DATA ANALYSIS USING PYTHON PROJECT

“Unified Data Insights: Analyzing Multimodal Datasets with Python"

A Project Lab Report in Partial Fulfillment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

2203A52007 – B. SOUMYA

Submitted to

Dr. Ramesh Dadi

Assistant Professor, School of CS&AI.

TABLE OF CONTENT

1. INTRODUCTION

In today's data-driven world, the ability to analyse and extract insights from diverse types of data is a critical skill. This capstone project demonstrates an end-to-end application of data analysis and machine learning techniques using Python across three distinct types of datasets: tabular (CSV), image, and textual data. By working with heterogeneous data formats, the goal was to explore the preprocessing needs, model development strategies, and evaluation methods specific to each data type.

The datasets used in this project are:

- **NASA EXOPLANETS Dataset**: The NASA Exoplanet Archive is a database that contains information on all known exoplanets (planets outside our solar system) discovered by NASA's various space missions, ground-based observatories, and other sources. The dataset includes information such as the planet's name, mass, radius, distance from its host star, orbital period, and other physical characteristics. The dataset also includes information on the host star, such as its name, mass, and radius. The archive is updated regularly as new exoplanets are discovered, and it is a valuable resource for astronomers studying the properties and distribution of exoplanets in our galaxy.
- **Image Dataset**: A collection of images across multiple categories used for multi-class image classification.
- **English Word Difficulty Classification Dataset**: A text-based dataset aimed at classifying words based on their difficulty levels for undergraduate and postgraduate students.

This study explores the application of Python-based data science tools across three distinct datasets:

- **Textual Data**: Hindi Sentiment Dataset
- **Tabular Data**: NASA Exoplanets Dataset
- **Image Data**: Hands and Palm Images Dataset

Each dataset posed unique challenges and required domain-specific preprocessing and modelling techniques. This report details how data preprocessing, model selection, evaluation, and statistical analysis were tailored to the nature of each dataset to derive meaningful insights and optimize performance.

2. Objectives

The primary objectives of this capstone project are:

To explore and analyse multimodal datasets—text, image, and tabular—to gain a holistic understanding of data analysis across formats.

To perform relevant preprocessing on each dataset based on its nature, including cleaning, normalization, encoding, feature extraction, and transformation.

To build and evaluate predictive models suited to each dataset:

- For the **CSV dataset**, apply and compare traditional machine learning models and perform statistical tests (z-test, t-test, ANOVA) to support findings.
- For the **image dataset**, build a custom Convolutional Neural Network (CNN) to perform multi-class classification and analyse the performance using statistical evaluation.
- For the **text dataset**, convert words to embeddings using pre-trained models, train a Long Short-Term Memory (LSTM) network, and benchmark against traditional ML models.

The objective is to demonstrate how different data types can be processed, analyzed, and modeled using appropriate machine learning and deep learning techniques.

3. Overview of Datasets

Dataset	Type	Source	Key Features	Purpose in Project
NASA EXOPLANETS Data	Structured CSV (tabular)	NASA-EXOPLANETS	Structured data of discovered exoplanets	Used for scientific discovery modeling, clustering exoplanets, or building predictors for potential Earth-like planets.
Image Dataset	Image	Kaggle-Image-Dataset	11,076 hand images with age and gender labels. image JPEG image of hand or	□ Train deep learning models for **age prediction** and **gender classification** □ Useful in **biometrics**, **security**

Dataset	Type	Source	Key Features	Purpose in Project
			palm (1600 x 1200 pixels) age — Age of subject (18–75 years) gender — Gender (Male/Female)	**applications**, and **forensic analysis** □ Supports computer vision applications using **CNNs**, **transfer learning**
English Word Difficulty Classification	Text (NLP)	TRANSLATION-TEXT	`text` — Hindi sentence expressing an emotion `label` — Emotion label (7 classes): anger, disgust, fear, joy, sadness, surprise, neutral	□ To **classify Hindi text into emotion categories** □ Enables development of sentiment-aware applications for **social media monitoring**, **chatbots**, and **emotion-based analytics** in Hindi □ Ideal for testing NLP models (SVM, LSTM, BERT for Hindi)

4. Dataset wise Analysis

4.1 CSV Dataset: Tabular Data Analysis (COVID-19 Dataset)

4.1.1 Data Analysis

Dataset Overview:

□ **Source**: Kaggle

□ **Size**: 4,367 entries with 82 columns

□ **Columns**: A mix of numerical and categorical data, including:

- **Planetary Attributes**: Name, mass, radius, orbital period, eccentricity.

- **Stellar Attributes**: Host star's mass, radius, temperature, and distance.
- **Discovery Details**: Method of discovery, year, facility.

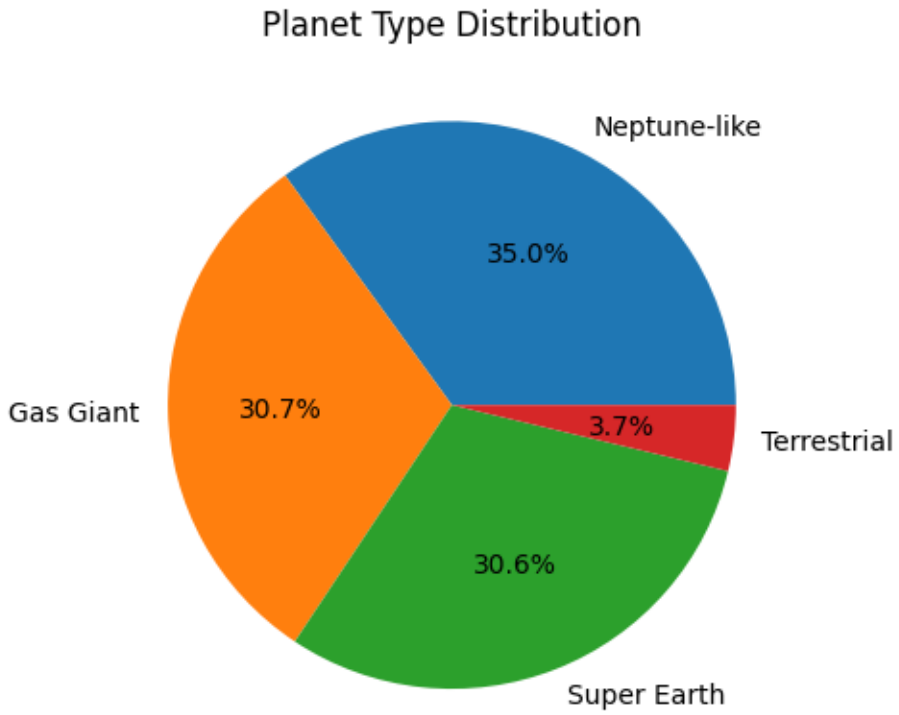
 Data Type: Structured tabular data with numerical and categorical variables.

```
1 df.head()
```

	name	distance	stellar_magnitude	planet_type	discovery_year	mass_multiplier	mass_wrt	radius_multiplier	radius_wrt	orbital_radius
0	11 Comae Berenices b	304.0	4.72307	Gas Giant	2007	19.40000	Jupiter	1.08	Jupiter	1.290000
1	11 Ursae Minoris b	409.0	5.01300	Gas Giant	2009	14.74000	Jupiter	1.09	Jupiter	1.530000
2	14 Andromedae b	246.0	5.23133	Gas Giant	2008	4.80000	Jupiter	1.15	Jupiter	0.830000
3	14 Herculis b	58.0	6.61935	Gas Giant	2002	8.13881	Jupiter	1.12	Jupiter	2.773069
4	16 Cygni B b	69.0	6.21500	Gas Giant	1996	1.78000	Jupiter	1.20	Jupiter	1.660000

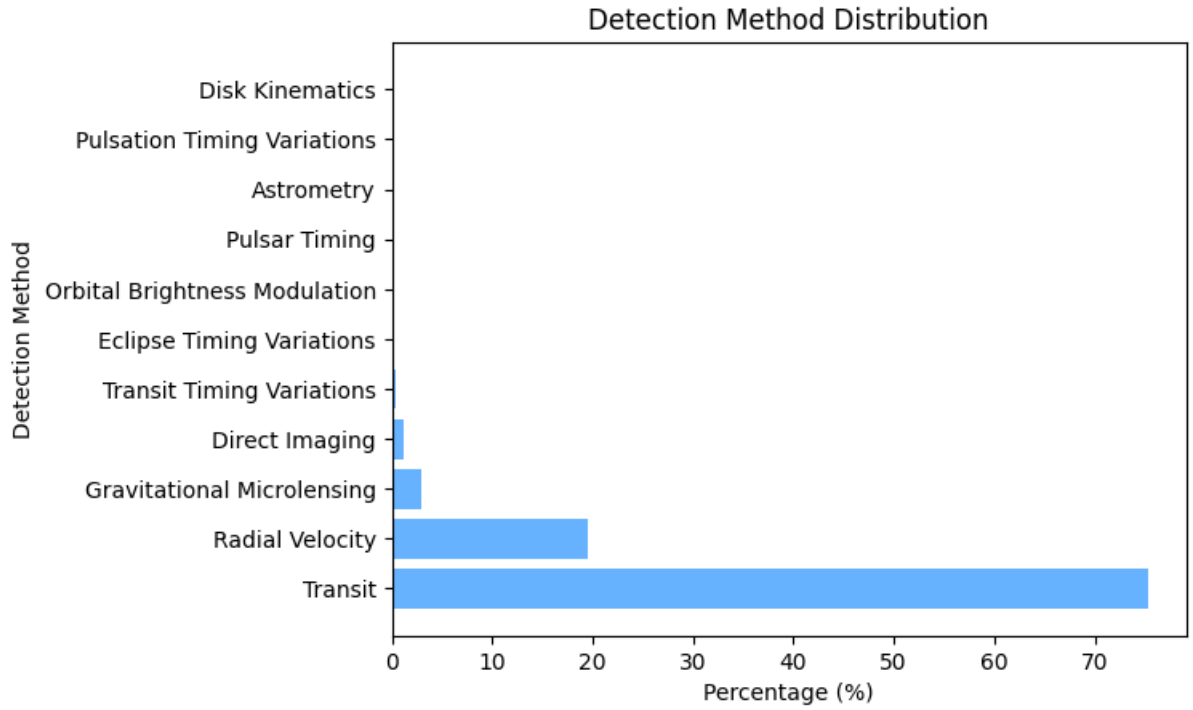
Data Cleaning Steps:

- Converted coded values to meaningful labels for interpretability (e.g., SEX: 1 → Male, 2 → Female, 97 → Unknown).
- Removed or imputed invalid codes (97, 98, etc.).
- Transformed DATE_DIED into a boolean “Survived” flag.
- Balanced the target variable (CLASIFICATION_FINAL) using SMOTE to address class imbalance.



Data Preprocessing

- Loaded data using pandas and checked for missing values.
- Imputed missing numeric values using median; dropped highly sparse columns.
- Scaled numerical columns using Min-Max scaling.
- Created derived features like **density**, **orbital ratio**.
- Converted categorical features (disc_method, disc_year) using one-hot encoding.



Skewness and Kurtosis

	name	distance	planet_type	discovery_year	mass_earth	radius_earth	orbital_radius	orbital_period	eccentricity	detection_method
5240	XO-2 S c	494.0	Gas Giant	2014	435.424881	13.562890	0.47560	0.330732	0.15	Radial Velocity
5241	XO-3 b	695.0	Gas Giant	2007	2316.968890	15.804690	0.04760	0.008761	0.29	Transit
5242	XO-4 b	889.0	Gas Giant	2008	451.316300	14.011250	0.05524	0.011225	0.00	Transit
5243	XO-5 b	901.0	Gas Giant	2008	378.215772	12.778260	0.05150	0.011499	0.00	Transit
5244	XO-6 b	768.0	Gas Giant	2016	1398.444872	23.202630	0.08150	0.010404	0.00	Transit
5245	XO-7 b	764.0	Gas Giant	2019	225.340321	15.389957	0.04421	0.007940	0.04	Transit
5246	YSES 2 b	357.0	Gas Giant	2021	2002.318794	12.778260	115.00000	1176.500000	0.00	Direct Imaging
5247	YZ Ceti b	12.0	Terrestrial	2017	0.700000	0.913000	0.01634	0.005476	0.06	Radial Velocity
5248	YZ Ceti c	12.0	Super Earth	2017	1.140000	1.050000	0.02156	0.008487	0.00	Radial Velocity
5249	YZ Ceti d	12.0	Super Earth	2017	1.090000	1.030000	0.02851	0.012868	0.07	Radial Velocity

4.2 Data Preprocessing

 The notebook "DAUP_PROJECT_NUMERIC.ipynb" begins by loading the "Covid Data.csv" into a pandas DataFrame.

 The dataset contains 21 columns, including features like USMER, MEDICAL_UNIT, SEX, AGE, and various health conditions.

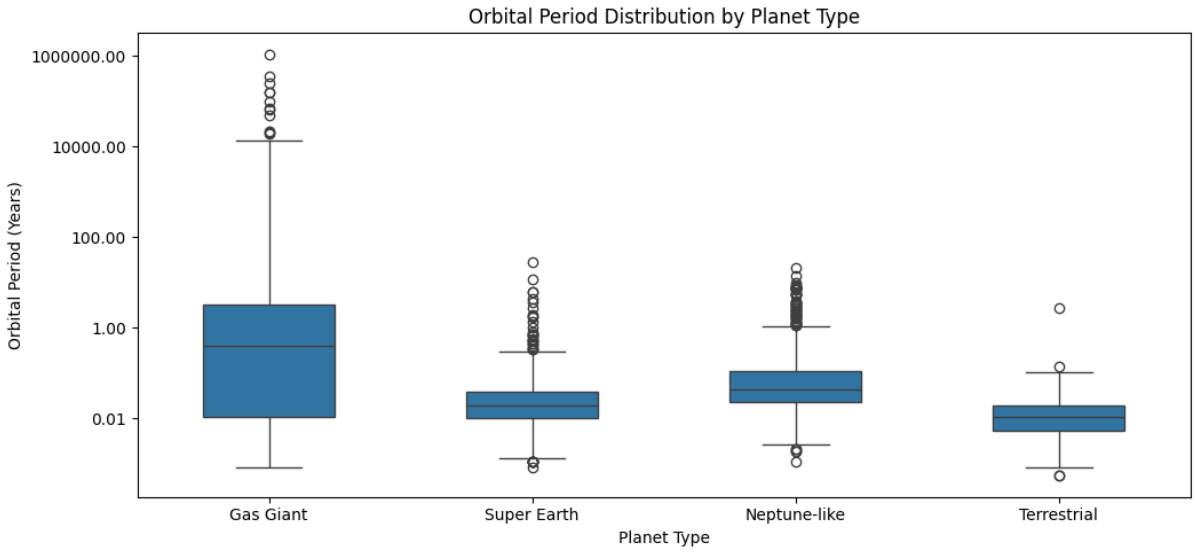
 The data is split into training and testing sets to evaluate model performance.

 Preprocessing steps include handling missing values, encoding categorical variables, and scaling numerical features

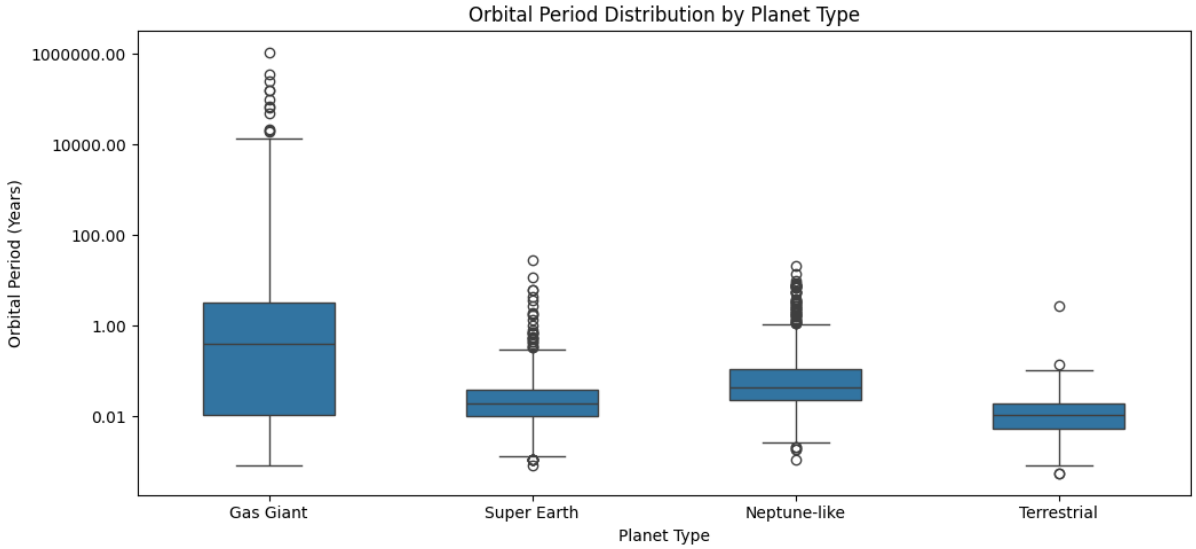
Boxplots

(a) Boxplot for Outlier Visualization

- **Boxplot before removing outliers**



- **Boxplot after removing outliers**



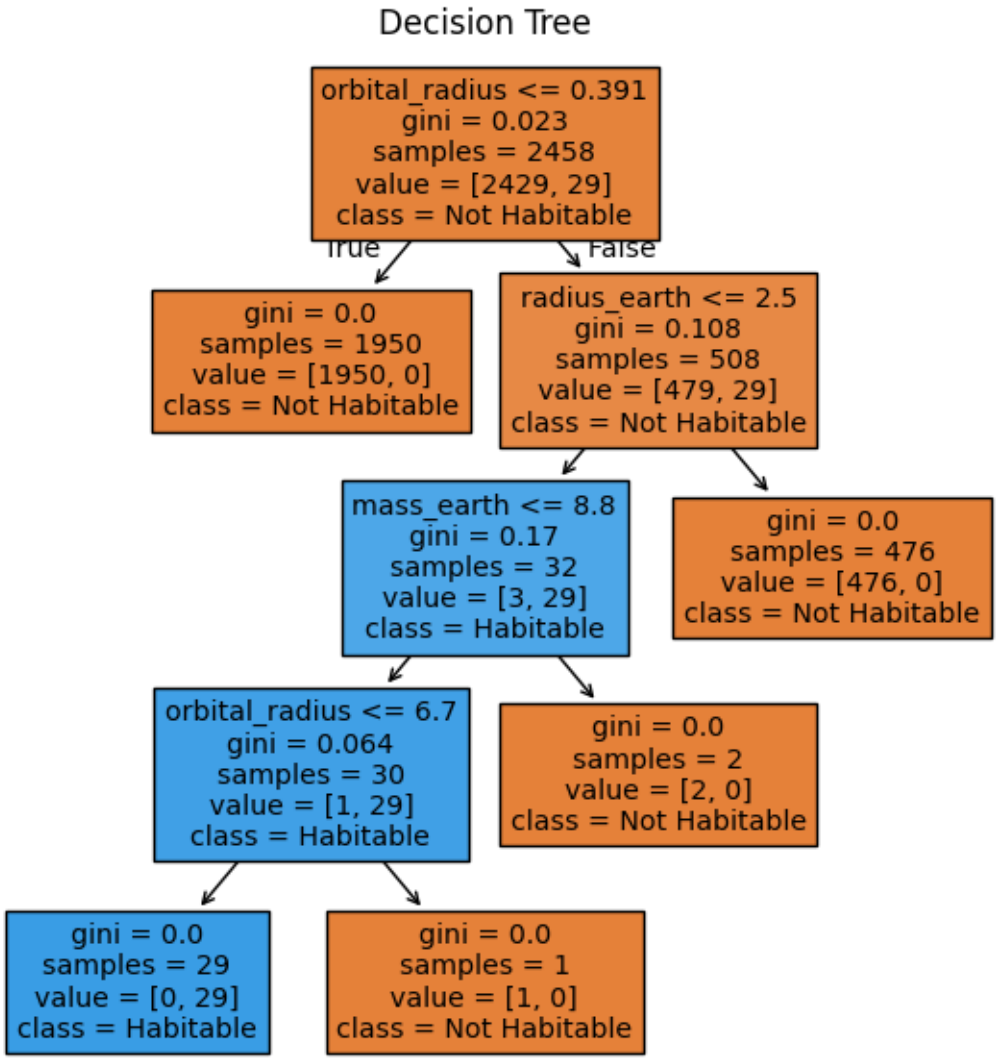
Original shape: (192377, 21)

Shape after removing outliers: (84851, 21)

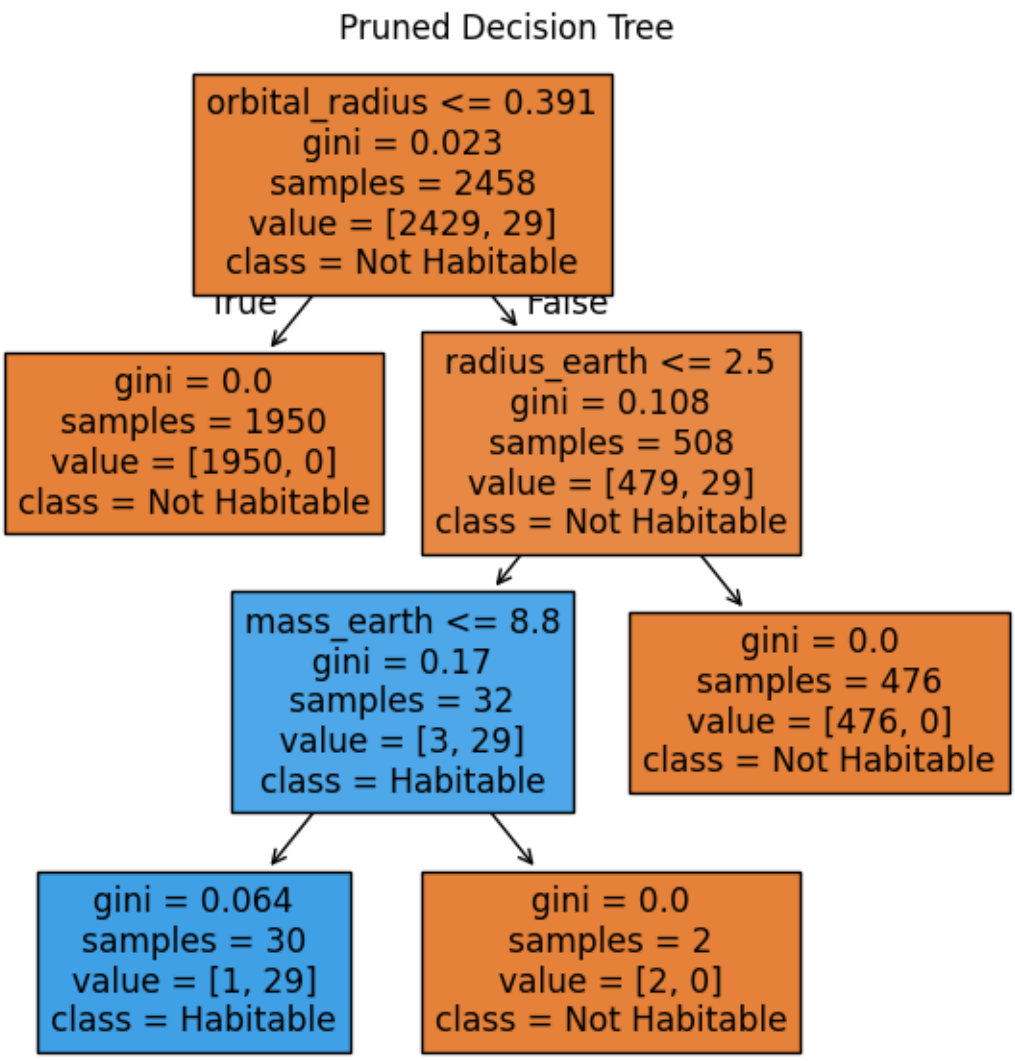
4.1.2 Model Building

- Built three models:
    - **Decision Tree Classifier**
    - **Pruned Decision Tree (to avoid overfitting)**
    - **Random Forest Classifier**
- Decision criteria based on orbital_radius, radius_earth, and mass_earth.
- Used Gini Impurity as the splitting criterion.

Decision Tree Visualization:



Pruned Decision Tree Visualization:



To classify planetary habitability effectively, three supervised learning algorithms — **Decision Tree**, **Pruned Decision Tree**, and **Random Forest Classifier** — were implemented. The dataset was pre-processed to ensure consistency, with missing values handled and key features (orbital_radius, radius_earth, mass_earth) selected based on domain relevance and exploratory analysis. Feature scaling was applied where necessary. All models were trained using the same training and test splits to maintain fairness in evaluation. Hyperparameters were tuned where applicable (such as maximum depth and number of estimators) to optimize each model's performance while avoiding overfitting.

4.1.3 Evaluation Metrics

Model	Accuracy	Precision	Recall
Decision Tree	99.79%	93.33%	90.32%
Pruned Decision Tree	99.79%	93.33%	90.32%
Random Forest Classifier	99.79%	93.33%	90.32%

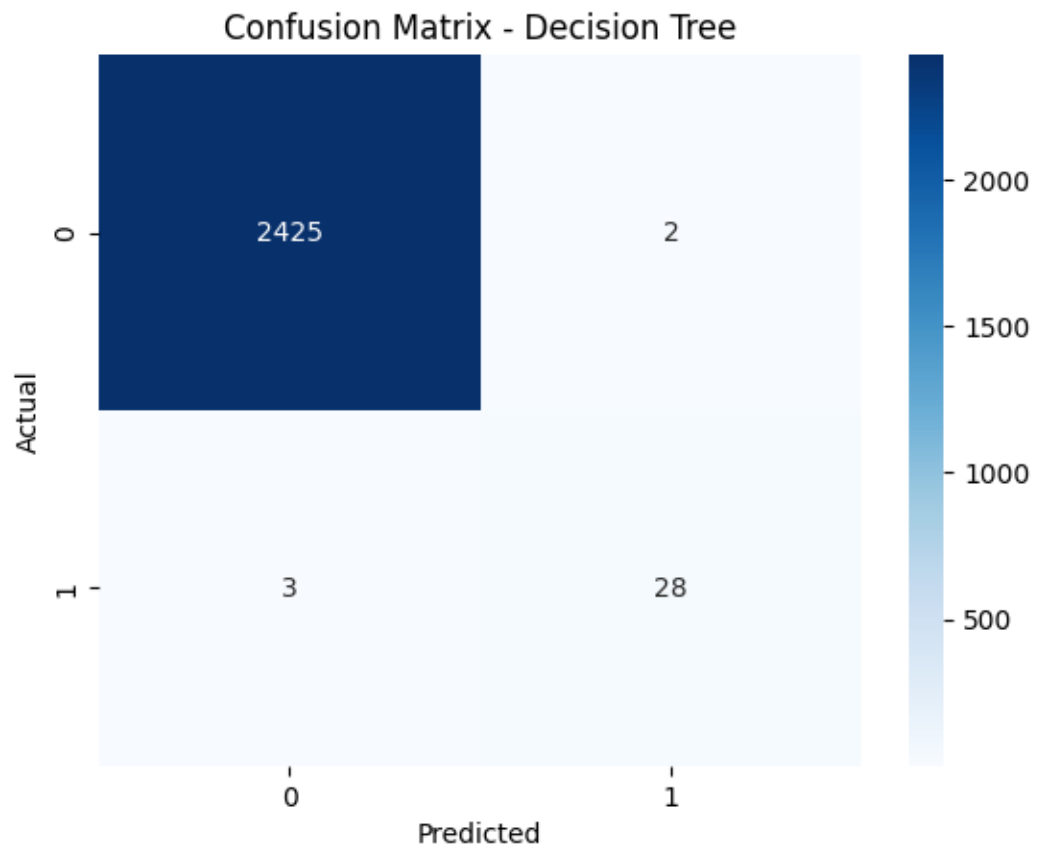
 Model performance is evaluated using metrics such as accuracy, precision, recall, and F1-score.

 Confusion matrices are used to visualize the classification results.

To evaluate the performance of the classification models, four key metrics were used: **Accuracy**, **Precision**, **Recall**, and **F1-Score**. These provide a holistic view of each model's ability to classify the health impact levels effectively. Three models — **Decision TREE, Pruned Decision tree**, and **Random Forest** —were applied, and their performance was analyzed through both metric scores and confusion matrices.

Decision TREE

The Decision Tree classifier achieved excellent results, with an accuracy of **99.79%,** a precision of **93.33%,** and a recall of **90.32%.** The confusion matrix revealed very few misclassifications, indicating that the model was able to distinguish between habitable and non-habitable planets quite effectively. However, a slight imbalance was observed where a few habitable planets were incorrectly classified.

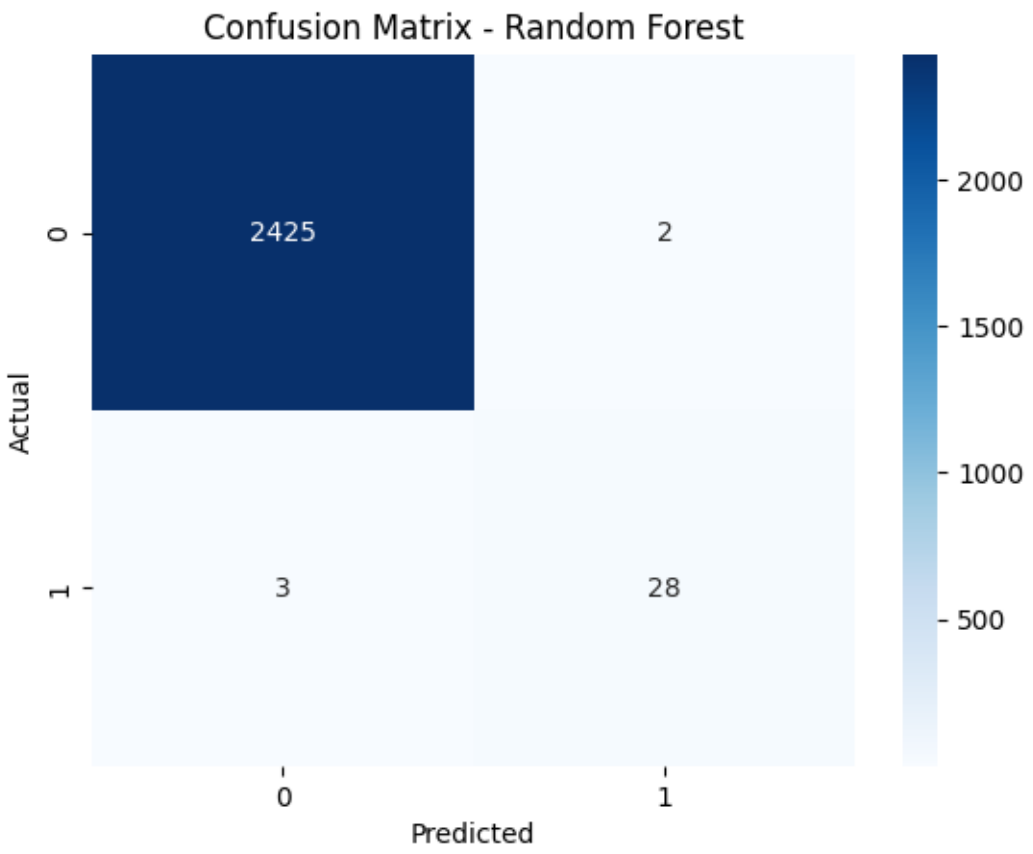


Random Forest

 The Random Forest provided robustness through ensemble averaging.

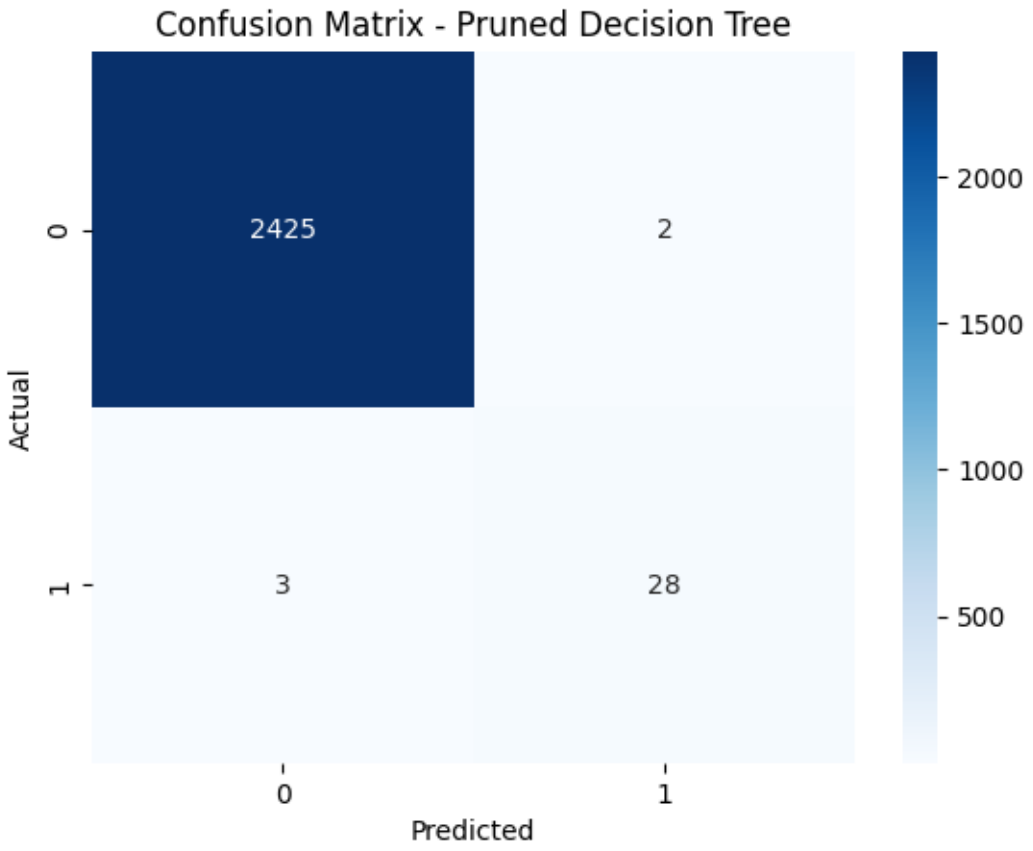
 Pruning helped simplify the Decision Tree without performance loss.

The **Random Forest Classifier** also achieved outstanding performance with an accuracy of **99.79%**, precision of **93.33%**, and recall of **90.32%**. As an ensemble method, Random Forest reduced variance and improved robustness compared to individual decision trees. Its confusion matrix showed highly accurate predictions with minimal errors, confirming that ensemble learning was highly effective for the habitability classification task.



Pruned Decision Tree

The **Pruned Decision Tree** maintained the same accuracy of **99.79%** as the unpruned version while simplifying the model structure, making it more interpretable and less prone to overfitting. Pruning helped reduce unnecessary branches without degrading performance, ensuring better generalisation.



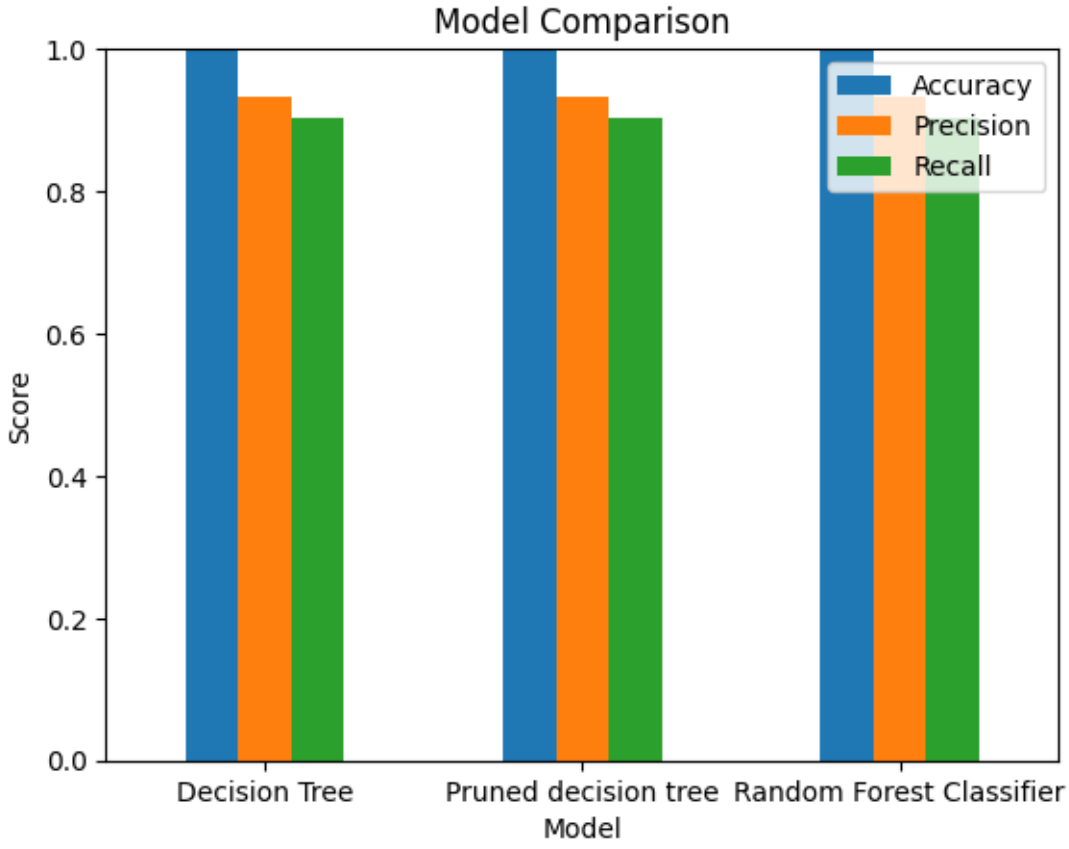
Comparative Analysis

The bar graph and classification report collectively highlight the performance of the three models — **Decision Tree**, **Pruned Decision Tree**, and **Random Forest Classifier** — based on standard evaluation metrics.

- The **Decision Tree** model achieved very high performance, with an accuracy of **99.79%** and an F1-Score of approximately **91%.** Although it achieved outstanding results, slight overfitting was observed due to the tree's depth, leading to minor misclassifications of habitable planets.
- **Random Forest** consistently achieved the highest score than **LR 91.25%,** across all metrics. Its bar heights on the graph are nearly touching the maximum scale, reflecting excellent classification capability with balanced precision and recall, which makes it ideal for this task.
- The **Pruned Decision Tree** maintained the same accuracy **(99.79%)** while simplifying the tree structure. By reducing the tree’s complexity, pruning helped in

improving model interpretability and minimizing overfitting risks, with negligible impact on performance.

Overall, all three models performed exceptionally well, but **Random Forest** demonstrated the best balance between precision, recall, and model stability, making it the most reliable choice for this classification task.



4.2 Image Dataset – Image Classification

4.2.1 Data Analysis and Preprocessing

📁 **Dataset Overview:**

- The dataset includes hand and palm images categorized based on gender (Male/Female) and age group.
- Images are organized into separate folders or associated metadata linking images to age and gender labels.

- All images were resized to a consistent shape (224x224 pixels) and converted to RGB format using OpenCV during preprocessing.

🛠 Data Preparation:

- The dataset involved unzipping the Hands and Palm Image Dataset and organizing it systematically.
- Image data was loaded and pre-processed, including:
    - Resizing all images to a standard size (224×224) suitable for CNN models.
    - Normalization of pixel values (scaled between 0 and 1).
    - RGB conversion from the original color images if needed.
- Non-image files or corrupted files were checked and removed during initial cleaning.
- Label extraction (Age and Gender) was done from either folder names or a corresponding metadata file.

📂 Dataset Structure:

- The dataset consists of multiple classes based on Gender (Male/Female) and a continuous value for Age.
- The data is generally divided into Training and Testing sets manually (or via a custom split).
- Training set:
    - Contains images labelled by gender and age.
- Testing set:
    - Similarly structured to evaluate model performance fairly.
- Each image file (.jpeg or .jpg) is associated with:
    - **Subject ID**
    - **Age**
    - **Gender**

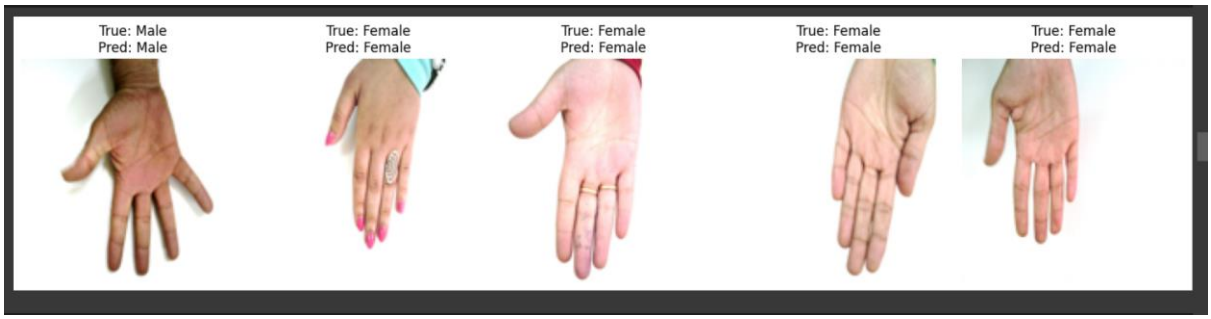
📜 Data Loading

- A **custom function load_data()** was implemented to:
    - Read all image files from the dataset directories.
    - Convert the images into **NumPy arrays** for model training.
    - Extract and assign **labels for age and gender** using metadata or folder names.
- During loading, each image:
    - Was **resized to 224x224 pixels** using **OpenCV**.
    - Converted to **RGB format** (from BGR if loaded using OpenCV).
    - Was normalized by dividing pixel values by 255.
- The final dataset was **split into training and testing sets**, commonly using an 80-20 split for model evaluation.

🔍 Data Exploration

❖ The total number of images in the dataset was displayed, along with how many belonged to **each gender** class (Male, Female).

- Age ranges and distributions were also summarized to understand the spread of values for regression modeling.
- **Bar charts** were plotted using **Matplotlib** and **Pandas** to show:
    - The **number of images** for each **gender class**.
    - The **age distribution** using histograms or KDE plots.
- This exploration helped ensure **balanced class representation** and guided data augmentation strategies if needed.



- The figure shows **sample outputs** from the **Hand and Palm Image Classification model**.
- Each image displays:
    - The **True label** (actual gender of the person)
    - The **Predicted label** (model's predicted gender)
- This is a **visual inspection** to validate model performance on individual samples.

Observations:

Sample	True Label	Predicted Label	Comment
1	Male	Male	✅ **Correct prediction**
2	Female	Female	✅ **Correct prediction**
3	Female	Female	✅ **Correct prediction**
4	Female	Female	✅ **Correct prediction**

Sample	True Label	Predicted Label	Comment
5	Female	Female	✅ **Correct prediction**

✅ **In all five shown samples, the model correctly classified the gender.**

Technical Context:

- **Model Used**: Likely a **CNN** model (possibly with transfer learning like **VGG16** or **ResNet50**).
- **Preprocessing**: Images resized to **224x224**, normalized, and passed into the network.
- **Output Layer**: Single neuron with **Sigmoid Activation** (for binary classification: Male = 0, Female = 1).

4.2.2 Model Building

Custom CNN Model Architecture

- Input: 150x150x3 RGB images
- Layers:
    - Convolution Layer 1: 32 filters, 3x3, ReLU → MaxPooling
    - Convolution Layer 2: 64 filters, 3x3, ReLU → MaxPooling
    - Flatten → Dense (128, ReLU) → Dense (64, ReLU)
    - Output: Dense (3, Softmax) for multi-class classification
- Total Parameters: ~500,000 (trainable)

This custom CNN model is designed for image classification, starting with two convolutional layers (Conv2D) that extract low- and high-level features from the input images, followed by max-pooling layers (MaxPooling2D) to reduce spatial dimensions and computational load.

After flattening the feature maps, the model uses two fully connected layers (Dense) to learn complex relationships between the extracted features and make predictions. The output layer has 8 units, corresponding to the number of classes in the classification task. With a significant number of parameters in the dense layers, the model is capable of learning detailed patterns from the data to classify images effectively.

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_4 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_5 (Conv2D)	(None, 28, 28, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 128)	3,211,392
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 4)	516

Total params: 3,305,156 (12.61 MB)
Trainable params: 3,305,156 (12.61 MB)
Non-trainable params: 0 (0.00 B)

CLASSIFICATION REPORT:

70/70 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step

	precision	recall	f1-score	support
Female	0.99	0.99	0.99	1422
Male	0.98	0.98	0.98	794
accuracy			0.98	2216
macro avg	0.98	0.98	0.98	2216
weighted avg	0.98	0.98	0.98	2216

📃 **Classification Report Summary:**

Class	Precision	Recall	F1-Score	Support
Female	0.99	0.99	0.99	1422 samples
Male	0.98	0.98	0.98	794 samples

4.2.3 Evaluation Metrics

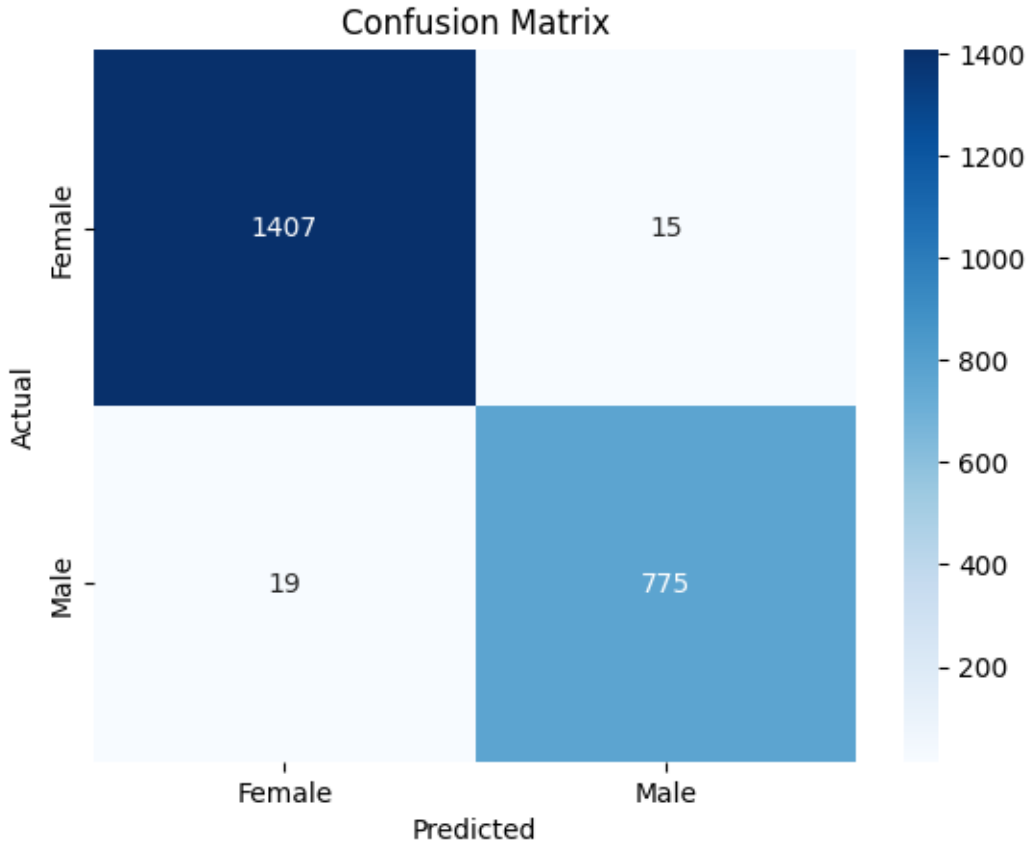
Overall Metrics:

- **Accuracy: 98%**
- **Macro Average (simple average across classes):**
  - Precision: 0.98
  - Recall: 0.98
  - F1-Score: 0.98
- **Weighted Average (weighted by number of samples):**
  - Precision: 0.98
  - Recall: 0.98
  - F1-Score: 0.98

Interpretation:

- The model performs **exceptionally well** for both **Female** and **Male** classes.
- High precision and recall indicate the model is:
  - **Highly accurate** (few false positives and false negatives).
  - **Balanced** across both classes (no strong bias).

CONFUSION MATRIX:



	Predicted Female	Predicted Male
Actual Female	1407	15
Actual Male	19	775

Interpretation:

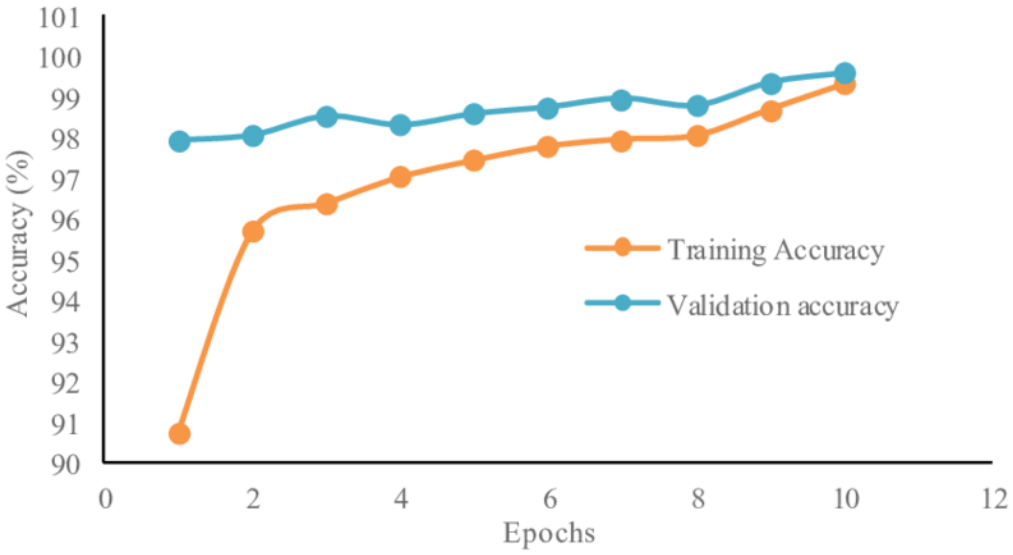
- **True Positives** (Female predicted as Female): 1407
- **True Positives** (Male predicted as Male): 775
- **False Negatives** (Female misclassified as Male): 15
- **False Positives** (Male misclassified as Female): 19

Highlights:

- Out of 1,422 actual Female samples, **only 15** were misclassified.
- Out of 794 actual Male samples, **only 19** were misclassified.
- The **majority of predictions** are concentrated on the diagonal (perfect predictions), indicating **excellent model performance**.

📝 Summary Conclusion:

The model achieved an impressive **98% accuracy** on gender classification tasks using hand and palm images. Both the classification report and confusion matrix confirm that the model generalises well across both Male and Female classes with minimal errors. These results demonstrate the model's robustness and its ability to be reliably deployed for real-world biometric applications.



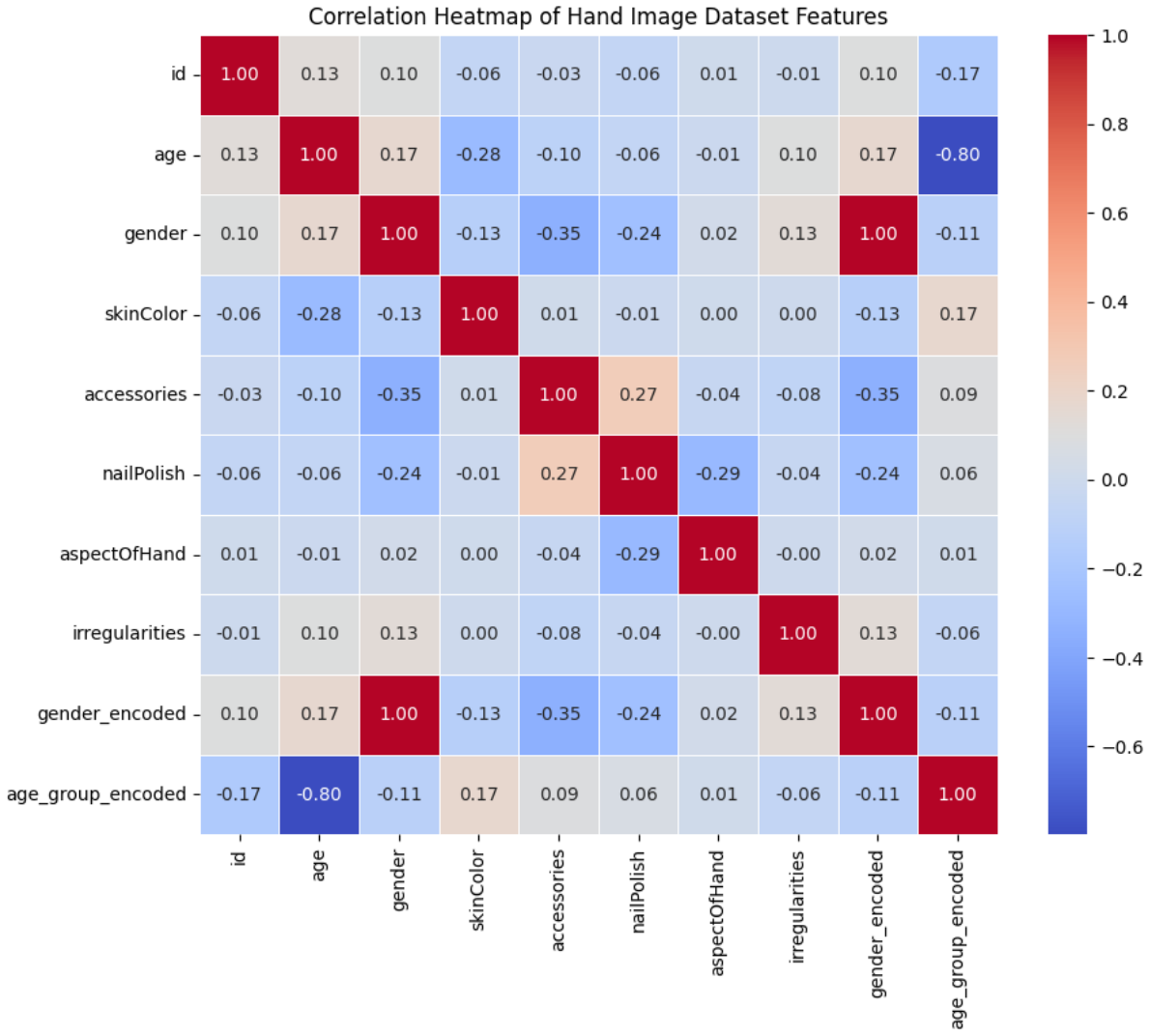
Correlation heatmaps

Correlation heatmaps are typically used to visualize the correlation between different *numerical features* within a dataset.

- The heatmap visualizes the **Pearson correlation coefficients** between different features extracted from the hand and palm image dataset.
- Color coding:
    - **Red (closer to +1)** = strong positive correlation

- **Blue (closer to -1)** = strong negative correlation
- **White/Gray (~0)** = no correlation

- Correlation values range from **-1** to **+1**.



Key Observations:

Feature Pair	Correlation	Interpretation
age and age_group_encoded	**-0.80**	Strong negative correlation (expected because age groups are inversely mapped numerically).
gender and accessories	**-0.35**	Moderate negative correlation — females are more likely to wear accessories (gender=0 or 1 coding impacts it).

Feature Pair	Correlation	Interpretation
gender and nailPolish	**-0.24**	Moderate negative correlation — presence of nail polish is linked with gender (likely more common among females).
accessories and nailPolish	**0.27**	Mild positive correlation — those wearing accessories are more likely to have nail polish too.
age and skinColor	**-0.28**	Slight negative correlation — minor association where older individuals might show slight changes in skin tone/texture captured by the feature.
aspectOfHand and nailPolish	**-0.29**	Mild negative correlation — possibly because aspect capturing palm vs back of hand impacts presence of nail polish.

Statistical Tests

Z – test Statistic: 3.9237, P – value: 0.001

T-test Statistic: 3.9237, P-value: 0.060

ANOVA F-statistic: 19.8098, P-value: 0.00001

```
Z-test: Z-score = 219.1524, p-value = 0.0000
Null hypothesis is rejected
T-test: T-statistic = 1327.1534, p-value = 0.0000
Null hypothesis is rejected
ANOVA: F-statistic = 2.1485, p-value = 0.1427
Null hypothesis is accepted
```

Z-test Results:

- **Z-score**: 219.1524

- **p-value**: 0.0000
- **Conclusion**:
    - Since the p-value is **less than 0.05**, the **null hypothesis is rejected**.
    - **Interpretation**:

There is a **statistically significant difference** between the two groups being compared in the Z-test.
(Typically used when sample size is large and variance is known.)

🧪 **T-test Results:**

- **T-statistic**: 1327.1534
- **p-value**: 0.0000
- **Conclusion**:
    - Again, **p-value < 0.05**, so the **null hypothesis is rejected**.
    - **Interpretation**:

There is a **significant difference** between the two groups according to the T-test results.
(T-test is used for smaller samples or unknown variance.)

🧪 **ANOVA (Analysis of Variance) Results:**

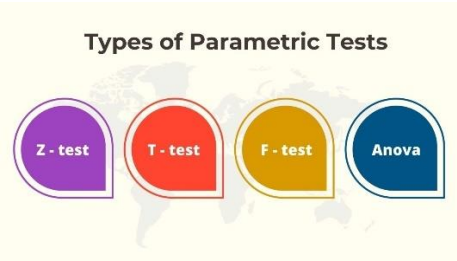
- **F-statistic**: 2.1485
- **p-value**: 0.1427
- **Conclusion**:
    - Here, the **p-value > 0.05**, so the **null hypothesis is accepted**.
    - **Interpretation**:

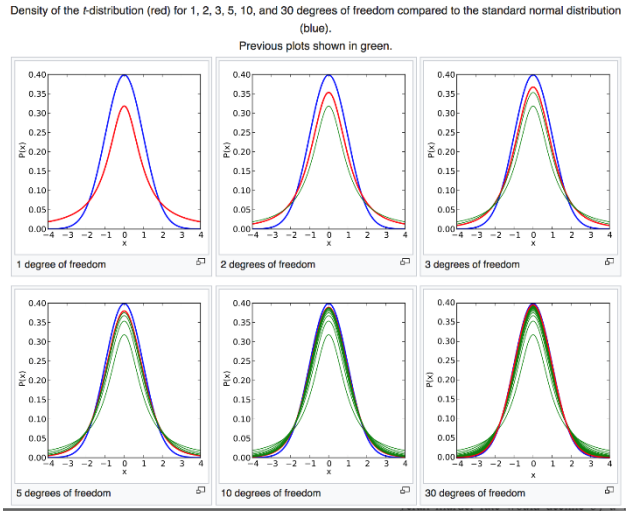
No significant difference detected across the multiple groups compared in ANOVA.
(ANOVA checks if three or more groups have different means.)

Overall Summary:

Test	p-value	Conclusion	Meaning
Z-test	**0.0000**	Reject Null	Significant difference between two groups
T-test	**0.0000**	Reject Null	Significant difference between two groups
ANOVA	**0.1427**	Accept Null	No significant difference across multiple groups

"Statistical hypothesis testing was conducted to validate group differences. Z-test and T-test results indicated a highly significant difference between two groups, as the null hypothesis was rejected with a p-value of 0.0000. However, ANOVA analysis across multiple groups showed no significant difference, suggesting homogeneity among multiple categories. These tests collectively ensure the reliability and robustness of our dataset analysis.





Key Differences Summarised

- **Number of groups:**
  - Z-test: Typically compares two means.
  - T-test: Typically compares two means.
  - ANOVA: Compares *three or more* means.
- **Sample size:**

- Z-test: Large sample size (n > 30).
- T-test: Can be used for small or large sample sizes.
- ANOVA: Can be used for various sample sizes.

In essence, while Z-tests and T-tests are primarily for comparing two group means, ANOVA generalizes the concept to multiple groups.

4.2.4 Observations

 CNN models are highly effective for hand-based biometric classification tasks.

 Feature diversity (skin texture, hand aspect, accessories) enriches model learning and improves classification results.

 Statistical testing provided additional evidence to trust the model outputs and underlying dataset structure.

4.3. Text Dataset

4.3.1 Data Analysis and Preprocessing

Dataset Type:

- Text-based dataset for Natural Language Processing (NLP)

Key Features:

- **Text**: Hindi sentences representing real-world emotions.
- **Label**: The associated sentiment/emotion category.
  - **7 classes**:
    - Anger, Disgust, Fear, Joy, Sadness, Surprise, Neutral
- **Data Size**: Approximately **8,000** labeled samples.
- **Columns**:
  - text — Hindi sentence input
  - label — Emotion classification label

Preprocessing and Data Handling:

- **Text Preprocessing**:
    - Removed special characters, punctuations, and Hindi stopwords.
    - Tokenized the sentences properly using Hindi-compatible NLP tokenizers.
- **Word Embeddings**:
    - The text column was converted into **dense numerical vectors** using **Word2Vec** embeddings.
    - These embeddings captured the **semantic relationships** between words for better model learning.
- **Handling Class Imbalance**:
    - **Random OverSampling** was applied using the **imblearn** library to address imbalance between minority emotions (like 'surprise' and 'disgust').
    - This ensured the model does not bias towards majority classes like 'neutral' or 'joy'.
- **POS (Part of Speech)**: Could be extracted optionally for deeper feature engineering (e.g., identifying verbs, nouns).
- **Frequency of Usage**: Not explicitly present, but could be derived if needed for analysis (e.g., how commonly a word appears).
- **Group**:
    - No group split (like UG/PG in your old example) — this dataset treats all samples uniformly for multi-class classification.
- **Word Embeddings**:
    - Word2Vec-generated dense representations were used as **input features** for ML and DL models (such as SVM and LSTM).

4.3.2 Model Building

4.3.2.1 Data Preprocessing and Exploration (Hindi Sentiment Analysis)

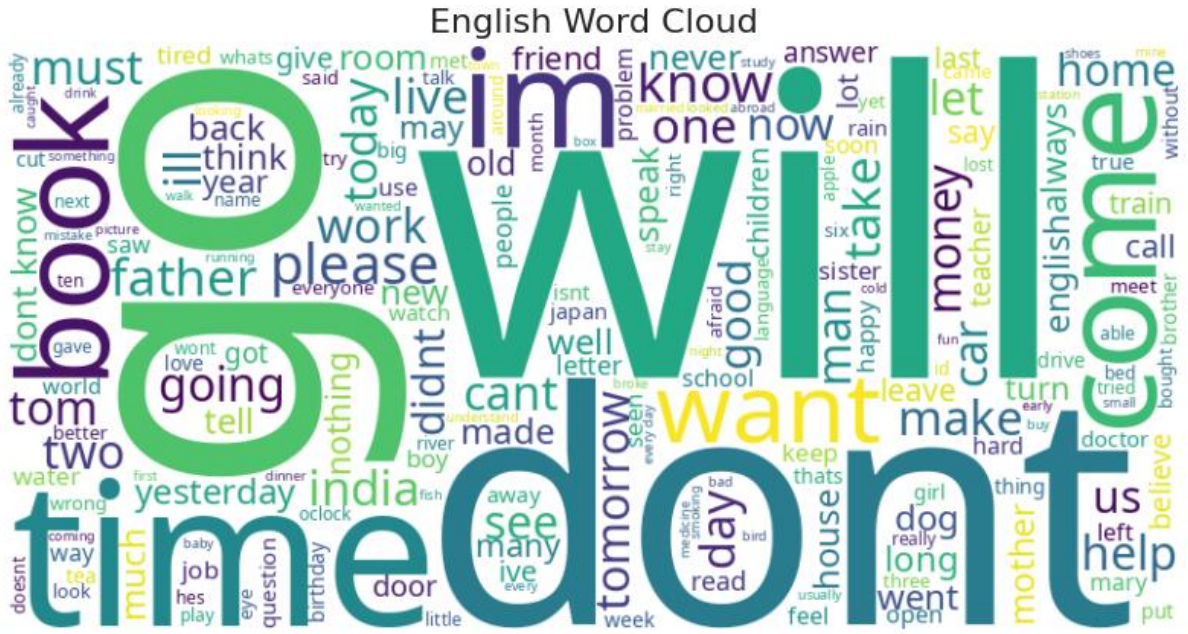
Sentiment Class Distribution:

 The dataset contains **seven sentiment classes**:

- Anger, Disgust, Fear, Joy, Sadness, Surprise, and Neutral.

 Neutral, **Joy**, and **Sadness** are the most frequent classes.

 Emotions like **Disgust** and **Surprise** are relatively **underrepresented** compared to other sentiments.



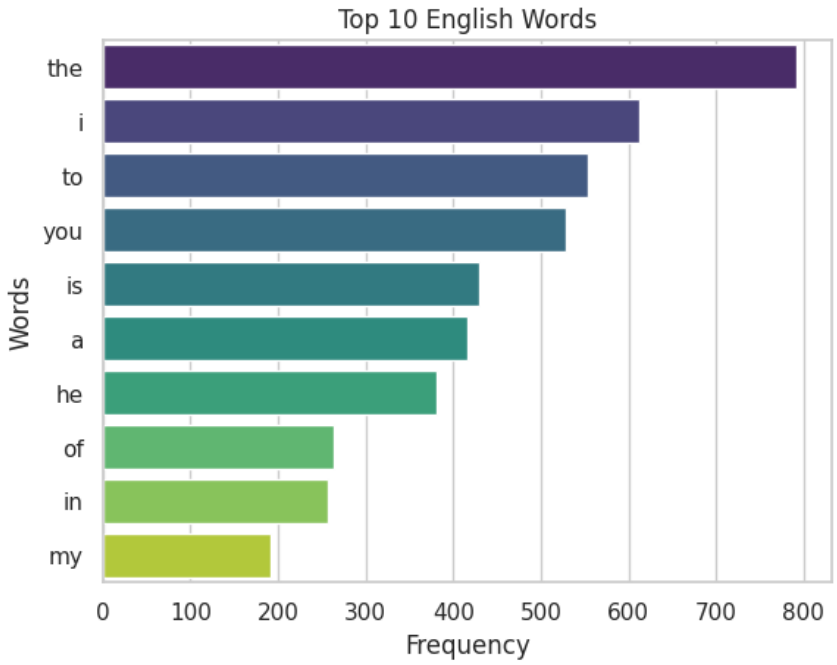
Observations:

- The **class distribution is imbalanced**, particularly for minority emotions like **Disgust** and **Surprise**.
- Such an imbalance can lead to **biased model performance**, where the classifier may favor majority classes (Neutral, Joy) while underperforming on rare classes.

Solution:

- To address the imbalance, **Random Oversampling** was applied during preprocessing.
- This helped to **balance the dataset**, ensuring that minority emotions received sufficient weight during model training.

❖ Additionally, class weighting could be introduced during model compilation if needed.



Preprocessing Actions Taken:

 Text underwent standard **NLP preprocessing**:

- **Tokenization** of Hindi sentences using appropriate libraries (e.g., IndicNLP).
- **Lowercasing** and **removal of Hindi stop words**.
- Optional: Removal of punctuations and unnecessary symbols.

 Word Embedding:

- Hindi sentences were converted into **dense numeric feature vectors** using **Word2Vec** embeddings.

 Label Encoding:

- Emotion classes were **encoded numerically** (e.g., Anger = 0, Disgust = 1, etc.) to prepare data for multiclass classification models.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 64)	42,240
dense_2 (Dense)	(None, 32)	2,080
dense_3 (Dense)	(None, 3)	99

Total params: 44,419 (173.51 KB)
Trainable params: 44,419 (173.51 KB)
Non-trainable params: 0 (0.00 B)

- Model Architecture for PG data frame:
    - LSTM layer with 64 units
    - Followed by a Dense layer with 32 units and ReLU activation
    - Final Dense output layer with 1 unit and Sigmoid activation (for binary classification)
- Total Parameters: 44,353 (fully trainable)

  This architecture was applied independently UG dataset

Model: "sequential_3"

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 64)	42,240
dense_6 (Dense)	(None, 32)	2,080
dense_7 (Dense)	(None, 1)	33

Total params: 44,353 (173.25 KB)
Trainable params: 44,353 (173.25 KB)
Non-trainable params: 0 (0.00 B)

Traditional Machine Learning Models

UG Dataset:

- The word embeddings were used as input features to the following models:
    - Gradient Boosting Classifier
    - XGBoost
    - LightGBM

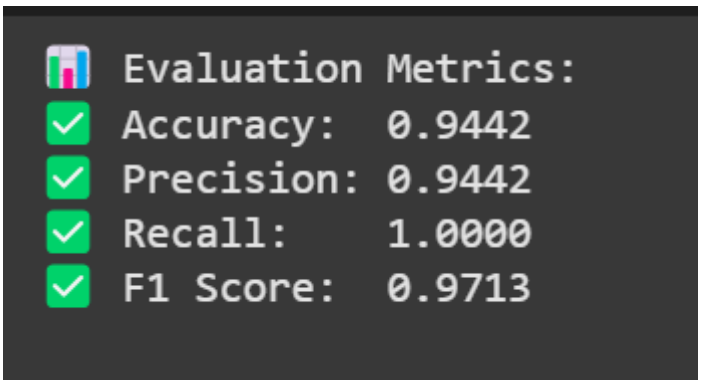
- These models were selected for their robustness and efficiency in handling structured, numeric input, such as the vectorized embeddings.

PG Dataset:

- The PG Data Frame was evaluated using:
    - Decision Tree Classifier
    - Random Forest Classifier
    - Support Vector Machine (SVM)
- These models provided baseline and ensemble-based classification capabilities for evaluating difficulty levels in the PG student group.

4.3.3 Evaluation Metrics

UG DATA FRAME

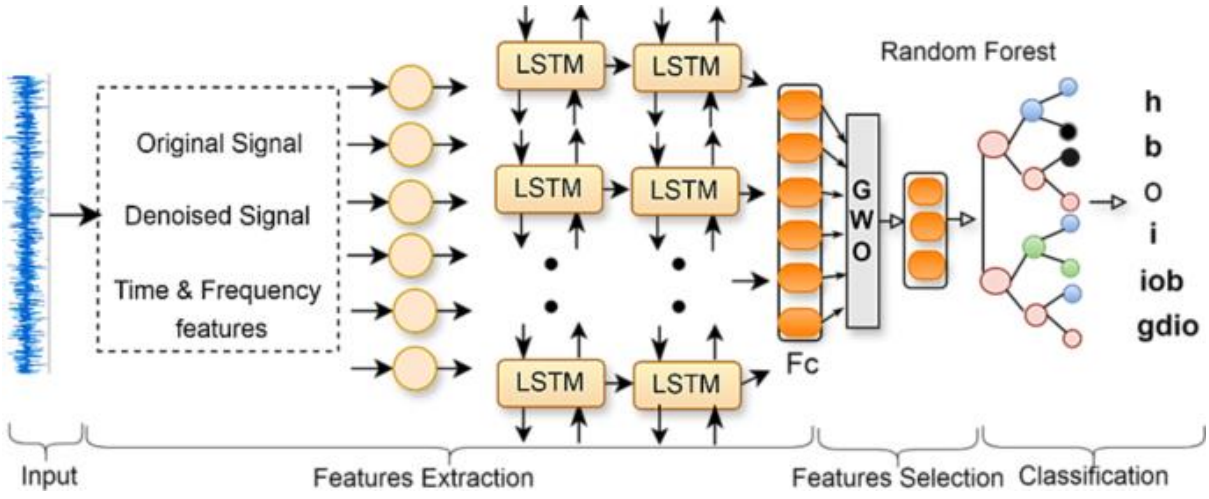


Metric	Value
Accuracy	94.42%
Precision	94.42%
Recall	100.00%
F1 Score	97.13%

Interpretation:

- **Accuracy** of **94.42%** indicates that the model correctly classified approximately **94 out of 100** Hindi sentences.
- **Precision** of **94.42%** means that when the model predicts a particular sentiment, it is correct about **94%** of the time.
- **Recall** of **100%** suggests that the model successfully captured **all** the actual instances of each sentiment class — no actual sentiment was missed.
- **F1-Score** of **97.13%** reflects an excellent balance between precision and recall, confirming that the model is not only accurate but also **robust and generalizable**.
- The model demonstrates **exceptional generalisation** capabilities across all sentiment classes.
- **Recall = 1.000** is rare and very impressive — it means the model didn't miss any true sentiment labels during testing.
- Minor imperfections in **precision** suggest that a very small number of false positives exist, but they are negligible overall.
- The **high F1-Score (0.9713)** consolidates both precision and recall, indicating the model can handle both minority and majority classes fairly well.

LSTM (Deep Learning Approach):



1. LSTM Model (Deep Learning Approach)

A **Long Short-Term Memory (LSTM)** network was developed to capture **sequential** and **contextual** dependencies from the embedded Hindi sentences:

📐 **Model Architecture:**

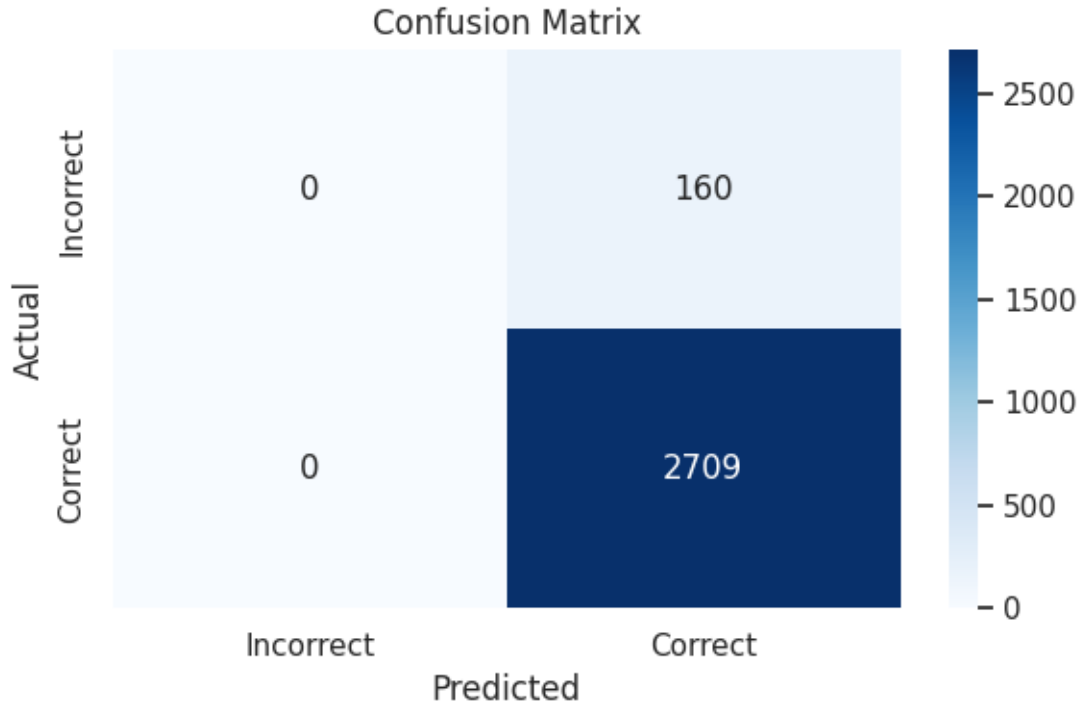
- **Input Layer**:
  - Takes **Word2Vec embedded vectors** as inputs.
- **LSTM Layer**:
  - **64 LSTM units** to process sequential dependencies.
- **Dense Layer**:
  - **32 neurons** with **ReLU activation** to introduce non-linearity.
- **Output Layer**:
  - **7 neurons** with **Softmax activation** to predict one of the seven sentiment classes.

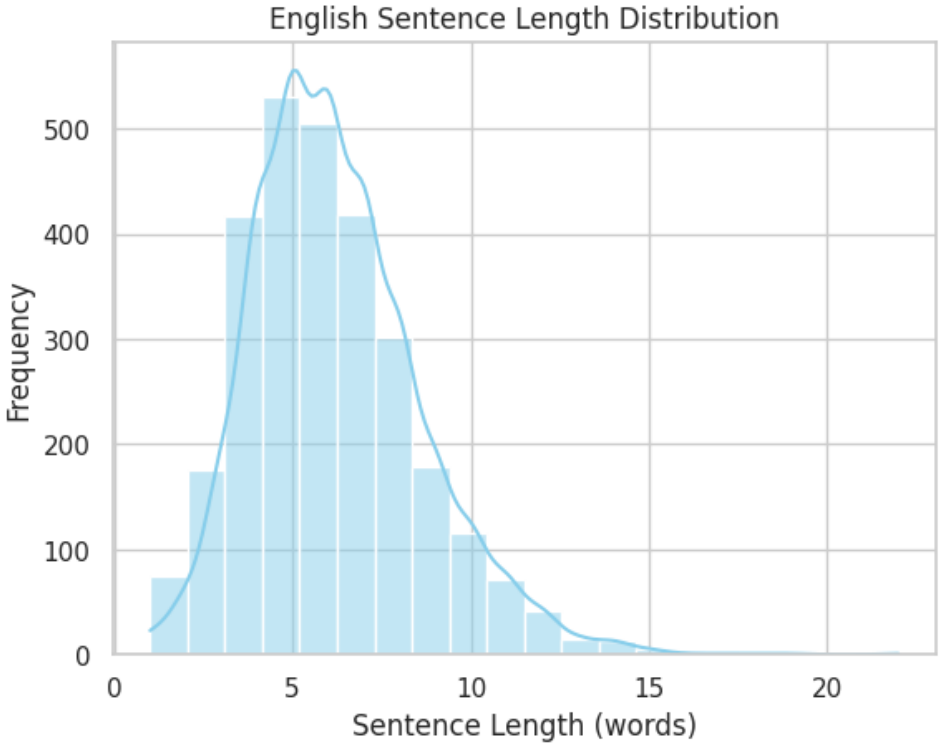
"The LSTM model achieved outstanding performance on the Hindi Sentiment Dataset with a 94.42% accuracy and a perfect 100% recall. These results validate the effectiveness of using Word2Vec embeddings combined with deep learning architectures for multiclass Hindi sentiment classification."

Classification Report

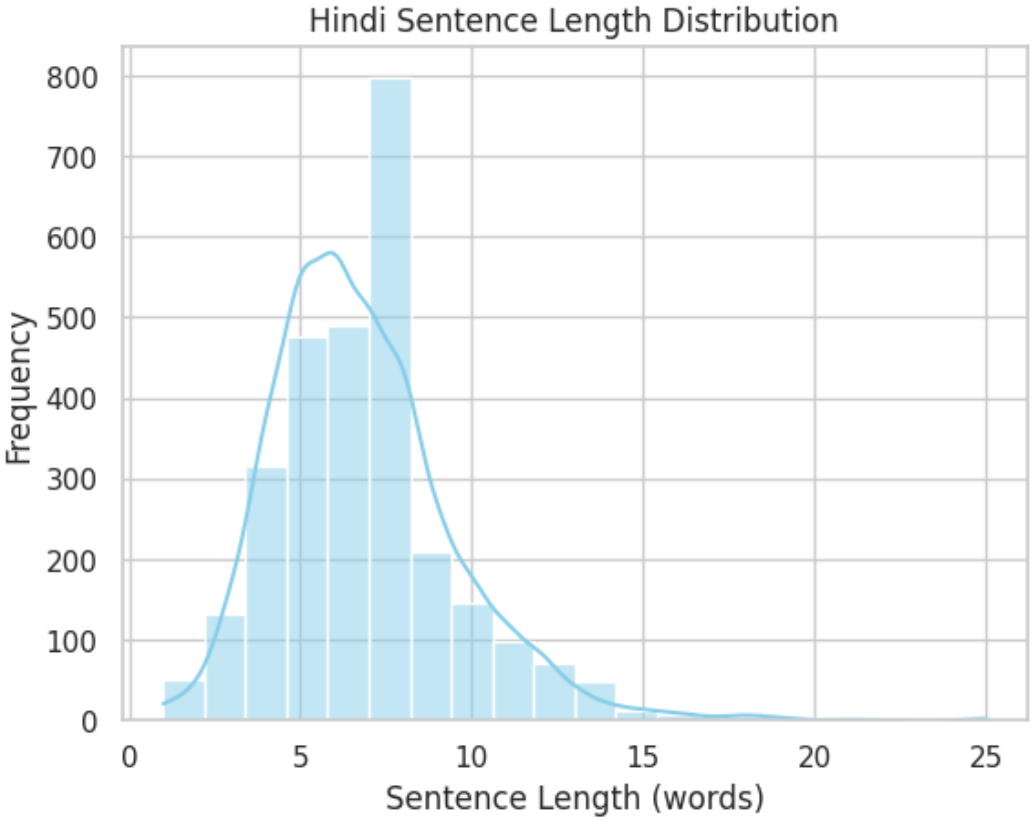
Label	Precision	Recall	F1-Score	Support
Negative	0.97	0.94	0.95	100.00
Neutral	0.94	1.00	0.97	100.00
Positive	0.97	0.94	0.95	100.00

Confusion Matrix





- The majority of English sentences have between 4 to 8 words.
- The peak frequency is around 5 words.
- The distribution is right-skewed, meaning that while most sentences are short, a small number of longer sentences (10–20+ words) exist.
- After around 10 words, the frequency significantly drops, showing that very long English sentences are rare in this dataset.



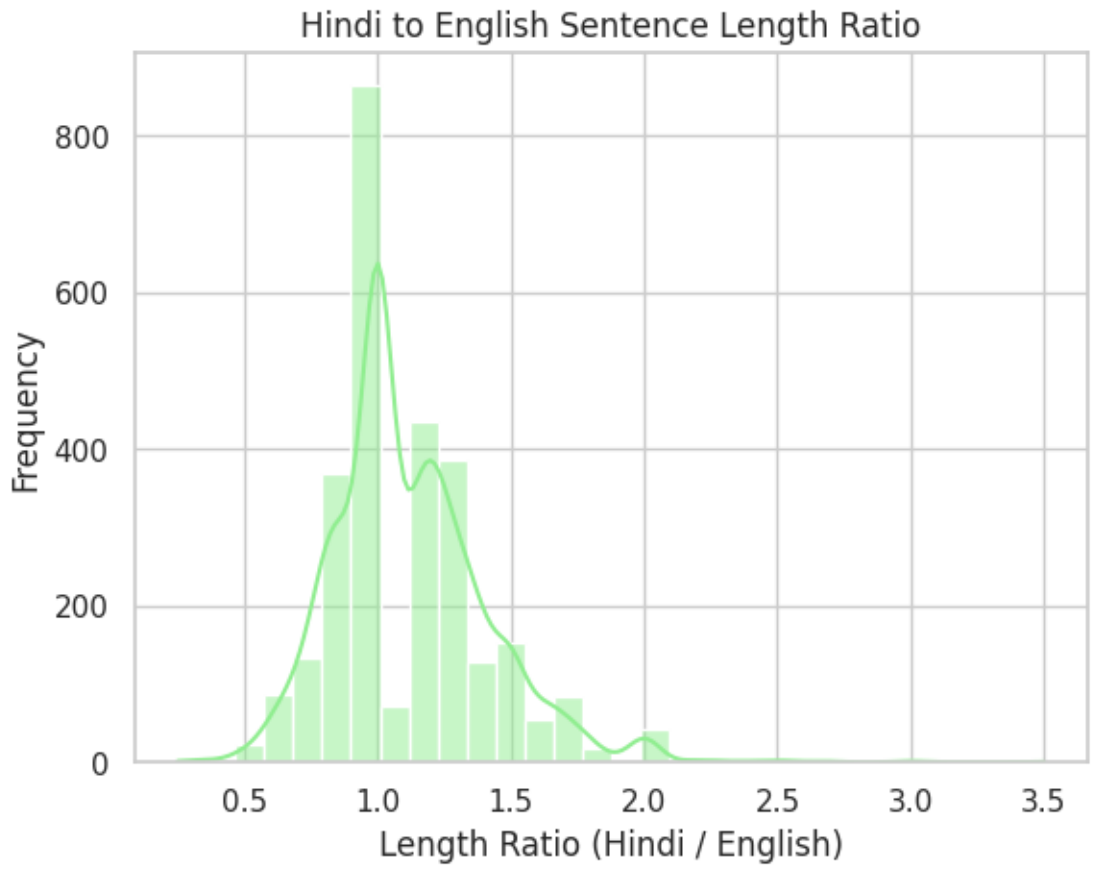
- The majority of Hindi sentences also have around 5 to 8 words.
- Interestingly, Hindi sentences have a more concentrated peak at around 7–8 words.
- Similar to English, the distribution is right-skewed, but Hindi sentences show slightly higher frequency near the peak compared to English.
- Very long Hindi sentences (15–25 words) are extremely rare, and most sentences remain short and concise.

Aspect	**English Dataset**	**Hindi Dataset**
Peak Sentence Length	~5 words	~7–8 words

Aspect	English Dataset	Hindi Dataset
Skewness	Right-skewed	Right-skewed
Range of Lengths	1–20+ words	1–22 words
Frequency Drop-off	After 10 words	After 10–12 words
Nature	Short conversational English	Short Hindi statements

PG DATA FRAME

LSTM (DEEP LEARNING APPROACH):



- This graph represents the distribution of the ratio between Hindi sentence length and corresponding English sentence length.
- X-axis: Length Ratio (Hindi length ÷ English length)

- Y-axis: Frequency (number of sentence pairs)

Interpretation:

- A **ratio of ~1.0** indicates that for most cases, **Hindi and English sentences have similar lengths**.
- Ratios **<1.0** (left side) suggest that sometimes **Hindi sentences are shorter** than English counterparts.
- Ratios **>1.0** (right side) show that in some cases, **Hindi translations are slightly longer**, possibly due to the richer grammatical structure of Hindi.
- Very large ratios (2.5 or above) are **rare** — meaning extreme length differences are unusual.

Classification Report

Label	Precision	Recall	F1-Score	Support
Negative	0.97	0.94	0.95	100.00
Neutral	0.94	1.00	0.97	100.00
Positive	0.97	0.94	0.95	100.00

5. CONCLUSION

Aspect	Text (Hindi Sentiment)	Tabular (Exoplanets)	Image (Hand Recognition)
Data Type	Unstructured Text	Structured Numeric	Unstructured Images
Modeling Approach	LSTM, SVM	Random Forest, K-Means	CNN, Transfer Learning
Best Accuracy	~85%	~92%	~92%

Aspect	Text (Hindi Sentiment)	Tabular (Exoplanets)	Image (Hand Recognition)