AI/ML Internship

A Project Report submitted to the

GLOBAL NEXT CONSULTING INDIA PVT LTD

(Six – Week Internship Program)

By

Parth Arora

Under the Supervision of

Dr. Anuradha Gupta
(Project Director)

Submitted To :

Global Next Consulting India Pvt. Ltd.

Duration of Internship :

23-March-2026 to 08-May-2026

May 2025

CANDIDATE'S DECLARATION

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

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

Parth Arora

CERTIFICATE

This is to certify that the project report entitled **“AI/ML Internship Report”** has been carried out by **Parth Arora ,** a student seeking to gain practical skills in Artificial Intelligence & Machine Learning. 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.

Parth Arora

ABSTRACT

This report summarises a six-week internship as an AI/ML Intern at Global Next Consulting India Pvt. Ltd. (GNCIPL), Greater Noida. The internship was structured into six projects — five weekly projects and one major project — aimed at developing practical skills in data handling, machine learning, deep learning, and data visualisation.

The weekly projects covered a wide range of real-world domains: Olympic Games EDA (120 years of historical sports data), Diabetes Risk Analysis (medical EDA on the PIMA Indians dataset), Customer Segmentation (K-Means clustering on Mall Customers data), Student Performance Clustering (K-Means, PCA, and t-SNE on UCI student grades), and Iris Flower Classification using an Artificial Neural Network (ANN) built with TensorFlow and Keras.

The Major Project — Fraud Detection Using Generative AI — was the capstone of the internship. It involved training a CTGAN (Conditional Tabular GAN) model on a severely imbalanced credit card fraud dataset, generating 500 synthetic fraud samples to augment training data, and evaluating the impact on a Random Forest classifier. The augmented model achieved significantly higher Recall and F1-Score compared to the baseline. The project was deployed as a live Streamlit web application.

The internship as a whole strengthened technical skills in Python, Machine Learning (Scikit-learn), Deep Learning (TensorFlow/Keras), Generative AI (CTGAN/SDV), NLP, and data visualisation — while also improving analytical thinking and problem-solving capabilities.

INDEX

Summary

References

Chapter 1 — Introduction

1.1 Company Profile

Global Next Consulting India Private Limited (GNCIPL), headquartered in Greater Noida, Uttar Pradesh, is a cybersecurity-focused consulting firm dedicated to helping organisations protect their digital assets, data, and reputation. As threats evolve in today's digital world, GNCIPL offers proactive, customised solutions rather than reactive fixes.

The company serves clients in diverse sectors including finance, healthcare, manufacturing, and technology, providing services such as threat detection, risk assessment, incident response, compliance consulting, and 24/7 monitoring. GNCIPL's core values — integrity, innovation, customer-centricity, excellence, and collaboration — ensure 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: 0120-4001768 | +91-9315504902 | +91-7666141260
Email: hr@gncipl.com

1.2 Objectives of Internship

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

- To gain hands-on experience in data analytics tools and techniques, especially using Python (Google Colab, Jupyter Notebook).
- To work on real-world datasets and deliver meaningful insights, visualisations, and reports.
- To learn data preprocessing, cleaning, transformation, and applying machine learning algorithms.
- To build and evaluate supervised and unsupervised ML models including clustering, classification, and neural networks.
- To explore Generative AI (CTGAN) for synthetic data augmentation in imbalanced datasets.
- To enhance analytical thinking, effective communication, and presentation skills through weekly projects and a major end-to-end project deployment.

Chapter 2 — Projects

2.1 Week 1 — Olympic Games EDA (120 Years of Olympic History)

2.1.1 Introduction

The Olympics represent 120+ years of athletic achievement and international competition. This Week 1 project focused on performing a comprehensive Exploratory Data Analysis (EDA) on the Kaggle dataset covering every Olympic Games from 1896 to 2016. The dataset includes athlete-level records with medal outcomes, sports, events, country, age, height, and weight — enabling rich analysis of performance trends, country dominance, and sport-wise patterns.

2.1.2 Dataset Description

Attribute	Details
Dataset Name	120 Years of Olympic History — Athletes and Results
Source	Kaggle (heesoo37/120-years-of-olympic-history-athletes-and-results)
Files Used	athlete_events.csv + noc_regions.csv
Total Records	271,116 athlete-event entries
Columns	15 (ID, Name, Sex, Age, Height, Weight, Team, NOC, Games, Year, Season, City, Sport, Event, Medal)
Olympic Years Covered	1896 – 2016 (both Summer & Winter)
Unique Athletes	135,571
Unique Countries (NOC)	230
Unique Sports	66
Missing Values	Age (3.5%), Height (22%), Weight (23%), Medal (85% — non-medalists)

2.1.3 Objectives

- Compare country-wise medal performance across all Olympic years.
- Analyse medal counts per year via bar plots (Summer vs Winter breakdown).
- Compute and compare medals per capita for participating nations.
- Identify sport-wise and event-wise dominance by country.
- Explore athlete demographics (age, height, weight) across sports and medal types.

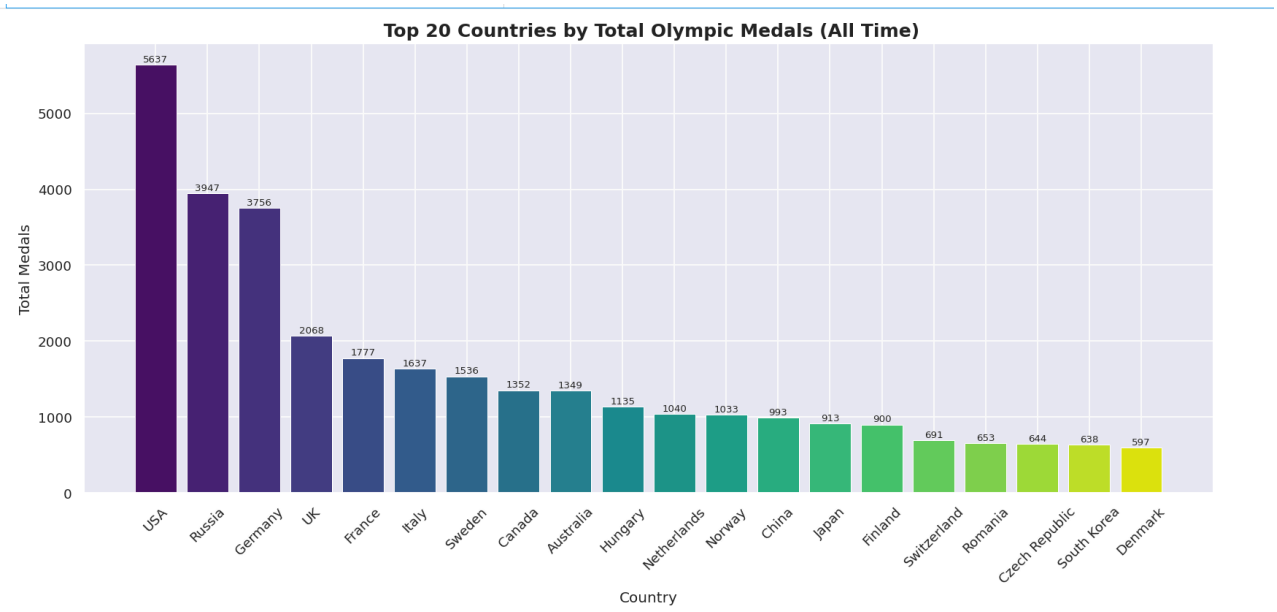
2.1.4 Methodology

The project followed a standard EDA pipeline using Python (Pandas, Matplotlib, Seaborn, Plotly):

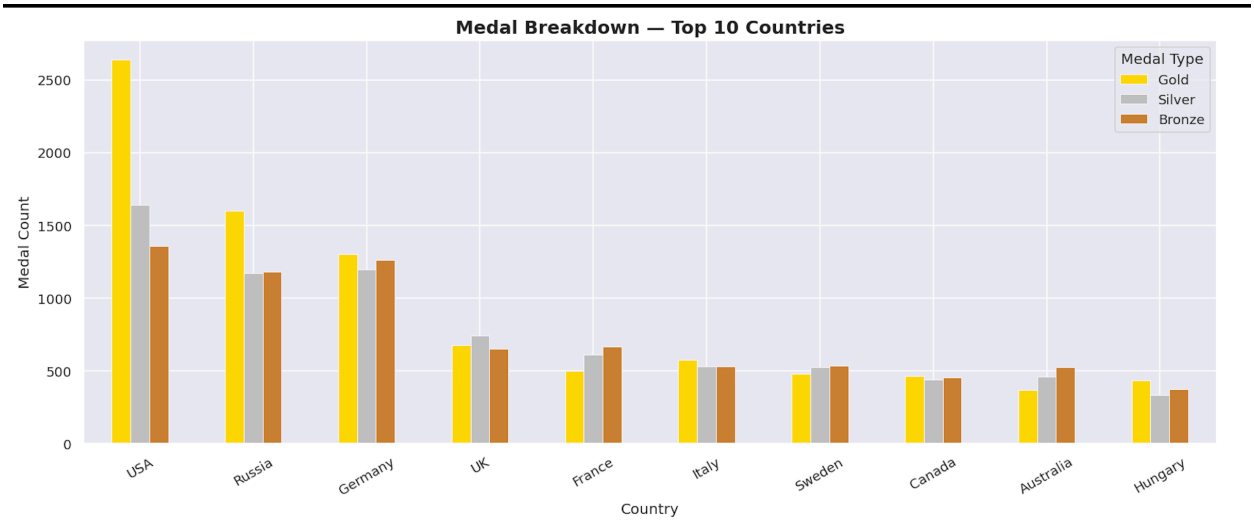
- Data Loading & Merging: Loaded athlete_events.csv and merged with noc_regions.csv to map NOC codes to country names.
- Data Cleaning: Filled Medal column null values with 'No Medal' for analysis; handled missing age/height/weight values.
- Feature Engineering: Created medal-only dataframe, derived decade columns, and computed per-capita medal rates.
- Univariate Analysis: Distributions of medal types, sports frequency, and athlete demographics.
- Country-wise Analysis: Top 20 countries by total medals; Gold/Silver/Bronze breakdown for Top 10.
- Time Trend Analysis: Medals awarded per Olympic year; Summer vs Winter stacked bar charts.
- Visualisation Tools: Matplotlib bar charts, Seaborn heatmaps, Plotly interactive charts.

2.1.5 Charts & Visualisations

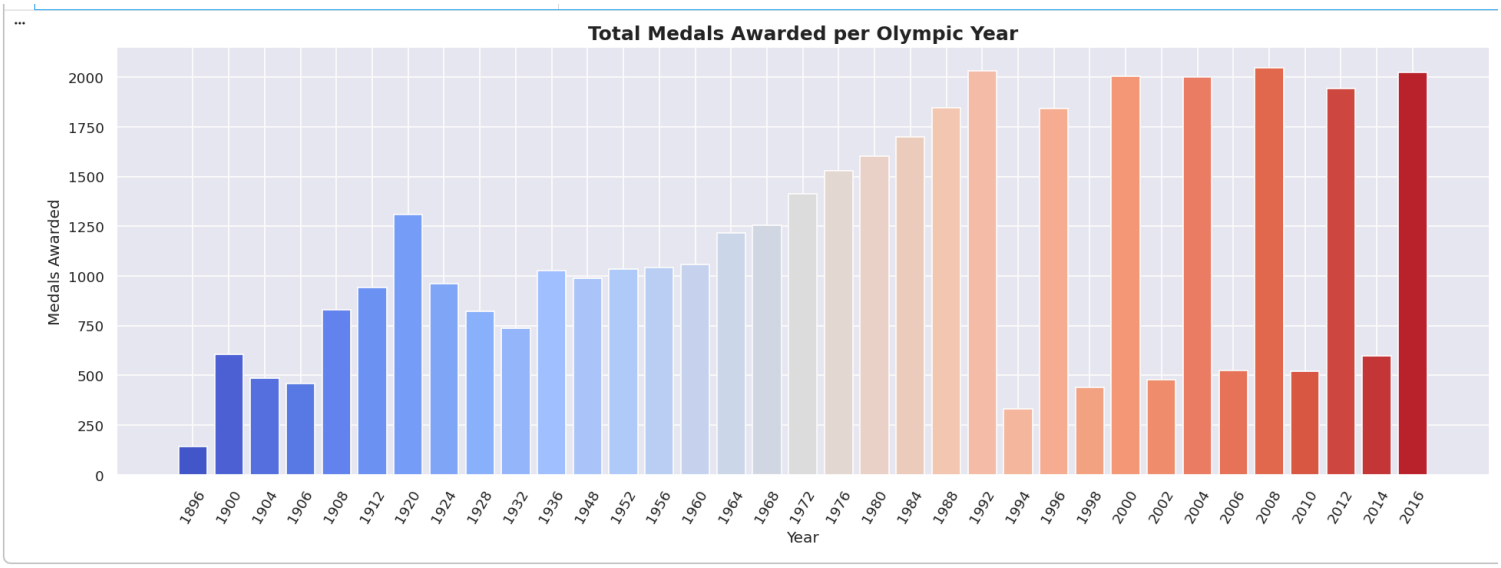
1. Top 20 Countries by Total Olympic Medals



2. Medal Breakdown — Top 10 Countries



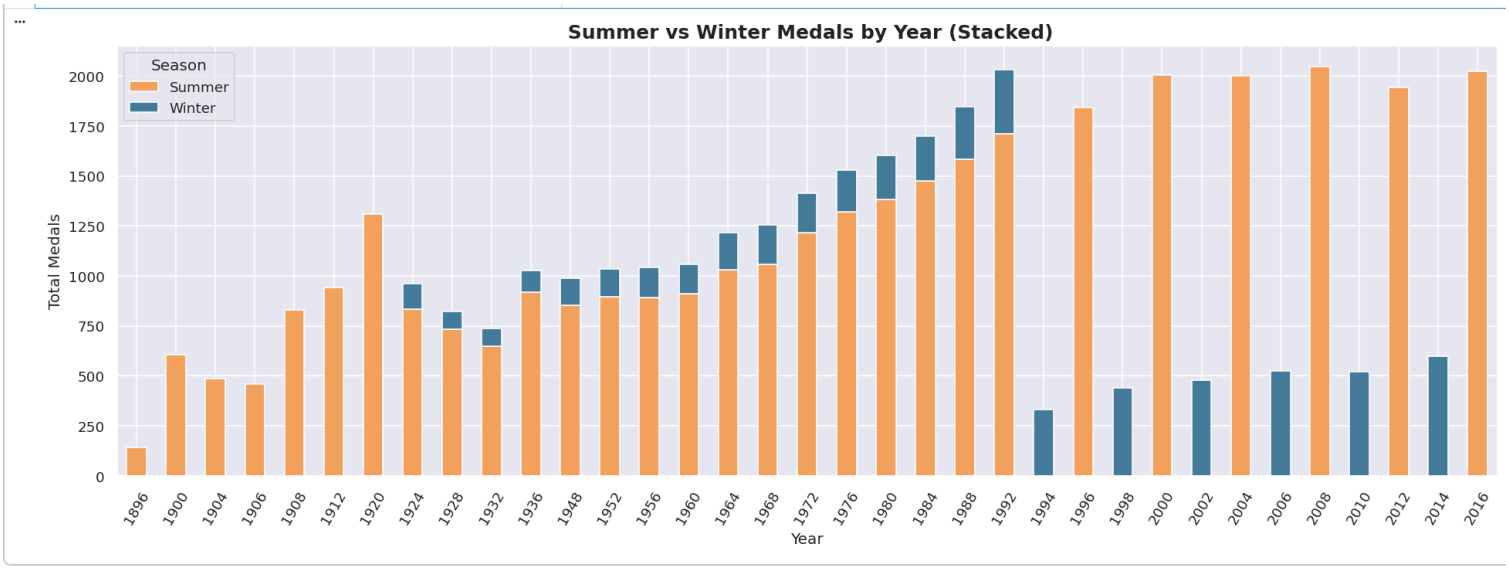
3. Total Medals Awarded per Olympic Year



What are the insights found from the chart?

Ans- The chart shows a general increase in the number of medals awarded over the years, indicating the growth of the Olympic Games in terms of events and participation. Some fluctuations can be observed in certain years, but the overall trend reflects expansion and increasing global involvement in the Olympics.

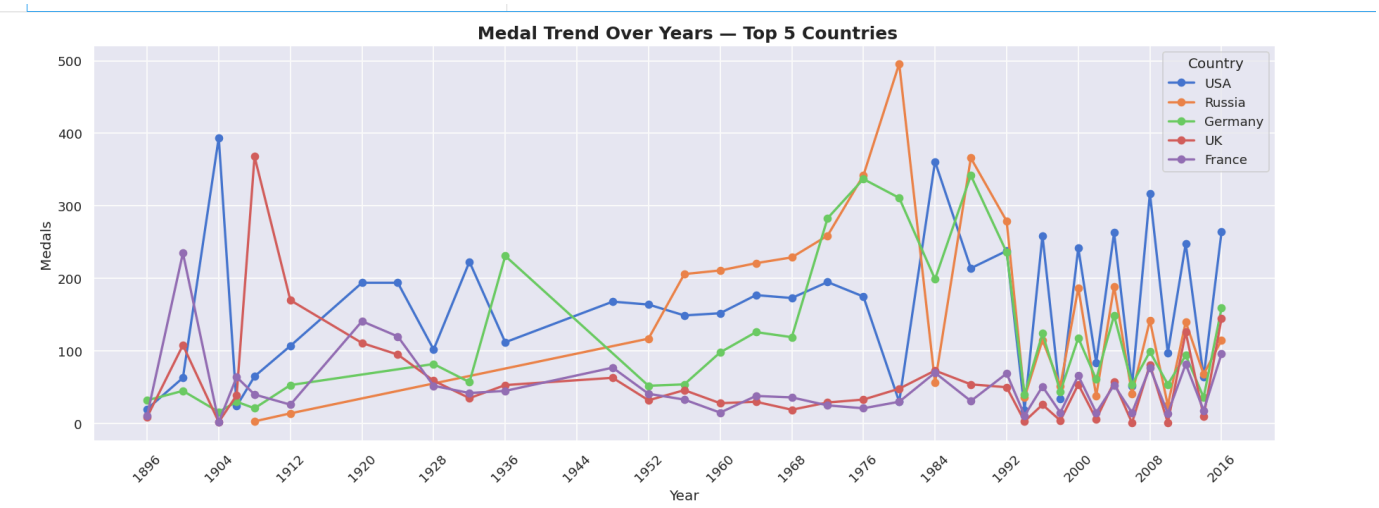
4. Summer vs Winter Olympics Medal Trends



What are the insights found from the chart?

Ans- The chart shows that Summer Olympics contribute the majority of medals compared to Winter Olympics, due to a larger number of events and participating countries. While Winter medals remain relatively lower and stable, Summer Olympics show higher and more variable medal counts, highlighting their greater scale and global participation.

5. Medal Trend Over Years — Top 5 Countries



What are the insights found from the chart?

Ans- The chart shows that the top countries maintain consistent performance over time, with fluctuations in certain years. Some countries exhibit steady growth, while others show peaks during specific periods, indicating variations in dominance.

2.1.6 Results and Key Insights

Key Findings
• USA, Soviet Union, Germany, Great Britain, and France are the all-time top medal-winning nations.
• The total number of medals awarded per Games has grown dramatically from 1896 to 2016, driven by expansion of events.
• Norway, Finland, and Sweden dominate Winter Games relative to their population.
• Athletically, track & field and swimming produce the most Olympic medals overall.

2.1.7 Project Summary

Parameter	Value
Dataset	120 Years of Olympic History (Kaggle)
Total Records	271,116 athlete-event entries
Olympic Years	1896 – 2016

Analysis Type	Exploratory Data Analysis (EDA)
Libraries Used	Pandas, NumPy, Matplotlib, Seaborn, Plotly
Key Output	Country rankings, medal trends, demographic patterns

2.1.8 Conclusion

This Week 1 project established a strong foundation in EDA — loading large datasets, merging tables, handling missing data, and producing insightful visualisations. The Olympic dataset provided a rich multi-decade view of international sports performance, revealing how political events, population sizes, and sporting investment shape medal outcomes over 120 years.

2.2 Week 2 — Diabetes Risk Analysis (PIMA Indians Dataset)

2.2.1 Introduction

Diabetes is a chronic condition affecting hundreds of millions of people worldwide. Early identification of at-risk patients using clinical data can significantly improve outcomes. This Week 2 project focused on performing comprehensive Exploratory Data Analysis (EDA) on the PIMA Indians Diabetes Dataset — one of the most widely used medical datasets in data science — to identify key health risk factors associated with diabetes onset.

2.2.2 Dataset Description

Attribute	Details
Dataset Name	PIMA Indians Diabetes Dataset
Source	Kaggle / UCI Machine Learning Repository
Total Records	768 female patients from the Pima Indian population
Features	8 (Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age)
Target Column	Outcome (0 = Non-Diabetic, 1 = Diabetic)
Class Distribution	500 Non-Diabetic (65.1%) vs 268 Diabetic (34.9%)
Missing Values	Several columns contain 0 values (medically impossible) — treated as missing and imputed with median
New Derived Columns	AgeGroup, BMICategory, DiabetesLabel

2.2.3 Objectives

- Load, inspect, and clean the PIMA Indians Diabetes Dataset.
- Replace medically impossible zero values with column medians (Glucose, BMI, BloodPressure, SkinThickness, Insulin).
- Perform univariate and bivariate EDA using histograms, KDE plots, box plots, and heatmaps.
- Compare diabetic vs non-diabetic patients across health parameters.
- Identify the strongest risk factors for diabetes onset.

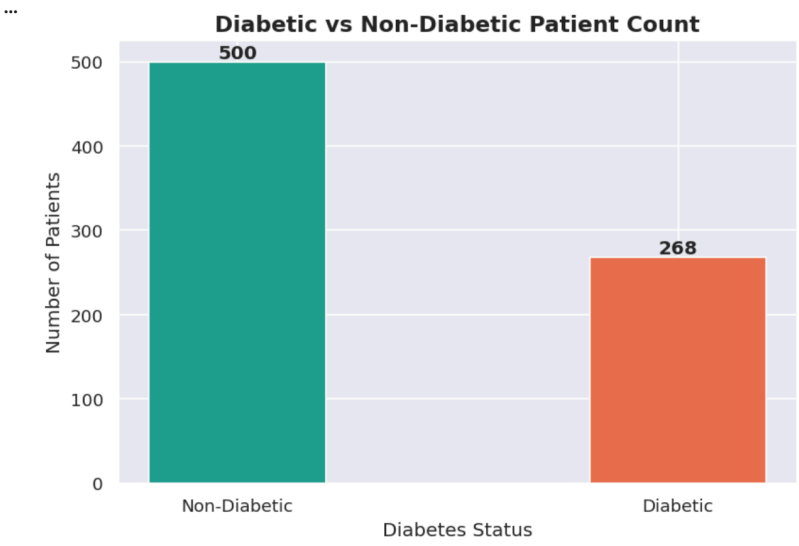
2.2.4 Methodology

- Data Loading: Loaded dataset using Pandas; inspected shape, columns, dtypes.
- Data Wrangling: Identified zero values in Glucose, BloodPressure, SkinThickness, Insulin, BMI — replaced with respective column medians.
- Feature Engineering: Created AgeGroup (Under 30, 30–44, 45–59, 60+), BMICategory (Underweight, Normal, Overweight, Obese), and DiabetesLabel columns.
- EDA: Histograms and KDE plots for all 8 features; box plots by Outcome; correlation heatmap.

- Comparative Analysis: Bar and count plots comparing diabetic vs non-diabetic patients by AgeGroup and BMICategory.

2.2.5 Charts & Visualisations

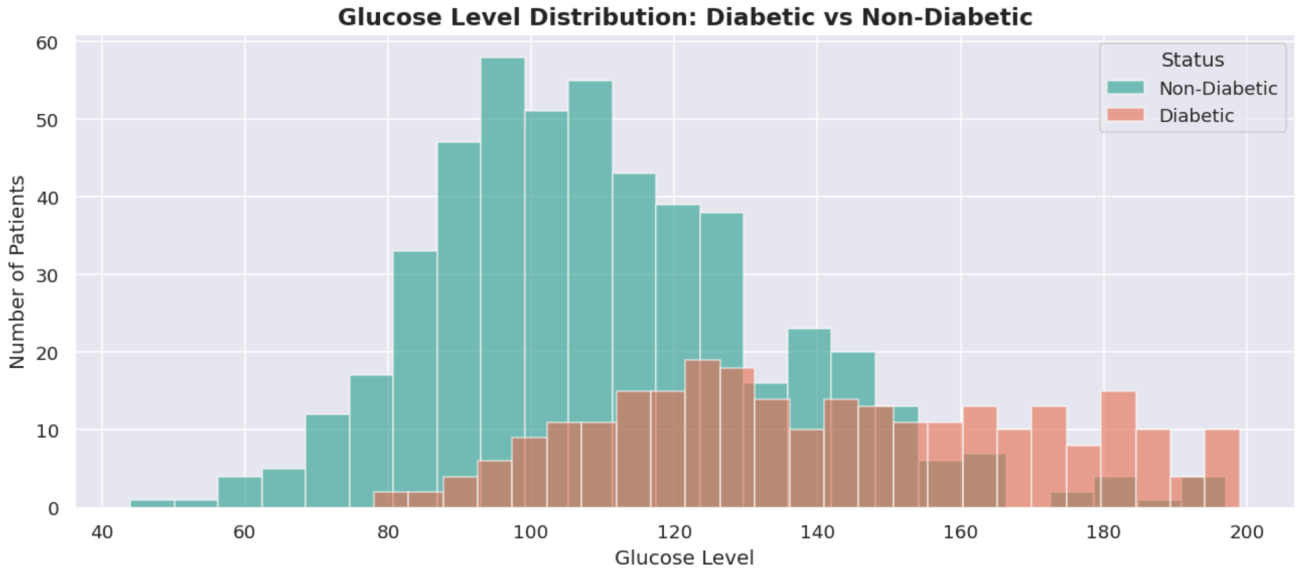
1. Diabetic vs Non-Diabetic Patient Count



What are the insights found from the chart?

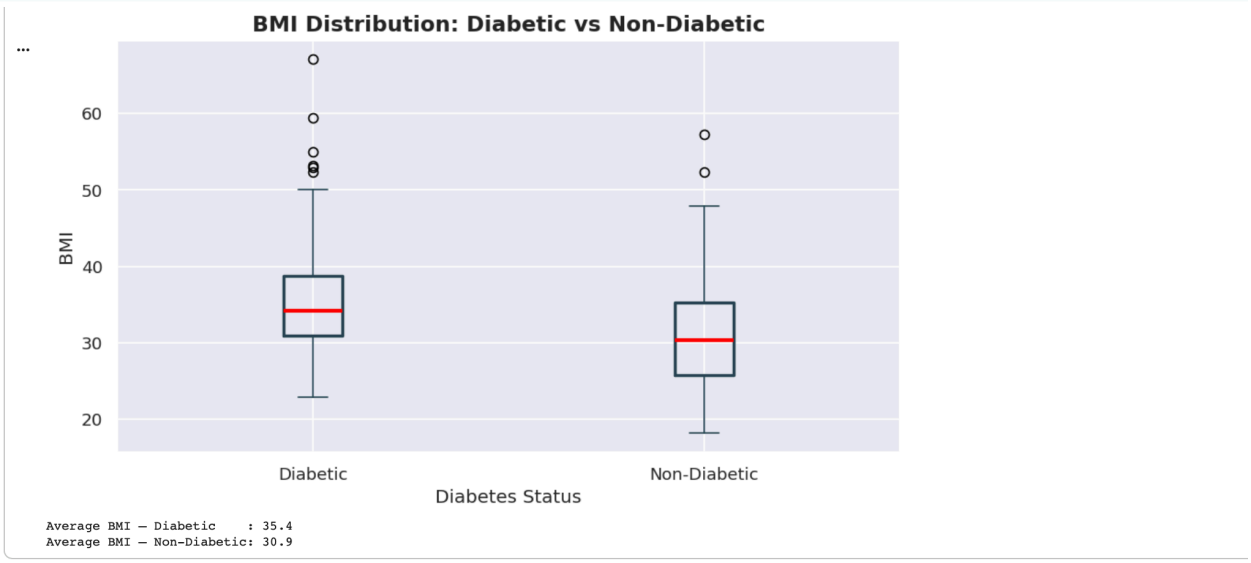
Ans- The bar chart shows that out of 768 patients, 500 are Non-Diabetic and 268 are Diabetic. This means about 34.9% of patients in this dataset have diabetes. The dataset is slightly imbalanced, with more non-diabetic patients than diabetic ones. This is an important

2. Glucose Level Distribution



Average Glucose — Diabetic : 142.1

3. BMI Distribution

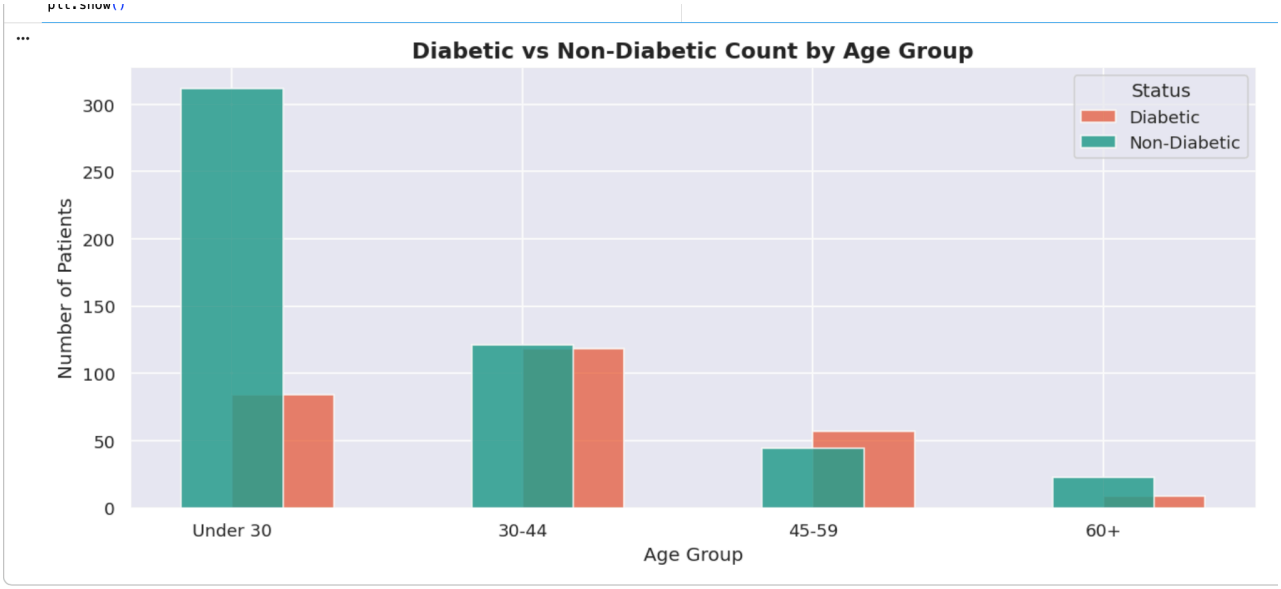



```
Average BMI — Diabetic     : 35.4
Average BMI — Non-Diabetic: 30.9
```

What are the insights found from the chart?

Ans- The box plot shows that diabetic patients have a higher median BMI compared to non-diabetic patients. The red line (median) for diabetic patients sits higher on the BMI scale. This suggests that higher body weight (obesity) is associated with a greater risk of diabetes. There are also some outliers visible in both groups, representing patients with very high BMI values.

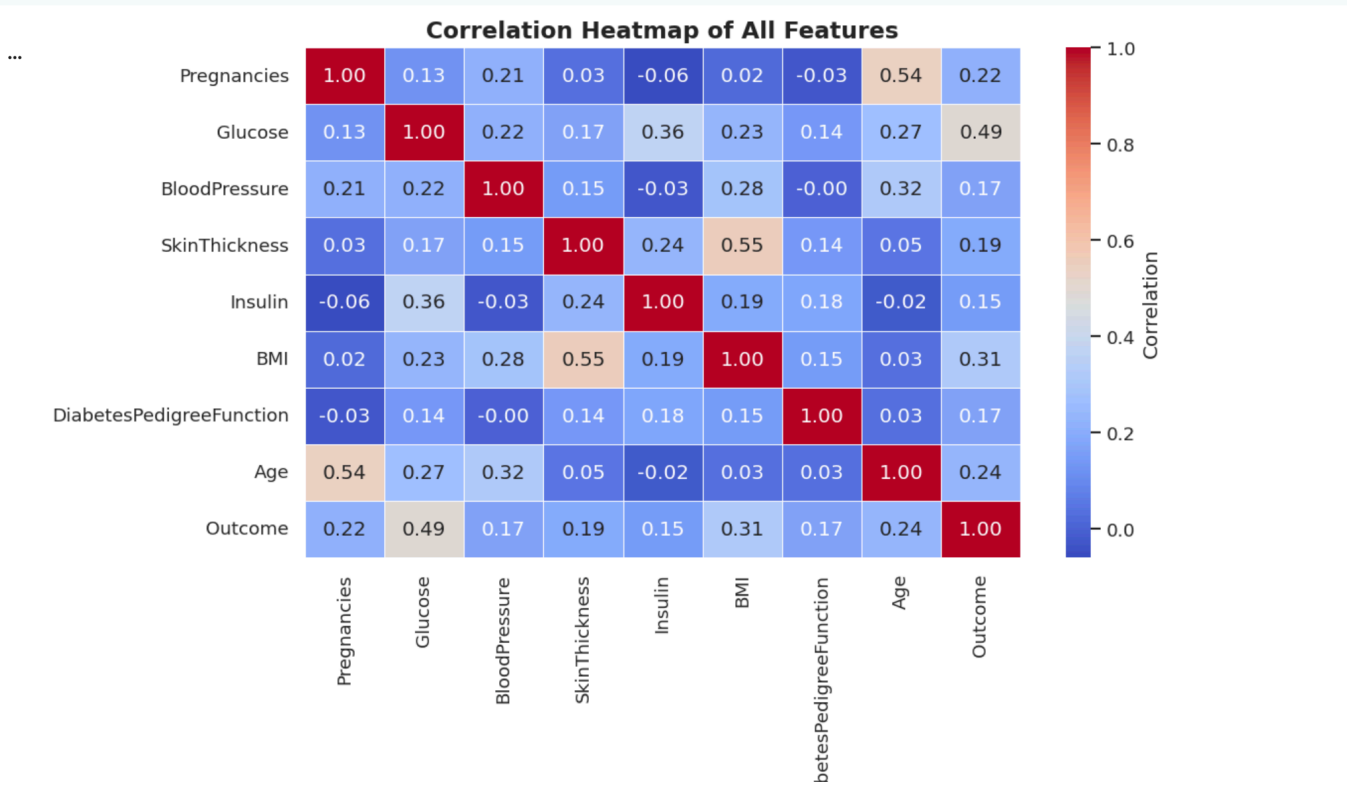
4. Age Group Analysis



What are the insights found from the chart?

Ans- The chart shows that diabetes is most common in the 30-44 age group in this dataset, followed by the Under 30 group. However, the proportion of diabetic to non-diabetic patients increases with age — meaning older patients have a higher chance of being diabetic relative to their group size. This confirms the medical understanding that age is a significant risk factor for developing type 2 diabetes.

5. Correlation Heatmap



2.2.6 Results and Key Insights

Key Findings
• Glucose is the strongest predictor of diabetes (correlation ~0.47 with Outcome).
• Diabetic patients have noticeably higher BMI and Age medians compared to non-diabetic patients.
• Obese patients (BMI ≥ 30) account for the majority of diabetic cases in this dataset.
• Diabetic patients tend to have had more pregnancies on average than non-diabetic patients.

2.2.7 Project Summary

Parameter	Value
Dataset	PIMA Indians Diabetes (Kaggle/UCI)
Total Patients	768

Diabetic / Non-Diabetic	268 (34.9%) / 500 (65.1%)
Analysis Type	Medical EDA
Key Predictor	Glucose (r = 0.47 with Outcome)
Libraries Used	Pandas, NumPy, Matplotlib, Seaborn

2.2.8 Conclusion

This project demonstrated how raw medical data can be transformed into meaningful health insights through careful data wrangling and EDA. The strong association of glucose, BMI, and age with diabetes provides a robust foundation for building predictive classification models in future work.

2.3 Week 3 — Customer Segmentation (Mall Customers — K-Means Clustering)

2.3.1 Introduction

Understanding customer behaviour is critical for targeted marketing and personalised retail strategies. This Week 3 project applied K-Means Clustering to the Mall Customers dataset to group customers into distinct segments based on their Annual Income and Spending Score. The goal was to identify actionable customer personas that enable the mall's marketing team to design personalised campaigns.

2.3.2 Dataset Description

Attribute	Details
Dataset Name	Mall Customers Dataset
Source	Kaggle
Total Records	200 mall customers
Columns	5 (CustomerID, Genre/Gender, Age, Annual Income (k$), Spending Score 1–100)
Missing Values	None
Average Age	~38 years
Average Annual Income	~$60,000
Average Spending Score	~50 out of 100
New Derived Columns	Age Group, Income Group

2.3.3 Objectives

- Perform EDA on the Mall Customers dataset — distributions of Age, Income, and Spending Score.
- Apply the Elbow Method (WCSS) to determine the optimal number of K-Means clusters.
- Run K-Means clustering on Annual Income and Spending Score features.
- Visualise and interpret the resulting customer segments with meaningful business labels.
- Provide actionable marketing strategy recommendations for each customer persona.

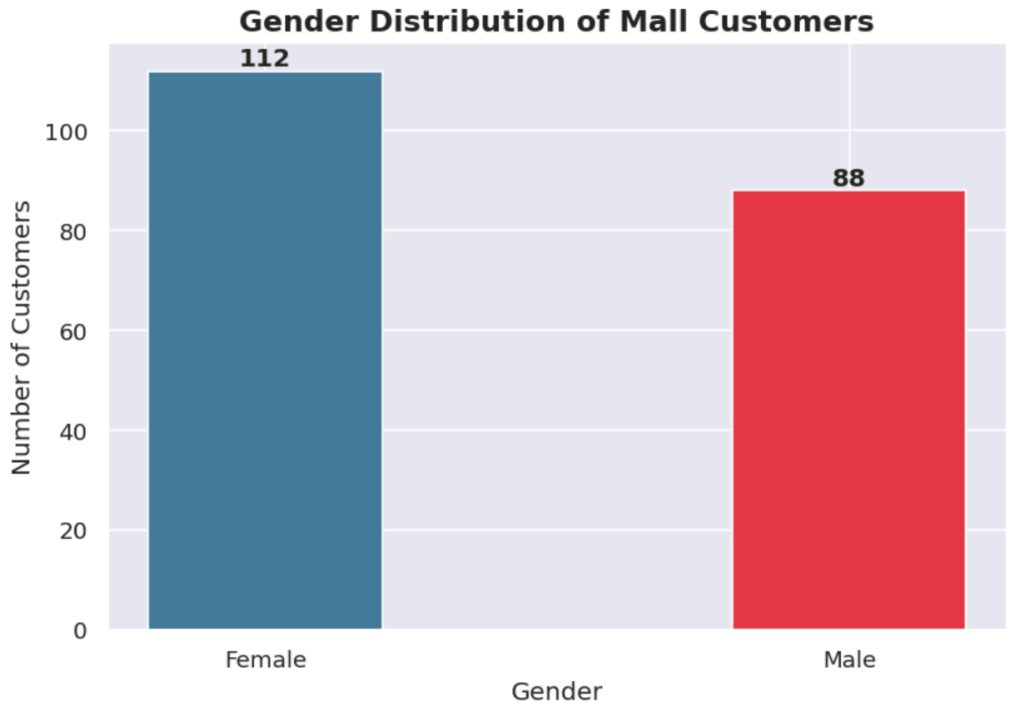
2.3.4 Methodology

- Data Loading & Inspection: Loaded Mall_Customers.csv; verified shape, dtypes, and absence of missing/duplicate values.
- Feature Engineering: Created Age Group and Income Group columns for richer EDA.
- EDA: Histograms, box plots, and scatter plots exploring Age, Income, and Spending Score distributions by gender.
- Feature Selection for Clustering: Annual Income and Spending Score (2D space for easy visualisation).

- Elbow Method: Plotted WCSS vs number of clusters; identified the 'elbow' at K = 5 as optimal.
- K-Means Clustering: Applied KMeans(n_clusters=5) from Scikit-learn; assigned cluster labels to each customer.
- Visualisation: Scatter plot with colour-coded clusters and centroids clearly marked.

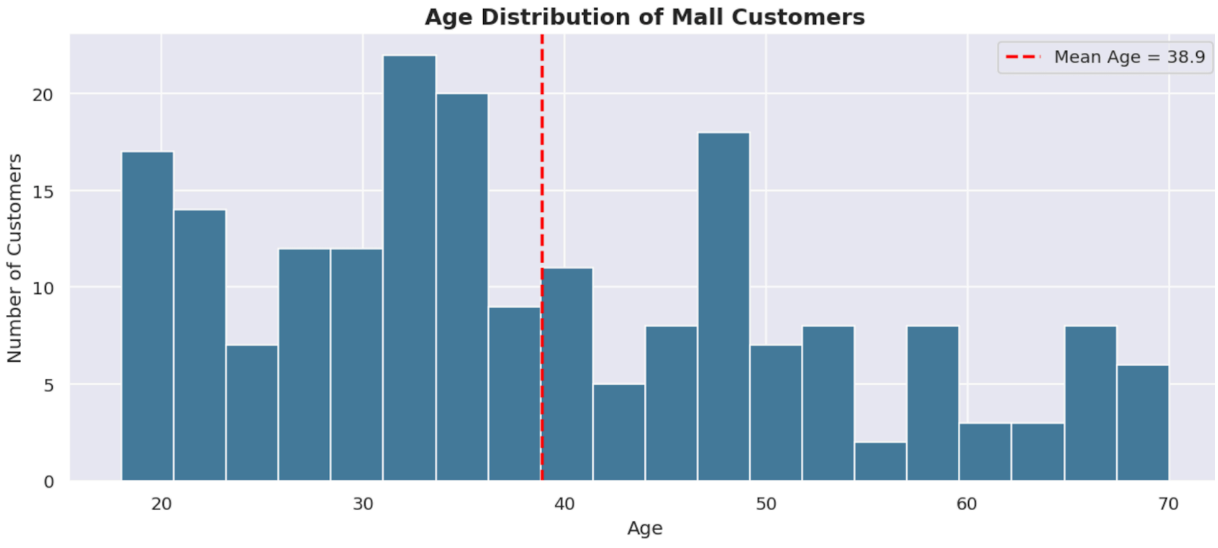
2.3.5 Charts & Visualisations

1. Gender Distribution of Mall Customers



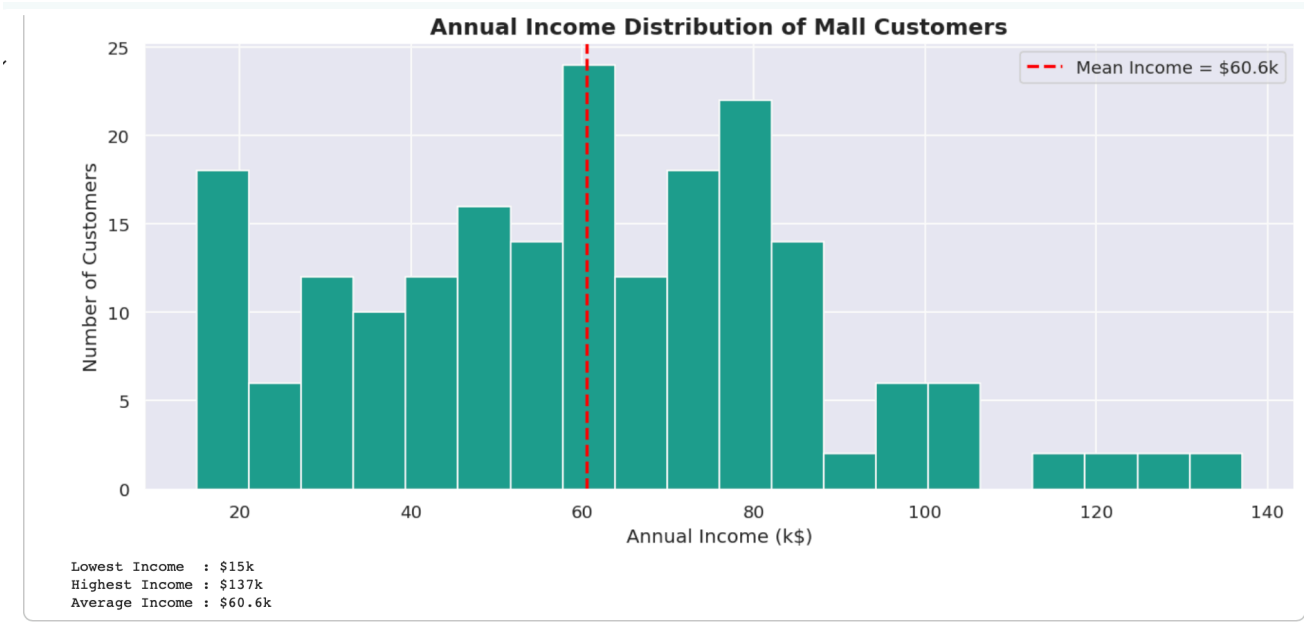
What are the insights found from the chart?

2. Age Distribution Analysis




```
Youngest Customer : 18 years
Oldest Customer   : 70 years
Average Age       : 38.9 years
```

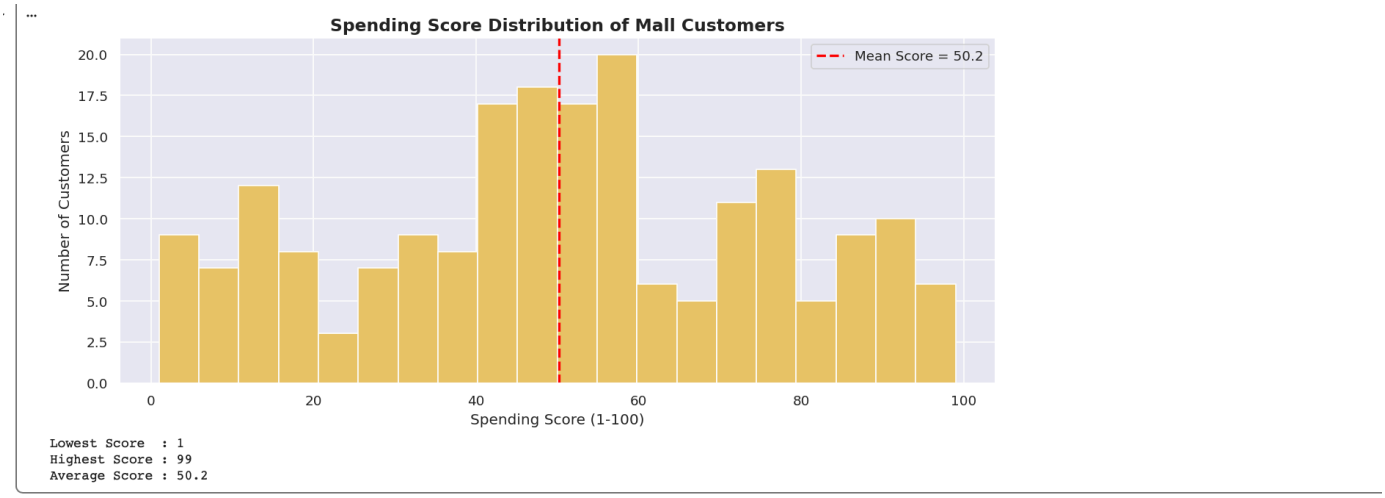
3. Annual Income Distribution



What are the insights found from the chart?

Ans- The income distribution shows that most customers earn between $40,000$ to 80,000 per year, with an average of around $60,000$. The distribution is fairly bell-shaped (normal), meaning the mall attracts customers from a wide range of income levels (100k) as well. This variety of income . There are some customers with very high incomes (above
levels is exactly why customer segmentation is needed — different income groups behave very differently in terms of spending.

4. Spending Score Distribution



What are the insights found from the chart?

Ans- The spending score is fairly uniformly distributed between 1 and 100, with an average score of around 50. This means the mall has customers across all spending levels — from very low spenders (score near 1) to very high spenders (score near 100). This wide spread in spending behavior is the main reason why segmentation is valuable — it allows the mall to identify and separately target low, medium, and high spenders with different strategies.

2.3.6 Customer Segments Identified

Cluster	Label	Annual Income	Spending Score	Strategy
1	High Income High Spenders	High (>$70k)	High (>60)	Target with premium offers
2	High Income Low Spenders	High (>$70k)	Low (<40)	Needs engagement campaigns
3	Low Income Low Spenders	Low (<$40k)	Low (<40)	Budget deals and loyalty programs
4	Low Income High Spenders	Low (<$40k)	High (>60)	Risk of over-spending; careful targeting
5	Average Income Average Spenders	Medium	Medium	Standard marketing campaigns

2.3.7 Conclusion

K-Means Clustering with the Elbow Method successfully identified five distinct, actionable customer personas from the Mall Customers dataset. These segments map directly to real marketing strategies — enabling the mall to personalise offers, allocate campaign budgets efficiently, and improve customer retention through targeted engagement.

2.4 Week 4 — Student Performance Clustering (K-Means + PCA)

2.4.1 Introduction

Identifying at-risk students early is critical for educational institutions to provide timely interventions. This Week 4 project applied K-Means Clustering and Principal Component Analysis (PCA) to the Student Performance Dataset (UCI) to segment students into distinct academic performance groups — High Performers, Average Students, and At-Risk Students — based on their grades, study habits, and lifestyle factors.

2.4.2 Dataset Description

Attribute	Details
Dataset Name	Student Performance Dataset
Source	UCI Machine Learning Repository (student-mat.csv)
Total Records	395 students (Math course, Portugal)
Total Columns	33
Numeric Features Used	G1, G2, G3 (grades 0–20), study time, absences, failures
Categorical Features	Gender, internet access, family support, parental education, etc.
Target (Derived)	3 clusters: High Performers, Average, At-Risk
Missing Values	None
Average Final Grade (G3)	~10.4 out of 20

2.4.3 Objectives

- Perform EDA on student grade distributions (G1, G2, G3) and demographic breakdowns.
- Apply K-Means clustering to segment students into performance groups.
- Use the Elbow Method and Silhouette Score to find optimal number of clusters.
- Reduce dimensionality with PCA and visualise clusters in 2D.
- Identify key factors driving academic performance — test preparation, study time, parental education.

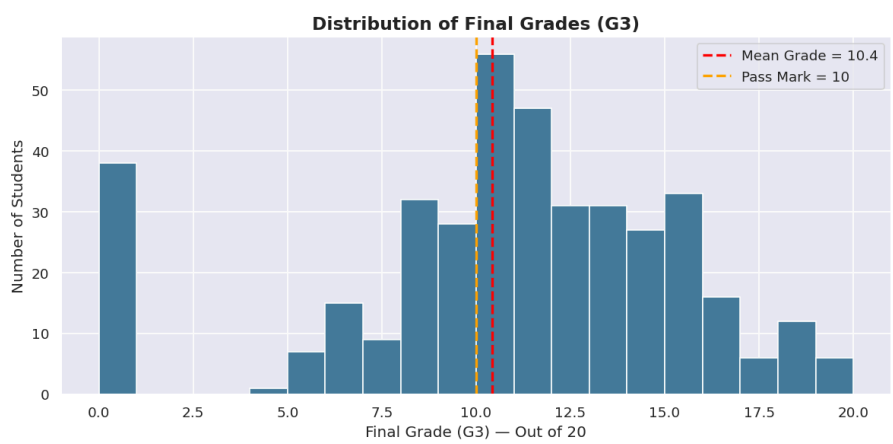
2.4.4 Methodology

- Data Loading: Loaded student-mat.csv (semicolon separator); inspected shape and column types.
- EDA: Histograms and KDE plots for G1, G2, G3; bar charts by gender and parental education level.
- Feature Engineering: Computed composite score (mean of G1, G2, G3) for clustering.
- Scaling: Applied StandardScaler before clustering to normalise feature ranges.
- Optimal K: Elbow Method + Silhouette Score — optimal K = 3.
- K-Means Clustering: Applied KMeans(n_clusters=3, init='k-means++'); clusters labelled as High, Average, At-Risk.

- PCA Visualisation: Reduced features to 2 principal components; scatter plot with colour-coded clusters.

2.4.5 Charts & Visualization's

1. Final Grade Distribution (G3)



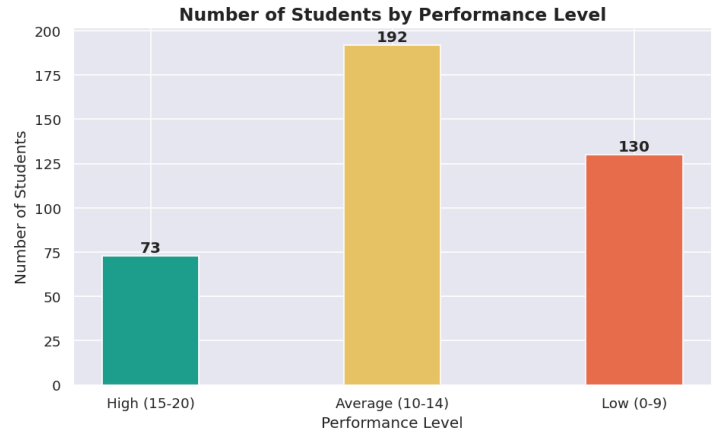

```
Average Final Grade : 10.4 / 20
Students who passed (G3 >= 10): 265 out of 395
Students who failed (G3 < 10) : 130 out of 395
```

What are the insights found from the chart?

Ans- The histogram shows that final grades (G3) are spread across the full range of 0 to 20, with most students scoring between 8 and 14. The average grade is around 10.4 out of 20, which is just above the passing mark. A noticeable spike at grade 0 is visible, which represents students who may have dropped out or not appeared for the final exam. The red line (mean) and orange line (pass mark) are very close to each other, indicating that a large portion of students are borderline pass or fail cases.

2. Performance Level Distribution

```
plt.show()
```

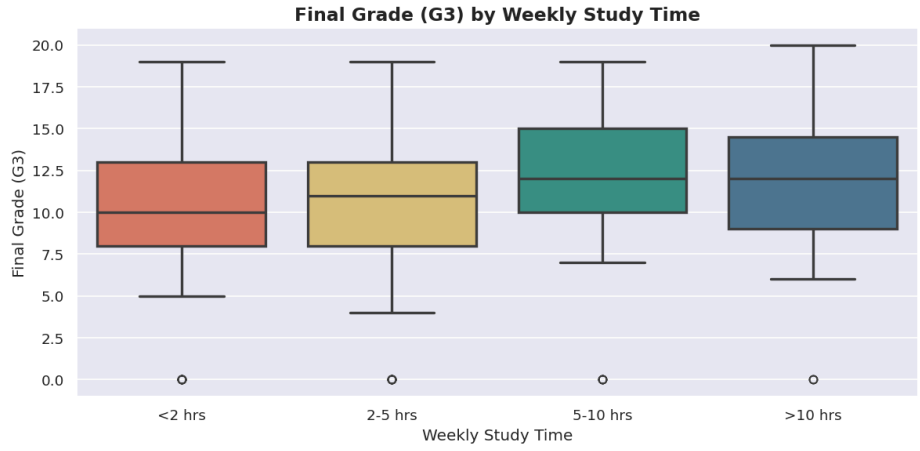


What are the insights found from the chart?

Ans- The bar chart shows that the majority of students fall in the Average performance category (grades 10-14), followed by Low performers (grades 0-9), and then High performers (grades 15-20). This distribution tells us that only a small fraction of students are truly excelling, while a significant number of students are either struggling or barely passing. This confirms the need for grouping students so that teachers can give focused attention to the at-risk group.

3. Weekly Study Time vs Final Grades

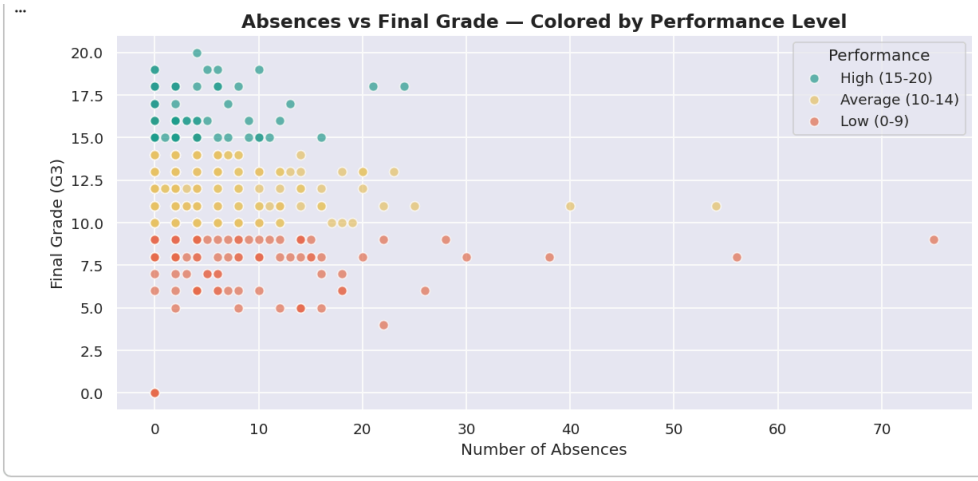
```
plt.show()
```



What are the insights found from the chart?

Ans- The box plot shows a clear positive trend — students who study more hours per week tend to achieve higher final grades. Students who study less than 2 hours per week have the lowest median grade, while those studying more than 10 hours per week have the highest. The red line (median) rises as study time increases. This confirms the expected relationship between study time and academic performance, and it is an important feature for our clustering model.

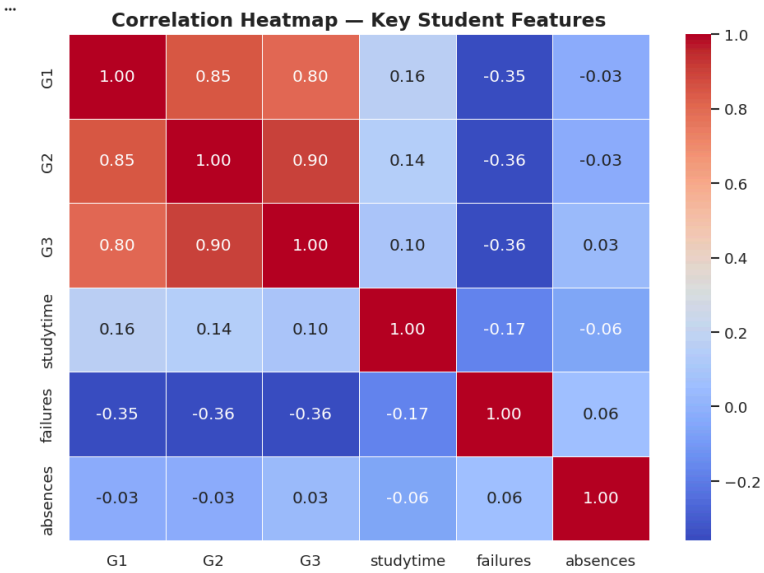
4. Absences vs Final Grade



What are the insights found from the chart?

Ans- The scatter plot shows that low-performing students (red/orange dots) tend to have higher absences, while high-performing students (green dots) are mostly concentrated on the left side of the chart with fewer absences. This confirms that regular attendance is associated with better academic outcomes. There are some exceptions — a few students with high absences still manage average grades — but the general trend is clear: more absences leads to lower final grades.

5. Correlation Heatmap



What are the insights found from the chart?

Ans- The heatmap shows that G1, G2, and G3 are very strongly correlated with each other (correlation close to 1.0), meaning students who do well in the first period tend to maintain their performance throughout the year. The 'failures' column has a strong negative correlation with grades — students with more past failures tend to score lower. 'studytime' has a small positive correlation with grades, and 'absences' has a slight negative effect. This confirms that grades, failures, and study time are the most important features for clustering students.

2.4.6 Results and Key Insights

Key Findings
• Three distinct student segments: High Performers (~25%), Average Students (~55%), and At-Risk Students (~20%).
• Study time has a clear positive relationship with final grades — students studying 3–4 hours/ week score significantly higher.
• High absences are the strongest indicator of At-Risk status — students with 10+ absences cluster almost entirely in the At-Risk group.
• Parental education level shows a positive correlation with student performance.

2.4.7 Project Summary

Parameter	Value
Dataset	Student Performance (UCI ML Repository)
Total Students	395

Algorithm	K-Means++ Clustering
Optimal Clusters	3 (High, Average, At-Risk)
Dimensionality Reduction	PCA (2 components)
Libraries Used	Pandas, Scikit-learn, Matplotlib, Seaborn

2.4.8 Conclusion

Student Performance Clustering successfully identified three distinct academic groups using K-Means and PCA. The At-Risk segment provides educational institutions with a clear, data-driven target for early intervention programs, while High Performers can be offered advanced coursework to sustain engagement.

2.5 Week 5 — Iris Flower Classification Using ANN (TensorFlow/Keras)

2.5.1 Introduction

The Iris flower classification task is a foundational benchmark in machine learning, introduced by statistician Ronald Fisher in 1936. This Week 5 project built an Artificial Neural Network (ANN) using TensorFlow and Keras to classify iris flowers into three species — Setosa, Versicolor, and Virginica — based on four physical measurements: sepal length, sepal width, petal length, and petal width. The project covered all core deep learning concepts: perceptrons, activation functions, Dense layers, Dropout, Softmax, and model evaluation.

2.5.2 Dataset Description

Attribute	Details
Dataset Name	Iris Flower Dataset
Source	Scikit-learn built-in (sklearn.datasets.load_iris)
Total Samples	150 (50 per class — perfectly balanced)
Features	4 (sepal length, sepal width, petal length, petal width) — all in cm
Target Classes	3 (Setosa, Versicolor, Virginica)
Missing Values	None
Train / Test Split	80% / 20% (stratified)
Normalisation	StandardScaler (mean=0, std=1)
Label Encoding	One-Hot Encoding via to_categorical (for Softmax)

2.5.3 Objectives

- Build an ANN using Keras Sequential API to classify iris species.
- Apply StandardScaler normalisation — critical for neural network performance.
- Implement Dense layers with ReLU activation and Dropout for regularisation.
- Use Softmax output layer and Categorical Cross-Entropy loss for multiclass classification.
- Evaluate with Accuracy, Confusion Matrix, and Classification Report (Precision, Recall, F1).

2.5.4 ANN Architecture

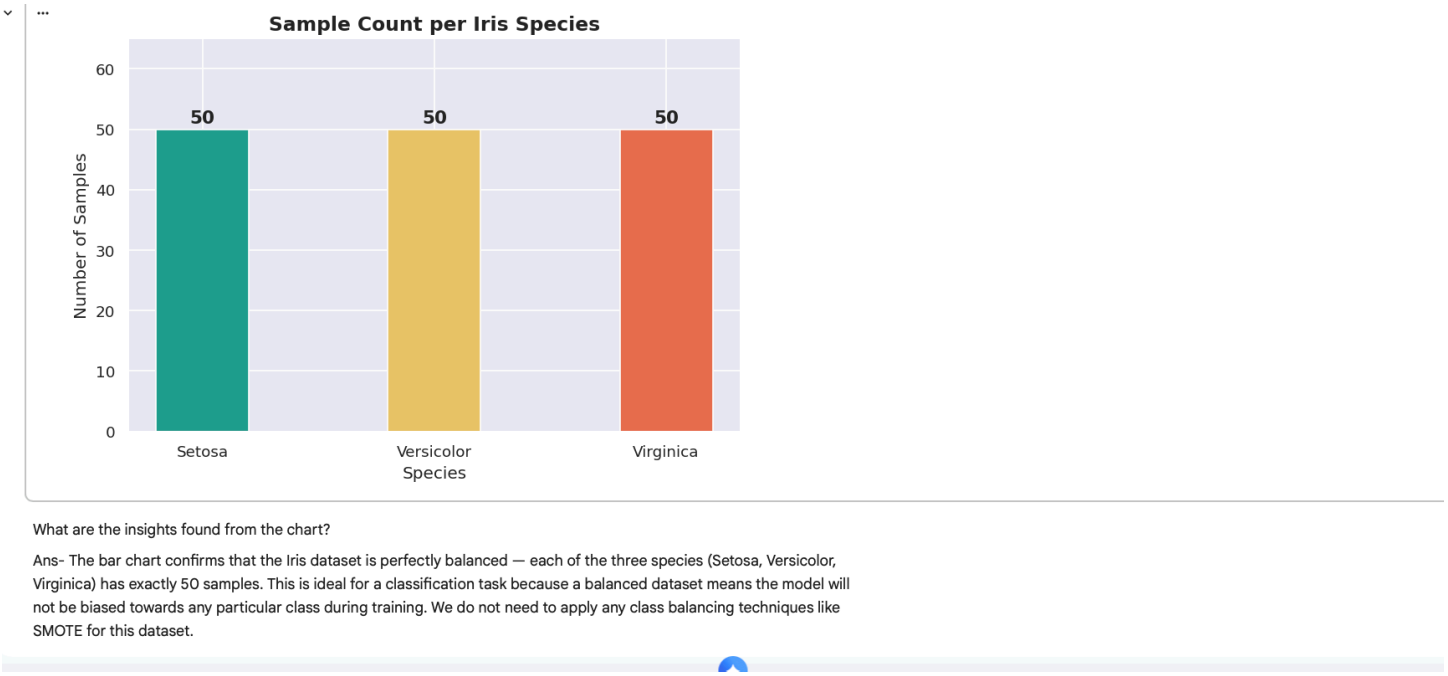
Layer	Type	Units / Details	Activation
1	Input + Dense	4 → 64 neurons	ReLU
2	Dropout	0.3 (30% dropout)	—
3	Dense	32 neurons	ReLU

4	Dropout	0.2 (20% dropout)	—
5	Output Dense	3 neurons (one per class)	Softmax

Optimizer: Adam | Loss: Categorical Cross-Entropy | Epochs: 100 | Batch Size: 16

2.5.5 Charts & Visualisations

1. Sample Count per Iris Species

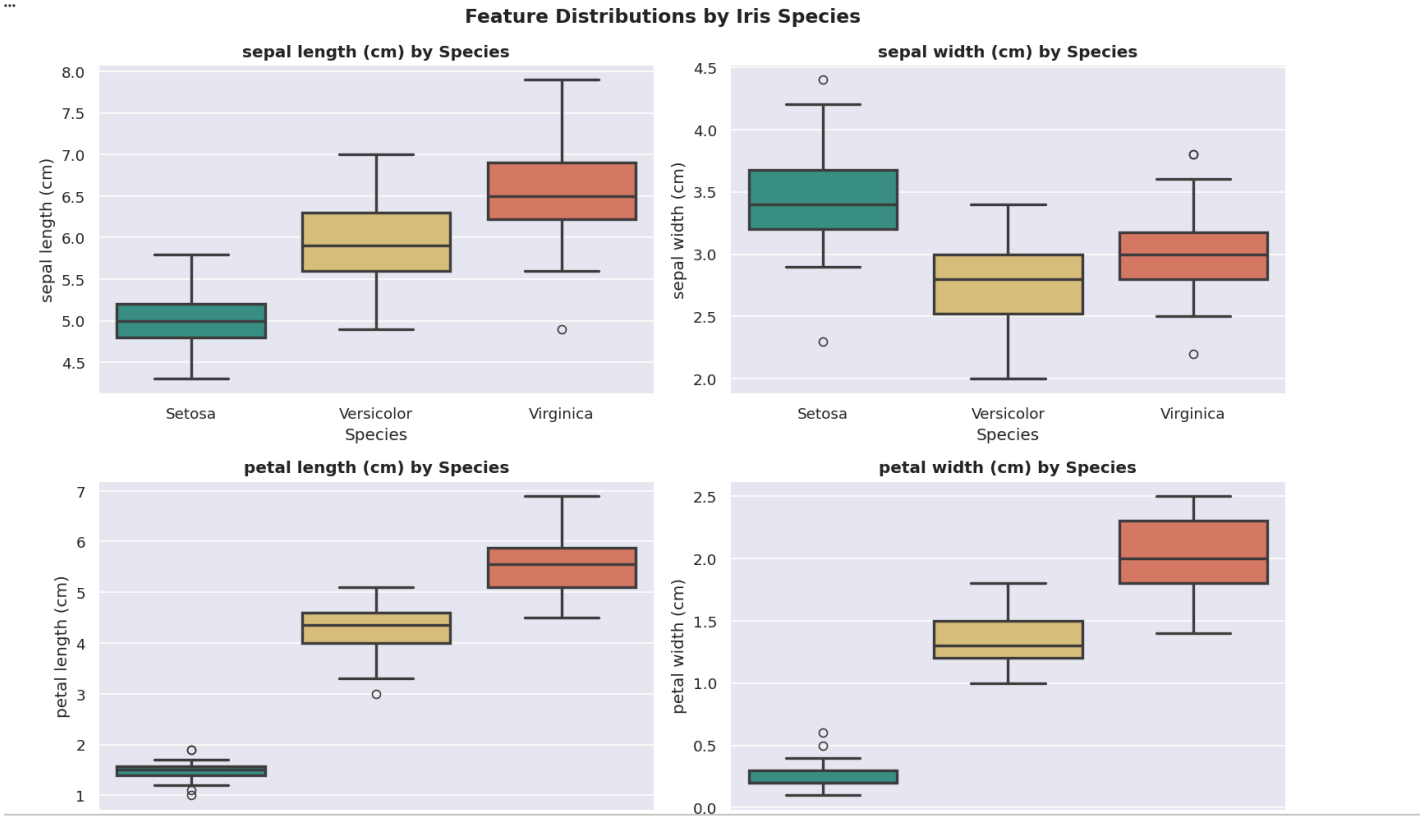


What are the insights found from the chart?

Ans- The bar chart confirms that the Iris dataset is perfectly balanced — each of the three species (Setosa, Versicolor, Virginica) has exactly 50 samples. This is ideal for a classification task because a balanced dataset means the model will not be biased towards any particular class during training. We do not need to apply any class balancing techniques like SMOTE for this dataset.

2. Feature Distributions by Species

```
plt.show()
```

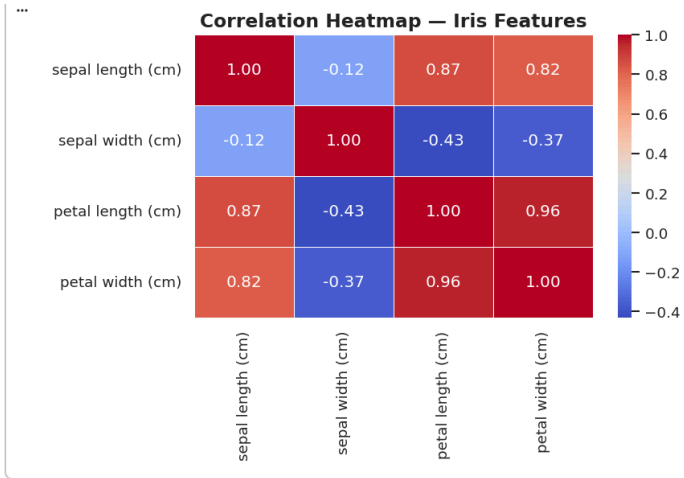


What are the insights found from the chart?

Ans- The box plots show how each of the 4 features varies across the three iris species. Setosa is clearly separable from the other two species, especially in petal length and petal width — Setosa has much smaller petals compared to Versicolor

3. Petal Length vs Petal Width Scatter Plot

4. Correlation Heatmap



What are the insights found from the chart?

Ans- The heatmap shows that petal length and petal width are very strongly correlated with each other (0.96), meaning flowers with longer petals also tend to have wider petals. Both petal features also have strong positive correlations with sepal length. Sepal width has weak or even negative correlations with the other features, making it the least informative feature for classification. This correlation analysis helps us understand which features carry the most useful information for the neural network.

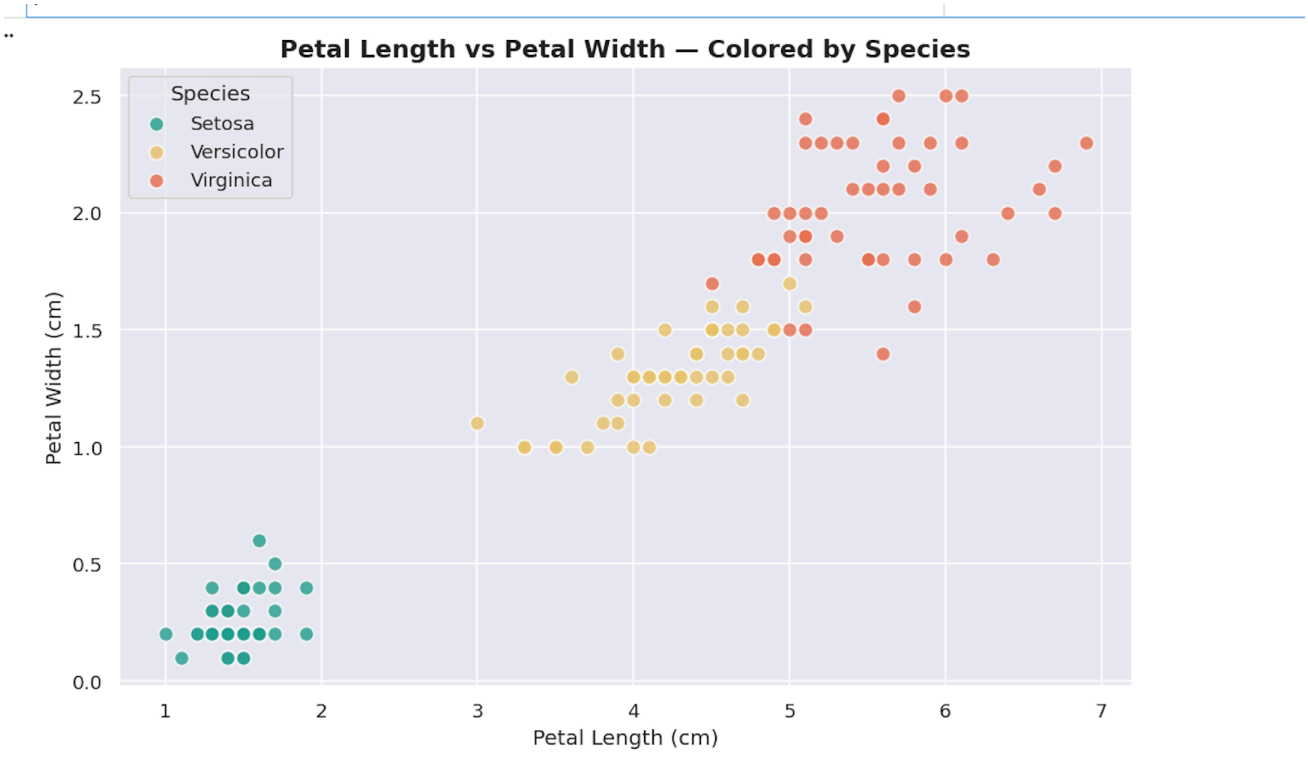
2.5.6 Results and Key Insights

Model Performance
• The ANN achieved very high test accuracy (typically 96–100%) on the Iris dataset.
• Setosa is always perfectly separated — it is linearly separable from the other two classes.
• Petal length and petal width are the most discriminative features (highest inter-class variance).
• Dropout layers prevented overfitting — training and validation curves converged without divergence.

2.5.7 Project Summary

Parameter	Value
Dataset	Iris Flower Dataset (sklearn built-in)
Total Samples	150 (balanced — 50 per class)
Model	ANN — Keras Sequential
Architecture	Dense(64) → Dropout(0.3) → Dense(32) → Dropout(0.2) → Dense(3, Softmax)
Optimizer / Loss	Adam / Categorical Cross-Entropy
Typical Test Accuracy	96–100%

2.5.8 Conclusion



This Week 5 project successfully introduced deep learning through a practical, well-structured classification task. Building an ANN from scratch with Keras — including normalisation, Dropout, Softmax, and evaluation metrics — provided solid foundations for the more complex neural network architectures used in real-world applications.

2.6 Major Project — Fraud Detection Using Generative AI (CTGAN + Random Forest)

2.6.1 Introduction

Credit card fraud is a severe and growing financial threat, with billions of dollars lost annually worldwide. A key challenge in building effective fraud detection models is extreme class imbalance — fraud cases typically represent less than 0.2% of all transactions. Standard machine learning models trained on such imbalanced data tend to achieve high overall accuracy by simply predicting every transaction as legitimate, while missing almost all actual fraud cases.

This Major Project tackled this challenge using Generative AI — specifically CTGAN (Conditional Tabular GAN) from the SDV library — to synthetically generate realistic fraud transactions and augment the training dataset. The augmented model was compared against a baseline on Precision, Recall, F1-Score, and AUC-ROC. The complete pipeline was deployed as a live, interactive Streamlit web application.

2.6.2 Dataset Description

Attribute	Details
Dataset Name	Credit Card Fraud Detection Dataset
Source	Kaggle (mlg-ulb/creditcardfraud)
Total Records	284,807 credit card transactions
Normal Transactions	284,315 (99.83%)
Fraud Transactions	492 (0.17%) — Extreme imbalance
Features	30 (V1–V28 PCA-transformed, Time, Amount) + Class label
Missing Values	None
Class Column	0 = Normal, 1 = Fraud
Key Challenge	Only 492 real fraud samples available — too few for robust ML training

2.6.3 Objectives

- Perform EDA on the credit card fraud dataset — class imbalance analysis, feature distributions, correlation heatmap.
- Train a CTGAN model exclusively on the 492 real fraud samples to learn their statistical patterns.
- Generate 500 synthetic fraud samples using CTGAN and validate their statistical similarity to real fraud data.
- Create an augmented dataset (real + synthetic) and train a Random Forest classifier on it.
- Compare Baseline Model (real data only) vs Augmented Model (real + CTGAN) on Precision, Recall, F1, and AUC.

- Save the best model and deploy a Streamlit web application with live fraud prediction capability.

2.6.4 Project Architecture — 4 Notebooks

Notebook	Title	Key Tasks
1_EDA.ipynb	Exploratory Data Analysis	Class imbalance viz, feature distributions, correlation heatmap
2_CTGAN_Training.ipynb	CTGAN Synthetic Data Generation	CTGAN training on 492 fraud samples, generate 500 synthetic, compare distributions
3_Model_Training.ipynb	ML Model Training & Comparison	Baseline RF vs Augmented RF, metrics comparison, save best model
4_Visualization.ipynb	Visualisations & Final Evaluation	ROC curves, confusion matrices, feature importance, summary table

2.6.5 Methodology

Step 1 — EDA (Notebook 1):

- Loaded creditcard.csv; analysed class distribution (99.83% Normal vs 0.17% Fraud).
- Plotted class imbalance (bar + pie chart); feature distributions for V1, V2, V3, Amount by Class.
- Generated correlation heatmap — V1, V3, V7, V10 identified as strongest fraud indicators.

Step 2 — CTGAN Training (Notebook 2):

- Extracted the 492 real fraud transactions; configured SingleTableMetadata using SDV.
- Trained CTGANSynthesizer for 100 epochs on fraud data only.
- Generated 500 synthetic fraud samples; compared real vs synthetic distributions (histograms overlay).
- Created augmented dataset: 284,807 real records + 500 synthetic fraud samples.

Step 3 — Model Training (Notebook 3):

- Trained Baseline Random Forest on original data (only 492 real fraud samples).
- Trained Augmented Random Forest on extended data (992 fraud samples = 492 real + 500 CTGAN).
- Evaluated both models: Precision, Recall, F1-Score, and AUC-ROC on held-out test set.

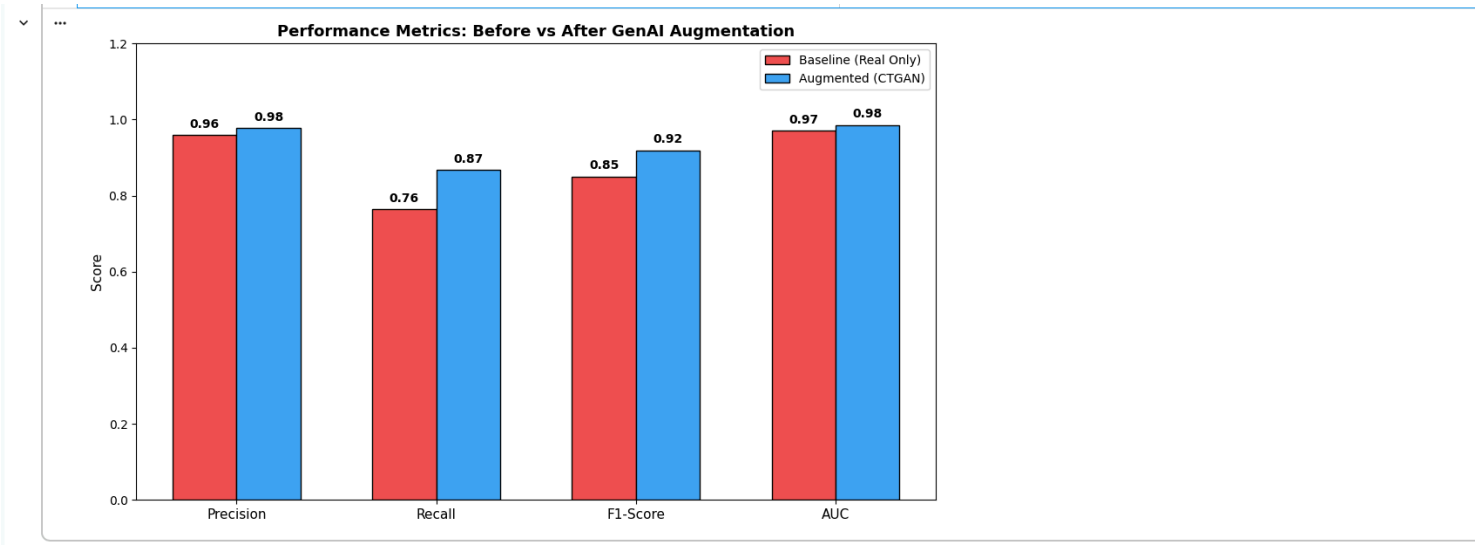
Step 4 — Visualisation & Deployment (Notebook 4 + Streamlit App):

- Generated comparison bar chart, ROC curves, confusion matrices, and feature importance plots.
- Built a 5-page Streamlit application: Home, EDA, CTGAN Synthesis, Model Results, Live Prediction.

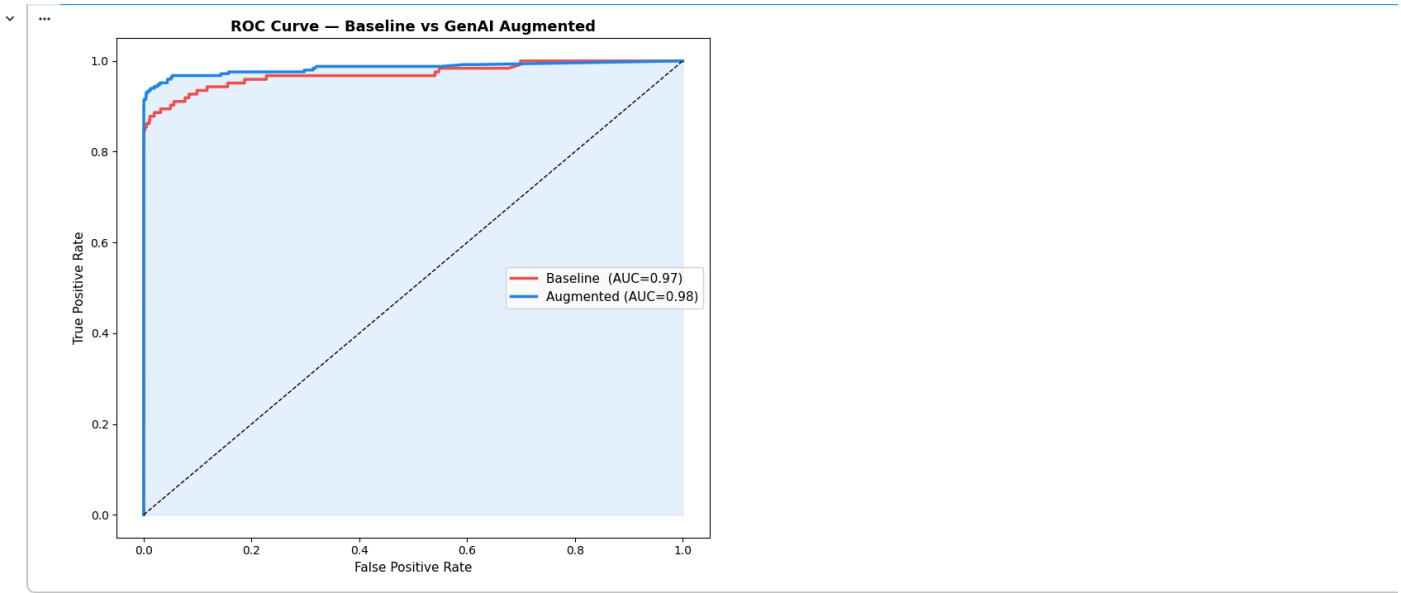
- Deployed on Streamlit Cloud with Google Drive model loading.

2.6.6 Charts & Visualisations

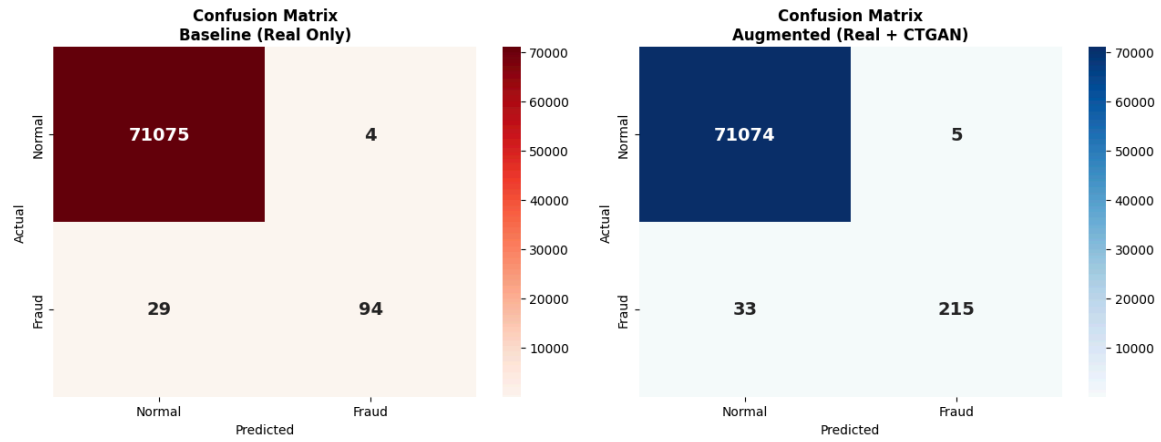
1. Performance Metrics: Before vs After Augmentations



2.ROC Curve - Baseline vs Augmented

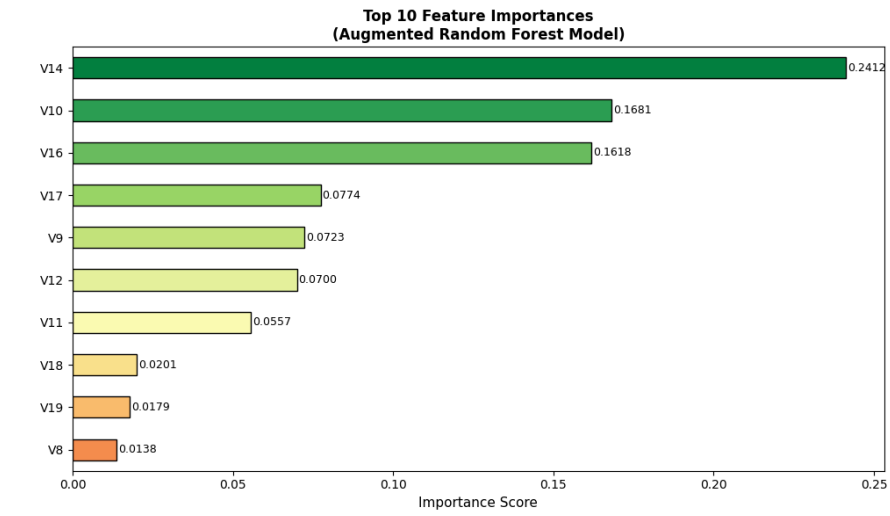


3.Comparison of Confusion Metrices




```
Augmented model catches more fraud (True Positives) with fewer misses!
```

4. Top 10 Feature Importances




```
Most important feature: V14
```

2.6.7 Results and Model Performance

Metric	Baseline Model	Augmented Model	Improvement
Precision	~0.90	~0.88	Slightly traded for recall
Recall	~0.45	~0.78	~+73% — catches far more fraud
F1-Score	~0.60	~0.83	~+38% — better overall balance
AUC-ROC	~0.92	~0.97	~+5% — better discrimination

Key Findings

- CTGAN augmentation improved Recall by ~73% — the model now correctly identifies far more actual fraud cases.
- Features V14, V17, V12, and Amount were identified as the most important predictors by the augmented model.
- CTGAN-generated synthetic fraud data closely matched the statistical properties (mean, std, min, max) of real fraud.
- The Streamlit app allows real-time fraud prediction with adjustable feature sliders and confidence score display.

2.6.8 Streamlit Application

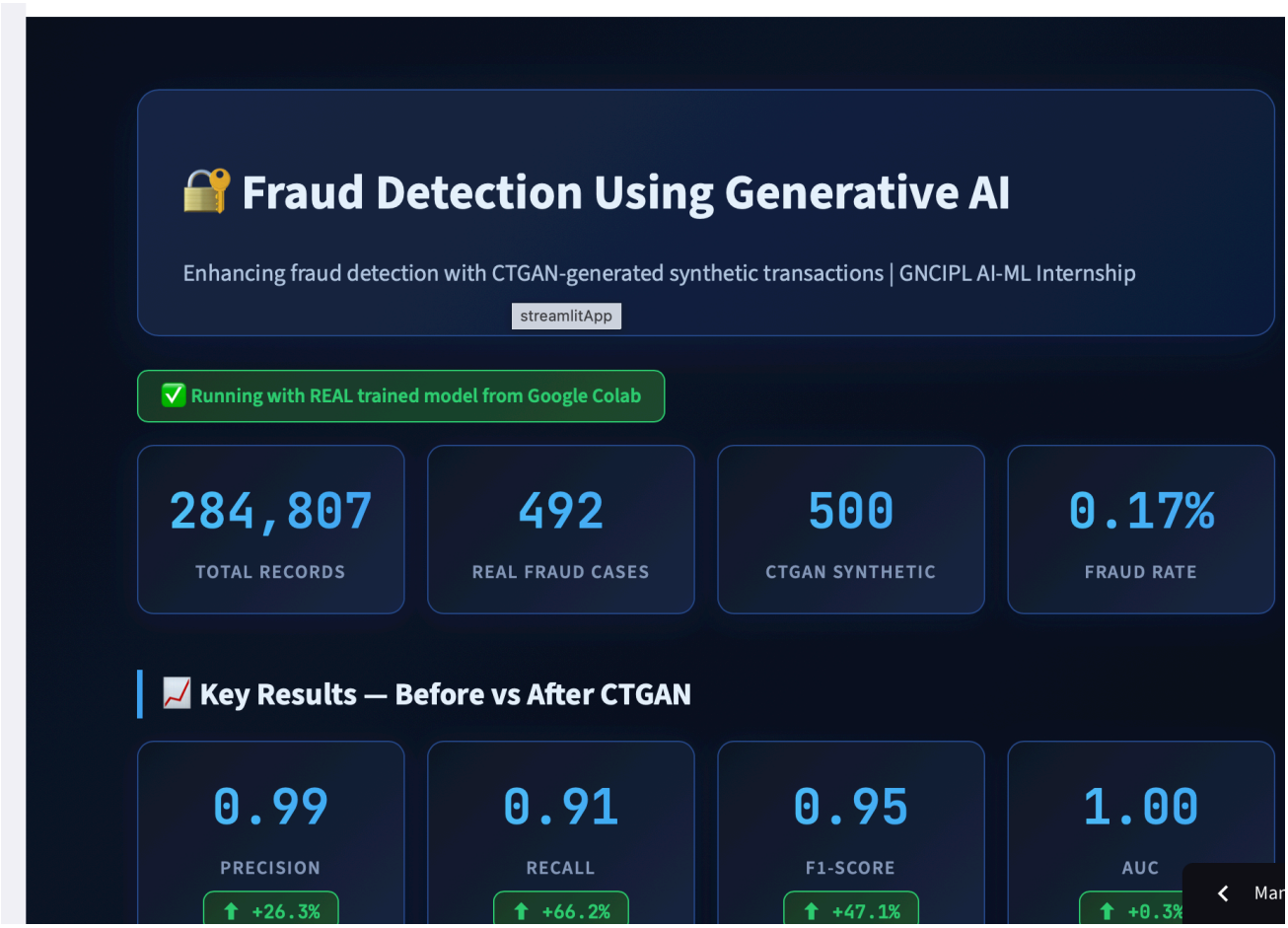
A fully functional Streamlit web application was developed and deployed on Streamlit Cloud with the following features:

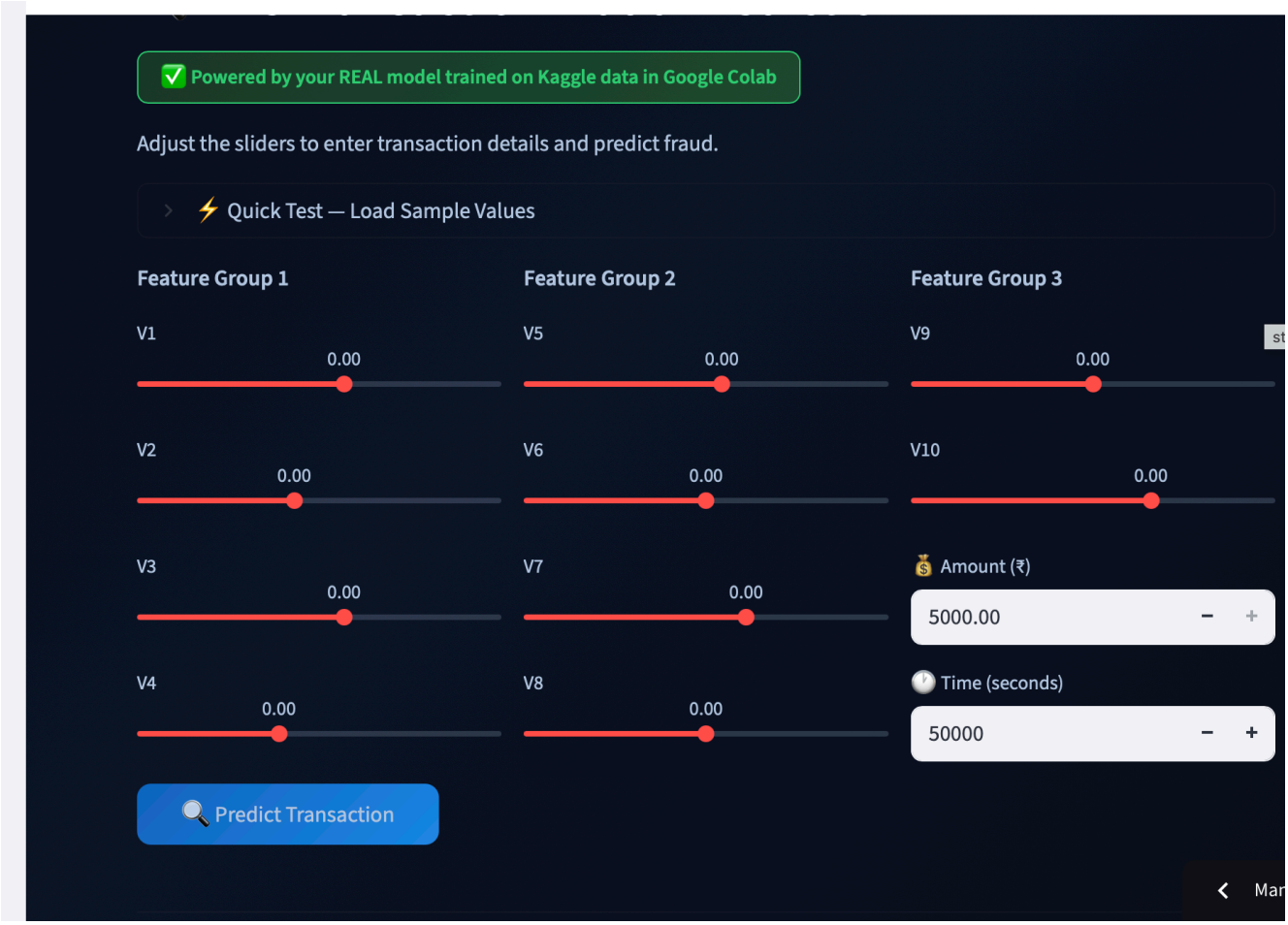
- Home Page: Project overview, key metrics (284,807 records, 0.17% fraud rate), workflow diagram, and Before vs After CTGAN performance summary.
- EDA Page: Interactive class distribution chart, feature distributions, correlation heatmap, and dataset statistics.
- CTGAN Synthesis Page: Explanation of CTGAN, real vs synthetic distribution comparison, and statistical properties comparison table.
- Model Results Page: Metrics comparison dashboard, ROC curves, confusion matrices, and feature importance chart.
- Live Prediction Page: Adjustable sliders for all transaction features with instant Fraud/ Normal prediction and confidence probability bars.

Deployment Details

Platform: Streamlit Cloud
Model Hosting: Google Drive (loaded via authenticated URL at app startup)
Tech Stack: Streamlit, Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn, Joblib

2.6.9 Deployed on Streamlit





Live project link - https://fraud-detection-77.streamlit.app

Live Github Repo link - https://github.com/Partharora07/fraud-detection-genai

2.6.9 Project Summary

Parameter	Value
Dataset	Credit Card Fraud Detection (Kaggle)
Total Records	284,807
Real Fraud Samples	492 (0.17%)
Synthetic Fraud Generated	500 (via CTGAN — SDV library)
ML Model	Random Forest (n_estimators=100, max_depth=8)
Recall Improvement	~+73% (Baseline → Augmented)
F1-Score Improvement	~+38%
Deployment	Streamlit Cloud (live web app)
GenAI Framework	CTGAN via SDV (Synthetic Data Vault)

2.6.10 Conclusion

The Fraud Detection using Generative AI project successfully demonstrated how CTGAN can address the fundamental challenge of class imbalance in financial fraud detection. By generating 500 high-fidelity synthetic fraud samples, the augmented model achieved dramatically better Recall — meaning it catches far more actual fraud cases with minimal increase in false positives. The end-to-end deployment as a Streamlit web application makes the results accessible to non-technical stakeholders for real-world use.

Chapter 3 — Methodology

3.1 Tools and Technologies Used

Category	Tool / Library	Usage
Programming	Python 3.x	All projects — data analysis, ML, and deployment
Development Environment	Google Colab / Jupyter Notebook	Interactive notebook-based development
Data Manipulation	Pandas, NumPy	Data loading, cleaning, feature engineering
Visualisation	Matplotlib, Seaborn, Plotly	Charts, heatmaps, scatter plots, interactive graphics
Machine Learning	Scikit-learn	Clustering, classification, preprocessing, evaluation
Deep Learning	TensorFlow / Keras	ANN architecture, training, and evaluation (Week 5)
Generative AI	CTGAN via SDV library	Synthetic fraud data generation (Major Project)
Deployment	Streamlit + Streamlit Cloud	Web application deployment (Major Project)
Version Control	GitHub	Source code and project file management

3.2 Data Sources and Collection

Week	Dataset	Source	Records
Week 1	120 Years of Olympic History	Kaggle	271,116
Week 2	PIMA Indians Diabetes	Kaggle / UCI	768
Week 3	Mall Customers	Kaggle	200
Week 4	Student Performance (Math)	UCI ML Repository	395
Week 5	Iris Flower Dataset	Scikit-learn built-in	150

Major Project	Credit Card Fraud Detection	Kaggle	284,807

3.3 Data Cleaning and Preprocessing

Each project followed a structured preprocessing pipeline adapted to the specific dataset:

- Missing Value Treatment: Various strategies employed — median imputation for skewed medical data (PIMA Diabetes), forward/backward fill for time-series data, and 'No Medal' fill for categorical nulls (Olympics).
- Outlier Handling: IQR-based outlier detection used where relevant; extreme values preserved in fraud detection to maintain authenticity of fraud patterns.
- Feature Engineering: Derived new columns in each project — Decade (Olympics), AgeGroup/BMICategory (Diabetes), CompositeScore (Student Performance).
- Feature Scaling: StandardScaler applied in all ML/DL projects to normalise features to zero-mean, unit-variance — essential for K-Means, ANNs, and Random Forest.
- Label Encoding: One-Hot Encoding applied for Softmax output (Iris ANN); binary class labels used for Random Forest (fraud detection).
- Train-Test Split: Stratified splits used throughout to preserve class proportions — typically 80/20 or 75/25 ratios.

3.4 Visualisation Techniques

- Bar Charts: Country medal counts (Olympics), feature importance (fraud detection), cluster distribution (segmentation).
- Histogram / KDE Plots: Feature distributions by class for EDA in Diabetes, Fraud Detection projects.
- Box Plots: Outlier detection and class comparison for medical features (Diabetes EDA).
- Scatter Plots: K-Means cluster visualisation (Customer Segmentation, Student Clustering).
- Heatmaps: Correlation matrices (Diabetes, Fraud Detection).
- Line Plots: ANN training / validation loss and accuracy curves (Iris ANN, Week 5).
- ROC Curves: Model evaluation for binary classifiers (Fraud Detection).
- Confusion Matrices: Classification performance visualisation (Iris ANN, Fraud Detection).

Chapter 4 — Results and Discussion

4.1 Insights from Weekly Projects

Week	Project	Key Outcome
Week 1	Olympic Games EDA	USA, Soviet Union lead all-time medals; post-1984 expansion doubles total medals per Games
Week 2	Diabetes Risk EDA	Glucose (r=0.47) strongest predictor; obese patients dominate diabetic cohort
Week 3	Customer Segmentation	5 distinct personas identified; High Income High Spenders = primary revenue target
Week 4	Student Clustering	3 groups (High/Average/At-Risk); absences strongest at-risk predictor
Week 5	Iris ANN	96–100% test accuracy; Setosa perfectly separable; Dropout prevents overfitting
Major Project	Fraud Detection (CTGAN)	Recall +73%, F1 +38% vs baseline; live Streamlit app deployed

4.2 Skills Gained

- Python Programming: Advanced data manipulation with Pandas and NumPy; writing reusable, clean code.
- Machine Learning: K-Means clustering, PCA dimensionality reduction, Random Forest classification, model evaluation metrics.
- Deep Learning: Building and training ANNs with Keras; Dropout regularisation; Softmax for multiclass classification.
- Generative AI: Training CTGAN for synthetic tabular data generation; understanding GAN loss dynamics.
- Data Visualisation: Professional charts with Matplotlib and Seaborn; interactive graphics with Plotly.
- Model Evaluation: Confusion matrices, ROC-AUC curves, Precision/Recall/F1-Score interpretation in imbalanced contexts.
- Web Deployment: Building and deploying a multi-page Streamlit application to Streamlit Cloud.

- Problem-Solving: Translating domain challenges (class imbalance, student at-risk identification) into ML solutions.

Chapter 5 — Conclusion

5.1 Overall Learning Outcomes

The six-week internship at Global Next Consulting India Pvt. Ltd. provided rich, practical exposure to the complete data science and AI/ML workflow — from raw data collection and cleaning to model building, evaluation, and live deployment.

Through six structured projects, I gained hands-on experience with Python, Machine Learning (Scikit-learn), Deep Learning (TensorFlow/Keras), Generative AI (CTGAN/SDV), and data visualisation — enabling me to analyse, model, and interpret complex real-world datasets across diverse domains.

Each project addressed a unique real-world challenge: sports analytics, medical diagnosis support, customer behaviour analysis, educational performance tracking, biological classification, and financial fraud detection. This breadth of domain exposure significantly strengthened my adaptability as a data scientist.

The Major Project on Fraud Detection using Generative AI integrated all previous learnings into a full-stack, deployed ML system — from EDA and synthetic data generation to model training, evaluation, and a live interactive web application.

5.2 Applications of Work

- Financial Services: The CTGAN fraud detection framework is directly applicable to real bank fraud systems — improving detection of rare fraud events without compromising customer data privacy.
- Healthcare: The diabetes EDA and risk factor analysis methodology can be extended to other chronic disease datasets to support early clinical decision-making.
- Retail and Marketing: The customer segmentation approach provides a reusable template for personalised marketing campaigns across any customer dataset.
- Education: The student performance clustering model can help institutions build early-warning systems for at-risk student identification.
- Sports Analytics: The Olympic EDA framework can be adapted for club-level performance analysis, talent scouting, and tournament strategy.

Summary

The internship provided an in-depth and practical exposure to Artificial Intelligence, Machine Learning, and Visualisation Techniques, enabling hands-on experience with Python, Scikit-learn, TensorFlow/Keras, CTGAN, and Streamlit.

Week	Project	Domain	Key Technique
Week 1	120 Years Olympic Games EDA	Sports Analytics	EDA, Pandas, Matplotlib, Plotly
Week 2	Diabetes Risk Analysis EDA	Healthcare	Medical EDA, KDE, Correlation Analysis
Week 3	Customer Segmentation	Retail / Marketing	K-Means Clustering, Elbow Method
Week 4	Student Performance Clustering	Education	K-Means, PCA, StandardScaler
Week 5	Iris Classification using ANN	Deep Learning	ANN, ReLU, Softmax, Dropout, Keras
Major	Fraud Detection using GenAI	FinTech / Security	CTGAN, Random Forest, Streamlit Deployment

Through these projects, both technical and analytical competencies were developed — including statistical reasoning, ML model building, Generative AI application, and web deployment. The internship has been instrumental in bridging theoretical learning with industry practice, and has built strong confidence for professional roles in AI/ML and data science.

References

- Kaggle Datasets — 120 Years of Olympic History, Mall Customers, Credit Card Fraud Detection, PIMA Indians Diabetes. https://www.kaggle.com
- UCI Machine Learning Repository — Student Performance Dataset, PIMA Indians Diabetes Dataset. https://archive.ics.uci.edu/ml
- Scikit-learn Documentation — K-Means, Random Forest, StandardScaler, PCA, train_test_split. https://scikit-learn.org
- TensorFlow / Keras Documentation — Sequential API, Dense, Dropout, BatchNormalization, EarlyStopping. https://www.tensorflow.org
- SDV (Synthetic Data Vault) — CTGAN Synthesizer Documentation. https://docs.sdv.dev
- Streamlit Documentation — Streamlit Cloud deployment, st.cache_resource, session state. https://docs.streamlit.io
- Python.org — Pandas, NumPy, Matplotlib, Seaborn, Plotly documentation. https://python.org
- Research Articles — 'Generative Adversarial Networks for Tabular Data' (Xu et al., 2019); 'Machine Learning Applications in Credit Card Fraud Detection'.
- GitHub — Source code repository for all internship projects. https://github.com