# AI-ML Internship

A Project Report submitted to the

**GLOBAL NEXT CONSULTING INDIA PVT LTD**

(Six – Week Internship Program)

By

**Y. Divya Reddy**

Under the Supervision of

***Dr. Anuradha Gupta***
***(Project Director)***

Submitted To :

**Global Next Consulting India Pvt. Ltd.**

Duration of Internship :

**23-March-2026 to 7-May-2026**


May 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 / Machine Learning Intern** role, is the result of original work carried out by me under the guidance of **Ms. Anuradha Gupta** during the period from March 2026 to May 2026.

I further declare that this report represents an authentic record of my own work and does not contain any falsely fabricated ideas, data, models, results, 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.

Y. Divya Reddy

# CERTIFICATE

This is to certify that the project report entitled **"AI-ML Internship Report"** has been carried out by **Y. Divya Reddy**, a AI/ML Intern specializing in **Data Analytics, Machine Learning**, and **Artificial Intelligence**. 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.

**Y. Divya Reddy**

# 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 these five minor projects as per each tool and one major project, aimed at developing practical skills in data handling, statistical analysis, and visualization.

A key project involved credit card fraud detection, where machine learning techniques were applied to identify fraudulent transactions from imbalanced datasets. Other projects focused on analyzing datasets, extracting insights, and building predictive models using Python and machine learning libraries such as Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn. Overall, the internship enhanced my technical skills in AI-ML and strengthened my analytical thinking, problem-solving ability, and data-driven decision-making skills.

# 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 Artificial Intelligence and Machine Learning tools and techniques, especially using Python (Google Colab, Jupyter Notebook), TensorFlow, Keras, Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn.

• To work on real-world datasets and develop meaningful machine learning models, data visualizations, and analytical insights.

• To learn data preprocessing, data cleaning, feature engineering, synthetic data generation, model training, and performance evaluation using machine learning and deep learning techniques.

• To enhance analytical thinking, problem-solving ability, effective communication, and presentation skills through weekly minor projects and a major end project.

# Chapter 2 - Projects

## 2.1 Credit Card Fraud Detection Trends (Week 1)

Credit card fraud detection is an important application of data analytics in the financial sector, where identifying fraudulent transactions helps reduce financial losses and improve transaction security. This project focuses on analysing credit card transaction trends using Exploratory Data Analysis (EDA) techniques.

The goal of the project is to understand transaction patterns, identify fraud trends, and explore the characteristics of fraudulent and non-fraudulent transactions using data analysis and visualization techniques.

The dataset contains 284,807 transaction records with 31 features, including transaction time, amount, anonymized variables (V1–V28), and a target variable indicating whether a transaction is fraudulent or legitimate.

Python libraries such as Pandas and NumPy were used for data understanding, preprocessing, and analysis, while visualization techniques were applied to study fraud distribution, transaction patterns, and feature relationships.

### 2.1.2 Objectives

#### ➢ Primary Objectives

• To understand and analyze the structure of the credit card transactions dataset.
• To perform data cleaning and preprocessing for effective analysis.
• To identify fraud transaction trends and behavioural patterns.
• To visualize transaction distributions and fraud occurrences using charts and graphs.
• To derive meaningful insights that support fraud detection systems.

#### ➢ Specific Analytical Goals

• Analyse the distribution of fraudulent and non-fraudulent transactions.
• Study transaction amount and time patterns related to fraud.
• Explore relationships among anonymized features (V1–V28).

• Identify data imbalance between fraud and legitimate transactions.

• Generate visual insights using statistical summaries and plots.

## 2.1.3 Methodology

### a) Dataset Preparation

➢ **Loaded the credit card transactions dataset containing 284,807 records and 31 features.**

➢ **Verified and standardized data types for numerical variables such as:**

• Transaction Time

• Transaction Amount

• Anonymized Features (V1–V28)

• Fraud Class Label

➢ **Checked for missing and inconsistent values.**

➢ **Performed data understanding and exploratory analysis to study:**

• Fraud vs Non-Fraud transactions

• Transaction amount distribution

• Time-based transaction behaviour

### b) Analysis Techniques

➢ **Used Pandas and NumPy libraries for:**

• Data handling and preprocessing

• Statistical summaries

• Fraud transaction analysis

➢ **Applied visualization techniques to analyse:**

• Fraud distribution

• Transaction amount trends

• Time distribution of transactions

• Feature correlations

➢ **Compared fraudulent and legitimate transactions using charts and summary statistics.**

### c) Visualization and Analysis

Interactive visual analysis was performed using:

• Fraud vs Non-Fraud Distribution Charts

• Transaction Amount Distribution

• Time-based Transaction Analysis

• Correlation Heatmaps

• Histograms and Scatter Plots

• Statistical Summary Tables

## 2.1.4 Results and Insights

### a) Dataset Insights

• Total transaction records: 284,807

• Total features: 31

• Fraudulent transactions are very small compared to legitimate transactions.

• The dataset is highly imbalanced, making fraud detection a challenging classification problem.

### b) Fraud Distribution Insights

• Legitimate transactions dominate the dataset, while fraud cases represent only a tiny percentage.

• Class imbalance highlights the importance of using proper fraud detection techniques and evaluation metrics.

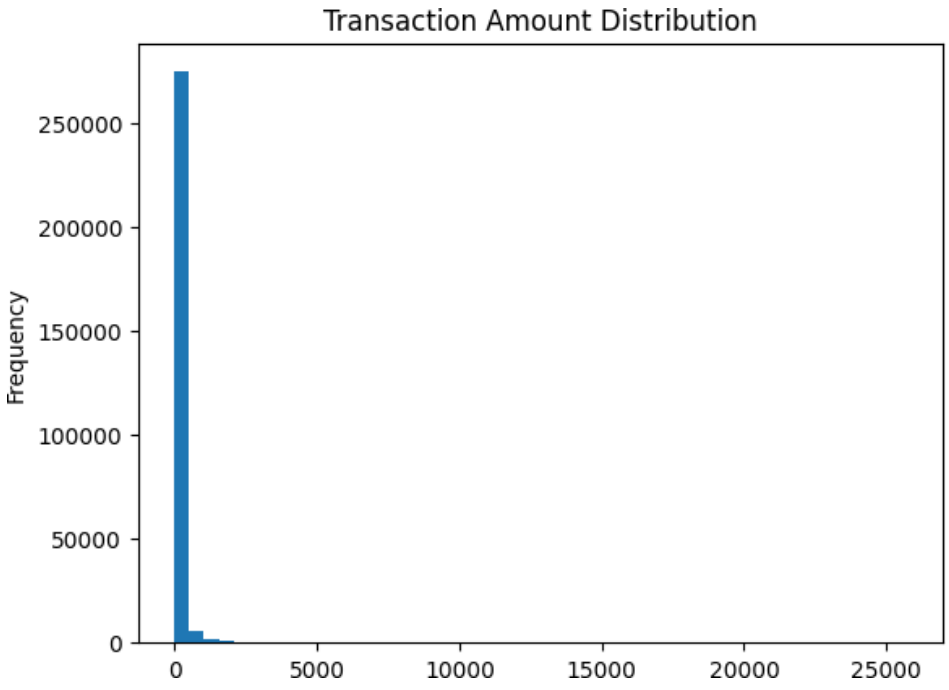
• Fraud detection models must focus on identifying rare fraudulent patterns without affecting genuine transactions.

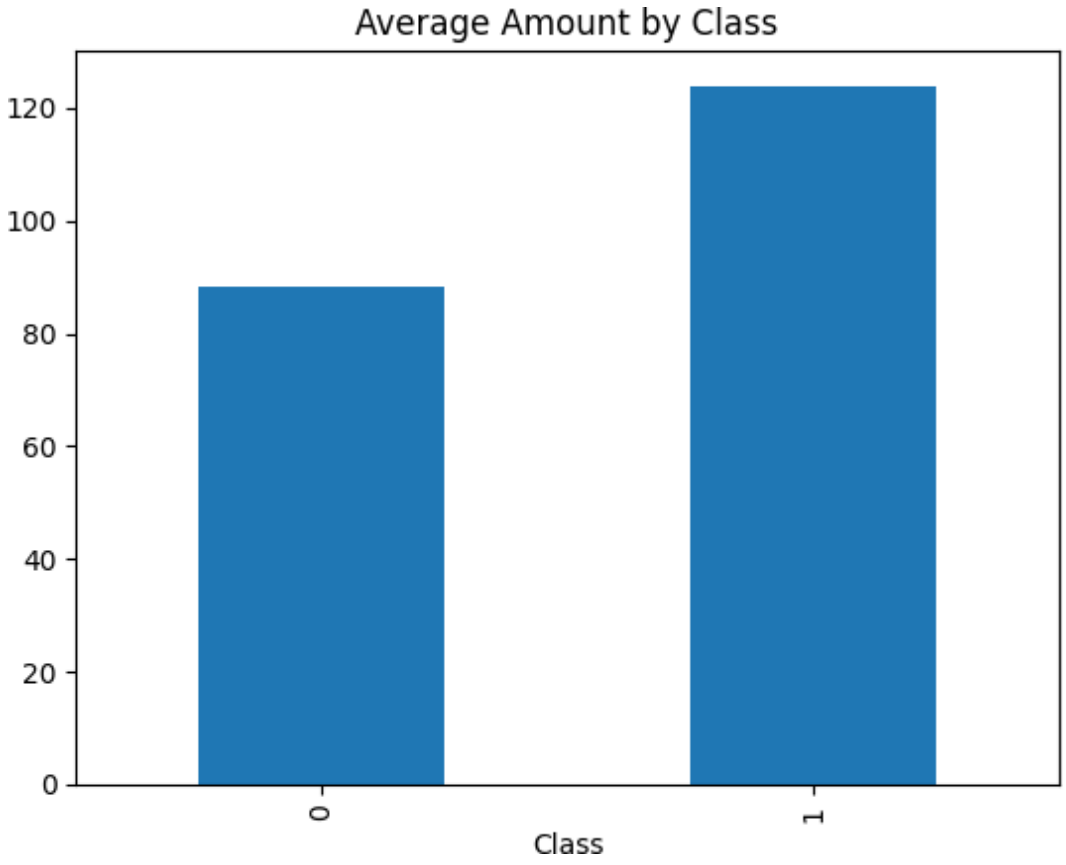
## c) Transaction Amount Insights

• Fraudulent transactions occur across multiple transaction amount ranges.

• Both low-value and high-value transactions can be fraudulent.

• Transaction amount alone is not sufficient to accurately identify fraud patterns.



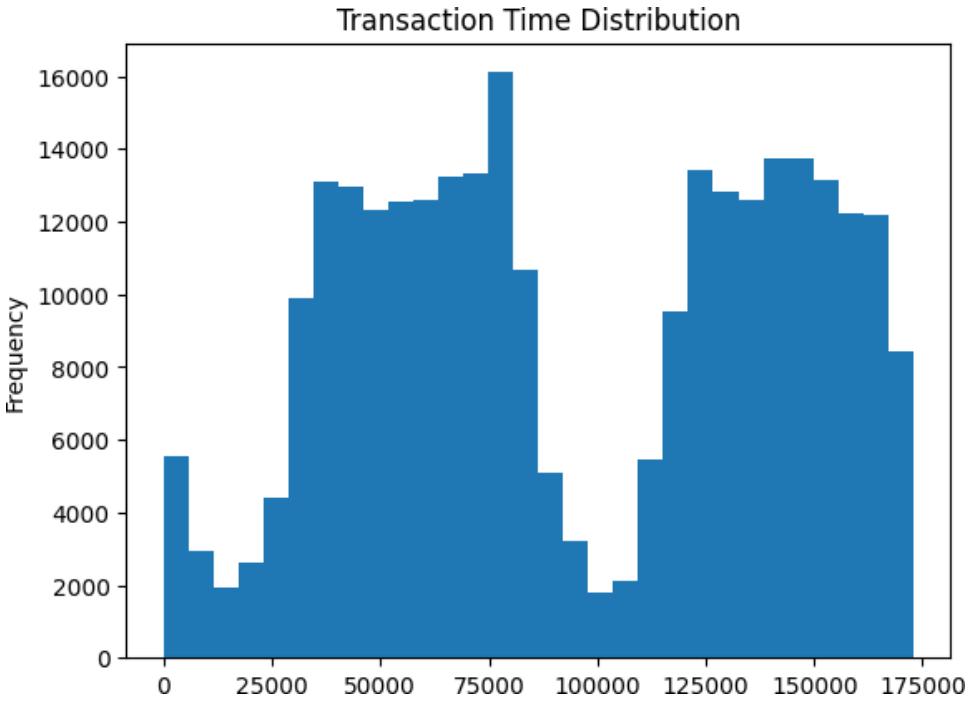



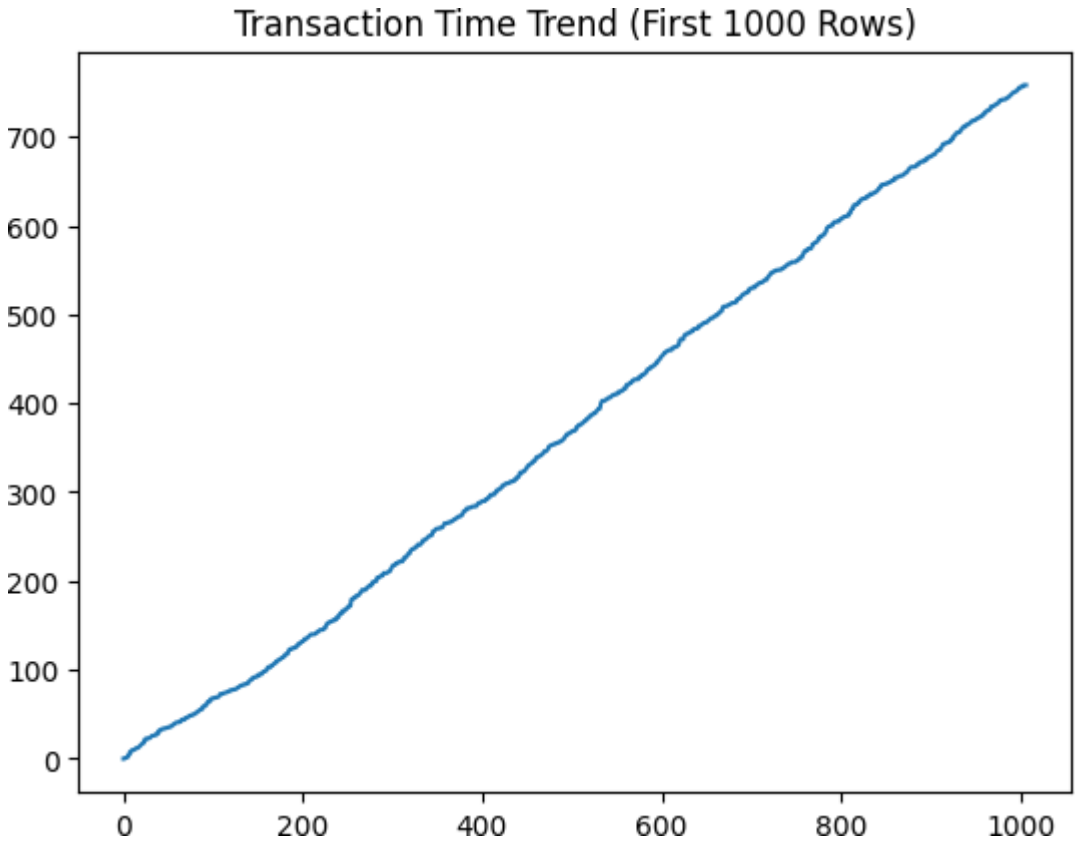
## d) Time-Based Insights

• Fraudulent activities are distributed across different time intervals.

• Certain time periods show slightly higher fraud occurrences, but no fixed pattern exists.

• Time-related features contribute better when combined with other transaction variables.



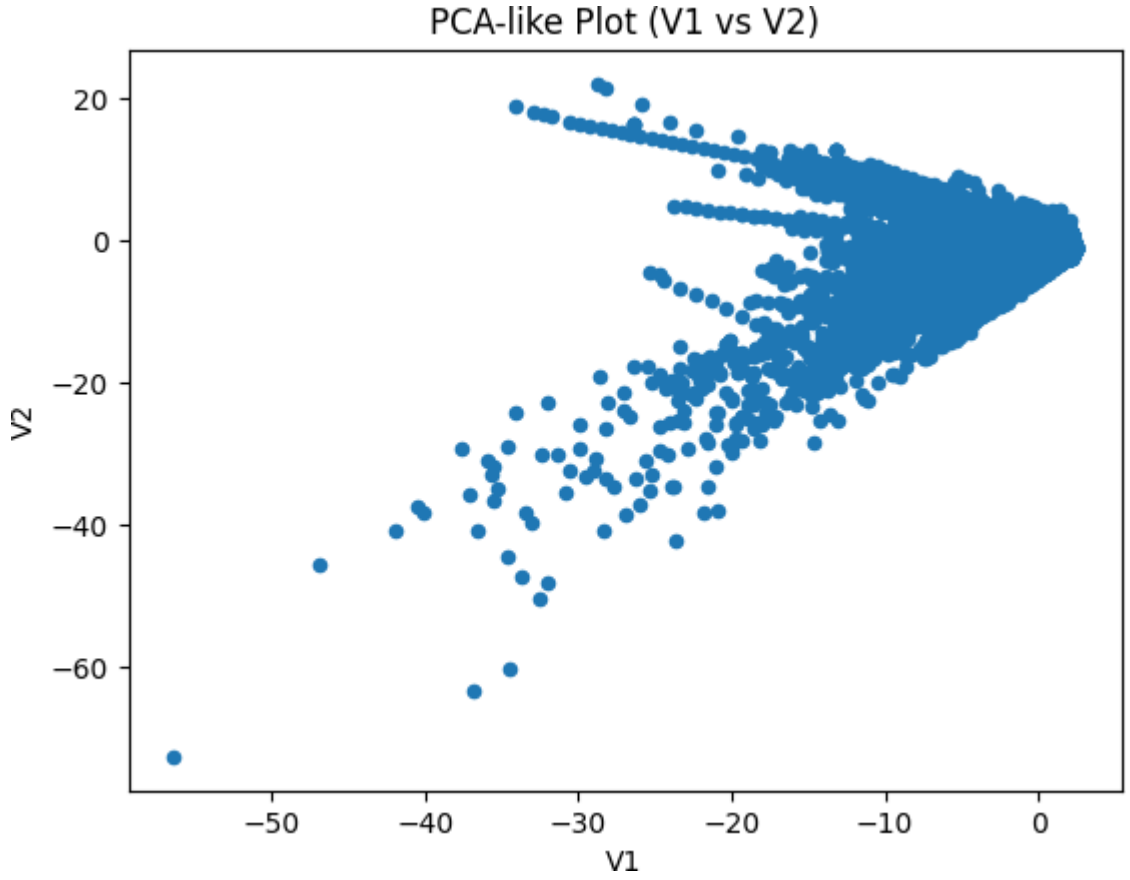
## e) Feature Correlation Insights

• Several anonymized features (V1–V28) show strong relationships with fraud behaviour.

• Correlation heatmaps help identify important variables influencing fraudulent transactions.

• Features with higher positive or negative correlations contribute significantly to fraud prediction.

| | Time | Amount | Class |
|---|---|---|---|
| **Time** | 1.000000 | -0.010559 | -0.012359 |
| **Amount** | -0.010559 | 1.000000 | 0.005777 |
| **Class** | -0.012359 | 0.005777 | 1.000000 |

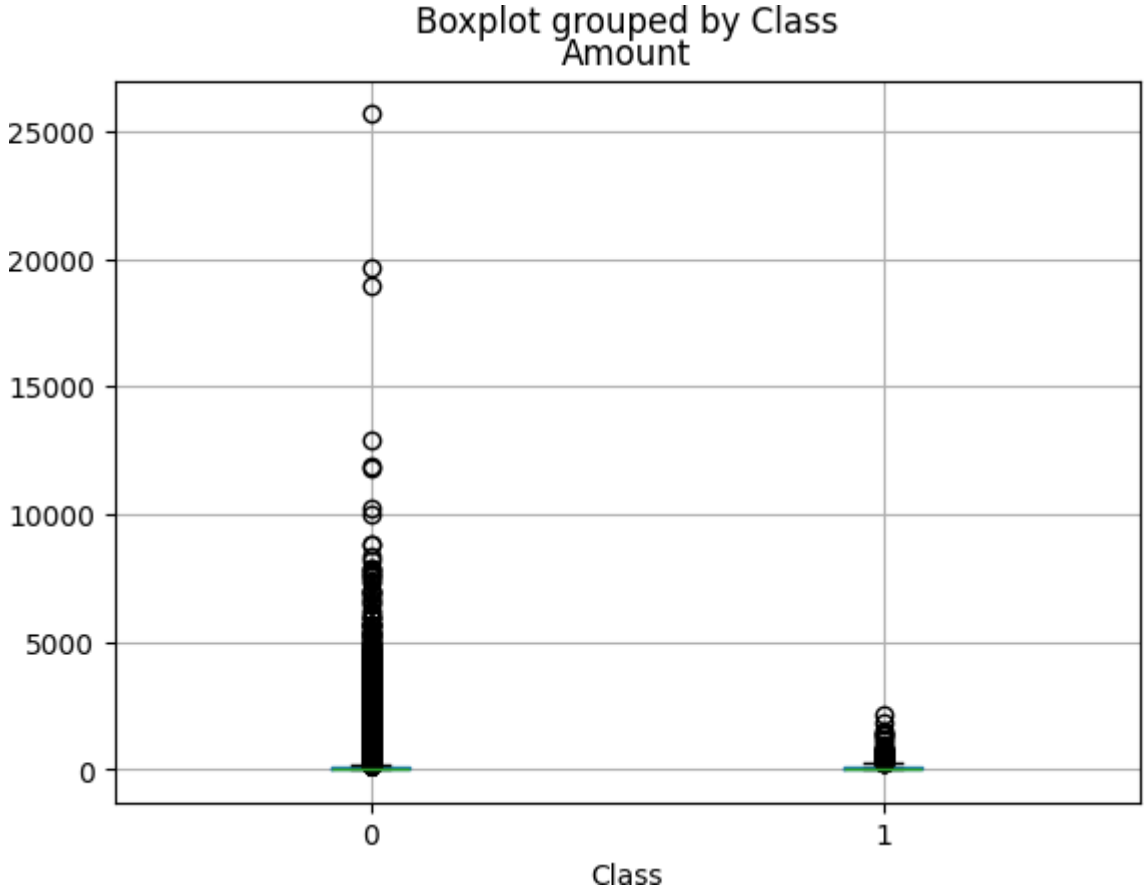



## f) Data Visualization Insights

• Histograms and boxplots reveal differences between fraudulent and non-fraudulent transaction distributions.

• Scatter plots and correlation analysis help identify hidden transaction patterns.

• Visualization techniques improve understanding of transaction behaviour and fraud trends.




## g) Machine Learning Insights

• Fraud detection models perform better after handling class imbalance.

• Feature scaling improves model performance and training efficiency.

• Machine learning algorithms can successfully identify suspicious transaction behaviour using transaction patterns and correlations.

## h) Overall Fraud Detection Insights

• Fraud detection requires analysing multiple transaction features together rather than relying on a single parameter.

• Imbalanced datasets require careful preprocessing and evaluation techniques.

• The project demonstrates how machine learning and data analysis techniques can help financial institutions identify fraudulent activities more effectively.

## 2.1.5 Recommendations

➢ **Improve Fraud Monitoring Systems**

• Implement advanced monitoring techniques to identify suspicious transaction patterns in real time.

• Use multiple transaction features together for better fraud identification accuracy.

➢ **Focus on Imbalanced Data Handling**

• Apply suitable sampling and balancing techniques while building fraud detection models.

• Ensure rare fraud transactions are properly represented during analysis and model training.

➢ **Enhance Transaction Analysis**

• Continuously monitor unusual transaction amounts and abnormal transaction behaviours.

• Analyse transaction patterns across different time intervals for early fraud detection.

➢ **Improve Feature Analysis**

• Study relationships among multiple transaction variables to identify hidden fraud patterns.

• Use correlation analysis and visualization techniques for better understanding of transaction behaviour.

➢ **Strengthen Financial Security Systems**

• Integrate data analytics and fraud detection mechanisms into banking systems.

• Support proactive fraud prevention using real-time analytical insights.

## 2.1.6 Conclusion

The Credit Card Fraud Detection Trends project demonstrates how Exploratory Data Analysis (EDA) can help identify fraud patterns in financial transactions. The analysis showed that fraudulent transactions are rare and cannot be detected using a single factor alone. By analysing multiple transaction features together, meaningful insights can be generated to support effective fraud detection and improve financial security systems.

# 2.2 Traffic Accident Analysis (Week 2)

## 2.2.1 Introduction

Traffic accidents have become a major public safety concern, causing injuries, fatalities, and economic losses across the world. Understanding the causes and patterns of accidents is essential for improving road safety and reducing accident severity. This project, **Traffic Accident Analysis**,

focuses on analysing traffic accident data using Exploratory Data Analysis (EDA) techniques to identify trends, accident patterns, and major contributing factors.

The dataset contains information related to driver details, vehicle information, road conditions, environmental factors, accident severity, casualties, and accident causes. Variables such as driver age, driving experience, weather conditions, road type, lighting conditions, vehicle ownership, and accident severity were analysed to understand their impact on accident occurrence.

This project focuses on analysing traffic accident data using Python libraries such as Pandas, Matplotlib, Seaborn, and Plotly. Through data cleaning, preprocessing, visualization, and correlation analysis, the goal was to uncover meaningful insights regarding road accidents and safety improvement measures.

## 2.2.2 Objectives

- To collect and analyse traffic accident data containing driver, vehicle, road, and environmental information.
- To clean and organize accident data containing parameters such as driver details, vehicle type, road conditions, weather conditions, casualties, and accident severity.
- To handle missing values, duplicate records, and inconsistent categorical data.
- To analyse the relationship between accident severity and factors such as weather, lighting, driver age, and driving experience.
- To identify the most common causes and locations of accidents.
- To compare accident severity across different vehicle types and junction types.
- To create visualizations for understanding accident patterns and trends.
- To generate insights that help improve road safety and accident prevention strategies.

## 2.2.3 Methodology

### ❖ Python Phase– Database Cleaning, Processing and Insights

- In the first phase, I worked with Python to clean the dataset, preprocess the data, and extract meaningful insights.

***Data Collection:***

I collected the traffic accident dataset from Kaggle. The dataset contained accident-related information such as driver details, vehicle information, weather conditions, road conditions, casualty severity, accident severity, and causes of accidents.

***Data Cleaning:***

I performed multiple cleaning steps:

¬ Removed duplicate rows from the dataset.

¬ Handled missing and null values using suitable imputation techniques.

¬ Standardized categorical values and column names for consistency.

¬ Converted categorical variables into numerical format using one-hot encoding.

¬ Checked and corrected data types for numerical and categorical features.

¬ Performed preprocessing tasks to prepare the data for visualization and analysis.

## ***Final Clean Dataset:***

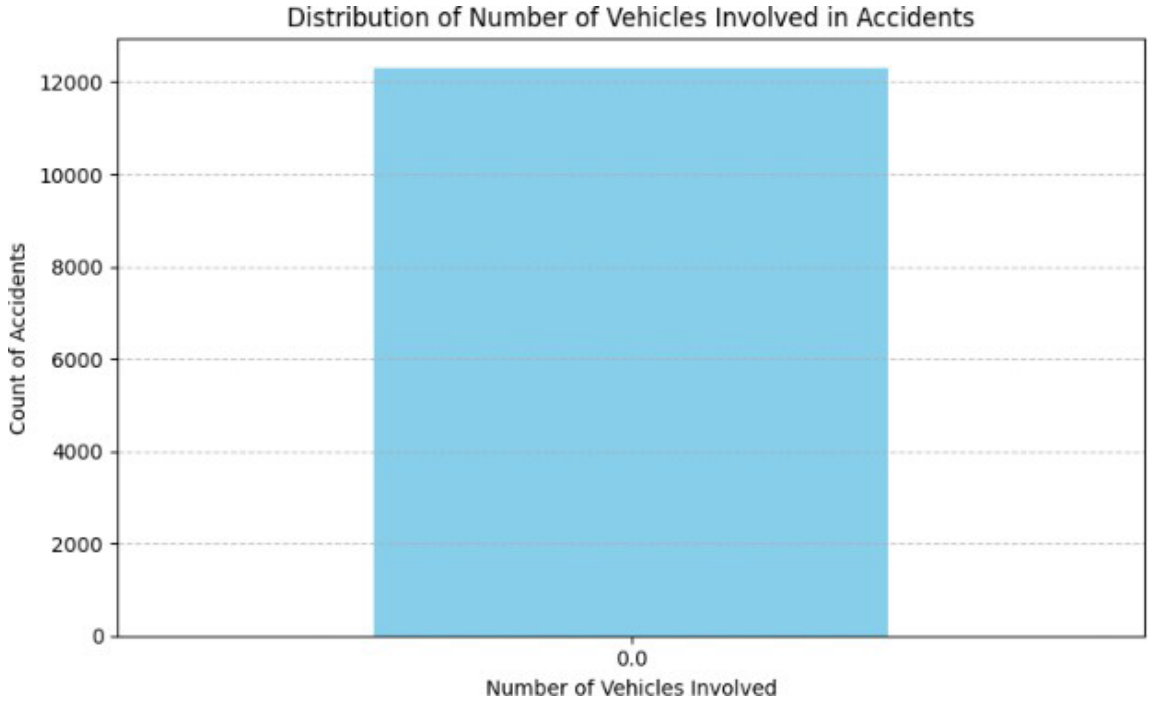
After preprocessing and transformations, the cleaned dataset was used for exploratory data analysis, visualization, and correlation analysis.

## ***EDA Insights:***

### *i) Distribution of Vehicles Involved in Accidents*

¬ Most accidents involve 1 or 2 vehicles.

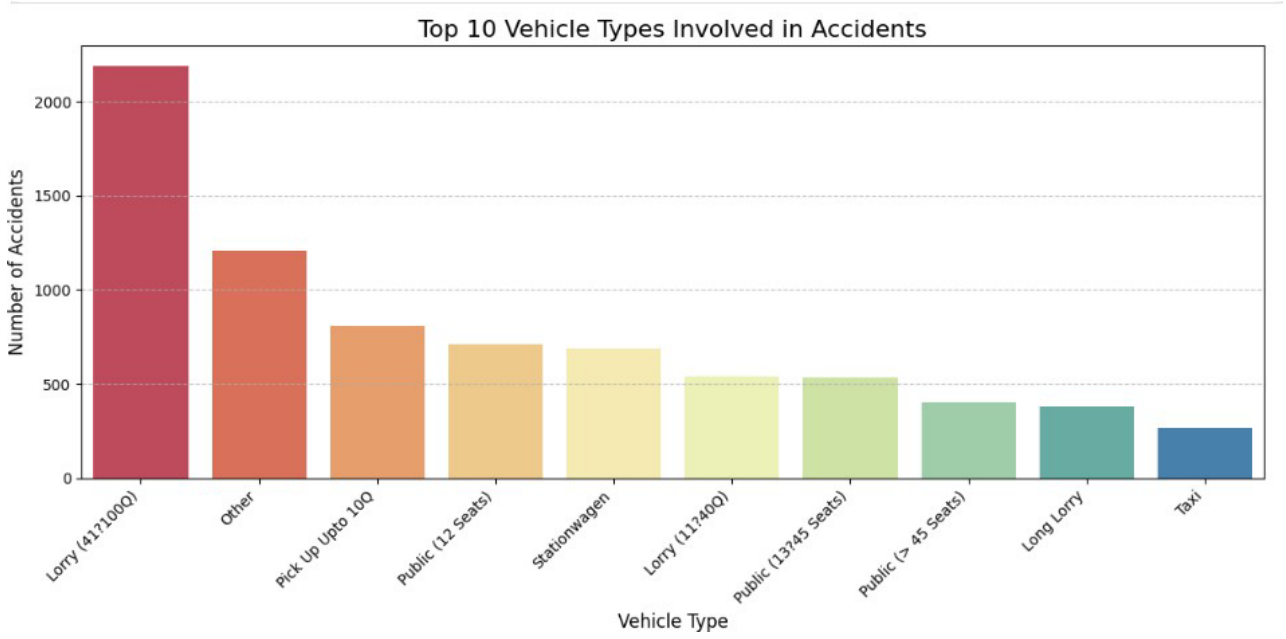
¬ The number of accidents decreases as the number of vehicles involved increases.



### *ii) Top Vehicle Types Involved in Accidents*

¬ Lorries, especially heavy vehicles, are involved in the highest number of accidents.

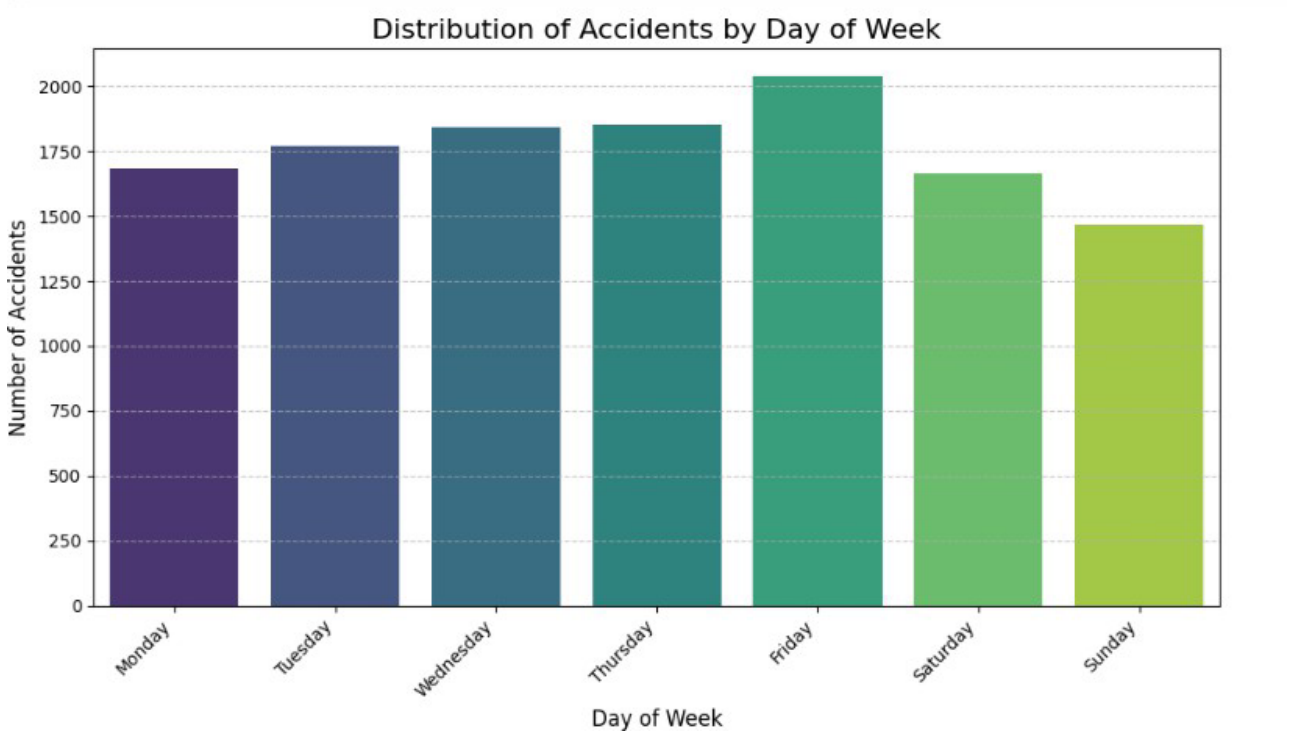
¬ Public transport vehicles also contribute significantly to accidents.




### *iii) Accidents by Day of Week*

¬ Thursdays and Fridays record the highest number of accidents.

¬ Weekends show comparatively lower accident frequency.

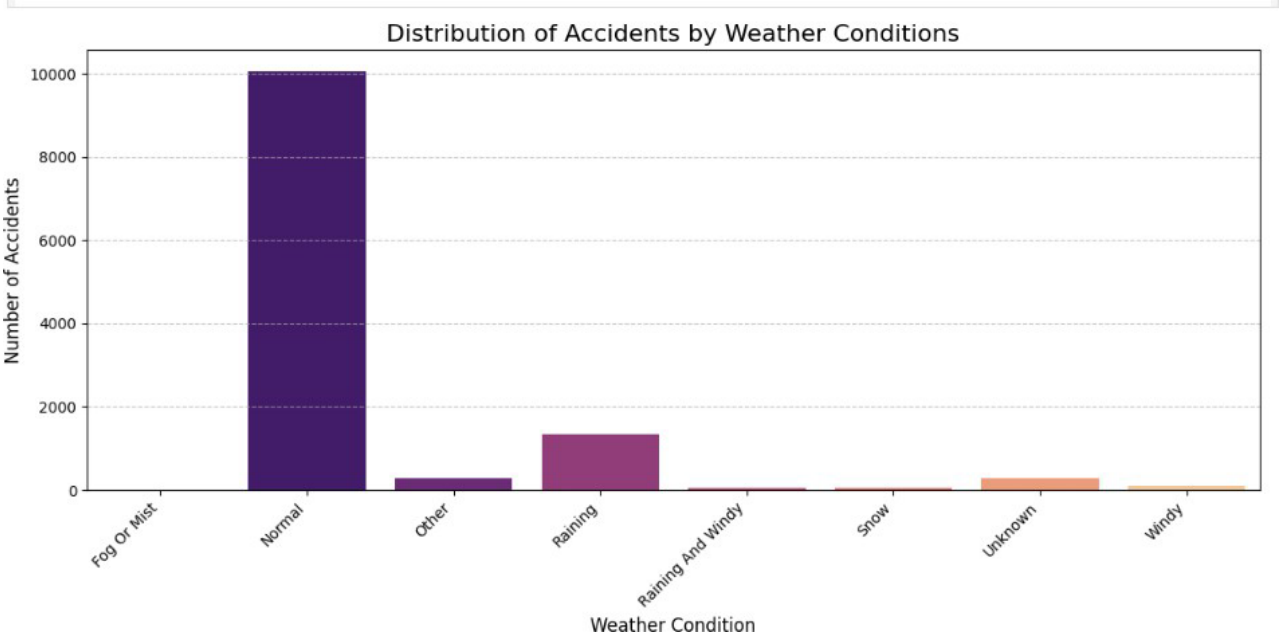


### *iv) Accidents by Weather Conditions*

¬ Most accidents occur under normal weather conditions.

¬ Rainy weather contributes to a significant number of accidents due to reduced visibility and slippery roads.

¬ Severe weather conditions such as fog and wind contribute to fewer accidents.

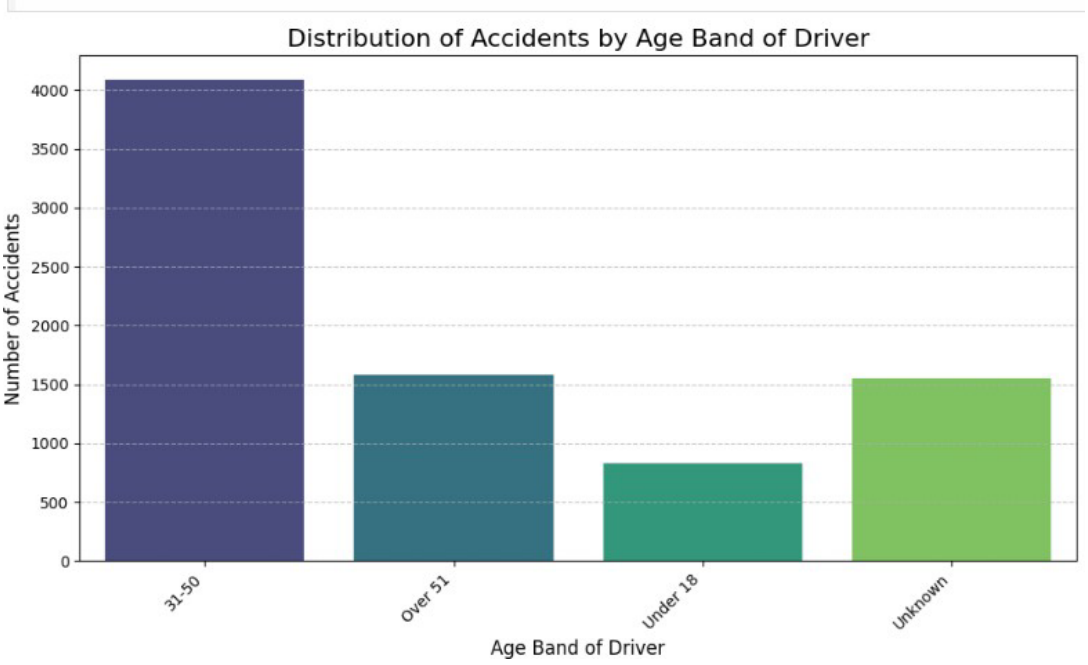



### *v) Accidents by Age Band of Driver*

¬ Drivers aged between 18–30 and 31–50 are involved in the majority of accidents.

¬ Drivers above 51 years have lower accident involvement.

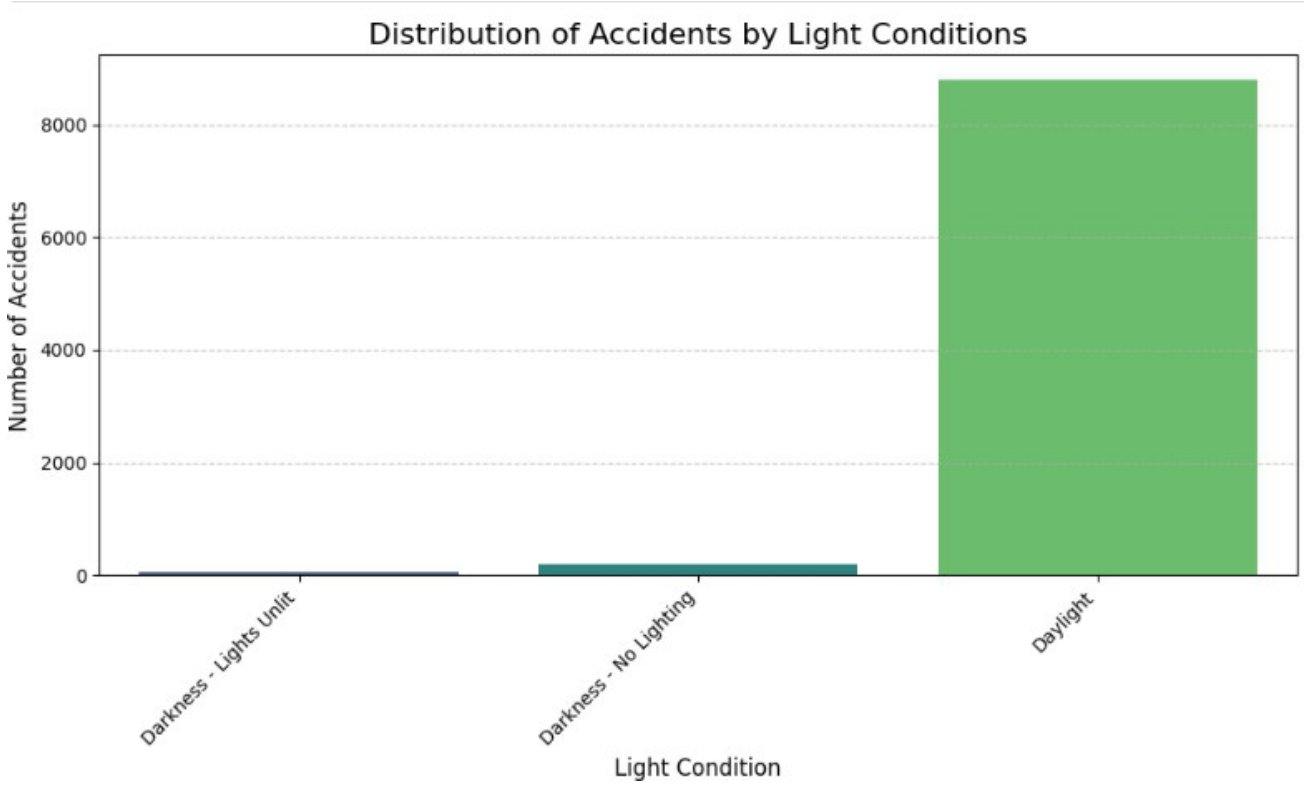
¬ A notable number of records contain unknown age information.



### *vi) Accidents by Light Conditions*

¬ Most accidents occur during daylight conditions.

¬ Poor lighting conditions during darkness contribute to a considerable number of accidents.

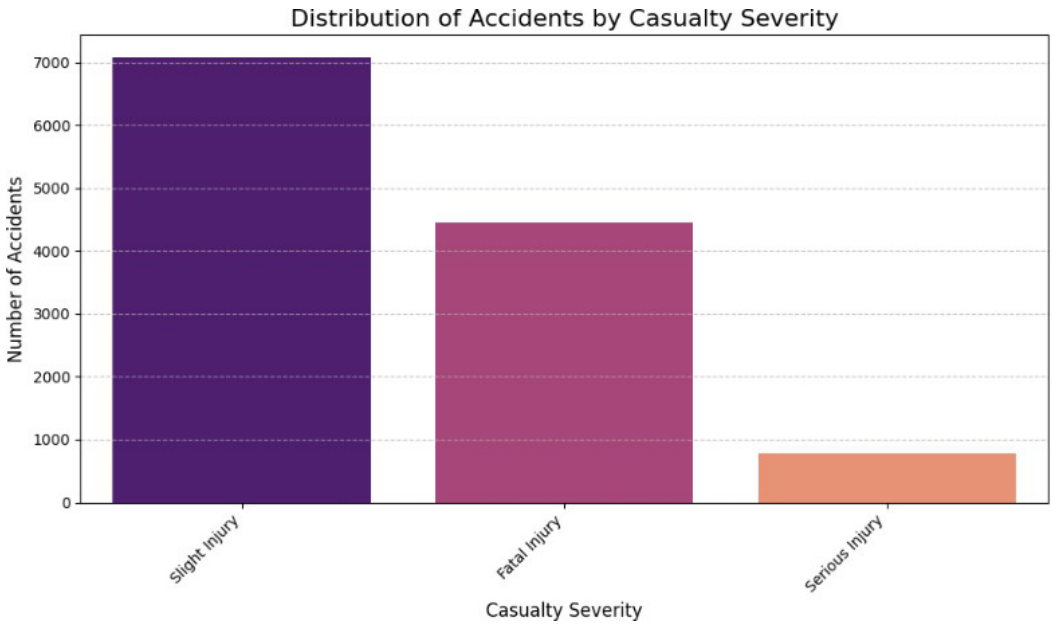



### *vii) Casualty Severity Analysis*

¬ Slight injuries are the most common accident outcome.

¬ Serious injuries are the second most common category.

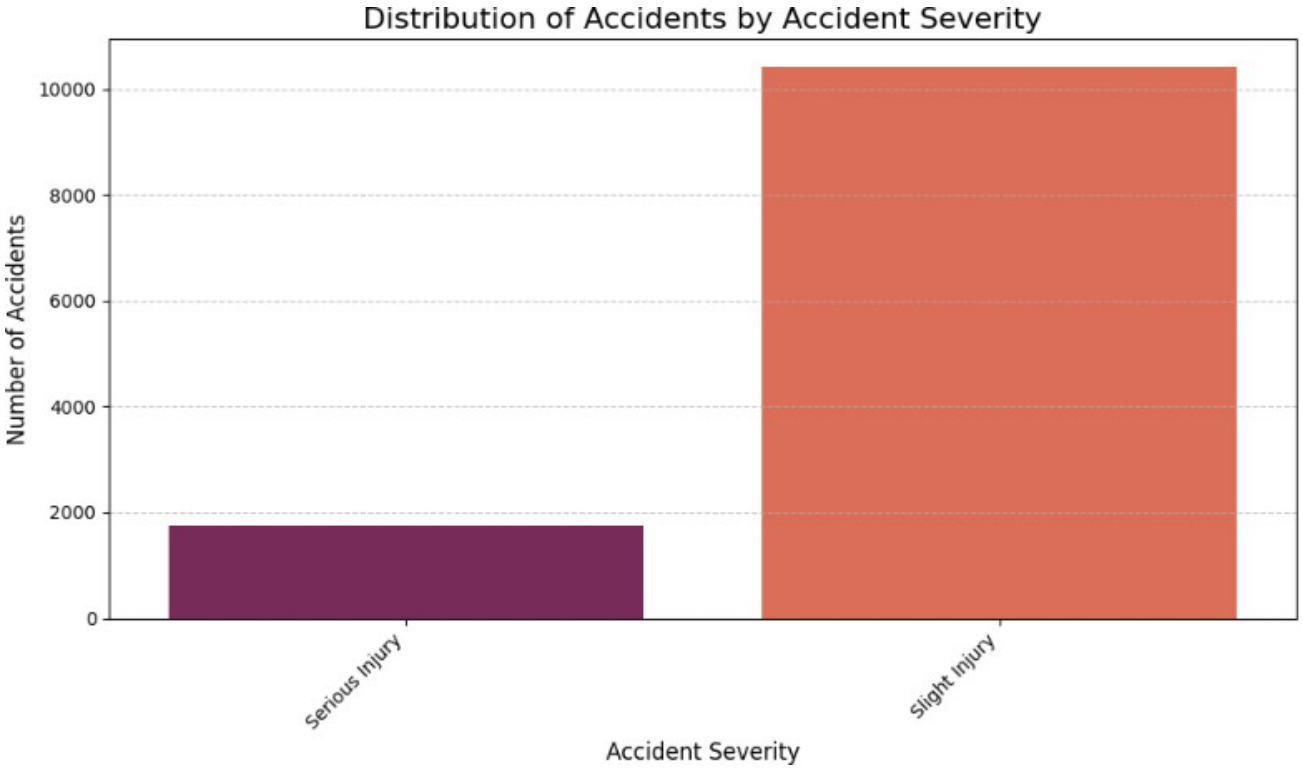
¬ Fatal injuries are relatively low but remain highly critical.



### *viii) Accident Severity Distribution*

¬ Majority of accidents result in slight injuries.

¬ Serious and fatal accidents occur less frequently but require attention for road safety planning.

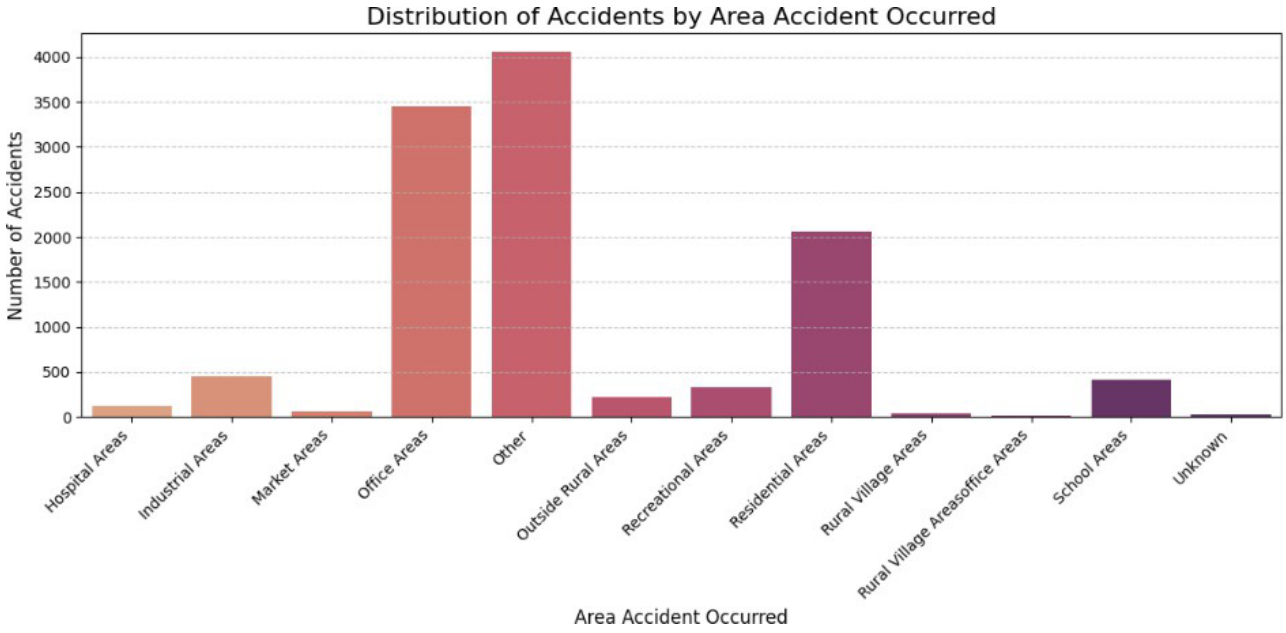



### *ix) Area-wise Accident Distribution*

¬ Residential and office areas record the highest number of accidents.

¬ School and market areas also show significant accident occurrence.

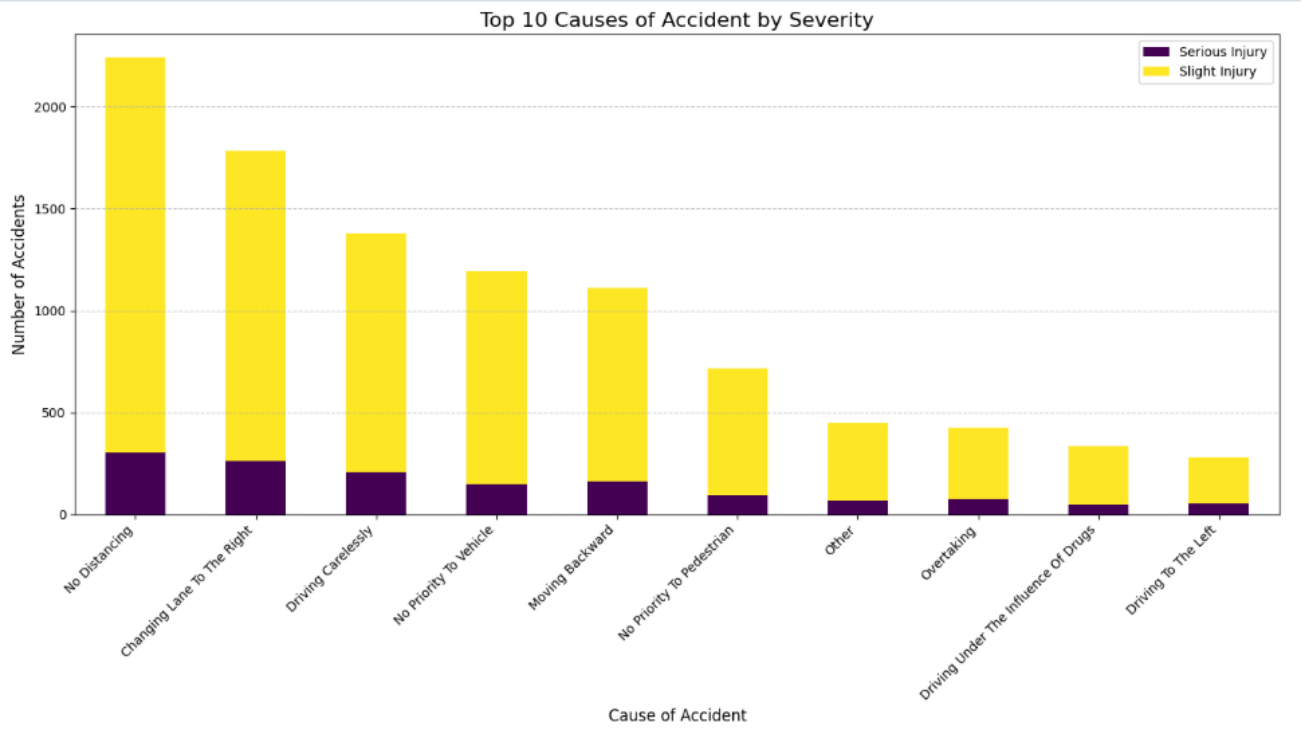
¬ Rural and industrial areas have comparatively fewer accidents.



### *x) Accident Severity by Cause of Accident*

¬ Driver-related causes contribute heavily to both serious and slight injuries.

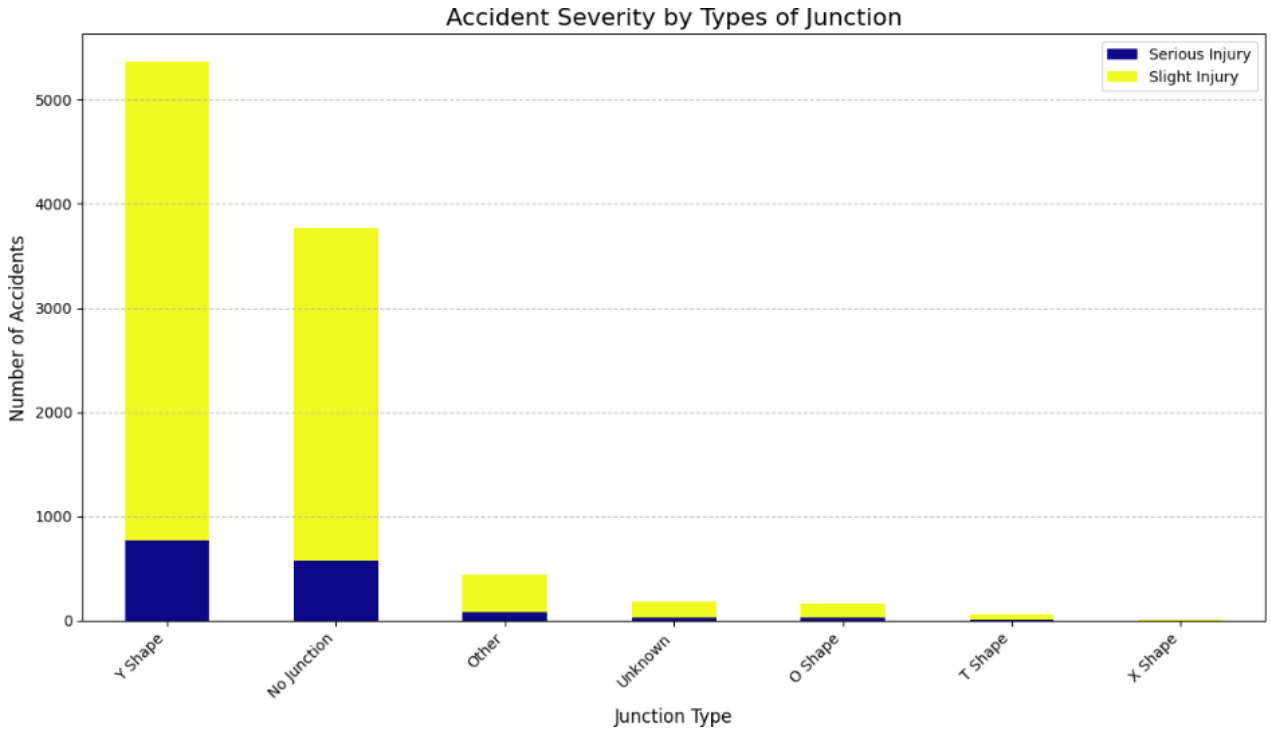
¬ Certain causes result in higher proportions of severe injuries.




### *xi) Accident Severity by Types of Junction*

¬ "No Junction" areas account for a large number of accidents.

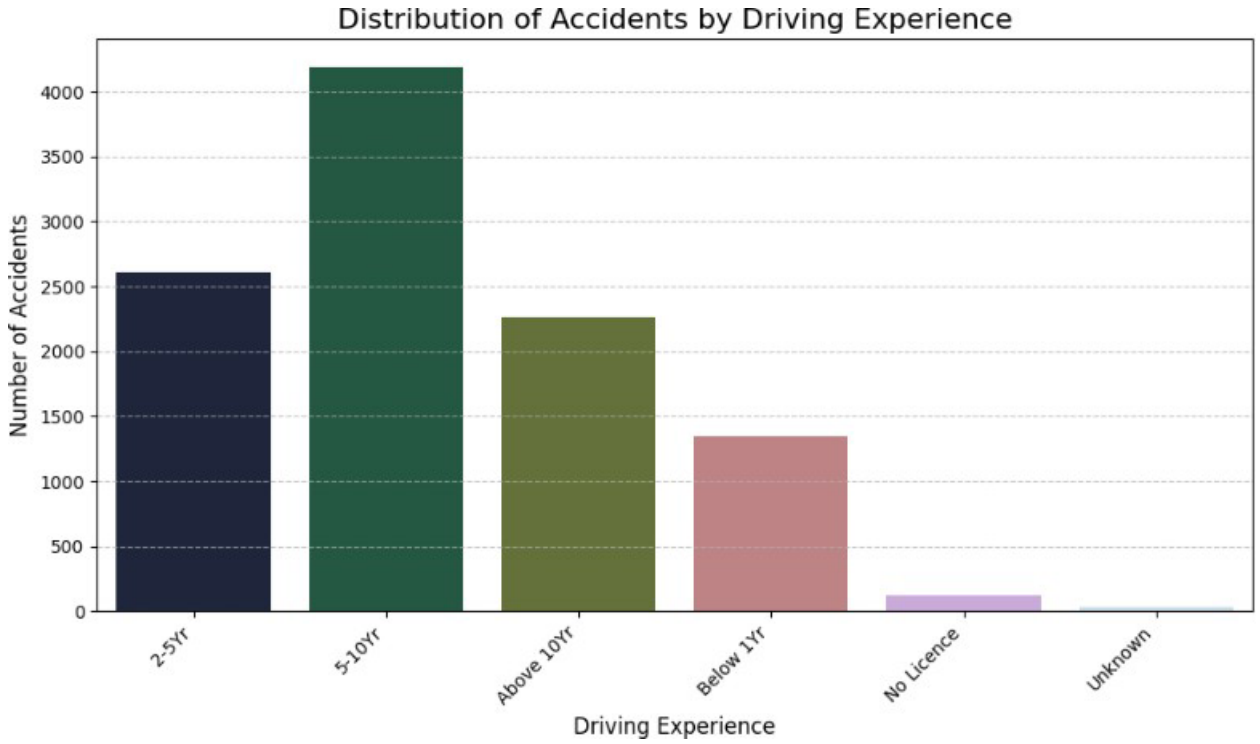
¬ X-shaped and T-shaped junctions contribute significantly to accident occurrence and severity.



### *xii) Accidents by Driving Experience*

¬ Drivers with 2–5 years and 5–10 years of experience are involved in more accidents.

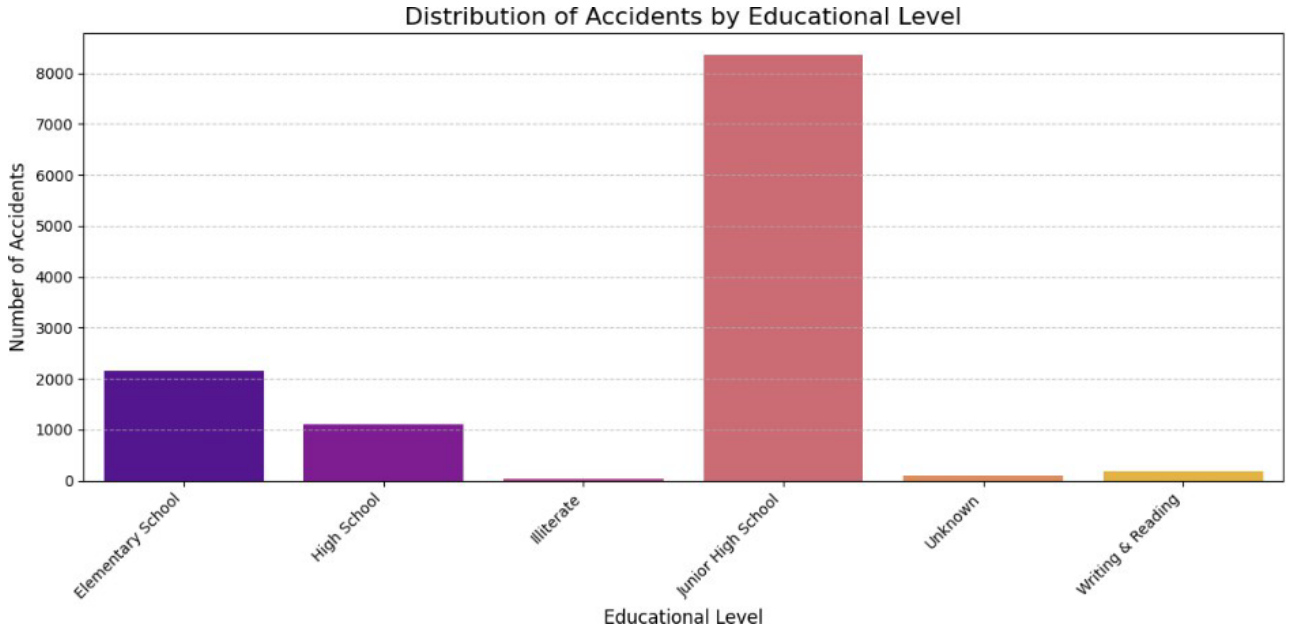
¬ Drivers with more than 10 years of experience show lower accident involvement.




### *xiii) Accidents by Educational Level*

¬ Drivers with elementary and junior high school education levels show higher accident involvement.

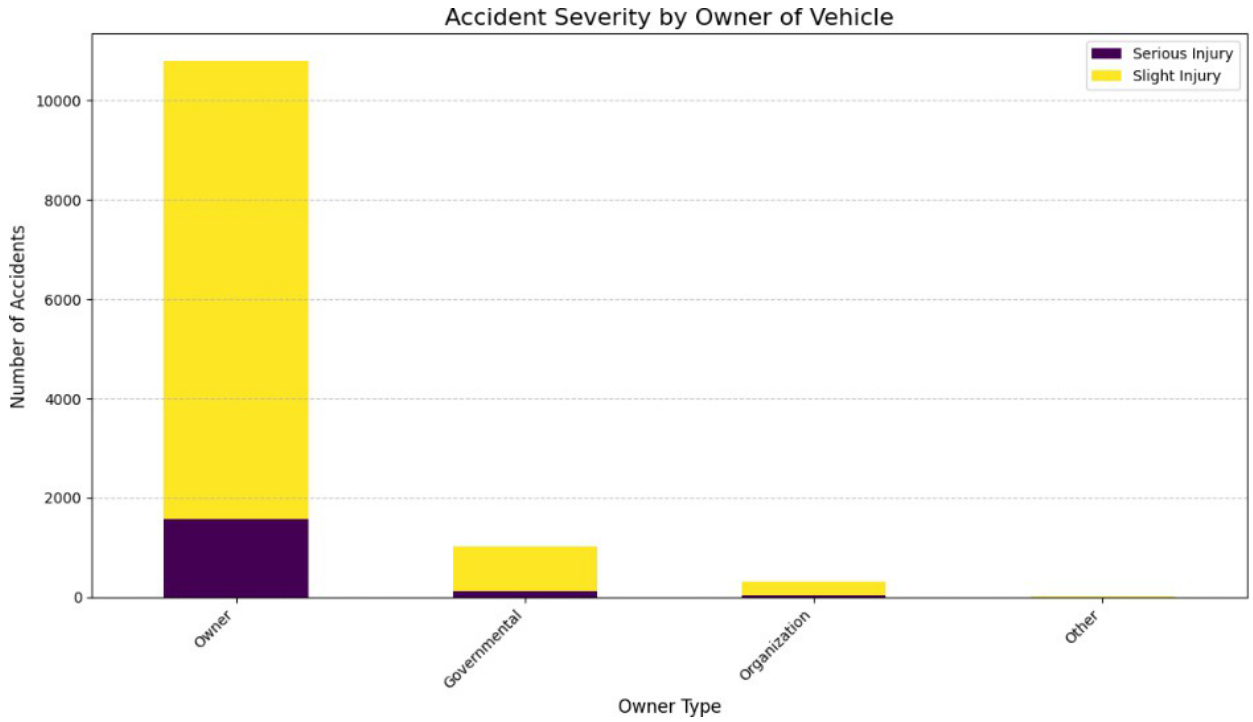
¬ The “Unknown” category indicates possible data collection gaps.



### *xiv) Accident Severity by Owner of Vehicle*

¬ Privately owned and governmental vehicles are involved in a large number of accidents.

¬ Slight injuries are more common across all ownership categories.

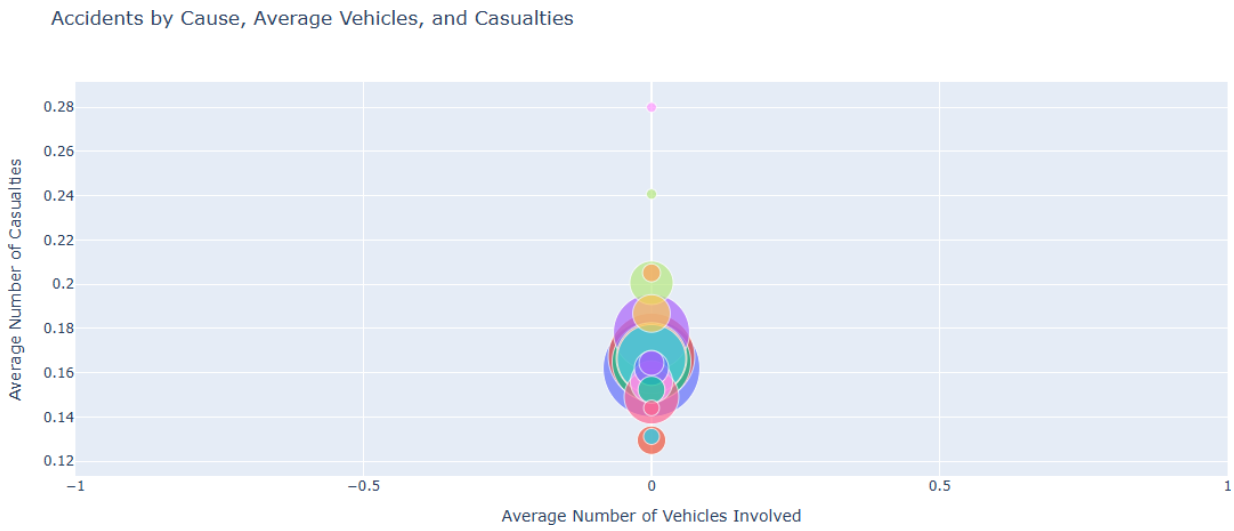



### *xv) Interactive Bubble Chart Analysis*

¬ Interactive bubble charts were created to analyse accident causes, casualties, and vehicle involvement.

¬ Larger bubbles represent causes with higher accident frequency.

¬ Bubble positions indicate average casualties and average vehicles involved.

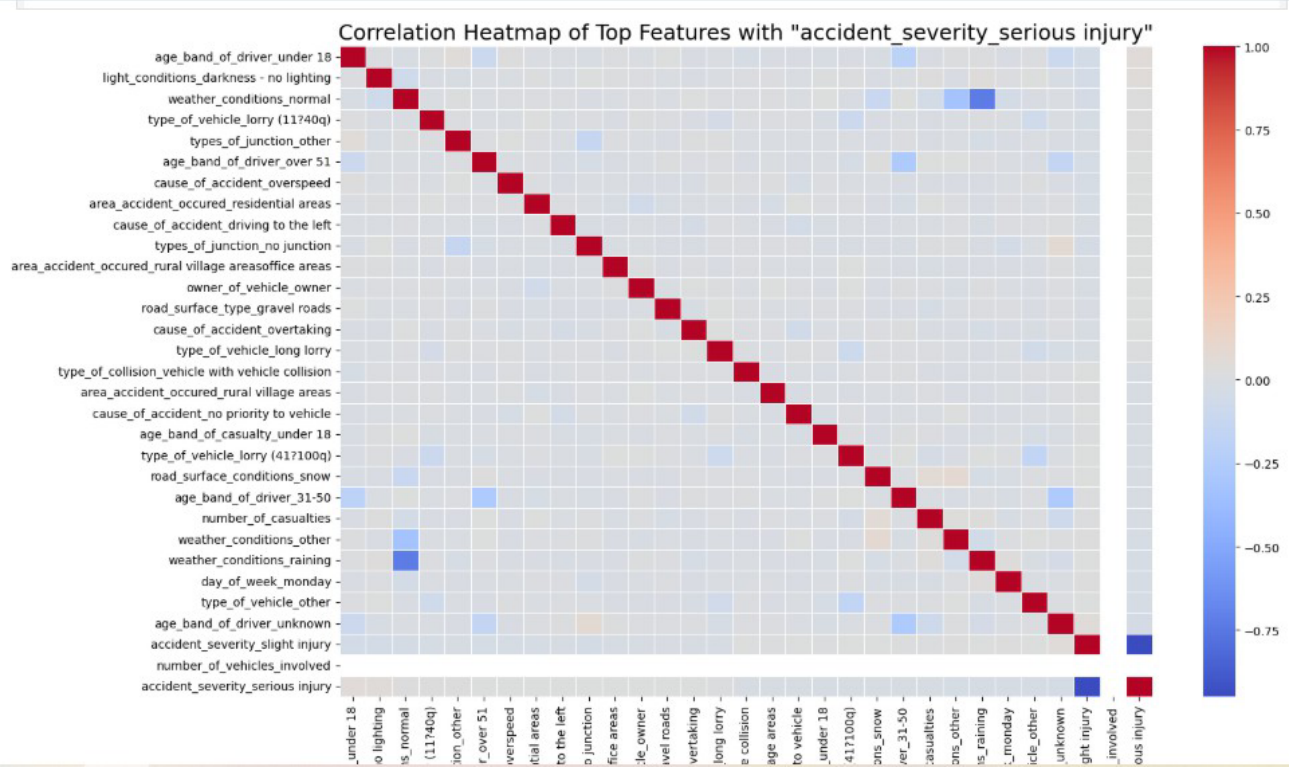


***xvi) Correlation Heatmap***

¬ Correlation analysis identified features strongly associated with serious injuries.

¬ High positive correlations indicate factors contributing to accident severity.

¬ Negative correlations highlight factors less associated with severe accidents.




## 2.2.4 Results and Insights

### i) Overall Accident Summary

¬ Majority of accidents result in slight injuries.

¬ Fatal accidents occur less frequently but remain a major public safety concern.

¬ Accident severity is strongly influenced by environmental and driver-related factors.

### ii) Vehicle Involvement Analysis

¬ Most accidents involve one or two vehicles.

¬ Heavy vehicles such as lorries contribute significantly to accident occurrence.

### iii) Driver Age and Experience Analysis

¬ Drivers aged 18–50 are involved in the highest number of accidents.

¬ Drivers with moderate driving experience (2–10 years) show higher accident frequency.

### iv) Weather and Environmental Impact

¬ Normal weather conditions account for the majority of accidents due to higher traffic movement.

¬ Rainy conditions increase accident risk because of reduced road grip and visibility.

### v) Light Condition Analysis

¬ Daylight conditions record the highest accident count.

¬ Poor lighting during nighttime increases accident severity risk.

### vi) Area and Junction Analysis

¬ Residential and office areas show the highest accident occurrence.

¬ Junctions such as X-shaped and T-shaped intersections require improved traffic management.

### vii) Accident Severity Analysis

¬ Slight injuries dominate accident outcomes.

¬ Serious injuries remain significant and are associated with risky driving behaviors and poor road conditions.

### viii) Correlation Analysis

¬ Correlation heatmaps helped identify key features affecting serious injuries.

¬ Strong relationships were observed between accident severity, casualties, and road-related conditions.

### ix) Interactive Visualization Insights

¬ Bubble charts effectively highlighted accident causes with high casualty involvement.

¬ Interactive visualizations improved understanding of accident severity and vehicle involvement patterns.

## ➢ EDA Phase Summary

The Traffic Accident Analysis project successfully used Python-based EDA techniques to analyse accident trends and identify important factors contributing to accidents. Visualizations and statistical analysis helped understand how driver behavior, environmental conditions, vehicle characteristics, and road infrastructure influence accident severity.

The project serves as an analytical tool for understanding traffic accident patterns and supports future road safety planning, awareness campaigns, and policy-making initiatives.

### 2.2.5 Conclusion

The Traffic Accident Analysis project demonstrates how Exploratory Data Analysis (EDA) techniques can be effectively used to understand accident patterns and identify key risk factors affecting road safety.

The analysis revealed that driver behavior, weather conditions, lighting conditions, road infrastructure, and vehicle types significantly influence accident occurrence and severity. Drivers aged between 18–50 and heavy vehicles such as lorries were found to be involved in a large number of accidents.

The study highlights that proper road safety awareness, improved traffic management, better road infrastructure, and safer driving practices can help reduce accident frequency and severity.

Overall, this project provides meaningful insights into accident trends and supports the development of effective strategies for improving road safety and minimizing traffic-related risks.

## 2.3 SEGMENTING CREDIT CARD USERS (Python & Machine Learning) (Week-3)

### 2.3.1 Introduction

Customer segmentation is an important business strategy that helps financial institutions understand customer behaviour and improve decision-making. Credit card companies manage customers with different spending habits, payment patterns, and credit usage behaviours. Analysing these behaviours manually is difficult due to the large volume of customer data.

This project, **Segmenting Credit Card Users**, focuses on grouping credit card customers into meaningful segments using Machine Learning techniques. The analysis uses customer transaction

and spending behaviour data to identify patterns such as high spenders, low-usage customers, installment users, and cash-advance dependent users.

The project is carried out using:

• Python for data preprocessing, cleaning, exploratory data analysis (EDA), feature scaling, PCA, and clustering

• Machine Learning algorithms such as K-Means Clustering for customer segmentation

These techniques help generate actionable business insights for customer targeting, marketing strategy, and risk analysis.

### 2.3.2 Objectives

The major objectives of this analysis are:

• To clean and preprocess credit card customer data using Python.

• To analyse customer spending and payment behaviour patterns.

• To perform feature scaling for better clustering performance.

• To reduce dimensionality using Principal Component Analysis (PCA).

• To apply K-Means Clustering for customer segmentation.

• To identify distinct customer groups based on financial behaviour.

• To generate visual insights that support business and marketing decisions.

• To understand relationships between spending, payments, and credit usage.

## 2.3.3 Methodology

The methodology follows a structured machine learning workflow including data preprocessing, feature engineering, clustering, and visualization.

### a) Python Phase – Data Cleaning, Preprocessing, and EDA

In the Python environment (Google Colab / Jupyter Notebook), the following steps were completed:

***Data Loading & Inspection***

• Imported dataset containing credit card customer information

• Checked datatypes, structure, null values, and duplicate records

• Analysed dataset dimensions and feature distributions

***Data Cleaning***

• Removed unnecessary columns such as customer ID where required

• Handled missing values using median imputation

• Standardized feature formats for consistency

• Checked outliers in spending and payment-related features

***Exploratory Data Analysis***

• Summary statistics for balance, purchases, payments, and credit limits

• Distribution analysis using histograms and boxplots

• Correlation analysis between financial variables

• Spending behaviour analysis using scatter plots and heatmaps

***Feature Engineering & Scaling***

• Applied StandardScaler for feature normalization

• Reduced dimensionality using PCA (Principal Component Analysis)

• Selected principal components for efficient clustering

***Machine Learning Phase***

• Applied Elbow Method to determine optimal clusters

• Implemented K-Means Clustering algorithm

• Grouped customers into meaningful behavioural segments

• Visualized clusters using PCA plots and scatter visualizations

## 2.3.4 Results & Insights

Below is a summary of the major analyses and insights generated during the project:

### i. Customer Spending Distribution Analysis

Insight:

Customer spending behaviour varies significantly across users. Some customers make very high purchases regularly, while others use credit cards minimally.

### ii. Balance vs Purchases Relationship (Scatter Plot)

Insight:

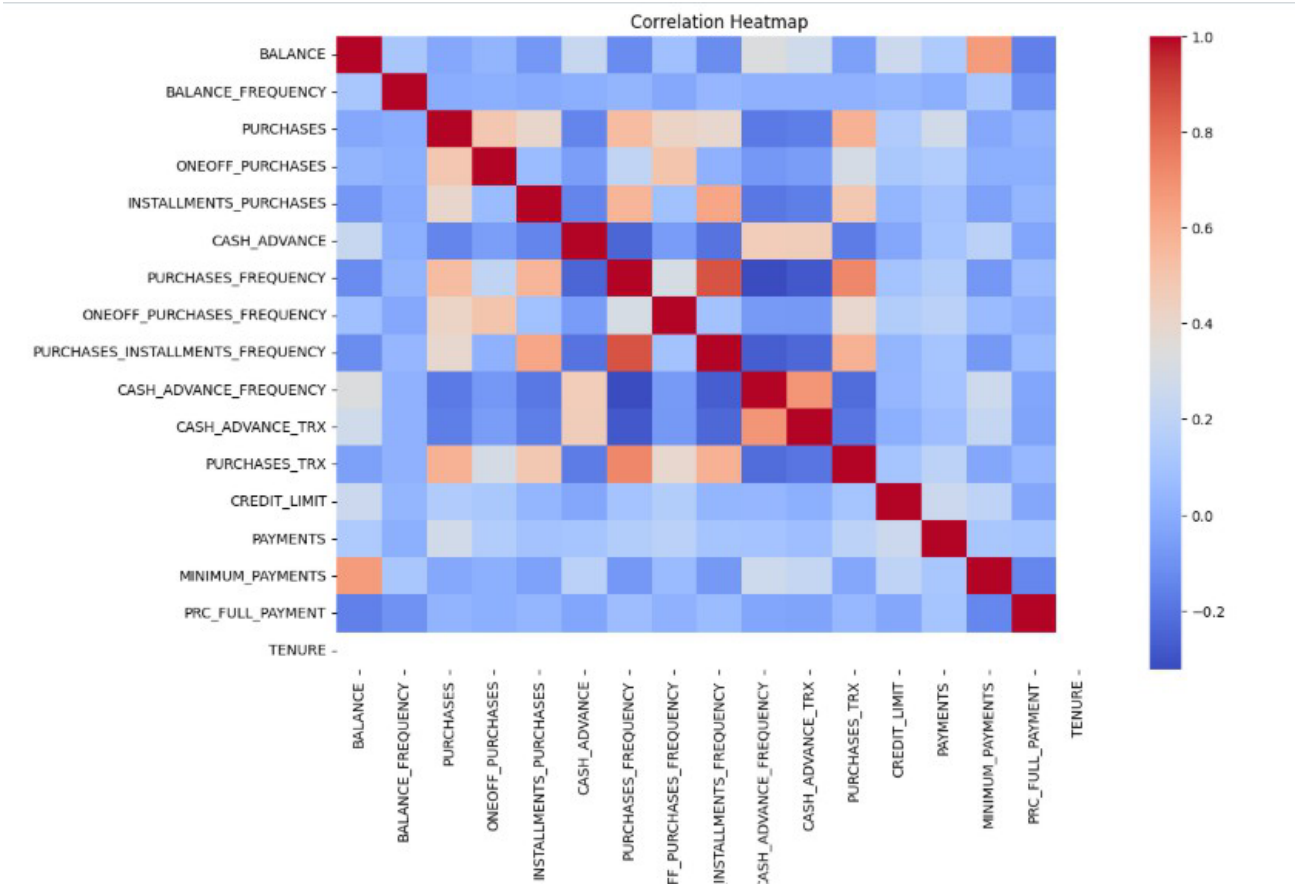
Customers with high balances generally show higher purchase activity. However, some customers maintain high balances with low purchase frequency, indicating possible debt accumulation patterns.

## iii. Credit Limit Distribution (Histogram)

Insight:

Most customers fall within medium credit limit ranges, while a small percentage possess very high credit limits associated with premium customer categories.

## iv. Correlation Heatmap of Financial Features




Insight:

Strong correlations exist between:

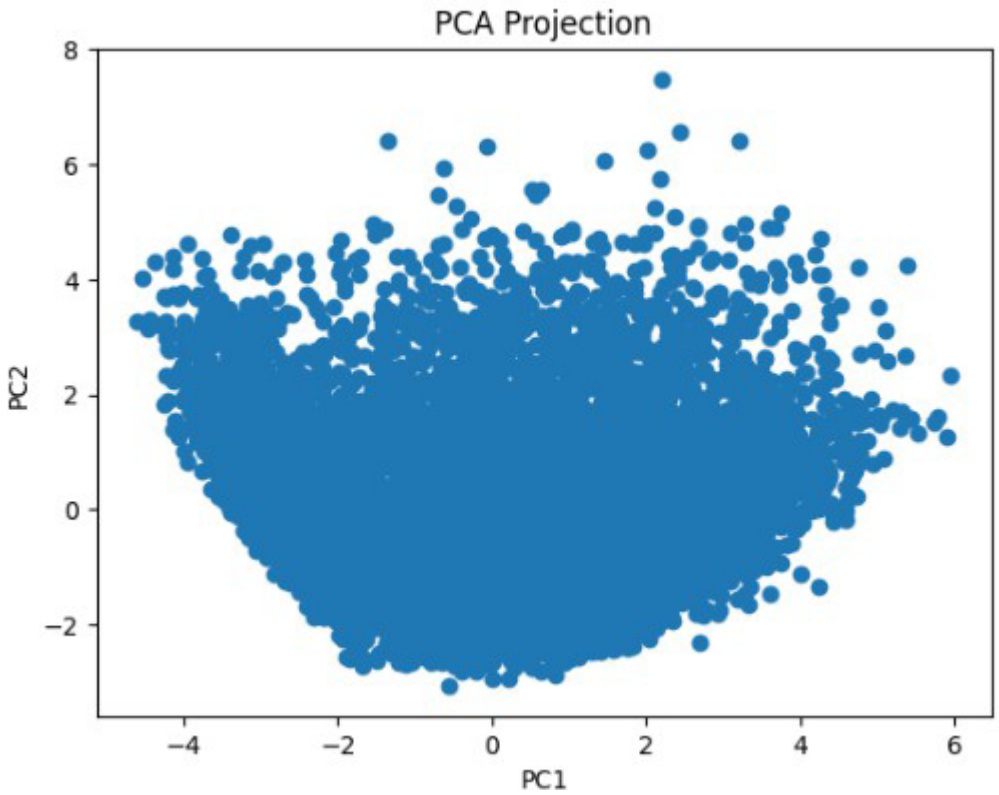
• Purchases and purchase transactions

• Cash advance and cash advance frequency

• Payments and credit limits

These relationships help identify customer financial behaviour patterns.

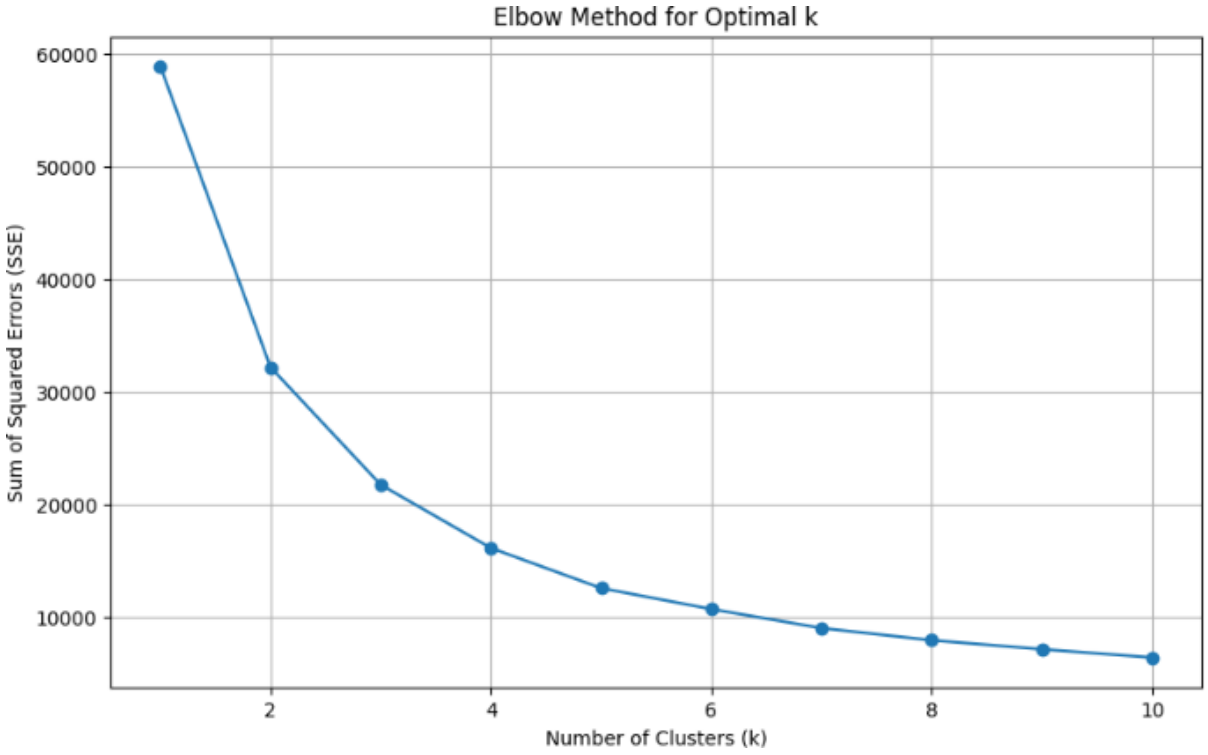
## v. PCA-Based Customer Visualizat




Insight:

PCA successfully reduced high-dimensional customer data into fewer components while preserving important behavioural patterns. This improved clustering efficiency and visualization clarity.

## vi. Optimal Cluster Identification (Elbow Method)




Insight:

The Elbow Method identified the optimal number of customer segments for K-Means clustering, ensuring balanced and meaningful grouping.

## vii. Customer Segmentation using K-Means Clustering

Insight:

The clustering model grouped customers into distinct segments such as:

• High spending premium customers

• Low activity customers

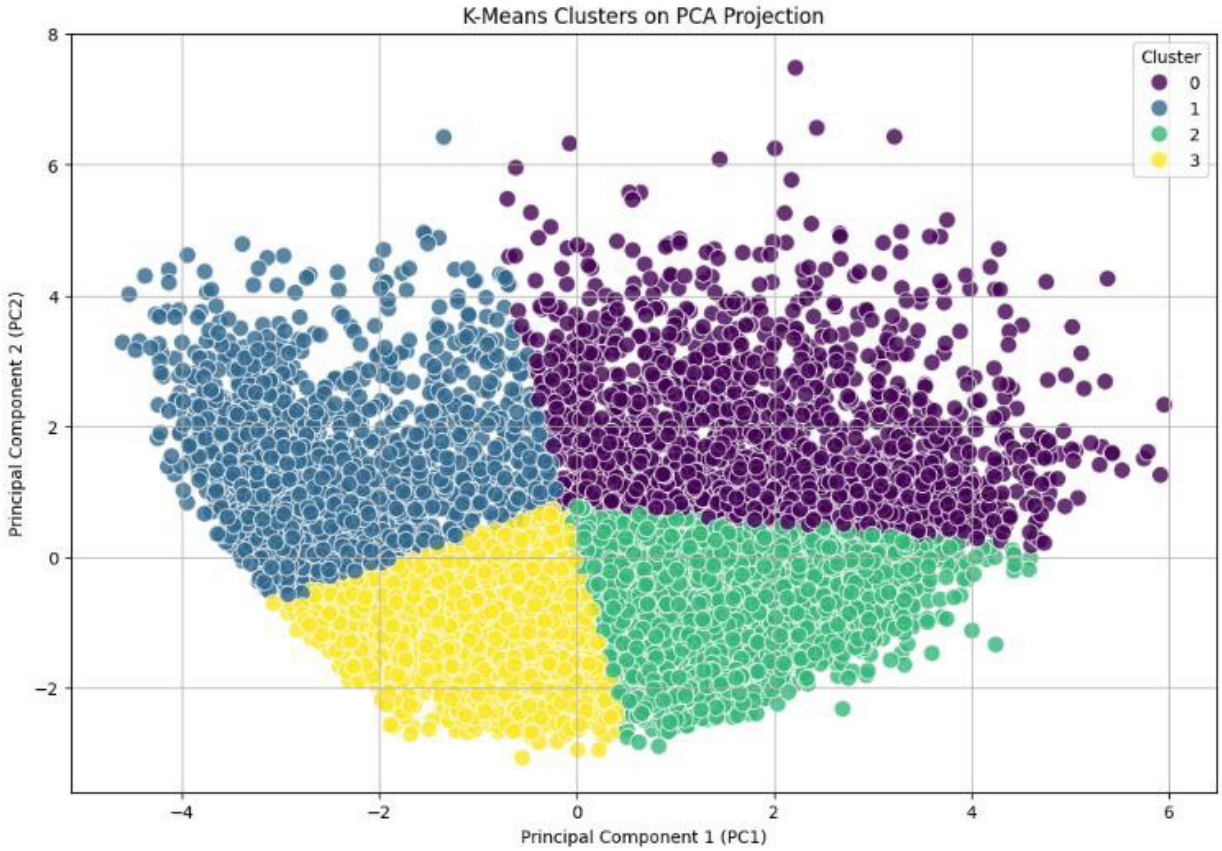
• Installment-based customers

• Cash advance dependent users

• Moderate balanced users

These clusters help businesses create targeted financial strategies.

### viii. Cluster Visualization (PCA Scatter Plot)




Insight:

The scatter plot clearly shows separation between customer groups, confirming effective clustering performance and behavioural differentiation among customers.

## 2.3.5 Conclusion

This project applied Python and Machine Learning techniques to segment credit card users based on their financial behaviour.

The analysis reveals:

• Customers exhibit diverse spending and payment patterns.

• K-Means clustering effectively groups customers into meaningful behavioural segments.

• PCA improves clustering efficiency by reducing feature dimensionality.

• High spenders, installment users, and low-usage customers can be clearly identified.

• Customer segmentation supports personalized marketing and business decision-making.

The machine learning workflow ensured effective preprocessing, clustering, and insightful customer behaviour analysis.

### 2.3.6 Key Findings

ϖ K-Means clustering successfully identified distinct customer groups.

ϖ High-spending customers represent premium business opportunities.

ϖ Some customers rely heavily on cash advances and revolving balances.

ϖ PCA improved visualization and clustering performance.

ϖ Strong relationships exist between purchases, payments, and credit limits.

ϖ Low-usage customers may require engagement and retention strategies.

ϖ Customer segmentation helps improve marketing, risk analysis, and customer targeting.

## 2.4 Customer segmentation(E-commerce) (Week-4)

### 2.4.1 Introduction

E-commerce companies manage a large number of customers with different purchasing habits, shopping frequencies, and spending patterns. Analysing customer behaviour manually becomes difficult due to the huge volume of transactional data generated every day.

This project, **Customer Segmentation Using RFM Analysis and K-Means Clustering**, focuses on grouping customers into meaningful segments using Machine Learning techniques. The project applies:RFM (Recency, Frequency, Monetary) Analysis ,Data Preprocessing ,Feature Scaling ,PCA (Principal Component Analysis) ,K-Means Clustering ,Hierarchical Clustering

to identify different types of customer behaviours such as:

- High-value customers
- Regular customers
- At-risk customers
- Low-value customers

The analysis is carried out using Python and Machine Learning libraries to generate valuable customer insights that support:

- Personalized marketing

- Customer retention strategies
- Sales improvement
- Business decision-making

### 2.4.2 Objectives

The major objectives of this project are:

- To clean and preprocess e-commerce customer transaction data.
- To analyse customer purchasing behaviour using RFM analysis.
- To create Recency, Frequency, and Monetary features.
- To apply feature scaling for improving clustering performance.
- To reduce dimensionality using PCA.
- To implement K-Means Clustering for customer segmentation.
- To compare clustering using Hierarchical Clustering.
- To identify meaningful customer groups based on purchasing behaviour.
- To generate visual insights for business and marketing strategies.
- To support customer targeting and customer retention using data-driven approaches.

### 2.4.3 Methodology

The methodology follows a structured Machine Learning workflow including preprocessing, feature engineering, dimensionality reduction, clustering, and visualization.

❖ **Python Phase – Data Cleaning, Preprocessing, and EDA**

In the Python environment (Google Colab/Jupyter Notebook), the following steps were completed:

**Data Loading & Inspection**

- Imported the Online Retail e-commerce dataset
- Checked datatypes, dataset structure, null values, and duplicates
- Analysed dataset dimensions and feature distributions
- Explored customer transaction information

## Feature Engineering & RFM Analysis

- Created TotalPrice feature using:
  - Quantity × UnitPrice
- Converted InvoiceDate into datetime format
- Generated RFM features:
  - Recency
  - Frequency
  - Monetary

## RFM Analysis

### Recency

Measures how recently a customer made a purchase.

### Frequency

Measures how often a customer purchases.

### Monetary

Measures the total amount spent by customers.

These features help understand customer behaviour and purchasing patterns.

## Outlier Detection & Treatment

- Used boxplots to identify outliers
- Applied IQR (Interquartile Range) method
- Replaced extreme outliers using median values
- Improved clustering stability and performance

## Feature Scaling

- Applied log transformation to reduce skewness
- Used StandardScaler for feature normalization
- Ensured equal contribution of all variables during clustering

## Dimensionality Reduction using PCA

- Applied Principal Component Analysis (PCA)

- Reduced high-dimensional RFM data into principal components
- Preserved most of the dataset variance
- Improved cluster visualization and clustering performance

### K-Means Clustering Implementation

- Used Elbow Method to identify optimal clusters
- Applied Silhouette Score analysis
- Implemented K-Means clustering algorithm
- Assigned cluster labels to customers

### Hierarchical Clustering

- Applied Agglomerative Hierarchical Clustering
- Compared clustering behaviour with K-Means clustering
- Analysed customer grouping similarities

### Python Visualizations

- Boxplots for outlier detection
- Correlation heatmaps
- Elbow Method visualization
- Silhouette Score plots
- PCA scatter plots for cluster visualization
- t-SNE visualization plots
- Cluster comparison bar charts
- Customer distribution plots

## 2.4.4 Results & Insights

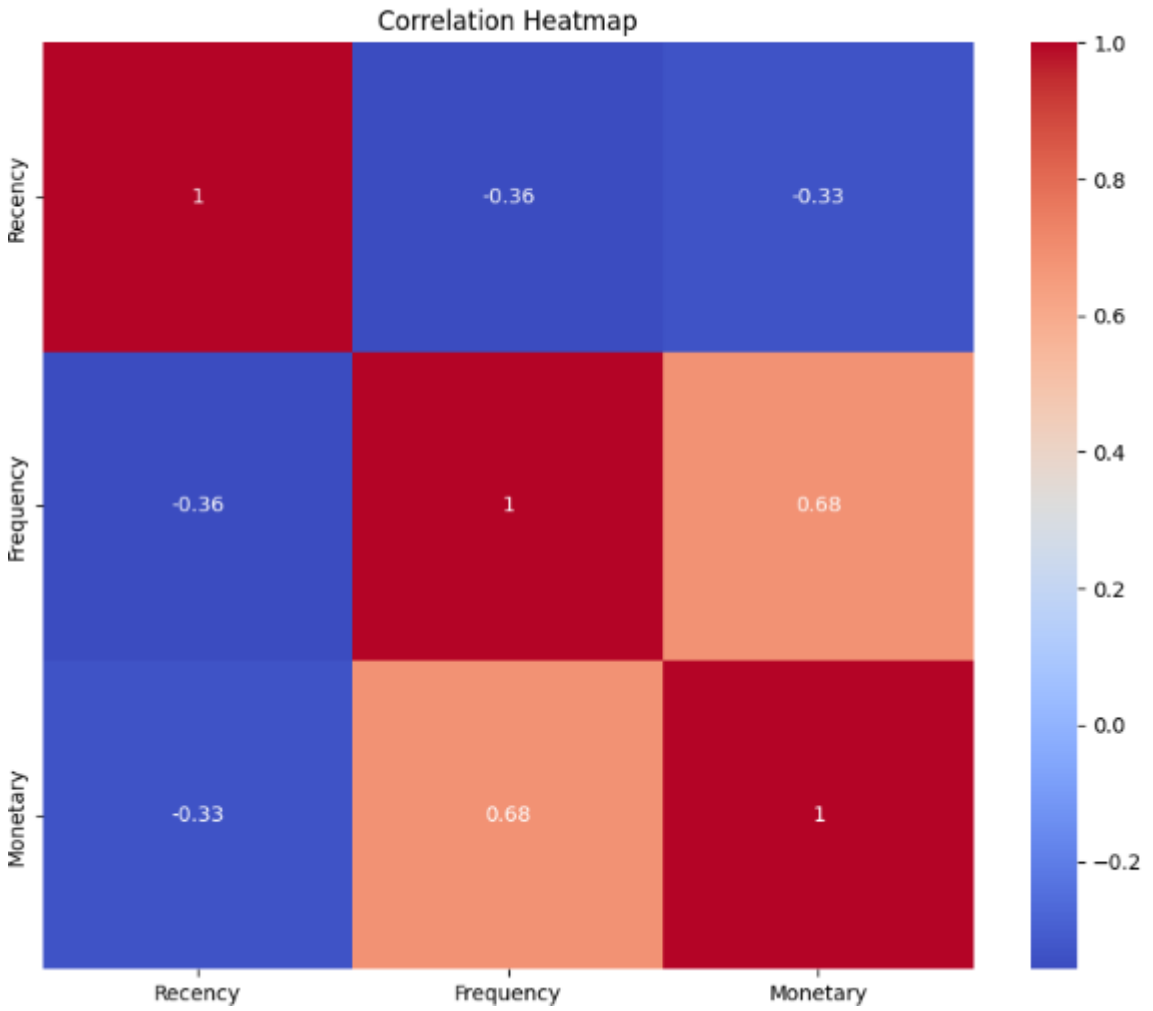
Below is a summary of the key analyses and insights generated from the Machine Learning workflow:

***i. Customer Spending Distribution Analysis***

**Insight:**

Customer purchase behaviour varies significantly across the dataset. Some customers make

high-value purchases frequently, while others use credit cards minimally.




**Insight:**

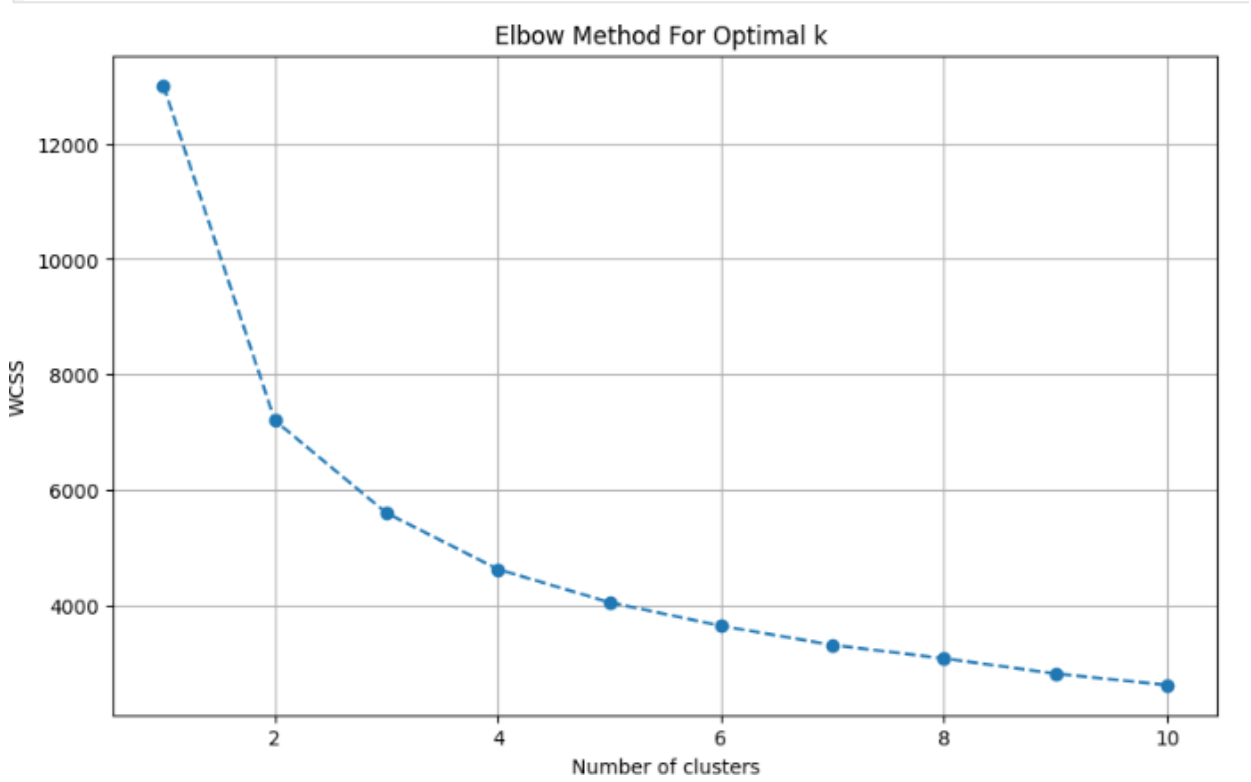
Several financial variables show strong positive and negative relationships. Features like PURCHASES, CREDIT_LIMIT, and PAYMENTS are highly related and influence customer segmentation.

### *ii. Feature Scaling & PCA Transformation*

**Insight:**

Feature scaling improved clustering accuracy by normalizing different financial variables. PCA successfully reduced dimensionality while preserving major data variance.

### *iii. Elbow Method for Optimal Clusters*




**Insight:**

The Elbow Method helped identify the ideal number of clusters required for effective customer segmentation. This improved the efficiency of the K-Means model.

### *iv. Customer Segmentation using K-Means Clustering*

**Insight:**

The clustering model grouped customers into multiple behavioral segments:

• High spending premium customers

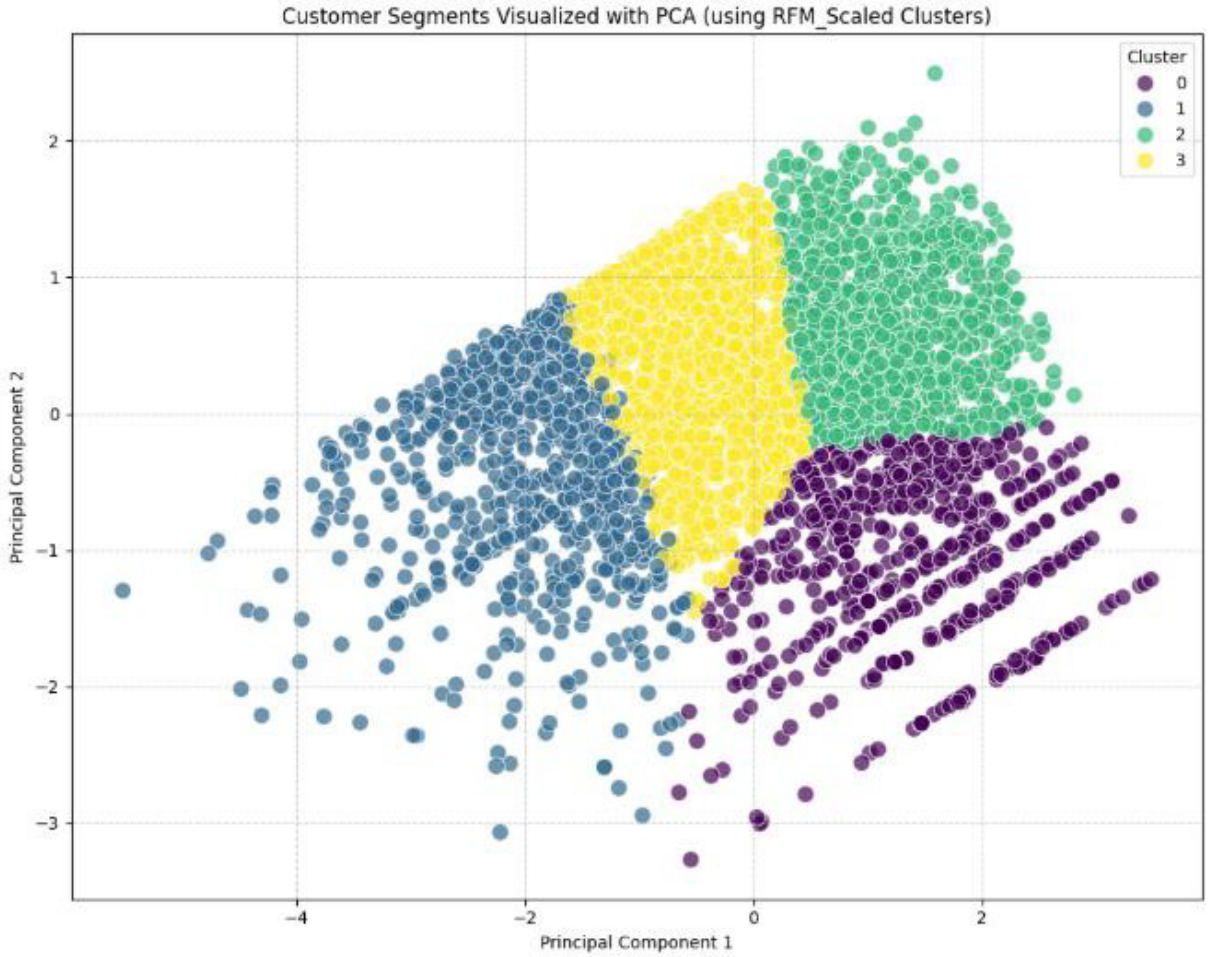
• Customers with frequent installment purchases

• Low engagement customers

• Customers dependent on cash advances

These groups help businesses design targeted financial services and marketing strategies.

### *v. PCA Cluster Visualization (Scatter Plot)*




**Insight:**

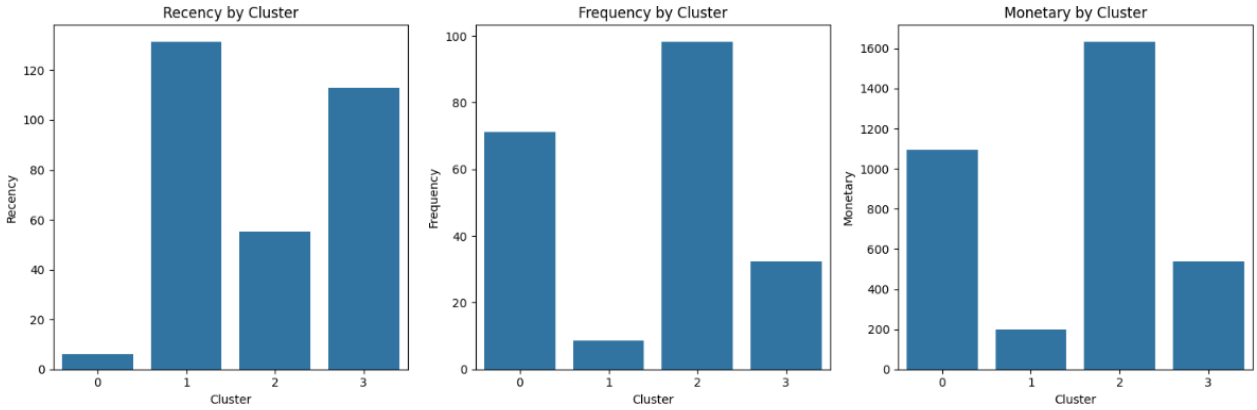
The PCA scatter plot clearly separates customer groups based on financial behaviour. Distinct clusters indicate meaningful segmentation patterns within the customer base.

### *vi. Credit Usage Behaviour Analysis*

**Insight:**

Some customers maintain high balances with low payments, indicating potential financial risk, while others consistently make full payments and maintain healthy credit usage.

### *vii. Payment & Purchase Pattern Analysis*




**Insight:**

Customers with higher purchase frequency often show higher payment consistency. Installment purchase behaviour differs significantly across customer segments.

## 2.4.4 Conclusion

This project applied Python and Machine Learning techniques to segment e-commerce customers based on their purchasing behaviour.

The analysis reveals:

- K-Means clustering effectively groups customers into meaningful behavioural segments.
- Feature scaling and PCA improve clustering quality and visualization.
- Customers show diverse purchasing and spending behaviours.
- High-value and low-engagement customers can be clearly identified.
- Customer segmentation helps businesses improve marketing and customer retention strategies.

The Machine Learning workflow ensured:

- Accurate preprocessing
- Effective RFM analysis
- Efficient clustering
- Meaningful business insights

Overall, the project demonstrates how Machine Learning and customer segmentation can improve business intelligence and customer relationship management in the e-commerce domain.

## 2.4.5 Key Findings

❖ K-Means clustering successfully segmented customers into behavioural groups.

❖ RFM analysis effectively captured customer purchasing behaviour.

❖ Feature scaling significantly improved clustering performance.

❖ PCA reduced dimensionality while preserving important information.

❖ High-value customers form a distinct premium segment.

❖ Some customers show low engagement and irregular purchasing behaviour.

❖ Strong relationships exist between purchase frequency and spending amount.

❖ Customer segmentation supports personalized marketing and retention strategies.

**Business Actions:**

- Reward high-value customers with loyalty programs and exclusive offers
- Re-engage at-risk customers using discounts and promotional campaigns
- Encourage regular customers through personalized recommendations
- Increase engagement of low-value customers through awareness campaigns
- Develop customer-focused marketing strategies based on purchasing behaviour

This project demonstrates the power of Machine Learning, RFM analysis, and customer segmentation in improving business decision-making, marketing strategies, and customer relationship management.

# 2.5 Credit Card Fraud Detection Using Deep Learning

## 2.5.1 Introduction

Credit card fraud detection is one of the most important applications of machine learning in the banking and financial sector. Fraudulent transactions are extremely rare compared to genuine transactions, making fraud detection a highly imbalanced classification problem.

This project focuses on building a Deep Learning model to classify credit card transactions as fraudulent or genuine. The dataset contains anonymized transaction features along with transaction amount and time information.

The analysis is carried out using Python libraries for preprocessing, imbalance handling, visualization, and neural network modelling.

The project includes:

• Data preprocessing and feature scaling

• Class imbalance handling using SMOTE

• Deep Learning model building using Keras

• Model evaluation using Accuracy, ROC-AUC, Precision, Recall, and F1-Score

• Visualization of confusion matrix, accuracy, and loss curves

The developed model helps financial institutions identify fraudulent transactions effectively and reduce financial losses.

## 2.5.2 Objectives

The major objectives of this project are:

• To preprocess and clean credit card transaction data.

• To handle severe class imbalance using SMOTE.

• To scale transaction features for better neural network performance.

• To build a Deep Learning binary classification model using Keras.

• To evaluate fraud detection performance using multiple evaluation metrics.

• To visualize model learning behaviour using accuracy and loss graphs.

• To improve fraud detection capability while minimizing false predictions.

## 2.5.3 Methodology

The methodology follows a structured Deep Learning workflow for fraud detection.

### a) Data Understanding & Exploration

In the Python environment (Google Colab), the following steps were completed:

***Import Libraries***

• Imported NumPy and Pandas for data handling

• Used Matplotlib and Seaborn for visualization

• Used Scikit-learn for preprocessing and evaluation

• Used TensorFlow/Keras for Deep Learning modelling

• Imported SMOTE from imbalanced-learn for handling imbalance

***Dataset Loading***

• Loaded the dataset creditcard.csv

• Displayed first 5 rows of the dataset

• Checked dataset dimensions and structure

• Verified datatype information and statistical summary

***Dataset Information***

• Total transaction records: 284,807

• Total features: 31

• Target variable: Class

0 → Genuine Transaction

1 → Fraudulent Transaction

***Exploratory Data Analysis***

• Checked class distribution using countplot

• Analysed imbalance between fraud and non-fraud transactions

• Observed that fraudulent transactions are extremely rare

## b) Data Preprocessing & Imbalance Handling

***Missing Value & Duplicate Handling***

• Checked missing values in all columns

• No missing values were found

• Removed duplicate records using drop_duplicates()

***Feature Scaling***

• Scaled Time and Amount features using StandardScaler

• V1 to V28 features were already transformed using PCA

***Train-Test Splitting***

• Separated features and target variable

• Split dataset into training and testing sets

• Used stratification to preserve fraud distribution

***SMOTE Oversampling***

• Applied SMOTE only on training data

• Balanced fraud and non-fraud transaction classes

• Improved minority class representation for better learning

Class Distribution After SMOTE:

• Non-Fraud Transactions: 226,602

• Fraud Transactions: 226,602

### c) Deep Learning Model Building

***Neural Network Architecture***

The Deep Learning model was built using Keras Sequential API.

Model Structure:

• Input Layer

• Dense Layer (16 neurons, ReLU activation)

• Dropout Layer (0.5)

• Dense Layer (8 neurons, ReLU activation)

• Dropout Layer (0.5)

• Output Layer (Sigmoid activation)

***Model Compilation***

• Optimizer: Adam

• Loss Function: Binary Crossentropy

• Evaluation Metric: Accuracy

***Model Training***

• Trained model using batch training

• Validation split used for monitoring performance

• Applied EarlyStopping to reduce overfitting

• Stored training history for visualization

## 2.5.4 Results & Insights

Below is a summary of the major analysis and model evaluation results:

### i. Dataset Insights

Insight:

• The dataset contains 284,807 transaction records with 31 features.

• Fraud transactions are extremely rare compared to genuine transactions.

• The dataset is highly imbalanced, making fraud detection challenging.

## ii. Class Distribution Analysis (Count Plot)


Insight:

• Legitimate transactions dominate the dataset.

• Fraudulent transactions account for only a very small percentage of total records.

• Class imbalance required oversampling techniques like SMOTE.

## iii. Feature Scaling & Preprocessing

Insight:

• Time and Amount features were standardized successfully.

• Scaling improved neural network training stability.

• Duplicate records were removed to improve data quality.

## iv. SMOTE Oversampling Results

Insight:

• SMOTE balanced both classes equally in the training dataset.

• The model became less biased toward non-fraud transactions.

• Fraud detection capability improved significantly after balancing.

## v. Deep Learning Model Performance

Insight:

• The neural network successfully learned fraud transaction patterns.

• Dropout layers reduced overfitting and improved generalization.

• The model achieved strong fraud classification performance.

## vi. ROC-AUC Score Analysis

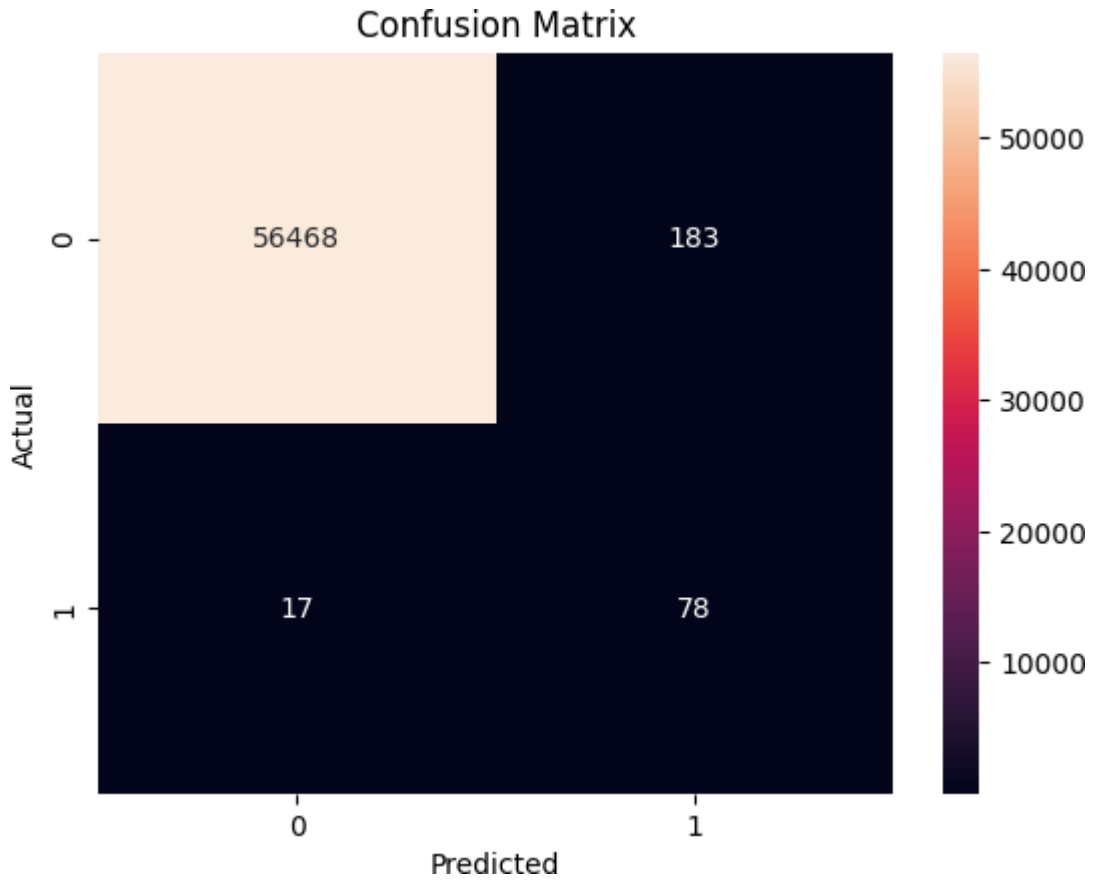
Insight:

• ROC-AUC Score achieved: 0.9534

• The model effectively distinguishes fraudulent and genuine transactions.

• A high ROC-AUC value indicates excellent classification capability.

## vii. Confusion Matrix Analysis




Insight:

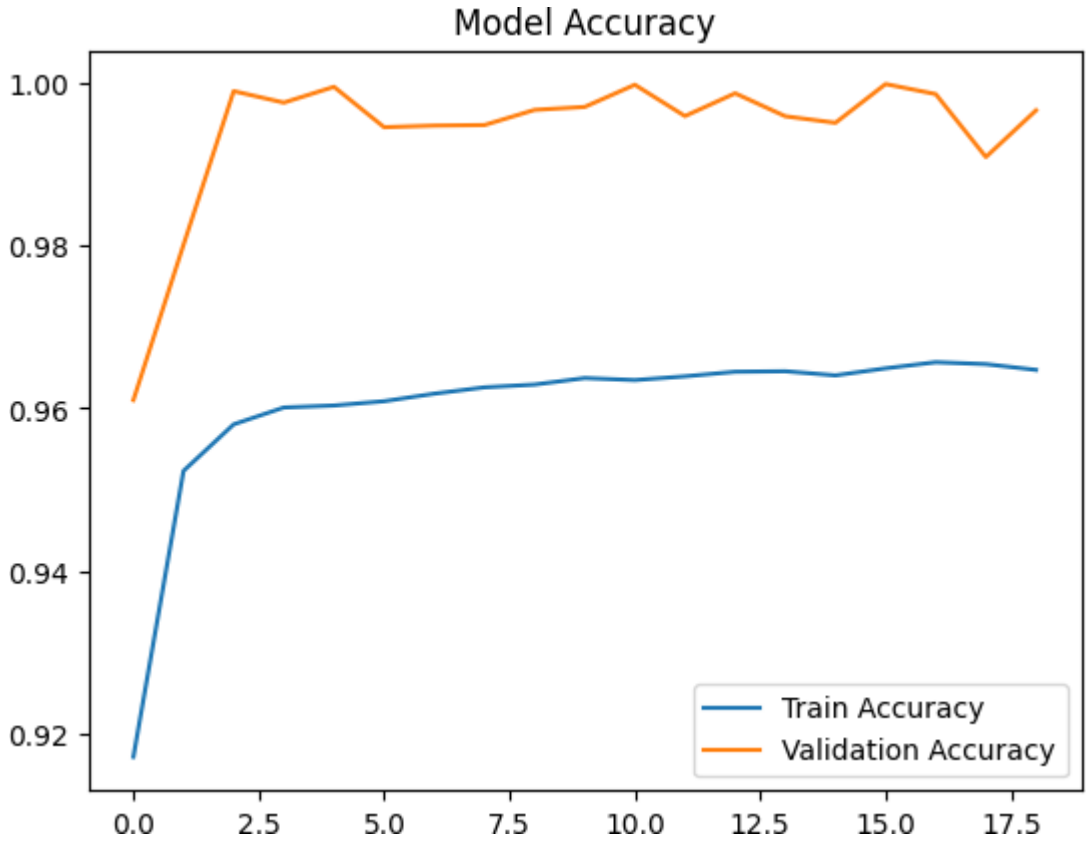
• The model correctly identified 56,468 genuine transactions.

• 78 fraud transactions were correctly detected.

• Only a small number of fraud cases were missed.

• The model showed strong fraud detection capability with minimal errors.

### viii. Accuracy Graph Analysis




Insight:

• Training accuracy steadily increased during learning.

• Validation accuracy remained close to training accuracy.

• The model generalized well without severe overfitting.

### ix. Loss Graph Analysis


Insight:

• Training loss continuously decreased over epochs.

• Validation loss also decreased consistently.

• EarlyStopping prevented unnecessary training and reduced overfitting.

## 2.5.5 Conclusion

This project successfully applies Deep Learning techniques for credit card fraud detection.

The analysis reveals:

• Credit card fraud datasets are highly imbalanced.

• SMOTE effectively balances fraud and non-fraud transactions.

• Feature scaling improves neural network performance.

• Dropout layers help prevent overfitting.

• The Deep Learning model achieves strong fraud detection capability.

• ROC-AUC score confirms excellent classification performance.

The combination of preprocessing, SMOTE, and Deep Learning created a robust fraud detection system capable of identifying fraudulent transactions effectively.

## 2.5.6 Key Findings

ϖ Fraud transactions represent only a very small fraction of total transactions.

ϖ SMOTE significantly improves fraud detection performance.

ϖ Feature scaling improves model training efficiency.

ϖ Dropout layers reduce overfitting and improve generalization.

ϖ The model achieved a high ROC-AUC score of 0.9534.

ϖ The neural network successfully detected most fraud transactions.

ϖ Validation accuracy remained stable throughout training.

ϖ EarlyStopping improved training efficiency and prevented overfitting.

# 2.5 Credit Card Fraud Detection Using CTGAN and XGBoost (Week6)

## 2.6.1 Introduction

Credit card fraud is one of the major financial security challenges faced by banking and digital payment systems. Fraudulent transactions are extremely rare compared to genuine transactions, making fraud detection a highly imbalanced classification problem. Traditional machine learning algorithms often struggle to correctly identify fraud cases due to the limited number of fraud samples.

This project focuses on building an intelligent Credit Card Fraud Detection System using CTGAN (Conditional Tabular GAN) and XGBoost. CTGAN is used to generate synthetic fraudulent transactions to balance the dataset, while XGBoost is used for efficient fraud classification.

The project also includes:

- Feature selection using Random Forest
- Data visualization using PCA
- Model evaluation using ROC-AUC and Confusion Matrix
- Deployment-ready model saving using Joblib

The final system reduces deployment complexity by using only the top 5 most important features while maintaining high fraud detection accuracy.

# 2.6.2 Dataset Description

The project uses the Credit Card Fraud Detection dataset containing anonymized credit card transaction records.

### Dataset Features

The dataset contains:

- 31 columns
- 284,807 transaction records

### Key Variables

- `Time` – Time elapsed between transactions
- `V1 - V28` – PCA-transformed anonymized features
- `Amount` – Transaction amount

- `Class`
  - 0 → Genuine Transaction
  - 1 → Fraudulent Transaction

### Dataset Characteristics

- Fraud cases are extremely rare
- Highly imbalanced dataset
- Majority class: Genuine transactions
- Minority class: Fraud transactions

### Initial Class Distribution

- Genuine Transactions: 284,315
- Fraud Transactions: 492

This imbalance makes fraud detection challenging and motivates the use of synthetic data generation techniques.

## 2.6.3 Data Preprocessing

Data preprocessing was performed to prepare the dataset for machine learning and synthetic data generation.

### Data Cleaning Steps

- Checked missing values
- Removed duplicate rows
- Standardized numerical features
- Prepared dataset for model training

### Duplicate Removal

Duplicate transaction rows were removed:

- Original records: 284,807
- After removing duplicates: 283,726

### Feature Scaling

The following features were standardized using `StandardScaler()`:

- Time
- Amount

This normalization improves model learning efficiency and stability.

### Class Distribution Visualization

A countplot was used to visualize fraud vs non-fraud transactions.

### Insight

- Fraud transactions form only a tiny percentage of total transactions.
- Severe imbalance can bias models toward predicting non-fraud transactions.
- Synthetic data generation becomes necessary to improve fraud learning.

## 2.6.4 Feature Selection

To reduce model complexity and improve deployment efficiency, feature importance analysis was performed using a Random Forest Classifier.

### Top Selected Features

The following top 5 features were selected:

- V17
- V14
- V12
- V10
- V16

### Insight

These variables contribute most significantly toward identifying fraudulent transaction behaviour. Using only important features:

- Reduces training time
- Simplifies deployment

- Improves model interpretability
- Allows lightweight fraud prediction systems

## 2.6.5 Synthetic Fraud Data Generation Using CTGAN

### CTGAN Overview

CTGAN (Conditional Tabular Generative Adversarial Network) is used to generate realistic synthetic fraud transactions.
Instead of simple oversampling, CTGAN learns the statistical distribution of fraud data and creates new artificial fraud samples with similar characteristics.

### Steps Performed

1. Extracted fraud transactions only
2. Created metadata using `SingleTableMetadata`
3. Trained CTGAN synthesizer
4. Generated 3000 synthetic fraud samples
5. Combined synthetic and original datasets

### Insight

The generated synthetic fraud samples help:

- Balance the dataset
- Improve model learning
- Reduce bias toward majority class
- Increase fraud detection capability

### Balanced Dataset Visualization

A countplot after CTGAN augmentation showed a much more balanced class distribution.

### Insight

- Fraud representation improved significantly
- XGBoost can now learn fraud patterns more effectively
- Reduces risk of ignoring minority class transactions

## 2.6.6 Machine Learning Methodology

### Algorithm Used

The project uses:

- XGBoost Classifier

### Why XGBoost?

XGBoost was selected because:

- High performance on tabular data
- Handles non-linear relationships effectively
- Robust against overfitting
- Excellent classification capability
- Efficient handling of large datasets

### Train-Test Split

The dataset was divided into:

- 80% Training Data
- 20% Testing Data

Stratified sampling was used to preserve class distribution.

### Model Training

The XGBoost model was trained using:

- Balanced dataset
- Selected top features
- Log-loss evaluation metric

## 2.6.7 Model Evaluation & Visualization

### Classification Report

**Metric Fraud Class Performance**

Precision 0.97

| Metric | Fraud Class Performance |
| --- | --- |
| Recall | 0.93 |
| F1-Score | 0.95 |

## Insight

- High precision indicates fewer false fraud alerts.
- High recall means most fraud transactions are correctly identified.
- Strong F1-score confirms balanced model performance.

## ROC-AUC Score

The model achieved:

- ROC-AUC Score = 0.9925

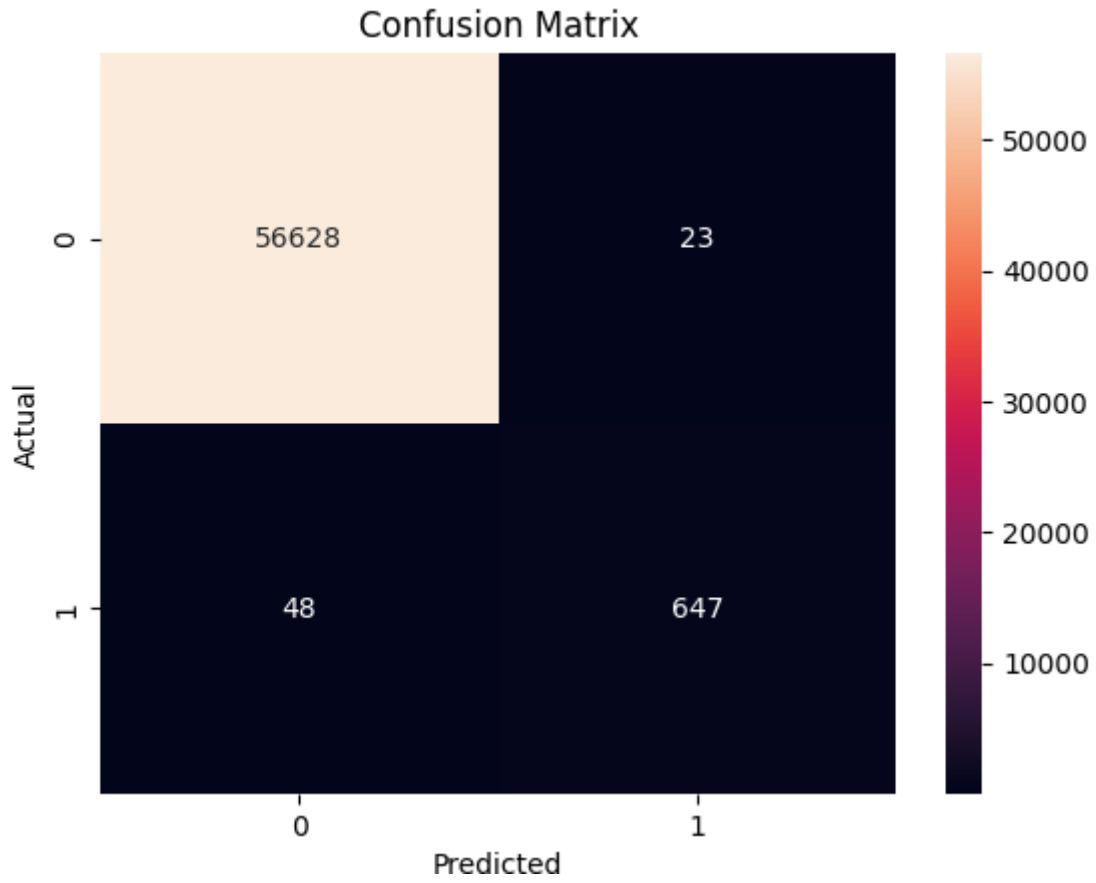
## Insight

- Excellent fraud classification capability
- Strong separation between fraud and genuine transactions
- Near-perfect detection performance

## Confusion Matrix Analysis

The confusion matrix showed:

- High true positive detection
- Very low false negatives
- Minimal false alarms

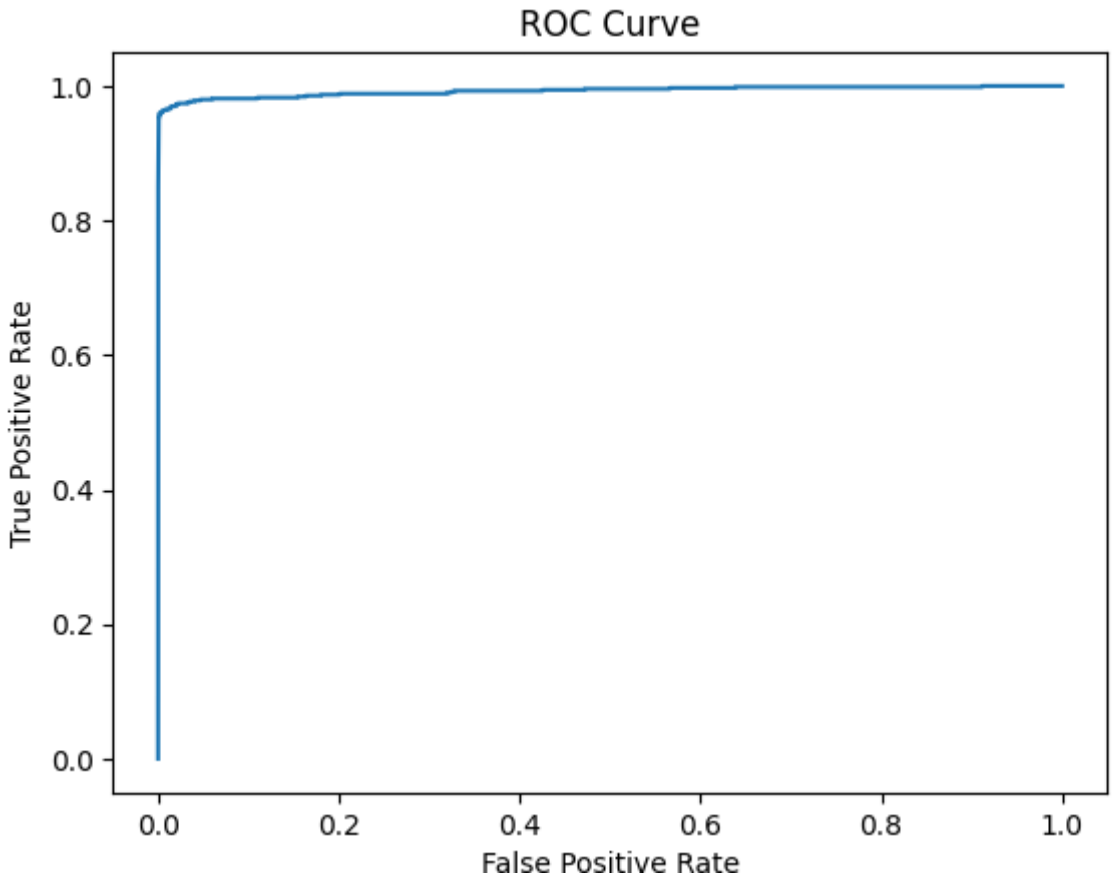



## Insight

- Most fraud cases were successfully identified.
- Few genuine transactions were incorrectly flagged.
- The model performs reliably in real-world fraud screening scenarios.

## ROC Curve

The ROC curve remained close to the top-left corner.



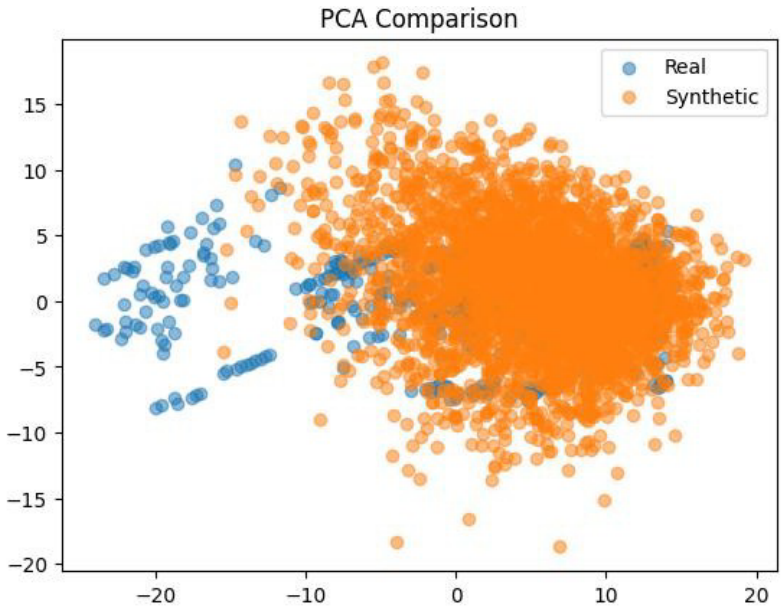

## Insight

- Indicates excellent classifier performance
- High true positive rate
- Low false positive rate

## PCA Visualization

PCA was used to compare:

- Real fraud transactions
- CTGAN-generated synthetic fraud transactions




---

### Insight

- Synthetic fraud samples closely overlap with real fraud patterns.
- CTGAN successfully learned fraud distribution behaviour.
- Synthetic data quality is high and useful for augmentation.

## 2.6.8 Model Deployment Preparation

The trained model and selected features were saved using Joblib.

### Saved Files

- `fraud_model.pkl`
- `features.pkl`

### Purpose

These files can be used for:

- Streamlit deployment
- Real-time fraud prediction
- Lightweight production systems

## 2.6.9 Results and Findings

### Major Findings

- CTGAN effectively solved class imbalance.
- Feature selection reduced input variables from 30 to 5.
- XGBoost achieved excellent fraud detection accuracy.
- ROC-AUC score exceeded 99%.
- Synthetic fraud samples closely matched real fraud behaviour.
- PCA confirmed high-quality data augmentation.

### Expected Outcome

The system provides:

- Faster fraud detection
- Improved fraud prediction accuracy
- Reduced deployment complexity
- Better financial security support

### Challenges

- Extreme class imbalance
- Synthetic data quality validation
- Preventing overfitting
- Selecting the most important features

## 2.6.10 Conclusion

This project successfully developed a high-performance Credit Card Fraud Detection System using CTGAN and XGBoost.

CTGAN effectively generated realistic synthetic fraud transactions to overcome the severe class imbalance problem, while XGBoost provided highly accurate fraud classification performance.

Feature selection significantly reduced model complexity by limiting prediction inputs to only five critical variables without sacrificing accuracy.

The project demonstrates:

- Advanced fraud analytics

- Effective synthetic data generation
- Strong machine learning performance
- Deployment-ready fraud prediction capability

Overall, the system provides a scalable, accurate, and practical solution for real-world fraud detection applications.

## 2.6.11 Key Takeaways

- Fraud datasets are highly imbalanced.
- CTGAN effectively generates realistic fraud samples.
- Feature selection simplifies deployment.
- XGBoost provides strong fraud classification performance.
- ROC-AUC is an important metric for fraud detection.
- PCA helps validate synthetic data quality.
- Synthetic augmentation improves minority class learning.
- Lightweight models are easier to deploy in real-world systems.

# CHAPTER 3: METHODOLOGY

## 3.1 Tools and Technologies Used

During the internship, multiple tools, technologies, and programming libraries were used for data preprocessing, exploratory data analysis (EDA), machine learning, deep learning, visualization, and predictive modelling. These tools helped in analysing structured datasets, building analytical models, and generating meaningful business insights.

### Programming Languages

- Python

### Development Platforms

- Google Colab
- Jupyter Notebook

### Python Libraries Used

**Data Processing & Numerical Computing**

- Pandas
- NumPy

**Data Visualization**

- Matplotlib
- Seaborn
- Plotly

**Machine Learning & Statistical Analysis**

- Scikit-learn
- XGBoost

**Deep Learning**

- TensorFlow
- Keras

**Synthetic Data Generation**

- CTGAN (Conditional Tabular GAN)

**Imbalanced Data Handling**

- SMOTE (Synthetic Minority Oversampling Technique)

### Machine Learning Techniques Applied

- Exploratory Data Analysis (EDA)
- Feature Scaling
- Principal Component Analysis (PCA)
- K-Means Clustering
- Random Forest Feature Selection
- Classification Models
- Deep Learning Neural Networks
- Synthetic Data Augmentation

### Visualization Techniques

- Histograms
- Scatter Plots
- Correlation Heatmaps
- Boxplots
- Countplots
- Bubble Charts
- PCA Scatter Visualizations
- Confusion Matrix Visualization
- ROC Curves

# 3.2 Data Sources and Collection

The datasets used during the internship were collected from publicly available platforms and research repositories. The datasets included financial transaction records, traffic accident data, and customer behaviour datasets.

### Major Data Sources

- Kaggle Datasets
- Public Machine Learning Repositories
- Research Articles and Open Data Platforms

### Datasets Used

| Project | Dataset Description |
|---|---|
| Credit Card Fraud Detection | Credit card transaction dataset containing fraud and non-fraud records |
| Traffic Accident Analysis | Traffic accident dataset containing driver, vehicle, road, and environmental details |
| Customer Segmentation | Credit card customer behaviour and spending dataset |
| Deep Learning Fraud Detection | Financial transaction dataset with imbalanced fraud records |

| Project | Dataset Description |
|---|---|
| CTGAN & XGBoost Fraud Detection | Credit card fraud dataset used for synthetic data generation and classification |

The collected datasets were imported into Python environments such as Google Colab and Jupyter Notebook for preprocessing, analysis, and model development.

# 3.3 Data Cleaning and Preprocessing

Data preprocessing was one of the most important stages of the internship projects. Raw datasets often contained missing values, duplicate records, inconsistent formats, and imbalanced classes that required cleaning before analysis and modelling.

### Major Preprocessing Steps

## a) Data Inspection

- Checked dataset dimensions and structure
- Verified datatypes of variables
- Identified missing values and duplicates
- Analysed statistical summaries

## b) Data Cleaning

- Removed duplicate rows
- Standardized column names and categorical values
- Corrected inconsistent data formats
- Removed unnecessary features where required

## c) Missing Value Handling

- Checked null values across datasets
- Applied median imputation and suitable preprocessing techniques
- Ensured datasets were complete before model training

## d) Feature Scaling

Feature scaling was applied to normalize numerical variables and improve machine learning model performance.
Techniques Used:

- StandardScaler
- Feature Normalization

## e) Dimensionality Reduction

Principal Component Analysis (PCA) was applied in customer segmentation and fraud detection projects to:

- Reduce high-dimensional data
- Improve clustering efficiency
- Enhance visualization quality

## f) Handling Imbalanced Data

Fraud detection datasets were highly imbalanced because fraudulent transactions were extremely rare compared to legitimate transactions.
Techniques Used:

- SMOTE Oversampling
- CTGAN Synthetic Data Generation

These methods improved minority class representation and enhanced model learning capability.

# 3.4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to understand dataset behaviour, identify trends, and extract meaningful insights.

### EDA Techniques Used

- Statistical summaries
- Correlation analysis
- Distribution analysis
- Feature relationship analysis
- Outlier detection
- Trend analysis

### Key EDA Activities

- Fraud transaction distribution analysis
- Accident severity analysis
- Customer spending behaviour analysis
- Correlation heatmap analysis
- Time-based behavioural analysis

EDA helped identify hidden patterns and important relationships between variables before machine learning model implementation.

# 3.5 Machine Learning and Deep Learning Methodology

Machine Learning and Deep Learning techniques were used to perform clustering, fraud detection, classification, and predictive analysis.

## Machine Learning Algorithms Used

### K-Means Clustering

Used for:

- Customer segmentation
- Behavioural grouping
- Marketing analytics

### Random Forest

Used for:

- Feature importance analysis
- Feature selection

### XGBoost Classifier

Used for:

- Fraud classification
- Predictive modelling
- High-performance transaction analysis

## Deep Learning Methodology

Deep Learning models were developed using TensorFlow and Keras.

### Neural Network Components

- Dense Layers
- Dropout Layers
- Sigmoid Activation Function
- Adam Optimizer
- Binary Crossentropy Loss Function

### Deep Learning Workflow

1. Data preprocessing
2. Feature scaling
3. Train-test splitting
4. Model training
5. Validation
6. Performance evaluation

# 3.6 Model Evaluation Techniques

Multiple evaluation metrics were used to measure machine learning and deep learning model performance.

### Evaluation Metrics Used

- Accuracy
- Precision
- Recall
- F1-Score

- ROC-AUC Score
- Confusion Matrix

### Visualization-Based Evaluation

- Accuracy Curves
- Loss Curves
- ROC Curves
- Confusion Matrix Heatmaps

These evaluation techniques helped measure model reliability, classification capability, and overall prediction performance.

# 3.7 Data Visualization Techniques

Visualization techniques were used extensively to improve understanding of trends, patterns, and model performance.

### Visualization Libraries

- Matplotlib
- Seaborn
- Plotly

### Types of Visualizations Created

- Histograms
- Boxplots
- Scatter Plots
- Countplots
- Heatmaps
- Bubble Charts
- PCA Visualizations
- ROC Curves
- Confusion Matrix Plots

Interactive and statistical visualizations helped convert raw analytical outputs into meaningful business insights.

# CHAPTER 4: RESULTS AND DISCUSSION

## 4.1 Overall Insights from Weekly Projects

The internship projects provided valuable insights into financial analytics, fraud detection, customer behaviour analysis, machine learning, and public safety analytics.
The projects demonstrated how data preprocessing, exploratory data analysis, machine learning, and deep learning techniques can transform raw datasets into meaningful analytical solutions.

### Credit Card Fraud Detection Insights

The fraud detection projects revealed that fraudulent transactions form only a very small percentage of total financial transactions, making fraud detection a highly imbalanced classification problem.
Major observations include:

- Fraud transactions cannot be identified using a single variable alone.
- Feature relationships and transaction patterns play an important role in fraud detection.
- SMOTE and CTGAN significantly improved minority class learning.
- XGBoost and Deep Learning models achieved high fraud detection performance.

### Traffic Accident Analysis Insights

The traffic accident analysis project identified major factors contributing to accident occurrence and severity.
Key findings include:

- Driver behaviour and environmental conditions strongly influence accidents.
- Weather and poor lighting increase accident severity risks.
- Heavy vehicles contribute significantly to accident frequency.
- Visualization techniques helped identify accident-prone conditions and locations.

### Customer Segmentation Insights

The customer segmentation projects successfully grouped customers based on spending habits, payment behaviour, and credit usage patterns.
Key findings include:

- High-spending customers form premium customer segments.
- Some users rely heavily on cash advances and minimum payments.
- PCA improved clustering visualization and efficiency.
- K-Means clustering effectively differentiated customer behaviour patterns.

# 4.2 Machine Learning and Deep Learning Performance Discussion

The internship involved practical implementation of multiple machine learning and deep learning algorithms.

### Machine Learning Performance

- K-Means Clustering successfully grouped customers into meaningful behavioural categories.
- Random Forest effectively identified important fraud-related features.
- XGBoost achieved excellent classification capability for fraud detection.

### Deep Learning Performance

The Deep Learning fraud detection model showed:

- High ROC-AUC performance
- Strong classification capability
- Stable validation accuracy
- Reduced overfitting using dropout layers and EarlyStopping

### Synthetic Data Generation Performance

CTGAN successfully generated realistic fraud transaction samples that closely matched original fraud patterns.
Benefits observed:

- Improved fraud representation
- Better minority class learning
- Enhanced model performance
- Reduced bias toward majority class

# 4.3 Visualization and Analytical Findings

Visualization played a major role in understanding dataset behaviour and communicating insights effectively.

### Important Visualization Outcomes

- Heatmaps identified feature relationships affecting fraud and accident severity.
- Scatter plots revealed customer spending and transaction behaviour patterns.
- PCA visualizations clearly separated customer segments.
- ROC curves demonstrated strong model classification capability.
- Bubble charts improved understanding of accident causes and casualty impact.

Visual analytics significantly improved interpretation and decision-making capability across all projects.

# 4.4 Skills Gained During Internship

The internship helped develop both technical and professional skills.

## Technical Skills Gained

- Data Cleaning and Preprocessing
- Exploratory Data Analysis (EDA)
- Data Visualization
- Machine Learning
- Deep Learning
- Feature Engineering
- Customer Segmentation
- Fraud Detection Analytics
- PCA and Clustering
- Synthetic Data Generation using CTGAN
- XGBoost Modelling
- Model Evaluation Techniques

## Software and Tools Learned

- Python
- Google Colab
- Jupyter Notebook
- Pandas
- NumPy
- Scikit-learn
- TensorFlow
- Keras
- Matplotlib
- Seaborn
- Plotly

## Professional Skills Developed

- Analytical Thinking
- Problem Solving
- Data Interpretation
- Report Preparation
- Visualization and Presentation
- Research Skills
- Business Insight Generation

# 4.5 Challenges Faced During Internship

Several practical challenges were encountered during project implementation.

**Major Challenges**

- Handling highly imbalanced fraud datasets

- Preprocessing large real-world datasets
- Managing missing and inconsistent values
- Reducing overfitting in Deep Learning models
- Selecting optimal clusters in segmentation analysis
- Evaluating synthetic data quality generated by CTGAN

These challenges improved practical understanding of real-world machine learning workflows and analytical problem-solving.

# 4.6 Business and Real-World Applications

The internship projects demonstrated multiple real-world applications of AI, Machine Learning, and Data Analytics.

### Financial Sector Applications

- Fraud transaction monitoring
- Real-time fraud detection systems
- Financial risk analysis

### Business Intelligence Applications

- Customer segmentation
- Personalized marketing
- Customer retention analysis

### Public Safety Applications

- Road accident analysis
- Traffic risk monitoring
- Road safety planning

### AI and Predictive Analytics Applications

- Classification systems
- Behaviour prediction
- Synthetic data generation
- Predictive modelling

The internship successfully demonstrated how analytical and AI-based techniques can support intelligent decision-making across multiple industries.

# CHAPTER 5: CONCLUSION

## 5.1 Overall Learning Outcomes

The internship at Global Next Consulting India Pvt. Ltd. provided practical exposure to the complete workflow of data analytics, machine learning, and deep learning — including data collection, preprocessing, exploratory data analysis (EDA), visualization, model building, and performance evaluation.

Through six structured projects focused mainly on credit card fraud detection, customer segmentation, and traffic accident analysis, I gained hands-on experience with **Python, Machine Learning, Deep Learning, CTGAN, XGBoost, K-Means Clustering, PCA,** and **data visualization libraries** such as **Matplotlib, Seaborn, and Plotly.**

The projects strengthened my understanding of handling real-world datasets, especially highly imbalanced financial datasets, feature engineering, preprocessing techniques, clustering methods, synthetic data generation, and fraud detection systems.

The internship also enhanced my ability to analyse customer behaviour, identify transaction fraud patterns, study accident severity factors, and generate meaningful business insights through statistical analysis and visualization techniques.

The final projects on Credit Card Fraud Detection using Deep Learning and CTGAN with XGBoost integrated advanced machine learning concepts such as SMOTE, neural networks, synthetic data generation, feature selection, and deployment-ready model development.

Overall, the internship improved both technical and professional skills including analytical thinking, problem-solving, data interpretation, teamwork, model evaluation, and effective communication.

## 5.2 Applications of Work

The knowledge, methodologies, and tools learned during this internship can be applied in multiple real-world domains such as:

• **Financial Fraud Detection:** Developing intelligent fraud detection systems for banking and digital payment platforms using Machine Learning and Deep Learning techniques.

• **Customer Segmentation & Business Analytics:** Analysing customer spending behaviour to create targeted marketing strategies, personalized financial services, and customer retention programs.

• **Risk Analysis & Financial Security:** Identifying suspicious transaction patterns and improving financial security systems through predictive analytics and anomaly detection.

• **Traffic Safety & Accident Analysis:** Analysing accident trends, environmental factors, and driver behaviour to support road safety planning and accident prevention strategies.

• **Machine Learning & AI Applications:** Applying clustering, classification, feature engineering, PCA, CTGAN, and XGBoost techniques to solve real-world business problems.

• **Data Visualization & Decision Support:** Creating analytical dashboards, visual reports, heatmaps, correlation analysis, and interactive charts to support data-driven decision-making.

• **Research & Deployment Systems:** Building deployment-ready predictive models for real-time fraud detection and intelligent business applications using Python and AI technologies.

# Internship Certificate

This is to certify that **Ms. Y. Divya Reddy** has successfully completed her **Six-Week AI-ML Internship Program** at **Global Next Consulting India Pvt. Ltd.** from **23-March-2026 to 10-May-2026**.

During the internship period, she worked on various projects related to **Data Analytics, Machine Learning, Deep Learning, Customer Segmentation, Traffic Accident Analysis, and Credit Card Fraud Detection** using tools and technologies such as **Python, Pandas, NumPy, Scikit-learn, TensorFlow, Keras, XGBoost, CTGAN, Matplotlib, Seaborn, and Plotly**.

She demonstrated sincere dedication, analytical thinking, technical skills, and enthusiasm toward learning throughout the internship period. Her performance and conduct were found to be satisfactory.

We wish her success in all future academic and professional endeavors.

**Ms. Anuradha Gupta**
Program Director
Global Next Consulting India Pvt. Ltd.

# SUMMARY

The internship provided in-depth practical exposure to various data analysis, machine learning, deep learning, and visualization techniques, enabling hands-on experience with tools and technologies such as Python, Machine Learning, Deep Learning, SQL, Pandas, NumPy, Matplotlib, Seaborn, Plotly, Keras, XGBoost, CTGAN, and data visualization methods. Each week focused on solving real-world analytical problems through structured workflows including data collection, preprocessing, exploratory data analysis (EDA), visualization, model building, evaluation, and reporting.

Across the six projects, the work covered multiple analytical and machine learning domains:

• **Week 1 (Credit Card Fraud Detection Trends)** – Analysed credit card transaction data using Python and EDA techniques to identify fraud patterns, transaction behaviour, and class imbalance issues. Applied statistical analysis, correlation analysis, histograms, scatter plots, and heatmaps to generate fraud-related insights and support financial security analysis.

• **Week 2 (Traffic Accident Analysis)** – Performed traffic accident analysis using Python by cleaning and preprocessing accident datasets containing driver, vehicle, road, and environmental

information. Analysed accident severity, weather impact, driver behaviour, road conditions, and casualty patterns using statistical analysis, correlation heatmaps, bubble charts, and interactive visualizations to support road safety planning.

• **Week 3 (Segmenting Credit Card Users using Machine Learning)** – Applied Machine Learning techniques such as K-Means Clustering and PCA to segment credit card customers based on spending behaviour, payments, cash advances, and credit usage patterns. Used feature scaling, clustering algorithms, and visualization techniques to identify meaningful customer groups for marketing and business intelligence applications.

• **Week 4 (Credit Card Customer Segmentation using Python & Machine Learning)** – Conducted advanced customer segmentation analysis using Python, PCA, StandardScaler, and K-Means Clustering. Analysed customer financial behaviour, payment consistency, purchase patterns, and credit usage to generate business insights supporting personalized marketing and customer relationship management strategies.

• **Week 5 (Credit Card Fraud Detection using Deep Learning)** – Built a Deep Learning-based fraud detection model using TensorFlow/Keras to classify fraudulent and genuine transactions. Applied SMOTE for handling severe class imbalance, feature scaling for preprocessing, and neural network optimization using dropout layers and EarlyStopping. Evaluated model performance using Accuracy, ROC-AUC, Precision, Recall, F1-Score, confusion matrix, and learning curves.

• **Week 6 (Credit Card Fraud Detection using CTGAN and XGBoost)** – Developed an advanced fraud detection system using CTGAN for synthetic fraud data generation and XGBoost for classification. Performed feature selection using Random Forest, dimensionality reduction using PCA, and fraud classification using balanced datasets. Built a lightweight deployment-ready fraud prediction system with high ROC-AUC performance and strong fraud detection capability.

Through these projects, both technical and analytical competencies were significantly strengthened, including data preprocessing, exploratory data analysis, feature engineering, machine learning, deep learning, clustering, synthetic data generation, fraud analytics, statistical reasoning, predictive modeling, and data visualization.

The internship enhanced the ability to transform raw datasets into meaningful insights and intelligent predictive systems that support real-world business decision-making, financial security, customer analytics, and public safety analysis.

Overall, this internship journey played a major role in bridging theoretical knowledge with practical industry applications, improving confidence, problem-solving ability, and readiness for professional roles in the fields of **Data Analytics, Machine Learning, Artificial Intelligence**, and **Business Intelligence.**

# REFERENCES


1. **Kaggle Datasets** – Credit Card Fraud Detection Dataset, Traffic Accident Analysis Dataset, Credit Card Customer Segmentation Dataset.
2. **Python Documentation** – Python programming concepts and implementation support for data preprocessing, machine learning, and deep learning workflows.
3. **Pandas Documentation** – Data cleaning, preprocessing, dataframe manipulation, and exploratory data analysis techniques.
4. **NumPy Documentation** – Numerical computing, array operations, and mathematical computations for machine learning workflows.
5. **Matplotlib Documentation** – Data visualization techniques including histograms, scatter plots, line graphs, and model evaluation graphs.
6. **Seaborn Documentation** – Statistical visualization, correlation heatmaps, distribution analysis, and confusion matrix visualizations.
7. **Plotly Documentation** – Interactive visualizations, bubble charts, and advanced analytical dashboards.

8. **Scikit-learn Documentation** – Machine Learning algorithms including K-Means Clustering, PCA, StandardScaler, SMOTE preprocessing, Random Forest, and evaluation metrics.
9. **TensorFlow Documentation** – Deep Learning model development and neural network implementation using TensorFlow and Keras.

**Keras Documentation** – Sequential neural network architecture, dropout layers, model training, and deep learning evaluation techniques.

10. **XGBoost Documentation** – XGBoost classifier implementation for advanced fraud detection and predictive modeling.
11. **SDV CTGAN Documentation** – Synthetic fraud data generation using CTGAN for handling imbalanced datasets.
12. **Imbalanced-learn Documentation** – SMOTE oversampling techniques for balancing fraud and non-fraud transaction classes.
13. Jupyter Notebook Documentation – Interactive Python environment used for machine learning development and exploratory data analysis.
14. **Google Colab Documentation** – Cloud-based notebook environment used for Deep Learning and Machine Learning implementation.
15. Research Articles & White Papers –

• "Machine Learning Techniques for Credit Card Fraud Detection"

• "Deep Learning Approaches for Fraud Transaction Classification"

• "Customer Segmentation using K-Means Clustering"

• "Traffic Accident Analysis and Road Safety Prediction using Data Analytics"
• "Synthetic Data Generation using GANs for Imbalanced Classification Problems"