**N01419700**

**Machine Learning**

**Fashion MNIST dataset**

**Abstract**

**Motivation :** The Fashion MNIST dataset is a huge, freely accessible library of fashion photos that is often used to train and evaluate various machine learning techniques. Fashion-MNIST was designed to be a replacement for the original MNIST database for benchmarking machine learning algorithms, as it uses the same image size, data format, and training/testing split structure. The primary goal will be to create a classification model that can find the many fashion industry segments from the Fashion MNIST dataset using TensorFlow and Keras.

**For supervised algorithm** we classify the Fashion MNIST dataset, we will employ logistic regression and a 7-layer deep neural network. For each of these models, we would train the model using all 60000 examples of the data before testing it on the remaining 10000 instances. The accuracy will be our starting point, and we will strive to improve on it.

On the Fashion MNIST dataset, we use **unsupervised learning**, K-Means clustering, and Gaussian Mixture Model clustering to perform cluster analysis using the unsupervised technique. The main aim is to cluster photos and find them as belonging to several other clusters, as well as to perform cluster analysis on the fashion MNIST dataset using unsupervised learning. The machine learning algorithm is evaluated on the training set to decide the model's efficacy, and the performance is assessed based on the model's clustering precision.

**Results : For supervised algorithm,** we can see that the ideal distance is around the 30-50th percentile, whereupon the precision starts to drop (from both ends of the range). It should be noted that while we were unable to improve upon our baseline accuracy using semi-supervised learning techniques, if we were given completely unlabeled data, these techniques proved to be useful for increasing the accuracy of our models after we manual process labelled a small but massive part of the data.

**For unsupervised algorithm,** K-Means clustering, Auto-Encoder based K-Means clustering, and Auto-Encoder based GMM Clustering all reach a clustering accuracy of 52.4%, 51.7%, and 52.7%, respectively. Therefore, Gaussian Mixture Model is significantly more effective and efficient than the other models and has the highest accuracy.

**Introduction**

**Context :** Zalando article photos make up the Fashion-MNIST dataset, which has a training set of 60,000 samples and a test set of 10,000 examples. Each illustration is a 28x28 grayscale graphic paired with a label drawn from one of ten classes. Fashion-MNIST is intended to be a drop-in replacement for the original MNIST dataset for evaluating machine learning methods. The image size and split structure between training and testing are the same.

**Content** : Each image is 28 pixels tall and 28 pixels wide, for a total of 784 pixels. Each pixel has a single pixel-value associated with it, which indicates how light or dark that pixel is, with larger numbers suggesting darker. The integer for this pixel value ranges from 0 to 255. There are 785 columns in the training and test data sets. The first column is made up of class labels (as seen above) and represents an article of apparel. The corresponding image's pixel values are contained in the remaining columns.

**Labels :**

0 T-shirt/top

1 Trouser

2 Pullover

3 Dress

4 Coat

5 Sandal

6 Shirt

7 Sneaker

8 Bag

9 Ankle boot

**Objective :** The main objective is to use TensorFlow and Keras to build a classification model that can identify the various fashion industry segments from the Fashion MNIST dataset.

We will build a CNN model to recognize the image categories and train it on the dataset to achieve our goal. We chose deep learning since the dataset consists of photos, and CNNs have been the algorithm of choice for image classification problems. We will use Keras to build CNNs and TensorFlow to manipulate data.

**Supervised learning :** Although clustering is typically employed to solve issues relating to unsupervised learning, we will use it as a pre-processing technique for semi-supervised learning. If we just have a few labels, we could conduct clustering and propagate the labels to all instances (or the nearest instances determined by percentile) in the same cluster. By using this method, the number of labels that are available to a future supervised learning algorithm can be significantly increased, which will enhance the program's performance.

**Unsupervised learning :** The optimal distance, after which accuracy diminishes, is the 30th to 50th percentile (from both ends of the range). It should be noted that while we were unable to improve upon our baseline accuracy using semi-supervised learning techniques, if we were given completely unlabeled data, these techniques proved to be useful for increasing the accuracy of our models after we manual process labelled a small but massive portion of the data.

**Methodology**

**Supervised learning**

To classify the Fashion MNIST dataset, we will employ logistic regression and a 7-layer deep neural network. For each of these models, we would first train the model using all 60000 instances of the dataset, and then test it on the remaining 10000 examples. Our place to start will be precision, and we will work to increase it.

Graphical user interface, application

Description automatically generated

The Fashion MNIST dataset comprises images with a dimension of 28 x 28, thus we flatten the image matrix into a vector as a pre-processing step for logistic regression. We also pickled our models so they may be reused without retraining because the training period was rather long because we were also experimenting with different cluster number values. These files will be included with the code.

Graphical user interface, application

Description automatically generatedFirst, a pipeline that divides the training set into 100, 200, and 300 clusters, replaces the images with their distances from the clusters, and then applies a logistic regression model has been developed.

Then, we used arbitrary n-labeled examples (for n=500, 1000, or 2000) to test the accuracy of our models.

Graphical user interface, application

Description automatically generated

We can observe that the neural network does not perform very well in this situation when compared to Logistic Regression because the training set is so short. The neural network, however, requires far less time to train. Additionally, we observe that the models function better with bigger cluster sizes, indicating the need for larger cluster sizes in future experiments.

The examples are then divided into 2000 clusters, and the centroids are used to train our model.

Graphical user interface, application

Description automatically generated

We don't detect a substantial difference in the results since when we propagated the centroids' labels to all instances, we also included outliers or examples that are confused about which cluster they belong to. So let us propagate the labels to the instances that are close to the cluster centroids (25 percentile) and train our models again.

Graphical user interface, text, application

Description automatically generated

In order to obtain optimal accuracy, we attempt to determine for 2000 clusters the ideal distance from the cen

Graphical user interface, text, application

Description automatically generated

We can observe that the ideal distance is approximately the 30-50th percentile, after which the accuracy decreases (from both ends of the range). Although we were unable to improve on our baseline accuracy using semi-supervised learning techniques, in the case of entirely unlabeled data, these strategies come in handy for boosting the accuracy of our models after manually labelling a modest but significant fraction of the data.

**Unsupervised learning :**

**Part A – K-Means Clustering**

K-Means clustering is a basic and common unsupervised machine learning approach that is used with unlabeled data. A group of data points that have been combined due to some commonalities is referred to as a cluster. The spot that represents a cluster's core is called a centroid. Each cluster receives a certain set of data points. The algorithm's primary goal is to identify data groups, and it uses recursion to allocate data points to each cluster. Based on features that are similar, data points are clustered. In K- Means clustering, Euclidean Distance is utilized and may be determined using the following formula:

Graphical user interface, application, Word

Description automatically generated

Every data point is first assigned to the closest cluster using the K-Means algorithm, which also attempts to make the centroids as tiny as feasible. In data mining, the K-Means algorithm begins with a random selection of centroids and then executes iterative calculations to maximize the coordinates of the centroids. The centroids of the K-Clusters and the labels of the training data are the output of K-Means clustering. By reducing inertia or the within-cluster sum of squares criterion, the K-Means method selects centroids:

Graphical user interface, application, Word

Description automatically generated

One way to gauge how internally coherent a cluster is is by its inertia. There is no normalised metric for inertia.

The distance measure is prone to inflation. Running a dimensionality reduction approach, such as Principal Component Analysis (PCA) or Auto-encoder, prior to k-means clustering might ease the difficulty and speed up the computations. K-Means algorithms are extremely quick, reliable, and simpler to comprehend. When data sets are separate, it is quite effective and produces the finest results. But. Noisy data and outliers are intractable by K-Means. For non-linear data selection, it fails. Customer segmentation can be done using K-Means clustering. Document clustering, image segmentation, recommendation engines, and so on.

Graphical user interface, application, Word

Description automatically generated

**Part B – Auto-encoder training based on K-Means clustering**

One way to gauge how internally coherent a cluster is by its inertia. There is no normalized metric for inertia.

The distance measure is prone to inflation. Running a dimensionality reduction approach, such as Principal Component Analysis (PCA) or Auto-encoder, prior to k-means clustering might ease the difficulty and speed up the computations. K-Means algorithms are extremely quick, dependable, and simpler to comprehend. When data sets are separate, it is quite effective and produces the finest results. But. Noisy data and outliers are intractable by K-Means. For non-linear data selection, it fails. Customer segmentation can be done using K-Means clustering.

The sparsity parameter is :

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Part C - Gaussian Mixture Model with Auto-Encoder for Training**

Gaussian Mixture Models are probabilistic models that use a soft clustering strategy to distribute points into distinct clusters. The assumption made by Gaussian Mixture Models (GMM) is that there exist a certain number of Gaussian distributions, each of which represents a cluster. Therefore, a Gaussian Mixture Model tends to combine data points that belong to the same distribution. Gaussian Mixture Models employ the soft clustering technique to assign data points to Gaussian distributions.

A Gaussian mixture model (GMM) looks for a combination of multidimensional Gaussian probability distributions. A probabilistic model for modelling regularly distributed subpopulations within a larger population is the Gaussian mixture model.

The probability density function of a Gaussian Distribution in one dimension is given by: (“Gaussian Mixture Model - GeeksforGeeks”)Graphical user interface, text, application, Word

Description automatically generated

The formula for a Gaussian Distribution's probability density function in one dimension is :

**Graphical user interface, application, Word

Description automatically generated**

**Result**

The project's goal is to perform cluster analysis on the fashion MNIST dataset using unsupervised learning.

One unsupervised machine learning technique that does not require labelled data is cluster analysis. To cluster photos and find it as one of several clusters is the goal. Three processes were conducted during this project:

Using the Sklearns library and the K-Means technique, the Fashion - MNIST dataset's original data space was clustered.

The condensed form of the unlabeled fashion MNIST dataset is clustered using the Keras and Sklearns libraries, and the K-Means clustering algorithm is based on Auto-Encoder.

The condensed representation of the unlabeled fashion MINTST dataset is clustered using the Keras and Sklearns libraries, based on a Gaussian Mixture Technique clustering model.

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

The model is assessed by plotting a graph of training loss and validation vs. epoch count during auto-encoder training.

Graphical user interface, application, Word

Description automatically generated

**Weka**

Chart

Description automatically generated

**Visualizing dataset**

Chart, scatter chart

Description automatically generated

**Logistic regression**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with low confidence

Graphical user interface, chart

Description automatically generated with medium confidence

**Naïve bayes**

Graphical user interface, application

Description automatically generated

**Random Forest**

Graphical user interface

Description automatically generated

**REP Tree**

A picture containing scatter chart

Description automatically generated

**J48 Tree**

**Conclusion**

To use unsupervised learning to do cluster analysis on the Fashion MNIST dataset. Clustering techniques employed include Gaussian Mixture Model and K-Means. The main aim is to cluster photos and find them as belonging to one of many clusters, as well as to perform cluster analysis on the fashion MNTST dataset using unsupervised learning. The machine learning algorithm is evaluated on the testing set to decide the model's efficacy, and the performance is assessed based on the model's clustering precision. An Auto-Encoder based K-Means clustering model is developed to cluster the condensed representation of the unlabeled fashion MNTST dataset using Keras and Sklearns library, and an Auto-Encoder based Gaussian Mixture Model clustering model is built to cluster the condensed representation of the unlabeled fashion MNIST dataset using Keras and Sklearns library are the three tasks that were completed. K-Means clustering, Auto-Encoder based K-Means clustering, and Auto-Encoder based GMM Clustering all obtain a clustering accuracy of 52.4%, 51.7%, and 52.7%, respectively. As a result, the Gaussian Mixture Model is far more exact and effective than the other models.

**References**

*[1] https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aal*

*[2] https://towardsdatascience.com/applied-deep-leaning-part-3-autoencoders-1c083af4d798*

*[3]* [*https://www.researchgate.nel/figure/3D-view-of-a-40-Gaussian-Mixture-Model-used-in-our-experiments\_figl\_45872040*](https://www.researchgate.nel/figure/3D-view-of-a-40-Gaussian-Mixture-Model-used-in-our-experiments_figl_45872040)

*[4]*[*"Basic classification: Classify images of clothing | TensorFlow Core"*](https://www.tensorflow.org/tutorials/keras/classification)*. TensorFlow. Retrieved 2021-10-07.*

*[5]* [*^*](https://en.wikipedia.org/wiki/Fashion_MNIST#cite_ref-5)[*"Build software better, together"*](https://github.com/search?q=fashion-mnist)*. GitHub. Retrieved 2022-01-30.*

*[6]* [*^*](https://en.wikipedia.org/wiki/Fashion_MNIST#cite_ref-6)[*"Papers using Fashion-MNIST (till 09.18)"*](https://docs.google.com/spreadsheets/d/1cGX7Juedn_KVUgjDk298v5uUjc_wPk930tKyEoZhTQM/edit?usp=embed_facebook)*. Google Docs. Retrieved 2022-01-30.*

*[7]* [*^*](https://en.wikipedia.org/wiki/Fashion_MNIST#cite_ref-7)*Meshkini, Khatereh; Platos, Jan; Ghassemain, Hassan (2020). Kovalev, Sergey; Tarassov, Valery; Snasel, Vaclav; Sukhanov, Andrey (eds.).*[*"An Analysis of Convolutional Neural Network for Fashion Images Classification (Fashion-MNIST)"*](https://link.springer.com/chapter/10.1007/978-3-030-50097-9_10)*. (“An Analysis of Convolutional Neural Network for Fashion Images ...”) Proceedings of the Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). Advances in Intelligent Systems and Computing. Cham: Springer International Publishing. 1156: 85–95.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1007/978-3-030-50097-9\_10*](https://doi.org/10.1007%2F978-3-030-50097-9_10)*.*[*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-3-030-50097-9*](https://en.wikipedia.org/wiki/Special:BookSources/978-3-030-50097-9)*.*[*S2CID*](https://en.wikipedia.org/wiki/S2CID_(identifier))[*226778948*](https://api.semanticscholar.org/CorpusID:226778948)*.*

*[8]* [*^*](https://en.wikipedia.org/wiki/Fashion_MNIST#cite_ref-8)*Kayed, Mohammed; Anter, Ahmed; Mohamed, Hadeer (February 2020).*[*"Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture"*](https://ieeexplore.ieee.org/document/9047776)*. 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE): 238–243.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1109/ITCE48509.2020.9047776*](https://doi.org/10.1109%2FITCE48509.2020.9047776)*.*[*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-1-7281-4801-4*](https://en.wikipedia.org/wiki/Special:BookSources/978-1-7281-4801-4)*.*[*S2CID*](https://en.wikipedia.org/wiki/S2CID_(identifier))[*214691687*](https://api.semanticscholar.org/CorpusID:214691687)*.*