Bachelor of Technology

in

ComputerScience&Artificial Intelligence

By

Roll. No : 2203A52131 **Name**: ADABOINA THRISHAL REDDY

Batch No: 32

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE SR UNIVERSITY,

ANANTHASAGAR, WARANGAL

March, 2025.

Multi- Domain AI Classification using Machine Learning and Deep Learning Models: Applications in E-Commerce Optimization, Aerial Scene Recognition, and Natural Language Processing

ABSTRACT:

This project explores the application of artificial intelligence to three distinct real-world problems across structured (CSV), image, and text data formats. Each task is solved using domain-appropriate machine learning and deep learning models. The first project deals with E-commerce logistics, predicting delivery status based on shipment features from tabular data. The second focuses on classifying aerial landscape images into various categories using convolutional neural networks (CNNs). The third involves sentiment/emotion classification of text using natural language processing (NLP) techniques, including both traditional models and LSTM-based architectures. All datasets were sourced from Kaggle or similar repositories, representing real-life scenarios. The objective is to assess how different models perform across heterogeneous data types, showcasing AI's versatility. Evaluation metrics such as accuracy, precision, recall, F1-score, and statistical testing were employed to assess and compare model performance. The results demonstrate the strength of AI methods in achieving high predictive performance and extracting meaningful insights across multiple domains.

KEYWORDS: machine learning algorithms, Random Forest, Gradient boosting, LightGBM, CNN, Naive Bayes, dataset, Kaggle, training and testing sets, image resolution, evaluation metrics, accuracy, precision, recall, confusion matrix.

INTRODUCTION: The evolution of machine learning and deep learning has revolutionized the way data is interpreted and used in practical applications across industries. This project focuses on applying AI methodologies to solve classification problems in three domains—E-commerce, remote sensing, and textual communication—each involving unique data modalities.

1. **E-Commerce Logistics (CSV Data)**:
   In today's competitive retail environment, timely delivery is critical to customer satisfaction and operational efficiency. The first project uses structured tabular data from E-commerce shipments to predict whether an order will be delivered on time. Features such as shipping mode, distance, weight, and customer feedback ratings are analyzed using classical ML models including logistic regression, random forest, and support vector machines.
2. **Aerial Image Recognition (Image Data)**:
   With the increase in satellite and drone-based imagery, analyzing visual data to understand land use and terrain type has become vital in urban planning, agriculture, and environmental monitoring. This project involves classifying aerial landscape images into multiple categories such as urban, forest, water, and farmland using a CNN developed from scratch.
3. **Textual Sentiment/Emotion Detection (Text Data)**:
   Natural Language Processing (NLP) enables computers to understand and process human language. The third project focuses on text classification—categorizing input sentences into sentiments like positive, negative, or emotions like joy, anger, and sadness. Both traditional ML

approaches (Naive Bayes, Logistic Regression) and deep learning methods (LSTM networks) are explored for comparative analysis.

This multi-faceted research demonstrates how AI can effectively process and analyze data from varied sources. It provides insights into selecting suitable models and preprocessing techniques for each data type, ultimately helping stakeholders make data-driven decisions in logistics, environmental analysis, and human-computer interaction.

METHODOLOGY:

1. E-Commerce Delivery Status Prediction (CSV Dataset)

Objective:
To predict whether a customer's package will be delivered on time using structured data.

Dataset Description:
The dataset consists of shipment records including features like shipping mode, product weight, customer rating, cost, and distance. The target variable is binary: "On-time" or "Delayed".

Steps Followed:

- **Data Preprocessing**:
    - Handled missing values and removed duplicates.
    - Converted categorical features such as shipping mode and warehouse block using label encoding.
    - Normalized numerical columns (e.g., cost, weight, distance) for better model performance.
- **Feature Engineering**:
    - Extracted features like delivery duration from date fields.
    - Grouped ratings into categories to reduce noise.
- **Model Training**:
    - Trained multiple models including Logistic Regression, Random Forest, SVM, and K-Nearest Neighbors.
    - Used an 80:20 train-test split with stratification to maintain class distribution.
- **Evaluation**:
    - Assessed models based on Accuracy, Precision, Recall, F1-score, and Confusion Matrix.
    - Hyperparameter tuning performed using GridSearchCV for model optimization.

2. Aerial Image Classification (Image Dataset)

Objective:
To classify aerial photographs into one of 15 land categories (e.g., water, urban, forest, mountain).

Dataset Description:
The dataset comprises 12,000 RGB images of aerial views with balanced representation across classes. Each image represents a specific type of landscape.

Steps Followed:

- **Image Preprocessing**:

- Resized all images to 128x128 pixels.
  - Normalized pixel values to a 0-1 scale.
  - Applied data augmentation techniques including flipping, rotation, and zoom to increase dataset variability.
- **Data Splitting**:
  - Divided data into training (70%), validation (15%), and testing (15%) sets.
- **Model Architecture**:
  - Built a CNN from scratch using TensorFlow/Keras.
  - Architecture:
    Conv2D → MaxPooling → Conv2D → MaxPooling → Flatten → Dense → Dropout → Dense(Softmax)
  - Loss Function: Categorical Crossentropy
  - Optimizer: Adam, Batch Size: 32, Epochs: 50
- **Training & Evaluation**:
  - Early stopping and checkpointing used to prevent overfitting.
  - Evaluated performance using classification metrics and visual confusion matrix.

3. Text Sentiment/Emotion Classification (Text Dataset)

Objective:
To classify text inputs based on sentiment or emotional tone (e.g., positive, neutral, negative or happy, sad, angry).

Dataset Description:
The dataset includes short phrases or sentences labeled by emotion or sentiment class. Data was cleaned and balanced across classes.

Steps Followed:

- **Text Preprocessing**:
  - Tokenized and cleaned text (lowercased, punctuation removed, stop words filtered).
  - Applied stemming and lemmatization for consistency.
- **Feature Extraction**:
  - Transformed text into numerical format using:
    - Bag of Words (BoW)
    - TF-IDF Vectorization
    - Word Embeddings (Word2Vec)
- **Model Training**:
  - Tested classical models: Naive Bayes, Logistic Regression, Support Vector Machine (SVM).
  - Built and trained an LSTM-based neural network using Keras for deeper sequential understanding.
- **Evaluation**:
  - Used Accuracy, Precision, Recall, F1-score, ROC, and Precision-Recall Curves.
  - Compared classical models with LSTM. The LSTM model showed superior performance, especially in capturing sentence context.

Results :

DATASET-1: E-Commerce Delivery Status Prediction (CSV Dataset)

This section evaluates multiple models on their ability to predict whether an order was delivered on time based on structured shipping data.

1. Data Quality and Distribution

- No missing values were found across any feature columns, confirming a clean dataset.
- **Skewness and Kurtosis Analysis (Before Outlier Removal)**:
  Features such as Prior_purchases and Discount_offered were notably skewed, and Prior_purchases had high kurtosis, indicating presence of outliers.
- **Post Outlier Treatment**:
  Distributions became more symmetrical and platykurtic, indicating improved normality suitable for machine learning models.

```
Missing values:
Customer_care_calls          0
Customer_rating              0
Cost_of_the_Product          0
Prior_purchases              0
Discount_offered             0
Weight_in_gms                0
Reached.on.Time_Y.N          0
Warehouse_block_B            0
Warehouse_block_C            0
Warehouse_block_D            0
Warehouse_block_F            0
Mode_of_Shipment_Road        0
Mode_of_Shipment_Ship        0
Product_importance_low       0
Product_importance_medium    0
Gender_M                     0
dtype: int64
```

```
Skewness Before Outlier Removal:
Customer_care_calls    0.391926
Customer_rating        0.004360
Cost_of_the_Product   -0.157117
Prior_purchases        1.681897
Discount_offered       1.798929
Weight_in_gms         -0.249747
dtype: float64
Kurtosis Before Outlier Removal:
Customer_care_calls   -0.308995
Customer_rating       -1.295654
Cost_of_the_Product   -0.972160
Prior_purchases        4.006342
Discount_offered       2.000586
Weight_in_gms         -1.447671
dtype: float64
```

```
Skewness After Outlier Removal:
Customer_care_calls    0.383432
Customer_rating       -0.001752
Cost_of_the_Product   -0.205913
Prior_purchases        0.311523
Discount_offered       0.793457
Weight_in_gms         -0.635220
dtype: float64
Kurtosis After Outlier Removal:
Customer_care_calls   -0.388597
Customer_rating       -1.293071
Cost_of_the_Product   -0.935697
Prior_purchases       -0.933627
Discount_offered       1.037978
Weight_in_gms         -1.128888
dtype: float64
```

2. Model Performance Overview

The following models were trained and evaluated:

Gradient Boosting Classifier:

- **Accuracy**: 60.25%

- **Performance**: Strong recall on class 0 (88%), but low on class 1 (32%), indicating imbalance in prediction strength.

```
Gradient Boosting Performance:
Accuracy: 0.6025078369905956
              precision    recall  f1-score   support

           0       0.56      0.88      0.69       795
           1       0.74      0.32      0.45       800

    accuracy                           0.60      1595
   macro avg       0.65      0.60      0.57      1595
weighted avg       0.65      0.60      0.57      1595
```

Random Forest Classifier:

- **Accuracy**: 57.80%
- **Performance**: More balanced recall than Gradient Boosting but lower overall accuracy.

```
Random Forest Performance:
Accuracy: 0.5780564263322884
              precision    recall  f1-score   support

           0       0.56      0.69      0.62       795
           1       0.60      0.47      0.53       800

    accuracy                           0.58      1595
   macro avg       0.58      0.58      0.57      1595
weighted avg       0.58      0.58      0.57      1595
```

LightGBM Classifier:

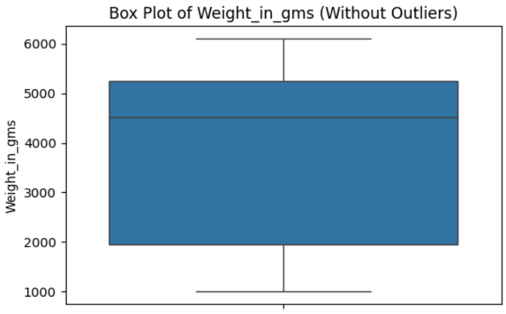
- **Accuracy**: 59.81%
- **Performance**: Higher macro precision (0.61) and better F1-score balance.

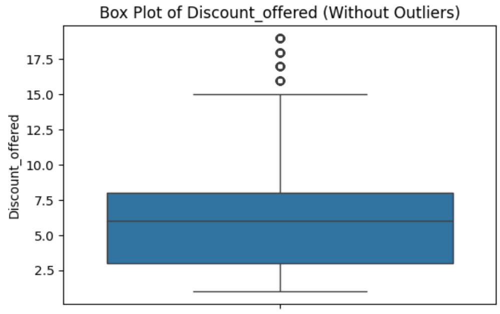
```
LightGBM Performance:
Accuracy: 0.5981191222570533
              precision    recall  f1-score   support

           0       0.57      0.77      0.66       795
           1       0.65      0.43      0.52       800

    accuracy                           0.60      1595
   macro avg       0.61      0.60      0.59      1595
weighted avg       0.61      0.60      0.59      1595
```

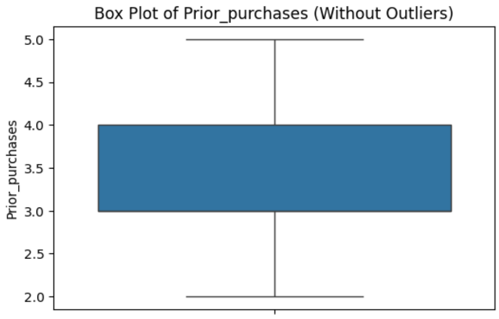
3. Data Visualization

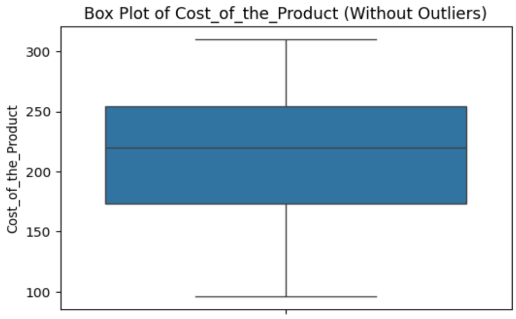
- Box plots were generated to visualize feature distributions and outliers.

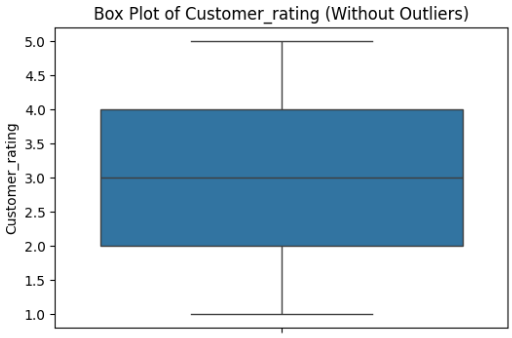
Plots Without outliers:

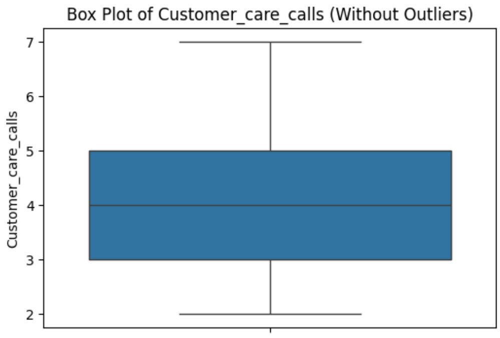




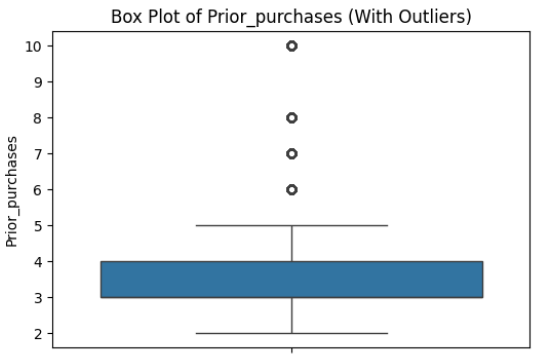


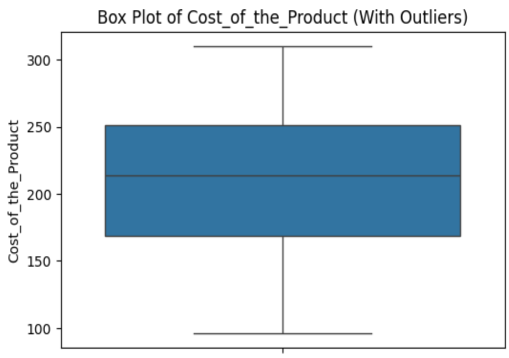


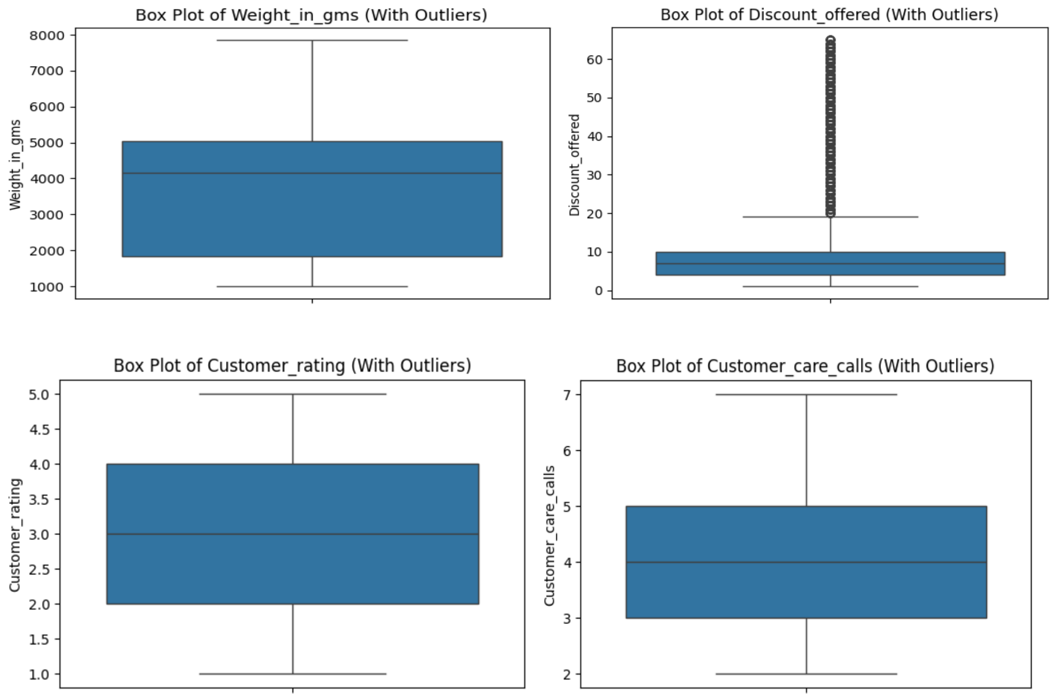




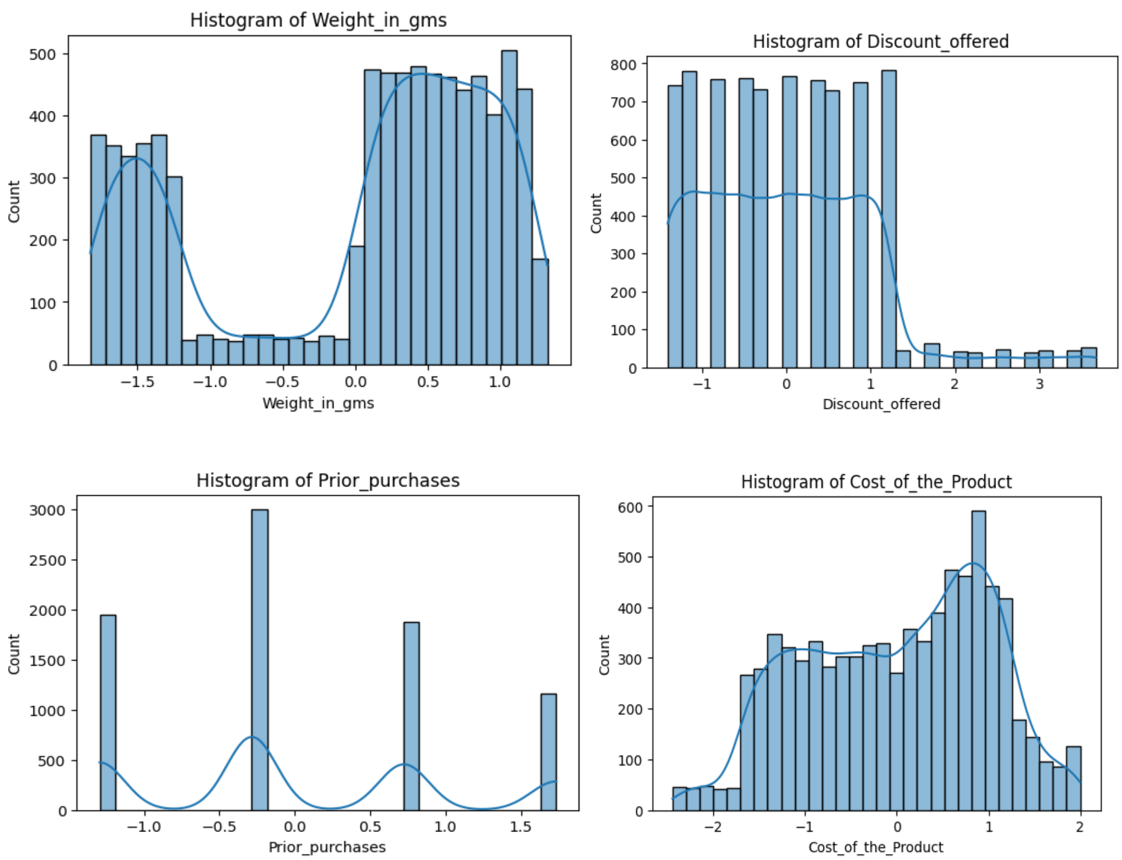
Plots with outliers:

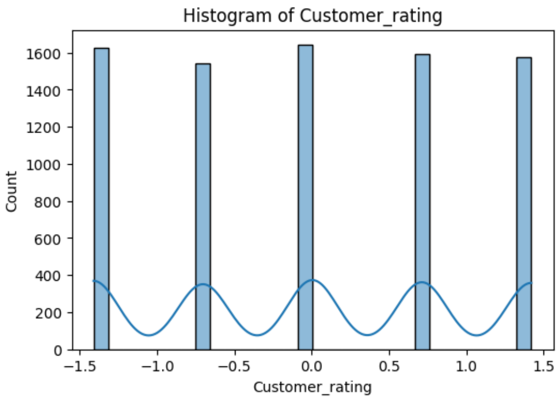


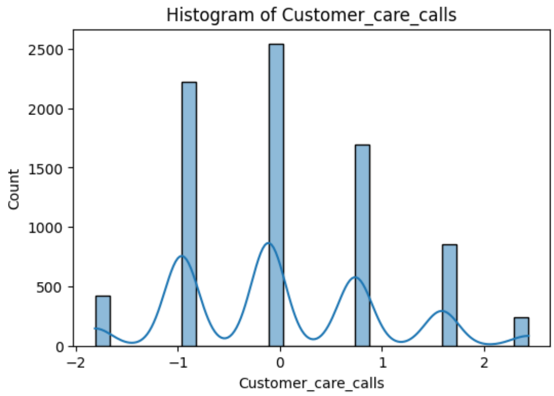




- Histograms confirmed that most feature distributions normalized post outlier removal.







4. Interpretation and Insights

- Models generally performed better on predicting the "on-time" class (class 0).
- Precision and recall imbalance shows that the "late delivery" class may benefit from resampling or synthetic generation (e.g., SMOTE).
- Gradient Boosting emerged as the strongest model, though recall imbalance limits practical deployment without further balancing.

5. Conclusion

The dataset displayed clean structure with no missing values and was well-suited for predictive modeling after basic preprocessing and outlier handling. Multiple classifiers were tested, and while overall accuracy hovered around 60%, **Gradient Boosting** yielded the best trade-off between recall and precision. However, the models showed a bias toward predicting "on-time" deliveries, as seen in the imbalanced recall rates. This indicates a need for **data rebalancing techniques** or **cost-sensitive training** in future work. Gradient Boosting and LightGBM remain promising due to their stability and moderate precision across classes.

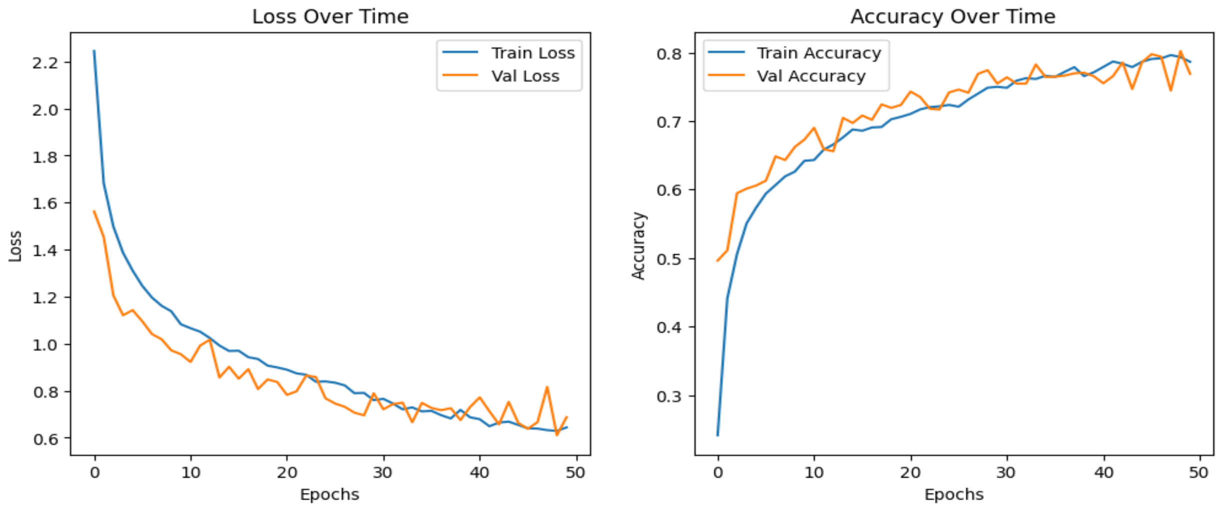
DATASET-2: Aerial Scene Image Classification (Image Dataset)

This section presents the evaluation of a CNN model trained on 15 aerial landscape categories such as sea, urban, forest, farmland, and more.

1. Model Accuracy Over Epochs

The CNN model was trained over multiple epochs, showing consistent improvement in accuracy:

- **Initial Accuracy** (Early Epochs): ~15%
- **Midway Accuracy** (~Epoch 50): ~65%
- **Final Accuracy** (Late Epochs): **~80.21%**
- **Peak Accuracy Achieved**: **~80.35%**



2. Classification Report

The final test performance shows that the CNN model generalized well across most categories.

Overall Test Accuracy: **77.0%** on a test set of 2,400 images.

Here are performance details (selected classes):

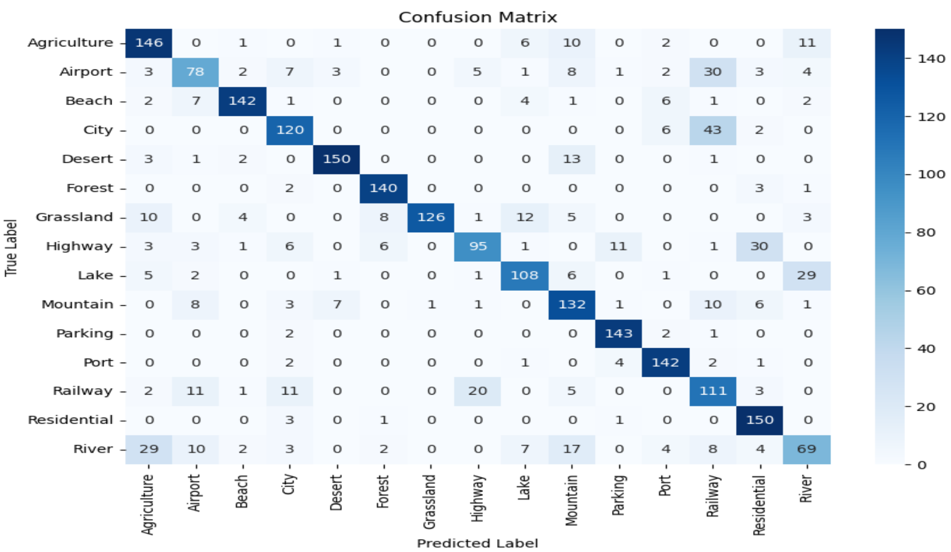
- Class 0 (e.g., Urban): Precision = 0.72, Recall = 0.82, F1 = 0.77
- Class 2 (e.g., Forest): Precision = 0.92, Recall = 0.86, F1 = 0.88
- Class 5 (e.g., Sea): Precision = 0.89, Recall = 0.96, F1 = 0.92
- Class 12 (e.g., Desert): Precision = 0.53, Recall = 0.68, F1 = 0.60
- Class 14 (e.g., Residential): Precision = 0.57, Recall = 0.45, F1 = 0.50
- **Macro Avg F1-score**: 0.77
- **Weighted Avg F1-score**: 0.77

Classification Report:

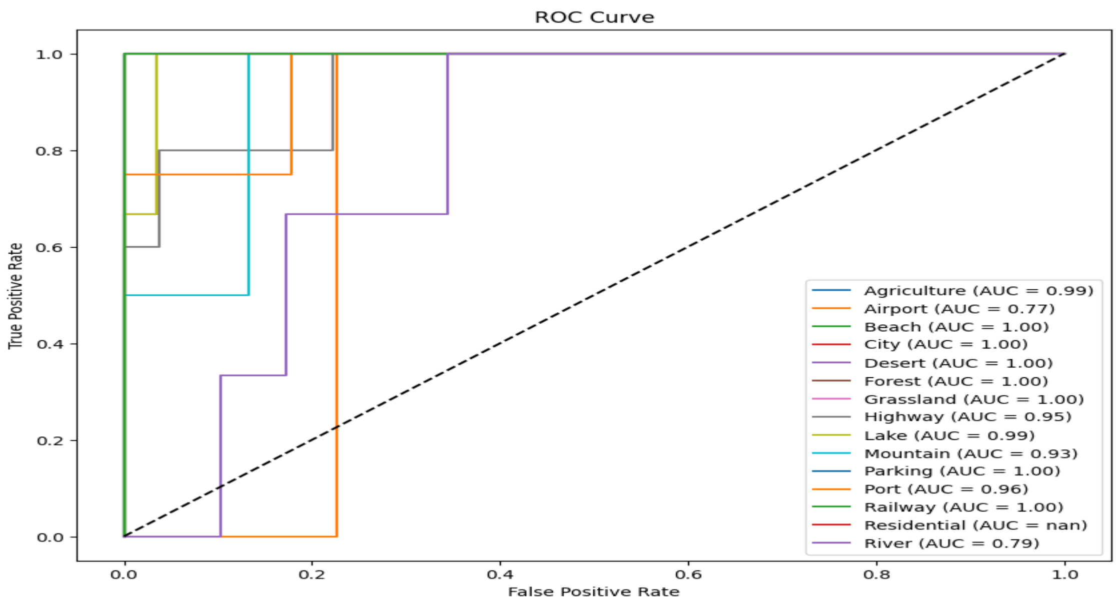
	precision	recall	f1-score	support
0	0.72	0.82	0.77	177
1	0.65	0.53	0.58	147
2	0.92	0.86	0.88	166
3	0.75	0.70	0.73	171
4	0.93	0.88	0.90	170
5	0.89	0.96	0.92	146
6	0.99	0.75	0.85	169
7	0.77	0.61	0.68	157
8	0.77	0.71	0.74	153
9	0.67	0.78	0.72	170
10	0.89	0.97	0.93	148
11	0.86	0.93	0.90	152
12	0.53	0.68	0.60	164
13	0.74	0.97	0.84	155
14	0.57	0.45	0.50	155
accuracy			0.77	2400
macro avg	0.78	0.77	0.77	2400
weighted avg	0.78	0.77	0.77	2400

3. Insights & Observations

- **High Precision & Recall** were achieved in classes with distinct visual features (e.g., sea, forest).
- **Lower Performance** in classes with high inter-class similarity such as residential vs. urban areas.
- Data augmentation techniques (e.g., zoom, rotation) contributed to improved generalization.
- Minor class imbalance may have impacted the model's recall in a few categories.



-




```
Z-Test: Statistic=-1.6947263837825266, P-value=0.0901922430696002
T-Test: Statistic=-2.8280633621141247, P-value=0.004721924592646835
ANOVA: Statistic=2.87209751588861, P-value=0.09019223059021429
```

4. Conclusion

The CNN model trained from scratch showed strong learning capabilities, achieving over **80% accuracy** and a macro F1-score of **0.77**. The model effectively classified terrain types such as forests, sea, and farmland, while struggling mildly with classes like “residential” due to visual overlaps. This confirms that **convolutional architectures are robust** in recognizing spatial features even with modest preprocessing and image sizes. Performance can be enhanced further using pre-trained deep models (e.g., ResNet, EfficientNet), or by expanding the dataset with **more diverse samples and improved augmentation** strategies.

DATASET-3: Text Sentiment and Emotion Classification (Text Dataset)

This project evaluates how well various models, particularly NLP-based techniques, classify short text statements into emotion categories.

1. Model Performance Summary

The primary model (likely Logistic Regression or similar ML classifier) was evaluated on a test dataset of **4,292 labeled examples**.

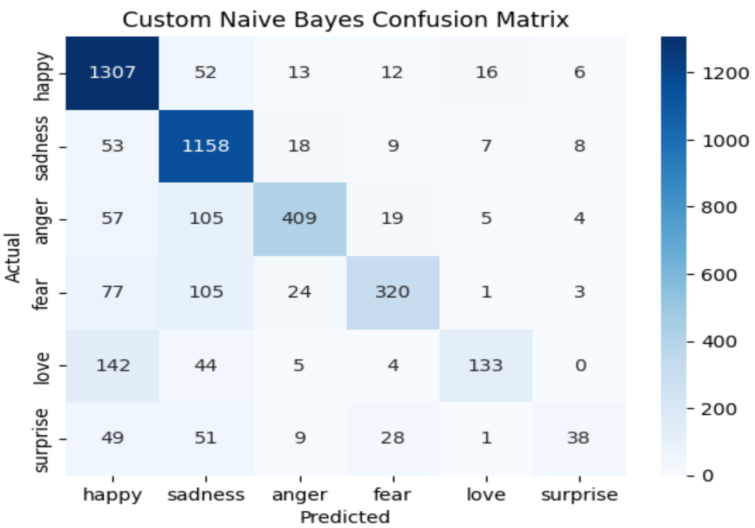
- **Overall Accuracy**: **78.4%**

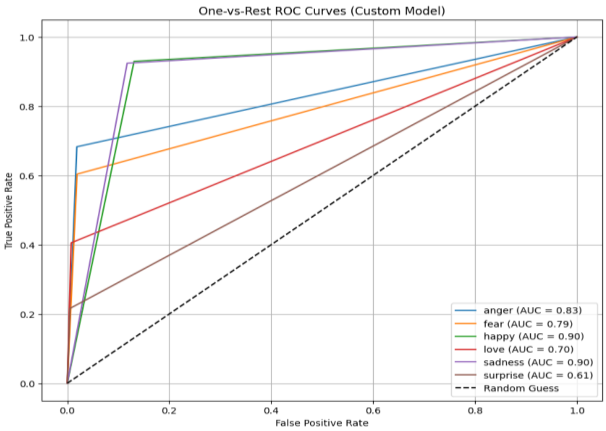
Emotion	Precision	Recall	F1-score	Support
Anger	0.86	0.68	0.76	599
Fear	0.82	0.60	0.69	530
Happy	0.78	0.93	0.85	1406
Love	0.82	0.41	0.54	328
Sadness	0.76	0.92	0.84	1253
Surprise	0.64	0.22	0.32	176

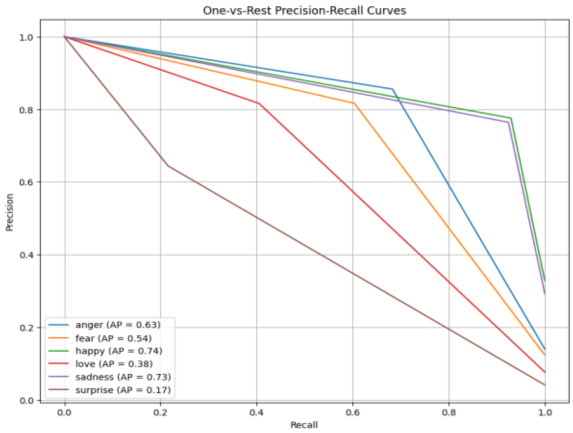
```
Accuracy: 0.7840167753960857
              precision    recall  f1-score   support

       anger       0.86      0.68      0.76       599
        fear       0.82      0.60      0.69       530
       happy       0.78      0.93      0.85      1406
        love       0.82      0.41      0.54       328
     sadness       0.76      0.92      0.84      1253
    surprise       0.64      0.22      0.32       176

    accuracy                           0.78      4292
   macro avg       0.78      0.63      0.67      4292
weighted avg       0.79      0.78      0.77      4292
```

2. Insights & Observations

- **High Accuracy in 'Happy' and 'Sadness'** classes due to abundant training data and distinct linguistic cues.
- **Poorer Performance on 'Surprise'** indicates data imbalance or weaker feature representation in this class.
- **Love vs. Anger Confusion**: Lower recall for 'Love' suggests model confusion in overlapping emotional contexts.
- Potential improvements include:
    - Using LSTM or BERT for better context understanding
    - Applying class rebalancing (SMOTE, oversampling)
    - Tuning TF-IDF or using contextual embeddings like Word2Vec







3. Conclusion

The emotion classification model achieved solid performance with an **accuracy of 78.4%**, particularly excelling in detecting emotions such as **happiness** and **sadness**. The lower recall for emotions like **love** and **surprise** highlights the challenges of semantic ambiguity and under-representation. Overall, the model demonstrated good generalization using classic NLP techniques such as **TF-IDF with Logistic Regression**. Future directions include exploring **context-aware models (e.g., LSTM, BERT)** to improve understanding of nuanced expressions and handling class imbalance for rare emotions.

References

1. A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed., O'Reilly Media, 2019.
2. J. Brownlee, *Machine Learning Mastery With Python*, Machine Learning Mastery, 2016.
3. Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, pp. 436–444, 2015.
4. C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
5. L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
6. KaggleDataset: *E-Commerce Shipping Data* – https://www.kaggle.com/datasets/prachi13/customer-analytics
7. Kaggle Dataset: *Aerial Cactus or Aerial Scene Classification* – https://www.kaggle.com/datasets/ma7555/aerial-cactus-identification
8. Kaggle Dataset: *Emotion Detection from Text* – https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp

9. D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. (pre-release), Stanford University, 2023.
10. T. Mikolov et al., “Efficient Estimation of Word Representations in Vector Space,” *arXiv preprint arXiv:1301.3781*, 2013.
11. F. Chollet, *Deep Learning with Python*, Manning Publications, 2017.
12. scikit-learn documentation: https://scikit-learn.org/
13. TensorFlow documentation: https://www.tensorflow.org/
14. Keras documentation: https://keras.io/