**Review of probability and statistics concepts Task**

NAME : TEJASRI. V

**Probabilistic modeling :**

Probabilistic modeling is a statistical approach that uses the effect of random occurrences or actions to forecast the possibility of future results. It is a quantitative modeling method that projects several possible outcomes that might even go beyond what has happened recently.

Probabilistic modeling considers new situations and a wide range of uncertainty while not underestimating dangers. The three primary building blocks of probabilistic modeling are adequate probability distributions, correct use of input information for these distribution functions, and proper accounting for the linkages and interactions between variables. The downside of the probabilistic modeling technique is that it needs meticulous development, a process that depends on several assumptions and large input data.

**Importance of Probabilistic ML Models :**

One of the most significant advantages of the probabilistic modeling technique is that it provides a comprehensive understanding of the uncertainty associated with predictions. Using this method, we can quickly determine how confident any mobile learning model is and how accurate its prediction is. It is heavily dependent on the opposing concepts of uncertainty and confidence.

In reality, it is extremely helpful when used to key [machine learning applications](https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-applications) such as illness detection and autonomous driving. Furthermore, probabilistic outcomes would be beneficial for many Machine Learning-related approaches, such as Active Learning.

**1. Probability Distributions:**

Probability distributions play a central role in Generative AI models. These models aim to learn and replicate the probability distribution of the training data. By capturing the underlying statistical patterns, the model can generate new samples that resemble the original data.

**2. Likelihood Estimation:**

In Generative AI, models need to estimate the likelihood of generating observed data given their parameters. This involves finding the model parameters that maximize the likelihood of generating the training data. Maximum Likelihood Estimation (MLE) is a common approach used to achieve this.

**3. Latent Space:**

Many Generative AI models operate in a latent space—a lower-dimensional representation of the data that captures meaningful features. Probability distributions in the latent space guide the generation process. Latent space manipulation allows for creative control over generated outputs.

**4. Sampling Techniques:**

Generating new data points involves sampling from learned probability distributions. Generative models utilize techniques like Monte Carlo sampling, Markov Chain Monte Carlo (MCMC), and variational sampling to draw new samples from the latent space.

**5. Mode Collapse:**

Mode collapse is a phenomenon where a Generative AI model generates a limited set of outputs, ignoring the full diversity of the training data. Addressing mode collapse often involves improving the model architecture, introducing regularization, or using novel training strategies.

**6. Evaluation Metrics:**

Evaluating the quality of generated data is crucial. Metrics such as Inception Score, Fréchet Inception Distance (FID), or Perceptual similarity are employed to measure how closely generated data matches the distribution of real data. These metrics provide a quantitative assessment of the model's performance.

**7. Bayesian Generative Models:**

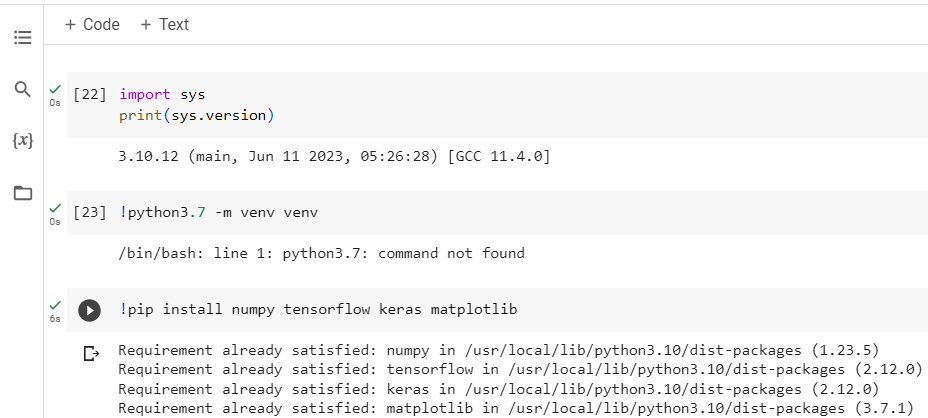
Bayesian approaches combine probabilistic modeling with Generative AI. Bayesian Generative Models provide uncertainty estimates along with generated samples, making them more reliable for decision-making tasks.

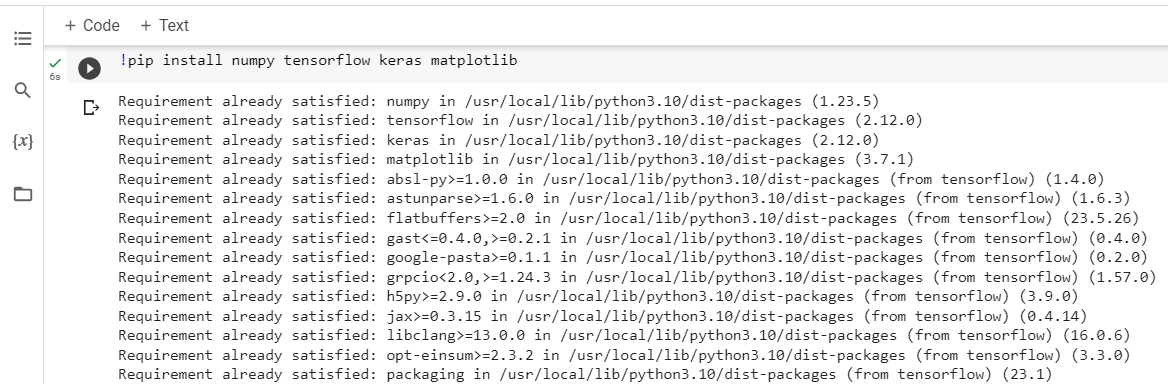
**8. Data Augmentation:**

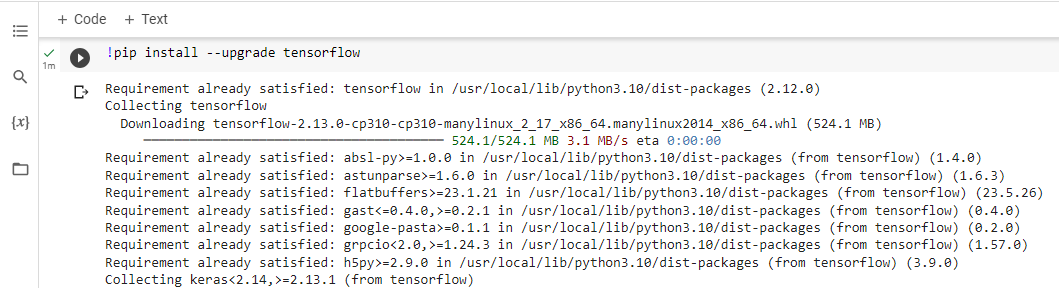
Generative models are used for data augmentation, generating synthetic samples to increase the diversity of training data. This is especially valuable when training other machine learning models, as it improves generalization.

**Task Name: Generating Handwritten Digits with Variational Autoencoders (VAE)**

1. Install the required libraries .

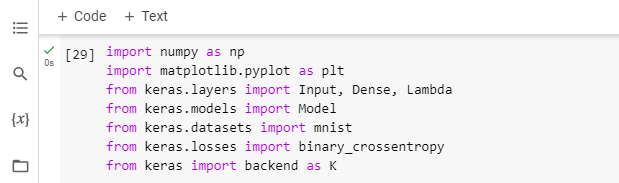






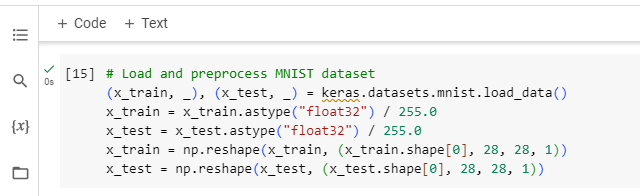
1. Import necessary libraries

Import the required libraries: 'numpy' for numerical operations, 'tensorflow' for creating and training neural networks, 'keras' for building and training models, and 'matplotlib.pyplot' for 'visualization'.



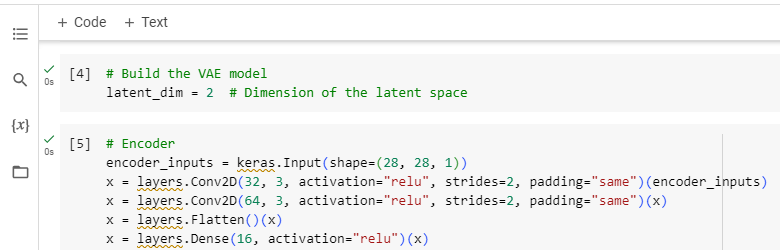
1. Load and preprocess MNIST dataset :

Load the MNIST dataset, which consists of handwritten digits. The data is then preprocessed: pixel values are normalized to the range '[0, 1]' by dividing by 255.0, and the data is reshaped to have a shape of '(num\_samples, 28, 28, 1)' to fit the input shape of the convolutional neural network.



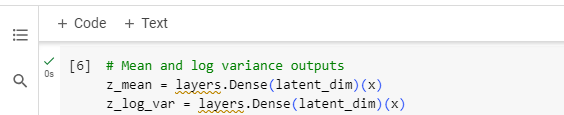
1. Build the VAE model and specify the Encoder Architecture :

Build the encoder part of the Variational Autoencoder (VAE). The encoder takes the input images and transforms them into a lower-dimensional latent space. It consists of convolutional and dense layers to extract features from the input images.



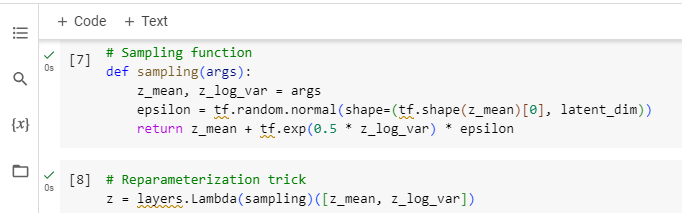
1. Mean and log variance outputs :

These layers produce the mean and log variance values that are used for sampling from the latent space.



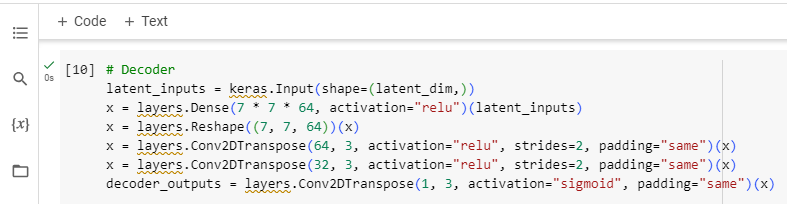
1. Sampling function using the reparameterization trick :

Define the sampling function using the reparameterization trick, which is crucial for training the VAE. It allows you to sample from the latent space while still being able to compute gradients during backpropagation.



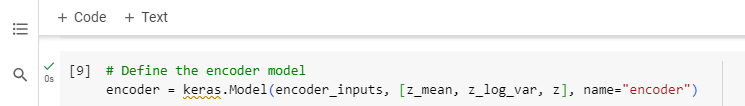
1. Decoder architecture :

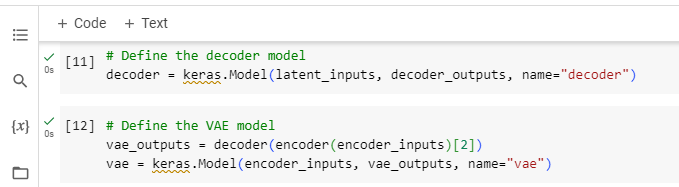
Construct the decoder part of the VAE. It takes samples from the latent space and reconstructs them back into images. The architecture is symmetrical to the encoder architecture, using transposed convolutional layers to upscale the data.



1. Define the encoder, decoder, and VAE models :

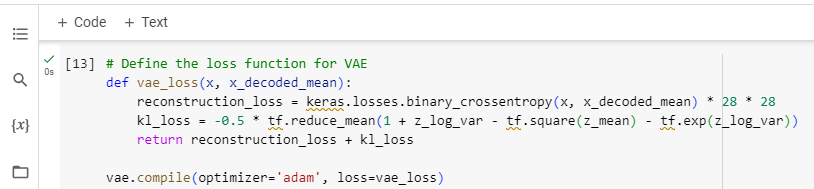
The encoder and decoder models are defined using the defined layers. The VAE model is constructed by connecting the encoder and decoder.





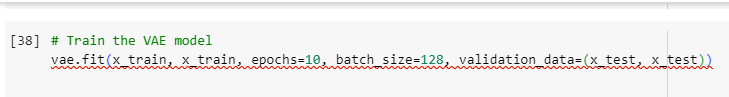
1. Define the VAE loss function :

Define the VAE loss function, which is a combination of reconstruction loss and a KL divergence term. The VAE model is compiled with the specified optimizer and loss function.



1. Train the VAE model :

VAE model is trained using the training data' (x\_train)' as both the input and target. It's trained for 10 epochs with a batch size of 128 and validated using the test data '(x\_test)'.



1. Generate new digits and Visualize the generated digits :

