AI/ML Internship

A Project Report submitted to the

GLOBAL NEXT CONSULTING INDIA PVT LTD

(Six – Week Internship Program)

By

Mansi Adiga

Under the Supervision of

Dr. Anuradha Gupta

(Project Director)

Submitted To:

Global Next Consulting India Pvt. Ltd.

Duration of Internship:

23-March-2026 to 6-May-2026

May 2026

CANDIDATE'S DECLARATION

I hereby declare that the internship report titled, **"AI/ML Internship (GNCIPL) Report"**, submitted as per the requirements for the completion of my six- week internship in Artificial Intelligence and Machine Learning, is a record of my 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 comprises the work completed during the internship period, which includes five mini projects and one major project, undertaken as part of the training program. The work presented in this report does not contain any falsely fabricated ideas, facts or sources. I confirm that this report has not been submitted, either in part or full, for the award of any degree, diploma, or certification.

I also declare that I have adhered to academic integrity and ethical standards throughout the course of this internship.

Mansi Adiga

CERTIFICATE

This is to certify that the project report titled, "**AI/ ML Internship (GNCIPL) Report",** has been carried out by **Mansi Adiga**, a second year Bachelor's in Data Science and Mathematics at Christ University Bangalore Central Campus.

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 would like to express my sincere gratitude to Global Next Consulting India Private Limited (GNCIPL), for giving me the opportunity to undertake the six-week internship in Artificial Intelligence and Machine Learning. This internship has been an invaluable learning experience, allowing me to gain practical exposure to real-world applications of AI/ML concepts.

I would like to extend my gratitude to my supervisor, Ms. Anuradha Gupta, for their guidance and encouragement throughout the internship. Their expertise and constant support played a crucial role in the successful completion of this report.

Finally, I would also like to acknowledge my peers and teachers whose support, guidance and suggestions have been helpful in the course of this internship.

Mansi Adiga

ABSTRACT

This report presents the work completed during the six- week internship in Artificial Intelligence and Machine Learning at Global Next Consulting India Pvt. Ltd., Noida. The internship was structured to provide a balance of theoretical understanding and hands-on implementation through daily sessions and the completion of five mini projects, followed by a comprehensive major project in the final week.

The mini projects focused on fundamental concepts such as data preprocessing, exploratory data analysis (EDA), feature engineering, dimensionality reduction techniques, hyperparameter tuning, and the implementation of core machine learning algorithms including classification, clustering, and other techniques, with the help of tools such as, Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn. The internship further covered neural networks and introductory deep learning concepts, including perceptrons, activation functions, and backpropagation, with practical implementation of Artificial Neural Network (ANN), Softmax activation, etc, using frameworks such as TensorFlow and Keras. The final major project integrated the knowledge gained from the earlier projects and involved generation and augmentation using Generative AI.

The internship projects and daily sessions strengthened my skills in Python, Machine Learning, Data Visualization, Supervised and Unsupervised Learning models, Neural Networks and Deep Learning, while also improving my presentation skills, analytical thinking and problem-solving approach. The experience gained through this internship serves as a strong foundation for further exploration and development in the field of Artificial Intelligence and Machine Learning.

INDEX

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 protect businesses from the rapidly evolving landscape of digital threats. With the objective of delivering proactive and comprehensive security solutions, the company specializes in safeguarding critical digital assets, sensitive data and organizational infrastructure. GNCIPL serves a diverse range of industries, including finance, healthcare, manufacturing, and technology, by offering customized cybersecurity strategies tailored to specific business requirements. The company also provides compliance support aligned with global standards such as GDPR, HIPAA and PCI-DSS.

Driven by core values of integrity, innovation, customer-centricity, excellence, and collaboration, GNCIPL is committed to delivering high-quality, reliable services. With a strong focus on innovation and client satisfaction, GNCIPL aims to establish itself as a global leader in cybersecurity while enabling organizations to operate securely and confidently in the digital age.

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

The primary objectives of the six-week AI/ML internship were as follows:

- To build a strong foundation in Python programming, mathematical and statistical concepts which are essential for machine learning algorithms.
- To develop proficiency in data handling and preprocessing techniques, which includes data cleaning, handling missing values, feature engineering, and transformations using NumPy and Pandas libraries.
- To gain practical experience in data visualization and exploratory data analysis (EDA) using Matplotlib and Seaborn.
- To understand and implement supervised and unsupervised machine learning algorithms, which includes regression, classification, and clustering algorithms, along with model evaluation using metrics such as accuracy, precision, recall and confusion matrix.
- To learn dimensionality reduction techniques such as Principal Component Analysis (PCA) as well as model optimization techniques such as hyperparameter tuning to improve model performance.
- To gain introductory knowledge of neural networks and deep learning, including perceptrons, activation functions, and backpropagation, with practical implementation using frameworks like TensorFlow and Keras.
- To understand the basics of deploying machine learning models using tools such as Flask or Streamlit for real-world application.
- To develop an end to end machine learning solution through a capstone project, using Generative AI techniques such as Conditional Tabular GAN (CTGAN), for augmentation and generation of synthetic data to address class imbalance and improve the generalization of the capstone machine learning model.

2. Mini Projects

2.1 Week 1 Project: Bank Loan Approval Patterns Analysis (EDA Project)

2.1.1 Introduction

In the modern financial system, loan approval is a critical process that directly impacts both banking institutions and applicants. Financial institutions must carefully evaluate each loan application to ensure that the borrower is capable of repayment while minimizing the risk of default. This decision-making process is influenced by multiple factors, including an applicant's income, credit score, financial history, and demographic characteristics.

This project focuses on exploring and analysing loan approval data to understand how different factors affect the outcome of loan applications. By using data visualization techniques, the project aims to uncover relationships between applicant attributes and loan approval status. The insights gained from this analysis can help in understanding the key factors that drive lending decisions and demonstrate the importance of data analytics in the financial domain.

2.1.2 Objective

The main project objectives are:

- To analyse loan approval patterns using applicant demographic and financial data.
- To identify the key factors influencing loan approval decisions, such as income, credit score, and previous loan defaults.
- To study the relationship between applicant characteristics and loan approval status through data visualization techniques.
- To use correlation analysis and heatmaps to understand relationships between numerical variables in the dataset.
- To derive meaningful insights from the dataset that can support better decision-making in the loan approval process.

2.1.3 Dataset Description

The Kaggle dataset, 'loan_data.csv', contains the data used for binary classification on loan approval. It contains information related to loan applicants, including demographic details, financial attributes, and credit-related variables. The dataset has 45000 rows and 14 columns. The description of each variable in the dataset is given below:

- person_age (float): Age of the person
- person_gender (categorical): Gender of the person
- person_education (categorical): Highest education level
- person_income (float): Annual income
- person_emp_exp (integer): Years of employment experience
- person_home_ownership (categorical): Home ownership status (e.g., rent, own, mortgage)
- loan_amnt (float): Loan amount requested
- loan_intent (categorical): Purpose of the loan
- loan_int_rate (float): Loan interest rate
- loan_percent_income (float): Loan amount as a percentage of annual income
- cb_person_cred_hist_length(float): Length of credit history in years
- credit_score (integer): Credit score of the person
- previous_loan_defaults_on_file (categorical): Indicator of previous loan defaults
- loan_status (target variable) (integer): Loan approval status: 1 = approved; 0 = rejected

2.1.4 Tools and Technologies Used

The tools used in the project are:

- Python (primary programming language)
- Pandas (data loading, data manipulation, data cleaning)
- Matplotlib (data visualization)
- Seaborn (advanced statistical visualizations)

2.1.5 Methodology and Interpretation

I. Data Understanding

The dataset, ‘loan_data.csv was imported using the Pandas library and explored to gain an understanding of its structure, dimensions, variable types, and overall composition. The functions used are:

- .head(): To display first five rows of the dataset
- .shape: To get the total number of rows and columns in the dataset
- .dtypes: To get the datatypes of each column in the dataset
- .info(): To get a technical summary of the dataset (datatypes, non-null counts, and memory usage)
- .columns.tolist(): To get a list of columns in the dataset
- .describe(): To get a statistical summary of all numerical columns in the dataset

II. Data Wrangling

Basic data cleaning steps were performed, including:

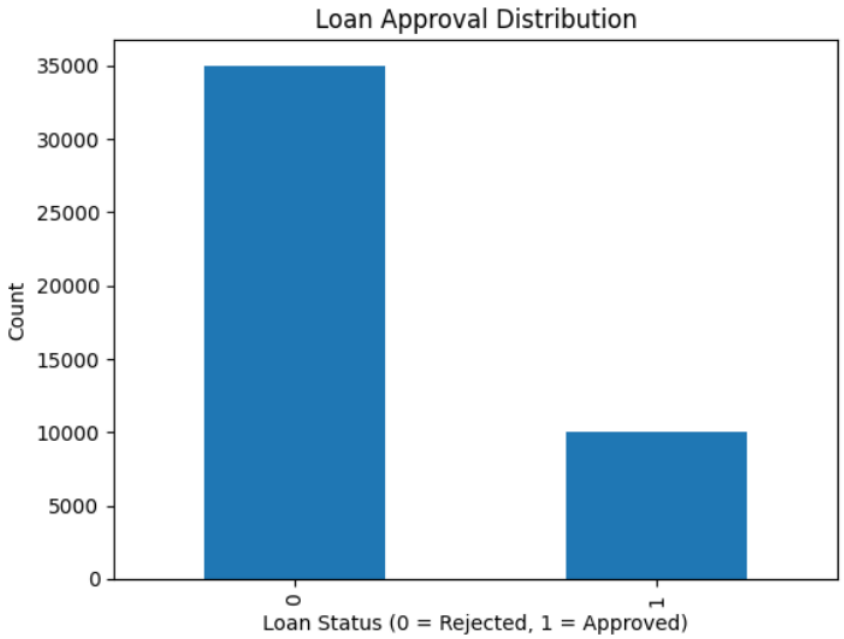
- Checking for missing values: no missing values were found
- Checking for duplicate values: no duplicate values were found
- Checking the number of unique values
- Verifying data types: the data types were appropriate

Hence, the dataset was cleaned and ready for data visualization.

III. Data Visualization and Analysis

Various visualizations were created to understand relationships between variables and loan approval status. The visualisations and the insights for each are:

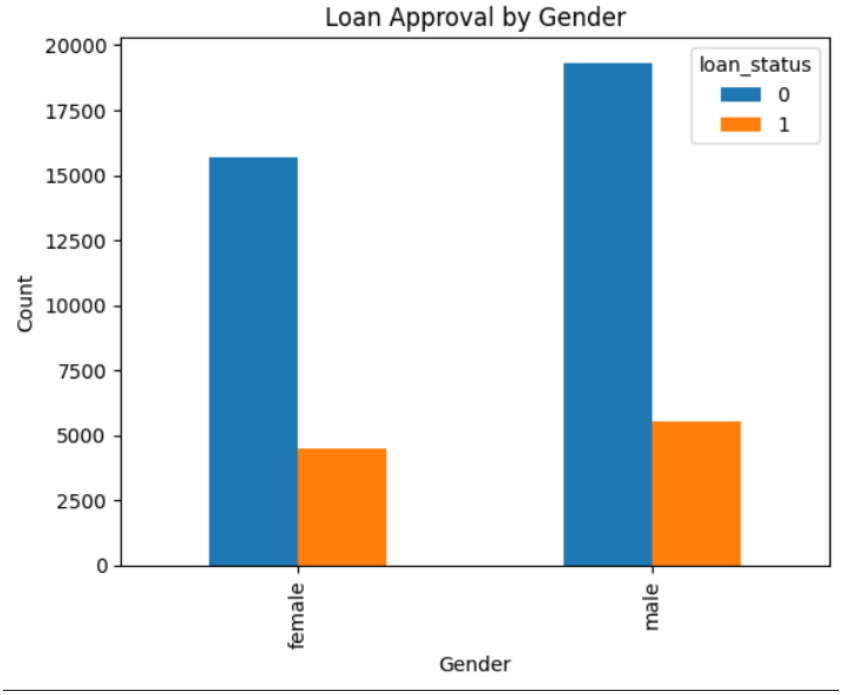
- **Chart-1: Loan Approval Distribution**



Insights:

- The above bar graph shows the overall distribution of approved and rejected loans.
- The number of rejected loans is around 35000 while the number of approved loans is around 10000.
- This shows that there is a huge imbalance between rejected loans and approved loans, indicating that a large number of applicants fail to meet loan approval criteria.

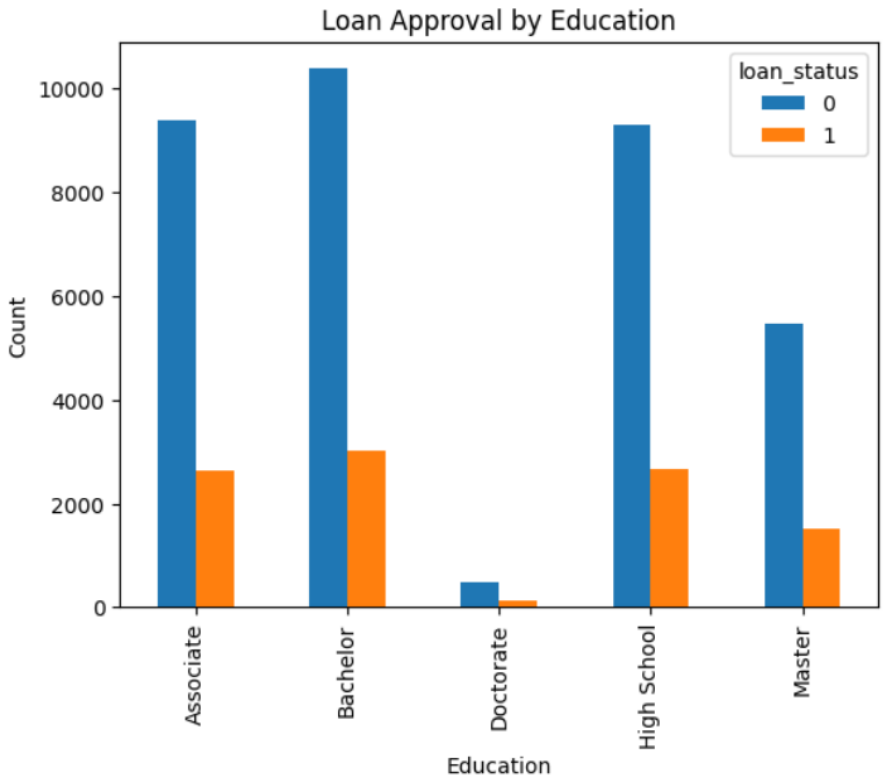
- **Chart-2: Loan Approval by Gender**



Insights:

- This chart compares loan approval across genders (male and female).
- In both genders, the number of rejected applications(blue) are significantly higher than the approved applications(orange).
- The proportion of approved applications relative to total application is almost similar in both genders which shows that gender does not have a strong impact on loan approval outcomes.

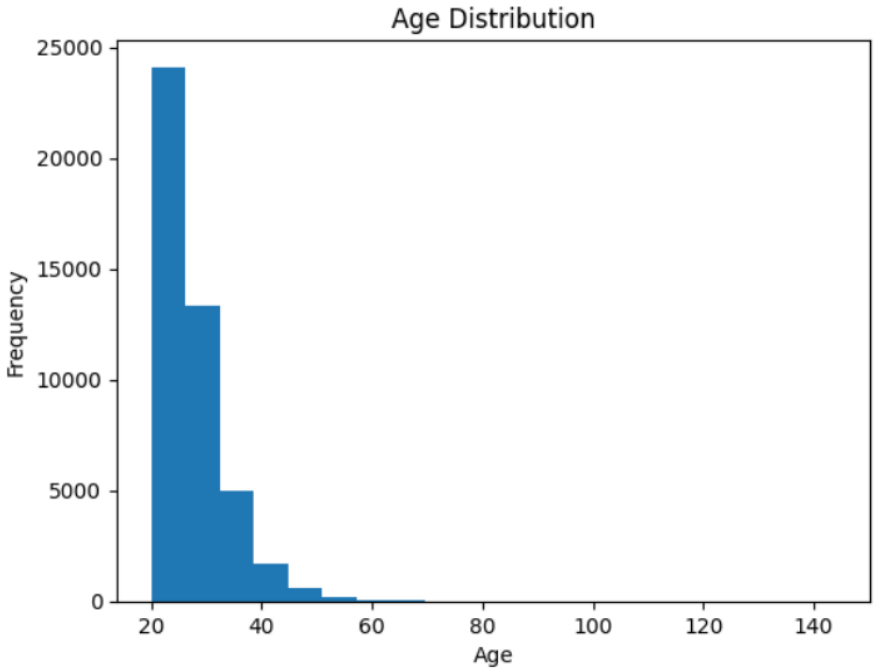
- **Chart-3: Loan Approval by Education**



Insights:

- This chart compares loan approval across education levels.
- Through all education levels, the number of rejected applications is higher than the approved applications.
- Most loan applications are from applicants with Bachelor's, Associate's, and High School education levels. However, education level does not appear to have a significant impact on loan approval outcomes.

- **Chart-4: Age Distribution**



Insights:

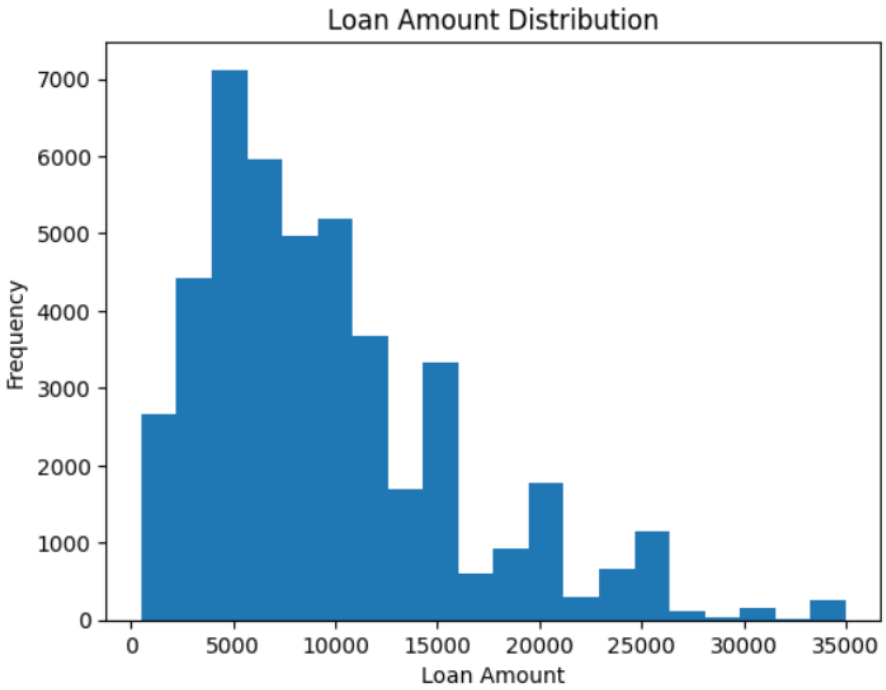
- The above histogram shows the age distribution in the dataset.
- Most applicants are young adults in their 20s.
- The number of applicants subsequently decreases as the age increases.

- **Chart-5: Loan Approval by Credit Score Group**

Insights:

- The above graph compares credit scores and loan approval rates.
- Applicants with credit scores in the range of 574 to 666 show the highest number of both approvals and rejections, indicating that the majority of applicants fall within this credit score range.
- Higher credit score groups show more approval rates than low credit score groups.
- This shows that credit scores are an important factor for loan approval rates.

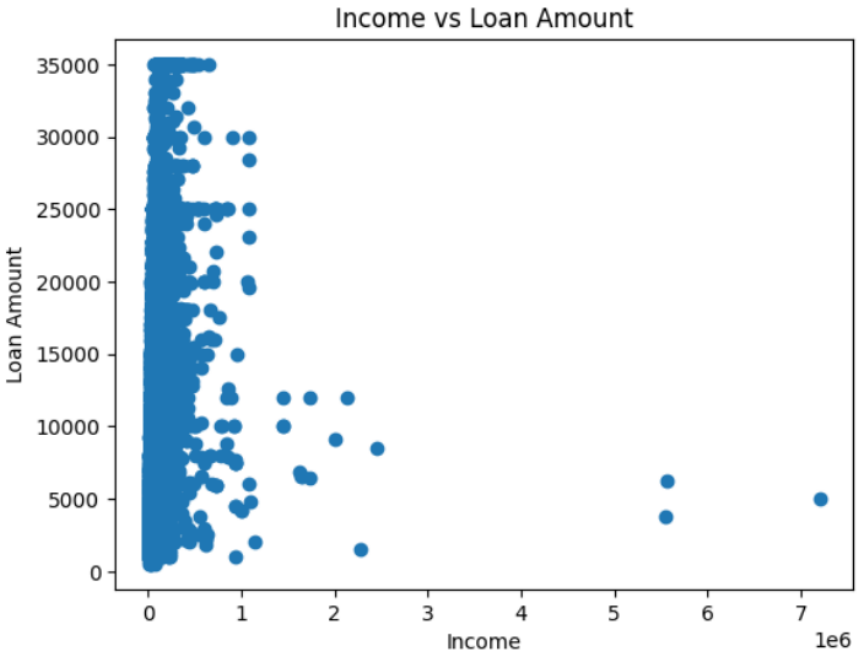
- **Chart-6: Loan Amount Distribution**



Insights:

- The above histogram shows the distribution of loan amount in the dataset.
- The highest frequency of loan amount is 7000. While there are peaks at 20000, 25000, 30000 and 35000, from the graph it can be inferred that loan amounts under 12000 are most common.

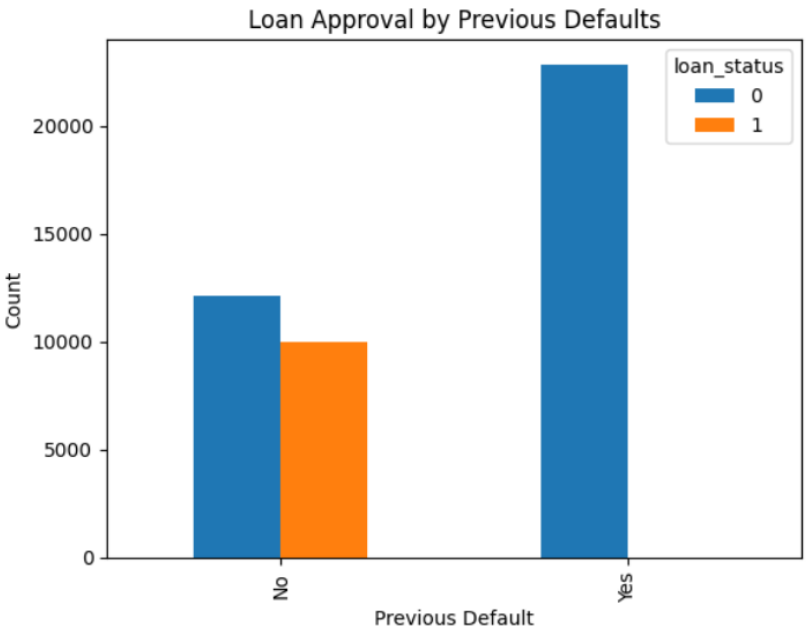
- **Chart-7: Income vs Loan Amount**



Insights:

- The above scatter plot compares loan amount and income.
- Most applicants have low income but loan amounts vary widely in that group.
- A few high-income outliers exist but they don't take large loans.
- This shows that there is no correlation between income and loan amount.

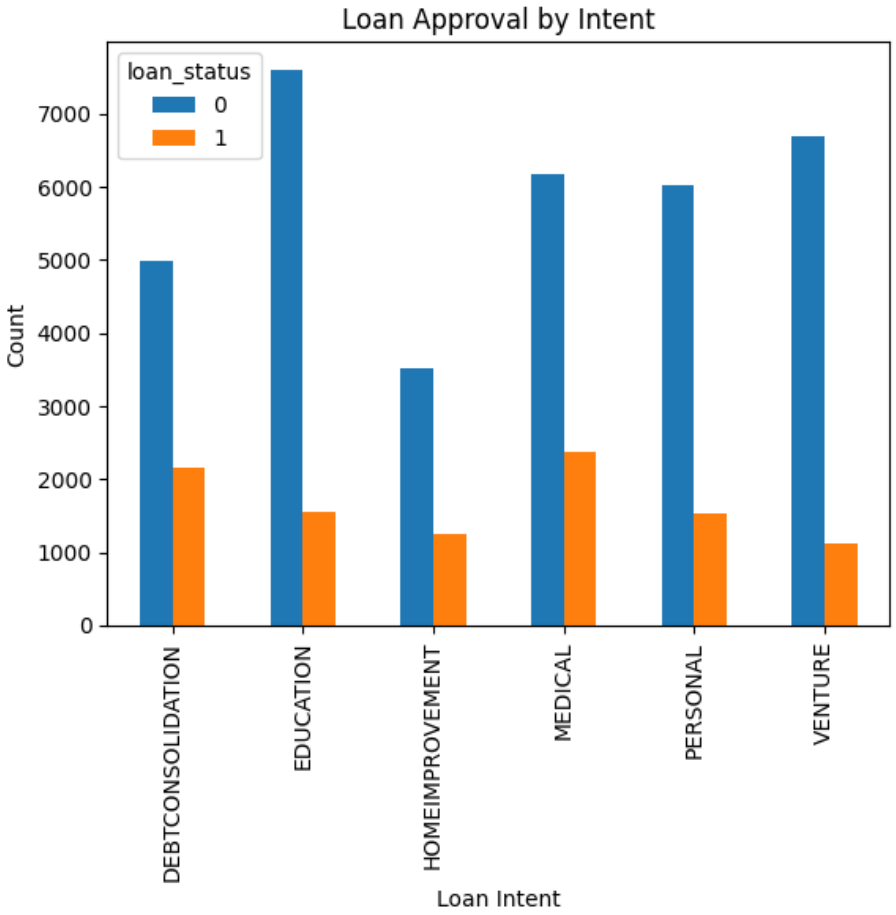
- **Chart-8: Loan Approval by Previous Defaults**



Insights:

- The above chart shows loan approval based on previous default.
- Applicants with no previous defaults have both rejected and approved applications whereas applicants with previous defaults have only rejected applications.
- This shows that previous loan defaults highly influence loan approval outcomes.

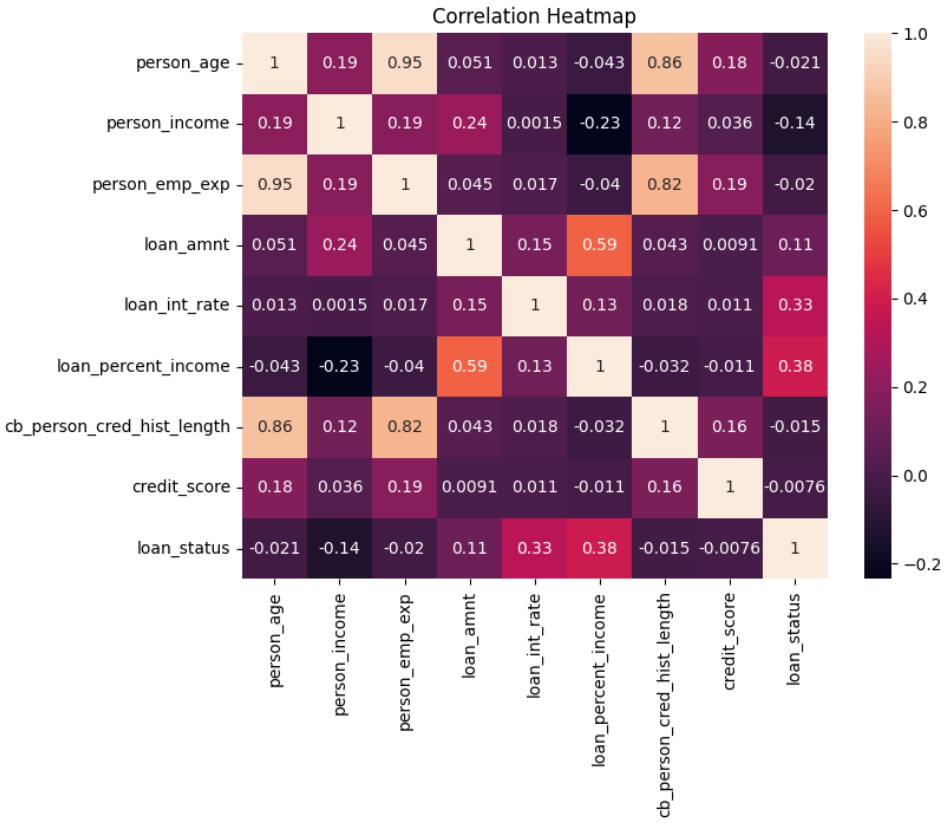
- **Chart-9: Loan Approval by Intent**



Insights:

- The above chart shows the loan approval rates across different loan intents.
- Rejected application are significantly higher than approved applications across all intents.
- Medical and Debt consolidation intents have relatively high approvals compared to others, whereas Education and Venture show high rejected applications.
- This shows that loan intents influence approval outcomes.

- **Chart-10: Correlation Heatmap**



Insights:

- The above heatmap shows the correlation between the attributes in the dataset.
- The attributes that have very strong correlation are person_emp_exp and person_age.
- Income has weak correlations with most variables.
- The most important factors affecting loan status are loan_percent_income and loan_int_rate.

2.1.6 Conclusion

This project helps in understanding how different factors affect loan approval decisions. The analysis indicates that the most important factors influencing loan approval are higher credit scores, the absence of previous loan defaults, and lower loan-to-income ratios. Applicants with a history of defaults are far more likely to face rejection, even if other financial factors are favourable. Overall, the loan approval process appears to prioritize financial stability and repayment capability, while maintaining strict evaluation standards to reduce the risk of default.

These insights can help banks make better decisions and improve their approval process. Thus, the project successfully identified key variables affecting loan approval and demonstrated the use of data analysis techniques to extract meaningful insights from real-world data.

2.2 Week 2 Project: Global Inflation Trends Analysis (EDA Project)

2.2.1 Introduction

The global economy is highly complex, and understanding economic trends and patterns is crucial for making informed decisions about investments, policies, and more. One key factor that impacts the economy is inflation, which refers to the rate at which prices increase over time.

Inflation is one of the most significant macroeconomic indicators affecting economic growth, purchasing power, and financial stability. This project focuses on analysing global inflation trends across multiple countries over the period 1970–2022 using Exploratory Data Analysis (EDA) techniques.

2.2.2 Objective

The main objectives of this project are:

- To analyse global inflation trends over time
- To compare inflation patterns across countries
- To identify periods of economic instability and inflation spikes
- To perform correlation and distribution analysis on inflation data
- To detect outliers and anomalies in inflation patterns
- To visualize inflation geographically using geospatial techniques

2.2.3 Dataset Description

The Kaggle dataset, “Global Price Inflation” dataset provides a collection of inflation rates across 206 countries from 1970 to 2022, covering four critical sectors of the economy. The dataset has 783 rows and 64 columns, where 6 columns have categorical data and 58 columns have numerical data.

The description of the variables in the dataset are:

- Country Code: A unique code assigned to each country in the dataset

- IMF Country Code: The three-letter code assigned by the International Monetary Fund (IMF) to each country
- Country: The name of the country
- Indicator Type: The type of inflation indicator (energy, food, consumer, producer)
- Series Name: The name of the specific inflation series
- 1970-2022: Inflation rates for each year, provided in monthly, quarterly, or annual intervals depending on the country and series
- Note: Any additional notes or context for the data, such as sources or explanations for any outliers or gaps in the data.

The data can be used to gain insights into the complex factors that impact the economy and make informed decisions about investments, policies, and more. Since the dataset was originally structured in a wide format, extensive preprocessing and transformation were performed to make it suitable for analysis.

2.2.4 Tools and Technologies Used

The tools used for this project include:

- Python (primary programming language)
- Pandas (data loading, data manipulation, data cleaning)
- Missingno (missing values visualization)
- Matplotlib (data visualization)
- Seaborn (advanced statistical visualizations)
- Plotly Express (data visualization)

2.2.5 Methodology and Interpretation

I. Data Understanding

The dataset, 'Global Dataset of Inflation.csv', was imported using the Pandas library and explored to gain an understanding of its structure, dimensions, variable types, and overall composition. The functions used are:

- .head(): To display first five rows of the dataset
- .shape: To get the total number of rows and columns in the dataset
- .dtypes: To get the datatypes of each column in the dataset
- .info(): To get a technical summary of the dataset (datatypes, non-null counts, and memory usage)
- .columns.tolist(): To get a list of columns in the dataset
- .describe(): To get a statistical summary of all numerical columns in the dataset

II. Data Wrangling

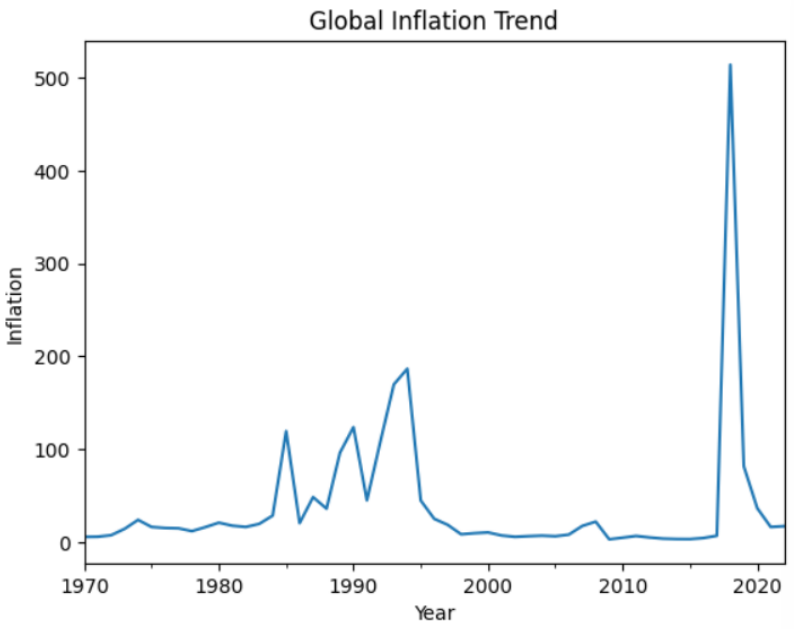
The following cleaning steps were performed for analysis:

- Checked for missing values: many missing values were found
- Visualized the missing values using missingno library
- Checked for duplicate values: no duplicate values were found
- Removed unnecessary unnamed columns
- Converted the dataset from wide format to long format using the melt function
- Converted year values into datetime format
- Converted inflation values into numeric format
- Removed rows containing missing inflation values
- Filtered countries with insufficient data availability
- Sorted the dataset

These steps ensured that the dataset was clean, structured, and suitable for exploratory analysis.

III. Exploratory Data Analysis (EDA)

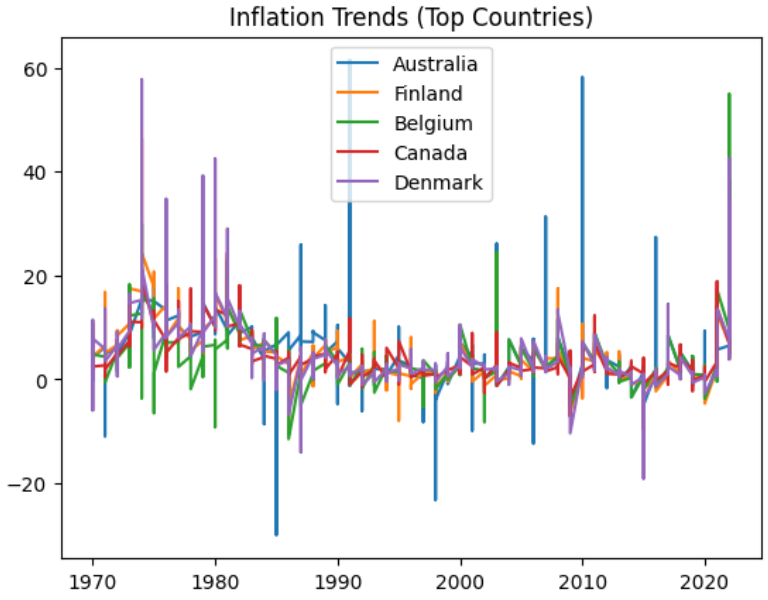
- **Chart-1: Global Inflation by Year**



Insights:

- This time series chart gives the global inflation trend over time (1970-2022).
- From the 1970s to 1980s inflation appears to be stable showing controlled economic conditions globally.
- In the late 1980s or early 1990s there appears to be a sudden rise in inflation due to some major economic disturbance.
- After the spike, inflation drops significantly and remains low and stable till the early 2020s showing that better monetary policies were implemented.
- Then there is a visible rise in inflation near 2018-2022 because global disruptions or pandemic related economic effects.

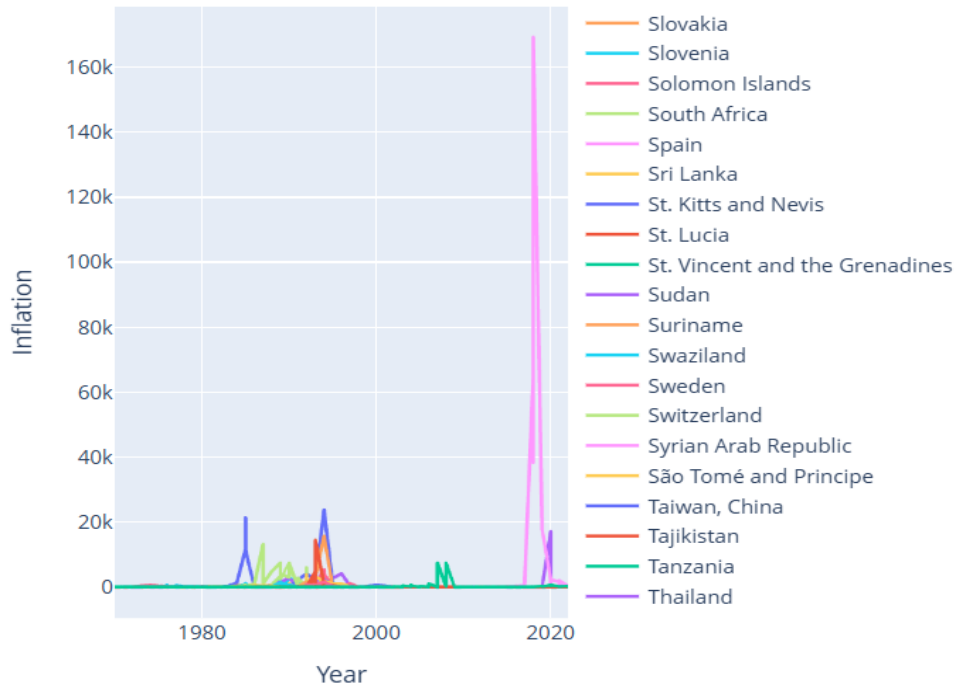
- **Chart-2: Inflation trends (Top 5 countries)**



Insights:

- The above time series chart shows the inflation trends in top 5 countries, namely Australia, Finland, Belgium, Canada and Denmark, over time (1970-2022).
- From 1970s to 1980s, Denmark(purple) seems to have the highest inflation but then comes down to near 0 in the following years. The inflation becomes negative at 2015 but increases again in 2022.
- The inflation in Australia varies significantly through the years. It is near 0 from 1970-1985 after which it has negative inflation and soon high inflation in the 1990s. It drops again in the 2000s and increases in 2010, after which it comes down to near 0 and stabilizes.
- Belgium's inflation seems to be just a bit more or less 0 throughout the years, except in the mid 2000s and 2022 where it seems to have increased.
- Finland and Canada have a stable inflation trend throughout the years where its mostly near zero.

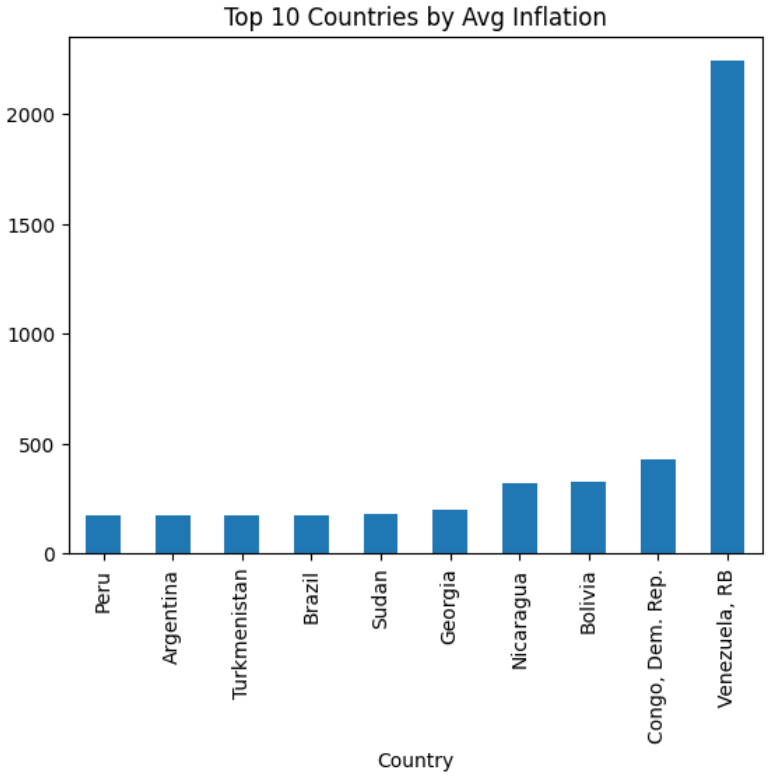
- **Chart-3: Inflation by Year (using Plotly)**



Insights:

- The above chart is an interactive time series chart showing the inflation through the years in various countries segregated by colour.
- The chart shows extreme inflation in Venezuela, RB in the year 2018 with the highest inflation of 169.2018k.
- In the years 1980-1995, the chart shows inflation clusters, with peaks reaching near 20k especially in the areas, Bolivia in 1984 and Congo, Dem. Rep in 1994. This shows there were economic crisis during that time in many countries.
- Despite the outliers, most of the countries maintain a stable inflation rate.

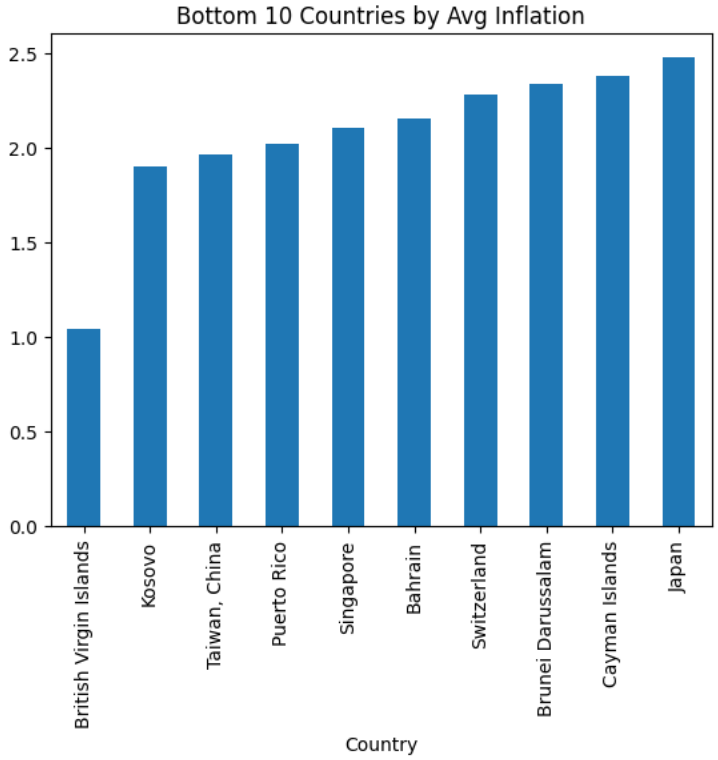
- **Chart-4: Average Inflation by Top 10 Countries**



Insights:

- The above chart compares the top 10 countries by their average inflation. Each bar gives the average inflation of that country, displayed in ascending order.
- In this case, the top 10 countries namely, Peru, Argentina, Turkmenistan, Brazil, Sudan, Georgia, Nicaragua, Bolivia, Congo, Dem. Rep and Venezuela, RB are compared.
- Venezuela shows the highest average inflation compared to the rest of the countries, followed by Congo, Dem. Rep and Bolivia.

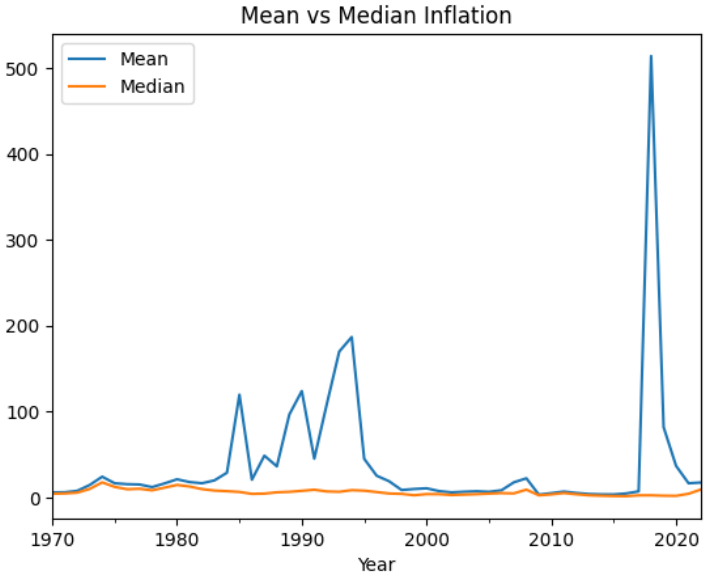
- **Chart-5: Average Inflation by Bottom 10 Countries**



Insights:

- The above chart compares the average inflation over the years among the bottom 10 countries, displayed in ascending order of inflation.
- Through the graph we can infer that, among the other, Japan has the highest inflation, of about 2.4.
- Japan is followed by, Cayman Islands, Brunei Darussalam and Switzerland.

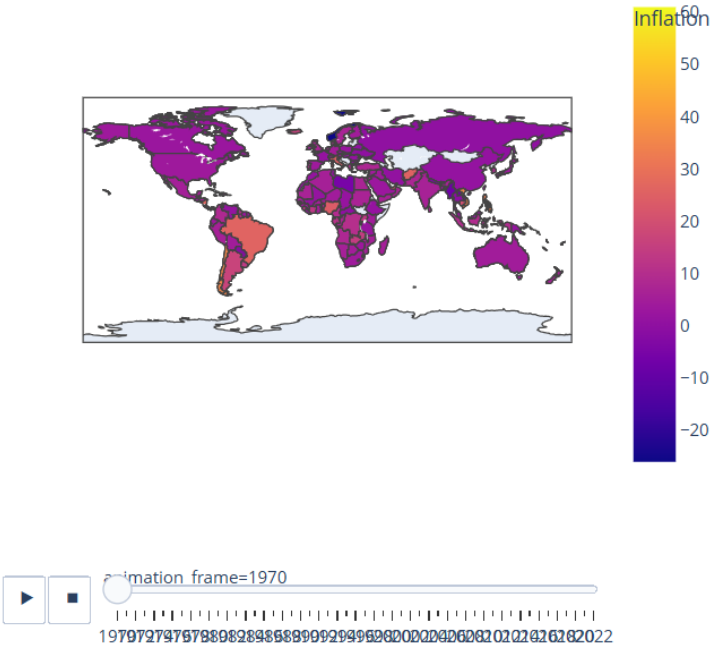
- **Chart-6: Mean vs Median Inflation Over The Years**



Insights:

- The above chart compares the mean and median inflation over time.
- Through the graph we can infer that there is a massive gap between the mean (blue line) and median (orange line). This shows that the global inflation data is highly right-skewed.
- The median line remains flat and near 0 throughout the years, while the mean inflation peaks at certain times, around 1985-1995 and significantly in the late 2010s.

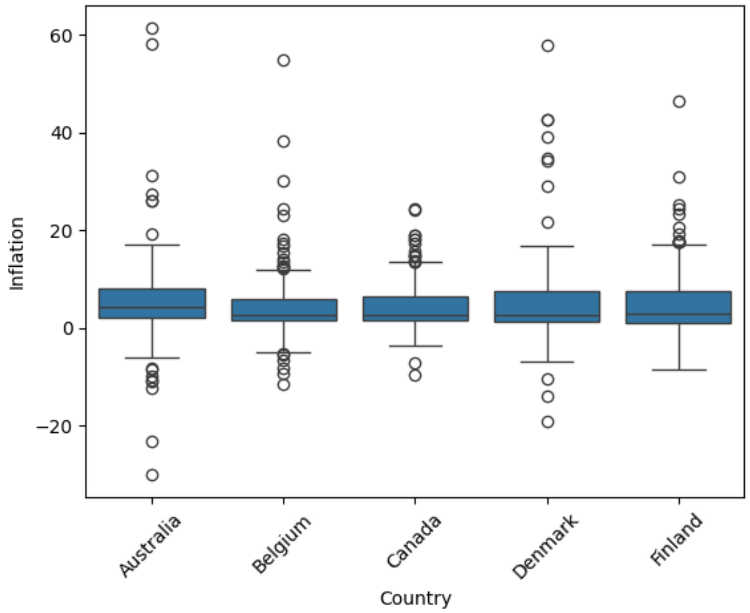
- **Chart-7: Geospatial Visualization**



Insights:

- The above geospatial visualization (chloropleth map) shows the inflation in each country over the years.
- The colour shows the inflation rate, where warmer colours show high positive inflation and cooler colours show zero to negative inflation.
- By using the animation slider, we can see how inflation rates change in each country.

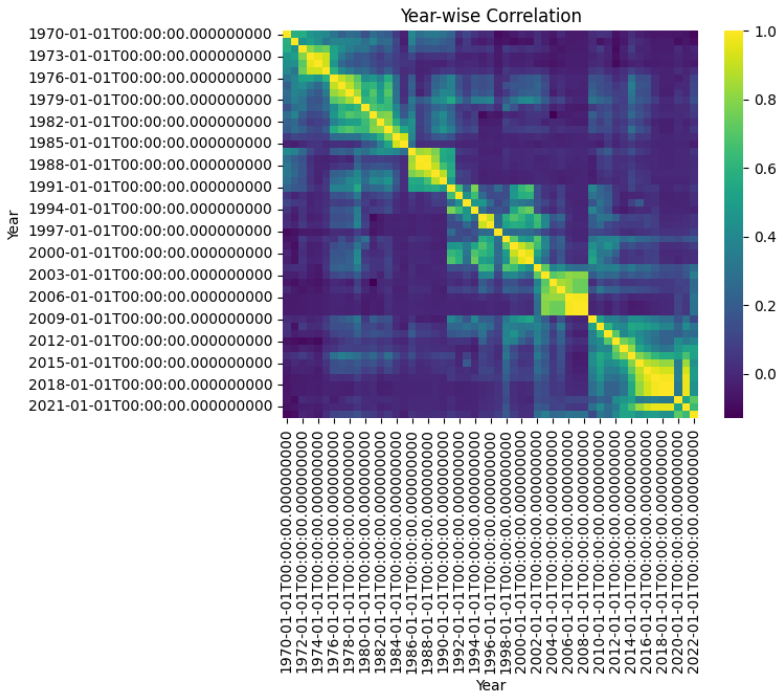
- **Chart-8: Boxplot Showing Inflation by Top 5 Countries**



Insights:

- The above box plot gives the inflation for the countries, Australia, Belgium, Canada, Denmark and Finland.
- The median inflation rate is consistently low, i.e., around 2-5%. This shows the long-term economic stability of the countries.
- All countries show a lot of outliers which show that while the inflation is usually stable, there have been times where the countries have experienced extreme peaks (low and high).

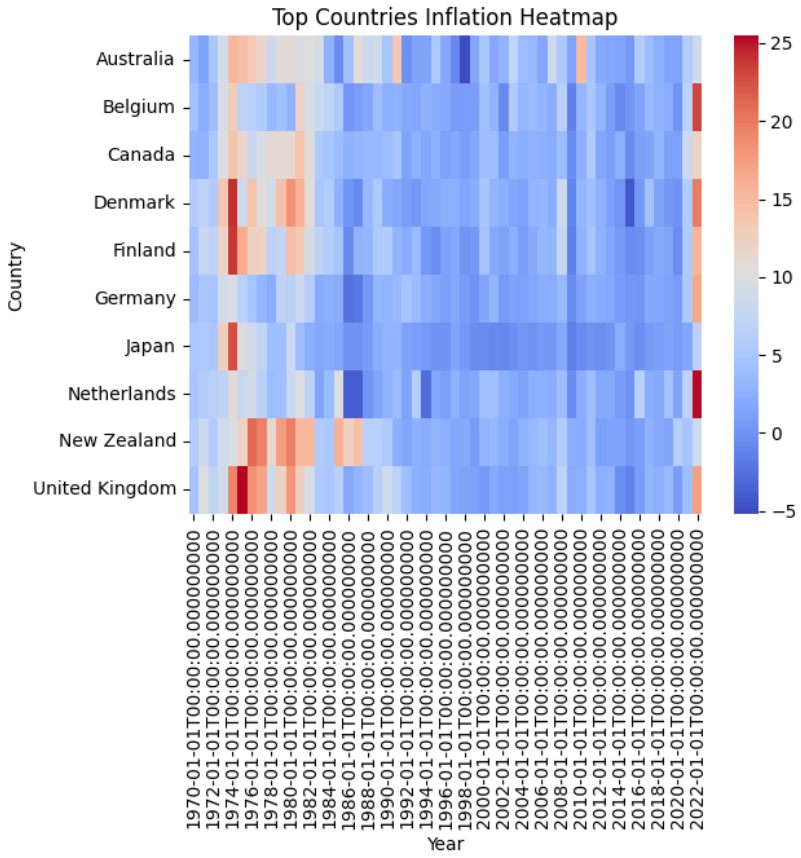
- **Chart-9: Year Wise Correlation**



Insights:

- The above heatmap shows the year wise correlation. The more the yellow, the more is the correlation and the more the purple the less is the correlation.
- The bright yellow diagonal shows each year's correlation with itself (1.0). The green/yellow areas around this diagonal indicates that the inflation pattern across the world are highly similar in the adjacent years.
- The purple areas in the far corners shows very low correlation, meaning the global inflation today is different from what it was 50 years ago.

- **Chart-10: Top Countries Inflation Heatmap**



Insights:

- The above heatmap gives the top 10 countries' inflation, where red shows high inflation and blue shows low inflation.
- From the 1970s-1980s, there are warm blocks on the left side of the chart, which shows high inflation of around 15-25% during that time.
- From 1990s-2020s the heatmap shows mostly blue bands, showing low and stable inflation among the countries.
- At 2022, there is a sudden shift to red bands, showing high inflation mostly due to the COVID-19 pandemic.

2.2.6 Conclusion

This project successfully analysed global inflation patterns using exploratory data analysis techniques. The study provided insights into historical inflation trends, economic cycles, country-level inflation variability, and major inflation anomalies. Through preprocessing, visualization, and

comparative analysis, the project demonstrated how EDA can be effectively used to understand complex macroeconomic data and derive meaningful insights from large time-series datasets. These insights can serve as a foundation for further predictive modelling or policy- oriented analysis.

2.3 Week 3 Project: Age vs Spending Cluster Analysis (Customer Segmentation Project)

2.3.1 Introduction

Customer segmentation is an important technique used by businesses to understand customer behaviour and improve marketing strategies. Different customers have different spending habits, preferences, and purchasing patterns. Identifying these patterns helps organizations target customers more effectively and improve customer satisfaction.

This project focuses on segmenting mall customers based on their age and spending behaviour using the K-Means clustering algorithm. By grouping similar customers together, the project helps identify valuable customer segments and provides useful business insights.

2.3.2 Objective

The main objectives of this project are:

- Analyse customer behaviour based on age and spending patterns
- Identify distinct customer groups using clustering
- Support better decision-making through customer segmentation

2.3.3 Dataset Description

The Kaggle dataset, “The Mall Customers Dataset”, contains basic customer information like ID, gender, age, education, marital status, annual income and spending score. There are 200 rows and 7 columns. Customer ID, Age, Annual income and Spending Score are numeric data whereas Gender, Education and Marital Status are categorical data. The description of each attribute in the dataset is as follows:

- CustomerID: Unique identifier for each customer

- Gender: Gender of the customer (Male/Female)
- Age: Age of the customer
- Education: Customer's education level
- Marital Status: Marital status of the customer
- Annual Income (k$): Annual income in thousand dollars
- Spending Score (1-100): Score assigned based on spending behaviour

2.3.4 Tools and Technologies Used

The tools and libraries used in the project are:

- Python (primary programming language)
- Pandas (data loading, data manipulation, data cleaning, data preprocessing)
- Scikit- learn (Feature scaling, KMeans model training, and silhouette score for evaluation)

2.3.5 Methodology

I. Data Understanding

The dataset, 'Mall Customers.xlsx', was imported using the Pandas library and explored to gain an understanding of its structure, dimensions, variable types, and overall composition. The functions used are:

- .head(): To display first five rows of the dataset
- .shape: To get the total number of rows and columns in the dataset
- .dtypes: To get the datatypes of each column in the dataset
- .info(): To get a technical summary of the dataset (datatypes, non-null counts, and memory usage)
- .columns.tolist(): To get a list of columns in the dataset
- .describe(): To get a statistical summary of all numerical columns in the dataset

II. Data Cleaning and Preprocessing

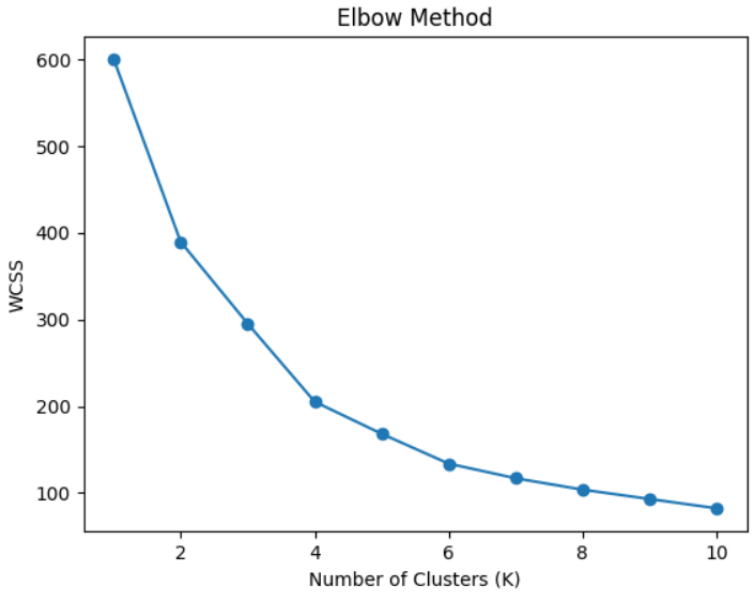
The cleaning and preprocessing steps performed included:

- Checking for missing values: no missing values were found
- Checking for duplicate values: no duplicate values were found
- Verifying appropriate data types
- Cleaning column names for consistency
- Label- encoding on Gender column
- One- hot encoding on Marital Status and Education columns
- Feature scaling using StandardScaler to normalize numerical values before clustering

III. K- Selection

- Elbow method

Elbow method is applied on scaled feature data to identify the point where the rate of decrease in WCSS changes sharply, which gives the optimal number of clusters.



In the above graph we can see a sharp decrease in WCSS from K=1 to K=6, after which the curve starts to flatten. This shows that adding clusters more than K=6, does not

improve cluster overlapping. Therefore, the optimal number of clusters is selected as K=6.

- Silhouette score

Silhouette score measures the cohesion, how close the data points in the same cluster are, and separation, how far clusters are from each other. If the score is closer to 1, it shows the clustering is better or that the clusters are well separated. Below is the silhouette score analysis:

K = 2 , Silhouette Score = 0.33547192894004574
K = 3 , Silhouette Score = 0.357793388710272
K = 4 , Silhouette Score = 0.4039582785148566
K = 5 , Silhouette Score = 0.41664341513732767
K = 6 , Silhouette Score = 0.4284167762892593
K = 7 , Silhouette Score = 0.417231894954916
K = 8 , Silhouette Score = 0.4082067042807375
K = 9 , Silhouette Score = 0.41769250624076476
K = 10 , Silhouette Score = 0.40655411010117015

The silhouette scores analysis shows K=6 has the highest score(closest to 1). As the elbow method also showed K=6 as the optimal cluster, K=6 is selected as the optimum number of clusters.

IV. Model Building and Visualization

The K-Means clustering algorithm was used to segment customers into K=6 different groups. The cluster visualization is given below:

The average age and spending score of each of the clusters are:

	Age	Spending_Score_1-100
Cluster		
0	56.333333	49.066667
1	26.794872	48.128205
2	41.939394	16.969697
3	32.692308	82.128205
4	25.000000	77.608696
5	45.523810	19.380952

2.3.6 Cluster Interpretation

The clustering analysis shows 6 distinct customer segments differentiated by spending score and age. The visualization shows clear separation between groups, which confirms effective clustering.

The group wise mean analysis highlights different customer profiles which provide meaningful insights to businesses to develop effective marketing strategies.

- Clusters 3 and 4 represent high spending customers with relatively younger age groups (32 and 25), indicating that younger age groups inhabit strong spending habits.
- Clusters 0 and 1 have the next highest spending customers with ages around 26 and 56.
- Clusters 5 and 2 have the least spending customers belonging to middle age groups (45 and 41), indicating conservative spending patters.

The visualization confirms that spending score is the dominant factor in segmentation, while age provides additional refinement within each group.

2.3.7 Conclusion

This project successfully applied the K-Means clustering algorithm to perform customer segmentation based on age and spending behaviour. The analysis identified distinct customer segments with varying spending patterns and age distributions. Younger customers were observed to have higher and more diverse spending behaviour, while middle-aged and older customers were generally associated with moderate or lower spending patterns. These insights demonstrate how customer segmentation can help businesses better understand customer behaviour and make data-driven decisions.

The generated customer segments can support targeted marketing, personalized promotions, customer retention strategies, and improved business planning. Overall, the project demonstrates how clustering techniques can transform raw customer data into actionable business insights.

2.4 Week 4 Project: Food Order Preference Clustering (Unsupervised Learning Project)

2.4.1 Introduction

In recent years, the food and hospitality industry has increasingly relied on data-driven approaches to understand customer behaviour and improve user experience. With the rapid growth of online food delivery platforms and digital ordering systems, large amounts of customer preference data are generated every day. Analysing this data can help businesses understand consumer habits, identify popular choices, and design personalized services that improve customer satisfaction and engagement.

This project applies unsupervised learning methods to analyze food preference data and identify distinct customer segments based on demographic information and food-related choices. By combining preprocessing, dimensionality reduction, clustering, and evaluation techniques, the project demonstrates how machine learning can be used to uncover meaningful patterns in consumer preference data.

2.4.2 Objective

The main objectives of the project are:

- Identify distinct groups of users based on their food ordering preferences using unsupervised learning techniques.
- Perform preprocessing and exploratory data analysis on food preference data.
- Apply dimensionality reduction using PCA.
- Implement clustering techniques to group similar users.
- Determine the optimal number of clusters using evaluation metrics.
- Interpret clusters and derive meaningful business insights for personalization and targeted recommendations.

2.4.3 Dataset Description

The Kaggle dataset, 'Food_Preference.csv', contains information related to customer food preferences and demographic details based on a survey held. The description of each of the attributes in the dataset are:

- Timestamp- Timestamp of the survey
- Participant_ID- Participant identification number
- Gender- Gender of the Participant
- Nationality- Nationality of the Participant
- Age- Age of the Participant
- Food- Food Preference of the Participant
- Juice- Juice Preference of the Participant
- Dessert- Dessert Preference of the Participant

2.4.4 Tools and Technologies Used

The tools and libraries used in the project are:

- Python (primary programming language)
- Pandas (data loading, data manipulation, data cleaning, data preprocessing)
- NumPy (handling large numerical datasets)
- Matplotlib (data visualization)
- Seaborn (advanced statistical visualization)
- Scikit-learn (Feature scaling, PCA, KMeans model training, and silhouette_score)

2.4.5 Methodology

I. Data Understanding

The dataset, 'Food_Preference.csv', was imported using the Pandas library and explored to gain an understanding of its structure, dimensions, variable types, and overall composition. The functions used are:

- .head(): To display first five rows of the dataset
- .shape: To get the total number of rows and columns in the dataset
- .columns.tolist(): To get a list of columns in the dataset
- .dtypes: To get the datatypes of each column in the dataset
- .info(): To get a technical summary of the dataset (datatypes, non-null counts, and memory usage)
- .describe(): To get a statistical summary of all numerical columns in the dataset

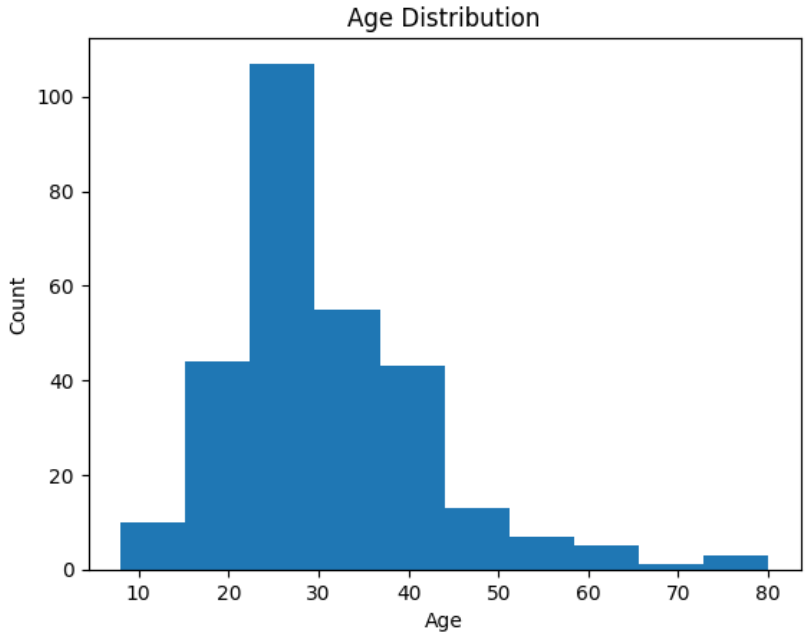
II. Data Cleaning and Preprocessing

The following preprocessing and cleaning steps were performed:

- Checked for missing values: 4 missing values were found
- Checked for duplicate values: no duplicate values were found
- Verified data type correctness
- Removed unnecessary columns (Timestamp and Participant_ID)
- Handled missing values in the Gender column using mode imputation.
- Applied One-Hot Encoding to categorical variables.
- Standardized features using StandardScaler.

III. Exploratory Data Analysis (EDA)

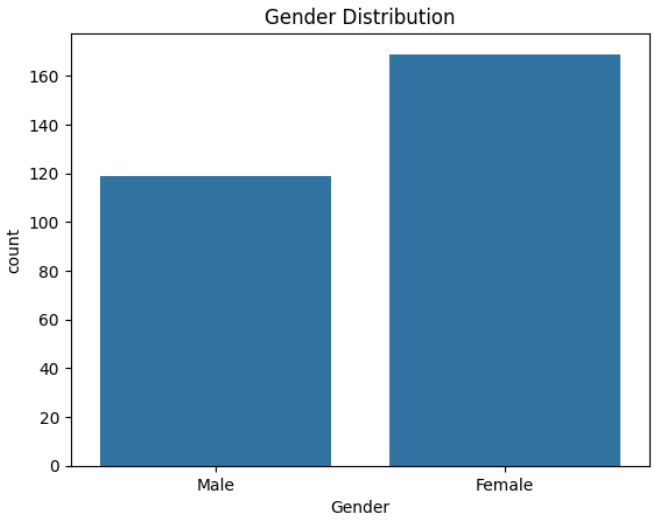
- **Chart-1: Age Distribution**



Insights:

- The above histogram gives the age distribution in the dataset.
- From the chart it can be inferred that the data is right skewed as the datapoints are clustered towards the left side of the chart and there is a tail extending towards the left.
- The age groups around 20-30 have the highest frequency, showing that the majority of the survey participants are young adults or that this age group is the most active regarding food ordering.

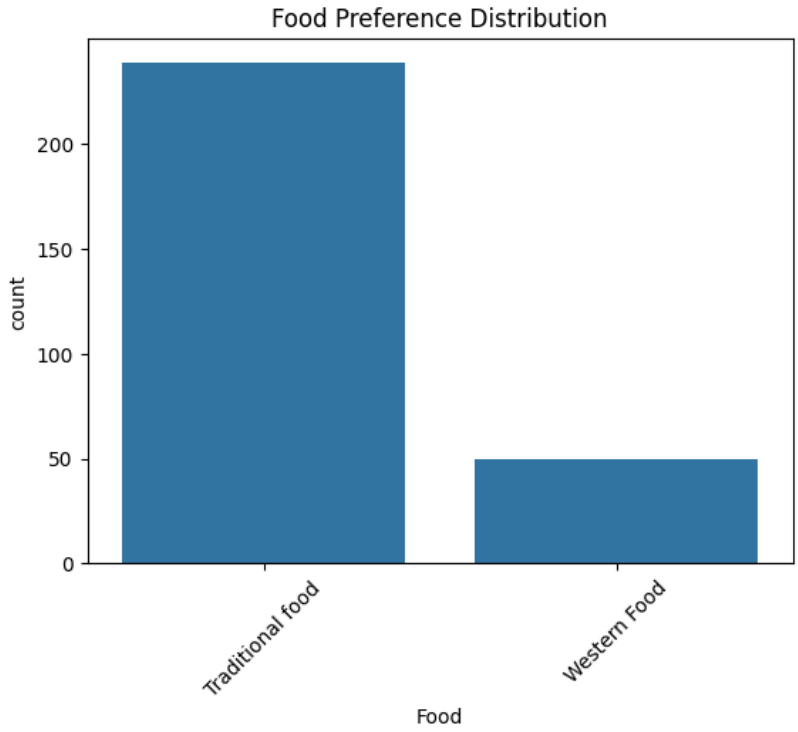
- **Chart-2: Gender Distribution**



Insights:

- The graph shows the gender distribution in the dataset.
- From the graph it can be inferred that females represent the majority of the participants in the survey.
- The number for females exceeds 160 whereas the count for males is around 120.

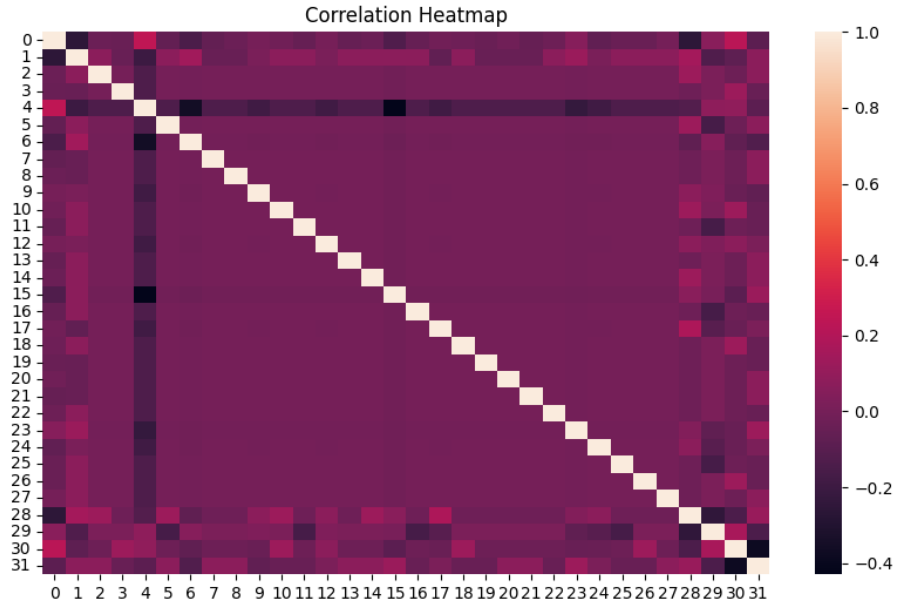
- **Chart-3: Food Preference Distribution**



Insights:

- The above graph shows the food preference distribution in the dataset.
- From the graph it can be inferred that the number of participants who prefer traditional food are way more than the number of participants who prefer western food.
- This shows that more participants value local/ traditional flavours than global flavours.

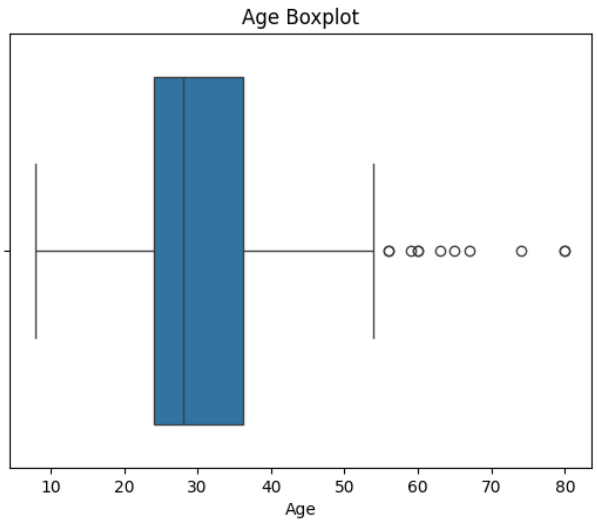
- **Chart-4: Correlation Heatmap**



Insights:

- The heatmap visualizes the correlation between all the features (encoded and scaled) in the dataset.
- The colour scale ranges from -0.4(weak negative correlation) to 1.0(strong positive correlation).
- The white diagonal line shows the perfect correlation each feature has to itself.
- The rest of the heatmap is dark, showing that there is close to zero correlation.
- The heatmap indicates that there is no single feature or factor that strongly influences food preference.

- **Chart-5: Age Boxplot**



Insights:

- The boxplot shows that the median age(line inside the box) is 28 years. 50% of the participants are below 28 years and the other 50% are above 28 years.
- Most of the participants are concentrated in the 24-36 age group.
- The circles on the right side of the plot are the outliers. They represent the participants who are significantly older than the rest of the participants.
- The box is shifted to the right, showing that the data is right-skewed.

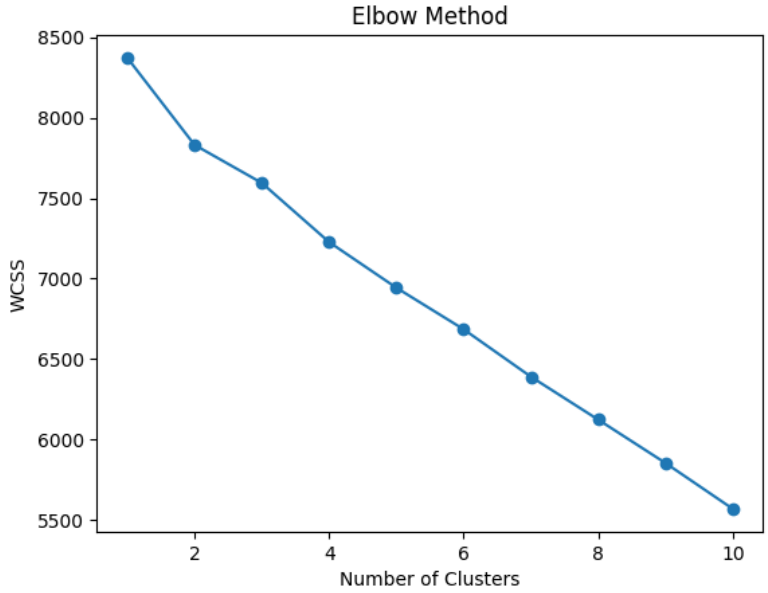
IV. Dimensionality Reduction

Principal Component Analysis (PCA) was applied to reduce dimensionality while preserving most of the variance in the dataset. Components explaining at least 90% variance were selected, i.e., 26 components. PCA was then applied with the optimal number of components, in this case 26.

V. Clustering

- Elbow method and Silhouette scores

  Elbow method is used to find the optimal number of clusters to be used for k means clustering. Elbow method plots WCSS against the number of clusters to determine the "elbow", which gives the optimal number of clusters, whereas the silhouette scores give the closeness of the points within the cluster and the separation of the clusters.



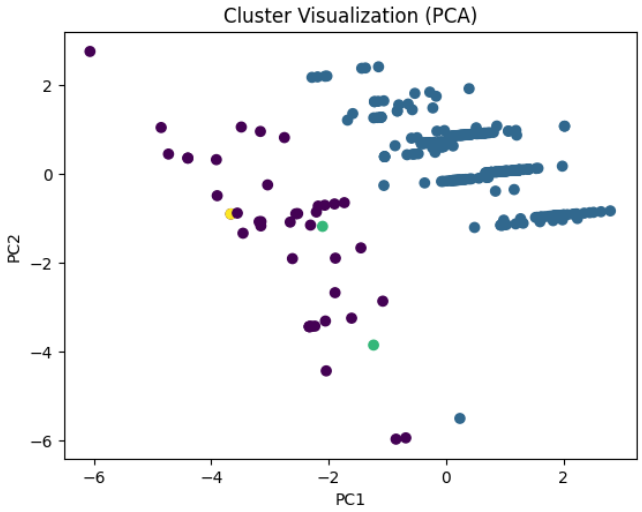
Silhouette scores:

K = 2 , Silhouette Score = 0.6142677545310526
K = 3 , Silhouette Score = 0.27674932025733945
K = 4 , Silhouette Score = 0.6320682020476576
K = 5 , Silhouette Score = 0.631474537941138
K = 6 , Silhouette Score = 0.6015865208252907
K = 7 , Silhouette Score = 0.2033944330578943
K = 8 , Silhouette Score = 0.2051601678827914
K = 9 , Silhouette Score = 0.4245041606056491
K = 10 , Silhouette Score = 0.5225440114908118

There is no sharp elbow seen in the elbow graph. But through the silhouette scores, K=4 was chosen as the optimal number of clusters as its value is the closest to 1.

- Apply K Means

K-Means clustering was implemented to segment users into K groups based on the k selected using the Elbow method and Silhouette scores, in this case 4 groups. Below is the visualization using the first 2 PCA components:



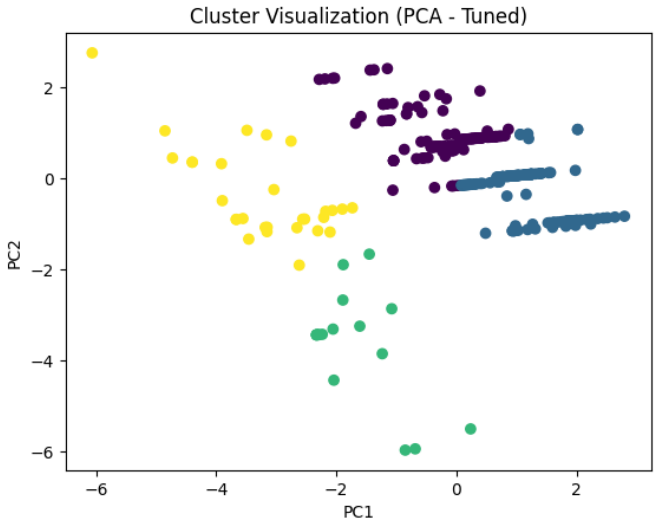
VI. Model Tuning

Manual tuning was performed by testing different:

- PCA components
- Cluster counts
- Initialization methods
- Iteration values

The best model parameters achieving the highest silhouette score of 0.48 were: {'n_components': 2, 'n_clusters': 4, 'init': 'k-means++', 'max_iter': 300}. K Means was again applied using these parameters.

The cluster visualization after model hyperparameter tuning is given below:



2.4.6 Results and Interpretation

I. Results

The cluster-wise counts are:

	count
Cluster	
1	128
0	114
3	29
2	17

dtype: int64

The cluster-wise summary is given below:

	Age
Cluster	
0	27.175439
1	35.851562
2	24.117647
3	24.655172

dtype: float64

```
 Gender
Cluster  Gender
0        Female    0.526316
         Male      0.473684
1        Female    0.726562
         Male      0.273438
2        Male      0.705882
         Female    0.294118
3        Male      0.620690
         Female    0.379310
Name: proportion, dtype: float64
```

```
 Nationality
Cluster  Nationality
0        Indian         0.991228
         Algerian       0.008772
1        Indian         1.000000
2        Indonesia      0.411765
         Japan          0.117647
         China          0.058824
         Korean         0.058824
         MY             0.058824
         Malaysian      0.058824
         Maldivian      0.058824
         Mauritian      0.058824
         Pakistan       0.058824
         Tanzanian      0.058824
3        Malaysian      0.310345
         Pakistani      0.103448
         Maldivian      0.068966
         Pakistani      0.068966
         Canadian       0.034483
         Indonesain     0.034483
         Indonesian     0.034483
         Indonesian     0.034483
         MALAYSIAN      0.034483
         MY             0.034483
         Malaysia       0.034483
         Malaysia       0.034483
         Malaysian      0.034483
         Muslim         0.034483
         Nigerian       0.034483
         Seychellois    0.034483
         Yemen          0.034483
Name: proportion, dtype: float64
```

```
 Food
Cluster  Food
0        Traditional food    0.692982
         Western Food        0.307018
1        Traditional food    0.984375
         Western Food        0.015625
2        Traditional food    0.882353
         Western Food        0.117647
3        Traditional food    0.620690
         Western Food        0.379310
Name: proportion, dtype: float64

 Juice
Cluster  Juice
0        Fresh Juice          0.789474
         Carbonated drinks    0.210526
1        Fresh Juice          1.000000
2        Fresh Juice          1.000000
3        Fresh Juice          0.724138
         Carbonated drinks    0.275862
Name: proportion, dtype: float64

 Dessert
Cluster  Dessert
0        Yes        0.754386
         Maybe      0.245614
1        Maybe      0.585938
         No         0.367188
         Yes        0.046875
2        Maybe      0.705882
         No         0.294118
3        Yes        0.758621
         Maybe      0.241379
Name: proportion, dtype: float64
```

II. Interpretation

Based on the outputs, here are the cluster profiles and labels:

- Cluster 0:

  Size: Large (114)

  Age: around 27

  Gender: Balanced

  Food: Mostly traditional

  Drinks: Mostly fresh juice

  Dessert: Strong preference for dessert

- Cluster 1:

  Size: Largest (128)

  Age: Highest (around 36)

  Gender: Female-dominated

  Food: Almost entirely traditional

  Drinks: 100% fresh juice

  Dessert: Mostly “Maybe” or “No”

- Cluster 2:

  Size: Small (17)

  Age: Youngest (around 24)

  Gender: Male-dominated

  Nationality: Highly diverse

  Food: Mostly traditional but more open than Cluster 1

  Drinks: 100% fresh juice

  Dessert: Mostly “Maybe”

- Cluster 3:

  Size: Medium-small (29)

  Age: Around 24-25

  Gender: Male-dominated

  Food: Highest western preference

  Drinks: Highest carbonated drink usage

  Dessert: Strong “Yes”

2.4.7 Conclusion

This project successfully demonstrated the application of unsupervised learning techniques to analyse and segment customer food preferences. Through systematic data preprocessing, exploratory data analysis, dimensionality reduction using PCA, and clustering using the K-Means algorithm, meaningful patterns were identified within the dataset. The implementation of

evaluation techniques such as the Elbow Method and Silhouette Score helped determine the optimal clustering structure, while hyperparameter tuning improved the overall clustering performance.

The final model identified four distinct customer groups with varying food habits, beverage choices, dessert preferences, and demographic characteristics. These clusters provided valuable behavioural insights that can support personalized recommendations, targeted marketing strategies, and improved customer engagement in the food and hospitality industry.

2.5 Week 5 Project: Iris Flower Classification Using Artificial Neural Network (ANN)

2.5.1 Introduction

Machine learning classification techniques are widely used to identify patterns and categorize data into predefined classes. In this project, an Artificial Neural Network (ANN) was developed to classify iris flowers into different species based on their physical measurements. The project demonstrates the application of deep learning concepts such as Dense layers, activation functions, Softmax classification, and model evaluation techniques.

The Iris dataset is one of the most commonly used benchmark datasets in machine learning because of its simplicity and effectiveness in demonstrating classification algorithms. The dataset contains measurements of iris flowers, including sepal length, sepal width, petal length, and petal width. Using these features, the ANN model learns patterns that distinguish the three species of iris flowers: Setosa, Versicolor, and Virginica.

This project provides practical experience in data preprocessing, neural network design, model training, and evaluation using Python and TensorFlow/Keras.

2.5.2 Objective

The main objectives of the project are:

- To develop an Artificial Neural Network (ANN) model for iris flower classification.
- To classify iris flowers into Setosa, Versicolor, and Virginica species based on flower measurements.
- To perform data preprocessing techniques such as scaling and encoding.
- To implement Dense layers and Softmax activation for multiclass classification.
- To train and evaluate the neural network using appropriate performance metrics.
- To analyse the model’s accuracy, precision, recall, F1-score, and confusion matrix.

- To gain practical understanding of deep learning workflows using TensorFlow/Keras.

2.5.3 Dataset Description

The project uses the built-in Iris dataset available in Scikit-learn. There are 150 rows and 5 columns. The description of each of the attributes in the dataset are:

- sepal length(cm)- Length of the sepal.
- sepal width(cm)- Width of the sepal.
- petal length(cm)- Length of the petal.
- petal width(cm)- Width of the petal.
- target- Type of iris flower:
  - 0: Setosa
  - 1: Versicolor
  - 2: Virginica

2.5.4 Tools and Technologies Used

The tools and libraries used in the project are:

- Python (primary programming language)
- NumPy (handling large numerical datasets)
- Pandas (data loading, data manipulation, data cleaning, data preprocessing)
- Scikit-learn (train- test splitting, feature scaling, model evaluation)
- TensorFlow / Keras (One- hot encoding, Sequential model training, and Dense layers implementation)

2.5.5 Methodology

I. Data Understanding

The iris dataset was imported using the Pandas library and was initially split into features and target. It was then explored to gain an understanding of its structure, dimensions, variable types, and overall composition. The functions used are:

- .head(): To display first five rows of the dataset
- .shape: To get the total number of rows and columns in the dataset
- .columns.tolist(): To get a list of columns in the dataset
- .dtypes: To get the datatypes of each column in the dataset
- .info(): To get a technical summary of the dataset (datatypes, non-null counts, and memory usage)
- .describe(): To get a statistical summary of all numerical columns in the dataset

II. Data Preprocessing

The cleaning and preprocessing steps which were done are:

- Checking for missing values: no missing values were found
- Checking for duplicate values: 1 duplicate value was found
- Removing duplicate rows
- Verifying correctness of data types
- Changing column names for consistency
- Splitting data into training and testing sets
- Feature scaling using StandardScaler
- One-hot encoding of target labels using to_categorical()

III. Model Architecture

The Artificial Neural Network (ANN) model for this project was developed using the Sequential API provided by TensorFlow/Keras. A Sequential model is suitable for feedforward neural networks where data flows linearly from the input layer through hidden layers to the output layer.

Network Structure**:**

- Input Layer: 4 input features
- Hidden Layer 1: 10 neurons with ReLU activation.
  ReLU introduces non-linearity into the model and helps the network learn complex patterns efficiently by converting negative values to zero while retaining positive values.
- Hidden Layer 2: 8 neurons with ReLU activation. This hidden layer improves the network's ability to capture deeper relationships between the input features and target classes.
- Output Layer: 3 neurons with Softmax activation. Softmax function converts the output values into probability distributions across the three classes. The class with the highest probability is selected as the final prediction.
  The model contained a total of 165 trainable parameters.

IV. Model Training

The model was compiled using:

- Optimizer: Adam
  The Adam optimizer was selected because it combines adaptive learning rates and momentum, allowing faster and more stable convergence during training.
- Loss Function: Categorical Crossentropy
  The categorical crossentropy loss function is appropriate for multiclass classification problems where target labels are one-hot encoded.
- Metric: Accuracy

Training was performed for:

- Epochs: 50
  The model was trained for 50 epochs, meaning the network processed the entire training dataset 50 times.
- Batch Size: 5
  A batch size of 5 was used, enabling frequent weight updates and effective learning on the relatively small dataset.
- Validation Split: 20%
  20% of the training data was reserved for validation during training.

Validation accuracy and validation loss were monitored to evaluate the model's ability to generalize to unseen data and to detect any signs of overfitting. The training process showed gradual improvement in accuracy and reduction in loss, indicating effective learning.

2.5.6 Results and Interpretation

I. Results

The trained ANN model achieved a test accuracy of 96.67%, indicating that it correctly classifies the vast majority of iris samples.

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	0.89	0.94	9
2	0.92	1.00	0.96	11
accuracy			0.97	30
macro avg	0.97	0.96	0.97	30
weighted avg	0.97	0.97	0.97	30

Confusion Matrix:

```
[[10  0  0]
 [ 0  8  1]
 [ 0  0 11]]
```

II. Interpretation:

- Class 0 (Setosa) shows perfect performance with precision, recall, and F1-score of 1.00, meaning all samples are correctly classified with no errors.
- Class 1 (Versicolor) has a recall of 0.89, indicating that one instance was misclassified.
- Class 2 (Virginica) achieves recall of 1.00 and a high precision of 0.92, showing strong performance.
- The confusion matrix confirms that all Setosa samples are correctly classified,
- One Versicolor sample is misclassified as Virginica and all Virginica samples are correctly classified.
- Overall, the model shows high accuracy and balanced performance across all classes, with only minor confusion between similar species.

2.5.7 Conclusion

The Iris Flower Classification project successfully demonstrates the implementation of an Artificial Neural Network for multiclass classification using TensorFlow/Keras. The model was trained using flower measurement features and was able to accurately classify iris flowers into Setosa, Versicolor, and Virginica species.

The project involved several important stages, including data preprocessing, feature scaling, one-hot encoding, neural network construction, model training, and performance evaluation. A lightweight ANN architecture with two hidden Dense layers and a Softmax output layer was designed to effectively learn patterns from the dataset.

The trained model achieved a test accuracy of approximately 96.67%, along with high precision, recall, and F1-scores across all classes. The confusion matrix showed that most samples were correctly classified, with only minimal confusion between Versicolor and Virginica due to similarities in their characteristics.

Overall, the project serves as a strong foundational example of applying deep learning techniques to structured datasets.

3. Major Project: Credit Card Fraud Detection Using Generative AI (CTGAN)

3.1 Introduction

Credit card fraud detection is a critical challenge in the financial sector due to the increasing number of online transactions and digital payment systems. Fraudulent transactions are extremely rare compared to legitimate transactions, resulting in highly imbalanced datasets. Traditional machine learning models trained on such datasets often fail to accurately identify fraudulent transactions because they become biased toward the majority class.

This project aims to improve fraud detection performance using Generative Artificial Intelligence techniques. A Conditional Tabular Generative Adversarial Network (CTGAN) was used to generate synthetic fraud samples, thereby reducing class imbalance and improving the learning capability of the classification model. The generated synthetic data was combined with the original dataset to train an improved fraud detection model.

The project demonstrates the application of Generative AI in solving real-world imbalance problems and improving predictive performance in financial systems.

3.2 Objectives

The main objectives of the project are:

- To analyse and preprocess the credit card transaction dataset
- To perform exploratory data analysis for identifying fraud patterns
- To handle class imbalance using synthetic data generation
- To generate synthetic fraud samples using CTGAN
- To train machine learning models for fraud detection

- To compare baseline and augmented model performance
- To deploy the final model using Streamlit library

3.3 Dataset Description

The Kaggle dataset, 'creditcard.csv', contains transactions made by credit cards in September 2013 by European cardholders. The features were transformed using Principal Component Analysis (PCA) to preserve confidentiality. The dataset includes both legitimate and fraudulent transactions. The description of each of the attributes are:

- V1, V2, … V28: The principal components obtained with PCA.
- Time: The seconds elapsed between each transaction and the first transaction in the dataset.
- Amount: The transaction amount.
- Class: The target variable, which takes value 1 in case of fraud and 0 otherwise.

3.4 Tools and Technologies Used

- Python (primary programming language)
- Pandas (data manipulation, data cleaning, data preprocessing)
- NumPy (handling large numerical datasets)
- Matplotlib (data visualizations)
- Seaborn (advanced statistical visualizations)
- Scikit-learn (data preprocessing, feature scaling, train-test splitting, Random Forest model training, model evaluation)
- SDV (CTGAN) (generating synthetic fraud samples, learning minority class distributions, improving severe class imbalance)
- Streamlit (deploying trained model, building an interactive web app, providing real-time fraud prediction)

3.5 Methodology

I. Data Understanding

The dataset, 'creditcard.csv', was imported using the Pandas library and explored to gain an understanding of its structure, dimensions, variable types, and overall composition. The functions used are:

- .head(): To display first five rows of the dataset
- .shape: To get the total number of rows and columns in the dataset
- .columns.tolist(): To get a list of columns in the dataset
- .dtypes: To get the datatypes of each column in the dataset
- .info(): To get a technical summary of the dataset (datatypes, non-null counts, and memory usage)
- .describe(): To get a statistical summary of all numerical columns in the dataset

II. Data Cleaning

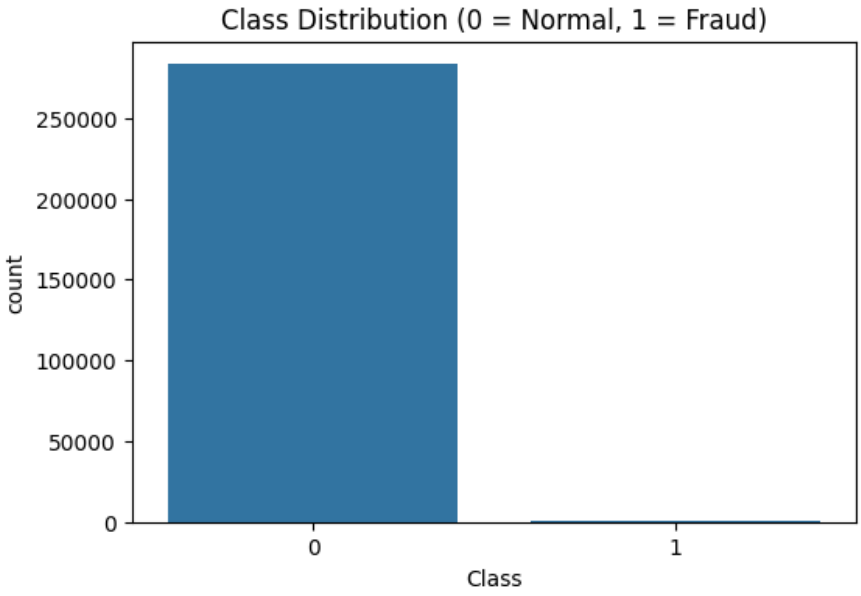
The following cleaning steps were performed:

- Checked for missing values: no missing values were found
- Checked for duplicate values: 1081 duplicate values were found
- Dropped the duplicate values
- Verified data types
- Checked for unique values
- Checked class distribution: the dataset was found to be highly imbalanced with the fraud transactions being only 0.17% of the dataset.

III. Exploratory Data Analysis (EDA)

Several visualizations were created to understand dataset characteristics and fraud patterns:

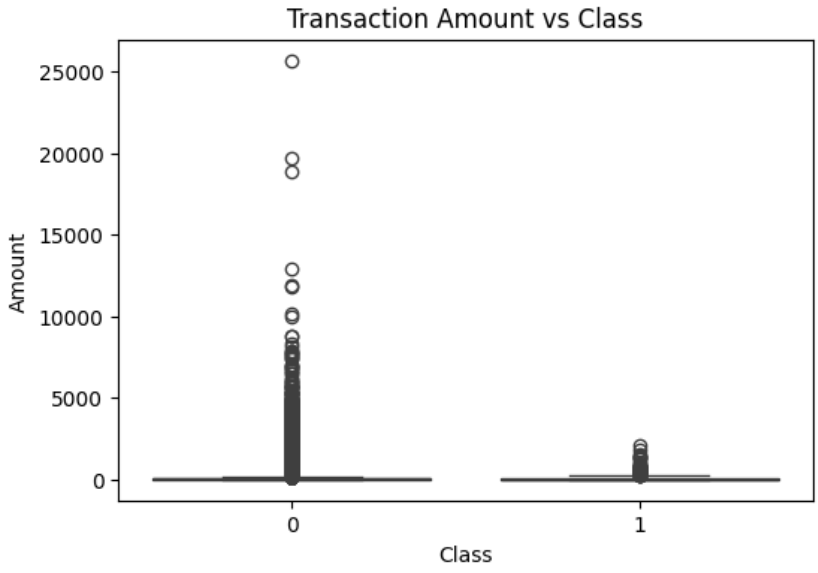
- **Chart-1: Class Distribution**



Insights:

- The above graph shows the imbalance between the two classes, i.e., normal (0) and fraud (1).
- Fraudulent transactions are extremely rare compared to normal transactions. This imbalance makes it difficult for machine learning models to learn fraud patterns effectively.

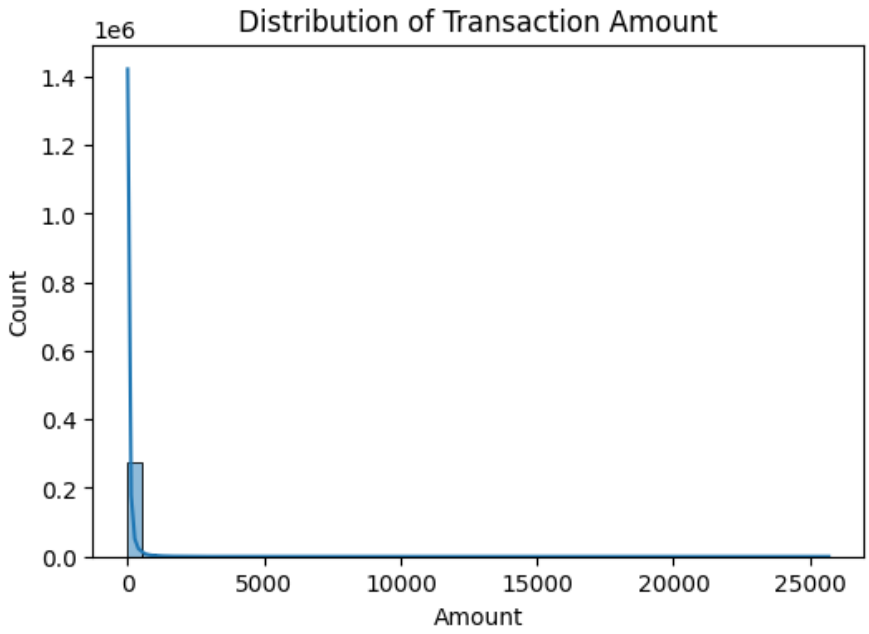
- **Chart-2: Transaction Amount Analysis**



Insights:

- The above boxplot shows that the normal transactions(0) have a wider range of values, including outliers, than compared to fraudulent transactions.
- Both classes show outliers, but the normal class has more extreme high value transactions.
- Fraudulent transactions seem to be concentrated to lower amounts, showing that fraudsters attempt smaller transactions to avoid immediate security alerts.

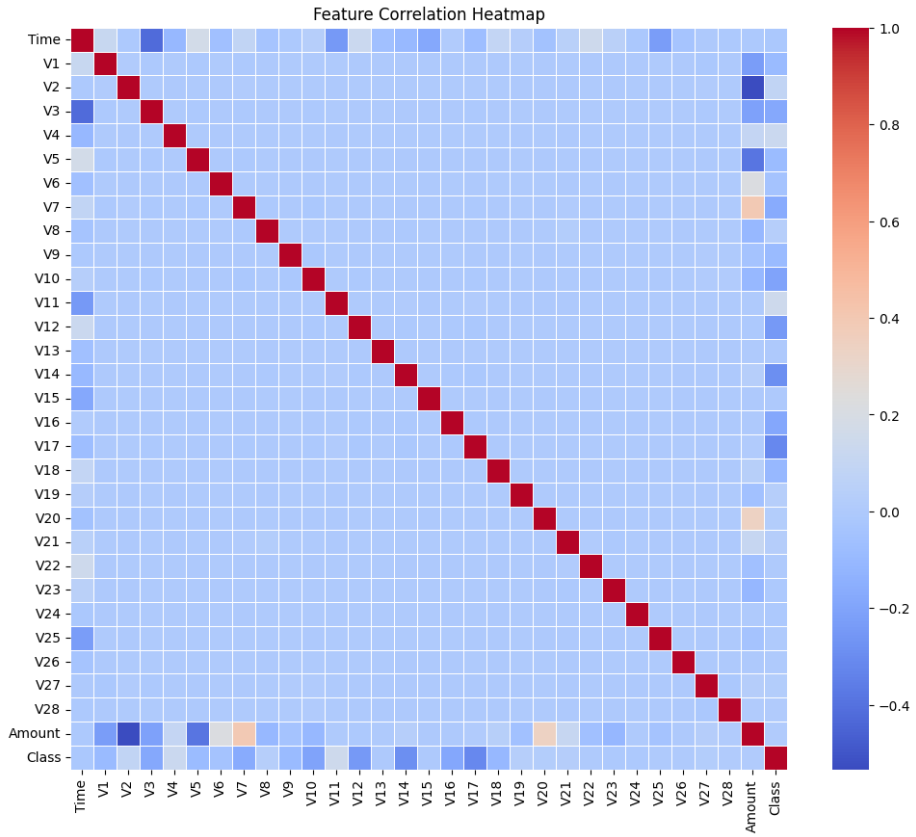
- **Chart-3: Distribution of Transaction Amount**



Insights:

- The above plot shows that the distribution is heavily right-skewed as there is a long tail extending to the right and most of the transactions are very small amounts. This might impact the model performance.

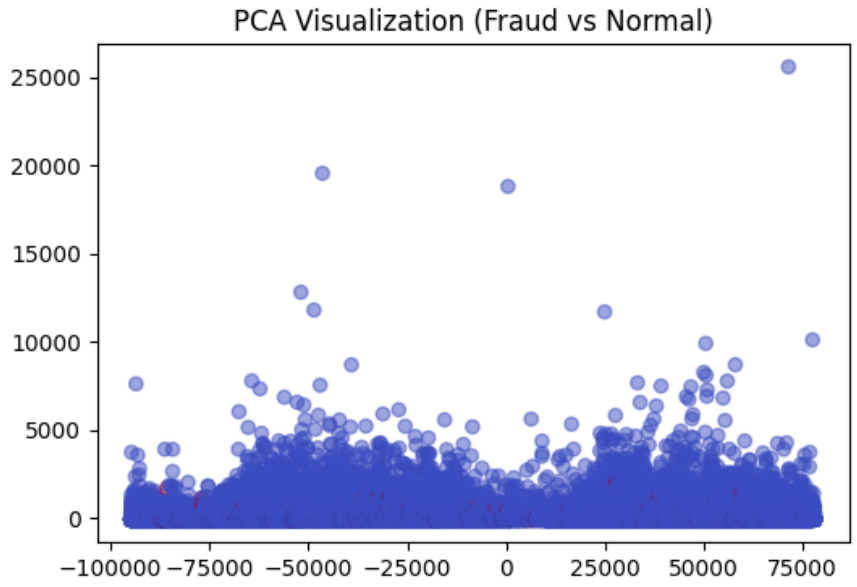
- **Chart-4: Correlation Heatmap**



Insights:

- Most features show weak correlation due to PCA transformation.
- Some features show low negative correlation with 'Class', indicating that as these values decrease, the probability of the transaction being fraudulent increases.
- Similarly, some features show low negative correlation with 'Amount'.
- Some features also show low positive correlation with 'Amount'.

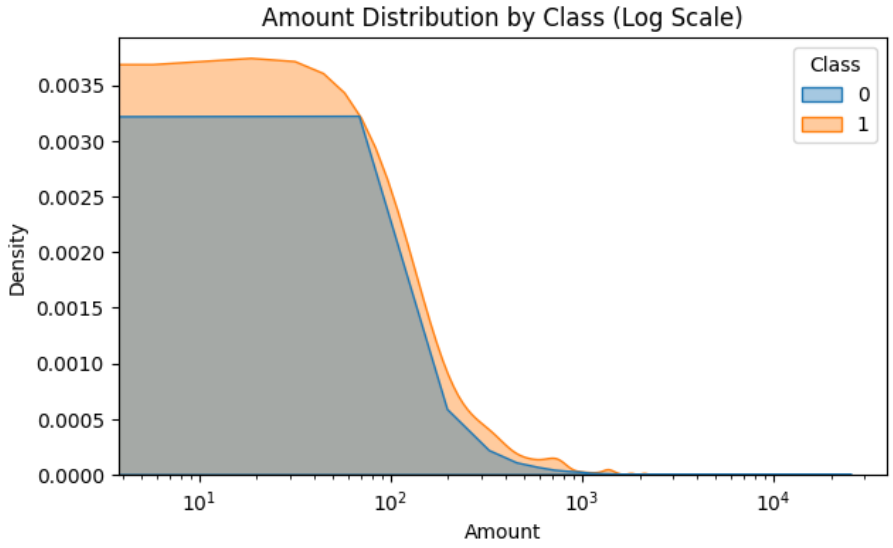
- **Chart-5: PCA Visualization**



Insights:

- This chart reduces 30+ features of the dataset to 2 principal components to allow for a 2D visual representation.
- The normal transactions and the fraud transactions are heavily overlapped, showing that fraud is not easily separable from normal transactions.
- This justifies the need for better data representation by augmentation.

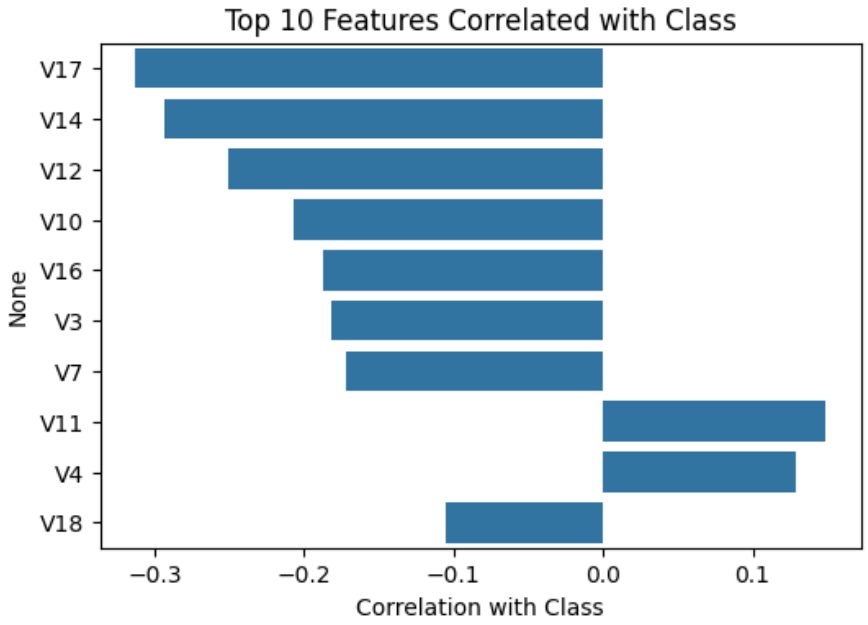
- **Chart-6: Class-wise Amount Distribution**



Insights:

- The log scale distribution shows the overlapping patterns between normal and fraudulent transactions. The log scale pulls the extreme outliers closer to the mean for better visualization.
- While the shapes are similar the fraud curve(orange) shows a slightly different peak compared to the normal curve(blue).

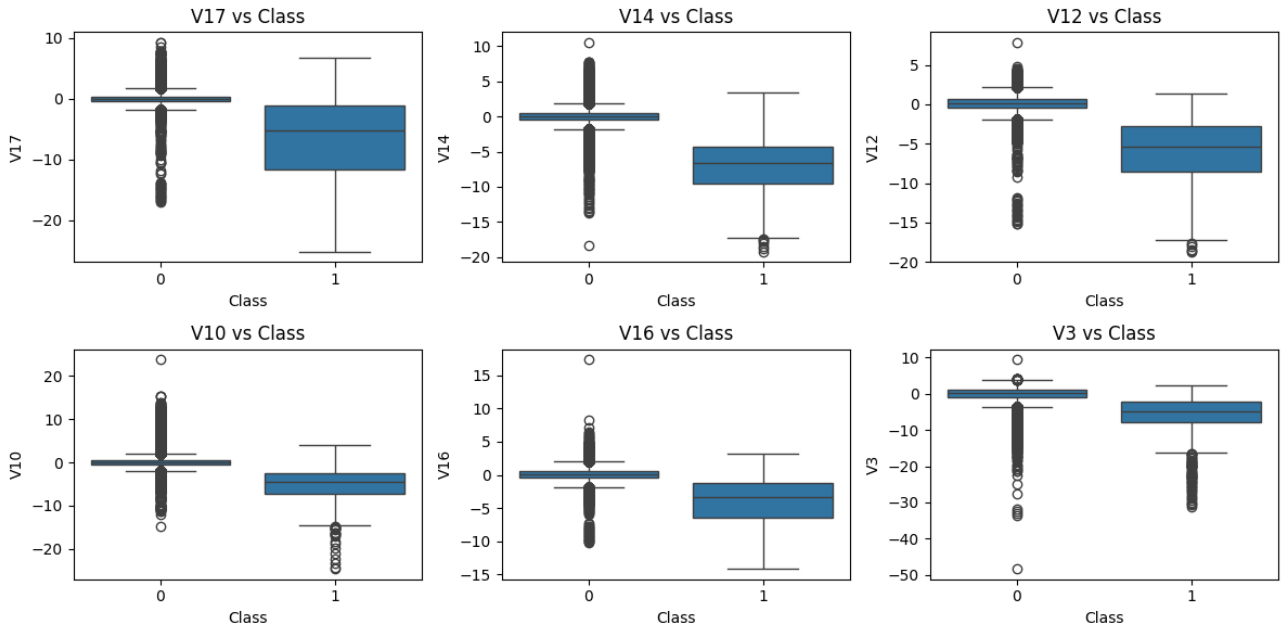
- **Chart-7: Top 10 Correlated Features with 'Class'**



Insights:

- The above chart shows which specific PCA features have the strongest correlation with "Class", i.e., the target variable.
- Most of the features, such as V17, V14, V12, V10, V16, V3, V7, show negative correlation with 'Class'. This indicates that as their values decrease, the probability of the transaction being fraudulent increases.
- The features, V11 and V4 show positive correlation with 'Class', indicating that as their values increases, the probability of the transaction being fraudulent increases.

- **Chart-8: Boxplot for Top Features**



Insights:

- These boxplots confirm the strong negative correlations identified in the previous chart.
- For all the top 6 features, the median value for the 'Fraud' class is significantly lower than the median value of the 'Normal' class.
- This shows that these features are effective discriminators that can help machine learning models to draw a boundary between the two classes.

IV. Data Preprocessing

The following preprocessing steps were performed:

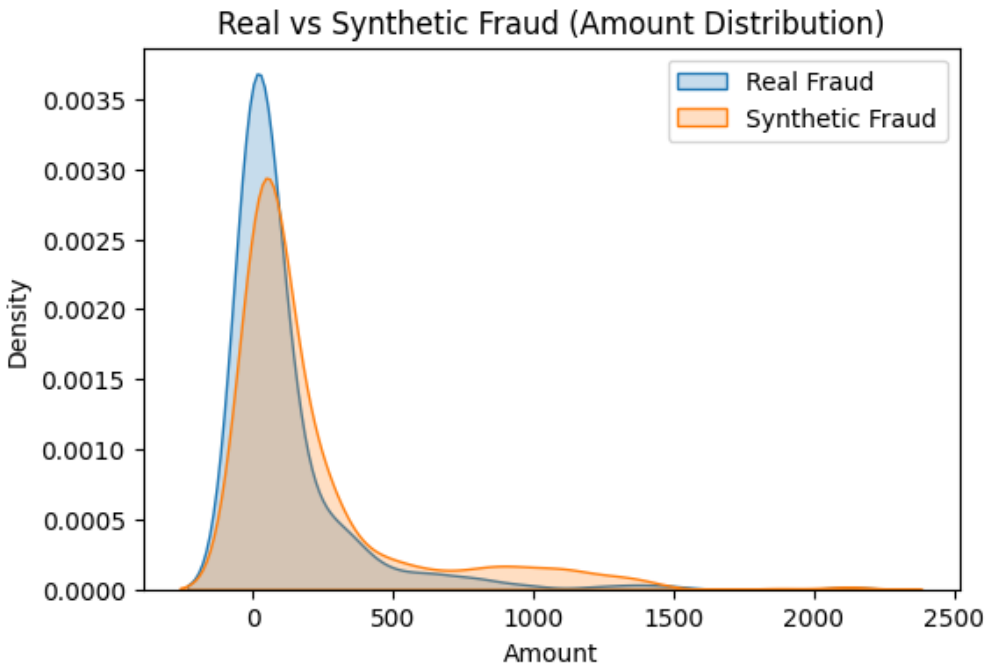
- Separated features and target
- Performed train-test split
- Extracted only the fraud data from the training set to train CTGAN model.

V. CTGAN Model Training

To address class imbalance, CTGAN was applied to generate synthetic fraud samples through the following steps:

- Imported CTGANSynthesizer from sdv.single_table
- Generated metadata using SDV
- Trained CTGAN model on fraud data
- Generated 1000 synthetic fraud samples

After generating the synthetic fraud samples, a visualization was created to compare real fraud data and synthetic fraud data.



The above visualization shows that the Synthetic Fraud curve(orange) closely follows the Real Fraud curve(blue). This indicates that the AI(CTGAN) has successfully learnt the underlying data distribution. Both curves peak at 0, which shows that the synthetic data accurately reflects the fact that most frauds occur at very low amounts. Minor variation is seen but that is desirable for generalisation.

VI. Data Augmentation

The real training data and the generated synthetic fraud data was combined and then shuffled. The new class distribution:

	count
Class	
0	198277
1	1331

dtype: int64

	proportion
Class	
0	99.333193
1	0.666807

dtype: float64

After augmentation, fraud cases increased from 0.17% to 0.67%. This reduced class imbalance while maintaining a realistic data distribution. This balance helps to improve the model's ability to detect fraud without overfitting.
Features and target were separated again for model training.

VII. Model Training

A Random Forest classifier was trained using:

i. Original imbalanced dataset without synthetic fraud data (Baseline Model)
ii. Augmented dataset with synthetic fraud samples (Augmented Model)

i. **Baseline Model**

The Random Forest model was trained using the original imbalanced training dataset.

During this stage:

- The model learned patterns from legitimate and fraudulent transactions
- However, the minority fraud class had very limited representation

ii. **Augmented Model**

A Random Forest classifier was trained using the augmented dataset.

The model learned from:

- Original legitimate transactions
- Original fraud transactions
- Synthetic fraud samples generated by CTGAN

VIII. Model Evaluation

The models were evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

3.6 Results and Analysis

I. Results

i. Baseline Model Performance:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	84976
1	0.96	0.78	0.86	142
accuracy			1.00	85118
macro avg	0.98	0.89	0.93	85118
weighted avg	1.00	1.00	1.00	85118

Confusion Matrix:

```
[[84971     5]
 [   31   111]]
```

ii. Augmented Model Performance:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	84976
1	0.88	0.80	0.83	142
accuracy			1.00	85118
macro avg	0.94	0.90	0.92	85118
weighted avg	1.00	1.00	1.00	85118

Confusion Matrix:

```
[[84960    16]
 [   29   113]]
```

II. Interpretation

i. Baseline Model Interpretation:

- Precision for fraud detection is high (0.96), indicating that predicted fraud cases are mostly correct.
- However, recall is relatively low (0.78), meaning the model fails to detect a significant portion of fraudulent transactions.
- F1-score (86%) shows balanced measure of precision and recall.

- From the confusion matrix, 31 fraud cases were missed, which is critical in financial applications.
- This demonstrates the limitation of training on imbalanced data.

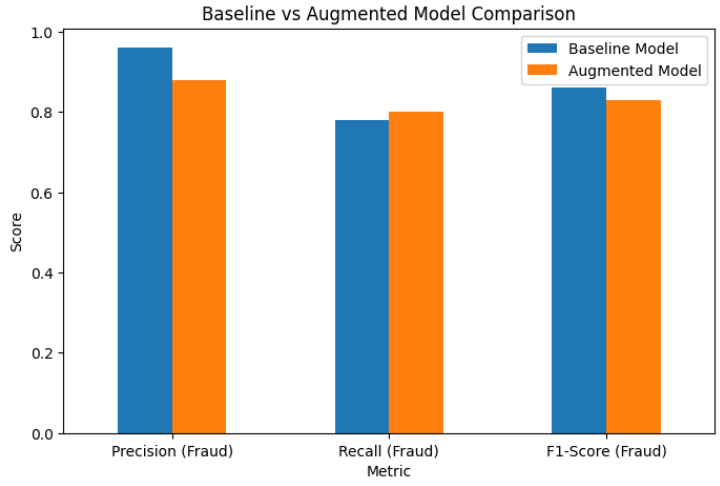
ii. Augmented Model Interpretation

- Multiple configurations were tested by varying the number of synthetic samples and applying class balancing. The best performance was achieved using 1000 synthetic samples without additional class weighting.
- This model improved recall from 0.78 to 0.80, enabling better detection of fraud cases.
- Precision remained high at 0.88, indicating controlled false positives.
- The model demonstrates an effective balance between precision and recall, making it suitable for fraud detection.

III. Final Comparison

A Data Frame was created to show the metrics for the Baseline Model and the Augmented Model. This was used to create various visualizations to compare the two models.

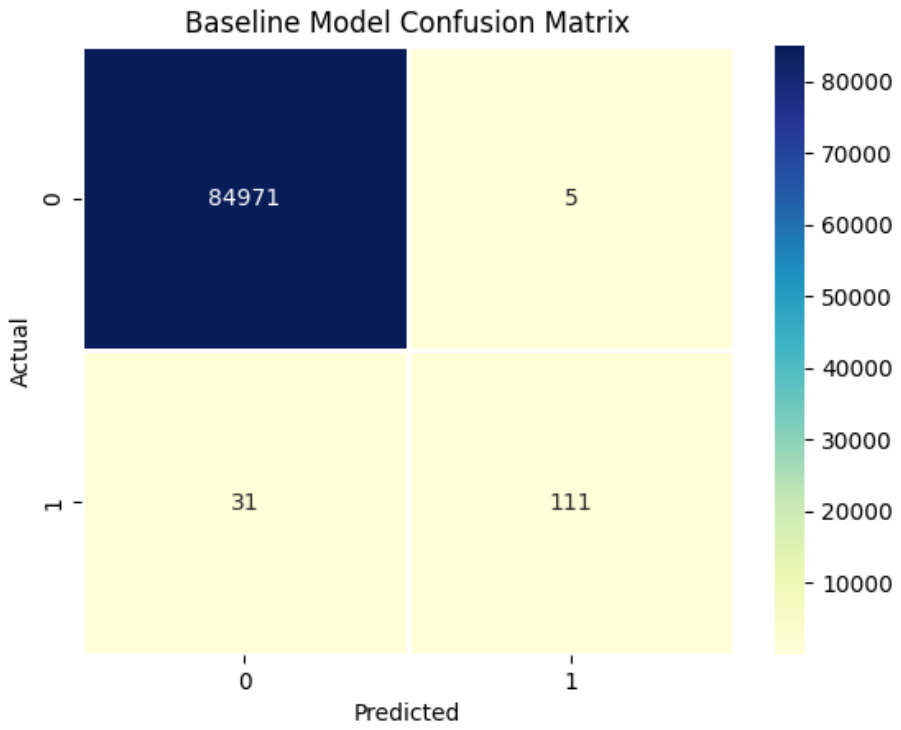
i. Bar Chart Comparison



Insights:

- The baseline model achieved high precision (0.96) but lower recall (0.78), missing several fraud cases.
- The augmented model improved recall to 0.80, enabling better detection of fraudulent transactions.
- Precision slightly decreased to 0.88, but remained high, showing controlled false positives.
- The F1-score remained comparable, indicating balanced performance.

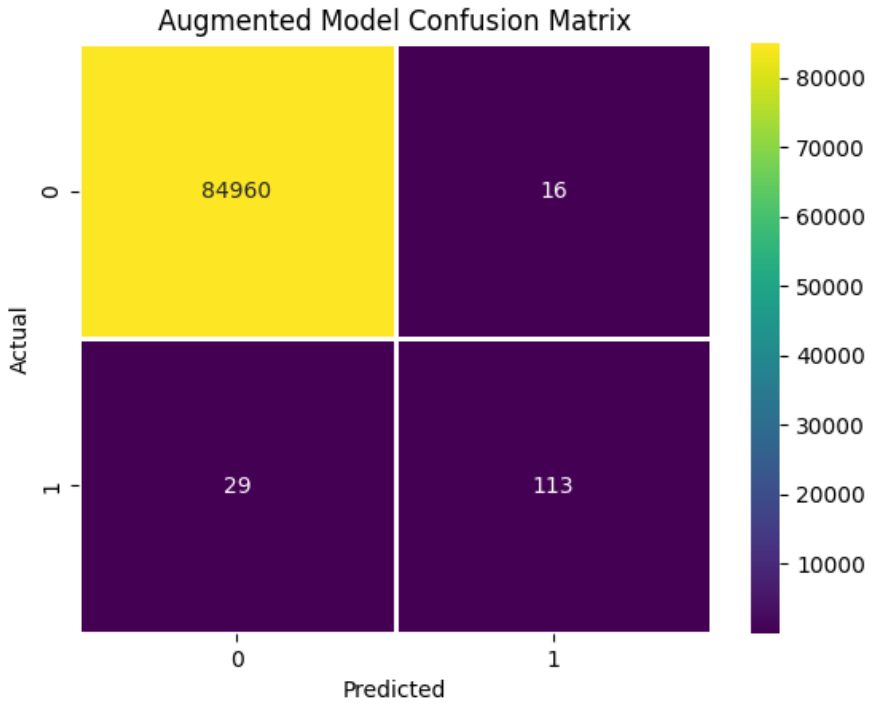
ii. **Confusion matrix Visualization for Baseline Model**



Insights:

- The model correctly classifies most normal transactions with very few false positives.
- But it fails to detect 31 fraudulent transactions. This indicates that the model is biased toward the majority class and struggles with identifying minority class instances, which the augmented model identifies better.

iii. Confusion matrix Visualization for Augmented Model



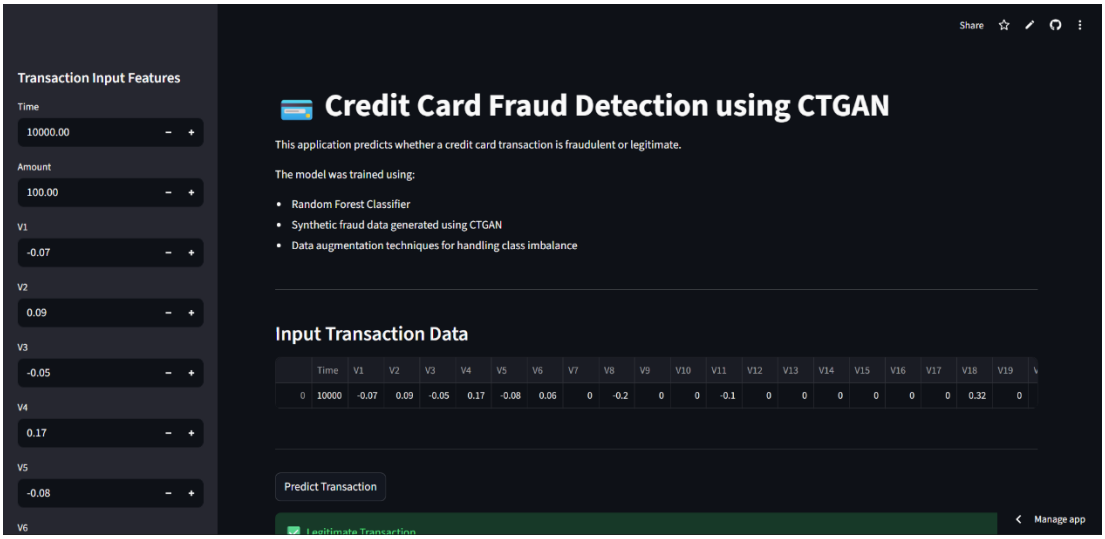
Insights:

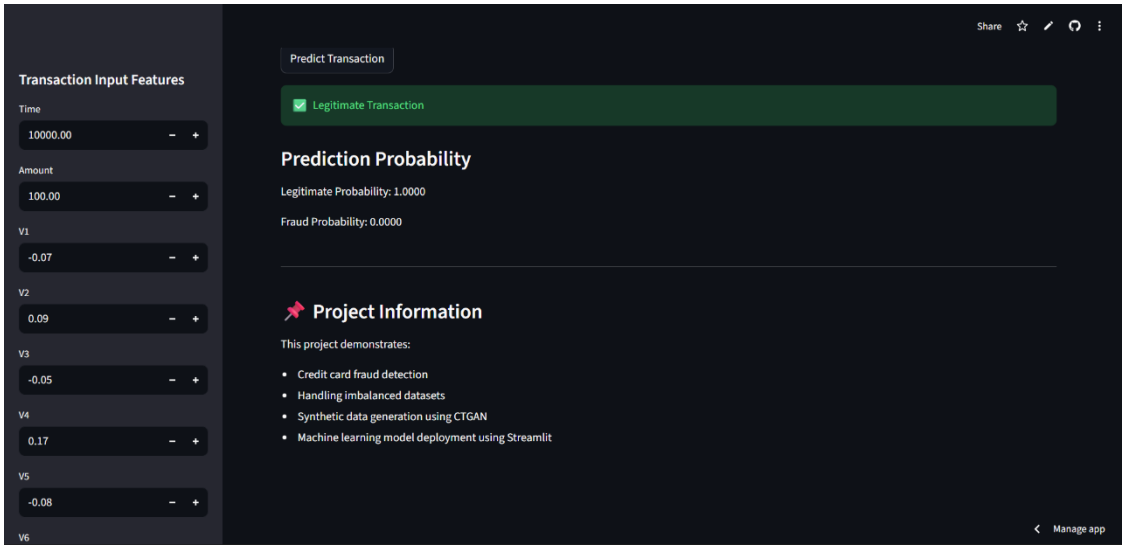
- The model shows improved detection of fraudulent transactions, reducing missed fraud cases from 31 to 29.
- True positive predictions increased, indicating better learning of fraud patterns.
- False positives increased slightly from 5 to 16, but remain relatively low.
- Overall, the model achieves a better balance between detecting fraud and minimizing false alarms.

3.7 Deployment

The final fraud detection model was successfully deployed using Streamlit, enabling real-time prediction through an interactive web application. The deployment integrates the trained Random Forest model and preprocessing scaler to classify transactions as fraudulent or legitimate based on user input features.

- Deployment Platform: Streamlit Community Cloud
- Deployment Type: Web-based interactive application
- Technologies Used: Streamlit, Python, Scikit-learn, Joblib
- Link: https://ai-ml-internship-major-project-8nsepyknvttcrzappjg9eub.streamlit.app/





3.8 Conclusion

This project successfully applied Generative AI techniques for improving credit card fraud detection. CTGAN-generated synthetic data helped address the class imbalance problem and improved the model's ability to identify fraudulent transactions.

The project also demonstrated that careful tuning of synthetic data quantity is essential to maintain balanced model performance. The final augmented model achieved improved recall while maintaining high precision, making it suitable for practical fraud detection applications.

Additionally, deployment using Streamlit converted the trained model into a real-time interactive application, demonstrating the end-to-end implementation of an AI-powered fraud detection system.

4. CONCLUSION

4.1 Overall Learning Outcomes

The internship contributed significantly to both technical and analytical skill development. The major learning outcomes are as follows:

- Developed a strong foundation in Python programming and AI/ML concepts.
- Gained practical experience in data preprocessing, cleaning, transformation, and feature engineering.
- Learned to perform Exploratory Data Analysis (EDA) using visualization libraries such as Matplotlib and Seaborn.
- Implemented supervised learning algorithms for classification tasks and unsupervised learning algorithms for clustering and segmentation.
- Understood dimensionality reduction techniques such as Principal Component Analysis (PCA).
- Acquired knowledge of Artificial Neural Networks (ANNs), activation functions, and deep learning workflows using TensorFlow/Keras.
- Learned model evaluation techniques using accuracy, precision, recall, F1-score, confusion matrix, and silhouette score.
- Gained practical exposure to Generative AI through the implementation of CTGAN for synthetic data generation and augmentation.
- Understood the importance of handling class imbalance in real-world datasets.
- Learned the basics of deploying machine learning applications using Streamlit.
- Improved presentation, documentation, analytical thinking, and problem-solving skills through project-based learning.

4.2 Application of Work

The projects completed during the internship demonstrate the practical applications of AI/ML techniques across multiple domains:

- Bank Loan Approval Analysis: Helps financial institutions understand factors affecting loan approval decisions and supports risk assessment.
- Global Inflation Trends Analysis: Assists economists and policymakers in understanding long-term inflation patterns and economic instability.
- Customer Segmentation Project: Supports businesses in targeted marketing, customer profiling, and personalized recommendation systems.
- Food Preference Clustering: Helps food delivery and hospitality businesses understand consumer behaviour for improved personalization and customer engagement.
- Iris Flower Classification using ANN: Demonstrates the application of deep learning techniques for multiclass classification problems.
- Credit Card Fraud Detection using CTGAN: Shows the practical use of Generative AI and machine learning in fraud prevention, anomaly detection, and financial security systems.

These applications highlight how Artificial Intelligence and Machine Learning can be used to solve real-world problems, support decision-making, improve operational efficiency, and enhance predictive capabilities across industries.

Internship Certificate

SUMMARY

This internship report presents the work completed during the six-week Artificial Intelligence and Machine Learning internship conducted at Global Next Consulting India Pvt. Ltd. The internship combined theoretical understanding with practical implementation through five mini projects and one major capstone project.

The internship began with foundational concepts in Python programming, data preprocessing, and exploratory data analysis, followed by supervised and unsupervised machine learning techniques, clustering, dimensionality reduction, and neural networks. The mini projects focused on applying these concepts to solve domain-specific problems using real-world datasets.

The final capstone project, “Credit Card Fraud Detection Using Generative AI (CTGAN),” involved the use of synthetic data generation to address severe class imbalance in fraud detection. By augmenting the minority class using CTGAN-generated synthetic samples, the augmented model demonstrated improved fraud detection capability compared to the baseline model. The final model was also deployed using Streamlit as an interactive web application.

Overall, the internship provided valuable exposure to the end-to-end AI/ML workflow, including data analysis, preprocessing, model development, evaluation, augmentation using Generative AI, and deployment. The experience significantly enhanced technical knowledge, practical implementation skills, and understanding of real-world machine learning applications.

REFERENCES

1. Python Software Foundation. *Python Documentation*. Available at: https://docs.python.org/
2. NumPy Developers. *NumPy Documentation*. Available at: https://numpy.org/doc/
3. Pandas Documentation. Available at: https://pandas.pydata.org/docs/
4. Scikit-learn Documentation. Available at: https://scikit-learn.org/stable/documentation.html
5. TensorFlow Documentation. Available at: https://www.tensorflow.org/
6. Keras Documentation. Available at: https://keras.io/
7. Streamlit Documentation. Available at: https://docs.streamlit.io/
8. SDV Documentation – CTGAN. Available at: https://docs.sdv.dev/
9. Kaggle Datasets:
    - Loan Approval Dataset
    - Global Inflation Dataset
    - Mall Customers Dataset
    - Food Preference Dataset
    - Credit Card Fraud Detection Dataset