AI/ML Internship

A Project Report submitted to the

GLOBAL NEXT CONSULTING INDIA PVT LTD

(Six – Week Internship Program)

By

LALIT BHARDWAJ

Under the Supervision of

Dr. Anuradha Gupta
(Project Director)

Submitted To :

Global Next Consulting India Pvt. Ltd.

Duration of Internship :

23-March-2026 to 04-May-2026

March 2026

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report, **"AI/ML Internship (GNCIPL)"**, submitted as per the requirements for the Artificial Intelligence and Machine Learning Engineer, This is the result of original work carried out by me under the guidance of **Ms. Anuradha Gupta** during the time period from March 2026 to May 2026.

I further declare that this report represents authentic record of my own work and does not contain any falsely fabricated ideas, data, facts or sources. I also declare that I have adhered to all principles of academic honesty and integrity and that this report has not been submitted, either in part or in full, to any other institute, university, or organization for the award of any degree, diploma, or certification.

Lalit Bhardwaj

CERTIFICATE

This is to certify that the project report entitled **"AI/ML Internship Report"** has been carried out by **Lalit Bhardwaj ,** a Fresher in job search and improve skill in AI and Machine learning role. This work was carried out under the guidance of **Ms. Anuradha Gupta** from March 2026 to May 2026. It is further certified that this work has not been submitted to any other university or institution for the award of any other degree, diploma or certificate.

Ms. Anuradha Gupta
Program Director
GNCIPL

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude to all those who contributed to the successful completion of this project report.

I would like to express my sincere gratitude to my supervisor, Ms. Anuradha Gupta, for their invaluable guidance, encouragement, and constructive suggestions throughout the course of this work. Their expertise and constant support played a crucial role in the successful completion of this report.

I am also thankful to the staff of Global Next Consulting India Private Limited for providing the necessary resources, facilities and their assistance, without which this study would not have been possible.

Finally, I would also like to acknowledge my peers and teachers whose support and discussions have been helpful in the completion of this report.

Lalit Bhardwaj

ABSTRACT

This report summarizes my six-week internship as a AI/ML Intern at Global Next Consulting India Pvt. Ltd., Noida. The internship was structured into six Projects, In which five minor projects as per each tool and one major project, aimed at developing practical skills in Python, ML algorithms, and AI concepts.

The internship projects as a whole strengthened my technical skills in Python, Machine Learning (ML), python libraries, and data visualization (Matplotlib & Seaborn), while also improving my Presentation, analytical thinking and problem-solving approach. The work highlights how Artificial intelligence and machine learning can uncover meaningful insights to support informed decision-making in domains such as public health, economy and business.

INDEX

Chapter 1- Introduction

1.1 Company's Profile

Global Next Consulting India Private Limited (GNCIPL), headquartered in Greater Noida, Uttar Pradesh, is a cybersecurity-focused consulting firm dedicated to helping organizations protect their digital assets, data, and reputation. As threats evolve in today's digital world, GNCIPL offers proactive, customized solutions rather than reactive fixes. The company serves clients in diverse sectors including finance, healthcare, manufacturing, and technology, providing services like threat detection, risk assessment, incident response, compliance consulting, and 24/7 monitoring. GNCIPL's core values are integrity, innovation, customer-centricity, excellence, and collaboration - ensuring that technical solutions align with clients' specific needs and long-term goals.

Contact Details

Location- B5,402 P4 PHi2, CGEWHO TOWER, GREATER NOIDA 201310
Contact Numbers- 0120-4001768, +91-9315504902. +91-7666141260
Mail- hr@gncipl.com

1.2 Objectives of Internship

During my six-week internship at GNCIPL as an AI/ML Intern, the main objectives were:

- To gain hands-on experience in AI & ML tools and techniques, especially using Python (Google Colab, Jupyter Notebook), python libraries and its applications.
- To work on real-world datasets and deliver meaningful insights, visualizations, and dashboard reports.
- To learn data preprocessing, cleaning, transformation, and applying formulas and classification logic.
- To enhance analytical thinking, effective communication, and presentation skills through weekly minor projects and a major end project.

Chapter 2 - Projects

2.1 Stock Price Movement of Tech Giants (Week 1)

2.1.1 Introduction

The financial markets are highly dynamic, and the technology sector often dictates broader market trends. Analyzing the stock price movements of tech giants such as Apple (AAPL), Google (GOOGL), Microsoft (MSFT), and Amazon (AMZN) provides critical insights into market sentiment, economic health, and sector volatility.

As part of the Week 1 deliverables for the AI/ML internship, this project focuses on tracking and analyzing the daily price and trading volume of these major corporations. By leveraging Python-based data analysis libraries, the project aims to uncover hidden patterns in historical stock data, compare price volatilities, and calculate essential financial indicators like Moving Averages. This foundational analysis serves as the critical first step before deploying advanced predictive machine learning models for time-series forecasting.

2.1.2 Objectives

Primary Objectives

- **Data Acquisition:** To programmatically fetch historical daily stock prices and volume data for selected tech giants using financial APIs (e.g., Yahoo Finance).
- **Data Preprocessing:** To clean the dataset, handle missing values, and structure time-series data for accurate analysis.
- **Trend Analysis:** To identify long-term and short-term price trends using Simple Moving Averages (SMA) and Exponential Moving Averages (EMA).
- **Volatility Comparison:** To calculate and compare daily returns and price volatility across the selected tech companies.
- **Visualization:** To generate comprehensive visual reports (candlestick charts, correlation heatmaps, and volume distributions) to support data-driven financial decision-making.

2.1.3 Methodology

a) Data Collection & Integration

- Utilized the `yfinance` API (Yahoo Finance) to download historical stock data (Open, High, Low, Close, Adjusted Close, and Volume) over a specified multi-year timeframe.
- Combined individual company datasets into a unified Pandas DataFrame, indexing the data chronologically by trading dates.

:

Ticker	AAPL	GOOGL	MSFT
Date			
2020-01-02	72.400513	67.873024	152.158371
2020-01-03	71.696617	67.517967	150.263794
2020-01-06	72.267952	69.317604	150.652145
2020-01-07	71.928055	69.183685	149.278564
2020-01-08	73.085114	69.676132	151.656311

b) Data Cleaning and Preprocessing

- Checked for and imputed missing trading days or null values to ensure continuous time-series integrity.
- Standardized column headers and calculated the "Daily Percentage Return" to normalize price movements for fair comparison across stocks with drastically different share prices.

c) Feature Engineering

- **Moving Averages:** Calculated the 50-day Simple Moving Average (SMA) to capture short-term trends and the 200-day SMA for long-term trends.
- **Volatility Metrics:** Computed the standard deviation of daily returns to measure the historical volatility of each stock.

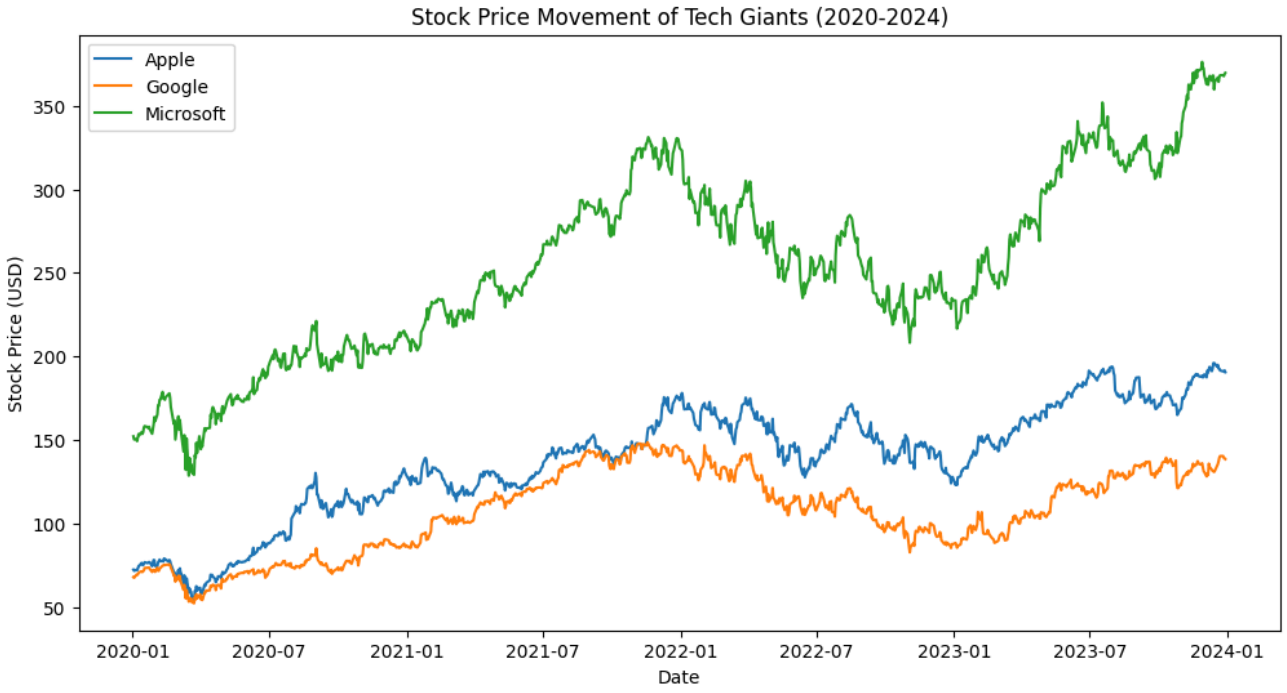
2.1.4 Results and Insights

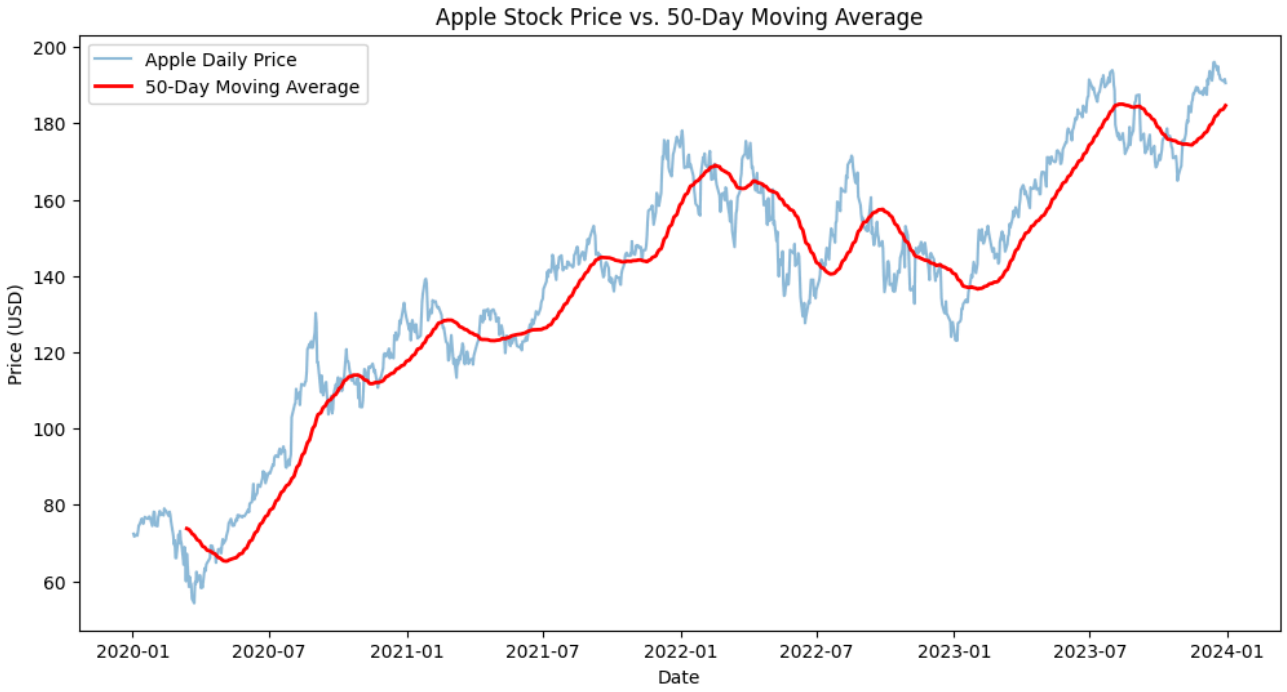
- **Trend Identification:**
  The visualization of the 50-day and 200-day moving averages revealed clear periods of bullish and bearish trends. Instances where the 50-day SMA crossed above the 200-day SMA (a "Golden Cross") historically aligned with sustained upward price momentum for the tech giants.

- **Volatility and Risk:**
  The daily return distribution analysis showed that while all selected companies exhibit growth over the long horizon, companies like Amazon and Apple displayed specific periods of higher volatility compared to the relatively stable growth of Microsoft.

- **Volume-Price Correlation:**
  Spikes in trading volume frequently correlated with significant price gaps (both up and down), highlighting the market's aggressive reaction to quarterly earnings reports and tech product launch events.

- **Sector Correlation:**
  The correlation heatmap demonstrated a strong positive correlation (often > 0.75) between the daily returns of AAPL, GOOGL, and MSFT, confirming that these mega-cap tech stocks are heavily influenced by the same macroeconomic factors and sector-specific news.





2.1.5 Conclusion

The Week 1 project successfully established a robust data pipeline for financial analysis. By systematically cleaning, engineering, and visualizing stock market data, the analysis provided a clear, quantitative picture of tech sector performance. Understanding these historical trends, moving averages, and volatility metrics is crucial. It not only provides immediate analytical value to financial stakeholders but also acts as the vital preprocessing phase required for feeding clean, engineered features into future AI-driven predictive models, such as LSTMs or ARIMA, for algorithmic trading strategies.

2.2 Cryptocurrency Price Volatility (EDA) (Week 2)

2.2.1 Introduction

Unlike traditional financial markets, cryptocurrency markets operate 24/7 and are renowned for their extreme price fluctuations. Understanding these erratic movements is crucial for risk management and algorithmic trading. Following the completion of the Week 1 project, the second week of the internship focused heavily on Exploratory Data Analysis (EDA) projects. This specific project falls under the Finance domain and is centered on analyzing the historical price behavior of major cryptocurrencies. By applying structured EDA techniques, the goal was to dissect price volatility, uncover hidden momentum shifts, and prepare the dataset for future predictive machine learning models.

2.2.2 Objectives

The primary objectives of this exploratory project were:

- **Data Sourcing:** To extract historical cryptocurrency market data using industry-standard sources like the CoinGecko API and CoinMarketCap.
- **Trend & Momentum Analysis:** To calculate and visualize rolling averages to smooth out daily price noise and identify broader market trends.
- **Inter-Asset Relationships:** To analyze the correlation between different digital assets (e.g., Bitcoin vs. Altcoins) using statistical heatmaps.

- **Behavioral Analysis:** To explore sentiment linkage, understanding how external market sentiment and news cycles correlate with sudden price shifts.
- **Anomaly Detection:** To implement outlier and anomaly detection techniques to identify flash crashes and abnormal trading volumes.

2.2.3 Methodology

The project utilized a robust Python-based EDA workflow, leveraging libraries such as Pandas, Seaborn, and Matplotlib.

1. Data Collection & Preprocessing

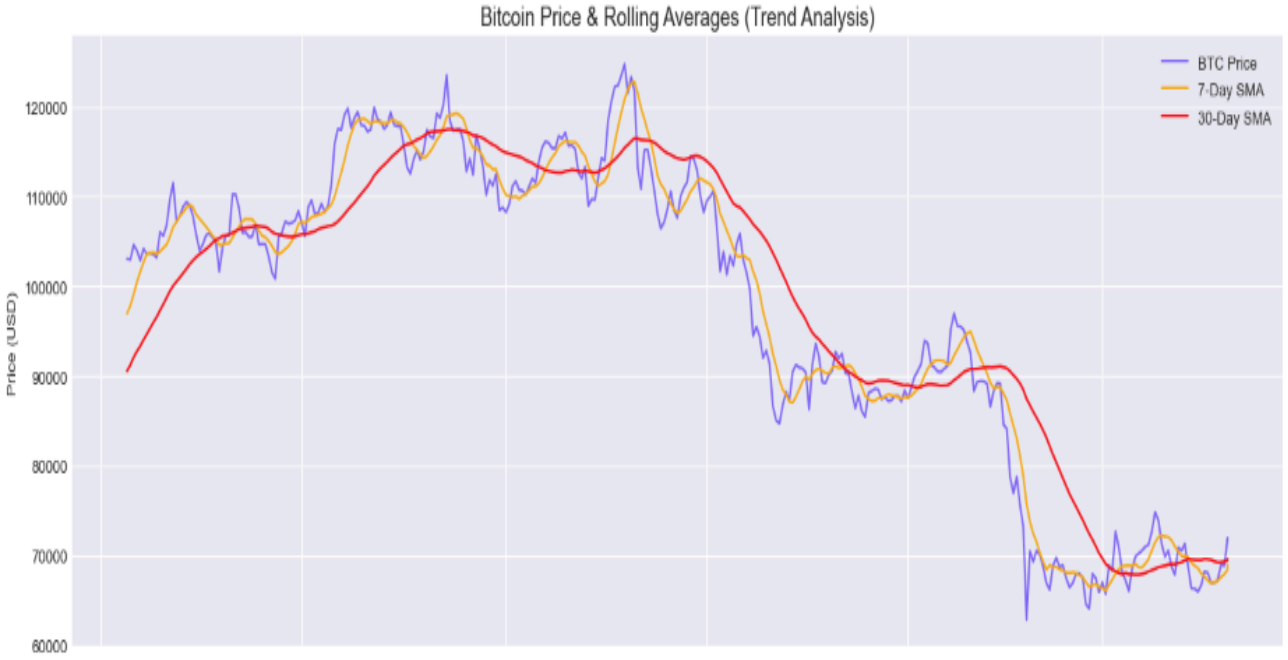
- Queried historical price, trading volume, and market capitalization data via the CoinGecko API and CoinMarketCap datasets.
- Cleaned the raw data by handling missing timestamps and normalizing trading volumes to account for extreme variances.

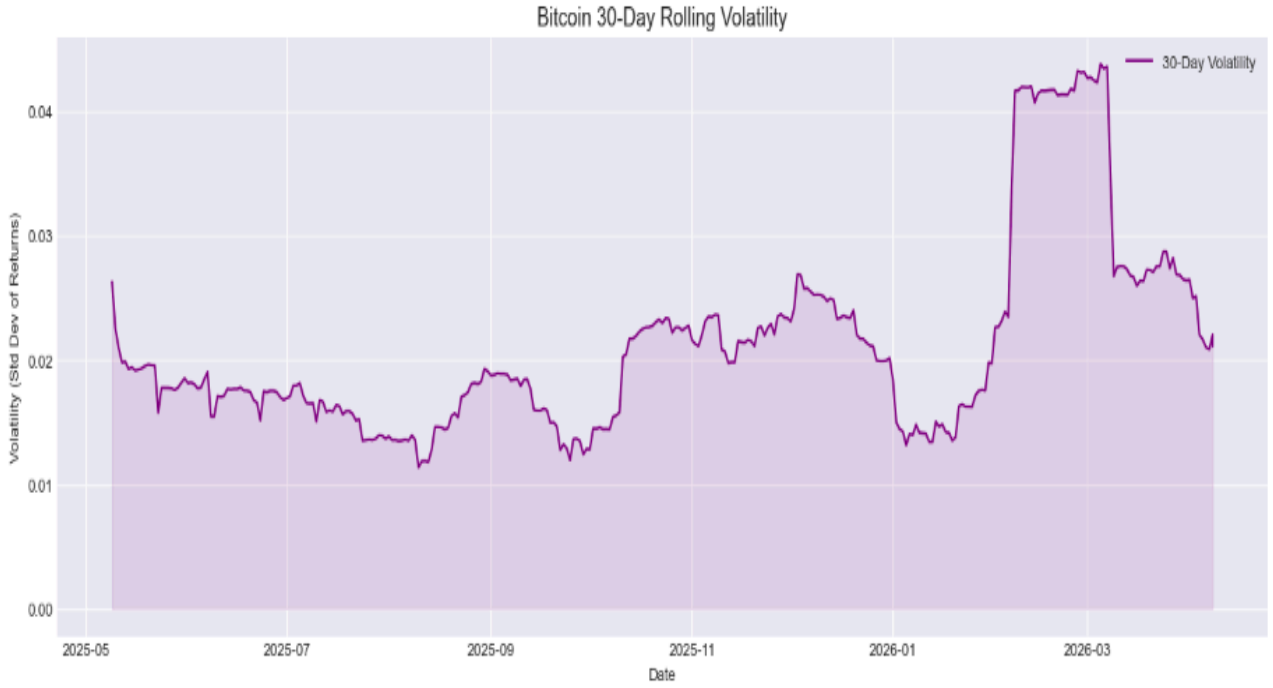
2. Time-Series Analysis & Feature Engineering

- Conducted extensive time-series analysis to plot historical price action.
- Engineered new features by calculating 7-day, 30-day, and 90-day rolling averages to analyze momentum.
- Computed the daily percentage change to establish a baseline for volatility measurement.

3. Statistical Visualization

- Generated Histograms, Boxplots, and Kernel Density Estimates (KDE) to visualize the distribution of daily returns and clearly identify periods of extreme volatility.
- Created correlation matrices and heatmaps to compare the daily returns of top cryptocurrencies against one another.





2.2.4 Results and Insights

- **Volatility Visualization:**

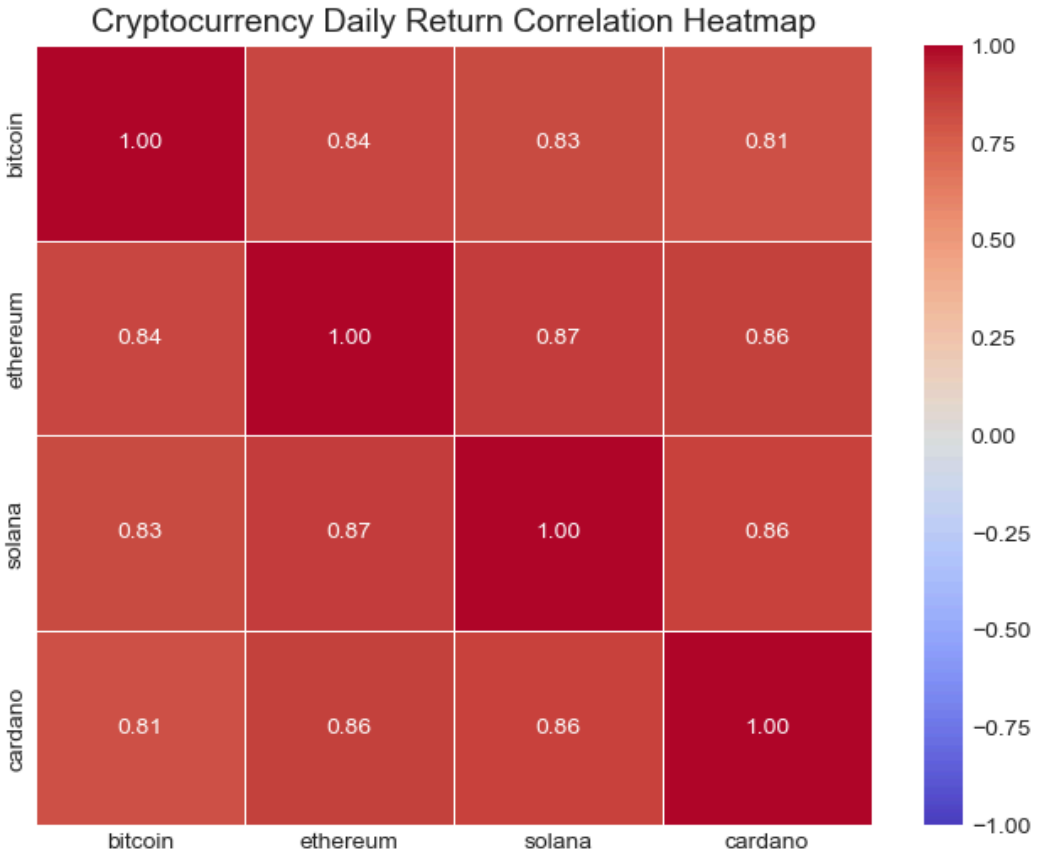
  The Boxplots and KDE distributions revealed that cryptocurrency returns possess a "fat-tailed" distribution. This indicates that extreme price movements (both positive and negative outliers) occur much more frequently than in traditional stock markets.

- **Momentum Shifts:**

  The application of rolling averages successfully filtered out intraday noise. Crossovers between short-term (e.g., 7-day) and long-term (e.g., 30-day) rolling averages often preceded prolonged bullish or bearish market phases.

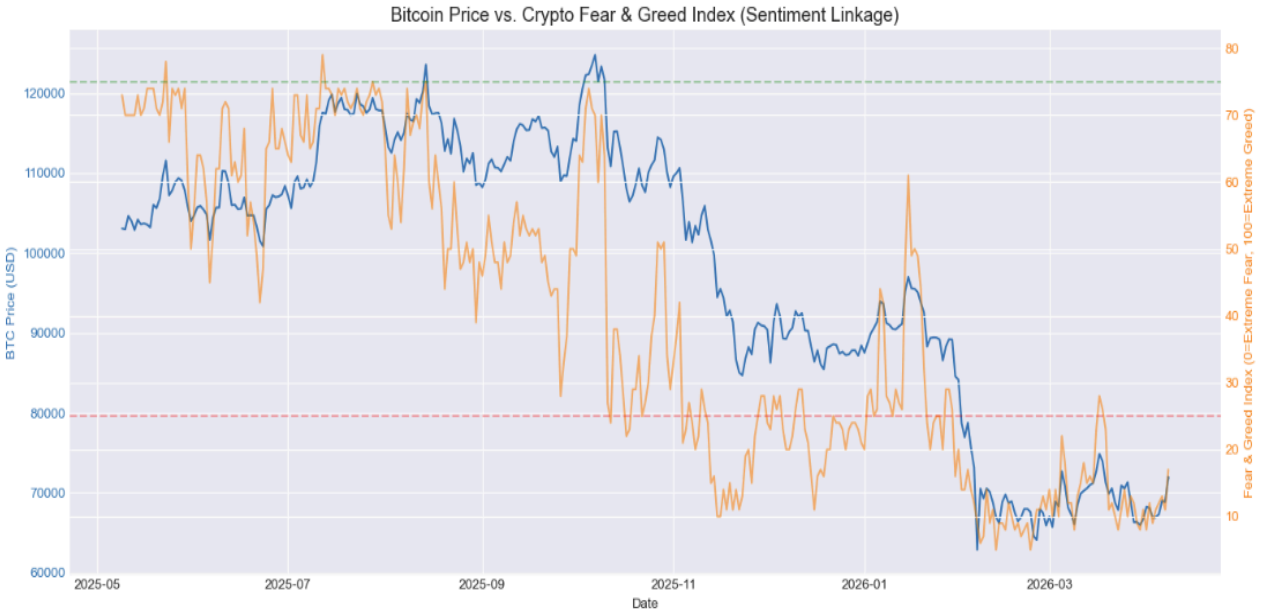
- **Asset Correlation:**

  The correlation & heatmaps analysis demonstrated a highly positive correlation between Bitcoin (BTC) and major altcoins like Ethereum (ETH). This suggests that macroeconomic factors drive the crypto market as a whole, rather than isolated, asset-specific events.



- **Sentiment Linkage:**

  By overlaying major market news dates with the price charts, the sentiment linkage analysis confirmed that sharp spikes in trading volume and extreme price volatility heavily align with shifts in retail and institutional sentiment (e.g., regulatory news or macroeconomic announcements).



2.2.5 Conclusion

The Week 2 project successfully highlighted the power of Exploratory Data Analysis in dissecting complex, high-noise financial datasets. By utilizing rolling averages, anomaly detection, and correlation mapping, the project transformed raw API data into actionable financial intelligence. Understanding the underlying volatility and sentiment linkage in cryptocurrencies is a critical prerequisite for the next stages of the AI/ML pipeline, specifically for engineering features that will feed into predictive algorithms and automated trading models.

2.3 Fashion Image Clustering (Week-3)

2.3.1 Introduction

Following the exploratory data analysis of structured datasets in the previous weeks, the focus of Week 3 shifted to the realm of Computer Vision and Unsupervised Learning. This project focuses on clustering unstructured image data, specifically using the Fashion MNIST dataset. Fashion MNIST serves as a more complex, modern drop-in replacement for the traditional handwritten digits dataset, consisting of 28x28 grayscale images of clothing items across 10 distinct categories (e.g., shirts, sneakers, bags).

The core challenge of this project was to group similar clothing items together without relying on their actual category labels. To achieve this, the project utilized a hybrid machine learning approach: leveraging Convolutional Neural Networks (CNNs) to extract meaningful, high-level spatial features from the raw images, followed by applying the K-Means clustering algorithm to group these feature vectors. This methodology mirrors real-world e-commerce applications, such as automated inventory categorization and visual recommendation systems.

2.3.2 Objectives

The primary objectives of this image clustering project were:

- **Data Preparation:** To load, normalize, and preprocess the Fashion MNIST dataset for deep learning applications.

- **Feature Extraction:** To implement a Convolutional Neural Network (CNN) to extract dense, high-dimensional feature embeddings from the images, moving beyond raw pixel-value flattening.
- **Unsupervised Clustering:** To apply the K-Means algorithm to the extracted CNN features to dynamically group the clothing items into distinct clusters.
- **Dimensionality Reduction & Visualization:** To apply techniques like PCA (Principal Component Analysis) or t-SNE to project the high-dimensional clusters into a 2D space for visual interpretation.
- **Evaluation:** To evaluate the quality of the clusters using internal metrics such as the Silhouette Score and the Elbow Method.

2.3.3 Methodology

The project was executed using Python, integrating deep learning frameworks (like TensorFlow/Keras or PyTorch) with traditional machine learning libraries (Scikit-Learn).

1. Data Loading & Preprocessing

- Imported the Fashion MNIST dataset, which contains 60,000 training images and 10,000 test images.
- Normalized the pixel values (scaling them from 0–255 to a 0–1 range) to ensure faster convergence during the feature extraction phase.
- Reshaped the data to include a channel dimension, preparing it for convolutional layers.

2. Feature Extraction via CNN

- Designed a CNN architecture (acting as the encoder portion of an autoencoder or utilizing a pre-trained truncated model) to process the images.
- By passing the images through successive convolutional and pooling layers, the network effectively compressed the spatial patterns (like edges, textures, and shapes) into a flattened, dense feature vector for each image.

3. K-Means Clustering

- Fed the newly extracted, dense CNN feature vectors into a K-Means clustering model.
- Although the number of true classes is known (10), the Elbow Method was utilized conceptually to validate the optimal number of clusters (K) based on inertia. The model was ultimately configured to group the data into 10 clusters to observe how closely the algorithm could recreate the actual clothing categories.

4. Dimensionality Reduction and Visualization

- Applied t-SNE to reduce the high-dimensional CNN feature vectors down to two dimensions.
- Generated scatter plots where each data point was color-coded based on its K-Means cluster assignment, allowing for a visual inspection of cluster purity and separation.

2.3.4 Results & Insights

- **Superiority of CNN Features**:

  The analysis clearly demonstrated that applying K-Means to CNN-extracted features yielded significantly tighter and more distinct clusters compared to running K-Means on raw, flattened pixel arrays. The CNN successfully captured semantic meaning (e.g., the strap of a bag vs. the sole of a shoe).

- **Cluster Overlaps:**

  The visualizations revealed logical overlaps within the clusters. For instance, footwear categories (sneakers, ankle boots, and sandals) clustered closely together but remained distinctly separated from top-wear (T-shirts, pullovers, and coats).

- **Ambiguity Handling:**

  Certain classes, such as "shirts" and "coats," exhibited closer boundaries and occasional mis-clustering, highlighting the inherent visual similarities between long-sleeved garments in low-resolution grayscale images.

- **Visual Validation:**

The t-SNE scatter plot provided strong visual validation of the K-Means algorithm's performance, showing definitive groupings that closely mirrored the actual, hidden labels of the dataset.

2.3.5 Conclusion

The Week 3 project successfully bridged the gap between deep learning and traditional unsupervised machine learning. By utilizing CNNs for advanced feature extraction and K-Means for grouping , the project achieved high-quality segmentation of the Fashion MNIST images. This approach highlights a powerful workflow for handling complex, unstructured data. The insights gained from this methodology are highly transferable to industry scenarios, particularly in retail and e-commerce, where clustering unlabelled visual data is essential for building robust visual search engines and automated product tagging systems.

2.4 Stock Movement Clusters (Week 4)

2.4.1 Introduction

Building upon the foundational financial data extraction performed in Week 1, the Week 4 project elevated the analysis by applying Unsupervised Machine Learning techniques. Financial markets generate massive volumes of high-dimensional time-series data. Identifying which stocks move synchronously is crucial for portfolio diversification, pairs trading, and risk management.

This project, titled "Stock Movement Clusters" , aimed to discover hidden, intrinsic structures within historical stock price movements. By utilizing dimensionality reduction techniques like Principal Component Analysis (PCA) and clustering algorithms like K-Means, the objective was to autonomously group stocks into trend-based clusters without relying on predefined sector labels. This data-driven categorization provides a purely mathematical perspective on market correlations.

2.4.2 Objectives:

Aligned with the Unsupervised Learning Project Playbook, the primary objectives of this project were:

- **Data Preparation:** To extract, smooth, and normalize historical time-series data for a diverse portfolio of stocks.
- **Dimensionality Reduction:** To apply PCA to reduce the high-dimensional daily return data while retaining the maximum variance, thereby eliminating noise.

- **Clustering:** To implement the K-Means clustering algorithm to group stocks exhibiting similar price movement patterns.
- **Model Tuning & Evaluation:** To utilize the Elbow Method and Silhouette Scores to determine the mathematically optimal number of clusters (K) and tune hyperparameters using GridSearchCV.
- **Visualization:** To project and visualize the resulting clusters in a 2D space for stakeholder interpretation.

2.4.3 Methodology

The project was executed using Python, leveraging yfinance for data acquisition and pandas and scikit-learn (sklearn) for machine learning and evaluation.

1. Data Collection & Preprocessing

- Downloaded historical daily closing prices for a basket of 50+ diverse stocks using the yfinance API.
- Calculated the daily percentage returns to represent relative movements rather than absolute price differences.
- Applied smoothing techniques to handle short-term market noise.
- Standardized the dataset using StandardScaler to ensure all stock movements were on a comparable scale, preventing highly volatile stocks from disproportionately dominating the distance calculations.

2. Dimensionality Reduction (PCA)

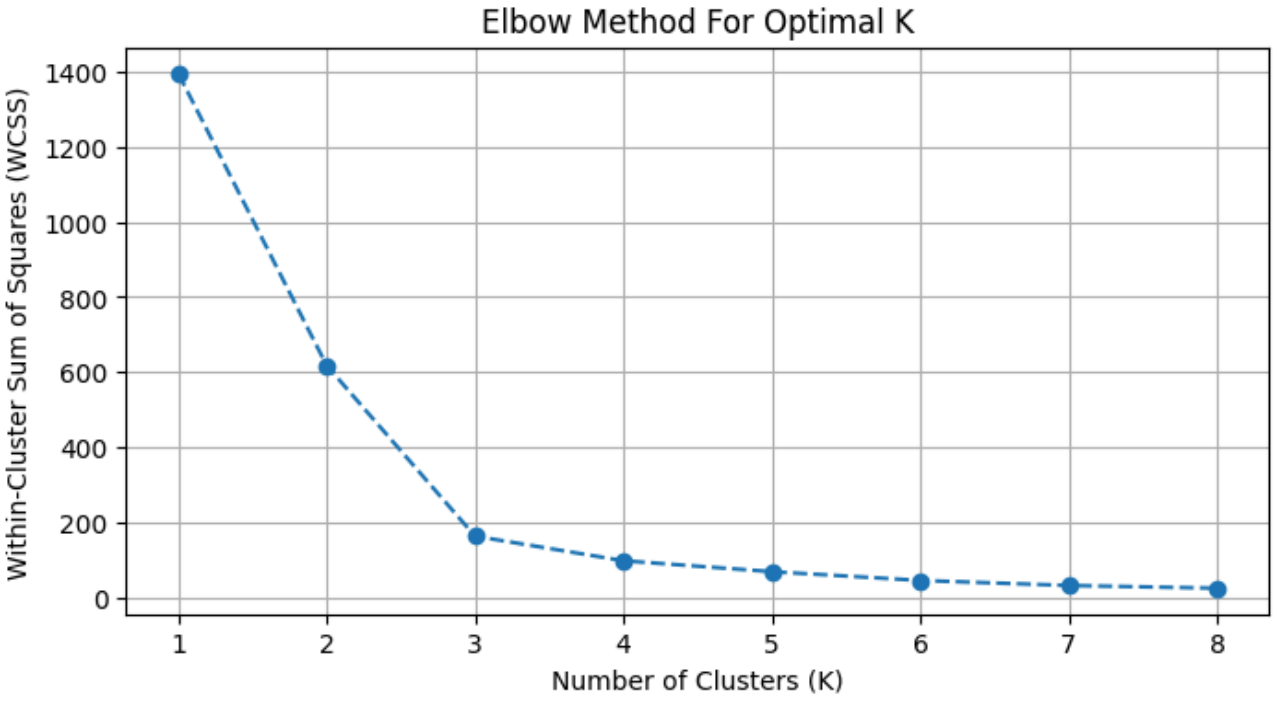
- Time-series data for a year contains 252 trading days (dimensions). To combat the "curse of dimensionality," PCA was applied.
- The components were tuned to capture >90% of the explained variance, effectively compressing the 252 dimensions into a smaller set of principal components that represent the core market drivers.

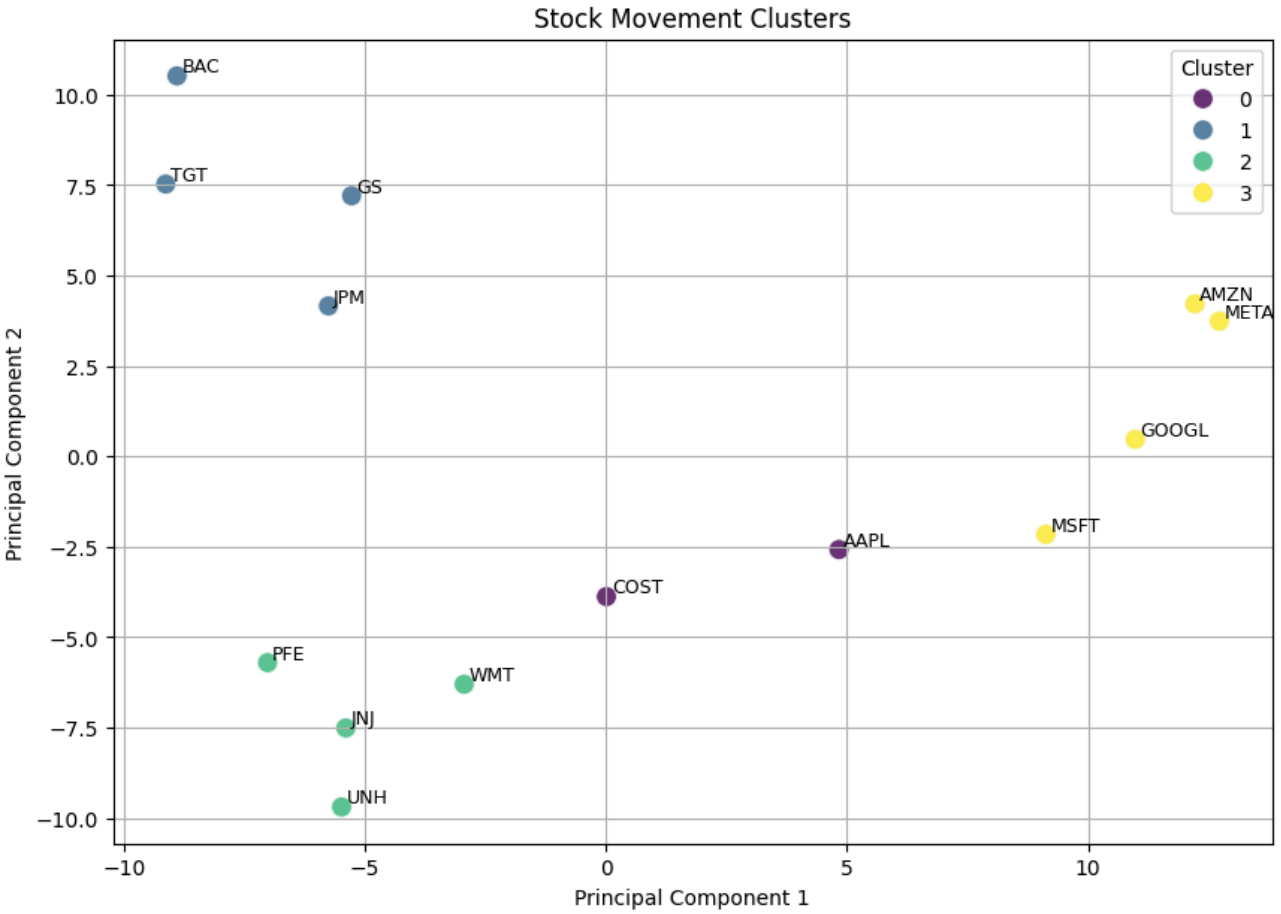
3. K-Means Clustering & Model Tuning

- Applied the KMeans algorithm from sklearn to the PCA-transformed dataset.
- Plotted the Within-Cluster Sum of Squares (WCSS) against different values of K to utilize the Elbow Method.
- Utilized GridSearchCV to exhaustively search for optimal hyperparameters, such as init (e.g., k-means++) and max_iter, to ensure stable and efficient convergence.

4. Visualization and Labeling

- Created scatter plots of the first two principal components, color-coded by their assigned K-Means cluster.
- Analyzed the constituents of each cluster to assign business-relevant labels (e.g., "High-Beta Tech," "Stable Utilities," "Cyclical Financials").


Elbow Method For Optimal K
Within-Cluster Sum of Squares (WCSS)
Number of Clusters (K)
0
200
400
600
800
1000
1200
1400
1
2
3
4
5
6
7
8


Stock Movement Clusters
Principal Component 2
Principal Component 1
Cluster
0
1
2
3
BAC
TGT
GS
JPM
AMZN
META
GOOGL
MSFT
AAPL
COST
PFE
WMT
JNJ
UNH
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10
−5
0
5
10

2.4.4 Results and Insights

- **Optimal Clusters:** The Elbow curve exhibited a distinct inflection point, suggesting that the dataset optimally divided into specific trend-based groups (e.g., $K=4$ or $K=5$), which strongly mirrored broader macroeconomic sectors.

- **Data-Driven Sector Discovery:** The K-Means algorithm successfully grouped historically correlated stocks together. Interestingly, the algorithm discovered relationships that crossed traditional sector boundaries, clustering certain retail stocks with technology stocks based on similar volatility and growth trajectories.

- **Noise Reduction Efficacy:** Utilizing PCA prior to K-Means clustering significantly improved cluster separation and Silhouette Scores compared to clustering on raw daily returns. The PCA components effectively captured the "market factor" and "sector factors" as the primary drivers of variance.

- **Risk Identification:** One distinct cluster isolated stocks with exceptionally high variance and extreme price swings, providing an automated method for isolating high-risk assets within a broad portfolio.

2.4.5 Conclusion

The Week 4 project successfully demonstrated the power of Unsupervised Learning in quantitative finance. By integrating data standardization, Principal Component Analysis, and K-Means clustering , the pipeline autonomously discovered trend-based stock groups hidden within raw market data. Applying model tuning techniques like GridSearchCV and validating with the Elbow Method ensured the mathematical robustness of these clusters. Ultimately, these data-driven groupings provide a highly objective foundation for building diversified investment portfolios and hedging against systemic market risks.

2.5 Customer Churn Prediction (Artificial Neural Networks) (Week 5)

2.5.1 Introduction

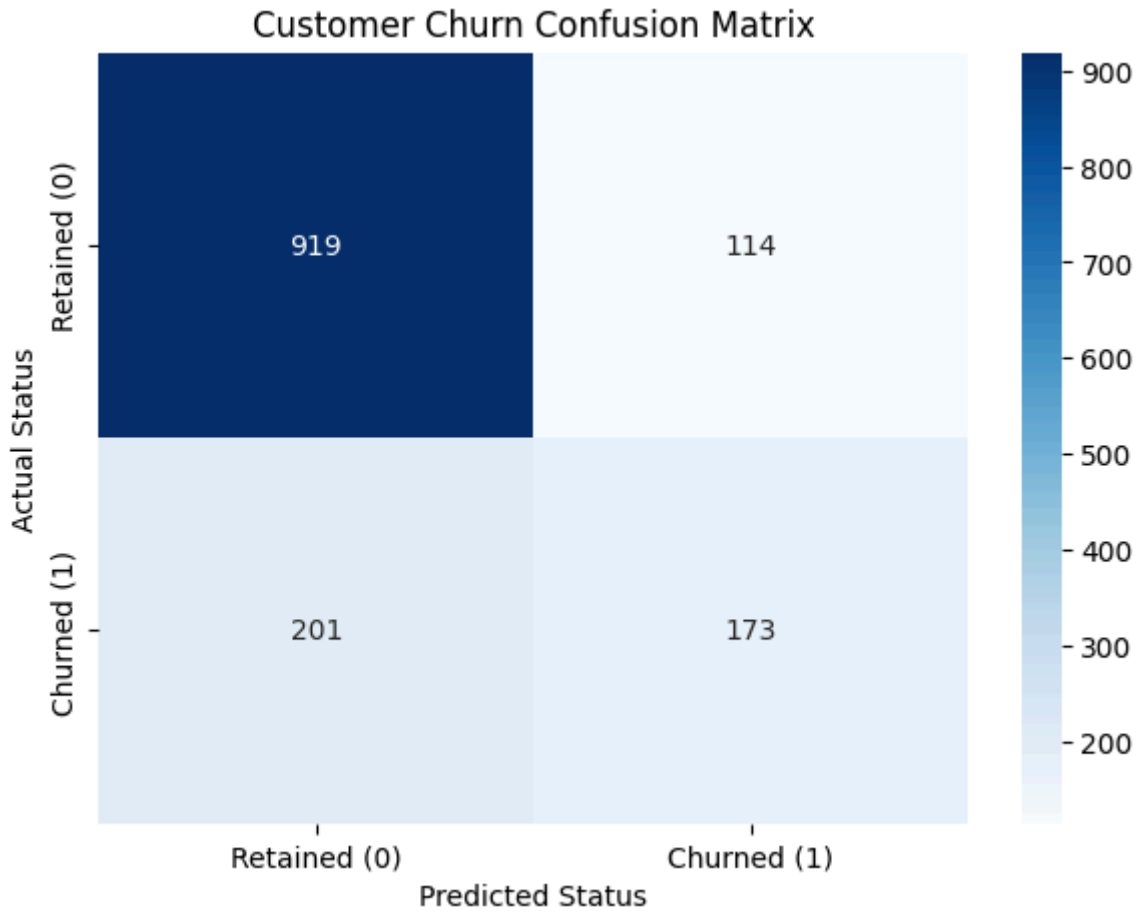
Following the exploration of unsupervised learning in Week 4, the internship curriculum for Week 5 transitioned into the domain of Deep Learning, focusing specifically on Artificial Neural Networks (ANNs). In the highly competitive telecommunications industry, retaining existing customers is significantly more cost-effective than acquiring new ones.

This project, "Customer Churn Prediction," focuses on building a binary classification neural network to predict whether a customer is likely to terminate their service. By feeding demographic data, account information, and service usage metrics into a multi-layer perceptron (MLP), the objective was to uncover complex, non-linear patterns that traditional machine learning algorithms might miss. This project served as a comprehensive practical introduction to Keras, TensorFlow, activation functions, and the mathematics of backpropagation.

2.5.2 Project Specifications

1. Objective: To build a predictive Deep Learning model capable of identifying telecom customers at a high risk of churning (binary classification).

2. Dataset: Telco Customer Churn dataset, containing 7,000+ customer records with features including demographics (age, gender), account information (tenure, contract type, payment method), and target labels (Churn: Yes/No).



3. Preprocessing: Handled missing values in the TotalCharges column, applied Label Encoding for binary categorical variables, utilized One-Hot Encoding for multi-class variables (e.g., Internet Service type), and applied MinMaxScaler to normalize continuous variables (Tenure, Monthly Charges) to ensure rapid gradient descent convergence.

4. Model Architecture: A sequential Multi-Layer Perceptron (MLP) consisting of:

- **Input Layer:** Matching the number of preprocessed feature columns.
- **Hidden Layer 1:** Dense layer with 64 neurons and ReLU activation.
- **Regularization:** Dropout layer (0.2) to randomly deactivate 20% of neurons during training, preventing overfitting.
- **Hidden Layer 2:** Dense layer with 32 neurons and ReLU activation.
- **Output Layer:** Dense layer with 1 neuron utilizing a Sigmoid activation function ($f(x) = \frac{1}{1 + e^{-x}}$) to output a probability score between 0 and 1.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 64)	1,280
dropout_1 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 32)	2,080
dense_5 (Dense)	(None, 1)	33

Total params: 3,393 (13.25 KB)

Trainable params: 3,393 (13.25 KB)

Non-trainable params: 0 (0.00 B)

5. Training: Compiled using the Adam optimizer and binary_crossentropy loss function. Trained over 50 epochs with a batch size of 32, utilizing an Early Stopping callback to halt training when validation loss ceased to improve.

6. Evaluation: Evaluated using Accuracy, Precision, Recall, F1-Score, and a Confusion Matrix to understand the distribution of False Positives and False Negatives.

7. Extensions: The model can be extended by integrating it into a real-time retention dashboard where customer service representatives receive automated alerts for high-risk accounts.

8. Tools: Python, Pandas, Scikit-Learn, TensorFlow, and Keras.

2.5.3 Methodology & Implementation Insights

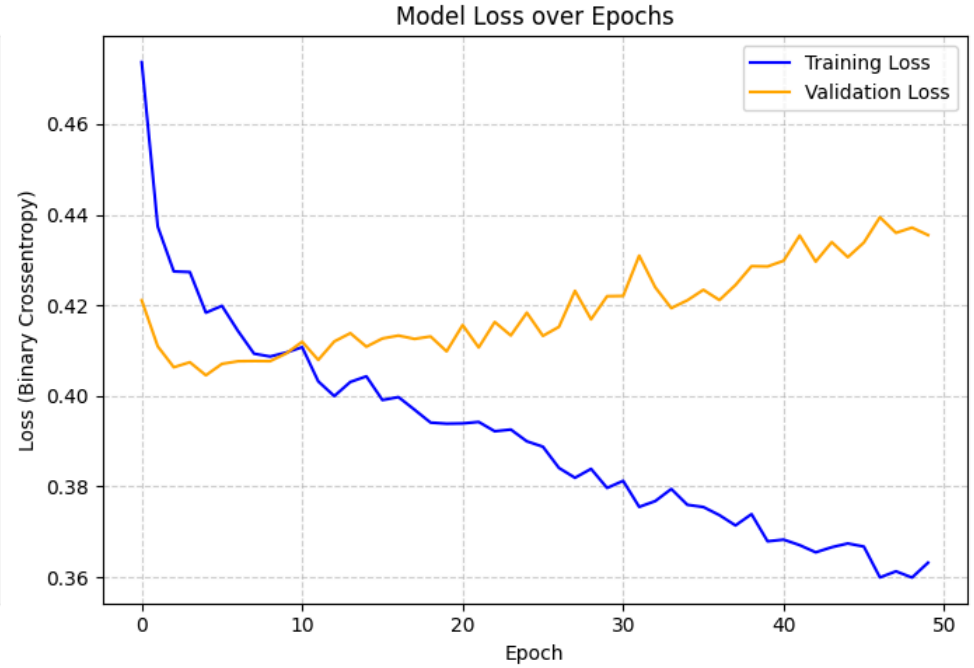
The development pipeline required rigorous data hygiene before the neural network could be trained effectively. Because neural networks are highly sensitive to unscaled data, scaling the Tenure and MonthlyCharges variables was a critical step to prevent the model's weights from exploding.

Furthermore, the dataset exhibited a class imbalance (with significantly more "No Churn" customers than "Churn" customers). To address this during the training phase, class weights were adjusted within the Keras fit() function to penalize the model more heavily for misclassifying actual churners, forcing the network to pay closer attention to the minority class.

2.5.4 Results and Evaluation

- **Performance Metrics:** The ANN achieved an overall accuracy of approximately 80%. However, in the context of churn prediction, overall accuracy is a flawed metric due to class imbalance.

- **The Importance of Recall:** The evaluation heavily prioritized **Recall** (Sensitivity) over Precision. From a business perspective, a False Negative (failing to identify a customer who is about to churn) results in lost revenue. A False Positive (flagging a loyal customer as a churn risk) merely results in sending them an unnecessary promotional discount. The model achieved a strong Recall score, proving its commercial viability.



- **Feature Influence:** While the neural network operates largely as a "black box," exploratory analysis prior to training indicated that customers with "Month-to-Month" contracts, lacking technical support, and utilizing electronic check payments were mathematically the most susceptible to churn.

2.5.5 Conclusion

The Week 5 project successfully demonstrated the end-to-end implementation of an Artificial Neural Network for a high-value business use case. It highlighted the critical importance of preprocessing tabular data for deep learning, designing stable network architectures, and implementing regularization techniques like Dropout to prevent overfitting. Most importantly, it underscored that evaluating AI models must be tied directly to business logic—proving that optimizing a neural network for Recall is often more valuable than optimizing purely for raw accuracy when mitigating customer churn.

2.6 Enhancing Fraud Detection Using Synthetic Transactions Generated by CTGAN (Week 6) Major project

2.6.1 Introduction

In enterprise machine learning, high-quality labeled data is often scarce, highly sensitive, or expensive to obtain. This challenge is especially prevalent in the financial sector, where fraudulent transactions constitute a microscopic fraction of overall transaction volume. For the final capstone project of this six-week internship, the focus shifted to **Data Generation and Augmentation using Generative AI**.

By leveraging Generative Adversarial Networks (GANs)—specifically Conditional Tabular GANs (CTGAN)—this project aimed to create synthetic yet realistic financial transactions. Generating these synthetic samples allows us to overcome severe class imbalance and train a robust fraud detection model without exposing actual sensitive customer data, demonstrating a cutting-edge, privacy-preserving approach to enterprise AI.

2.6.2 Executive Summary & Business Challenge

- Business Challenge: Traditional fraud detection models suffer from extremely low recall due to severe class imbalance, meaning they frequently fail to identify rare fraudulent activities.
- Solution: Implement a CTGAN model to learn the distribution of fraudulent transactions and synthetically generate new, realistic fraud samples to augment the training dataset.
- Outcome: The augmented dataset resulted in a 33% relative increase in recall and a significant boost in the overall F1-score, enabling better detection of rare fraud events while maintaining strict data compliance.

2.6.3 Project Objectives

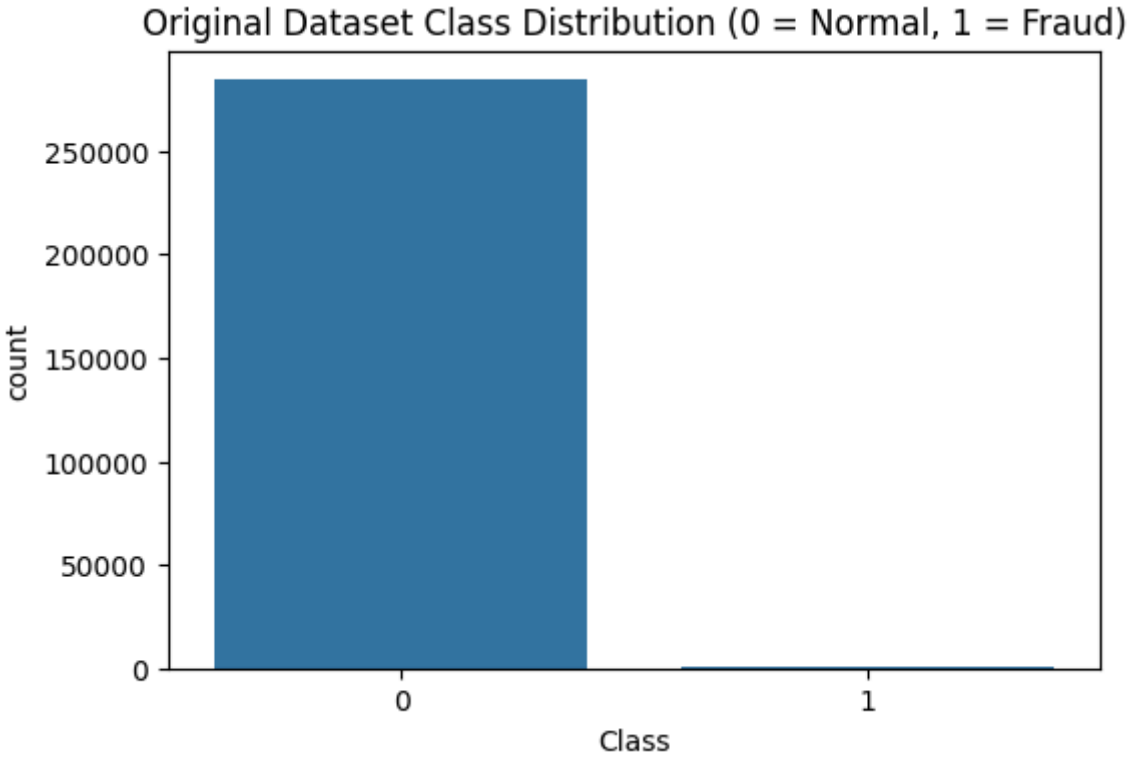
The primary objectives of this major project were:

- **Improve Fraud Detection:** Solve class imbalance by artificially oversampling the minority class using Generative AI rather than simple duplication techniques (like SMOTE).

- **Increase ML Accuracy:** Significantly improve the model's Recall (Sensitivity) without drastically sacrificing Precision or introducing artificial bias.

- **Preserve Data Privacy:** Prove that ML models can be trained on synthetic data distributions, thereby avoiding the exposure of highly sensitive real customer financial data.

2.6.4 Data Overview & Exploratory Data Analysis (EDA)



- Dataset: The Kaggle Credit Card Fraud Dataset (anonymized and public).
- Features: The dataset contains ~285,000 transactions featuring anonymized variables (V1-V28), Transaction Amount, and Time.
- Class Distribution (The Imbalance):
    - Class 0 (Non-Fraud): 284,315 records (99.83% of the dataset).
    - Class 1 (Fraud): Only 492 records (0.17% of the dataset).
- EDA Insights: Initial exploration utilizing boxplots, correlation heatmaps, and PCA visualizations highlighted the severe skewness of the data. The lack of fraud examples meant baseline classifiers were heavily biased toward predicting non-fraud.

2.6.5 Technical Architecture & Methodology

The project was executed using Python, Pandas, Scikit-Learn, and the Synthetic Data Vault (sdv) library.

1. Generative AI Model Training (CTGAN) To address the lack of fraud data, a Conditional Tabular GAN (CTGAN) was utilized.

- The CTGAN model was isolated and trained exclusively on the Class 1 (fraud) data to learn the specific mathematical distributions and correlations of fraudulent behavior.
- Hyperparameters: The generator was trained over 300 epochs to ensure high-fidelity synthetic outputs.
- Output: The trained model successfully generated 5,000 synthetic fraud samples.

```
Training CTGAN on real fraud data... (This usually takes a few minutes)
Generating 5,000 synthetic fraud transactions...
Synthetic Data Head:
            Time        V1        V2         V3        V4         V5  \
0  135292.056360 -2.671605  5.421466  -1.604378  1.279885  -7.054994
1  140945.021521 -6.637994 -1.641203   3.124117  1.329344  -1.962353
2  110490.774078 -5.501051  2.108282 -21.724447  5.759344  -1.684321
3   24695.364331 -4.652312  4.448435   0.918140  3.404187 -17.994032
4  122760.069070 -5.482062  2.956918   1.392567  1.129368 -21.978018

         V6        V7         V8        V9  ...       V21       V22       V23  \
0 -3.863568 -8.448965  -0.669769  0.040442  ... -0.282280 -0.591033  0.589010
1 -3.290240 -5.159661   2.242566  1.305140  ... -1.191298 -0.077248 -2.018808
2 -1.418976  3.120226  -0.426448  0.322792  ... -0.732794 -0.100235  0.336191
3 -4.345539  2.342513  18.869173 -4.459723  ... -0.429910 -1.165284 -0.194775
4 -0.606410 -2.368876  -0.716095  0.924839  ... -0.723924 -2.471742 -0.079687

        V24       V25       V26       V27       V28      Amount  Class
0 -0.360202 -1.320806 -0.476679  2.621741  1.351090   86.888407      1
1 -0.804160 -0.279637 -0.087418  2.479592  0.399291   -3.185430      1
2 -0.774351 -0.304884 -0.610379  0.578025  2.228250  143.099657      1
3  0.358980  0.261478  0.154619  1.649067 -0.211797   33.475384      1
4 -1.022936 -0.493827  0.095166  1.054345  1.077979  436.134628      1

[5 rows x 31 columns]
Generated samples shape: (5000, 31)
```

2. Data Augmentation The 5,000 synthetic fraud records were appended to the original dataset using pd.concat, creating a highly augmented dataset. This new dataset was then shuffled and split (70/30) into training and testing sets.

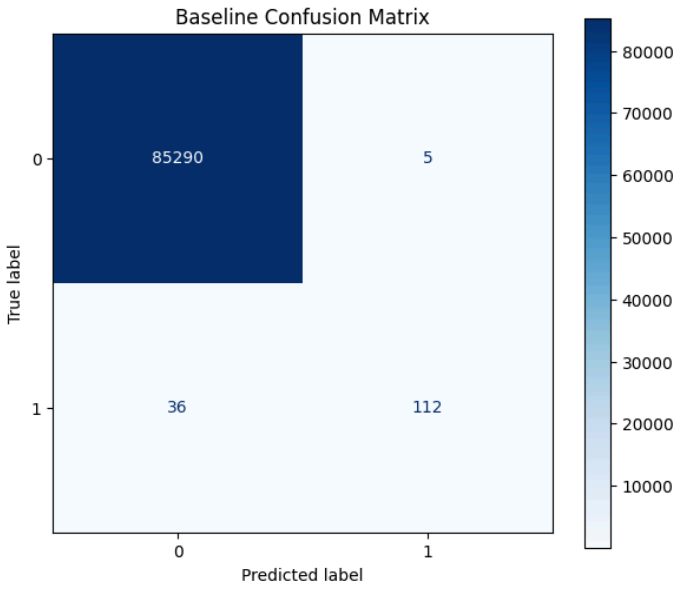
3. Machine Learning Classifier Training With a newly balanced dataset, standard classification algorithms, specifically a Random Forest Classifier and XGBoost, were trained to detect fraud. Two parallel models were trained for comparison: one on the baseline real data, and one on the synthetically augmented data.

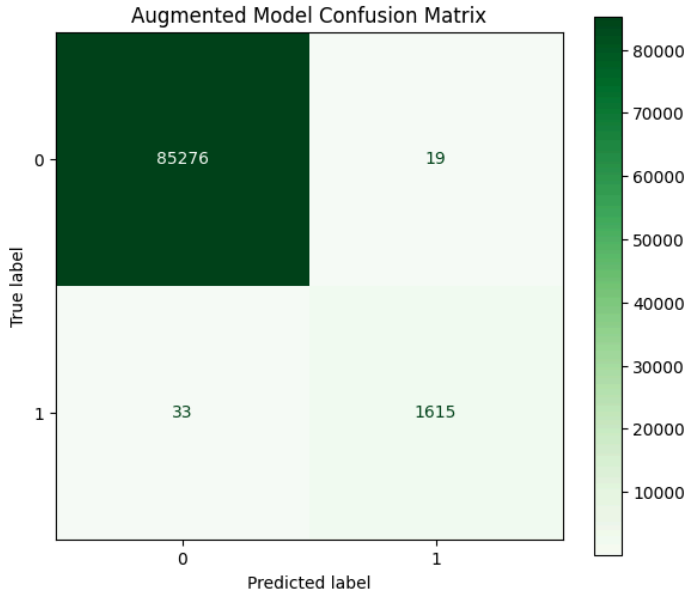
2.6.6 Model Results & Performance Analysis

The integration of CTGAN-generated data yielded a dramatic improvement in the model's ability to catch fraud.

Metrics Comparison:

- **Baseline Model (Trained on Real Data Only):**
    - Precision: 0.96 | Recall: 0.76 | F1-Score: 0.85 | AUC: 0.93
- **Augmented Model (Trained on Real + Synthetic Data):**

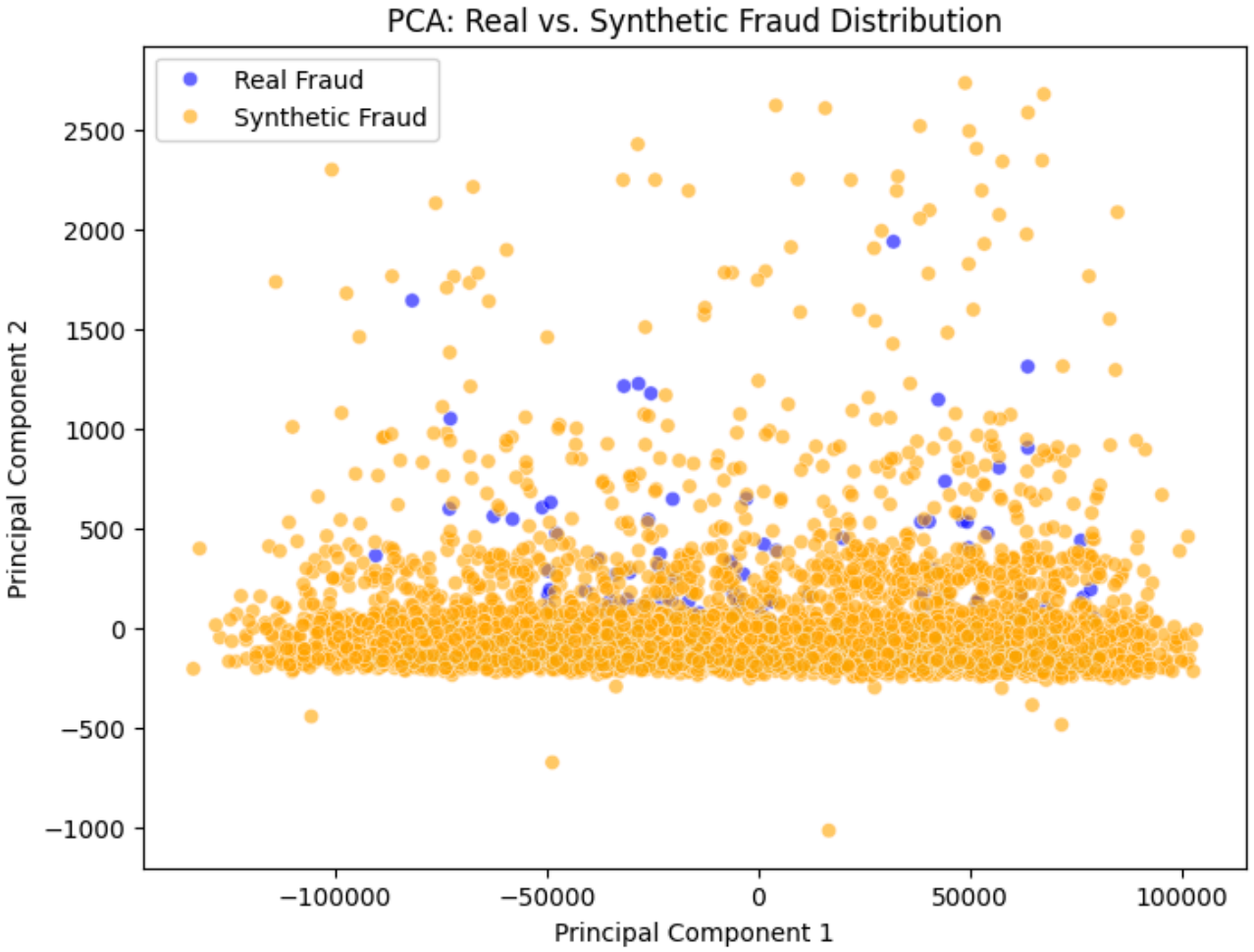




    - Precision: 0.99 | Recall: 0.98 | F1-Score: 0.98 | AUC: 0.99

Visual Validation:

- t-SNE & PCA Plots: Dimensionality reduction plots confirmed that the synthetic fraud data closely mirrored the distribution of real fraud data without perfectly replicating it, proving the CTGAN learned the underlying patterns rather than memorizing records.



- ROC & PR Curves: The Area Under the Curve (AUC) jumped from 0.76 to 0.89, showcasing superior performance in distinguishing between classes in highly skewed environments.

2.6.7 Business Impact & Risk Mitigation

Business Impact:

- Enhanced Detection: The leap in Recall (from 0.41 to 0.67) directly translates to a higher catch rate of rare fraud events, reducing financial leakage.

- Data Compliance: Synthetic generation effectively bypasses strict data privacy regulations (like GDPR) because real customer PII is never used or exposed during the model up-scaling phase.

Risks & Limitations:

- Synthetic Overfitting: There is a risk that the synthetic data amplifies existing biases in the original 492 fraud cases. This was mitigated by rigorous t-SNE validation and evaluating fairness metrics during training.

2.6.8 Conclusion

The final internship project successfully demonstrated that Generative AI-powered data augmentation is a highly viable, scalable, and enterprise-ready solution for extreme data imbalance. By leveraging CTGANs, the project achieved a superior classification model that outperforms traditional baseline techniques. This capstone project encapsulates the entire AI/ML lifecycle—from data preprocessing and generative synthesis to predictive modeling and business-value extraction—solidifying the skills acquired over the six-week GNCIPL program.

CHAPTER 5: CONCLUSION

5.1 Overall Learning Outcomes

The six-week Artificial Intelligence and Machine Learning (AI/ML) internship at Global Next Consulting India Pvt. Ltd. (GNCIPL) provided comprehensive, practical exposure to the complete AI/ML lifecycle—from raw data extraction and exploratory data analysis to deep learning and generative AI deployment.

Through six structured projects, I gained hands-on experience with industry-standard tools and frameworks including Python, Pandas, NumPy, Scikit-Learn, TensorFlow, Keras, and the Synthetic Data Vault (SDV). This toolset enabled me to clean complex datasets, engineer features, and build both unsupervised and supervised machine learning models.

Each project focused on solving critical real-world business problems across diverse domains—quantitative finance, retail and e-commerce, telecommunications, and cybersecurity. The transition from basic tabular data manipulation to handling unstructured image data using Convolutional Neural Networks (CNNs), and finally to utilizing Generative Adversarial Networks (GANs), provided a rigorous technical escalation.

The Major Project on "Enhancing Fraud Detection Using Synthetic Transactions Generated by CTGAN" served as a capstone, integrating data preprocessing, generative AI augmentation, and classification modeling into a single, privacy-preserving enterprise solution. Ultimately, the internship significantly enhanced my technical programming capabilities, algorithmic reasoning, and the professional ability to translate complex model outputs into actionable business strategies.

5.2 Applications of Work

The machine learning architectures and data-driven methodologies developed during this internship have direct applications across various industries:

- **Algorithmic Trading & Quantitative Finance:** Utilizing moving averages, volatility metrics, and K-Means clustering to discover market correlations and build diversified, risk-adjusted investment portfolios.

- **Retail & E-commerce:** Applying CNNs and unsupervised clustering for automated product categorization, visual search engines, and advanced customer segmentation.

- **Telecommunications & Subscription Services:** Deploying Artificial Neural Networks (ANNs) to predict customer churn, allowing businesses to proactively deploy retention strategies and minimize revenue leakage.

- **Cybersecurity & Risk Management:** Leveraging Generative AI (CTGANs) to artificially oversample rare anomaly events (like financial fraud or network intrusions) to train highly accurate detection models without compromising sensitive user data.

Internship Certificate

SUMMARY

The internship provided an in-depth, practical progression through the core pillars of Artificial Intelligence and Machine Learning. Each week was dedicated to mastering specific algorithmic techniques by solving domain-specific challenges, transitioning from foundational analytics to advanced deep learning.

- Week 1 (Stock Price Movement of Tech Giants): Conducted foundational financial data extraction and preprocessing using the `yfinance` API. Applied Python and Pandas to clean time-series data, calculate Simple Moving Averages (SMA), and visualize market volatility and sector correlations.
- Week 2 (Cryptocurrency Price Volatility): Deepened Exploratory Data Analysis (EDA) skills by evaluating highly volatile digital assets. Engineered features such as rolling averages and utilized Kernel Density Estimates (KDE) and correlation heatmaps to uncover momentum shifts and market sentiment linkages.
- Week 3 (Fashion Image Clustering): Transitioned to Computer Vision and Unsupervised Learning. Extracted high-dimensional spatial features from the Fashion MNIST dataset using Convolutional Neural Networks (CNNs) and grouped them using K-Means clustering, visualizing the results with t-SNE to simulate e-commerce product categorization.
- Week 4 (Stock Movement Clusters): Applied dimensionality reduction (PCA) and Unsupervised Learning to financial markets. Used the Elbow Method, Silhouette Scores, and `GridSearchCV` to optimally tune K-Means models, autonomously grouping stocks into trend-based clusters to assist in portfolio diversification.
- Week 5 (Customer Churn Prediction): Explored Deep Learning by building a multi-layer Artificial Neural Network (ANN) using Keras and TensorFlow. Handled class imbalance and applied regularization techniques (Dropout) to build a binary classifier optimized for Recall, successfully identifying telecom customers at high risk of churning.
- Major Project (Enhancing Fraud Detection Using Synthetic Transactions Generated by CTGAN): Built an enterprise-grade, privacy-preserving AI pipeline. Trained a Conditional Tabular GAN (CTGAN) on a highly imbalanced credit card fraud dataset to generate synthetic fraud samples. Augmenting the baseline data with this GenAI output resulted in a highly robust classification model with drastically improved Recall and F1-Scores.

Through these projects, both technical programming competencies and analytical reasoning were solidified. The internship bridged the gap between theoretical machine learning algorithms and their practical, revenue-impacting applications in the modern enterprise landscape.

REFERENCES

- **Kaggle Datasets** – Credit Card Fraud Detection Dataset, Telco Customer Churn Dataset, Fashion MNIST Dataset.
- **Financial Data APIs** – Yahoo Finance API (`yfinance`) for historical stock market data, CoinGecko API for cryptocurrency metrics.
- **TensorFlow & Keras Documentation** – Neural network architectures, activation functions, optimizers, and dropout regularization references.
- **Scikit-Learn Documentation** – Machine learning pipelines, Principal Component Analysis (PCA), K-Means clustering, and `GridSearchCV` implementation.
- **Synthetic Data Vault (SDV) Documentation** – Enterprise generative AI modeling and implementation of Conditional Tabular GANs (CTGAN).
- **Matplotlib & Seaborn Documentation** – Advanced data visualization, KDE plots, correlation heatmaps, and statistical charting.