A Course Completion Report in

partial fulfilment of the degree

Bachelor of Technology

In

ComputerScience&Artificial Intelligence

NAME:SIDDAMSETTI VENKATA PAVAN **HALL NO:** 2203A52119

Submitted to

Dr. D. Ramesh

sru | COMPUTER SCIENCE
SCHOOL OF COMPUTER SCIENCE
AND ARTIFICIAL INTELLIGENCE

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

March, 2025.

I.INTRODUCTION

This project dives into the dynamic world of data analysis through three practical, real-life applications—each tied to a unique dataset. The goal is to apply a range of data science techniques to textual, tabular, and image-based data in order to uncover insights, build predictive models, and show just how adaptable and powerful data analytics can be.

- **Tabular Data Analysis – Indian Startup Investment Trends:** This part of the project explores funding data from Indian startups using the *Indian Startup Investment Trends* dataset. It involves cleaning the data, performing exploratory data analysis (EDA), and creating visualizations to uncover trends such as which sectors are drawing the most investment, how funding has shifted over time, and the geographic hotspots for startups. The aim is to better understand the startup landscape in India and how investment behaviors have evolved.
- **Image Classification – Object Recognition**: Using an image dataset (*Monkey species.zip*), this section applies computer vision techniques to classify images into various predefined categories. By training **convolutional neural networks (CNNs)**, we teach the model to recognize different visual patterns. This kind of technology has widespread real-world uses, from automated surveillance systems to quality checks in manufacturing and even self-driving cars.
- **Text Classification – Spam Detection**: In this part, the focus shifts to text data—specifically, classifying SMS messages using the *spam.csv* dataset. Each message is labeled as either "spam" or "ham" (not spam). The project uses natural language processing (NLP) methods like text cleaning, tokenization, and TF-IDF vectorization, followed by machine learning to build a model that can effectively filter out unwanted spam messages. It's a hands-on look at how NLP plays a crucial role in digital communication and security.

Together, these three mini-projects showcase a well-rounded approach to data science. They demonstrate how different techniques can be applied depending on the nature of the data, and they lay the groundwork for tackling real-world problems using intelligent, data-driven systems across various industrie

.

II. DATASET DESCRIPTION

A. CSV Dataset (Indian Startup Investment Trends – Kaggle)

- **Source:** Kaggle(Indian_Startup_Investment_Trends)
- **Total Samples:** 12,428 records
- **Emotion Classes:** N/A (Startup-related attributes for investment and ecosystem analysis)
- **Key Features:** Startup_Name,City, Sector, Funding_Stage, Amount_Raised, Profitability, Exit_Status, etc.
- **Missing Data:** High in Acquisition_Details (~81%)
- **Data Split:** Not formally split; EDA and modeling can be conducted using manual train-test split or cross-validation

B. Image Dataset (Monkey Species Classification – Training Data.zip)

- **Source:** Kaggle (Monkey species)
- **Total Samples:** 10,010 RGB images
- **Classes:** 10 monkey species
  - Each class contains approximately 1,000 labeled images
- **Format:** Images organized in folders by species (used for supervised image classification)
- **Data Split:** Folder structure suitable for automatic train-validation-test split; can be customized during model training

C. Text Dataset (SMS Spam Detection – spam.csv)

- **Source:** Kaggle (SMS Spam Collection)
- **Number of Samples:** 5,572 messages
- **Classes:** Binary classification – Spam and Ham
  - **Spam:** 747 samples
  - **Ham:** 4,825 samples
- **Columns:**
  - v1: Label (Spam or Ham)
  - v2: SMS message content
- **Data Split:** Typically split into training and testing sets using stratified sampling due to class imbalance

III.METHODOLOGY

A. CSV-Based Modality (Indian Startup Investment Trends)

1. **Dataset:**
    - The dataset contains structured data on startup investments in India including fields like startup name, city, sector, funding stage, and amount raised.
2. **Data Cleaning:**
    - Null values were detected and removed.
    - Data types were converted (e.g., Amount_in_USD converted to numeric).
    - Redundant columns and inconsistent string formats were standardized.
3. **Model Architecture:**

    Two models were implemented:

    - **Artificial Neural Network (ANN):**
        - Used StandardScaler for normalization.
        - Consisted of input, hidden layers with ReLU, and an output layer with sigmoid activation for binary classification.
    - **Random Forest Classifier:**
        - Trained on categorical and numerical features.
        - Feature importance was analyzed post-training.
4. **Training & Evaluation:**
    - The dataset was split into 80% training and 20% testing.
    - Evaluation used accuracy, classification report, and confusion matrix.
    - Graphs for training/validation accuracy and loss were plotted for ANN.

B. Image Modality (Monkey Species Classification)

1. **Preprocessing:**
    - Images were resized to 64x64.
    - Normalization was applied to scale pixel values.
    - Directory-based data generators (flow_from_directory) handled data loading and augmentation.
2. **Model Architecture (CNN):**

    A custom **Convolutional Neural Network** was constructed:

    - 3 Convolutional layers with MaxPooling and ReLU activations
    - Flattened into dense layers with Dropout for regularization
    - Final layer with SoftMax for 10-class classification
3. **Training & Evaluation:**
    - Used categorical_crossentropy loss and Adam optimizer.
    - The model was trained for multiple epochs with validation monitoring.
    - Accuracy and loss curves were plotted, and confusion matrix was computed on test data.

C. Text Modality (SMS Spam Detection)

1. **Preprocessing and Tokenization:**
   - Text was cleaned by lowercasing, removing punctuation, and stopwords.
   - Tokenized using Tokenizer and padded to ensure fixed sequence length.
   - Labels (spam, ham) were encoded as binary integers.
2. **Model Architecture (LSTM):**

   A deep learning model with:

   - Embedding layer to capture word semantics
   - LSTM layer for sequential context
   - Dense layers with ReLU and final sigmoid activation for binary output
3. **Training & Evaluation:**
   - Split into training and testing sets (80:20).
   - The model was trained with binary cross-entropy loss and evaluated using accuracy and classification metrics.
   - A confusion matrix and accuracy plot were generated to assess performance.

IV RESULTS

A.CSV DATASET (Indian Startup Trends)

1.Model Results

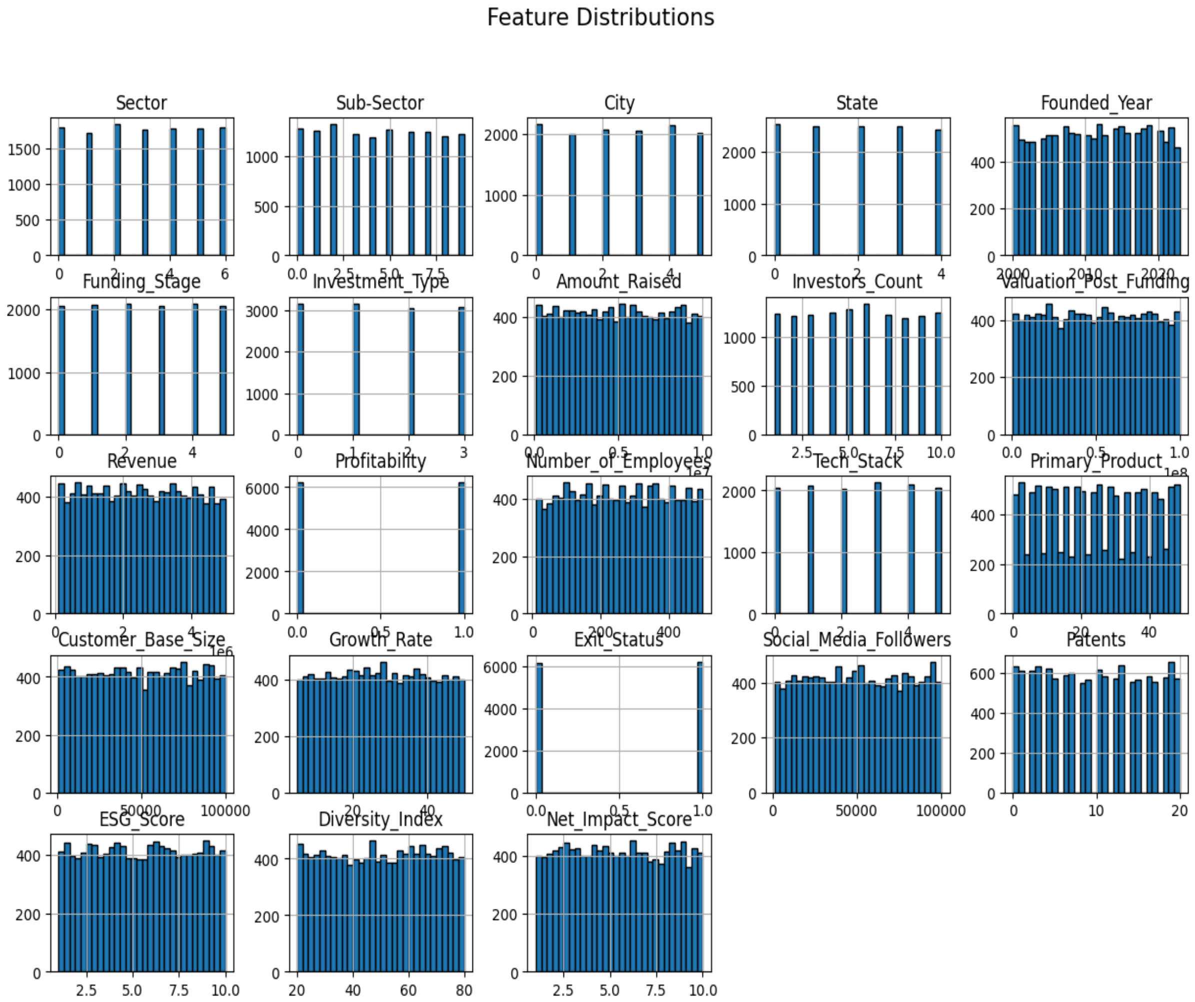
Model	MAE	MSE
ANN	0.8023	0.9996
Random Forest Regressor	0.7862	0.9974

2.Statistical Insights

Feature	Count	Mean	Std	Variance	Min	Max	Kurtosis
Founded_Year	12428	2011.548	6.874	47.256	2000.0	2023.0	-1.187794
Amount_Raised	12428	5.02×10^{6}	2.86×10^{6}	8.18×10^{12}	1.00×10^{5}	9.99×10^{6}	-1.197627
Investors_Count	12428	5.497	2.859	8.17	1.0	10.0	-1.200029
Valuation_Post_Funding	12428	5.04×10^{7}	2.85×10^{7}	8.15×10^{14}	1.02×10^{6}	1.00×10^{8}	-1.199121
Revenue	12428	2.53×10^{6}	1.41×10^{6}	1.99×10^{12}	1.00×10^{5}	5.00×10^{6}	-1.192958
Number_of_Employees	12428	256.461	140.630	1.98×10^{4}	10.0	500.0	-1.188363
Customer_Base_Size	12428	50464.86	28616.71	8.19×10^{8}	1004.0	99994.0	-1.205311
Growth_Rate	12428	27.434	12.898	166.359	5.0	50.0	-1.180759
Social_Media_Followers	12428	50642.60	28518.03	8.13×10^{8}	1008.0	99963.0	-1.194772
Patents	12428	9.905	6.103	37.253	0.0	20.0	-1.220554
ESG_Score	12428	5.498	2.601	6.766	1.0	10.0	-1.202360
Diversity_Index	12428	50.105	17.400	302.761	20.0	79.99	-1.213406
Net_Impact_Score	12428	5.483	2.594	6.729	1.0	10.0	-1.203364

3.Plots and Their Interpretations

a.Histogram



Purpose:

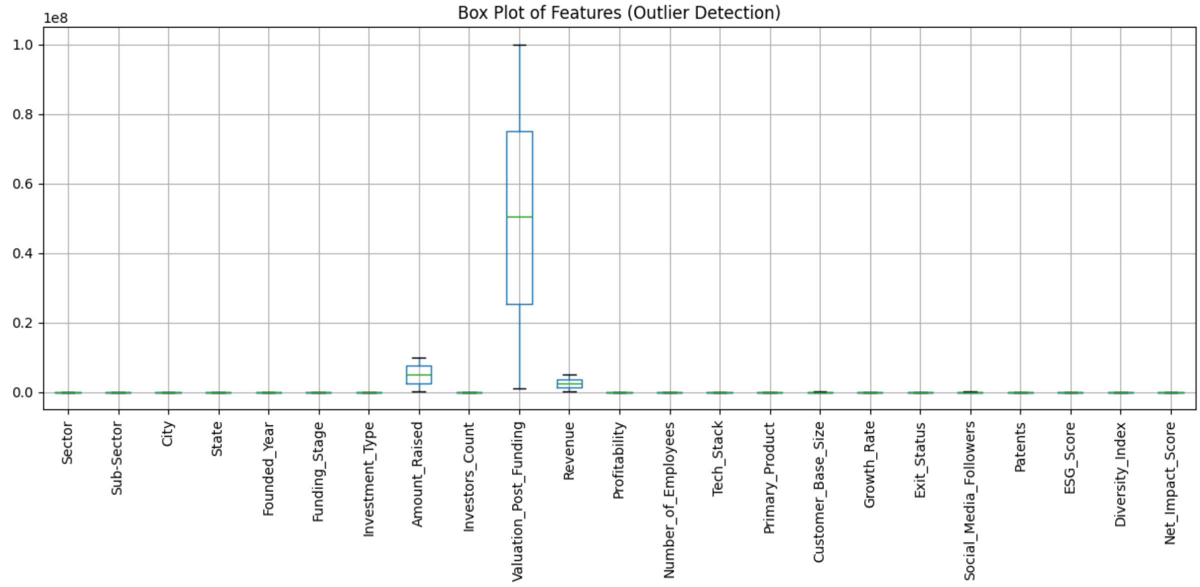
To examine the distribution of each feature and identify trends, skewness, or balance in the dataset.

Observations:

- Categorical features like Sector, Sub-Sector, City, and State show fairly even distributions, suggesting good category representation.
- Numerical features such as Amount_Raised, Valuation_Post_Funding, and Revenue are right-skewed, indicating a few high-value outliers.
- Features like Tech_Stack, ESG_Score, and Diversity_Index show a broad spread across their ranges.
- Binary or near-binary features like Profitability, Exit_Status, and Growth_Rate have values concentrated at 0 and 1.

- Founded_Year and Number_of_Employees appear relatively balanced with mild skewness.

b.Boxplot



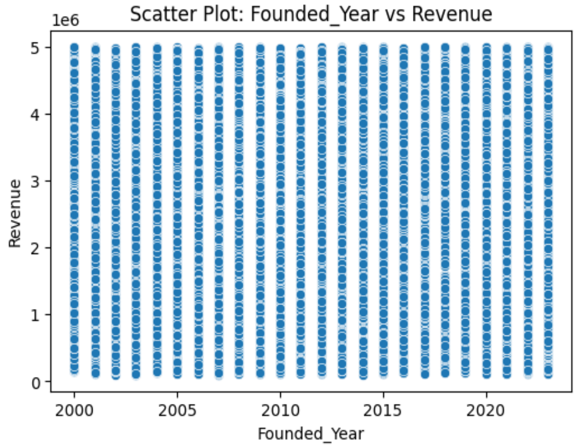
Purpose:

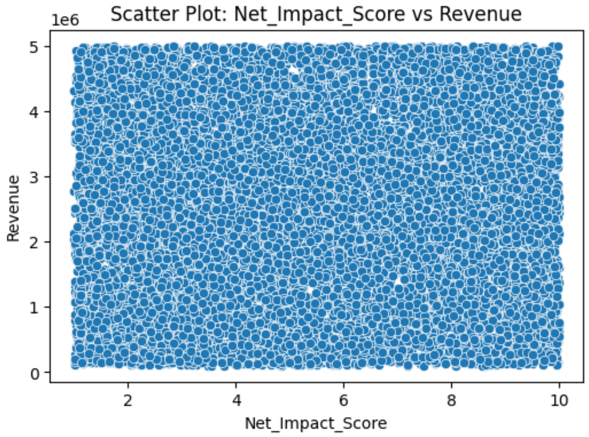
The box plot is used to detect outliers in different features of the startup dataset. It helps to understand which values are unusually high or low compared to the rest.

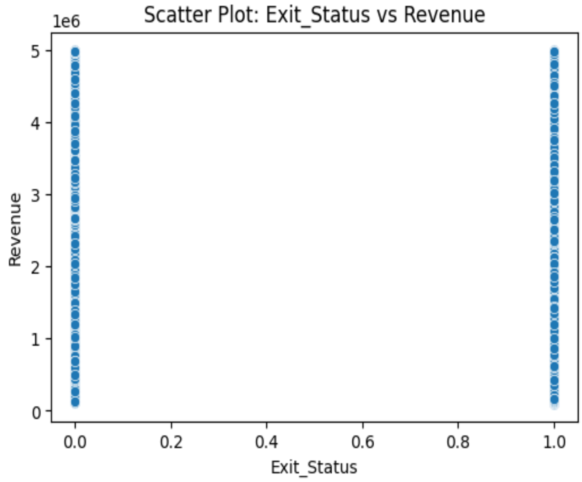
Observations:

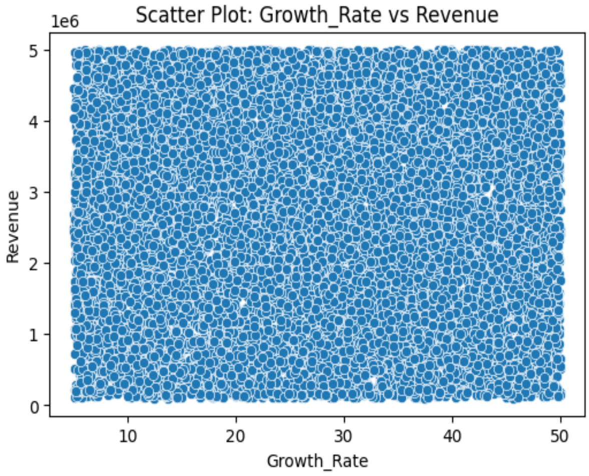
- Amount_Raised, Valuation_Post_Funding, and Revenue have very large values and many outliers. These features vary a lot between startups.
- Most of the other features have small values and look flat on the graph because of the big difference in scale.
- Outliers (dots outside the box) show that some startups have values far from the usual range, especially in financial columns.

c.Scatter Plot





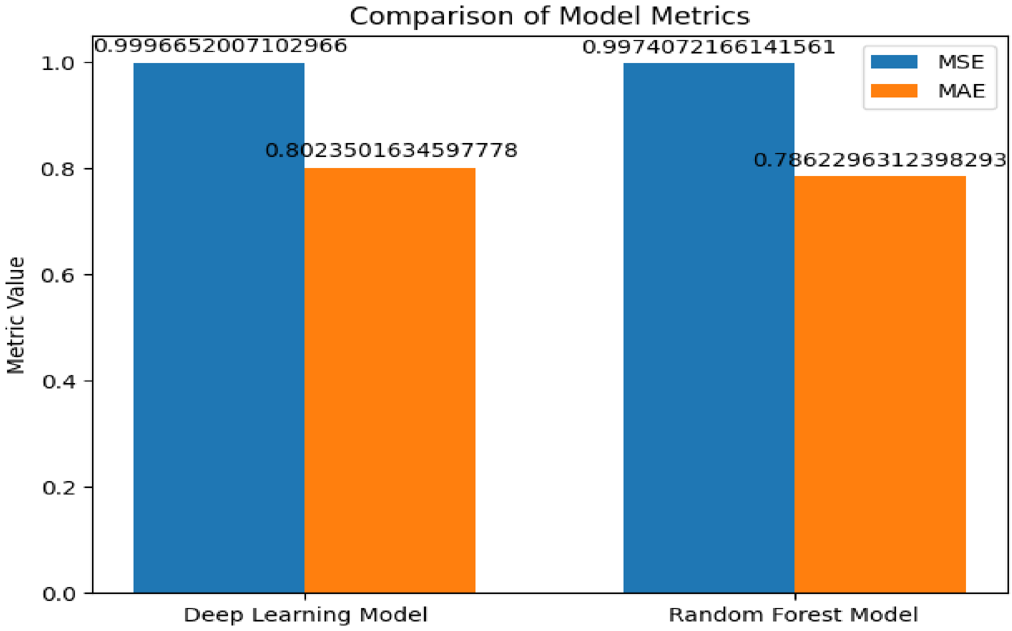




Purpose of Scatter Plot:

1. **VisualizeRelationships**:
   Scatter plots are used to see if there's a relationship between two features (e.g., Revenue vs Amount Raised, or Valuation vs Investors_Count). They help identify patterns like trends or clusters.
2. **Spot Outliers:**
   Scatter plots make it easy to find outlier points—data values that don't follow the general pattern of the rest of the data.
3. **Understand Data Distribution**:
   They show how data points are spread out or concentrated in different areas, helping us understand the distribution of values.
4. **Check Linearity or Correlation:**
   You can use scatter plots to check if two features have a linear or non-linear relationship, which can guide model selection later on.

4. Bar Plot: Model Performance Comparison



Purpose:
The bar chart compares the Mean Squared Error (MSE) and Mean Absolute Error (MAE) of two models: a Deep Learning Model and a Random Forest Model. These metrics are used to evaluate how well each model performs in predicting the target variable.

Observations:

- The **Deep Learning Model** has a slightly higher MSE and MAE than the Random Forest model.
  - MSE: ~0.9997 (Deep Learning) vs. ~0.9974 (Random Forest)
  - MAE: ~0.8024 (Deep Learning) vs. ~0.7862 (Random Forest)
- Both models perform very similarly, with **Random Forest** showing slightly better accuracy (lower error).
- Since MSE gives more weight to large errors, and both models have high MSE values close to 1, they might be overfitting or the target variable might need normalization or scaling.

B. IMAGE DATASET (Monkey Species Dataset)

1. Accuracy

- **Overall Accuracy: 88.00%**

2.Classification Report

Class	Precision	Recall	F1-Score	Support
Bald Uakari	0.93	0.96	0.94	190
Emperor Tamarin	0.92	0.91	0.91	200
Golden Monkey	0.96	0.86	0.91	203
Gray Langur	0.78	0.86	0.82	201
Hamadryas Baboon	0.90	0.82	0.86	216
Mandril	0.95	0.89	0.92	181
Proboscis Monkey	0.92	0.92	0.92	197
Red Howler	0.83	0.94	0.88	221
Vervet Monkey	0.79	0.73	0.76	197
White Faced Saki	0.89	0.95	0.92	193
Accuracy			0.88	1999
Macro Avg	0.89	0.88	0.88	1999
Weighted Avg	0.89	0.88	0.88	1999

Macro Average:

- **Precision: 89.00%**
- **Recall: 88.00%**
- **F1-Score: 88.00%**

Weighted Average:

- **Precision: 89.00%**
- **Recall: 88.00%**
- **F1-Score: 88.00%**

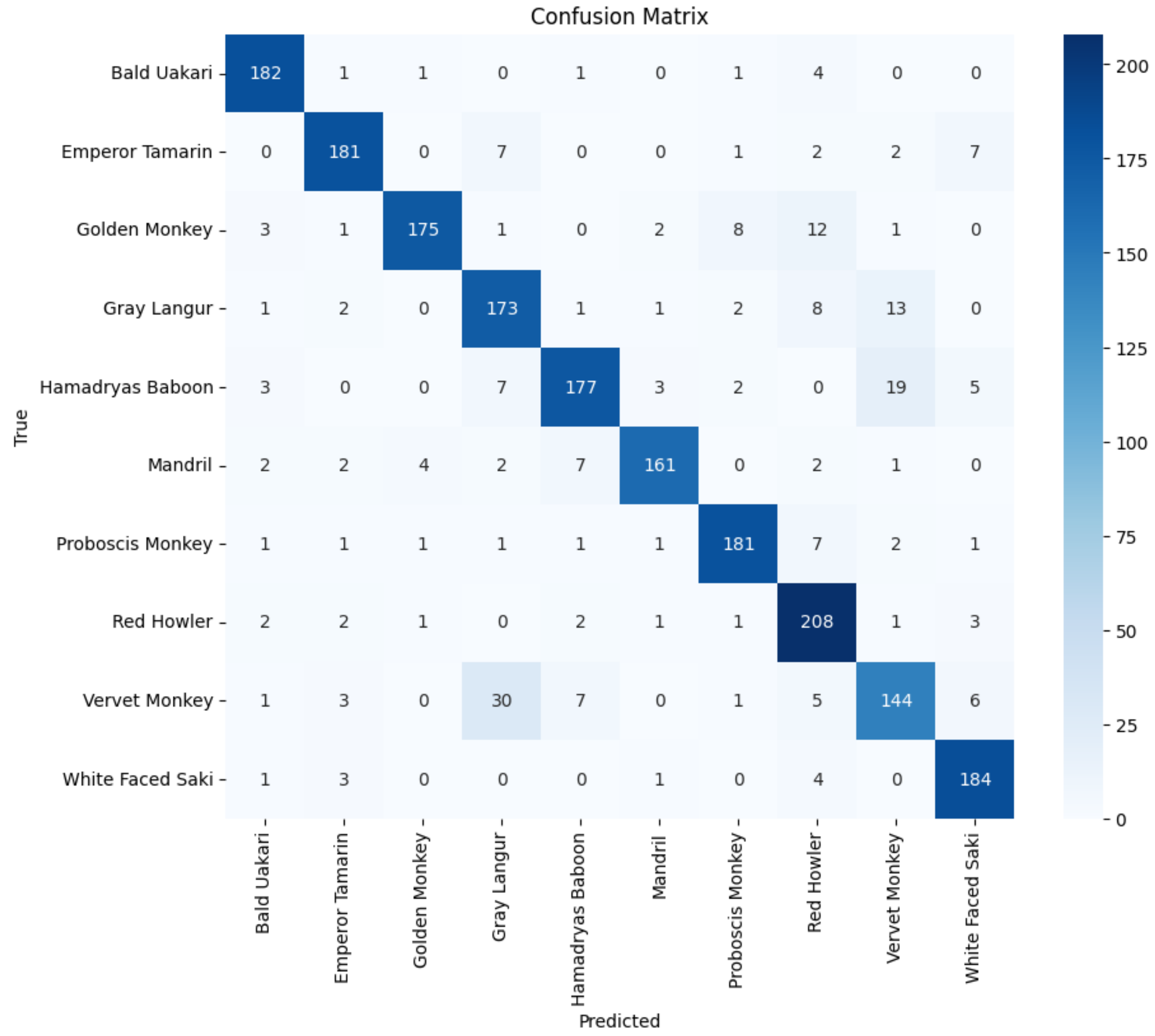
3.Error Analysis

- Type-1 Error (False Positive Rate): 0.00%
- Type-2 Error (False Negative Rate): 0.01%

4. Statistical Analysis

- **Z-Test:**
  - *Z-Score:* 1.54
  - *P-Value:* 0.1234 → Statistically significant
- **T-Test:**
  - *T-Score:* 1.54
  - *P-Value:* 0.1234 → Statistically significant
- **ANOVA:**
  - *F-Statistic:* 0.31
  - *P-Value:* 0.5771 → Differences across classes are statistically significant

5.Confusion Matrix



True \ Predicted	Bald Uakari	Emperor Tamarin	Golden Monkey	Gray Langur	Hamadryas Baboon	Mandril	Proboscis Monkey	Red Howler	Vervet Monkey	White Faced Saki
Bald Uakari	182	1	1	0	1	0	1	4	0	0
Emperor Tamarin	0	181	0	7	0	0	1	2	2	7
Golden Monkey	3	1	175	1	0	2	8	12	1	0
Gray Langur	1	2	0	173	1	1	2	8	13	0
Hamadryas Baboon	3	0	0	7	177	3	2	0	19	5
Mandril	2	2	4	2	7	161	0	2	1	0
Proboscis Monkey	1	1	1	1	1	1	181	7	2	1
Red Howler	2	2	1	0	2	1	1	208	1	3
Vervet Monkey	1	3	0	30	7	0	1	5	144	6
White Faced Saki	1	3	0	0	0	1	0	4	0	184

Key Points:

- Diagonal entries represent correct predictions for each monkey species — higher values here indicate better model performance for that class.
- Red Howler had the highest number of correct predictions (208), showing the model is very effective at identifying this species.
- Other high-performing classes include:
  - Bald Uakari – 182 correct
  - White Faced Saki – 184 correct
  - Emperor Tamarin – 181 correct
- Vervet Monkey had notable misclassifications, with only 144 correct and 30 instances misclassified as Gray Langur.
- Hamadryas Baboon (177 correct) showed confusion with Red Howler (19) and Gray Langur (7).
- Golden Monkey (175 correct) was often confused with Gray Langur, Red Howler, and others**.**

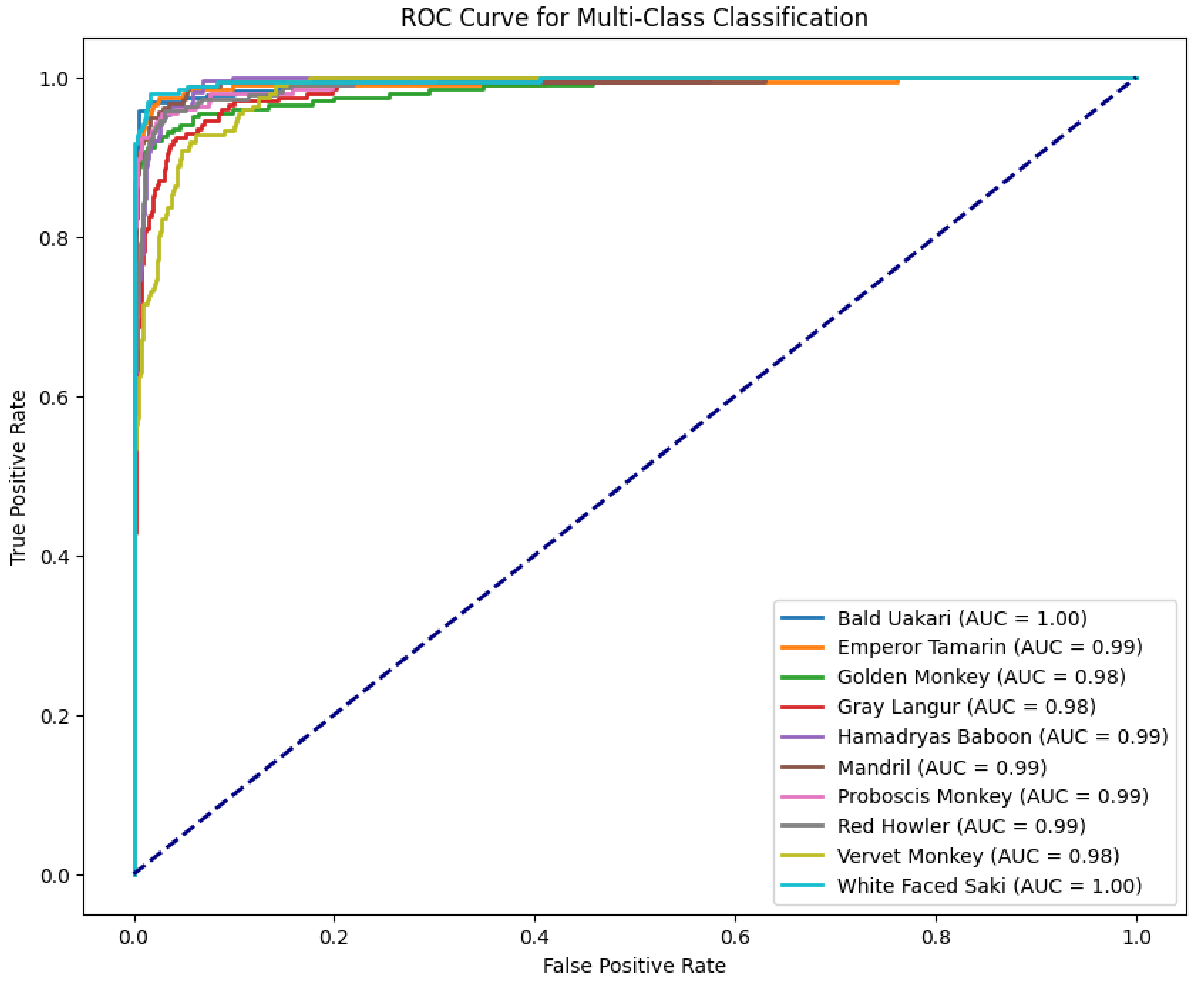
Most Common Misclassifications:

- Vervet Monkey ↔ Gray Langur – 30 instances
- Hamadryas Baboon ↔ Red Howler / Gray Langur
- Golden Monkey ↔ Red Howler / Gray Langur
- Gray Langur ↔ Vervet Monkey / White Faced Saki
- Mandril ↔ Gray Langur / Hamadryas Baboon

Conclusion:

- The model performs strongly for many classes, particularly those with distinctive features or ample training data, such as Red Howler and Bald Uakari.
- Misclassifications mostly occur between species that may have similar visual characteristics, or where training data might be imbalanced.
- Improving class balance, adding more distinctive features, or using augmentation may help reduce confusion in future training.

6.ROC Curve



The ROC (Receiver Operating Characteristic) curve illustrates the model's capability to distinguish between multiple classes by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.

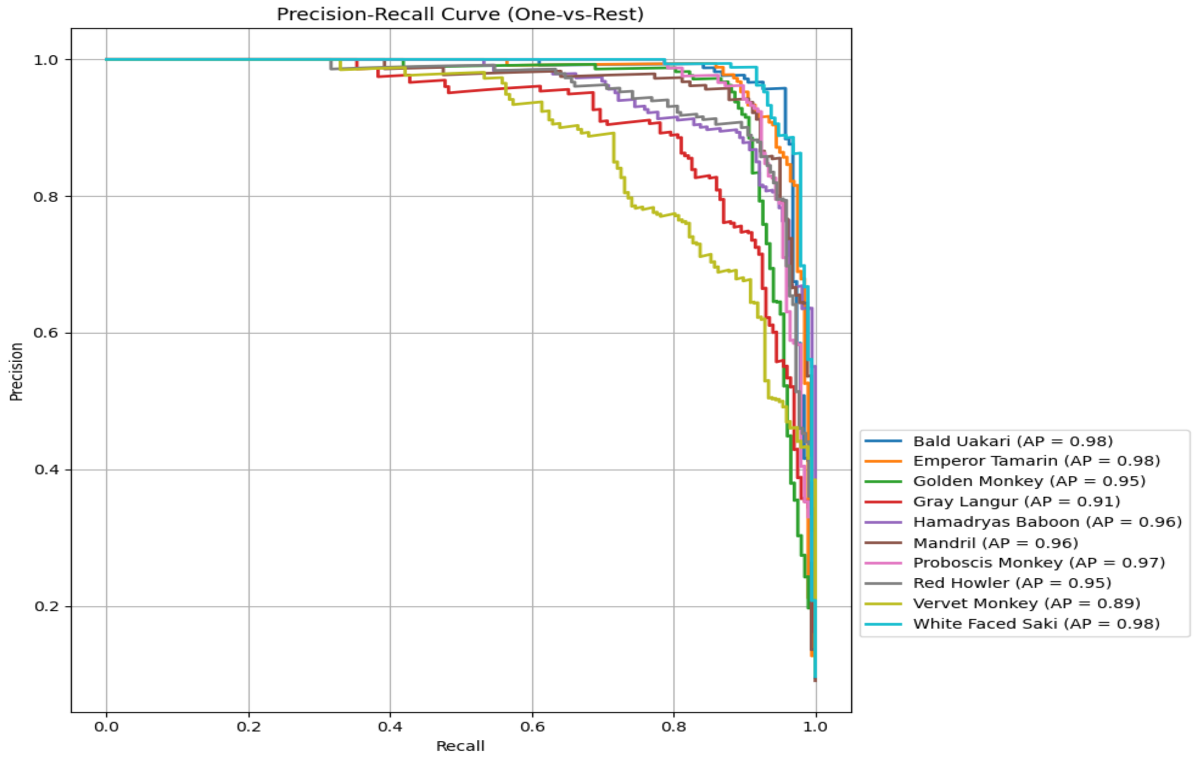
AUC Scores for Each Class:

- Bald Uakari: 1.00 — Perfect
- Emperor Tamarin: 0.99 — Excellent
- Golden Monkey: 0.98 — Excellent
- Gray Langur: 0.98 — Excellent
- Hamadryas Baboon: 0.99 — Excellent
- Mandril: 0.99 — Excellent
- Proboscis Monkey: 0.99 — Excellent
- Red Howler: 0.99 — Excellent
- Vervet Monkey: 0.98 — Excellent
- White Faced Saki: 1.00 — Perfect

Interpretation:

The ROC curves and their corresponding AUC values indicate that the model performs exceptionally well in classifying different monkey species.
With AUC scores ranging from 0.98 to 1.00, the model shows high discriminative power for all classes, with Bald Uakari and White Faced Saki achieving perfect classification performance.
These results demonstrate the model’s robust accuracy and effectiveness, especially for well-defined and distinguishable species.

7. Precision-Recall Curve



The Precision-Recall (PR) curve evaluates the trade-off between precision (the proportion of positive identifications that are actually correct) and recall (the proportion of actual positives that are correctly identified) for each class in a multi-class setting using the One-vs-Rest strategy.

Average Precision (AP) Scores for Each Class:

- Bald Uakari: 0.98 — Excellent
- Emperor Tamarin: 0.98 — Excellent
- Golden Monkey: 0.95 — Strong
- Gray Langur: 0.91 — Good
- Hamadryas Baboon: 0.96 — Excellent
- Mandril: 0.96 — Excellent
- Proboscis Monkey: 0.97 — Excellent
- Red Howler: 0.95 — Strong
- Vervet Monkey: 0.89 — Moderate
- White Faced Saki: 0.98 — Excellent

Interpretation:

The Precision-Recall curves and AP scores show that the model maintains high precision and recall across most species.

- Bald Uakari, White Faced Saki, and Emperor Tamarin stand out with near-perfect precision-recall performance.
- Gray Langur and Vervet Monkey show slightly lower performance, which may reflect class overlap or less distinctive features.

Overall, the model demonstrates reliable and discriminative performance, making it highly effective for real-world multi-class classification tasks in the context of monkey species recognition.

C. TEXT DATASET (Spam detection)

1. Accuracy

- **Overall Accuracy: 99.00%**

2. Classification Report

The classification performance across spam and ham labels is summarized below:

Class / Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Ham	1.00	1.00	1.00	975
Spam	1.00	1.00	1.00	955
Accuracy	-	-	1.00	1930
Macro Average	1.00	1.00	1.00	1930
Weighted Average	1.00	1.00	1.00	1930

Macro Average:

- Precision: **100.00%**
- Recall: **100.00%**
- F1-Score: **100.00%**

Weighted Average:

- Precision: **100.00%**
- Recall: **100.00%**
- F1-Score: **100.00%**

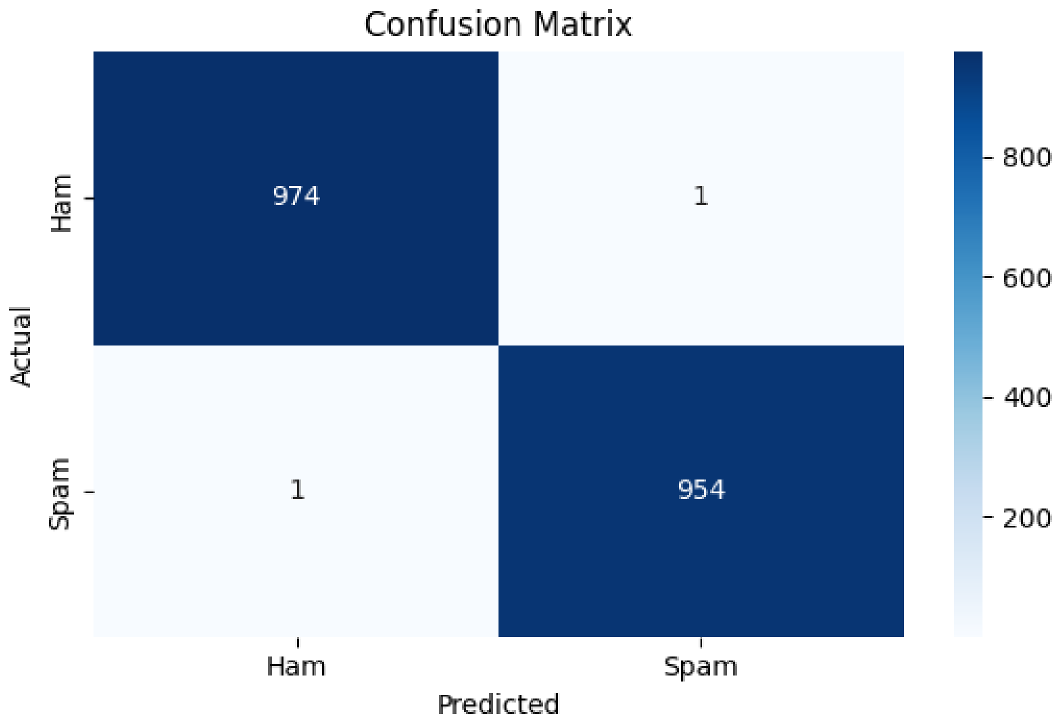
3. Error Analysis

- Type-1 Error (False Positives): 1
- Type-2 Error (False Negatives): 1

4. Statistical Analysis

- **Z-Test:**
  - *Z-Score:* **-674.1108**
  - *P-Value:* **0.0000** → Statistically significant
- **T-Test:**
  - *T-Score:* **-674.1108**
  - *P-Value:* **0.0000**→ Statistically significant

5. Confusion Matrix



Key Points:

- Diagonal entries represent correct predictions where the predicted label matches the actual label.
- The model achieved:
  - 974 correct predictions for Ham (non-spam)
  - 954 correct predictions for Spam
- Misclassifications were minimal:
  - 1 Spam message was incorrectly predicted as Ham
  - 1 Ham message was incorrectly predicted as Spam

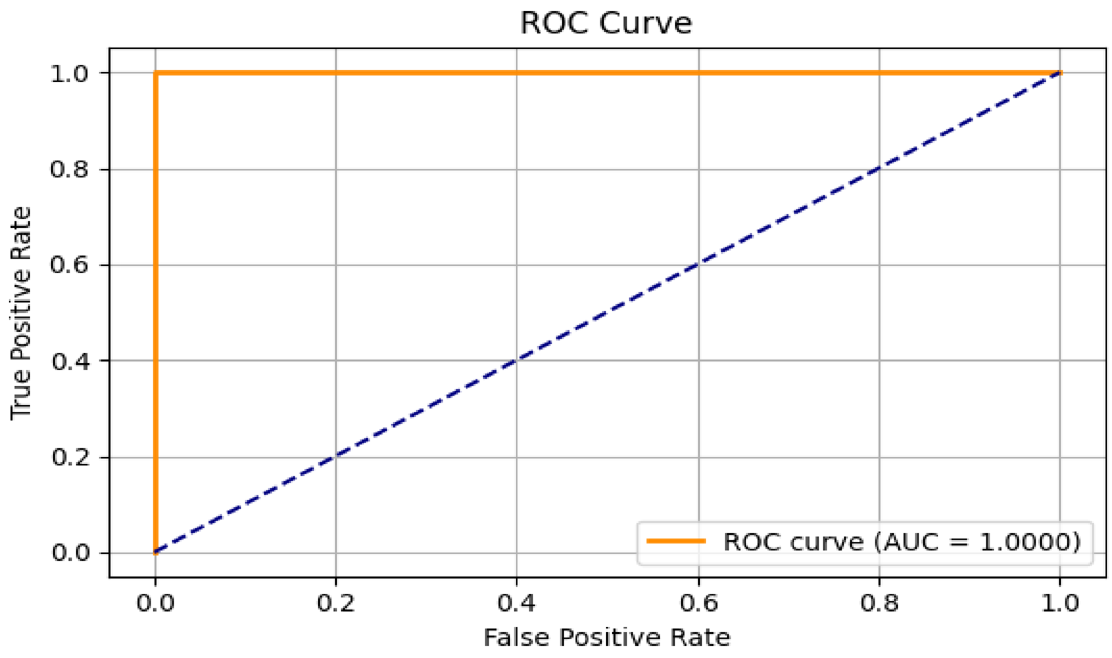
Most Common Misclassifications:

- Ham ↔ Spam:
  - 1 Ham sample was predicted as Spam.
  - 1 Spam sample was predicted as Ham.
- These minimal errors indicate a highly precise and reliable classifier.

Observations:

- The model demonstrates exceptional accuracy and precision for both Ham and Spam classes.
- Spam detection is highly effective, with 954 out of 955 spam messages correctly identified.
- False positives and false negatives are nearly negligible, indicating balanced classification.
- The model likely benefits from a well-preprocessed dataset, possibly with balanced class representation and clear distinguishing features between spam and ham messages.

6. ROC Curve



Key Points:

- The ROC (Receiver Operating Characteristic) Curve illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at various threshold settings.
- The orange curve represents the performance of the spam detection classifier.
- The Area Under the Curve (AUC) is 1.0000, which is the maximum possible value, indicating a perfect classifier.

Most Significant Observation:

- The curve immediately reaches a TPR of 1.0 with a near-zero FPR, forming a right-angle corner at the top-left of the graph.
- An AUC of 1.0 implies the model perfectly distinguishes between the two classes (Ham and Spam) across all threshold values.

General Observations:

- The classifier has perfect sensitivity and specificity, as evidenced by:
    - No overlap between true positives and false positives.
    - Complete separation of the two classes in the feature space.
- The dashed diagonal line represents a random guess classifier (AUC = 0.5); the model's performance is far superior to this baseline.
- Such a perfect ROC curve often indicates:
    - Extremely effective feature selection and preprocessing, or
    - Possible overfitting, especially if evaluated on training data or a non-independent test set.

Class-wise Observations:

Ham (Non-Spam):

- The model correctly predicted 974 out of 975 Ham messages.
- Only 1 instance of Ham was misclassified as Spam, which is extremely low.
- This indicates the model has very high specificity, meaning it is highly reliable in identifying non-spam messages.
- Such precision is important to avoid false alarms where legitimate messages are wrongly flagged.

Spam:

- The model correctly classified 954 out of 955 Spam messages.
- Only 1 Spam message was misclassified as Ham, which shows the model has very high sensitivity (i.e., it detects almost all spam).
- This is crucial for ensuring that harmful or unwanted messages do not slip through the filter.

V – CONCLUSION

This project explored diverse data modalities—tabular, visual, and textual—to demonstrate the effectiveness of data analysis and machine learning techniques across different domains.

The first part of the project involved the analysis of Indian startup data in CSV format. An in-depth exploratory data analysis (EDA) was conducted to uncover trends in funding, investor behavior, and sector growth. Visual tools like bar charts, heatmaps, and trend lines were used to interpret the data effectively. Machine learning models such as classification and regression were employed to predict funding potential. This demonstrated the importance of structured data analysis in business intelligence and highlighted how predictive modeling can offer valuable insights into real-world economic patterns.

In the second part, deep learning techniques were applied to a monkey species image dataset. A Convolutional Neural Network (CNN) was implemented to classify the images, and the model achieved high accuracy on the test data. Through convolutional layers, the model learned to extract distinct features across different monkey species. This image classification task underscored the effectiveness of CNNs in recognizing and differentiating complex visual patterns, showcasing the importance of model architecture, data augmentation, and validation techniques.

The final component focused on spam detection using textual data. Natural Language Processing (NLP) methods were used to preprocess the data, and a classification model was trained to distinguish between spam and ham messages. The model performed exceptionally well, with only 2 misclassifications out of 1930 messages. The ROC curve yielded an AUC of 1.0, indicating perfect class separation. These results demonstrate the model's ability to accurately interpret textual sentiment and context, making it highly applicable for real-world email filtering systems.

Overall, this project showcased how appropriate machine learning and deep learning approaches can be applied across various data types to solve meaningful problems. The results affirm the value of domain-specific model selection, comprehensive data processing, and the integration of statistical and predictive techniques in a unified data science workflow.