**Applicant Name:** Khushi Pitroda

## Introduction:

Generative Adversarial Networks (GANs) are a class of powerful machine learning models that can generate synthetic data that closely resembles real data. GANs consist of two neural networks: a generator and a discriminator. The generator creates fake data samples, while the discriminator tries to distinguish between real and fake data. Through an iterative training process, GANs learn to generate increasingly realistic and high-quality outputs. GANs have applications in image synthesis, text generation, and more, offering exciting possibilities for creative content generation and data augmentation.

## Technical Details:

**Name of the Dataset:**

* Fashion MNIST

## Description of the Dataset:

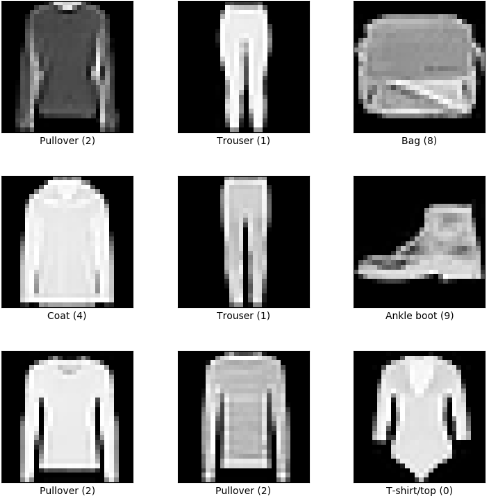


Fashion MNIST is a dataset provided by TensorFlow that serves as a replacement for the handwritten digit MNIST dataset. It consists of 60,000 grayscale images of fashion items, categorized into 10 classes such as T-shirts, trousers, dresses, and more. Each image is 28x28 pixels.

**The data set includes information about:**

* + Below are pixel values representing a 28x28 grayscale image from the Fashion MNIST dataset. Each value ranges from 0 to 255, where 0 represents black and 255 represents white:

## Sample values of the Dataset:



**Source of the Dataset:**

* **Tensorflow -** <https://www.tensorflow.org/datasets/catalog/fashion_mnist>

## Data Mining Functionality:

* **Classification**
  + Logistic Regression
  + Decision Tree

## Description of the Data Mining Functionality:

* **Classification:** Classification is a task in data mining that involves assigning a class label to each instance in a dataset based on its features. The goal of classification is to build a model that accurately predicts the class labels of new instances based on their features.
* **Data Augmentation:** GANs can be used to augment training data by generating synthetic samples that expand the dataset. This helps in improving model generalization and addressing issues related to limited or imbalanced data.
* **Pattern Mining:** Pattern mining involves discovering frequent patterns, sequences, or associations in structured or sequential data. It helps uncover hidden patterns and dependencies that can be used for decision- making or recommendation systems.

## Implementation Details: Implementation Environment:

* Python environment

## Platform used:

* Jupyter Notebook

## Software Required:

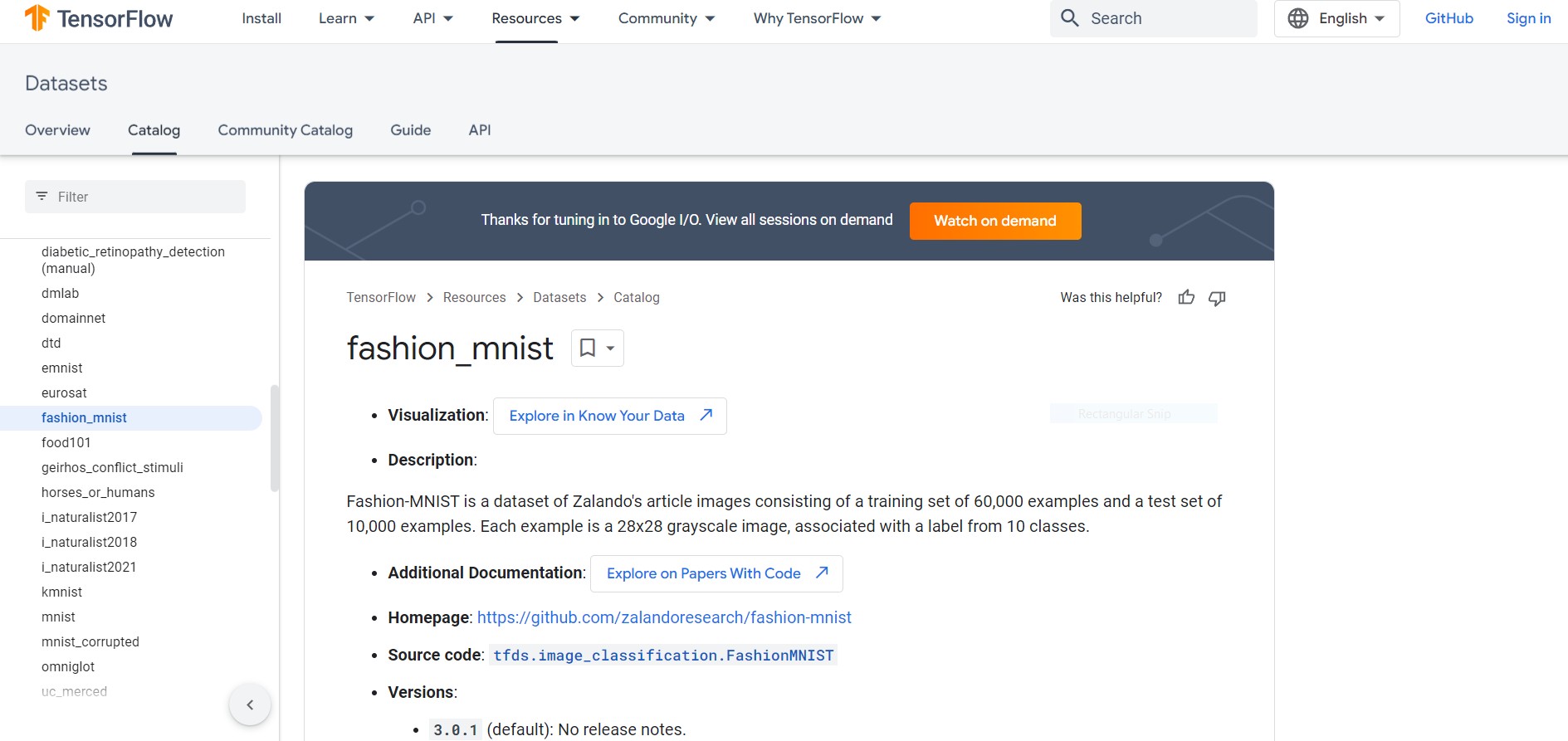
* Python
* Jupyter Notebook
* TensorFlow

## Hardware Required:

* Computer or server with a CPU
* RAM
* Storage
* GPU

## Screenshots of the implementation:

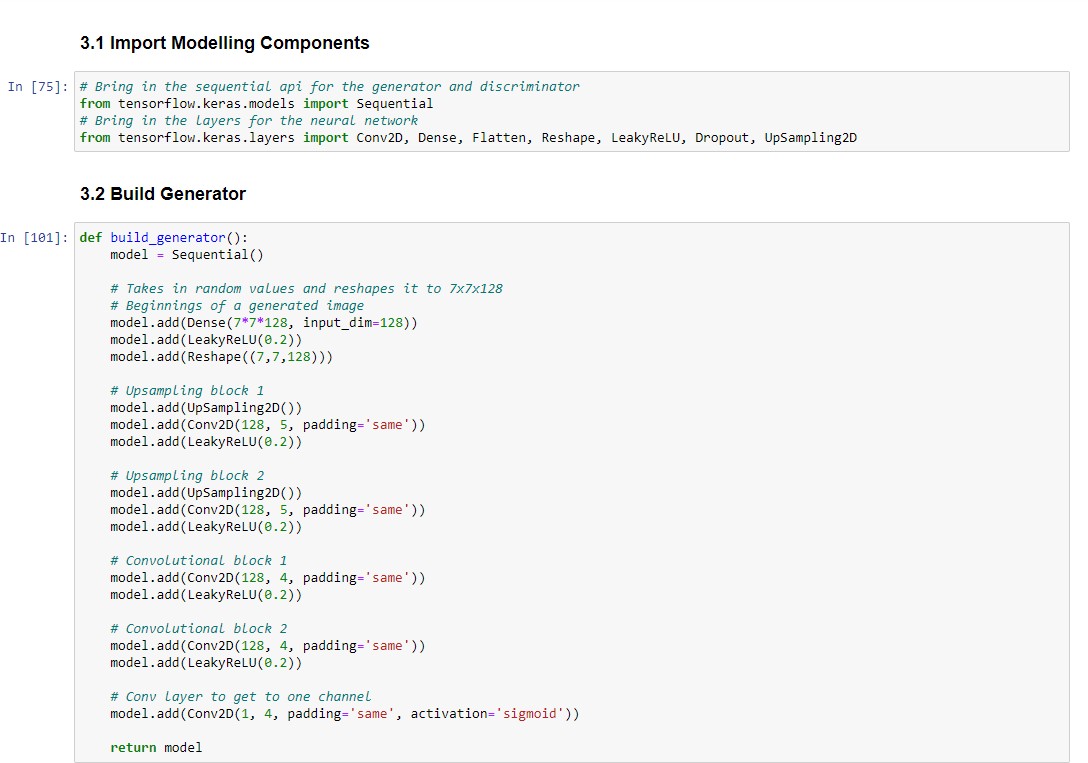
* **Generative Adversarial Networks (GANs):**
* **Importing Dataset:**



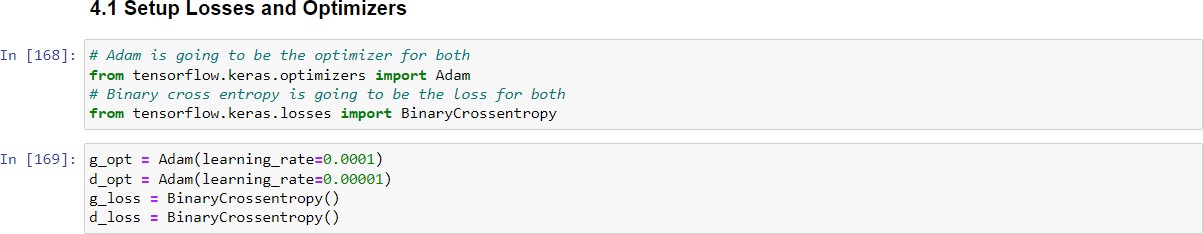
* Visualising Data



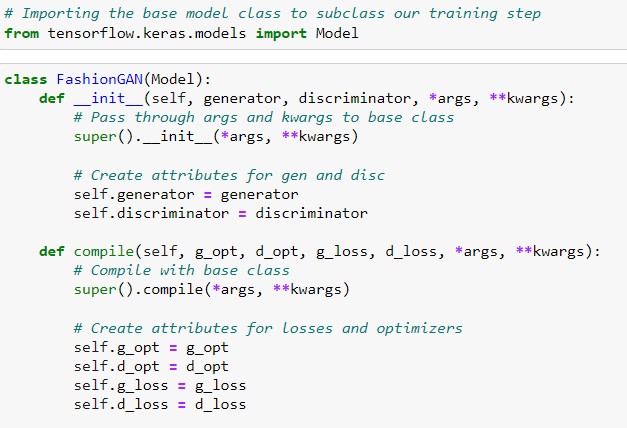
* Building the neural network for generator:

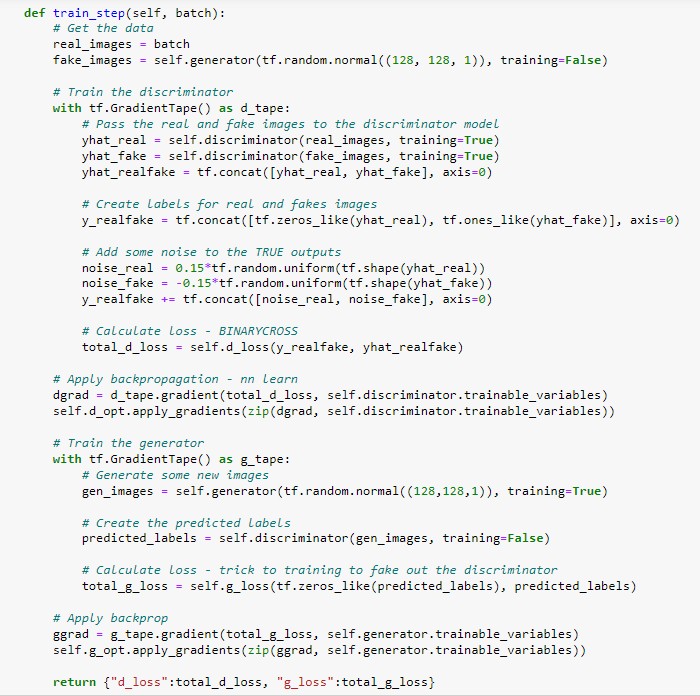


* Initialising optimisers and loss functions:

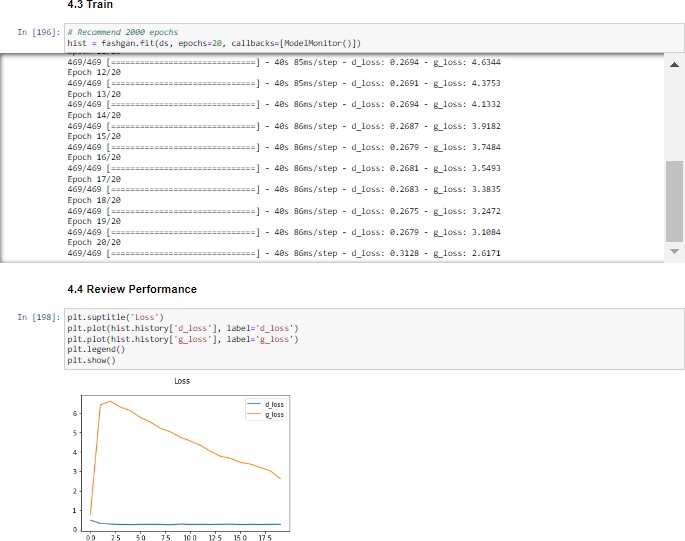


* Build Class Model:

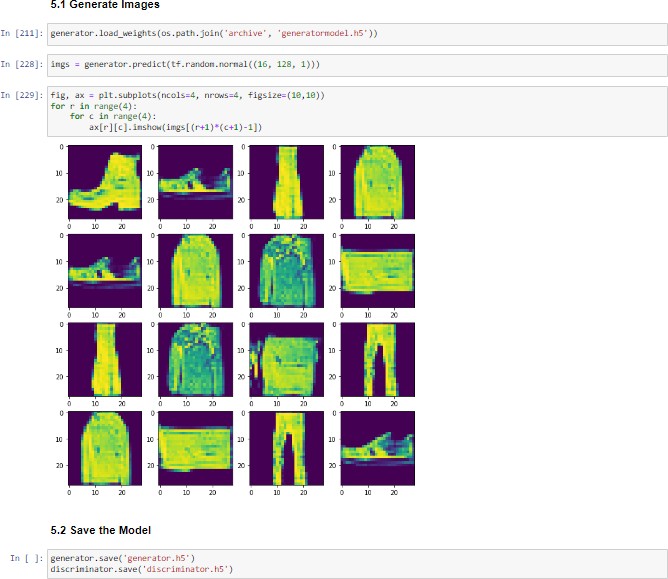




* Training the models:



* Generating Images:



## Results and Discussions:

* GANs have achieved impressive results in generative modeling by producing realistic and high-quality synthetic data. They have revolutionized various domains such as image synthesis, text generation, and video generation. GANs continue to push the boundaries of generative modeling and drive advancements in machine learning research.

## Conclusion:

* In conclusion, GANs have revolutionized generative modeling by enabling the generation of realistic synthetic data. They have applications in various domains and hold promise for advancements in virtual reality, data synthesis, and creative content generation. Despite challenges, GANs continue to push the boundaries of generative modeling and drive innovation in machine learning.

## Conclusion:

* **Logistic Regression Classification:**
  + From the confusion matrix, we can conclude that prediction of ‘no’ is 85.8% correct and prediction of ‘yes’ is 67% correct.
  + Overall accuracy of the model is 81.4%.

## Decision Tree Classification:

* + From the confusion matrix, we can conclude that prediction of ‘no’ is 82.6% correct and prediction of ‘yes’ is 64.1% correct.
  + Overall accuracy of the model is 79%.