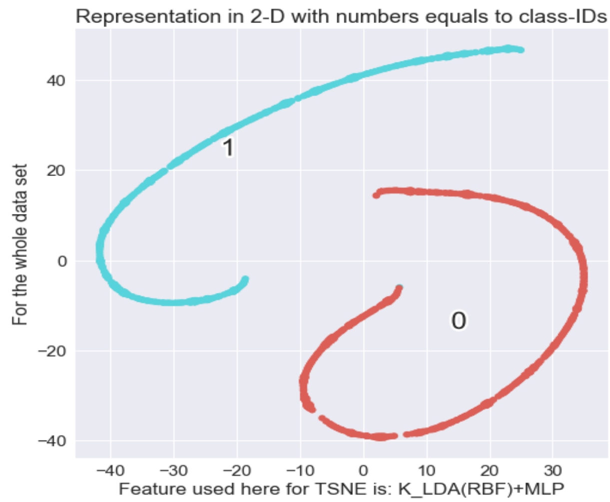
	Method	Reduced Space	Classification error	Accuracy	f1-score	precision	recall
0	PCA+MLP	40	1.612903	98.387097	0.983845	0.984224	0.983594
1	K_PCA(RBF)+MLP	40	2.822581	97.177419	0.971737	0.971465	0.972295
2	LDA+MLP	1	0.000000	100.000000	1.000000	1.000000	1.000000
3	K_LDA(RBF)+MLP	1	0.000000	100.000000	1.000000	1.000000	1.000000
4	K_LDA_(SIGMOID)+MLP	1	0.000000	100.000000	1.000000	1.000000	1.000000
5	K_PCA(POLY)+MLP	40	4.032258	95.967742	0.957671	0.953243	0.963388

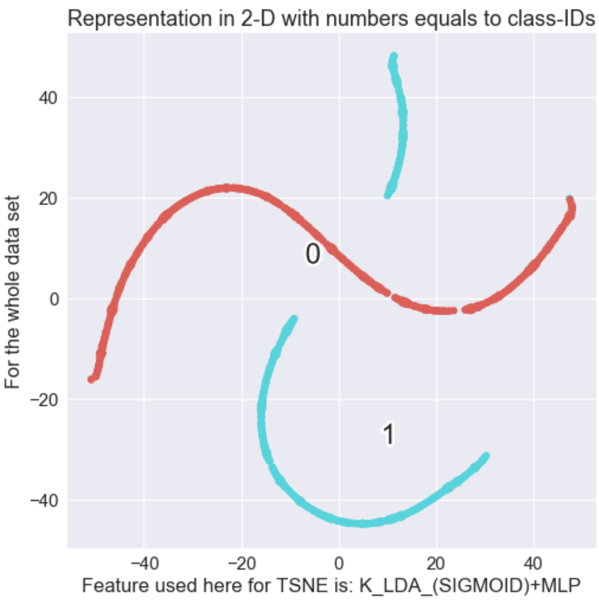
tSNE plots for whole data set 0-> Cartoon 1-> Real

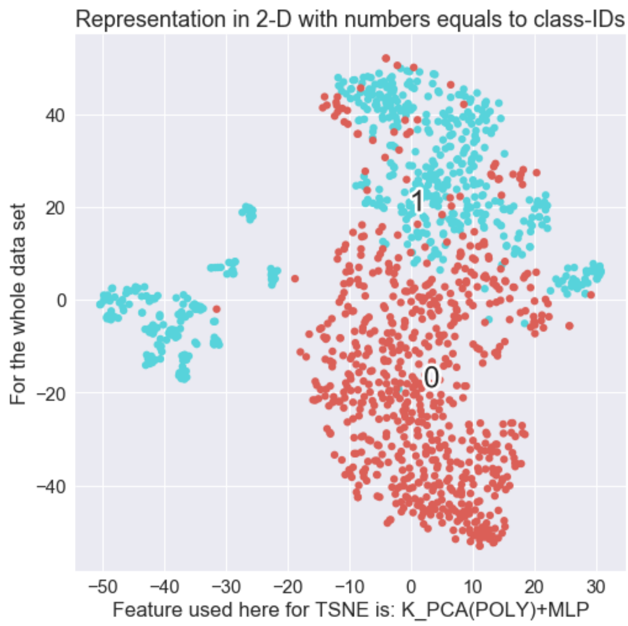








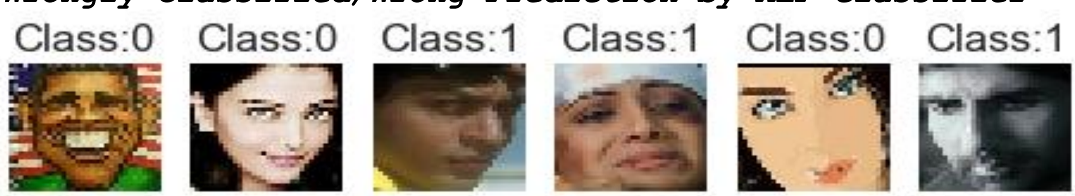




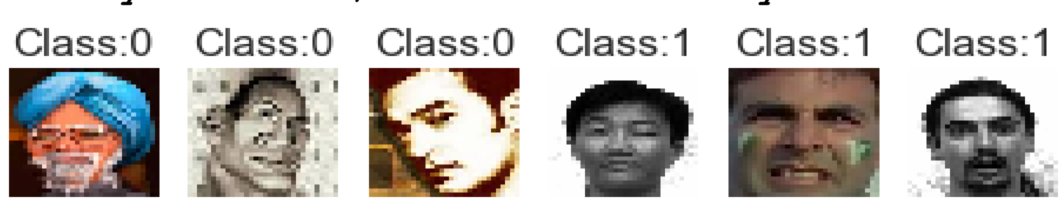
- We see that accuracy for LDA and KLDA is good because the data is separated nicely and we can see from tSNE plots also, the data is clearly separated.

QUANTITATIVE RESULTS

Wrongly Classified/Wrong Prediction by MLP classifier



Correctly Classified/Correct Prediction by MLP classifier



4 c) Empirical results with all the representations and variations in K.(For 3 data sets)