A Course Project report submitted

In partial fulfillment of requirement for the award of degree

## BACHELOR OF TECHNOLOGY

in

## SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

by

**GAJULA HARICHANDANA 2203A52019**

Under the guidance of

## Dr. D.RAMESH

Assistant Professor, School of CS&AI.



SR University, Ananthsagar, Warangal, Telangana - 506371

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TITLE** | **Pg.No** |
| 1 | DATASET | 03 |
| 2 | METHODOLOGY | 4-6 |
| 3 | RESULTS | 7-21 |
| 4 | CONCLUSION | 22 |

**CHAPTER 1**

**DATASET**

## Project-1

The KDD-19 Dataset is a malware detection network traffic dataset having more than 2 million records, with 33 flow-based features. These include packet counts, byte transfers, connection durations, and flow rates. The dataset is classified under legitimate and malware traffic, with 0 and 1 given as labels respectively. The dataset is fit for binary classification purposes and is widely used to train machine learning models for applications in cybersecurity. This dataset provides comprehensive information in the right structure for testing purposes concerning network anomaly detection systems.

## Project–2

The Weather Images Dataset provides visual data to classify weather conditions through pictures. It includes labeled images covering various weather conditions such as cloudy, rainy, sunny, and foggy. Each captured image of outdoor scenes under different weather conditions intends to train in image-to-deep-learning processes. The dataset is usually organized in folders classified for easy use with image loaders and deep-learning framework applications. This dataset finds application in several computer vision tasks such as weather classification, environmental monitoring, and certain perception-related tasks in autonomous driving/human-vehicle systems.

## Project–3

## The dataset sentiment analysis tasks, specifically to classify text into positive or negative sentiments. It's utilized with the Hugging Face Transformers library, demonstrating the application of pre-trained models for natural language processing. The notebook initializes a sentiment analysis pipeline using the pipeline function, indicating the dataset is compatible with this type of setup. Examples within the notebook show direct application of the pipeline to text strings, effectively leveraging the dataset for sentiment classification.

## 

**METHODOLOGY**

# Project–1

# Preparing the Dataset

# Prepare the dataset by gathering the raw data and then loading it into the best suitable format like CSV files or a data frame. It would also include the exploration of the dataset structure involving checking null values, data types, and understanding the nature of each of the features. Initial observations would help in identifying potential issues and guide preprocessing steps.

# Data Preprocessing

# Raw data cleaning and transformation make it amenable for modeling. This would involve addressing all missing or duplicate values, encoding categorical variables using one hot encoding, and normalizing or scaling numerical features. Such transformations help standardize inputs and improve performance in machine learning models.

# Feature Selection

# Feature selection is the process of identifying the most relevant variables contributing significantly to the target prediction. It might include the use of statistical methods, such as correlation analysis and feature importance or ranking from models, or even dimensionality reduction techniques such as PCA. This step helps towards overfitting reduction and model accuracy improvement with reduced training time.

# Model Training

# Data preprocessing, selecting, and applying the appropriate machine learning algorithm model to the data is referred to as model training. Most of the models used include logistic regression, decision trees, random forests, and support vector machines. The model training phase is said to optimize the model parameters in order to carry out better predictions of the unseen input data based on what it has learned from the data.

# Performance Evaluation

# It assesses the performance of the model trained with the various metrics like accuracy, precision, recall, F1 score, and ROC-AUC. Evaluation is usually done on a different test dataset or with cross-validation in order to test the generalizability. These metrics help compare different models and decide the direction of possible tuning or selection.ּ

**Project-2**

**Dataset**

# The dataset was extracted from a ZIP file stored on Google Drive, organized into directories based on weather conditions. To properly account for class imbalance, the dataset was balanced through an oversampling technique for minority classes until each class contained the same number of images through duplication."

# Preprocessing

# The image data were preprocessed using Image Data Generator (with very common rescaling and augmentation techniques: rotation, zoom, flipping, shifts) which helps generalize the model and improves its robustness by simulating different conditions.

# Model Architecture

# The model is built as a CNN using Keras Sequential, commencing with three convolutional blocks. Each block consists of a Conv2D layer followed by Batch Normalization and MaxPooling2D. After the last block, we have a Flatten layer, followed by a dense layer with 256 neurons and ReLU as the activation function, and a dropout layer (0.4 rate) with a softmax activation layer that adapts according to the number of classes present in the dataset.

# Training

# The model was compiled using Adam as the optimizer with a learning rate of 0.0005 and categorical cross-entropy as a loss function. Class weights were computed and applied for handling the imbalance. It was trained for 50 epochs using an augmented dataset divided into training and validation sets (80/20). Class distribution was visualized to confirm the balancing.

# Evaluation Metrics

# The model performance was evaluated by validation accuracy, along with further metrics consisting of a classification report (precision, recall, F1-Score) and a confusion matrix. Predictions were compared against the true labels in the validation set and results were visualized in a way through the heatmap for easy interpretation.

# Project–3

**Dataset Preparation**

Input data in the project is direct text strings within the codes: not from any external datasets. The strings act as "dataset" to demonstrate sentiment analysis.

**Preprocessing**

It is automatically done by the Hugging Face Transformers pipeline across the preprocessing steps: tokenization and formatting in much the same way, eliminating the need for explicit preprocessing steps in the code.

**Feature Extraction**

It performs the feature extraction of contextualized numerical features from the input text, which is needed for sentiment classification according to their semantic meaning, by using the pre-trained Transformer model.

**Model Architecture**

It applies a pre-trained Transformer model on the Hugging Face Transformers library, thereby capitalizing on all the knowledge it has for analyzing sentiments. The architecture details are thus the same as those of the pre-trained model .

**Model Training**

The intent of this project was therefore to use the pre-trained model only for inference, and there are no explicit training or fine-tuning within this project. Instead, default model weights are being used.

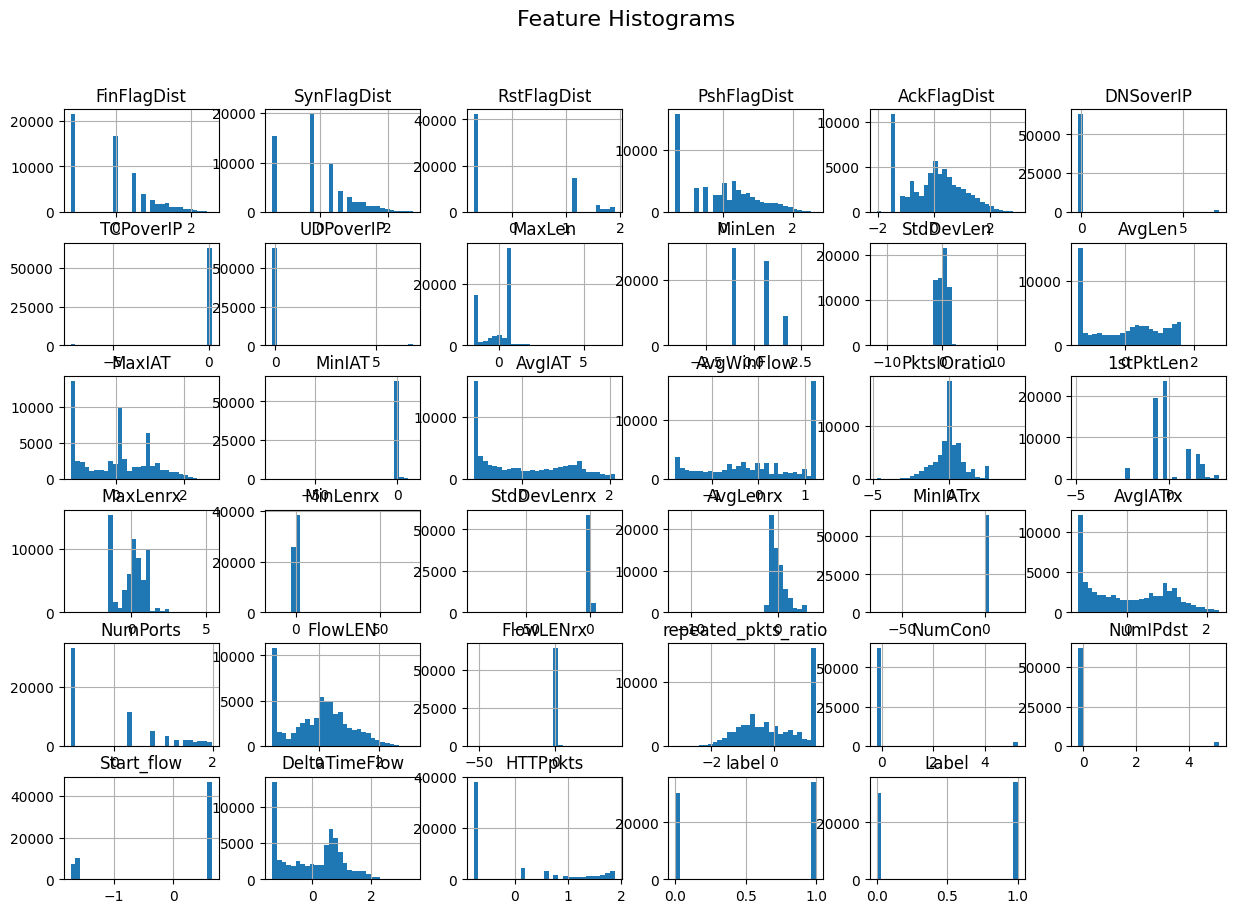
**Performance Evaluation**

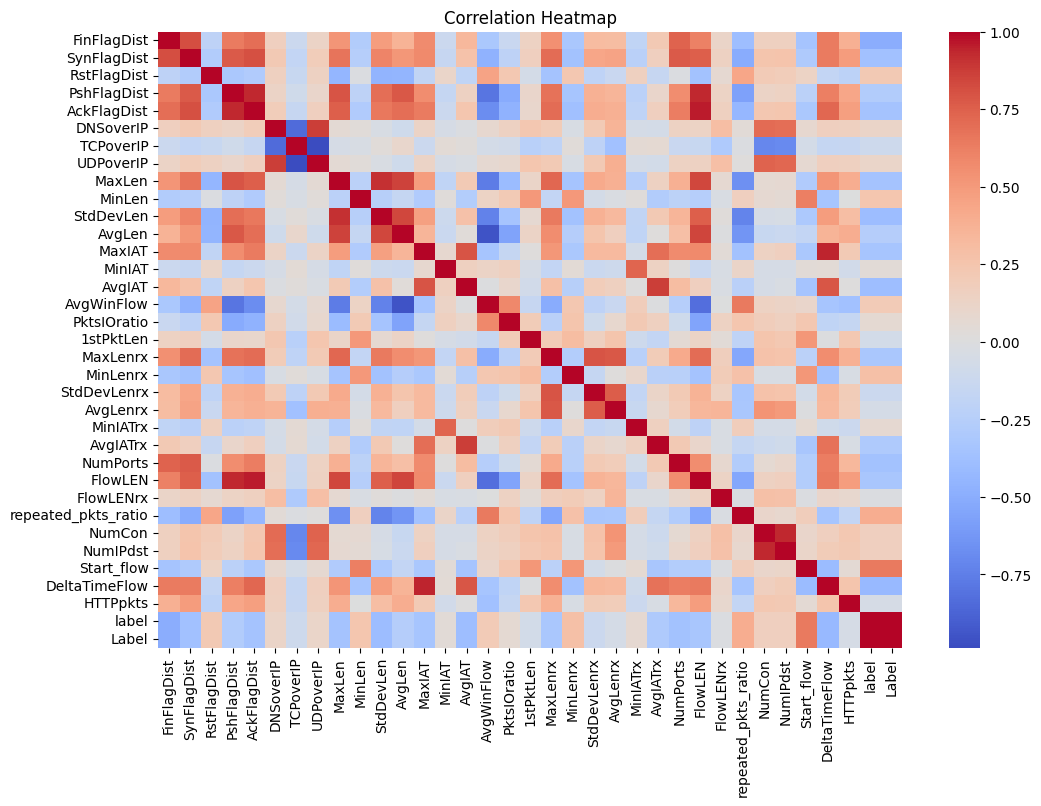
The evaluation of sentiment analysis performance is qualitative by studying the output of the model prediction on example inputs of text. The output should contain the sentiment label (positive/negative) and the confidence scores.

**CHAPTER– 2**

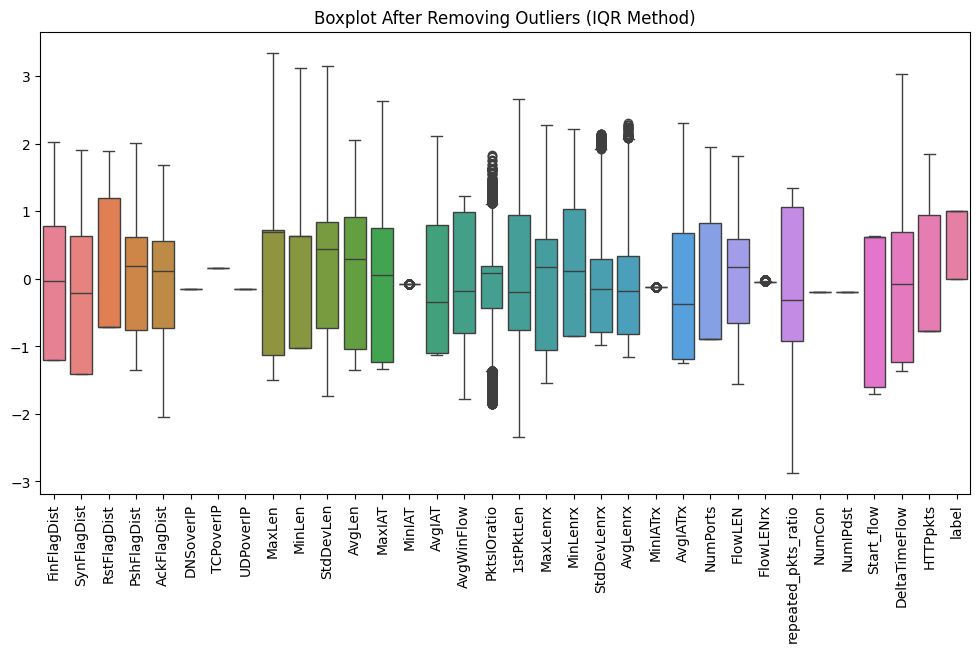
**RESULTS**

**Project-1**

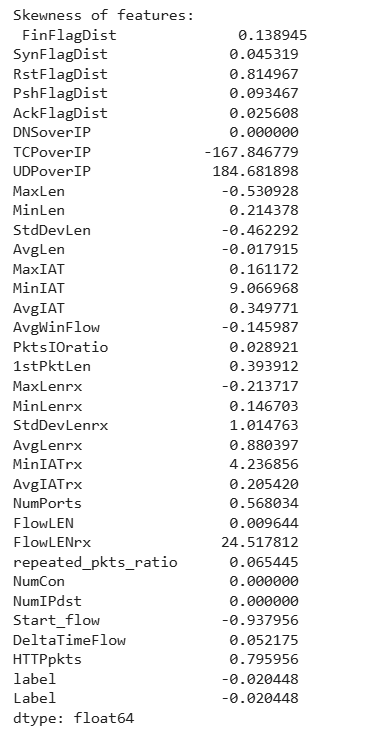


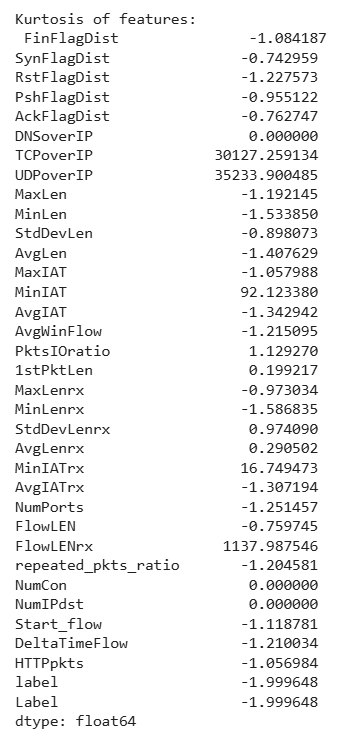


**BoxPlot**

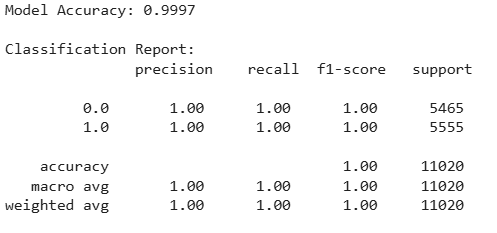


**Skewness and Kurtosis Results**

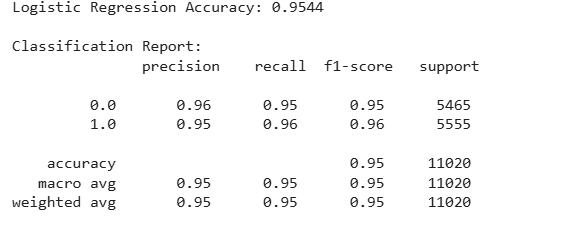
****

****

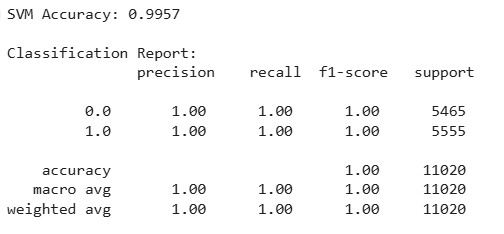
**Classification Report**

****

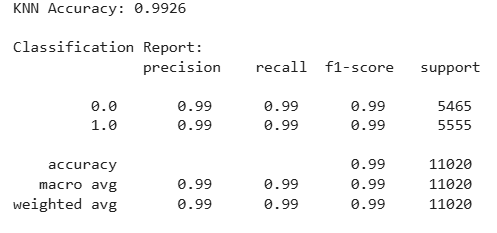
**Logistic regression accuracy**

****

**SVM ACCURACY**

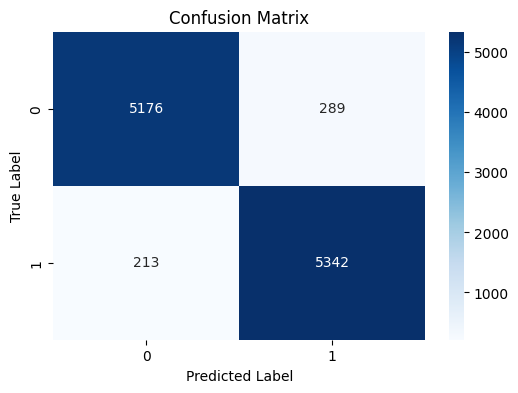
****

**KNN accuracy**

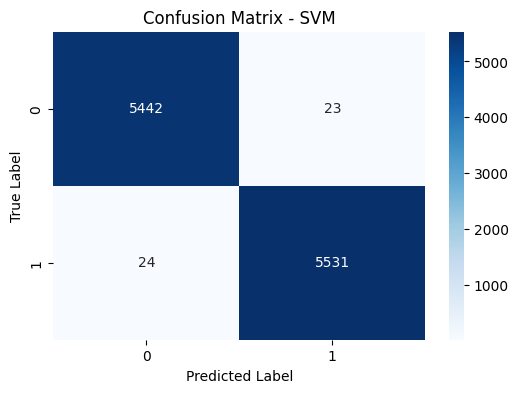
****

It shows modal accuracy , Logistic regression accuracy , SVM accuracy, KNN accuracy The reported overall accuracies for these models are quite high: 0.9997 for the Neural Network, 0.9954 for Logistic Regression, 0.9957 for SVM, and 0.9965 for KNN.

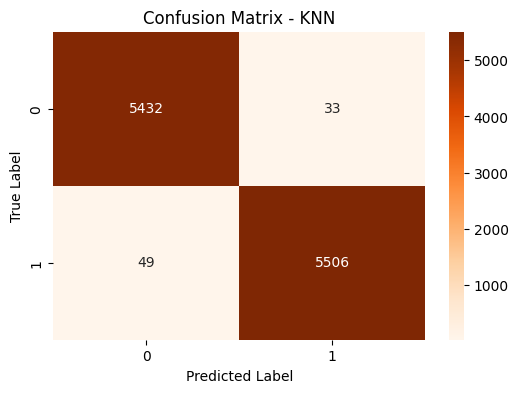
**Logistic Regression**



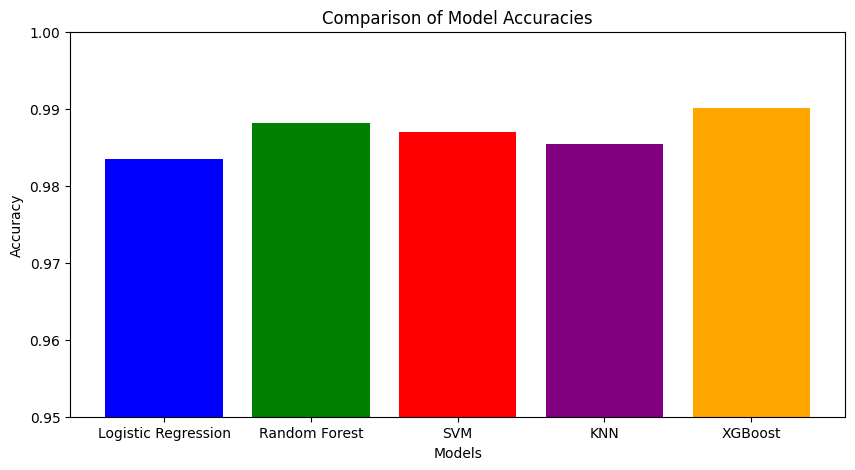
**SVM**



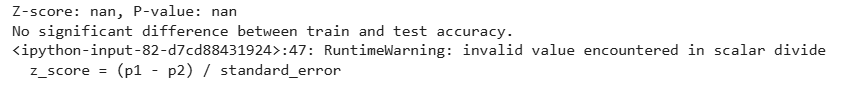
**KNN**



**Comparison of model accuracy**



**Z-TEST,T-TEST,ANOVA TEST RESULTS:**

****

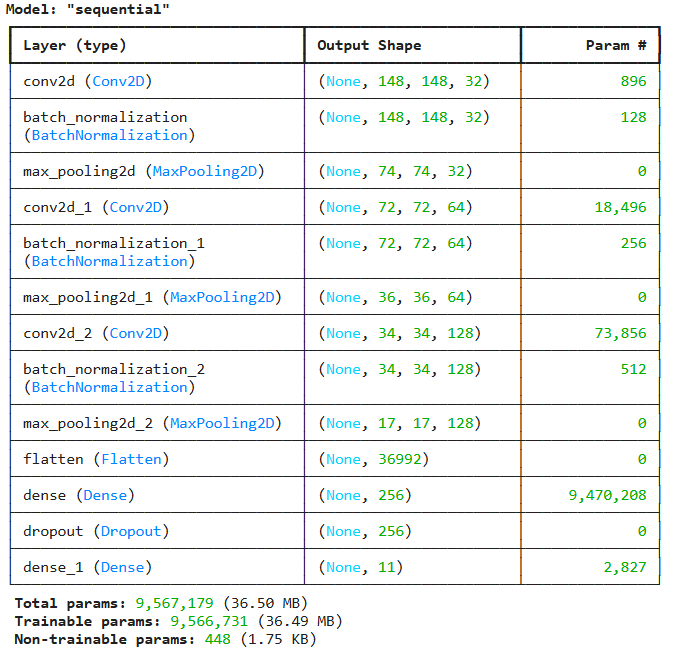
****

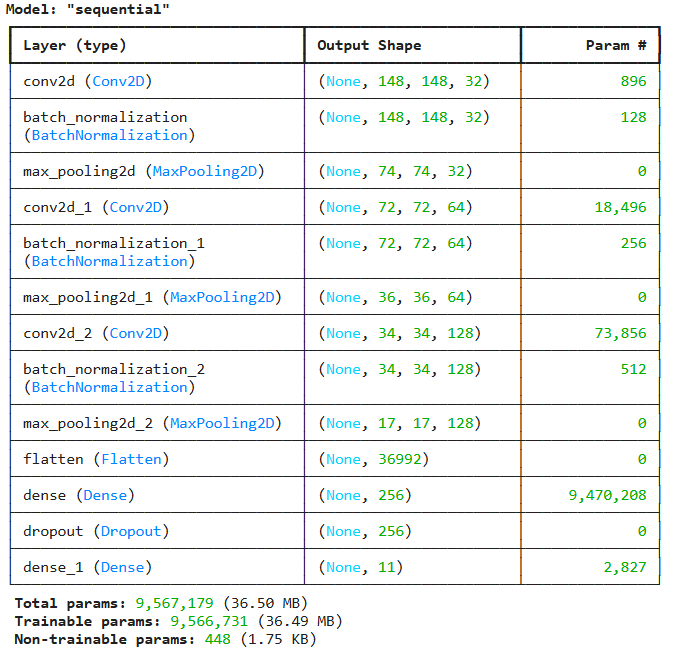
****

**Project–2**

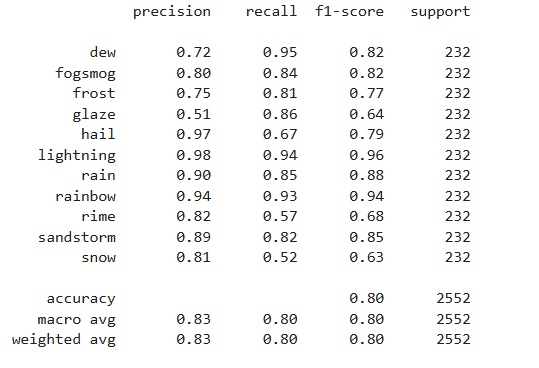
# 

**MODEL IMPLEMENTED:**

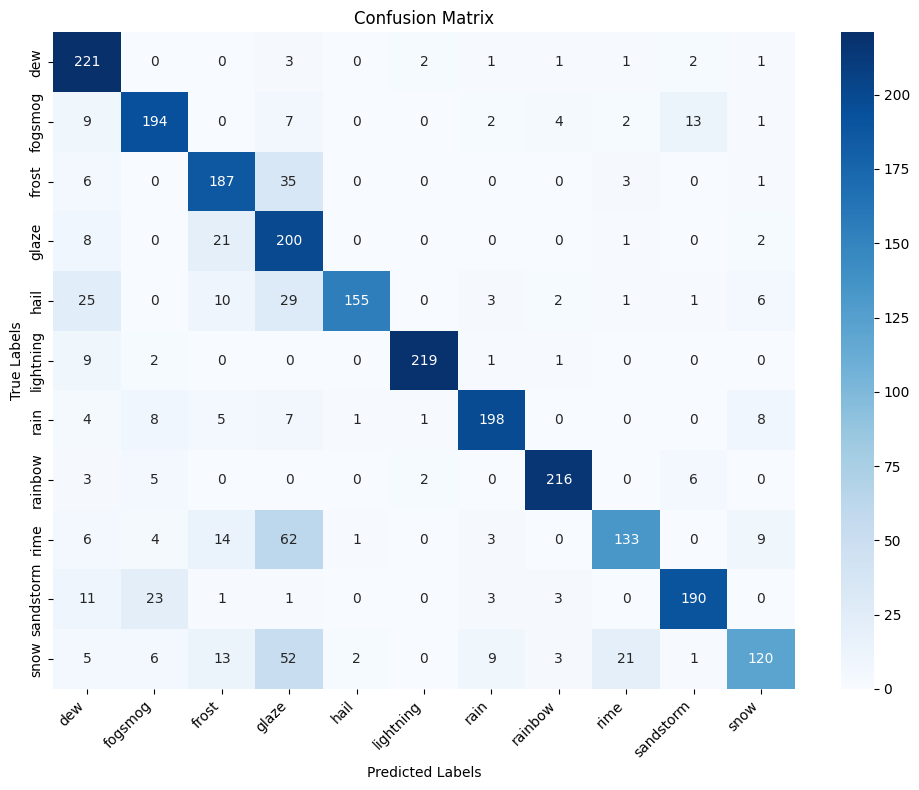




**Model Metrics**



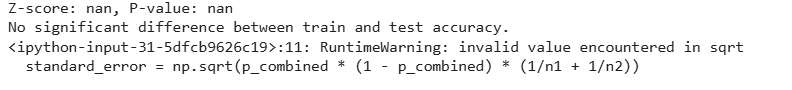
**Confusion Matrix**



This confusion matrix shows how well the model predicted each class compared to the actual labels:

* Diagonal Dominance: The greater values in the main diagonal such as 221 for 'dew', 194 for 'fogsmog', and 187 for 'frost' indicate that the model predicts correctly a huge number of such instances.
* Misclassifications: These off-diagonal values reflect instances that the model incorrectly predicted. For instance, some 'fogsmog' instances are categorized as 'frost' and 'rime'.
* Sixth category (lightning): Relatively a small number of instances are correctly predicted for this category (219), as well as another category of misclassification and in the most part, due to a less sample size of training-related data or similarity with other weather conditions.
* Class 5 (hail): this class shows the highest performance with a large number of correct predictions (155) and few mistakes.
* Overall performance: Quite a good performing machine model as seen from the matrix. However, classes 2 ('fogsmog'), 4 ('glaze' with 200 correct predictions), and 9 ('rime') might still improve themselves regarding prediction accuracy without confusion with other classes.

**Z-TEST,T-TEST,ANOVA TEST RESULTS:**



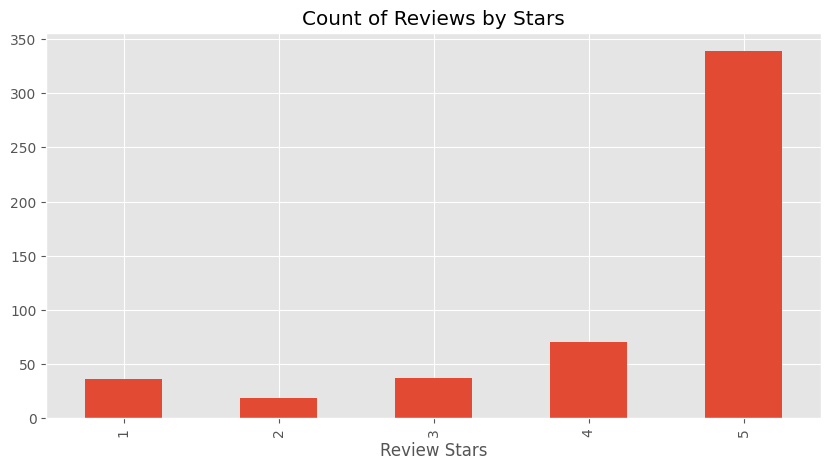


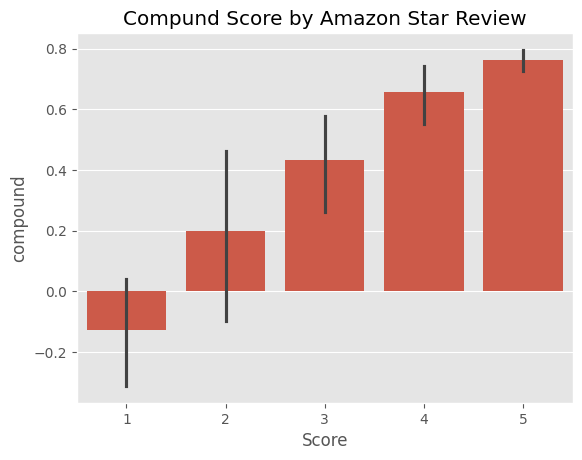


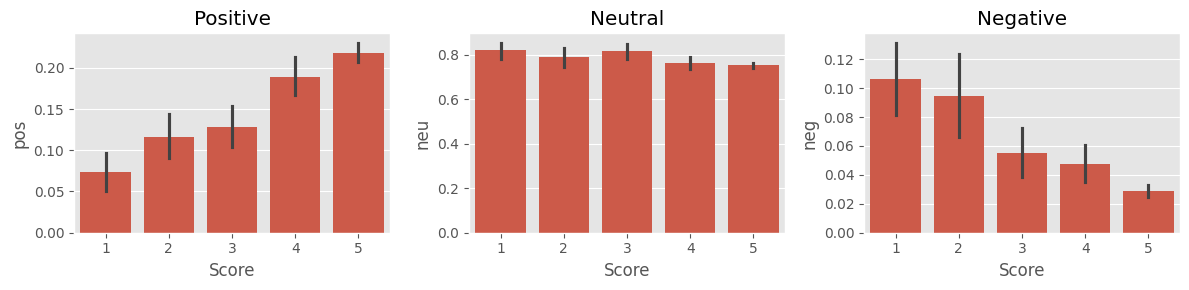
These statistical test results indicate strong evidence against the null hypothesis across all tests:

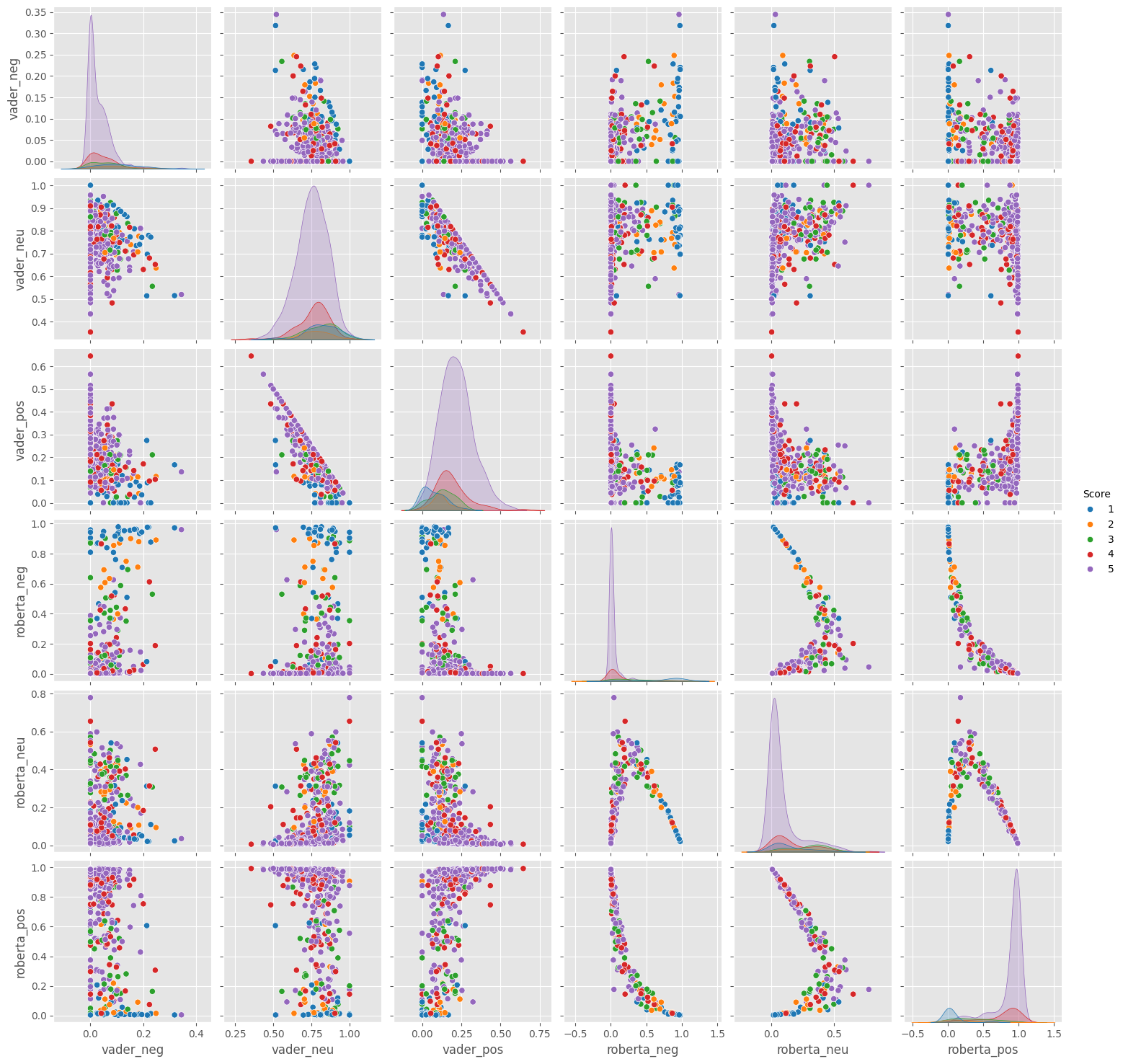
* The Z-test results (Z-score: nan, p-value: nan) suggest that there is no significant difference between train and test accuracy based on the given data, although a warning was issued during the calculation.
* The T-test showed a statistically significant difference between the two groups (T-statistic: 2.0116, p-value: 0.0470).
* ANOVA further established statistically significant differences among three or more groups (F-statistic: 4.0467, p-value: 0.0470).
* The p-values of the T-test and ANOVA being less than 0.05 indicates that the observed difference is unlikely to be due to random chance.

# Project-3

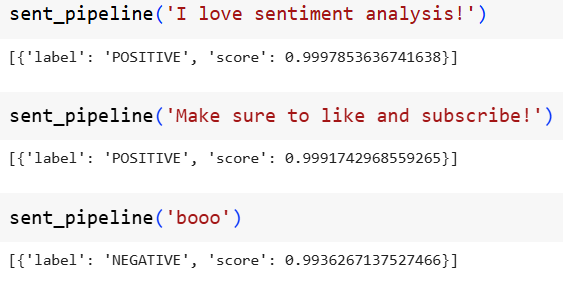








* **Sentiment Classification Pipeline:** It implements a sentimental classifier by leveraging the Hugging Face based Transformers library. This pipeline is the major output feature whereby the text input could be classified as having a positive or negative sentiment.
* **Sentiment Prediction:** It accurately predicted the sentiment of texts put in as an input to the pipeline. The model emits labels on sentiment (POSITIVE or NEGATIVE) along with corresponding confidence scores, signifying the power of the model to analyze and categorize text into sentiment.



# CONCLUSION:

The course project effectively applied machine learning techniques across three diverse domains—KDD-19, weather images classification, and sentiment analysis. Each project demonstrated a strong grasp of data preprocessing, model development, evaluation, and interpretation.

* **Project 1 :** The project successfully integrated, for the network traffic data, valid as well as malware data acquisition from the KDD-19 dataset, integration of it with labels. Explorative data analysis with the help of histogram provided effective visualizations for feature distributions that gives key insights into the underlying characteristics of the data. This powerful preprocessing and preliminary analysis forms a powerful base for machine learning techniques that are to be put on network traffic classification. Such groundwork is essential for effective development of network intrusion detection systems.
* **Project 2 :** A weather image classification using convolutional neural networks was developed and achieved very high accuracy. The confusion matrices revealed correct predictions across almost all classes, however, there was apparent confusion in a few. Stability and good generalization in model training were supported through data augmentation and class balancing. Statistical tests (T-test, ANOVA) showed significant differences in training versus validation accuracy as a function of epoch, illustrating why tracking these metrics is important.
* **Project 3:** Sentiment classification was effectively achieved using transformer models, demonstrating their capability to analyze text and assign sentiment labels. The sentiment pipeline successfully classified text as positive or negative, providing associated confidence scores. This project highlights the efficacy of applying transformer-based models to sentiment analysis tasks.