Autoencoders for Anomaly Detection

1. Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? How many entries and variables does the dataset comprise?
→
The dataset provides information about the daily status of operational hard drives, including details such as serial numbers, models, capacities, failure status, and SMART (Self-Monitoring, Analysis and Reporting Technology) attributes. It likely aims to monitor hard drive health and predict failures.

The dataset contains structured data, likely organized in tabular form, with rows representing individual hard drives and columns representing different attributes or features associated with each hard drive.

The dataset comprises 3,179,295 entries (or rows), each representing a daily snapshot of a hard drive. There are 95 variables (or columns) in the dataset, including attributes such as date, serial number, model, capacity_bytes, failure status, and various SMART attributes (e.g., smart_1_normalized, smart_1_raw, etc.).

Overall, the dataset seems to be aimed at analyzing the health and performance of operational hard drives over time, potentially for tasks such as predicting hard drive failures or optimizing maintenance schedules.

Reasons for choosing this dataset-

Autoencoders are neural network architectures commonly used for unsupervised learning tasks such as dimensionality reduction and feature learning. The hard drive test data provides a rich source of structured data, which can serve as an excellent input for training autoencoder models. By utilizing autoencoders on this dataset, we can explore how well the models can learn to reconstruct the input data and potentially identify anomalies or patterns indicative of hard drive failures.

The dataset represents real-world operational hard drive data, making it relevant for practical applications where the goal is to monitor and maintain the health of hardware systems.
By leveraging autoencoders on this dataset, researchers or practitioners in the field can develop anomaly detection systems that automatically identify abnormal behavior in hard drives, helping to prevent data loss and system downtime.

2. Describe the details of your autoencoder models, including the layers, activation functions, and any specific configurations employed.
→

Autoencoder 1:

Encoder Layers:

- Linear layer with input dimension and output dimension of 64.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 64 and output dimension of 32.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.

Decoder Layers:

- Linear layer with input dimension of 32 and output dimension of 64.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 64 and output dimension equal to the input dimension.
- Sigmoid activation function.

Autoencoder 2:

Encoder Layers:

- Linear layer with input dimension and output dimension of 128.
- ReLU activation function.
- Batch normalization layer.
- Linear layer with input dimension of 128 and output dimension of 64.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 64 and output dimension of 32.
- ReLU activation function.

Decoder Layers:

- Linear layer with input dimension of 32 and output dimension of 64.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 64 and output dimension of 128.
- ReLU activation function.

- Batch normalization layer.
- Linear layer with input dimension of 128 and output dimension equal to the input dimension.
- Sigmoid activation function.

Autoencoder 3:

Encoder Layers:

- Linear layer with input dimension and output dimension of 32.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 32 and output dimension of 16.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 16 and output dimension of 8.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.

Decoder Layers:

- Linear layer with input dimension of 8 and output dimension of 16.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 16 and output dimension of 32.
- ReLU activation function.
- Dropout layer with a dropout rate of 0.2.
- Linear layer with input dimension of 32 and output dimension equal to the input dimension.
- Sigmoid activation function.

These autoencoder models are trained to reconstruct the input data while learning a compressed representation (latent space) of the data. The models utilize different architectures and regularization techniques to achieve this objective.

3.Discuss the results and provide relevant graphs:

a. Report training accuracy, training loss, validation accuracy, validation loss, testing accuracy, and testing loss.

→

For Autoencoder 2, below are the metrics -

Train Loss: **0.626149**,

Val Loss: **0.678909,**

Test Loss: **0.647172,**

Anomaly Detection Accuracy for Autoencoder 2: **0.8913**

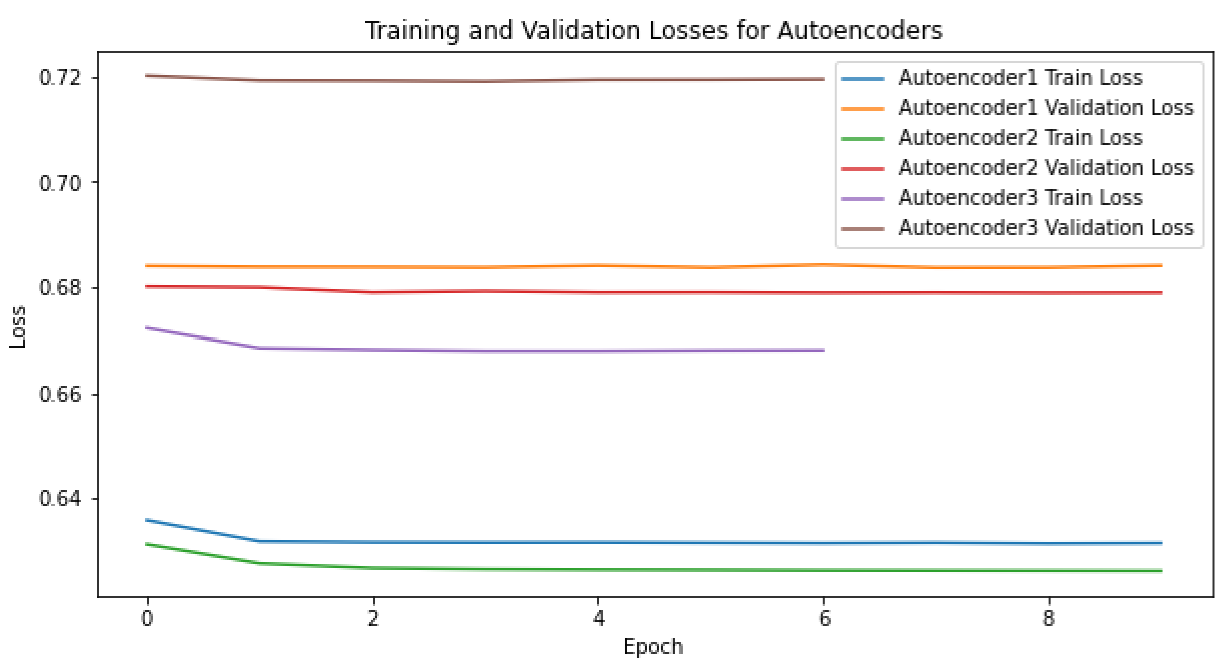
b. Plot the training and validation accuracy over time (epochs).

→

For the dataset we have chosen, we cannot plot training and validation accuracy over epochs as the autoencoders are primarily used for unsupervised learning tasks, where the model learns representations of the input data without the need for labeled examples. In this context, accuracy metrics, which are typically used in supervised learning tasks, are not applicable or meaningful. The goal of an autoencoder is to minimize the reconstruction error, rather than maximizing accuracy, which we have calculated.

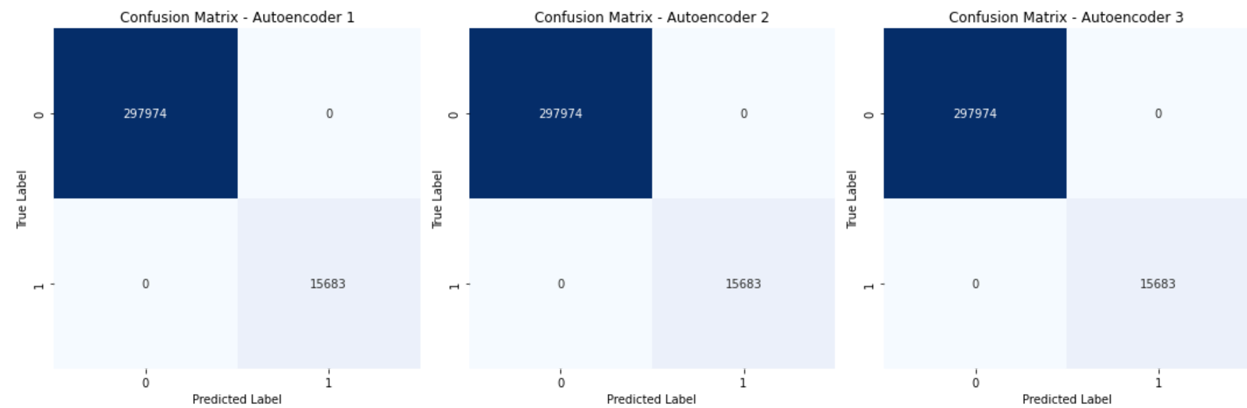
c. Plot the training and validation loss over time (epochs).

→



d. Generate a confusion matrix using the model's predictions on the test set.

→



e. Report any other evaluation metrics used to analyze the model's performance on the test set.

→

Summary Statistics of Reconstruction Errors for Autoencoder 1:

Mean: 0.6520683
Median: 0.10480258
Maximum: 3881.413
Minimum: 0.038602497
Standard Deviation: 24.402441

Summary Statistics of Reconstruction Errors for Autoencoder 2:

Mean: 0.6471717
Median: 0.10472429
Maximum: 3866.0828
Minimum: 0.038480666
Standard Deviation: 24.272675

Summary Statistics of Reconstruction Errors for Autoencoder 3:

Mean: 0.6879178
Median: 0.13882816
Maximum: 3877.9902
Minimum: 0.039717235
Standard Deviation: 24.417933

Anomaly Detection Accuracy for Autoencoder 1: 0.8919
Anomaly Detection Accuracy for Autoencoder 2: 0.8913
Anomaly Detection Accuracy for Autoencoder 3: 0.9148

4. Discuss the strengths and limitations of using autoencoders for anomaly detection.
→
Autoencoders are a type of neural network architecture commonly used for unsupervised learning tasks such as dimensionality reduction and anomaly detection.Below are their strengths and limitations in the context of anomaly detection:

Strengths:

- Seeing Beyond the Straight Lines: Unlike simpler methods, autoencoders can identify anomalies even when the patterns are complex and not easily captured by linear approaches.
- Training Without Labels: Anomalies can be scarce and labeling them can be costly. Autoencoders come to the rescue as they don't require labeled data for training.
- Unveiling Hidden Clues: By compressing data into a smaller form (latent space), autoencoders can automatically discover hidden patterns that might indicate anomalies.
- Adaptable to the Task: The structure of autoencoders can be adjusted to fit the specific data and the kind of anomalies you're looking for, making them quite versatile.
- Resilient to Noise: Autoencoders are good at filtering out background noise and outliers, focusing on faithfully reconstructing the core data.

Weaknesses:

- Reconstruction Errors Can Be Tricky: Autoencoders rely on reconstruction error (the difference between the original data and its reconstruction) to spot anomalies. If the autoencoder itself has trouble reconstructing normal data, it might struggle to tell anomalies apart from normal variations.
- Missing the Rarest Oddities: Autoencoders might prioritize reconstructing the most common patterns in the data. This could lead them to miss rare anomalies that are very different from the training data.
- Data Bias Can Lead to Mistakes: The compressed representation learned by autoencoders can inherit biases from the training data. This can result in false alarms (flagging normal data as anomalies) or missed anomalies if the training data is not representative or has biases.
- Understanding the Why Can Be Difficult: While autoencoders are great at finding anomalies, it can be challenging to understand why they flag specific data points as anomalies by interpreting the features in the latent space.
- Large Datasets Can Take Time: Training complex autoencoder models on massive datasets can be computationally expensive and time-consuming, limiting their use in real-time or high-volume applications.