A Capstone Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

by

2203A52193 THADISHETTY RAHUL

Under the guidance of

Dr.Ramesh Dadi

Assistant Professor, School of CS&AI.

SR University, Ananthsagar,Warangal,Telagnana-506371

CONTENTS

DATASET

Project-1: Exam Performance Dataset

The exams.csv dataset appears to contain information related to student exams. This likely includes features such as student IDs, scores in different subjects (e.g., math, science, language), and possibly demographic information or study habits. The dataset can be used to analyze student performance, identify factors influencing academic success, and potentially predict future outcomes. By exploring the correlations between different variables, educators and researchers can gain insights into learning patterns, evaluate teaching methods, and develop interventions to improve student achievement. This data is valuable for educational data mining, performance analysis, and the development of predictive models in the academic domain.

Project-2: Soil types Dataset

The Soil Types Dataset, sourced from Kaggle, contains a variety of features related to soil properties. These attributes likely include physical and chemical characteristics such as nutrient levels, pH values, texture, and possibly geographical information. The dataset is designed to allow for the analysis and classification of different soil types based on these features. By examining the relationships between these variables, researchers and practitioners can gain insights into soil health, agricultural suitability, and environmental factors. This data can be instrumental in developing models for soil classification, predicting soil quality, and supporting precision agriculture initiatives. The dataset offers a valuable resource for understanding soil science and its applications in various fields.

Project-3: Hate speech and Offensive Language

The Hate Speech and Offensive Language Detection dataset, available on Kaggle, comprises text data labeled to identify instances of hate speech, offensive language, and neither. This dataset is specifically curated to train machine learning models capable of automatically detecting toxic language in online communication. The text samples within the dataset vary in content, style, and the presence or absence of harmful speech. By analyzing the linguistic patterns and contextual cues within this data, models can be developed to filter and moderate online platforms, identify cyberbullying, and understand the nuances of harmful language. This project is crucial for fostering safer online environments and advancing natural language processing techniques for social good.

METHODOLOGY

Project 1 : Exam Performance

Data Loading and Initial Exploration: The exams.csv dataset was loaded into a Pandas DataFrame. The initial steps involved displaying the first few rows to understand the data structure and checking for missing values across the different columns. Categorical features within the dataset were then converted into numerical representations using Label Encoding to prepare them for machine learning models.

Feature Definition and Initial Data Characteristics: The 'math score' column was identified as the target variable for prediction. The remaining columns were designated as features. Before any data cleaning, the skewness and kurtosis of the numerical features were calculated to understand the shape and tailedness of their distributions. Histograms and boxplots were generated for each numerical feature to visually inspect their distributions and identify potential outliers.

Outlier Removal using IQR: To address potential outliers in the numerical features, the Interquartile Range (IQR) method was employed. Data points falling below (Q1 - 1.5 \times IQR) or above (Q3 + 1.5 \times IQR) for any of the numerical columns were considered outliers and subsequently removed from the dataset to create a cleaner dataset (df_cleaned). Following the outlier removal, histograms and boxplots were regenerated to visualize the impact of this process on the distributions. The skewness and kurtosis were also recalculated on the cleaned data to observe any changes in the distributional characteristics.

Data Preprocessing: After handling missing values and outliers, the data was preprocessed to prepare it for machine learning models. Numerical features were scaled using StandardScaler to ensure they have zero mean and unit variance. Categorical features were encoded using One-Hot Encoding to convert them into a binary vector representation, with the first category dropped to avoid multicollinearity. These preprocessing steps were implemented using a ColumnTransformer to apply the transformations to the appropriate columns.

Model Training and Evaluation: Three regression models were selected for predicting the 'math score': Linear Regression, Random Forest Regressor, and XGBoost Regressor. The preprocessed data was split into training and testing sets. Each model was trained on the training data using a Pipeline that combined the preprocessing steps with the model. The performance of each trained model was then evaluated on the test set using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R^2 (coefficient of determination) score. Scatter plots of the actual vs. predicted 'math score' were also generated for each model to visually assess their predictive performance..

Project 2: Soil Types

Data Collection and Preprocessing: The image dataset, organized into training and testing directories, was loaded using ImageDataGenerator from Keras. Images were resized to a consistent resolution of 150x150 pixels. Pixel values were normalized to the [0, 1] range by rescaling. To enhance the model's ability to generalize, image augmentation techniques such as rotation, width and height shifts, and horizontal flipping were applied to the training data. The dataset was split into training and validation sets (20% of the training data) using the validation_split parameter in ImageDataGenerator. The data generators were configured to produce batches of 32 images, and the class mode was set to 'categorical', implying a multi-class classification problem.

Model Structure: A Convolutional Neural Network (CNN) was designed for the image classification task using the Keras Functional API. The model architecture consists of a series of convolutional layers with ReLU activation functions to learn hierarchical features from the images. Each pair of convolutional layers is followed by a MaxPooling layer to reduce the spatial dimensions and increase the robustness to translations. The network includes three such convolutional blocks with an increasing number of filters (32, 64, and 128). The output from the convolutional layers is flattened and passed through two dense (fully connected) layers with ReLU activation (256 and 128 units). The final layer is a dense layer with a number of units equal to the number of classes in the dataset, using a softmax activation function to output probability distributions over the classes. The model was compiled using the Adam optimizer and categorical cross-entropy as the loss function, with accuracy as the evaluation metric.

Model Training: The CNN model was trained for 5 epochs using the training data generated by train_generator, with the val_generator providing validation data to monitor performance on unseen data during training. The fit method of the Keras model was used to perform the training process.

Evaluation Metrics: After training, the model's performance was evaluated on the test dataset using the evaluate method to obtain the test loss and accuracy. Furthermore, predictions were made on the test data to generate a classification report, which includes precision, recall, F1-score, and support for each class. A confusion matrix was also computed and visualized using a heatmap to provide a detailed breakdown of the model's predictions against the true labels. To further assess the model's ability to discriminate between classes, a Receiver Operating Characteristic (ROC) curve was plotted for each class, along with the Area Under the Curve (AUC) score.

Visualizations: During and after training, several visualizations were generated to understand the model's behavior and performance. Plots of training and validation loss and accuracy over the epochs were displayed to assess the model's convergence and identify potential overfitting. Random samples of training images were displayed with their corresponding class labels to provide a visual understanding of the data.

Project 3 : Hate Speech and Offensive Language Detection

Dataset Preparation: The dataset, containing tweets labeled into three categories: "hate speech," "offensive language," and "neither," was loaded into a Pandas DataFrame. The 'tweet' column was identified as the text data for analysis, and the 'class' column contained the corresponding labels. The text data was converted to string format. The labels were then one-hot encoded to be suitable for the neural network model. The dataset was split into training and testing sets, with 20% of the data reserved for testing.

Feature Extraction: The text data (tweets) was processed using the Keras Tokenizer. This involved converting the text into sequences of integers, where each integer represents a word. A vocabulary size of 10,000 was set, with out-of-vocabulary words being replaced by an "<OOV>" token. The tokenizer was fitted on the training text data, and then both the training and testing text data were converted into sequences. To ensure uniformity in input length for the LSTM model, the sequences were padded to a maximum length of 100, with padding applied at the end of the sequences.

Model Architecture: A sequential neural network model was constructed using Keras. The architecture begins with an Embedding layer, which transforms the integer sequences into dense vector representations of the words. This embedding layer has an input dimension of 10,000 (the vocabulary size) and an output dimension of 128 (the embedding vector size). Following the embedding layer is an LSTM (Long Short-Term Memory) layer with 64 units to capture the sequential dependencies in the text. Dropout (0.2) and recurrent dropout (0.2) were applied to the LSTM layer to mitigate overfitting. The final layer is a dense layer with a number of units equal to the number of classes (3) and a softmax activation function, which outputs the probability distribution over the three classes. The model was compiled using the Adam optimizer and categorical cross-entropy as the loss function, with accuracy as the primary evaluation metric.

Model Training: The model was trained for 5 epochs with a batch size of 32. A steps_per_epoch of 20 was set, which determines the number of batches processed in each epoch. A validation split was used during training, with the test set serving as the validation data to monitor the model's performance on unseen data and prevent overfitting.

Performance Evaluation: After training, the model's performance was evaluated on the test set. The predicted probabilities for each class were obtained, and the class with the highest probability was considered the predicted class. A classification report was generated, which includes precision, recall, F1-score, and support for each class ("Hate Speech," "Offensive Language," "Neither"). Overall accuracy, precision (weighted), recall (weighted), and F1-score (weighted) were also calculated. A confusion matrix was computed to visualize the distribution of true and predicted labels. Furthermore, a Receiver Operating Characteristic (ROC) curve

was plotted for each of the three classes, along with the Area Under the Curve (AUC) score, to assess the model's ability to discriminate between the classes.

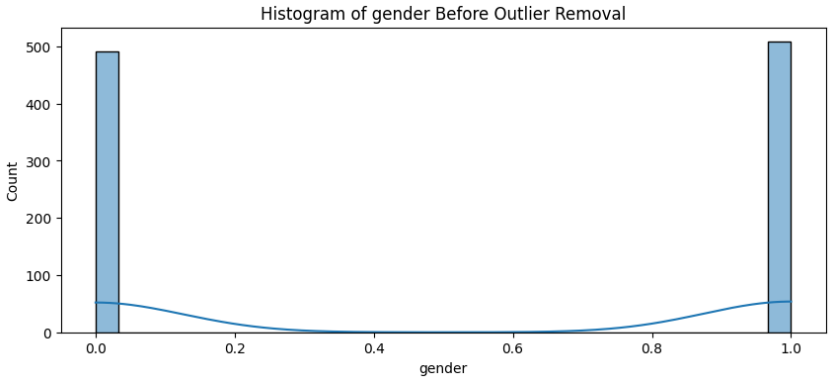
Visualizations: Several key visualizations were generated to provide insights into the model's training and performance. Plots of training and validation accuracy and loss over the epochs were displayed to assess the model's learning progress and identify potential overfitting. The confusion matrix was visualized as a heatmap to show the counts of true positives, true negatives, false positives, and false negatives for each class. ROC curves for each class were plotted to evaluate the model's performance at different classification thresholds.

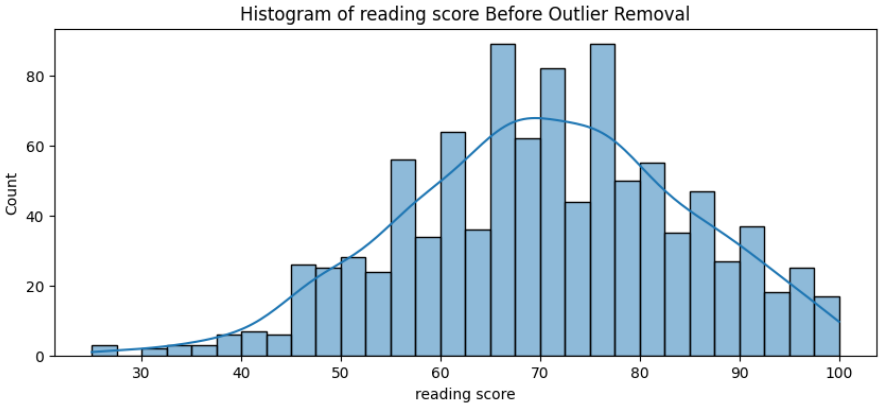
This structured approach outlines the process of building and evaluating an LSTM-based model for multi-class text classification, specifically for detecting hate speech and offensive language in tweets, with a focus on data preprocessing, model architecture, training, and a comprehensive evaluation using various metrics and visualizations.

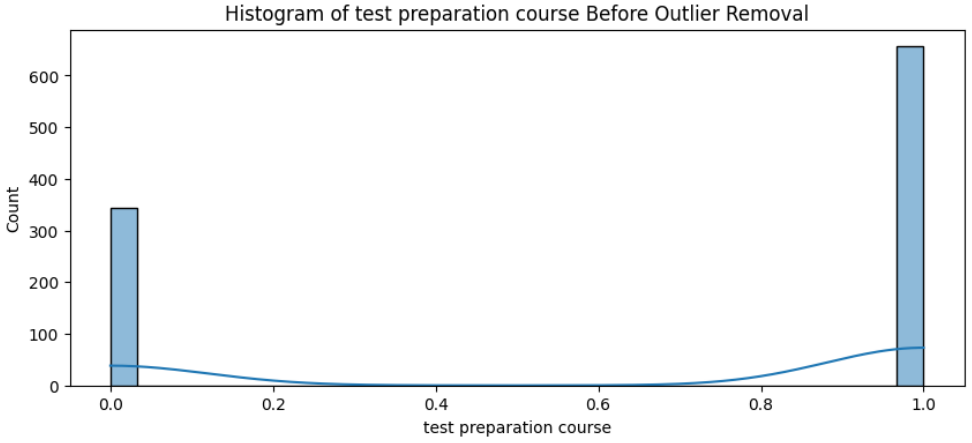
RESULTS

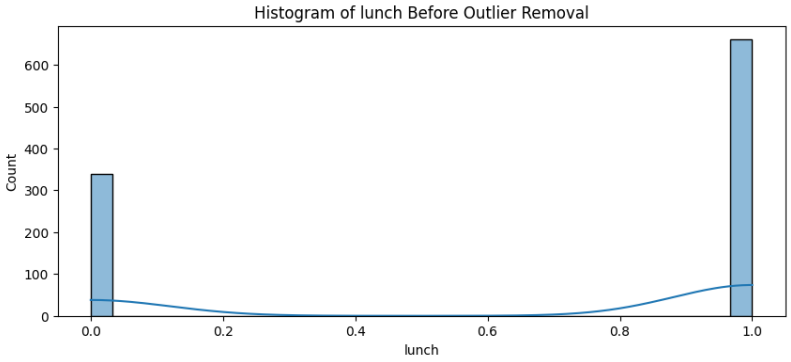
PROJECT-1

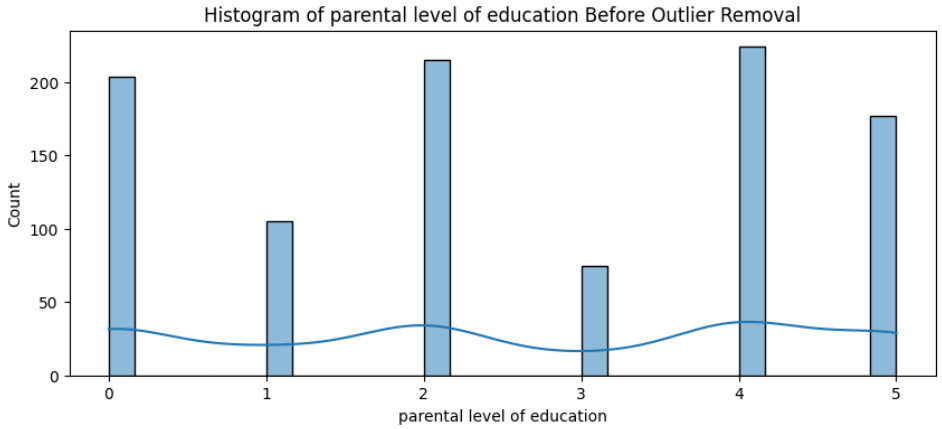
HISTOGRAMS

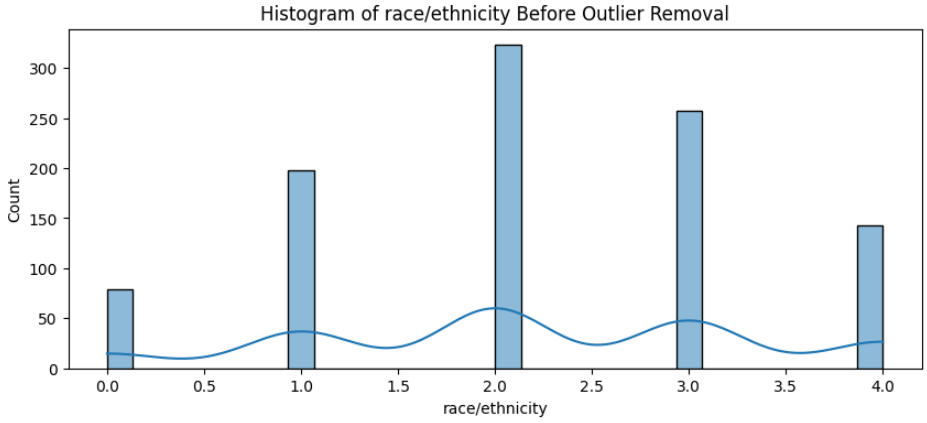


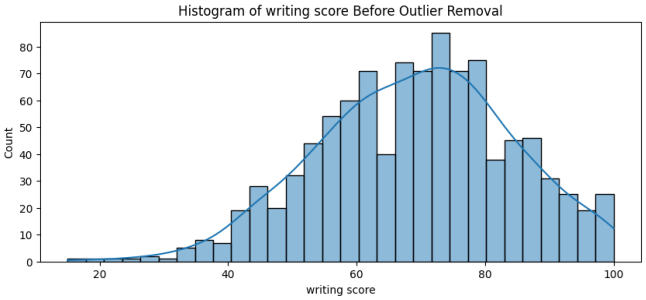


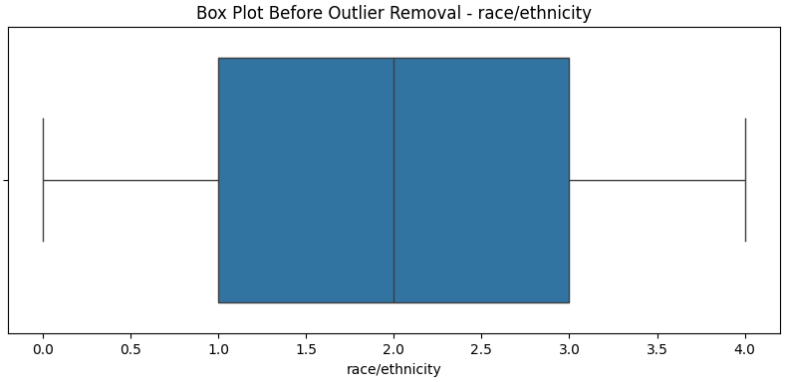


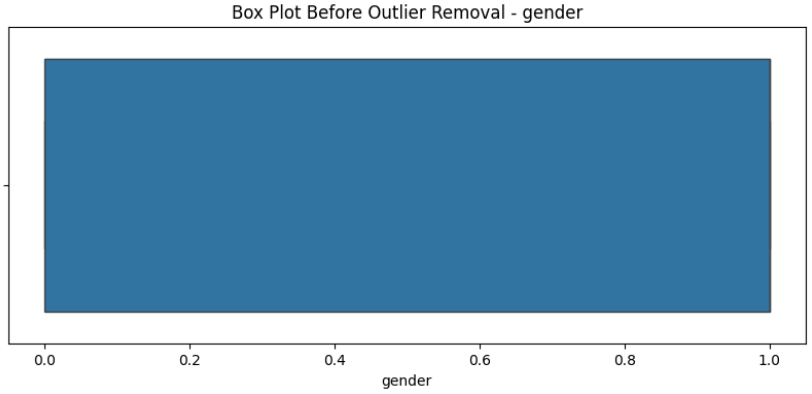

Histogram of lunch Before Outlier Removal
Count
lunch

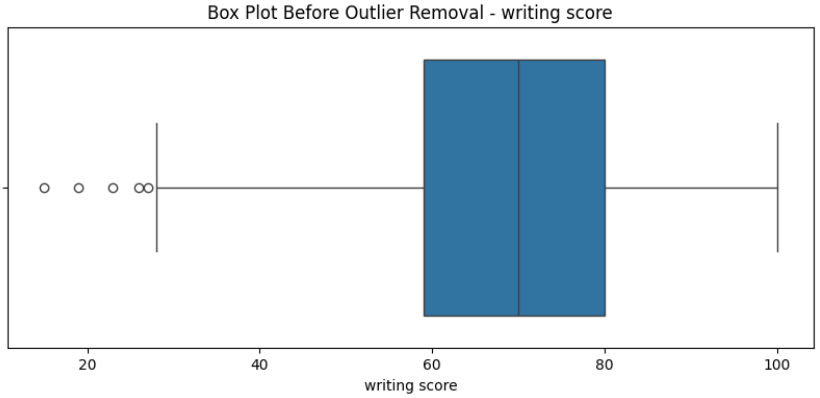

Histogram of parental level of education Before Outlier Removal
Count
parental level of education

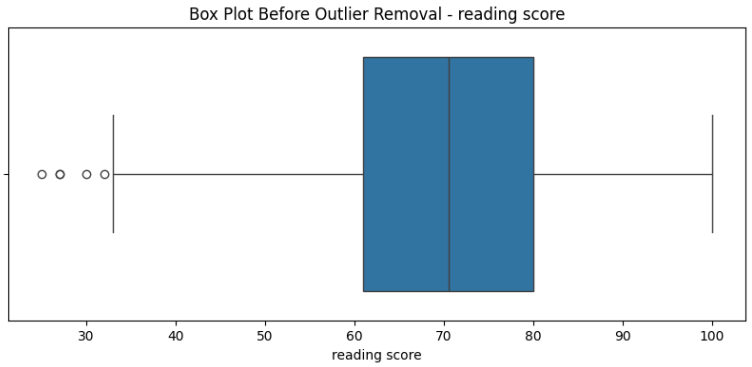

Histogram of race/ethnicity Before Outlier Removal
Count
race/ethnicity

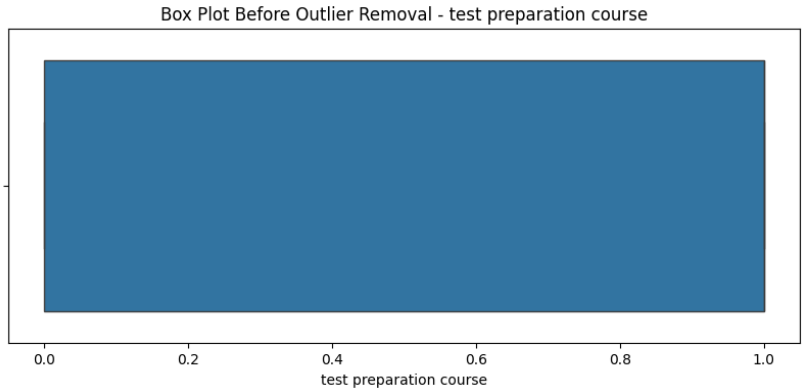

Histogram of writing score Before Outlier Removal
Count
writing score

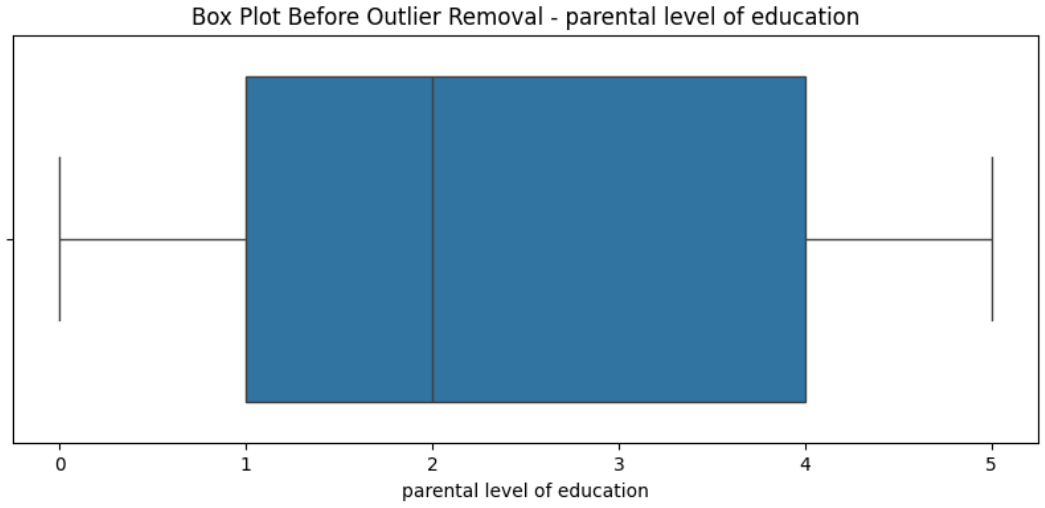

Box Plot Before Outlier Removal - race/ethnicity
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
race/ethnicity


Box Plot Before Outlier Removal - gender
0.0
0.2
0.4
0.6
0.8
1.0
gender


Box Plot Before Outlier Removal - writing score
20
40
60
80
100
writing score

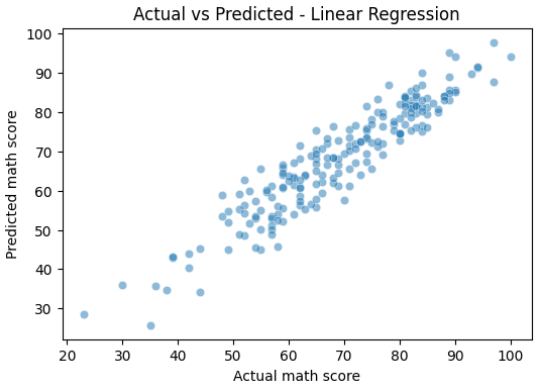

Box Plot Before Outlier Removal - reading score
30
40
50
60
70
80
90
100
reading score

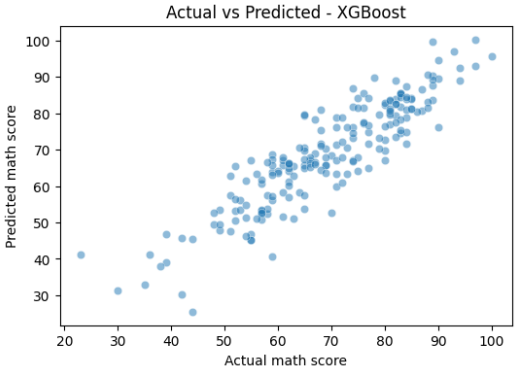


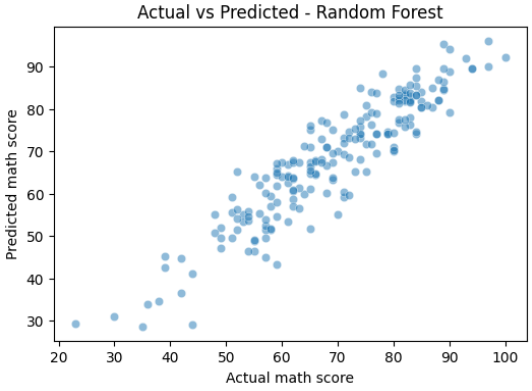



```
Skewness after outlier removal:
 gender                        -0.028214
race/ethnicity                 -0.115988
parental level of education    -0.079260
lunch                          -0.681096
test preparation course        -0.648099
reading score                  -0.091080
writing score                  -0.095260
dtype: float64
Kurtosis after outlier removal:
 gender                        -2.003239
race/ethnicity                 -0.755227
parental level of education    -1.374638
lunch                          -1.539209
test preparation course        -1.583157
reading score                  -0.463348
writing score                  -0.477142
dtype: float64
```

SCATTERPLOT







Model Performance After Outlier Removal:

	MAE	MSE	RMSE	R2 Score
Linear Regression	4.156556	25.312061	5.031109	0.873210
Random Forest	4.363337	30.424632	5.515853	0.847601
XGBoost	4.993756	41.466553	6.439453	0.792291

Data Distribution and Model Performance:

The distribution of the text data in the tweet dataset was implicitly handled through the tokenization and embedding process. Further analysis of the raw text's skewness and kurtosis isn't directly applicable in the same way as numerical features. However, the class distribution in the dataset (as reflected in the support values of the classification report) provides insights into the balance or imbalance of the categories.

In terms of model performance:

- The LSTM model achieved an **accuracy of 0.8835**, indicating that it correctly classified approximately 88.35% of the tweets into their respective categories (hate speech, offensive language, or neither).
- The **weighted precision was 0.8852**, suggesting that when considering all classes proportionally, about 88.52% of the instances predicted as a certain class were actually of that class.
- The **weighted recall was 0.8835**, indicating that the model correctly identified approximately 88.35% of all actual instances of each class.

- The **weighted F1-score was 0.8819**, which provides a balanced measure of the precision and recall, suggesting a good overall performance of the model across all classes.

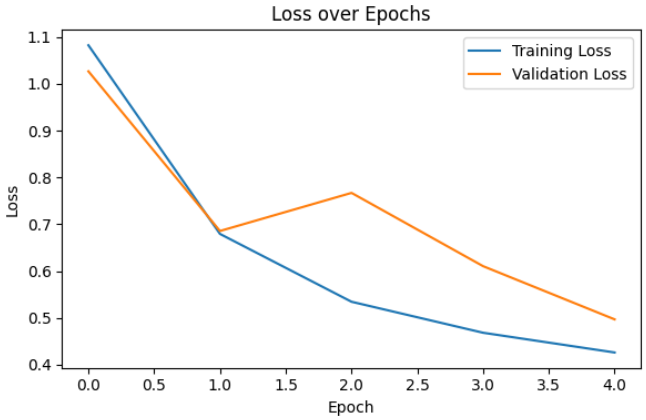
The classification report further details the performance for each individual class. The ROC curves and AUC scores for each class provide insights into the model's ability to distinguish between the different categories at various thresholds. The confusion matrix visually represents the model's classification accuracy for each class and highlights any patterns of misclassification.

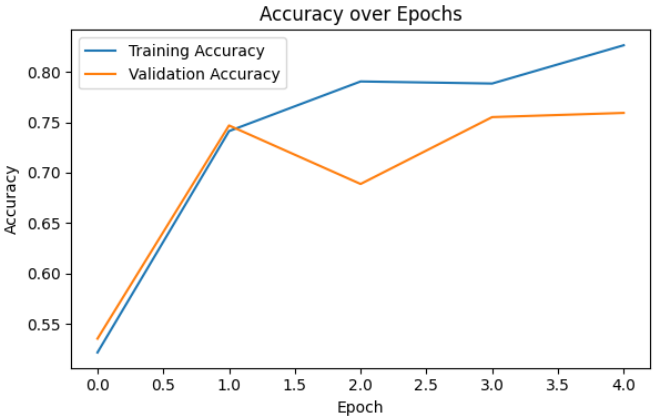
Overall, the LSTM model demonstrates a strong ability to classify tweets into the "hate speech," "offensive language," and "neither" categories, as indicated by the high accuracy and balanced precision and recall scores. The detailed classification report and confusion matrix offer a more granular understanding of the model's performance on each specific class.

PROJECT-2

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 150, 150, 3)	0
conv2d_12 (Conv2D)	(None, 150, 150, 32)	896
conv2d_13 (Conv2D)	(None, 150, 150, 32)	9,248
max_pooling2d_9 (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_14 (Conv2D)	(None, 75, 75, 64)	18,496
conv2d_15 (Conv2D)	(None, 75, 75, 64)	36,928
max_pooling2d_10 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_16 (Conv2D)	(None, 37, 37, 128)	73,856
conv2d_17 (Conv2D)	(None, 37, 37, 128)	147,584
max_pooling2d_11 (MaxPooling2D)	(None, 18, 18, 128)	0
flatten_3 (Flatten)	(None, 41472)	0
dense_7 (Dense)	(None, 256)	10,617,088
dense_8 (Dense)	(None, 128)	32,896
dense_9 (Dense)	(None, 4)	516

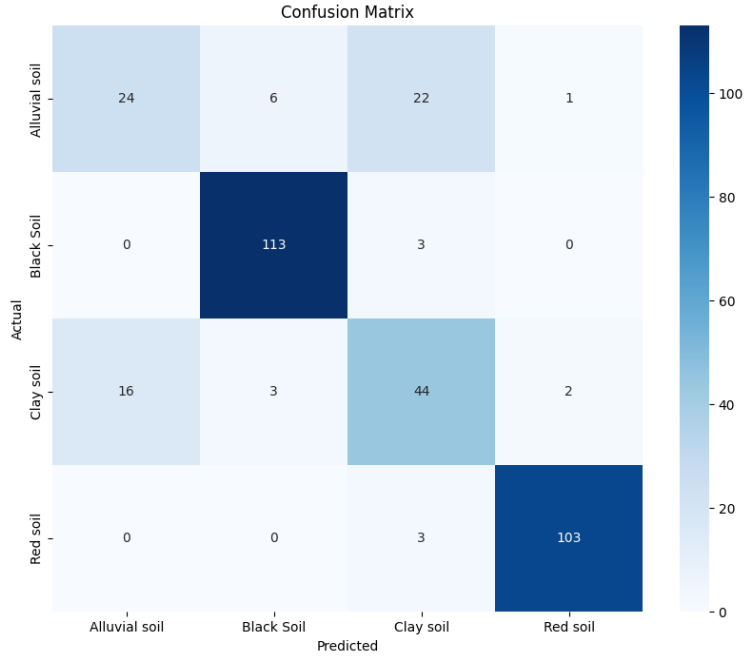
Total params: 10,937,508 (41.72 MB)
Trainable params: 10,937,508 (41.72 MB)
Non-trainable params: 0 (0.00 B)

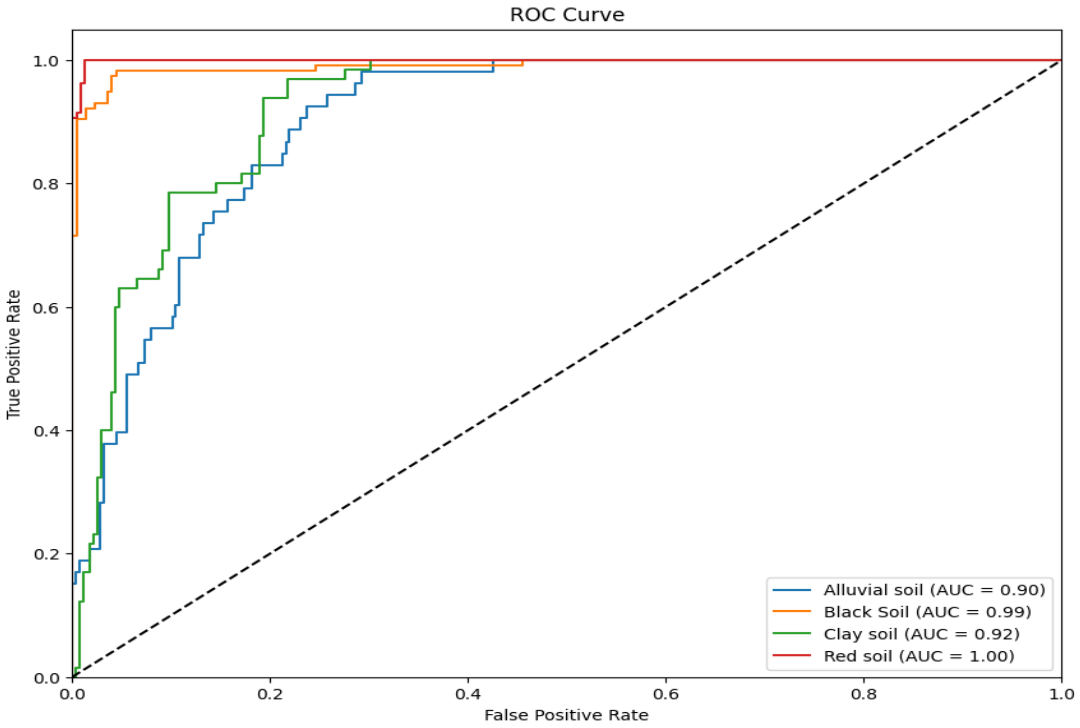




Classification Report:

	precision	recall	f1-score	support
Alluvial soil	0.60	0.45	0.52	53
Black Soil	0.93	0.97	0.95	116
Clay soil	0.61	0.68	0.64	65
Red soil	0.97	0.97	0.97	106
accuracy			0.84	340
macro avg	0.78	0.77	0.77	340
weighted avg	0.83	0.84	0.83	340





Final Metrics:

Final Training Loss: 0.4260

Final Validation Loss: 0.4968

Final Training Accuracy: 0.8265

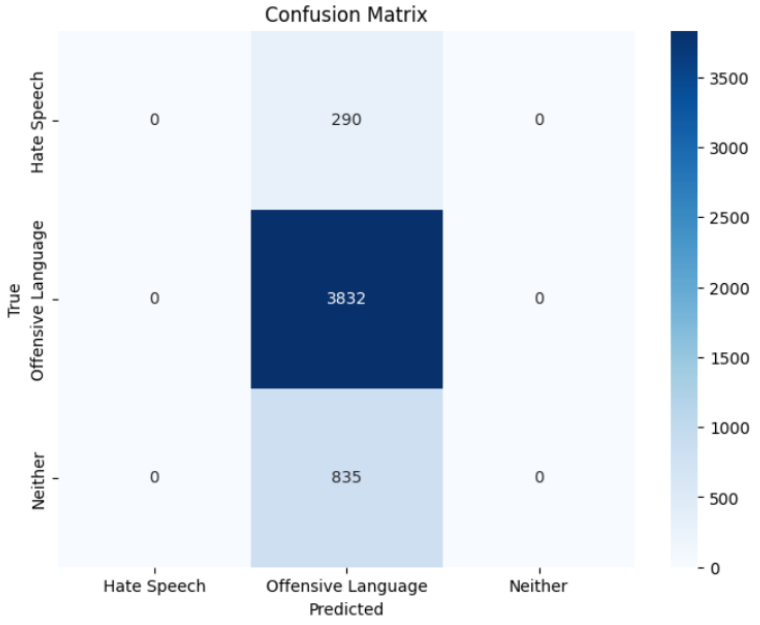
Final Validation Accuracy: 0.7593

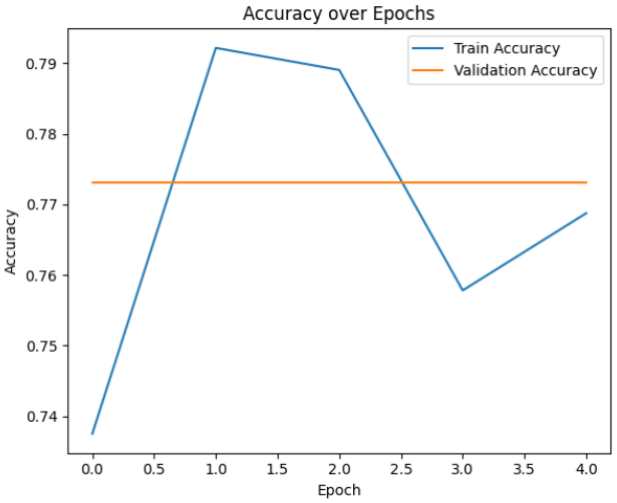
PROJECT-3

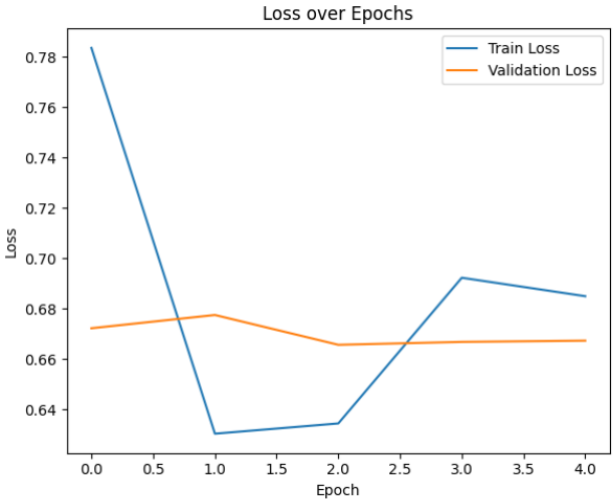
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 100, 128)	1,280,000
lstm_1 (LSTM)	(None, 64)	49,408
dense_1 (Dense)	(None, 3)	195

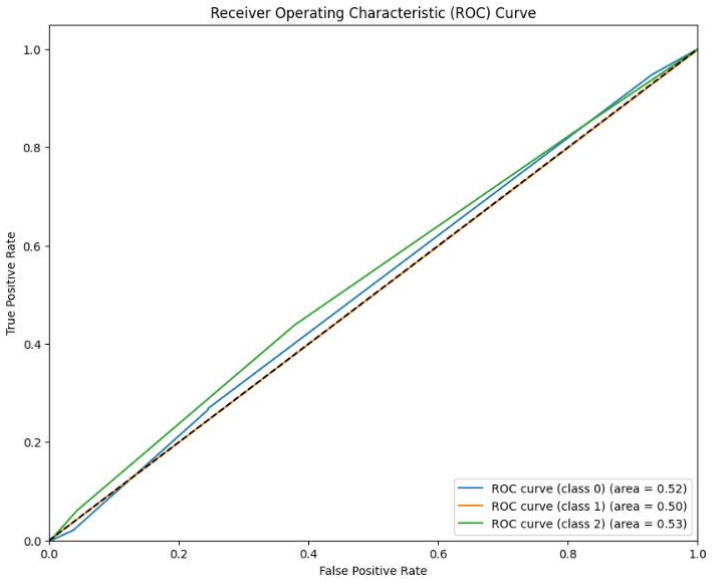
Total params: 1,329,603 (5.07 MB)
Trainable params: 1,329,603 (5.07 MB)
Non-trainable params: 0 (0.00 B)

```
Classification Report:
                    precision    recall  f1-score   support

       Hate Speech       0.00      0.00      0.00       290
Offensive Language       0.77      1.00      0.87      3832
           Neither       0.00      0.00      0.00       835

          accuracy                           0.77      4957
         macro avg       0.26      0.33      0.29      4957
      weighted avg       0.60      0.77      0.67      4957

Accuracy: 0.7730
Precision: 0.5976
Recall: 0.7730
F1 Score: 0.6741
```








Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (class 0) (area = 0.52)
ROC curve (class 1) (area = 0.50)
ROC curve (class 2) (area = 0.53)