Dimensionality Reduction And Classification On Custom Handwritten Character Dataset

Aaryan Kumar
Department of ECE
Gainesville, USA
aaryan.kumar@ufl.edu

***Abstract*—The purpose of this report is to observe, evaluate and explain different machine learning methodologies used for the purpose of dimensionality reduction tasks on the provided handwritten symbols dataset. For the experiment, I have used Logistic Regression classifier, and SVM classifier for classification tasks. For the dimensionality reduction tasks, I have used Recursive Feature Elimination (RFE), PCA, LDA, MDS, LLE, ISOMAP, and t-SNE methodologies. This report contains implementation of said methodologies for different tasks. The report further contains observation, evaluation, and analysis of the results.**

Keywords—machine learning, dimensionality reduction, logistic regression, SVM.

I. Introduction

Dimensionality reduction is a very important task while building machine learning models. It is done to prevent the curse of dimensionality, which refers to the various phenomena that arise when analyzing and organizing data in high-dimensional space. As the number of features increase, our data become sparser, which results in overfitting, and we therefore need more data to avoid it. Over the years many different methodologies have been developed to do the task of dimensionality reduction. In this project I used these methodologies to reduce the dimensions of the given handwritten symbol dataset for the purpose of classification. The training dataset provided has a total of 9000 samples, each ample having 6720 features (or dimensions). There are total of 10 classes in dataset, having almost equal proportions. The distribution of samples for each class is as shown in the Figure 1

Figure 1

Each sample is an image of size 300x300 pixels. The example of each class is shown in Figure 2. It is to be noted that the images need be downsized to make is computationally less expensive for the machine learning models to process them.

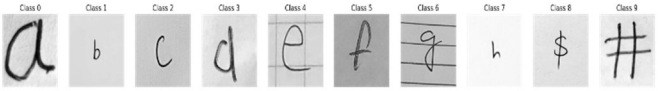

Figure 2

Different dimensionality reduction techniques were used based on the task given. This report discusses all such tasks and evaluates the results.

II. Data Preprocessing

The images in the dataset are resized from 300x300 pixels to 50x50 pixels. Doing this makes it easier for the models to compute, without losing much of the information. The resized images are shown in Figure3.

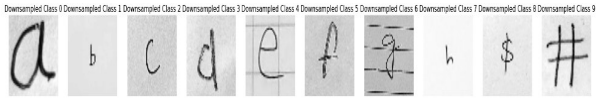

Figure 3

There is no other further need of doing any kind of preprocessing on the dataset.

III. Machine Learning Model Training

For the project, there are a total of 4 different machine learning tasks, each of which is described below-

A. Implementation of Recursive Feature Elimination (RFE)

The first task of the project was to implement Recursive Feature Elimination (RFE) algorithm to select the subset of features from the dataset. The RFE was implemented with two different estimators, i.e. Logistic Regression classifier, and State Vector Machine (SVM).

Two different pipelines were created for each estimator method. Logistic regression classifier was created with Lasso regularization, and 'liblinear' solver. The SVM classifier was created using LinearSVC method, taken directly from scikit learn examples. The RFE step in each pipeline was given a step size of 10 and was preceded by a data scaling step.

The models were trained on the training dataset, followed by being saved as pickle files.

B. *Implementation of Principle Component Analysis (PCA)*

The second task of the project required to implement PCA on the dataset to select the number of components that explain at least 90% of the explained variance and create a reduced dataset. I selected SVM to train the classifier on the original resized dataset, and the reduced dataset.

The SVM classifier was created using 'rbf' kernel. The pipeline created had 2 steps, the first being data scaling, and the second being the SVM classifier. The two models were trained and hyperparameter tuning was done using the GridSearchCV method on the original dataset, and the reduced dataset. The best performing models were then saved as pickle files.

C. *LDA; t-SNE; and PCA Visualization*

In the third task of the project, three different pipelines were created for the purpose of reducing the dataset to 2-dimensional data and visualize it. The first pipeline used Linear Discriminant Analysis to reduced data to 2-D. The second pipeline used t-SNE to reduce the data to 2-D. The third pipeline used PCA to reduce the data to 2-D. All the pipelines have a scaling function as the first step. The models were then trained on the resized dataset.

The trained models were then saved as pickle files.

D. *Manifold Learning Algorithm Implementation*

The fourth task of the project required to implement 3 manifold learning algorithms for dimensionality reduction and build a classifier using the said algorithms.

I selected MDS, ISOMAP, and LLE as the manifold learning algorithms and SVM with 'rbf' kernel as the classifier. Three pipelines were created for each algorithm with data scaling as the first step. The pipeline then fed the lower-dimensional feature space to the SVM classifier. The hyperparameter tuning was done for the number of components required for the algorithms. The best performing trained models were selected.

The trained models were then saved as pickle files.

IV. Machine Learning Model Testing Results And Evaluation

All the saved machine learning models were tested and evaluated and following was observed

A. *Implementation of Recursive Feature Elimination (RFE)*

The saved models for logistic regression with RFE, and SVM with RFE were loaded into the test notebook and were evaluated on following parameters:

1. Performance

After loading the saved models for logistic regression classifier, and SVM classifier, they were used to predict on the test dataset. The classification report for Logistic Regression classifier and SVM classifier are shown in Figure 4 and Figure 5 respectively.

	precision	recall	f1-score	support
0.0	0.40	0.43	0.41	274
1.0	0.33	0.32	0.32	273
2.0	0.36	0.53	0.43	285
3.0	0.41	0.37	0.39	296
4.0	0.38	0.30	0.33	309
5.0	0.38	0.31	0.34	296
6.0	0.37	0.40	0.38	291
7.0	0.35	0.40	0.37	280
8.0	0.41	0.39	0.40	291
9.0	0.44	0.37	0.40	285
accuracy			0.38	2880
macro avg	0.38	0.38	0.38	2880
weighted avg	0.38	0.38	0.38	2880

Figure 4

	precision	recall	f1-score	support
0.0	0.41	0.41	0.41	274
1.0	0.35	0.25	0.29	273
2.0	0.33	0.62	0.43	285
3.0	0.42	0.37	0.39	296
4.0	0.45	0.27	0.34	309
5.0	0.39	0.26	0.32	296
6.0	0.44	0.40	0.42	291
7.0	0.34	0.40	0.37	280
8.0	0.38	0.49	0.43	291
9.0	0.45	0.41	0.43	285
accuracy			0.39	2880
macro avg	0.40	0.39	0.38	2880
weighted avg	0.40	0.39	0.38	2880

Figure 5

It can be observed that SVM performed marginally better than the logistic regression classifier.

2. Selected Features and Mask examples

Heatmaps were generated for Logistic Regression classifier (Figure 6) and SVM classifier (Figure 7) to show which features were selected. (note- yellow shows the selected features.) Furthermore, mask examples were generated for both classifiers, and are shown in Figure 8 and Figure 9 respectively.

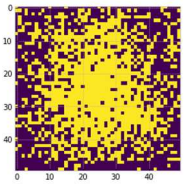

Figure 6

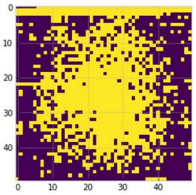

Figure 7

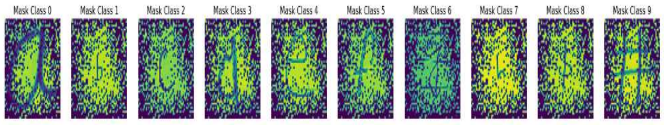

Figure 8

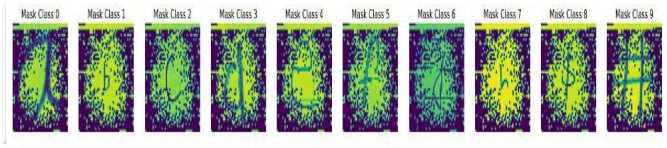

Figure 9

B. *Implementation of Principle Component Analysis (PCA)*

The two saved models were loaded and tested on the test dataset. The following was observed and evaluated:

1. Training time and Complexity

   Training was faster using reduced data set. It only took 134.329 seconds to train on reduced data set whereas it took 1886.6072 seconds to train on the original data set. Furthermore, the computational cost for training on reduced data set was less as there were only 146 components.

2. Performance Evaluation

   Target label for the test data was generated using both models and their classification reports were printed. Figure 10 shows the classification report for the model trained on original data set and figure 11 shows the classification report for the model trained on reduced data set.

	precision	recall	f1-score	support
0.0	0.38	0.49	0.43	274
1.0	0.38	0.41	0.39	273
2.0	0.34	0.67	0.45	285
3.0	0.52	0.45	0.48	296
4.0	0.54	0.33	0.41	309
5.0	0.40	0.39	0.39	296
6.0	0.56	0.47	0.51	291
7.0	0.40	0.39	0.39	280
8.0	0.59	0.38	0.46	291
9.0	0.59	0.46	0.52	285
accuracy			0.44	2880
macro avg	0.47	0.44	0.44	2880
weighted avg	0.47	0.44	0.44	2880

Figure 10

	precision	recall	f1-score	support
0.0	0.41	0.49	0.44	274
1.0	0.39	0.44	0.42	273
2.0	0.34	0.65	0.45	285
3.0	0.53	0.48	0.50	296
4.0	0.56	0.35	0.43	309
5.0	0.42	0.41	0.41	296
6.0	0.56	0.50	0.53	291
7.0	0.39	0.38	0.38	280
8.0	0.59	0.42	0.49	291
9.0	0.62	0.48	0.54	285
accuracy			0.46	2880
macro avg	0.48	0.46	0.46	2880
weighted avg	0.48	0.46	0.46	2880

Figure 11

It can be observed that the performance of model trained on reduced dataset is better than that of the model trained on original dataset.

3. Visualizing top 10 Eigenvectors

   The top 10 eigenvectors were visualized (Figure 12), and it was observed that the eigenvectors represent combination of features relating the original classes. It can be seen that there are outlines that represent boundaries of the classes.

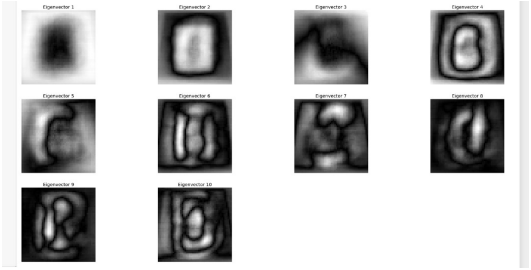

Figure 12

4. Visualizing examples of image reconstruction from PCA projection.

   The examples of image reconstruction for training data (Figure13) and for test data (Figure 14) were plotted as follows:

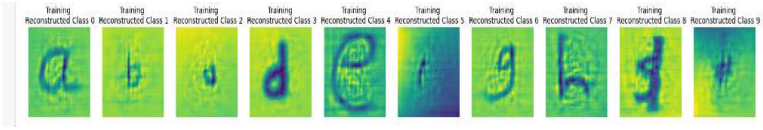

Figure 13

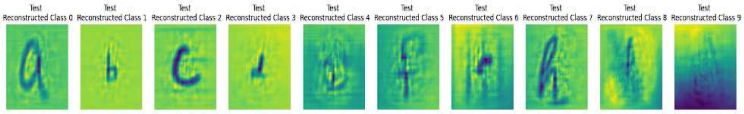

Figure 14

C. *LDA; t-SNE; and PCA Visualization*

The saved models were loaded and implemented on the test and training dataset. The plots for LDA visualization are shown in Figure 15,16. The plots for t-SNE visualization are shown in Figure 17,18 The plots for PCA visualization are shown in Figure 19,20.

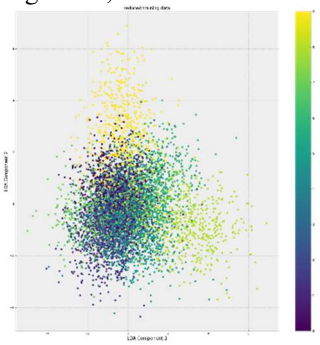

Figure 15

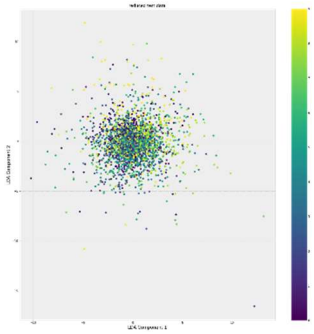

Figure 16

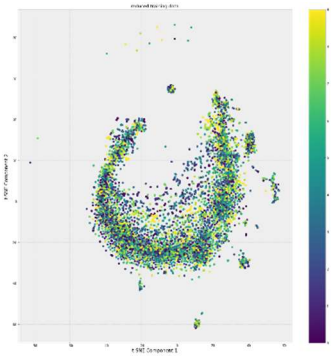

Figure 17

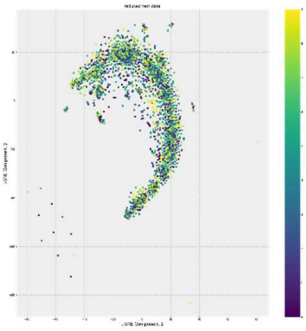

Figure 18

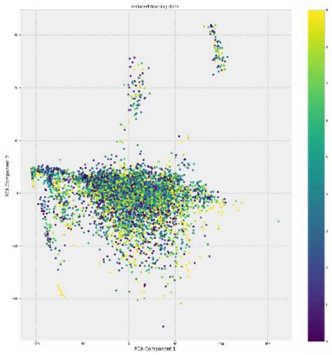

Figure 19

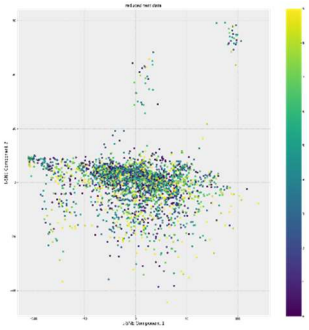

Figure 20

After comparing projections of LDA and t-SNE with the projections of PCA, it is clear that LDA should be preferred over the other two because by observing the projections, it can be concluded that each class is separable.

D. Manifold Learning Algorithm Implementation

The saved models were loaded and following analysis was done:

1. Performance Evaluation

   Each model was used to predict test labels using the test datasets. The classification reports for the classifier using MDS, ISOMAP, and LLE are shown in Figure 21,22,23 respectively.

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	274
1.0	0.00	0.00	0.00	273
2.0	0.00	0.00	0.00	285
3.0	0.00	0.00	0.00	296
4.0	0.11	1.00	0.19	309
5.0	0.00	0.00	0.00	296
6.0	0.00	0.00	0.00	291
7.0	0.00	0.00	0.00	280
8.0	0.00	0.00	0.00	291
9.0	0.00	0.00	0.00	285
accuracy			0.11	2880
macro avg	0.01	0.10	0.02	2880
weighted avg	0.01	0.11	0.02	2880

Figure 21

	precision	recall	f1-score	support
0.0	0.21	0.30	0.25	274
1.0	0.20	0.30	0.24	273
2.0	0.23	0.33	0.27	285
3.0	0.19	0.17	0.18	296
4.0	0.22	0.12	0.15	309
5.0	0.21	0.21	0.21	296
6.0	0.22	0.23	0.23	291
7.0	0.27	0.15	0.19	280
8.0	0.36	0.19	0.25	291
9.0	0.24	0.27	0.25	285
accuracy			0.23	2880
macro avg	0.23	0.23	0.22	2880
weighted avg	0.23	0.23	0.22	2880

Figure 22

	precision	recall	f1-score	support
0.0	0.24	0.30	0.27	274
1.0	0.24	0.25	0.25	273
2.0	0.27	0.29	0.28	285
3.0	0.18	0.22	0.20	296
4.0	0.27	0.22	0.24	309
5.0	0.27	0.19	0.22	296
6.0	0.21	0.26	0.23	291
7.0	0.28	0.24	0.26	280
8.0	0.26	0.27	0.27	291
9.0	0.28	0.24	0.26	285
accuracy			0.25	2880
macro avg	0.25	0.25	0.25	2880
weighted avg	0.25	0.25	0.25	2880

Figure 23

It can be concluded that LLE manifold learning has the best performance, and the highest accuracy. MDS performed the worst and took the longest to train.

2. Visualize and interpret what the first 2 dimensions in the trained manifold learning algorithm

   Following plots (Figure24) were obtained from the first 2 dimensions of manifold learning algorithms:

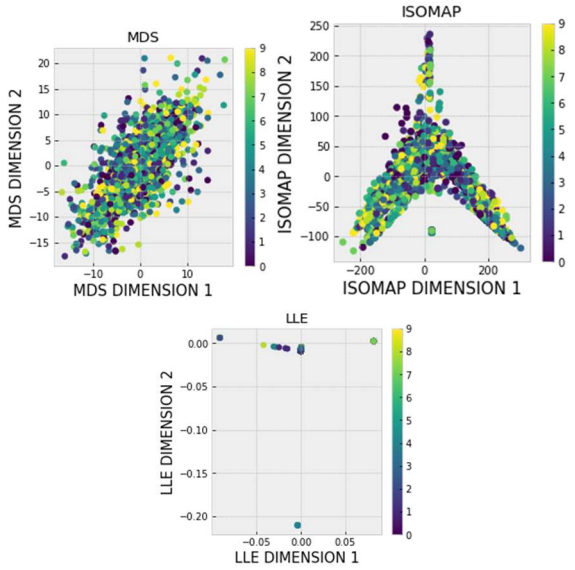

Figure 24

It can be clearly observed how different manifold learning algorithms separate the classes based on the shape, edges, thickness of the characters. MDS has separated classes based on shape and tilt, ISOMAP has separated classes based on orientation and brightness. LLE has separated classes based on edges and orientation.

CONCLUSION

The different machine learning models were trained, tested, and evaluated using different dimensionality reduction methods on the custom handwritten character dataset data for the tasks of classification.

ACKNOWLEDGMENT

I would like to acknowledge my professor, Dr. Catia Silvia, and course TA Jackson Cornell for helping and providing valuable insight for completion of this project.

REFERENCES

[1] EEL 5934 Course Lectures, office hours and Course notes

[2] EEL 5934 discussions on slack, and during lectures/office hours

[3] Scikit-learn documentation (https://scikit-learn.org/stable/index.htm