Gulfam Hussain
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
gulfamhu@buffalo.edu

Machine Learning – Project 3
Unsupervised Machine Learning Techniques

Abstract

As part of this projects, the classification is to be perform cluster analysis on fashion MNIST dataset using unsupervised learning. Cluster analysis is one of the unsupervised machine learning technique which doesn't require labeled data. The following three tasks are being performed and the analysis have been recorded in this project report:

1. Use KMeans algorithm to cluster original data space of Fashion-MNIST dataset using Sklearns library.
2. Build an Auto-Encoder based K-Means clustering model to cluster the condensed representation of the unlabeled fashion MNIST dataset using Keras and Sklearns library.
3. Build an Auto-Encoder based Gaussian Mixture Model clustering model to cluster the condensed representation of the unlabeled fashion MNIST dataset using Keras and Sklearns library.

Introduction:

Unsupervised learning is a type of self-organized learning that helps find previously unknown patterns in data set without pre-existing labels. The only requirement to be called an unsupervised learning method is to learn a new feature space that captures the characteristics of the original space while maximizing some objective function. It is also known as self-organization and allows modeling probability densities of given inputs.

Two of the main methods used in unsupervised learning are principal component and cluster analysis. Cluster analysis is used in unsupervised learning to group, or segment, datasets with shared attributes in order to extrapolate algorithmic relationships. Cluster analysis is a branch of machine learning that groups the data that has not been labelled, classified or categorized. Instead of responding to feedback, cluster analysis identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data.

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).

We are going to use to clustering techniques KMeans and Gaussian Mixture Model (GMM) in this project to perform the clustering on the given dataset.

Also, an autoencoder would be used to train the model before applying these clustering techniques.

Dataset:

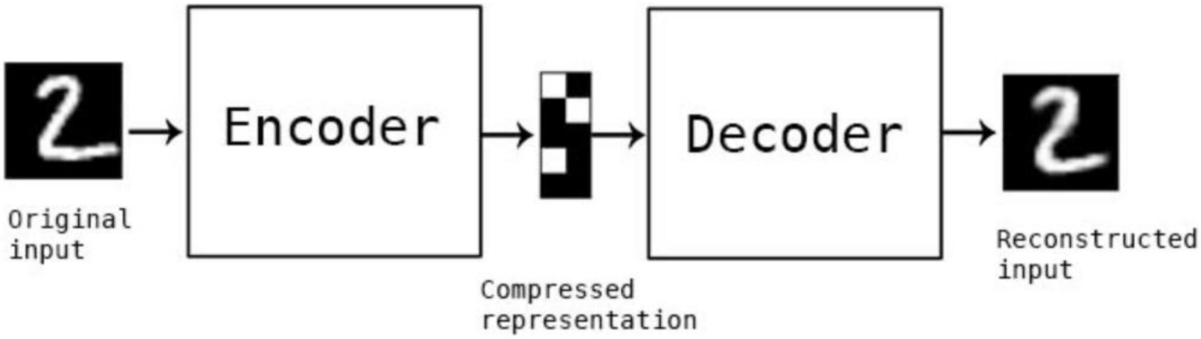
For training and testing of our clustering models, we will use the Fashion-MNIST dataset. The Fashion-MNIST is a dataset of Zalando's article images, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 784 columns. You can import the Fashion MNIST dataset using keras library. Each training and test example are assigned to one of the labels.
This Fashion MNSIT dataset has been provided as part of this project which can be downloaded through keras.

Preprocessing:

The given dataset has been preprocessed before using into the project. Two sets training and test dataset have been divided and the same will be used for the implementation of unsupervised clustering techniques.

Architecture:

Clustering technique using Auto Encoder:

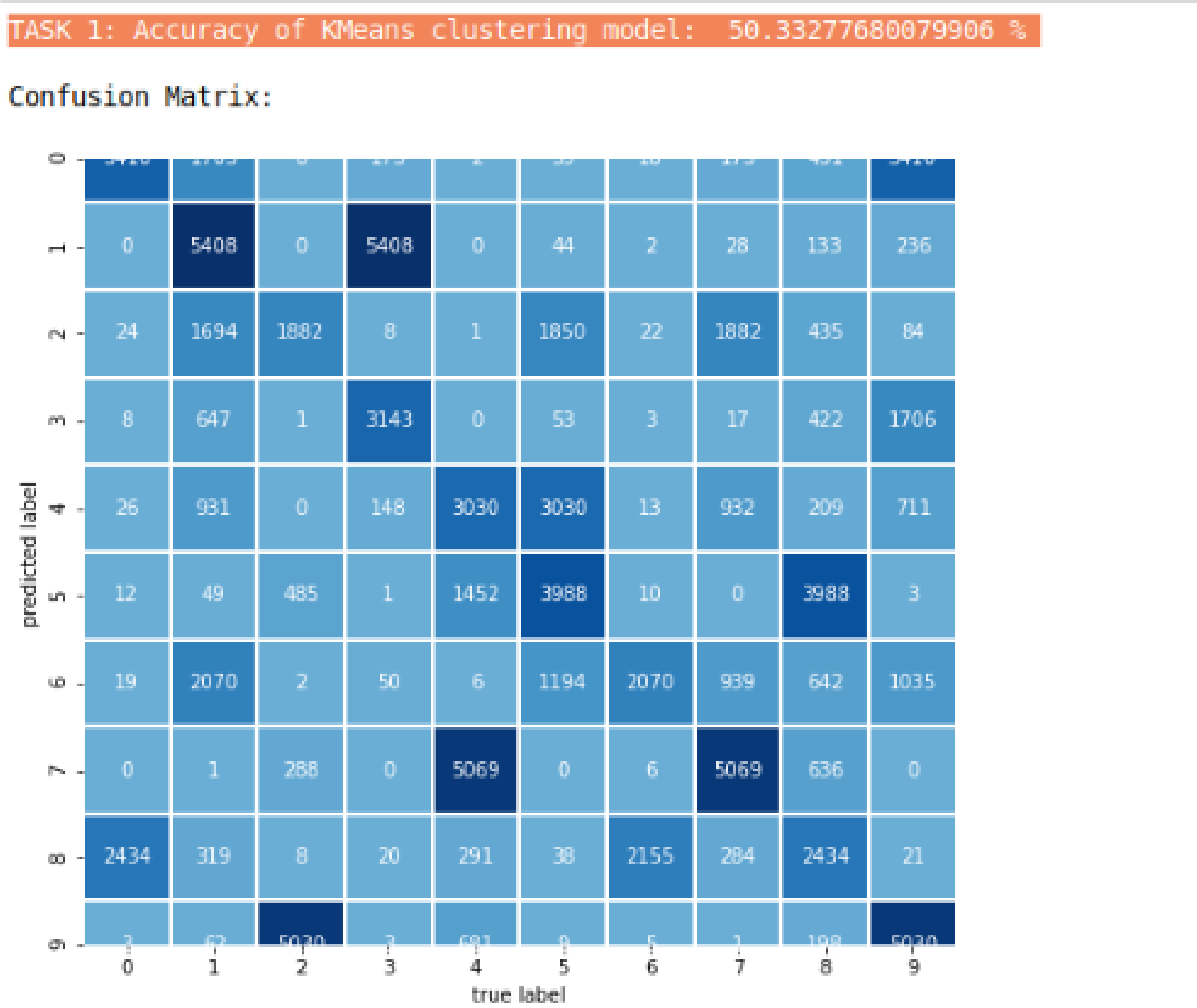


Results:

Task 1:

The baseline KMeans model using the Sklearns was implemented on the Fashion MNIST data and an overall accuracy of ~50% was achieved.

Clustering accuracy with K-Means as the baseline model and confusion matrix

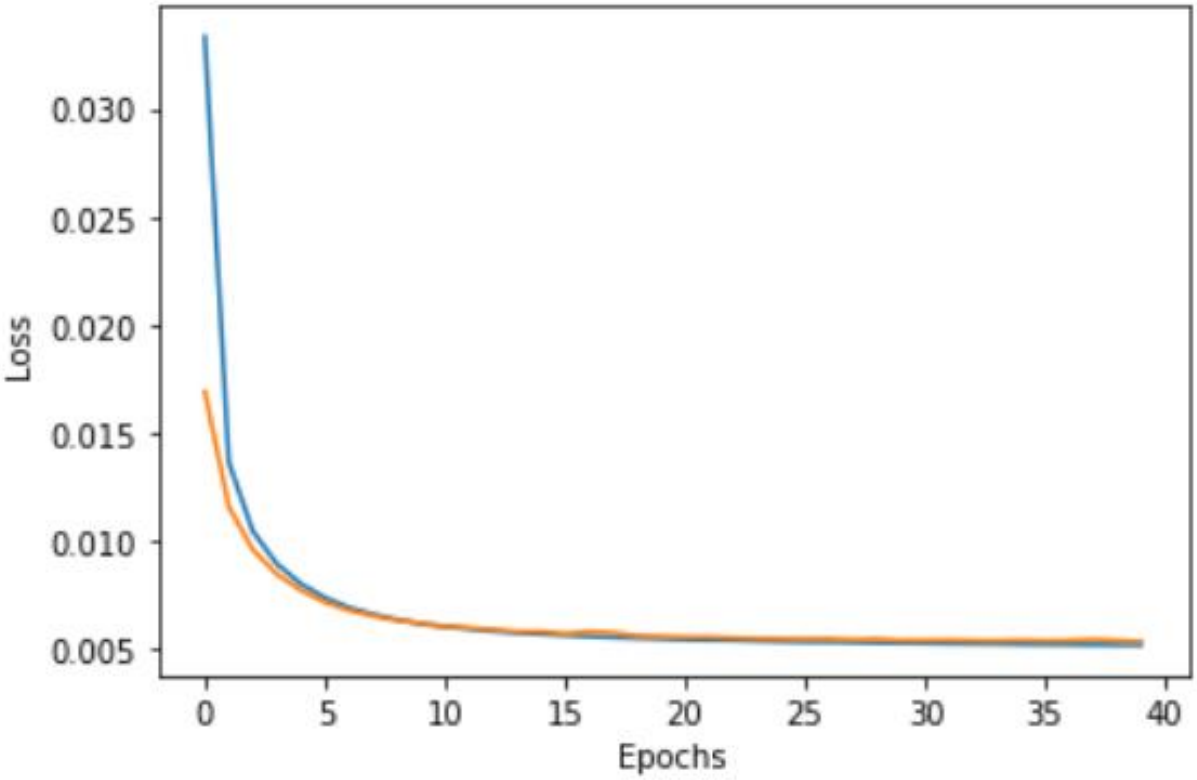


```
TASK 1: Accuracy of KMeans clustering model:  50.33277680079906 %
```

```
Confusion Matrix:
```

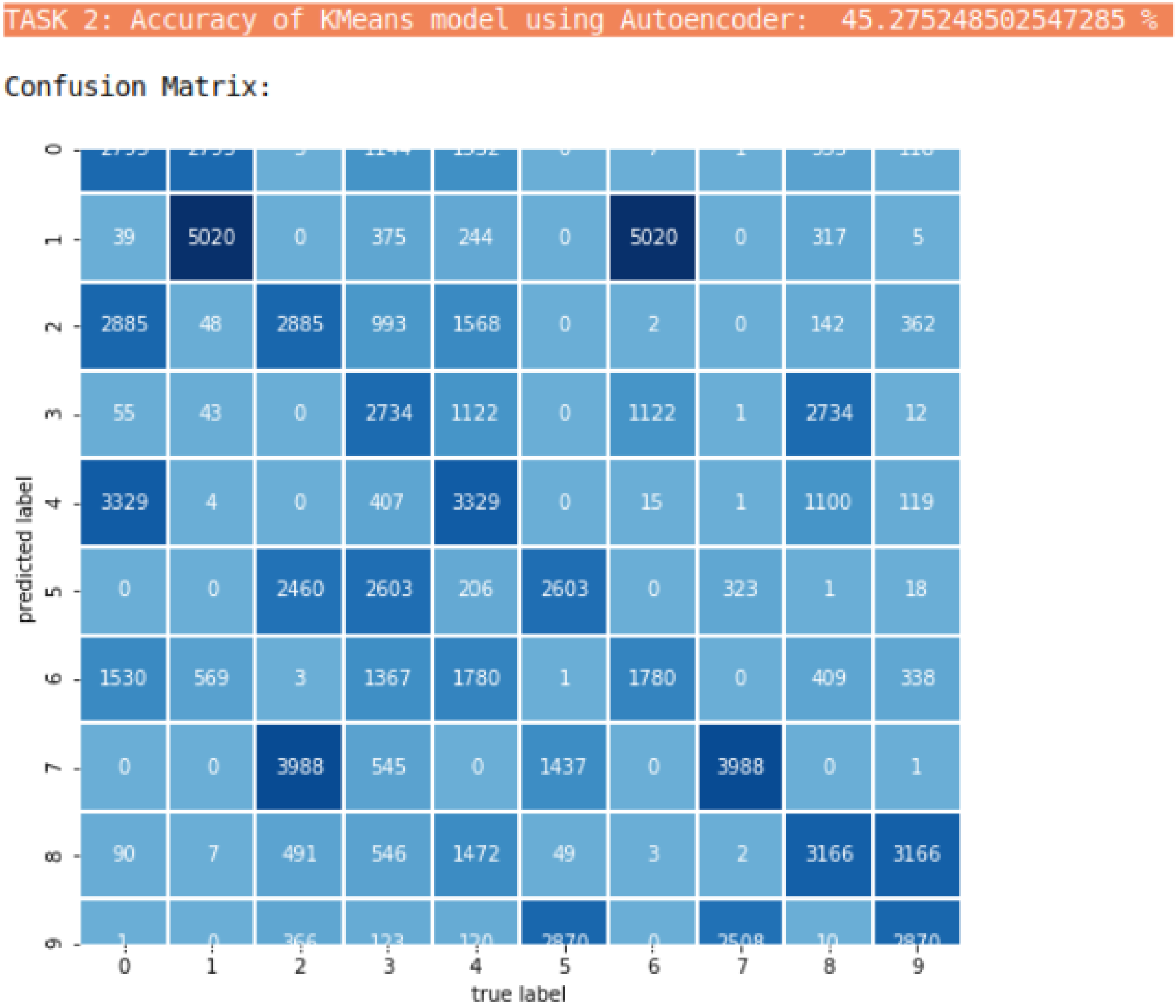
	0	1	2	3	4	5	6	7	8	9
0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	0	5408	0	5408	0	44	2	28	133	236
2	24	1694	1882	8	1	1850	22	1882	435	84
3	8	647	1	3143	0	53	3	17	422	1706
4	26	931	0	148	3030	3030	13	932	209	711
5	12	49	485	1	1452	3988	10	0	3988	3
6	19	2070	2	50	6	1194	2070	939	642	1035
7	0	1	288	0	5069	0	6	5069	636	0
8	2434	319	8	20	291	38	2155	284	2434	21
9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Graph of training loss and validation loss vs number of epochs while training for autoencoder



Task 2:

Confusion matrix for Auto-Encoder based K-Means clustering prediction and the clustering accuracy.



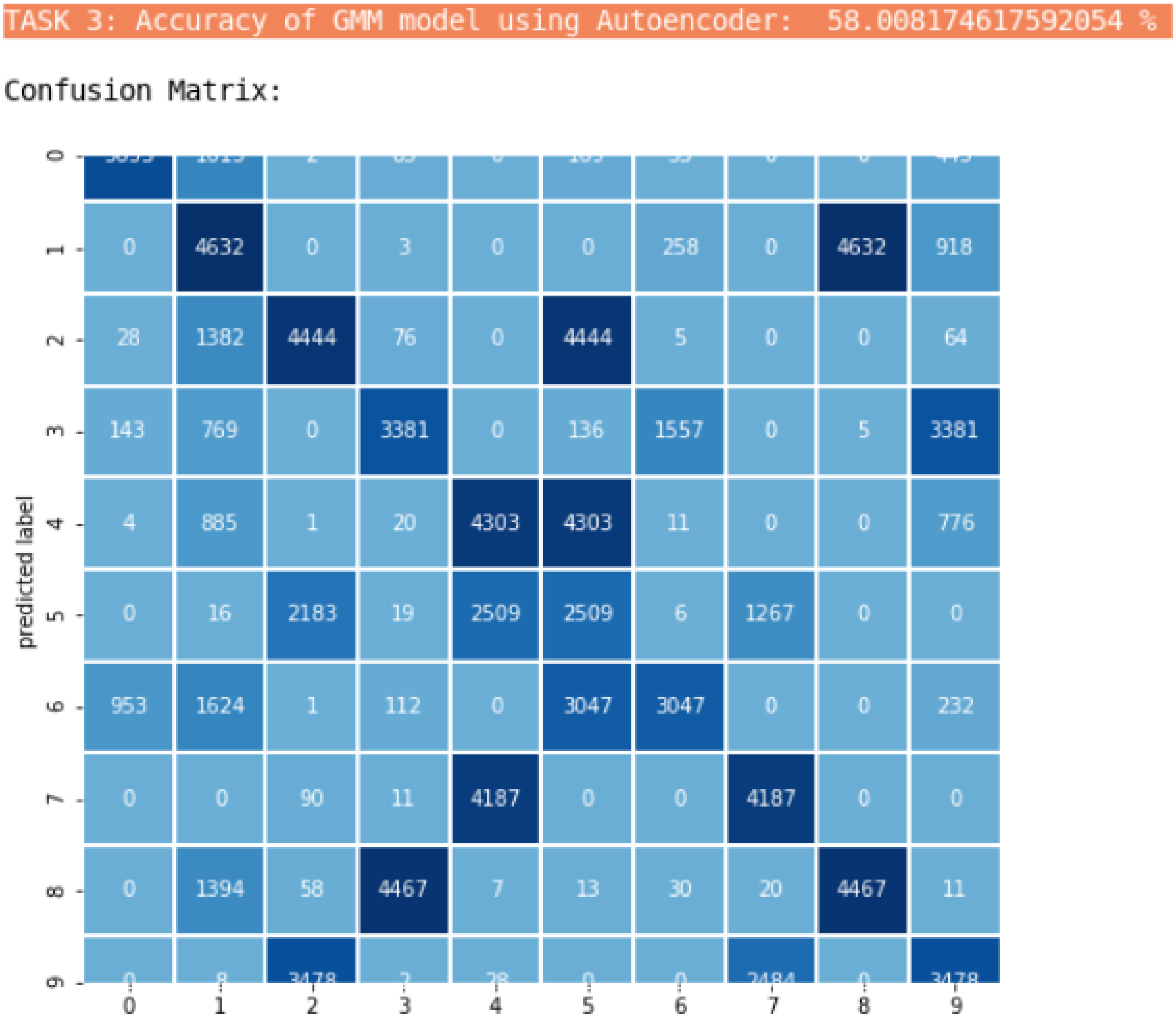
```
TASK 2: Accuracy of KMeans model using Autoencoder:  45.275248502547285 %
```

```
Confusion Matrix:
```

predicted label \ true label	0	1	2	3	4	5	6	7	8	9
0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	7	1	[illegible]	[illegible]
1	39	5020	0	375	244	0	5020	0	317	5
2	2885	48	2885	993	1568	0	2	0	142	362
3	55	43	0	2734	1122	0	1122	1	2734	12
4	3329	4	0	407	3329	0	15	1	1100	119
5	0	0	2460	2603	206	2603	0	323	1	18
6	1530	569	3	1367	1780	1	1780	0	409	338
7	0	0	3988	545	0	1437	0	3988	0	1
8	90	7	491	546	1472	49	3	2	3166	3166
9	1	0	366	123	120	2870	0	2508	10	2870

Task 3:

Confusion matrix for Auto-Encoder based GMM clustering prediction and the clustering accuracy.

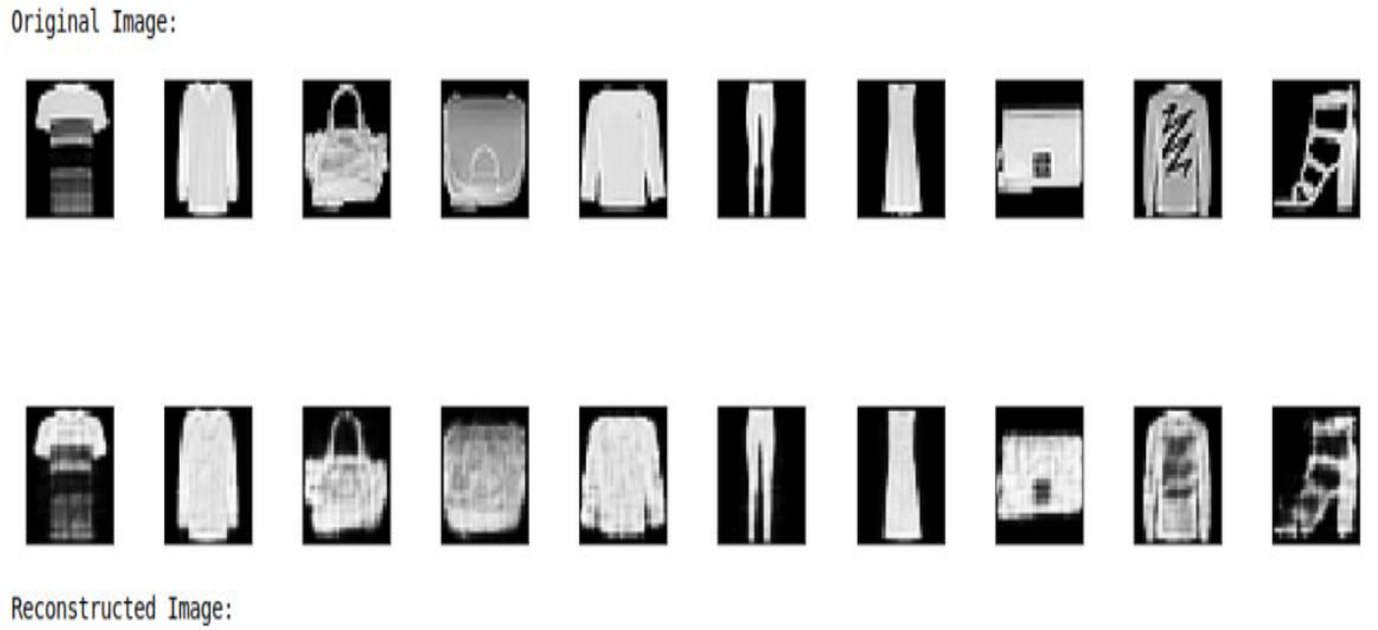


```
TASK 3: Accuracy of GMM model using Autoencoder:  58.008174617592054 %
```

```
Confusion Matrix:
```

	0	1	2	3	4	5	6	7	8	9
0	[illegible]	[illegible]	2	[illegible]	0	[illegible]	55	0	0	[illegible]
1	0	4632	0	3	0	0	258	0	4632	918
2	28	1382	4444	76	0	4444	5	0	0	64
3	143	769	0	3381	0	136	1557	0	5	3381
4	4	885	1	20	4303	4303	11	0	0	776
5	0	16	2183	19	2509	2509	6	1267	0	0
6	953	1624	1	112	0	3047	3047	0	0	232
7	0	0	90	11	4187	0	0	4187	0	0
8	0	1394	58	4467	7	13	30	20	4467	11
9	0	8	3478	2	28	0	0	2484	0	3478

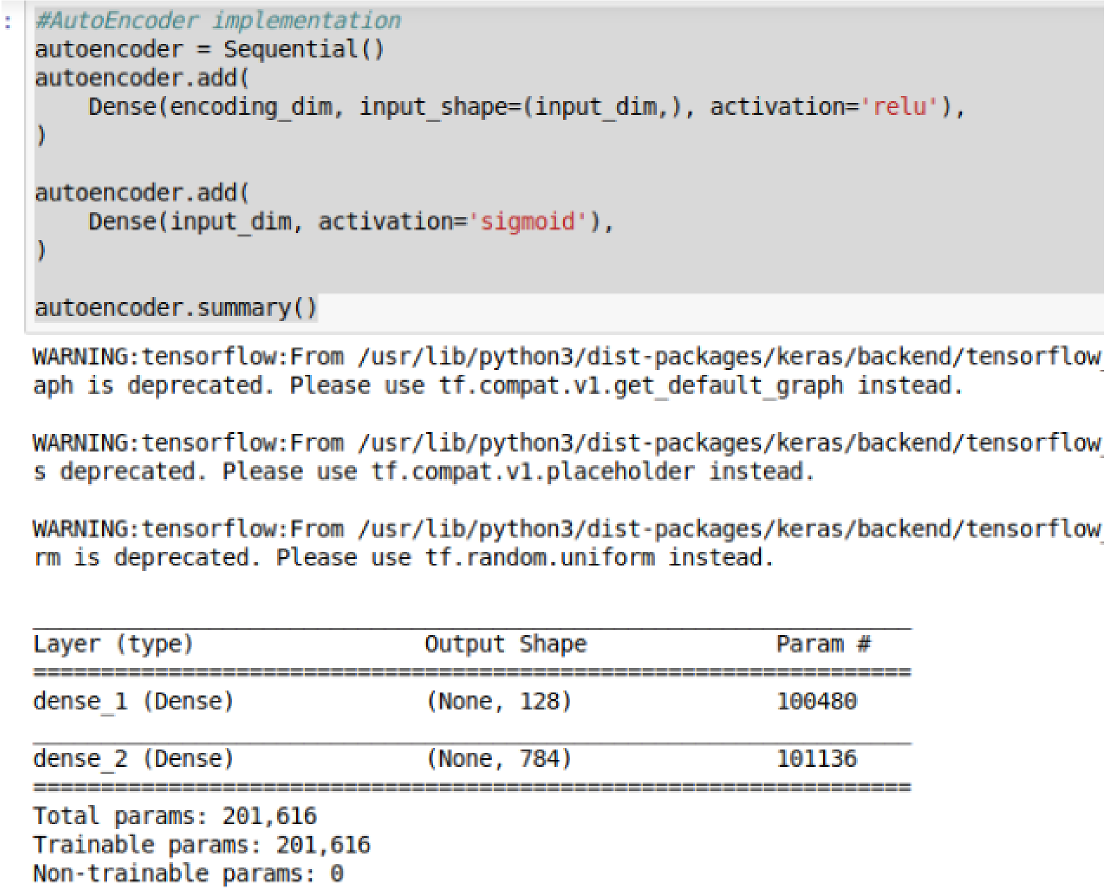
Original and reconstructed images:



```
Original Image:
```

```
Reconstructed Image:
```

Autoencoder summary:

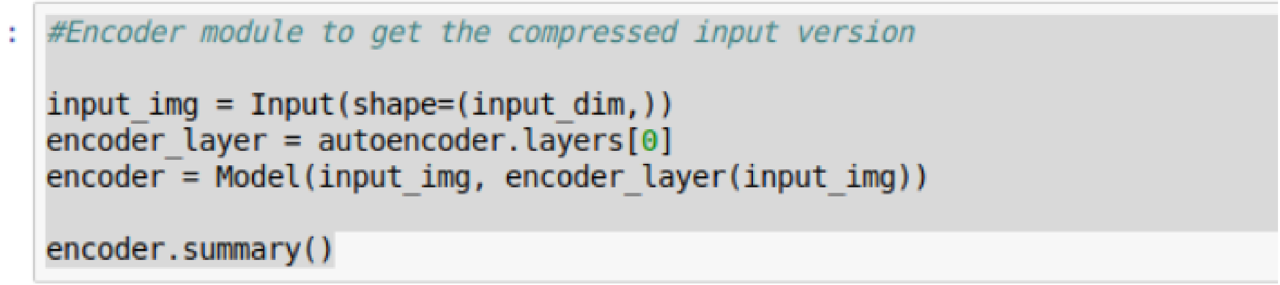


```
#AutoEncoder implementation
autoencoder = Sequential()
autoencoder.add(
    Dense(encoding_dim, input_shape=(input_dim,), activation='relu'),
)

autoencoder.add(
    Dense(input_dim, activation='sigmoid'),
)

autoencoder.summary()
```

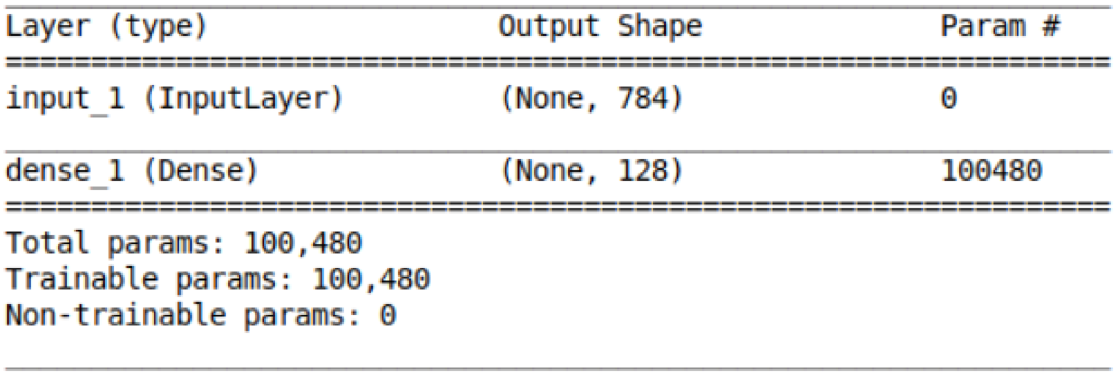
```
WARNING:tensorflow:From /usr/lib/python3/dist-packages/keras/backend/tensorflow_
aph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/lib/python3/dist-packages/keras/backend/tensorflow_
s deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/lib/python3/dist-packages/keras/backend/tensorflow_
rm is deprecated. Please use tf.random.uniform instead.

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               100480
_________________________________________________________________
dense_2 (Dense)              (None, 784)               101136
=================================================================
Total params: 201,616
Trainable params: 201,616
Non-trainable params: 0
```

Encoder module:

After training the autoencoder module, the weights of autoencoder was used to design an encoder module. The prediction of encoder module was then used in KMeans and GMM model to perform clustering and the corresponding accuracy scores were obtained.



```
#Encoder module to get the compressed input version

input_img = Input(shape=(input_dim,))
encoder_layer = autoencoder.layers[0]
encoder = Model(input_img, encoder_layer(input_img))

encoder.summary()
```



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 784)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               100480
=================================================================
Total params: 100,480
Trainable params: 100,480
Non-trainable params: 0
_________________________________________________________________
```

Conclusion:

This particular project based on unsupervised machine learning clustering techniques provided an opportunity to learn more on unsupervised learning approaches and their implementation in real world problems in the field of AI. The three models were successfully implemented to perform the operation on the Fashion MSNIT dataset and the accuracy results with confusion metrices were obtained for baseline KMeans model and KMeans and GMM models with autoencoder.

References:

[1] Class slides and project 3 description documents

[2] Google image

[3] Wikipedia

[4] https://www.dlology.com/blog/how-to-do-unsupervised-clustering-with-keras/