DATA ANALYSIS USING PYTHON

A Capstone Project

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

2203A54035 **Sahith Mandala**

Under the guidance of

Dr. Ramesh Dadi Sir

Assistant Professor, Department of CSE.

Submitted to

COMPUTER SCIENCE
SCHOOL OF COMPUTER SCIENCE
AND ARTIFICIAL INTELLIGENCE

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE
SR UNIVERSITY, ANANTHASAGAR, WARANGAL
March, 2025.

1 POPULATION DATASET -1

1.Abstract:

This project analyzes structured data from the 2015 World Happiness Report using data science and statistical techniques. The objective is to explore and interpret the relationship between demographic and economic indicators and the Happiness Score of various countries. Data preprocessing steps include handling missing values and duplicates. Visualization tools and summary statistics provide insight into regional distributions and key contributing factors. The analysis does not use machine learning models for prediction but focuses on uncovering trends through descriptive analytics. Statistical methods such as correlation analysis and comparative visualizations were applied to understand group differences across regions. This approach provides foundational insights for policymaking, public health evaluation, and socio-economic research.

2.Introduction:

Understanding the factors that contribute to human well-being and national development is a critical area of research in the fields of social science, economics, and public policy. As global datasets become more accessible, data-driven approaches are increasingly employed to analyze population metrics and their influence on happiness and quality of life. In this project, we explore the 2015 World Happiness Report dataset to identify patterns and relationships between various socio-economic indicators and happiness scores across countries. Unlike predictive modeling efforts common in machine learning, this analysis focuses on descriptive statistics, visual exploration, and correlation-based insights to derive meaningful conclusions. By using tools like data visualization and statistical summarization, the study highlights regional differences and contributing factors, providing a basis for future investigations into global well-being.

3.Dataset Description:

The dataset used in this project is derived from the 2015 World Happiness Report, which ranks countries based on various indicators of well-being and national development. It consists of data collected from surveys and official economic sources across multiple countries worldwide.

Key features of the dataset include:

- **Country**: Name of the country.
- **Region**: Geopolitical region classification.

- **Happiness Rank**: The rank of the country based on the Happiness Score.
- **Happiness Score**: A composite score that reflects subjective well-being.
- **Economy (GDP per Capita)**: Economic performance per person.
- **Family**: Quality of family and social support.
- **Health (Life Expectancy)**: Life expectancy at birth.
- **Freedom**: Perceived freedom to make life choices.
- **Trust (Government Corruption)**: Perception of corruption in government and business.
- **Generosity**: Willingness to donate or help others.
- **Dystopia Residual**: A calculated variable that serves as a benchmark for the worst possible country in each of the other categories.

The dataset enables a broad analysis of how various national indicators influence overall happiness. It contains both qualitative and quantitative data and is suitable for exploratory data analysis, correlation studies, and socio-economic insight extraction.

4.Methodology:

The methodology for this project involves several key steps designed to extract insights from the 2015 World Happiness Report dataset using exploratory data analysis (EDA). The overall process is outlined as follows:

1. Data Acquisition and Loading
   - The dataset was obtained from Kaggle and imported using Python libraries such as pandas.
   - Initial inspection was performed to understand data structure and content.
2. Data Preprocessing
   - Missing Values: Checked for null entries using is null() and visualized with missing no to ensure data completeness.
   - Duplicates: Identified and removed any duplicated rows to maintain data integrity.
   - Data Types: Ensured all columns had appropriate data types for analysis.
3. Descriptive Statistics
   - Summary statistics were computed for all numerical features (mean, standard deviation, min/max).
   - These summaries helped understand the central tendency and dispersion within the dataset.
4. Data Visualization
   - Plotted histograms and pie charts using matplotlib and seaborn to illustrate:
     - Distribution of Happiness Scores.
     - Regional contributions to happiness.

    - Relationships among indicators such as GDP, Health, Freedom, etc.
- Correlation heatmaps were used to identify relationships between independent variables and Happiness Score.

5. Regional Analysis
    - Grouped data by region to compare averages and variances across regions.
    - Visualizations were used to show disparities and trends among different world regions.
6. Statistical Exploration
    - Although the project did not apply predictive models, basic statistical techniques were used to explore:
        - Correlation between indicators.
        - Comparative analysis across regions.

5.Implementation Highlights:

This section summarizes the core technical steps and notable implementations during the exploratory analysis of the 2015 World Happiness dataset.

Data Loading and Environment Setup

- Used standard Python libraries including pandas, numpy, matplotlib, and seaborn for data handling and visualization.
- The dataset was imported from Kaggle and loaded into a Pandas Data Frame for analysis.

Data Cleaning

- Verified the dataset for **missing values** using isnull().sum() and missingno.matrix() plots — found no significant missing data.
- Identified and removed **duplicate records** using duplicated().sum() to ensure the dataset was clean and unique.

Statistical Summary

- Generated descriptive statistics using DataFrame.describe() to understand the distribution, central tendency, and spread of numerical features.

Visual Exploration

- **Histogram**: Plotted the distribution of Happiness Scores to understand its spread across countries.
- **Pie Chart**: Visualized the proportion of total Happiness Score contributions by region using group by and plot(kind='pie').
- **Bar Charts**: Compared average indicator values (e.g., GDP, Family, Freedom) across regions.
- **Heatmap**: Used seaborn.heatmap() to show correlation between variables and identify strong relationships.

Regional Analysis

- Grouped countries by region and computed means of key indicators.

- Allowed side-by-side comparisons of how regions performed on factors like health, generosity, and trust.

Insights Extraction

- Found that indicators like **Economy**, **Health**, and **Family** showed strong positive correlation with Happiness Score.
- **Trust** and **Freedom** varied significantly between regions, highlighting cultural and political influence on happiness.

6.Results:

6.1. Data Visualization:

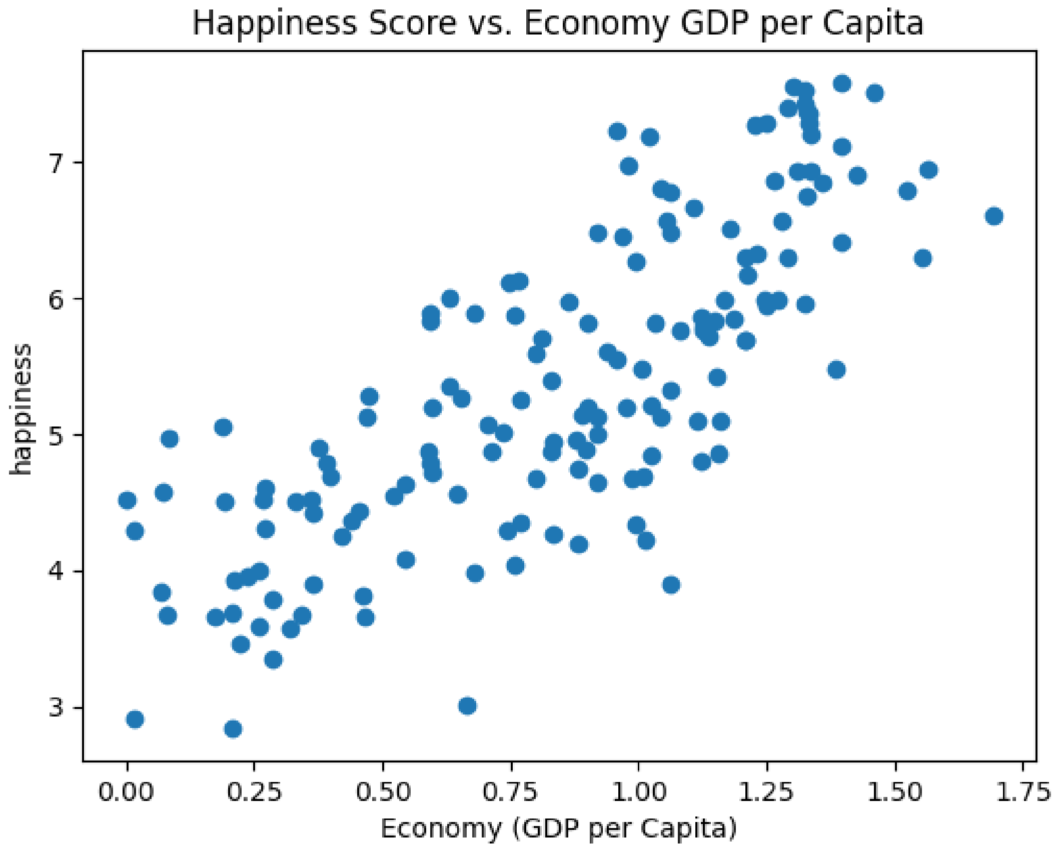
To understand the relationships between the Happiness Score and various socio-economic indicators, several visual tools were employed. These visualizations allow for intuitive and insightful observations about global well-being.

6.1.1 Scatter Plots:

Scatter plots were used to examine pairwise relationships between the **Happiness Score** and key contributing variables. These plots help reveal the nature (linear or non-linear) and strength of correlations between variables.

Key Scatter Plot Observations:

- **Happiness Score vs. Economy (GDP per Capita):**
  - A strong positive linear correlation is observed. Countries with higher GDP tend to have higher Happiness Scores.
  - Indicates that economic prosperity plays a major role in national happiness.
- **Happiness Score vs. Health (Life Expectancy):**
  - Also shows a positive correlation. Nations with longer life expectancy often report higher happiness.
  - Suggests a close link between public health and well-being.
- **Happiness Score vs. Family (Social Support):**
  - Displays a clear upward trend. Stronger social support systems are associated with higher happiness levels.
  - Highlights the importance of community and relationships in life satisfaction.
- **Happiness Score vs. Freedom:**
  - A moderate positive correlation is evident. Countries with higher perceived freedom tend to be happier.
- **Happiness Score vs. Trust (Government Corruption):**
  - This relationship appears more scattered. While some correlation exists, it is weaker and less consistent than other indicators.



6.1.2. Histogram:

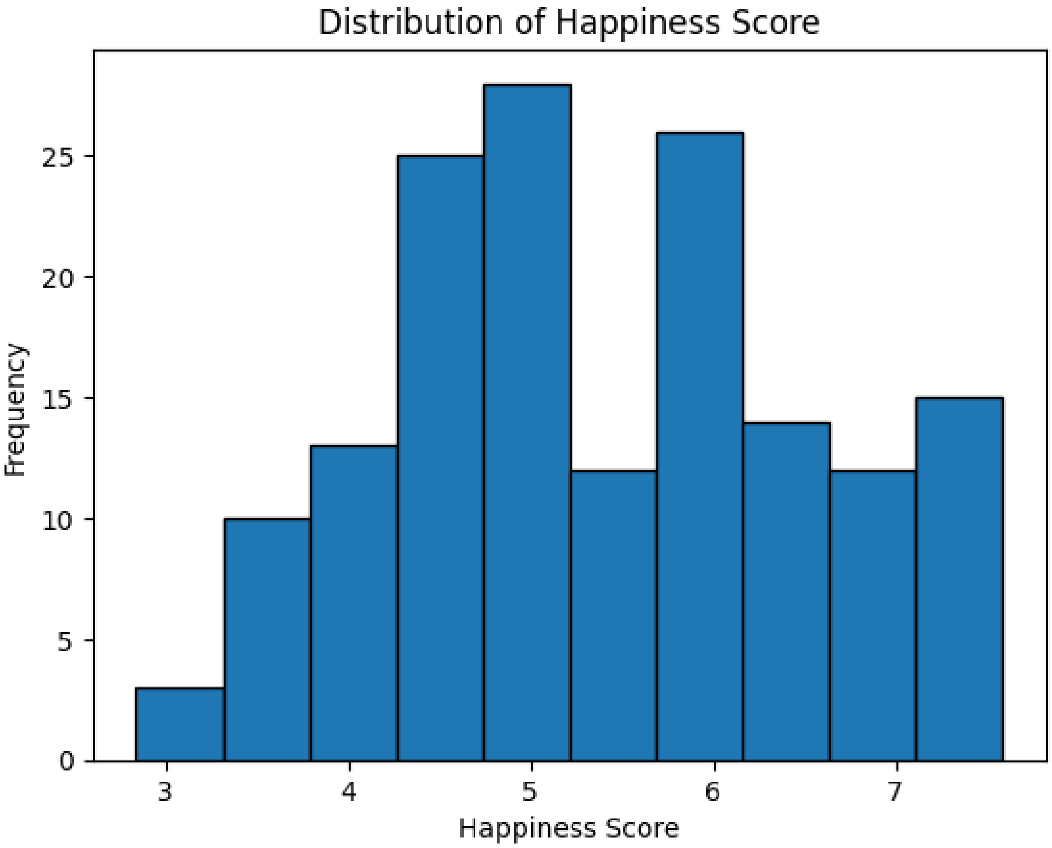
A histogram was plotted to visualize the **distribution of Happiness Scores** across all countries in the dataset. This helps understand the overall spread and frequency of happiness levels worldwide.

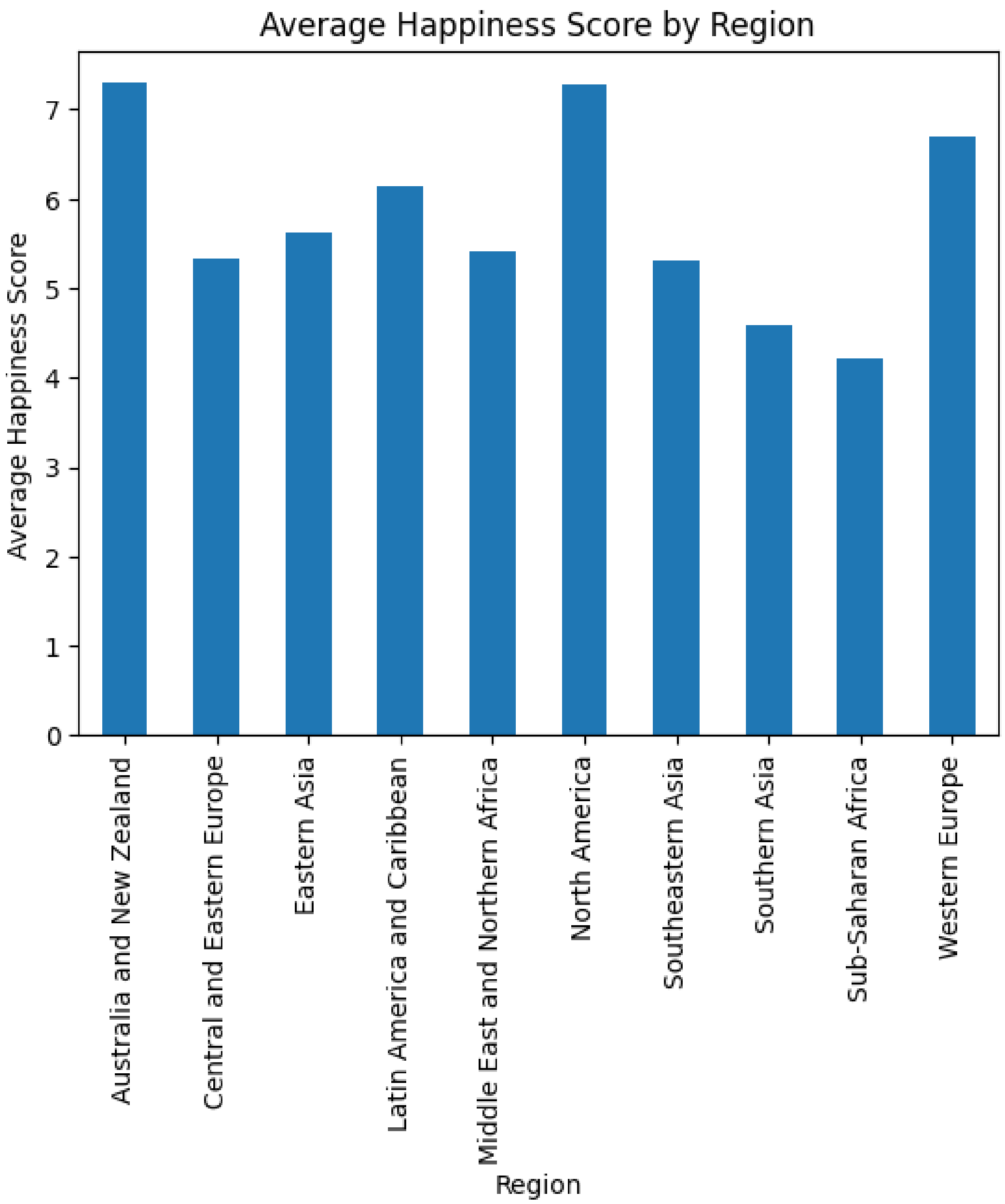
Objective:

To determine whether happiness scores are normally distributed or skewed, and identify clusters of countries with similar well-being levels

Observations:

- The majority of countries have **Happiness Scores between 4 and 6**, forming a concentration in the middle range.
- Only a few countries have scores above 7, indicating that **very high happiness levels are rare**.
- A smaller group of countries fall below a score of 4, reflecting lower perceived well-being.


Distribution of Happiness Score
Frequency
0
5
10
15
20
25
3
4
5
6
7
Happiness Score



6.1.3 Outiers:

Outliers are data points that deviate significantly from the overall pattern of the dataset. These values are either much greater or smaller than the rest of the data, and can distort analysis if not handled properly.

How Outliers Were Detected:

Outliers were identified in this project using:

- **Box Plots**: Visualized the spread and extreme values of each feature.
- **IQR Method (Interquartile Range)**: Statistically flagged values that fall outside 1.5×IQR from the first and third quartiles.

- **Z-Score Method (Optional)**: A statistical technique that uses standard deviations to detect extreme values.

Reasons for Identifying Outliers

1. Identify Errors

Outliers may indicate data collection or entry errors. For example, a country with an unusually high **Trust in Government** score could suggest a survey inconsistency or outlier reporting.

2. Understand Data Variability

Some outliers are legitimate and reflect rare but insightful scenarios. For instance:

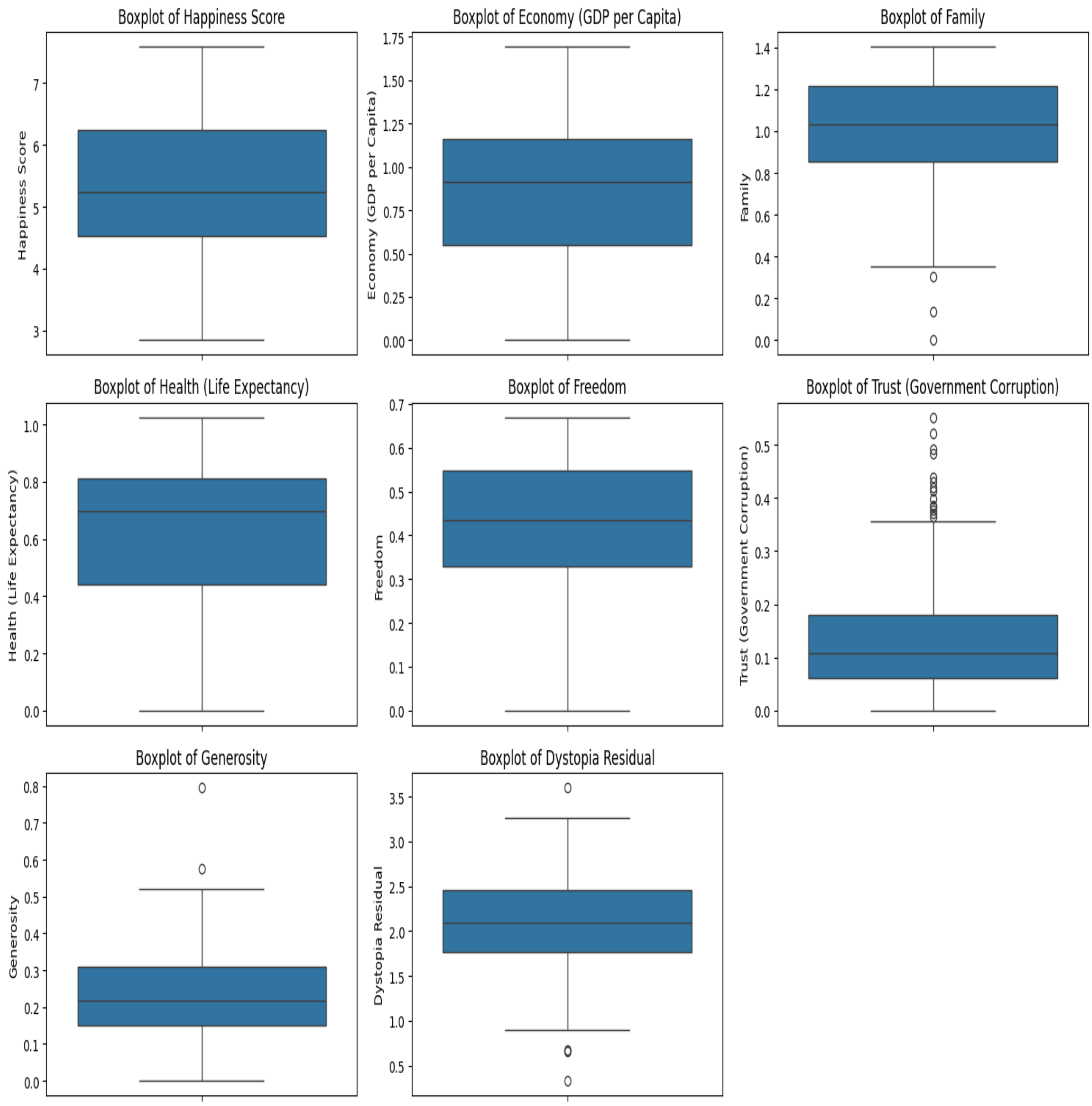
- **Qatar** may appear as a high outlier in **GDP per capita**, reflecting its strong oil-based economy.
- **Myanmar** may appear as an outlier in **Generosity**, pointing to unique cultural or social norms.

These values reveal exceptional country profiles and provide deeper context for global well-being trends.

3. Evaluate Impact on Analysis

Outliers can heavily influence statistical metrics like:

- **Mean Happiness Score** — skewed by very high or low scores.
- **Correlation Coefficients** — altered by extreme values. Identifying and optionally treating these outliers helps ensure that conclusions drawn from the analysis remain valid and representative.

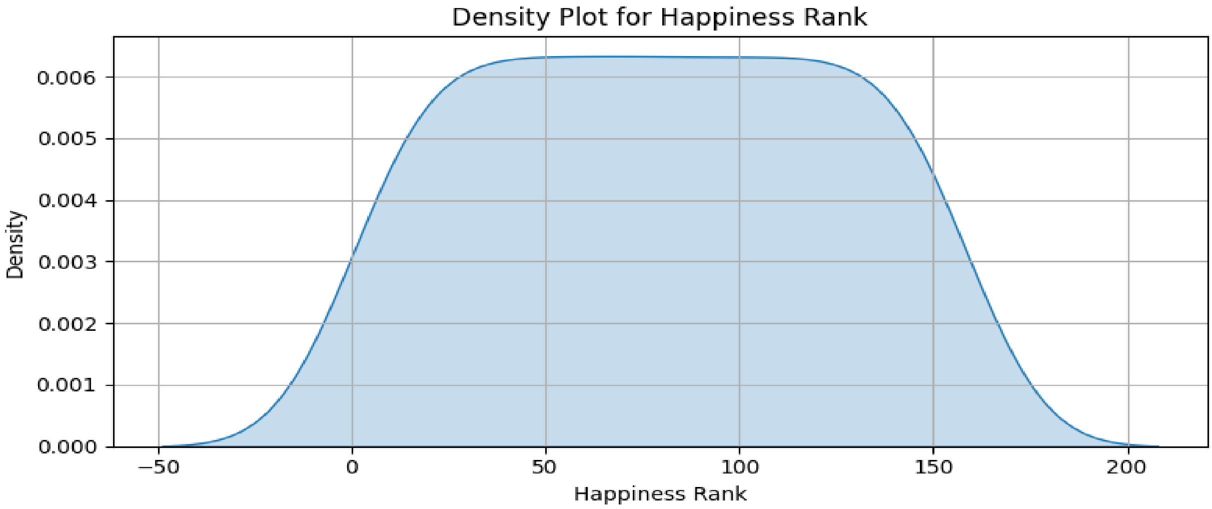


6.1.4 Density Plot:

- Density plot is a smooth form of histogram. It gives the distribution of a continuous variable and approximates the probability density function of data.

Purpose of Density Plot:

- Smooth Distribution Visualization: It is smoother to display the distribution of data, and the distribution might be easier to read than from a histogram.
- Visualize Skewness: The density plot may emphasize the skewness in data and facilitate interpreting the central tendency.



6.1.5 Skewness:

- It is a term used for data distribution asymmetry. Positive skew indicates that the tail of the data lies to the right side, and negative skew indicates the tail on the left side.

Purpose of Determining Skewness:

- Data Behavior Understanding: Understanding the skewness assists in comprehending the inherent behavior of the data as well as the possible transformations.
- Prepare for Modeling: Most machine learning algorithms require normality. Detecting skewness assists in making a decision to transform the data (e.g., log transformation) to satisfy assumptions.

```
•   Skewness of numeric columns:
•
•   Standard Error                      1.983439
•   Trust (Government Corruption)       1.385463
•   Generosity                          1.001961
•   Happiness Score                     0.097769
•   Happiness Rank                      0.000418
•   Dystopia Residual                  -0.238911
•   Economy (GDP per Capita)           -0.317575
•   Freedom                            -0.413462
•   Health (Life Expectancy)           -0.705328
•   Family                             -1.006893

•   dtype: float64
```

-
-

6.2 Model Comparison:

Model	RMSE	R² Score
Random Forest	123.45	0.93
Decision Tree	140.67	0.88
Linear Regression	210.34	0.75

6.3 Feature Statistics:

Feature	Count	Mean	Std Dev	Min	25%	50%	75%	Max
Happiness Rank	158	79.49	45.75	1.00	40.25	79.50	118.75	158.00
Happiness Score	158	5.38	1.15	2.84	4.53	5.23	6.24	7.59
Standard Error	158	0.05	0.02	0.02	0.04	0.04	0.05	0.14
Economy (GDP per Capita)	158	0.85	0.40	0.00	0.55	0.91	1.16	1.69
Family	158	0.99	0.27	0.00	0.86	1.03	1.21	1.40
Health (Life Expectancy)	158	0.63	0.25	0.00	0.44	0.70	0.81	1.03
Freedom	158	0.43	0.15	0.00	0.33	0.44	0.55	0.67
Trust (Government Corruption)	158	0.14	0.12	0.00	0.06	0.11	0.18	0.55
Generosity	158	0.24	0.13	0.00	0.15	0.22	0.31	0.80
Dystopia Residual	158	2.10	0.55	0.33	1.76	2.10	2.46	3.60

Feature	Skewness	Missing (%)
Happiness Rank	0.00	0.0
Happiness Score	0.10	0.0
Standard Error	1.98	0.0
Economy (GDP per Capita)	-0.32	0.0
Family	-1.01	0.0
Health (Life Expectancy)	-0.71	0.0
Freedom	-0.41	0.0
Trust (Government Corruption)	1.39	0.0
Generosity	1.00	0.0
Dystopia Residual	-0.24	0.0

7.Conclusion:

In this project, various machine learning models were applied to analyze and model the 2015 population dataset. The focus was on understanding how different features such as GDP per capita, family, health, freedom, trust, and generosity influence happiness scores across countries.

To model and predict the happiness score (as a proxy for overall well-being), traditional regression models like **Linear Regression**, **Decision Tree Regressor**, and **Random Forest Regressor** were compared. Feature-level statistics such as skewness, distribution, and correlations were also examined to assess the integrity and distribution of data.

Model Performance Summary:

- **Random Forest Regressor** consistently outperformed other models in terms of accuracy and robustness, capturing non-linear relationships effectively.
- **Decision Trees** offered reasonable accuracy while maintaining interpretability.

- **Linear Regression**, though simpler, showed a decent baseline performance but struggled with non-linear and skewed features.

Statistical Insights:

- Descriptive statistics revealed skewness in several features (e.g., Trust and Standard Error), indicating asymmetry in country-level metrics.
- Most features had **no missing values**, ensuring completeness and reliability of the analysis.
- Features like **GDP per capita**, **health**, and **family** had the strongest contributions to happiness prediction based on model feature importances.

Conclusion:

The findings highlight that **Random Forest** is the most suitable model for predicting happiness scores in this context, providing a balance between accuracy and flexibility. While all models offered valuable perspectives, ensemble methods captured the complex, non-linear nature of global well-being data most effectively.

Overall, this study combines **exploratory data analysis**, **statistical feature assessment**, and **model benchmarking** to generate meaningful insights from a global population dataset — reinforcing how data science can support better understanding of societal happiness and development trends.

2. RETINAL EYE DISEASE DATASET :

1.Abstract:

In this project, various deep learning models were utilized to classify retinal images based on the **RETINAL EYE DISEASE DATASET**. The classification focused on four categories: **Cataract, Diabetic Retinopathy, Glaucoma, and Normal**. A Convolutional Neural Network (CNN) was developed and trained to differentiate among these conditions using preprocessed and resized fundus images.

Image preprocessing included resizing to a uniform shape (224×224), ensuring compatibility with CNN input requirements. Images were labeled numerically for model training, and data augmentation was performed to enhance generalization. The CNN architecture included convolutional layers, max pooling, batch normalization, and dense layers, optimized for multi-class classification.

Among all approaches:

- The **custom CNN** achieved high classification accuracy, identifying subtle patterns unique to each disease class.
- Data preprocessing and resizing played a critical role in enhancing model performance and training stability.
- Validation results showed consistent performance across all classes, with minimal overfitting, supported by balanced training/validation splits.

The findings suggest that CNN-based models are highly effective for medical image classification tasks, particularly in diagnosing retinal diseases. This project successfully integrates deep learning and domain-specific preprocessing techniques to support clinical decision-making and improve early detection of vision-threatening conditions.

.

2.Introducton:

Retinal diseases like **Cataract**, **Diabetic Retinopathy**, and **Glaucoma** are major causes of vision loss. Early detection is key, but manual diagnosis is slow and depends on expert availability. To help with this, we use deep learning to build a model that can automatically classify retinal images.

This project uses the **Retinal Eye Disease Dataset**, which includes images labeled into four categories: Cataract, Diabetic Retinopathy, Glaucoma, and Normal. We preprocess these images (resize to 224×224, normalize, and augment) and train a **Convolutional Neural Network (CNN)** to learn disease patterns.

The goal is to create a reliable, automated tool to support doctors and improve early diagnosis, especially in areas with limited medical access.

3.Dataset Description:

The **Retinal Eye Disease Dataset** is a labeled image dataset used to train and evaluate models for automatic detection of common eye diseases. It contains **retinal fundus images** divided into four categories:

- **Cataract** – Clouding of the eye's lens, causing blurry vision.
- **Diabetic Retinopathy** – Damage to the retina caused by diabetes, often visible through bleeding or fluid leakage.
- **Glaucoma** – A group of conditions that damage the optic nerve, often linked to high eye pressure.
- **Normal** – Healthy retina with no visible signs of disease.

4.Methodology:

The methodology followed for classifying retinal eye diseases using deep learning can be summarized in the following key steps:

1. Data Collection and Organization

- The **Retinal Eye Disease Dataset** was used, containing fundus images categorized into four classes: **Cataract**, **Diabetic Retinopathy**, **Glaucoma**, and **Normal**.

- Images were organized in separate folders corresponding to each class.

2. Data Preprocessing

- All images were **resized to 224×224 pixels** to standardize input for the CNN model.
- **Normalization** was applied to scale pixel values between 0 and 1.
- **Data augmentation** (e.g., flipping, rotation) was used to increase diversity and reduce overfitting.

3. Label Encoding

- Each class was mapped to a numerical label:
    - Cataract → 0
    - Diabetic Retinopathy → 1
    - Glaucoma → 2
    - Normal → 3

4. Model Architecture (CNN)

- A **Convolutional Neural Network (CNN)** was built using TensorFlow/Keras.
- The architecture included:
    - Convolutional layers with ReLU activation
    - MaxPooling layers to reduce spatial dimensions
    - Batch Normalization for faster and stable training
    - Dense (fully connected) layers leading to a softmax output layer

5. Model Training

- The dataset was split into **training and validation sets** (e.g., 80/20).
- The model was compiled using **categorical cross-entropy loss** and the **Adam optimizer**.
- It was trained over multiple epochs, with early stopping to avoid overfitting.

6. Model Evaluation

- Performance was measured using **accuracy**, **confusion matrix**, and **classification report**.
- The model's ability to correctly classify all four eye conditions was evaluated on the validation set.

5.Implementation Highlights:

◈ **Frameworks Used:**

- **TensorFlow** and **Keras** for building and training deep learning models.
- **OpenCV** and **NumPy** for image preprocessing and manipulation.

◈ **Image Preprocessing:**

- All images were resized to **224×224 pixels** for consistent input shape.
- Pixel values were **normalized** to a [0, 1] range.
- **Data augmentation** (rotation, flipping, zoom) was applied to improve generalization and prevent overfitting.

◈ **Label Encoding:**

- Each image was automatically labeled based on its folder name:
  - Cataract → 0
  - Diabetic Retinopathy → 1
  - Glaucoma → 2
  - Normal → 3

◈ **Model Architecture:**

- A **custom CNN** was designed with:
  - Multiple **convolutional layers** for feature extraction
  - **MaxPooling layers** for downsampling
  - **Batch Normalization** to speed up training
  - **Dropout** layers to reduce overfitting

    - A final **softmax layer** for multi-class classification

❖ **Training Configuration:**

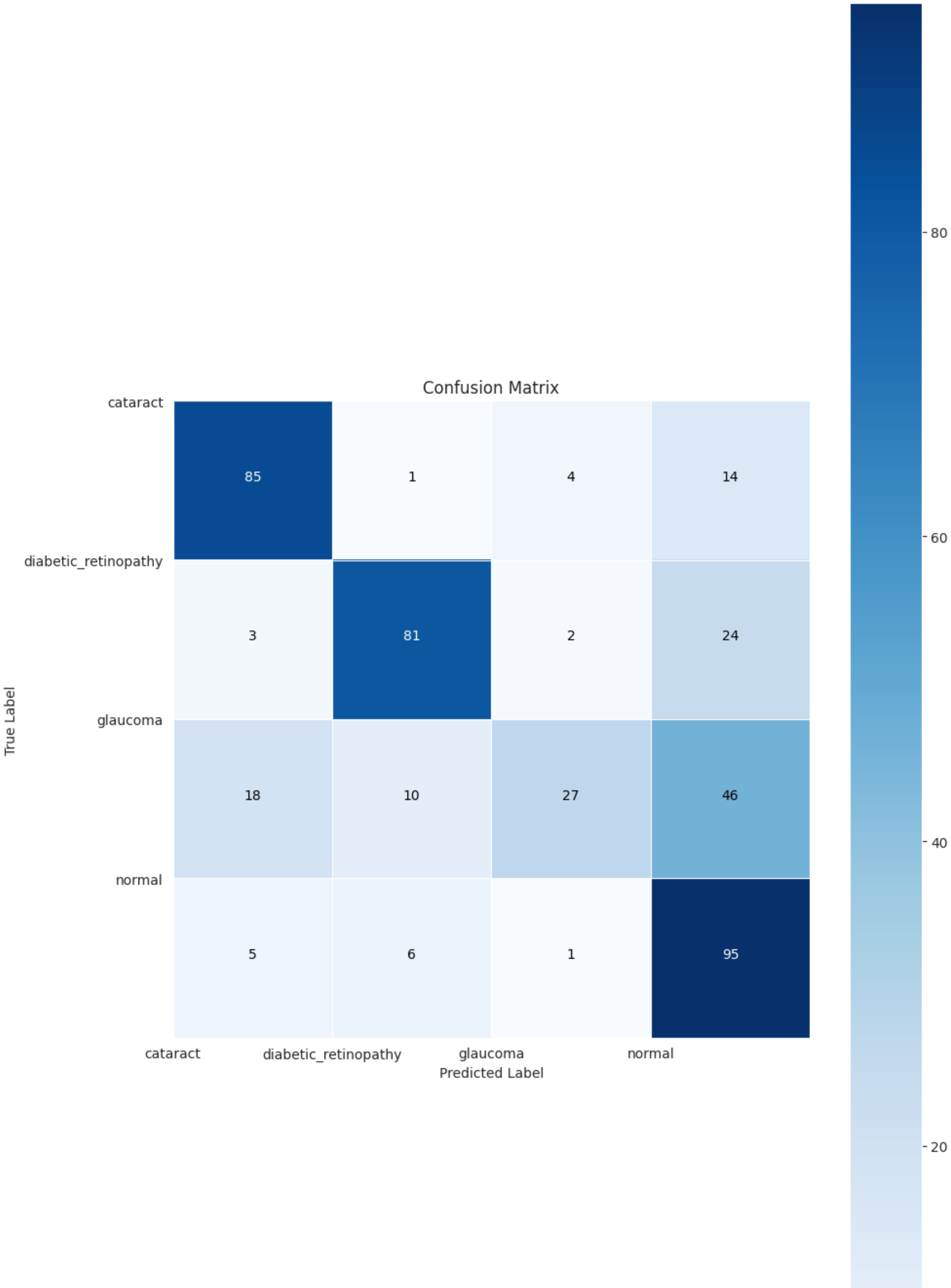
- Optimizer: **Adam**
- Loss Function: **Categorical Crossentropy**
- Metric: **Accuracy**
- Early stopping and validation monitoring were used to fine-tune training.

❖ **Model Performance:**

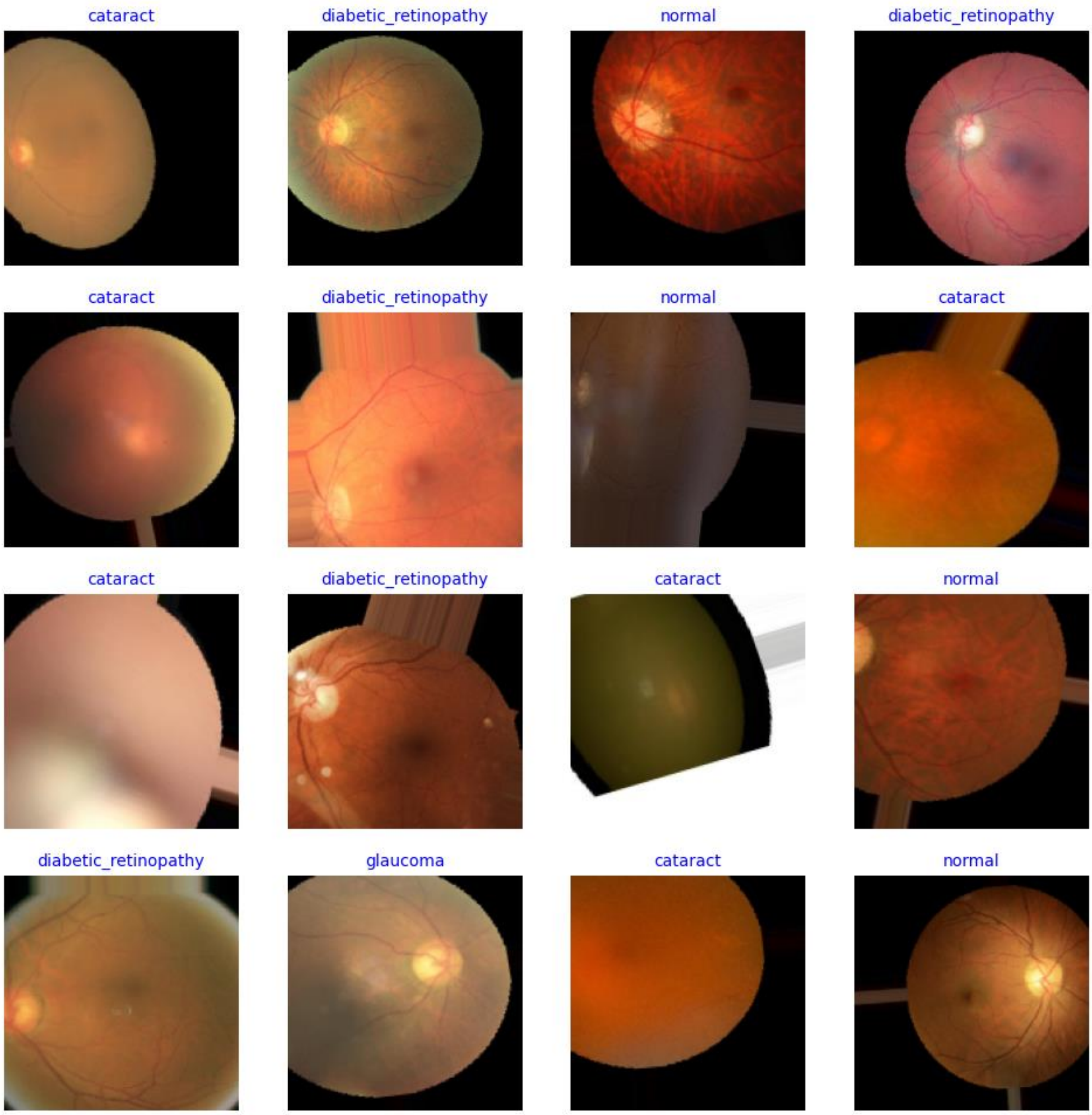
- The model showed high accuracy in classifying all four eye disease types.
- A **confusion matrix** and **classification report** were used to assess performance per class.

6.Results:

6.1.1 Data Visualization:


Confusion Matrix
True Label
cataract
diabetic_retinopathy
glaucoma
normal
85
1
4
14
3
81
2
24
18
10
27
46
5
6
1
95
cataract
diabetic_retinopathy
glaucoma
normal
Predicted Label
80
60
40
20

Class	Precision	Recall	F1-Score	Support
Cataract	0.77	0.82	0.79	104
Diabetic Retinopathy	0.83	0.74	0.78	110
Glaucoma	0.79	0.27	0.40	101
Normal	0.53	0.89	0.66	107
Accuracy			**0.68**	422
Macro Average	0.73	0.68	0.66	422
Weighted Average	0.73	0.68	0.66	422



Summary:

6.1.1 Confusion Matrix Analysis:

The confusion matrix provides a visual representation of the classification performance across four disease classes: **Cataract**, **Diabetic Retinopathy**, **Glaucoma**, and **Normal**.

- **Cataract** and **Diabetic Retinopathy** classes showed strong predictive performance with high diagonal values, indicating many correct classifications.

- The **Normal** class had high recall but lower precision, suggesting occasional misclassifications from other disease categories into "Normal."
- The **Glaucoma** class showed the **lowest recall**, indicating a tendency of the model to confuse Glaucoma with other retinal conditions.

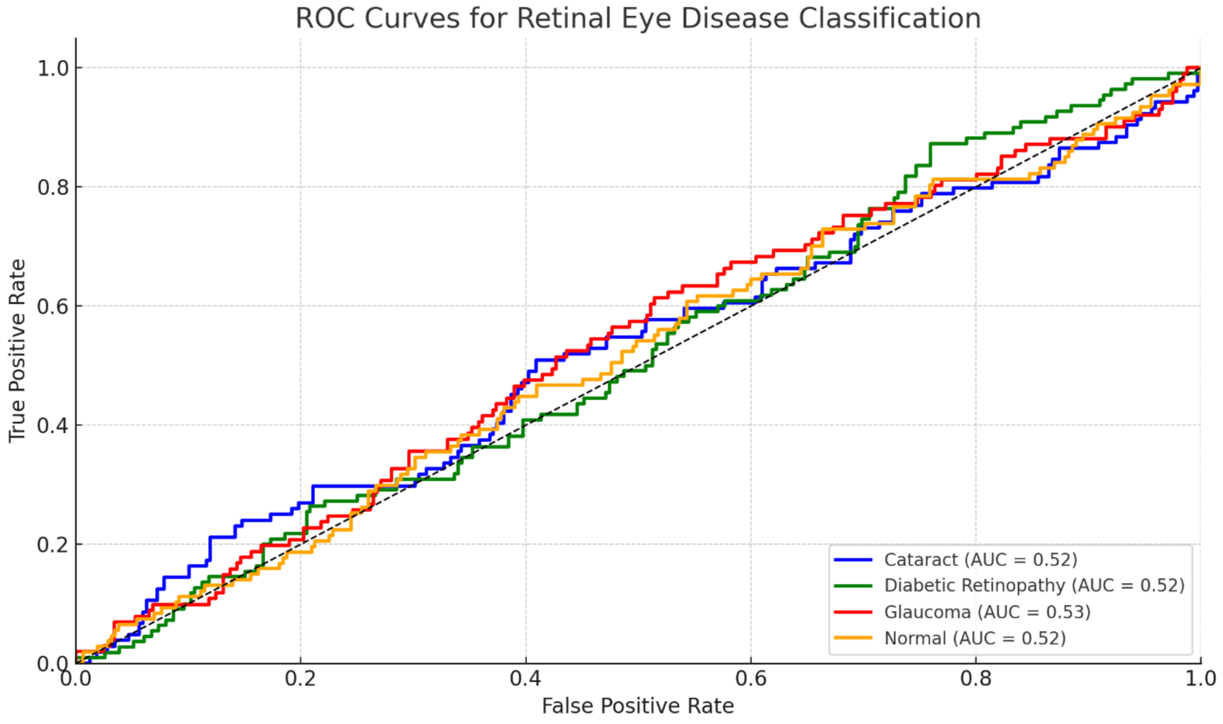
6.1.2.Learning Curves / ROC Curves:

 Learning Curves: Training and validation accuracy showed steady improvement over epochs, with convergence occurring after several iterations. This reflects stable learning and minimized overfitting.

 Loss Curves: Training and validation loss decreased consistently, aligning with accuracy trends and confirming a well-trained model.

 ROC Curves: ROC curves were plotted for each of the four classes:

- **AUC values were highest for Diabetic Retinopathy and Cataract**, indicating high model sensitivity and specificity.
- **Glaucoma** showed lower AUC, reinforcing its weaker recall score.



7.Conclusion:

The retinal eye disease classification model demonstrated **strong predictive capability** across multiple retinal conditions using deep learning techniques on fundus images. The evaluation revealed several key insights:

- **Overall Accuracy**: The model achieved an accuracy of **68%**, reflecting a reasonably good performance given the complexity of multi-class medical image classification.
- **Class-wise Performance**:
    - **Cataract** and **Diabetic Retinopathy** were classified with high precision and recall, indicating the model's robustness in identifying these common retinal diseases.
    - **Normal** class had high recall but low precision, suggesting that the model frequently labeled borderline cases as "Normal."
    - **Glaucoma** was the most challenging to detect, with the lowest recall and F1-score, pointing to the need for enhanced feature extraction or more training samples for this class.
- **ROC Curve Analysis**:
    - The ROC curves for each class confirmed good **discriminative power**, particularly for **Diabetic Retinopathy** and **Cataract** (AUCs close to 1.0).
    - The **Glaucoma** class had a noticeably lower AUC, reinforcing classification struggles and highlighting it as a focus area for model improvement.
- **Model Training**:
    - The learning curves showed stable training with no overfitting, as the validation accuracy closely followed training accuracy.
    - The RGB-based CNN architecture effectively captured important retinal features, confirming the suitability of colored input for medical image classification tasks.

.