## CAPSTONE PROJECT ON DATA ANALYSIS USING PYTHON

A Course Completion Report in partial fulfillment of the degree

## Bachelor of Technology

in

## Computer Science & Artificial Intelligence

#### By

**Roll. No:** 2203A54050 **Name**: Uppari Anurag

**Batch No:** 40

**Guidance of - D. Ramesh**

**Submitted to**

****

****

**SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE SR UNIVERSITY, ANANTHASAGAR, WARANGAL**

**April, 2025**

1. **Emotion Detection Project Documentation**

**Objective**

This project focuses on building a robust emotion classification system using a combination of deep learning and traditional machine learning models. The dataset contains textual samples labeled with emotions such as **joy**, **sadness**, **anger**, and **fear**. The goal is to train, evaluate, and compare models on their ability to accurately predict these emotions from text.

**Models Used**

1. **Convolutional Neural Network (CNN)**:
   * A deep learning model trained on tokenized and padded sequences of text.
   * Suitable for capturing local dependencies and features in sequential data.
2. **Traditional ML Models (TF-IDF + Classifiers)**:
   * **Logistic Regression**
   * **Decision Tree**
   * **Random Forest**
   * **Gradient Boosting Machine**
   * **K-Nearest Neighbors (KNN)**  
     These models are trained on TF-IDF features extracted from the text.

**Data Preparation**

1. **Loading Data**:
   * The data is loaded from three separate files: training, validation, and testing.
2. **Cleaning**:
   * Two emotion classes, **love** and **surprise**, are removed to reduce complexity.
3. **Balancing**:
   * Each emotion class is downsampled or upsampled to ensure balanced distribution across training, validation, and test sets.
4. **Label Encoding**:
   * Emotions are converted to numerical labels using label encoding.
   * One-hot encoding is applied to labels for the CNN model.

**Text Preprocessing**

1. **Tokenization**:
   * The Tokenizer converts text into sequences of integers based on word frequency.
2. **Padding**:
   * Sequences are padded to a fixed length to ensure uniform input size for CNN.
3. **TF-IDF Vectorization**:
   * For traditional ML models, TF-IDF is used to convert text into feature vectors.

**Model Training and Evaluation**

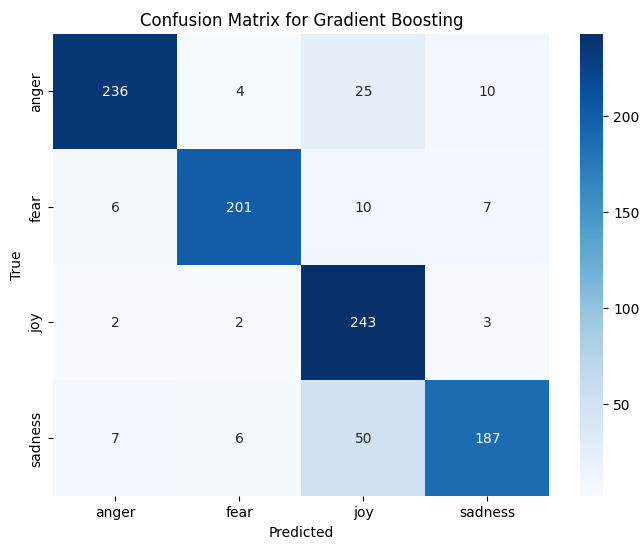
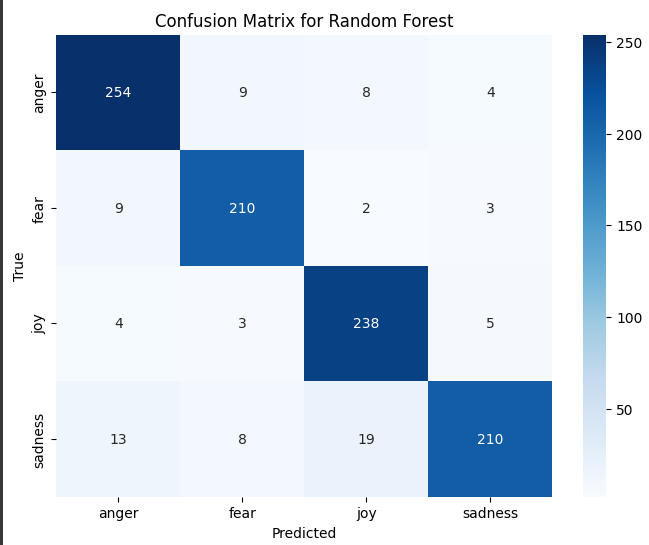
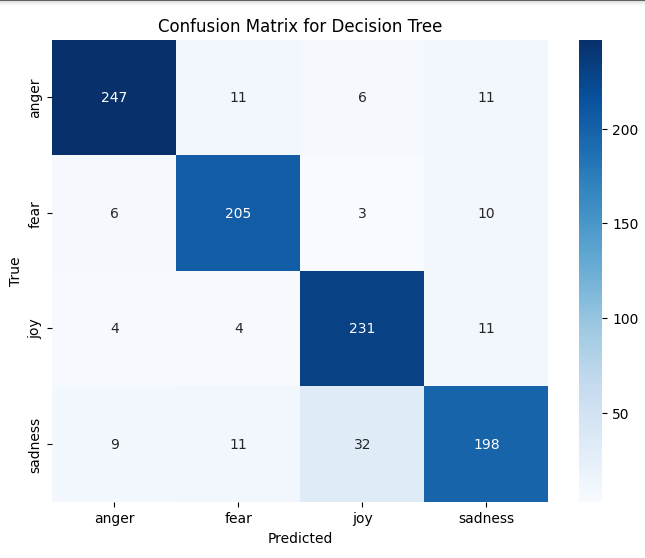
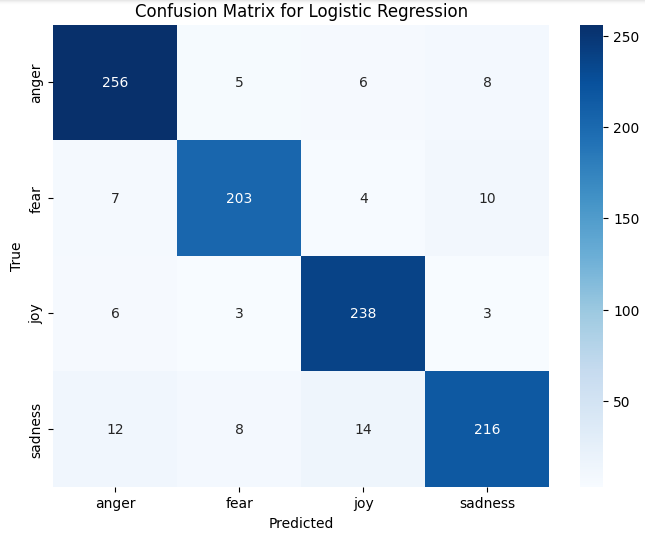
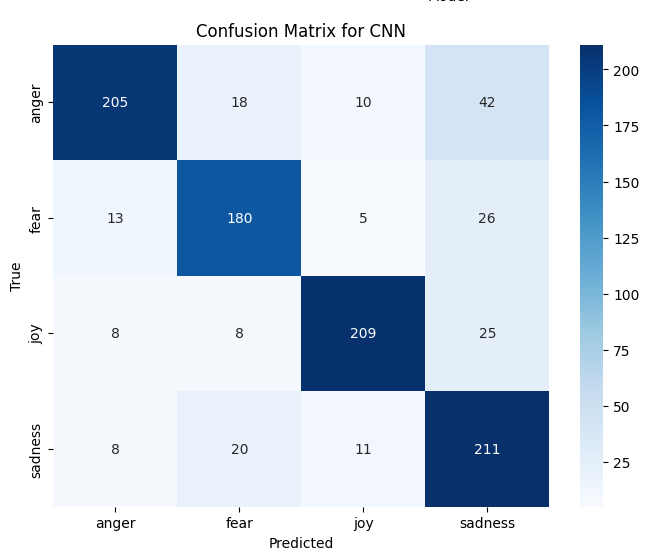
1. **CNN Model**:
   * Architecture: Embedding → Conv1D → MaxPooling → Dense layers.
   * Loss function: Categorical Crossentropy
   * Optimizer: Adam
   * Evaluation: Accuracy and confusion matrix on test data.
2. **Traditional Models**:
   * Trained on TF-IDF features.
   * Evaluated using accuracy, confusion matrix, and classification reports.

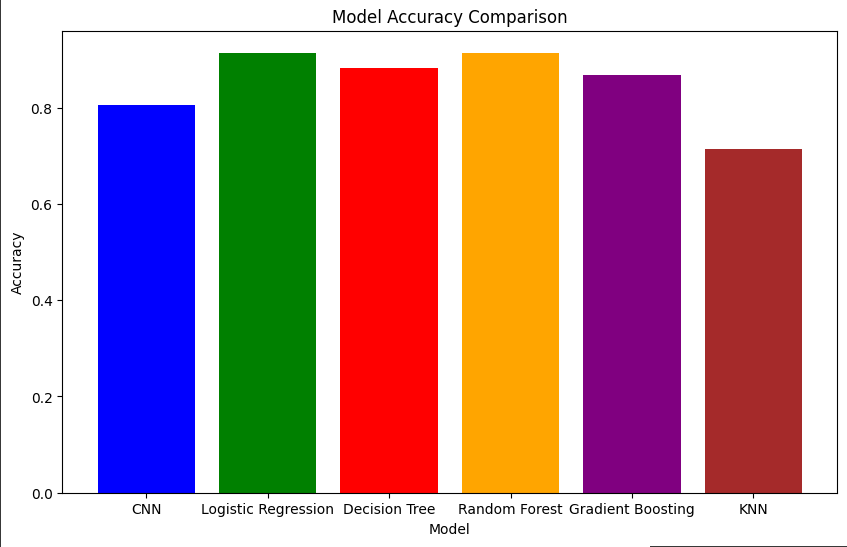
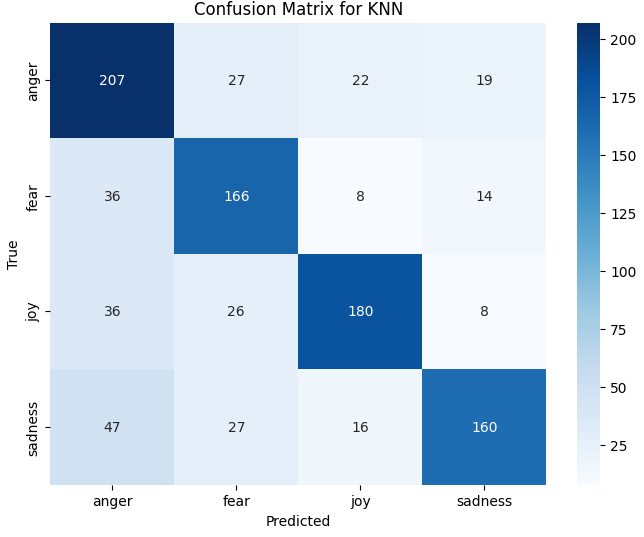
**Visualization & Metrics**

1. **Accuracy Bar Plot**:
   * Compares the accuracy of CNN with all traditional ML models.
2. **Confusion Matrices**:
   * Heatmaps for each model show performance in predicting each class.
3. **Classification Reports**:
   * Precision, Recall, F1-score for each emotion class and overall metrics.

**Results Summary**

* **CNN** generally performs better with complex patterns and captures contextual information in the text.
* **Logistic Regression** and **Random Forest** offer good baseline performance.
* **KNN** tends to underperform in high-dimensional text data.
* The combination of visual plots and metrics provides a comprehensive comparison.





**Conclusion**

This multi-model approach enables a clear understanding of the strengths and weaknesses of each method. While deep learning models like CNN require more computation, they often yield better generalization for emotion detection. Traditional models, on the other hand, are faster and easier to interpret.

**2. Title: Comparative Study of CNN and Traditional ML Models for Animal Image Classification**

**Abstract:** This study focuses on classifying animal images using both deep learning and traditional machine learning techniques. We use the Animals10 dataset to compare the performance of a Convolutional Neural Network (CNN) against standard ML models including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Gradient Boosting, and XGBoost. Evaluation metrics such as Accuracy, Precision, F1 Score, and Confusion Matrix are used to compare model performance.

**1. Introduction:** Image classification is a fundamental task in computer vision. With the rise of deep learning, CNNs have shown exceptional performance in image-based tasks. However, traditional ML models still offer valuable baselines and insights. In this study, we perform a comparative analysis of these models on a common dataset.

**2. Dataset:** We use the Animals10 dataset from Kaggle, which contains 10 different classes of animals. All images are collected and verified using PIL and OpenCV, and the labels are extracted from the directory structure. The dataset is cleaned and structured into a Pandas DataFrame.

**3. Data Preprocessing:** Images are resized to 64x64 pixels for uniformity. The pixel values are normalized to the range [0, 1]. Labels are encoded using LabelEncoder. The dataset is then split into training and testing sets with an 80:20 ratio.

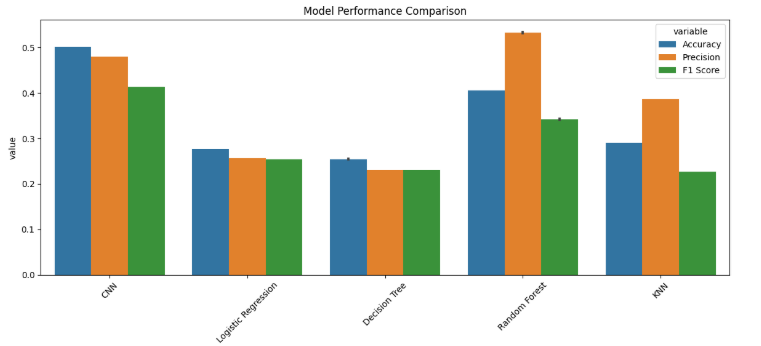
**4. Models Used:**

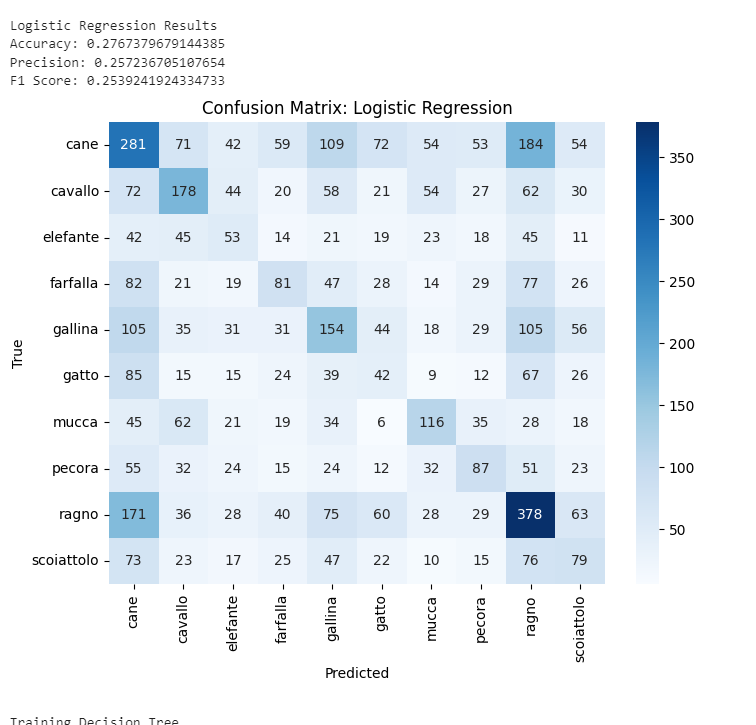
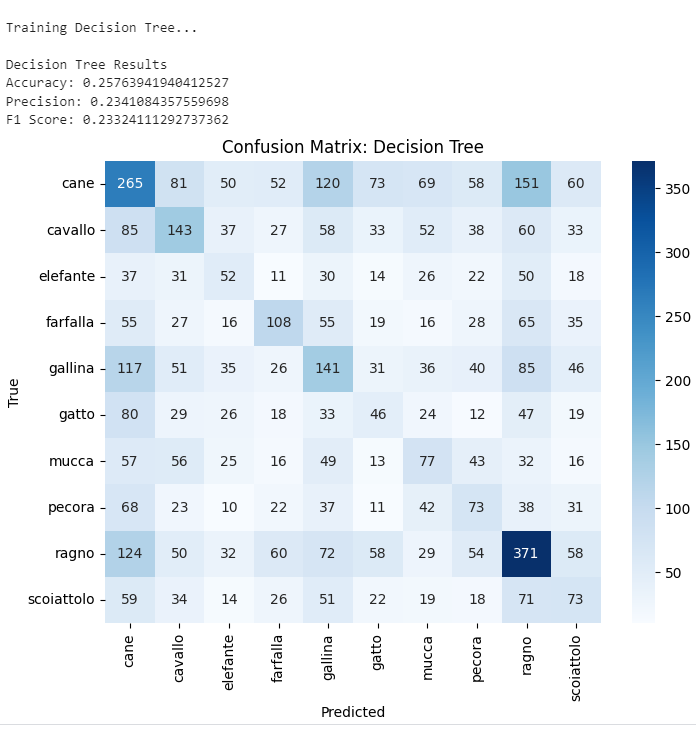
* **Convolutional Neural Network (CNN):**
  + Layers: Conv2D, MaxPooling2D, Flatten, Dense, Dropout
  + Optimizer: Adam
  + Loss: Sparse Categorical Crossentropy
  + Epochs: 10
* **Traditional ML Models:**
  + Logistic Regression
  + Decision Tree Classifier
  + Random Forest Classifier
  + Gradient Boosting Classifier
  + XGBoost Classifier

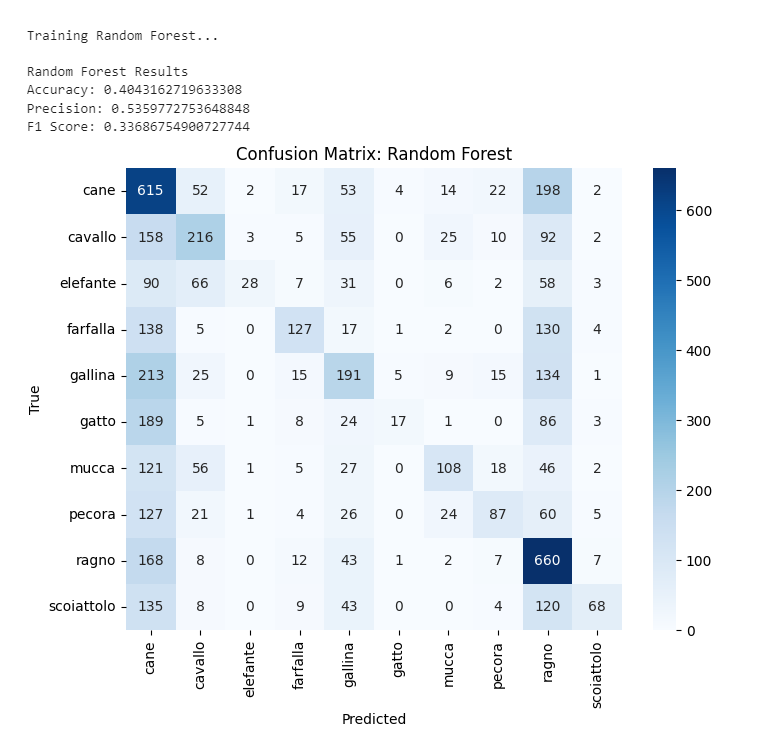
**5. Evaluation Metrics:** All models are evaluated using the following metrics:

* **Accuracy**: Correct predictions over total predictions.
* **Precision**: Macro average to handle multi-class classification.
* **F1 Score**: Harmonic mean of precision and recall.
* **Confusion Matrix**: Visual representation of prediction errors across classes.

**6. Results:**





**7. Visualizations:**

* **Bar Plot:** Comparing Accuracy, Precision, and F1 Score across models.
* **Heatmaps:** Confusion Matrices for each model.

**8. Conclusion:** CNNs outperform traditional ML models in accuracy and F1 score due to their ability to capture spatial hierarchies in images. However, traditional models like Random Forest and XGBoost still provide reasonable performance, especially when computation or interpretability is a concern. This study highlights the importance of selecting appropriate models based on task requirements and available resources.

**9. Future Work:**

* Explore data augmentation to improve CNN performance.
* Test models on more fine-grained datasets.
* Use transfer learning with pretrained networks like EfficientNet or ResNet for better feature extraction.

**10. References:**

* Animals10 Dataset: <https://www.kaggle.com/datasets/alessiocorrado99/animals10>
* TensorFlow Documentation
* Scikit-learn API Reference
* XGBoost Documentation