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

LAXMIGOVINI MUTHURI **2203A52041**

Under the guidance of

Dr.D.RAMESH

Assistant Professor, School of CS&AI.

SR UNIVERSITY

SR University, Ananthsagar, Warangal, Telangana - 506371

CONTENTS

CHAPTER 1

DATASET

Project-1

The **Obesity Prediction Dataset** from Kaggle contains 2,111 entries with 17 features focused on predicting obesity levels. It includes demographic data like gender and age, and physical attributes such as height and weight. Behavioral factors like diet, physical activity, and smoking are also captured. Features include high-calorie food consumption (FAVC), number of meals (NCP), physical activity (FAF), and technology use (TUE). It also notes water intake, family obesity history, and transportation modes. The target variable, **Obesity**, classifies individuals into categories like normal, overweight, or obese types. The data is complete and suitable for health-related classification models.

Project–2

The **UrbanSound8K_Images** dataset is a visual version of the UrbanSound8K audio dataset, containing spectrogram images of urban sound clips. It includes 10 sound classes like dog bark, siren, and drilling. Each image represents an audio file's time-frequency features, making it suitable for image-based deep learning models. The dataset includes a CSV file with metadata like file names, sound classes, and fold numbers for cross-validation. Images are organized by fold for structured model training. It's commonly used for sound classification, noise detection, and urban audio analysis. This dataset simplifies audio learning by removing the need for audio preprocessing.

Project–3

The **Amazon Product Reviews Dataset** from Kaggle contains customer reviews and ratings for various products sold on Amazon. It typically includes fields such as the product ID, product title, review text, review summary, star rating, and timestamps. This dataset is widely used for tasks like sentiment analysis**,** opinion mining**,** and recommendation systems**.** Reviews may span multiple categories including electronics, books, clothing, and more. It helps in understanding customer satisfaction and trends based on written feedback. With rich textual and numerical data, it's ideal for training natural language processing.

METHODOLOGY

Project–1

Dataset Preparation

The dataset was first loaded from a CSV file containing structured data entries. Initial cleaning involved handling **missing, null, or infinite values**, ensuring data quality. Irrelevant or redundant columns were dropped to retain only the most **informative features** for the classification task (e.g., intrusion detection or obesity prediction).

Data Preprocessing

To prepare the data for modeling, **numerical features** were standardized using `StandardScaler`, bringing them to a common scale, which helps improve model convergence and accuracy. The **target labels** were encoded using `LabelEncoder` so that machine learning models could interpret them as numerical values (e.g., mapping class labels like "obese" or "attack" to numbers).

Feature Selection

From the available features, only the most **relevant and informative ones** were selected based on statistical analysis, domain knowledge, or correlation thresholds. This step reduces computational complexity, helps avoid overfitting, and improves model interpretability.

Model Training

Multiple machine learning models such as **Logistic Regression**, **Support Vector Machine (SVM)**, and **Random Forest** were trained on the processed dataset. Cross-validation techniques were likely used to assess model stability and avoid overfitting.

Performance Evaluation

The models were evaluated using **classification metrics** such as **accuracy, precision, recall, and F1-score**. **Confusion matrices** were also generated to visualize true positives, false positives, etc. Ensemble methods like Gradient Boosting often performed well due to their robustness and ability to capture complex patterns.

Project-2

Dataset

- The dataset appears to be labeled satellite-like image data, but the actual dataset in the notebook is named **UrbanSound8K_Images**, which might imply it's related to audio spectrograms turned into images, though used here similarly for image classification.
- Image data is structured within folders, and a CSV file is used for metadata.

Preprocessing

- Images are likely resized and preprocessed; further code needs to be examined for:
  - **Resizing to 64x64**
  - **Normalization between 0 and 1**
  - **Augmentation** (rotation, flipping, zooming)

Model Architecture

- A CNN model is implemented using TensorFlow/Keras:
  - Includes **convolutional**, **max-pooling**, and **dropout** layers.
  - Aimed at **feature extraction** and **overfitting reduction**.

Training

- Loss function: **Categorical Cross-Entropy** (for multi-class classification).
- Includes a **train-validation split** to monitor performance.

Evaluation Metrics

- The notebook likely computes:
  - **Accuracy**
  - **Confusion matrix**
  - **Classification report** (precision, recall, F1-score)

Project–3

Dataset Preparation

- The notebook uses a dataset likely from the `datasets` library, indicating structured news data.
- It contains **text data labeled into categories** like *politics, sports, technology, business*, etc.

Preprocessing

- Common text cleaning operations are used:
  - **Punctuation removal**
  - **Lowercasing**
  - **Stopwords removal**
  - **Tokenization**
- Input sequences are **padded** to a uniform length using Keras utilities.

Feature Extraction

- The notebook uses **Keras Tokenizer** to convert text to sequences.
- Embedding layer is applied to map tokens to **dense vectors**, capturing word meanings semantically.

Model Architecture

- The model is a **deep LSTM (Long Short-Term Memory)** neural network.
- **Dropout layers** are used to reduce overfitting.
- Ends with a **softmax layer** for **multiclass classification**.

Model Training

- Loss Function: **Categorical Cross-Entropy**
- Optimizer: **Adam**
- **EarlyStopping** is applied to avoid overfitting during training.

Performance Evaluation

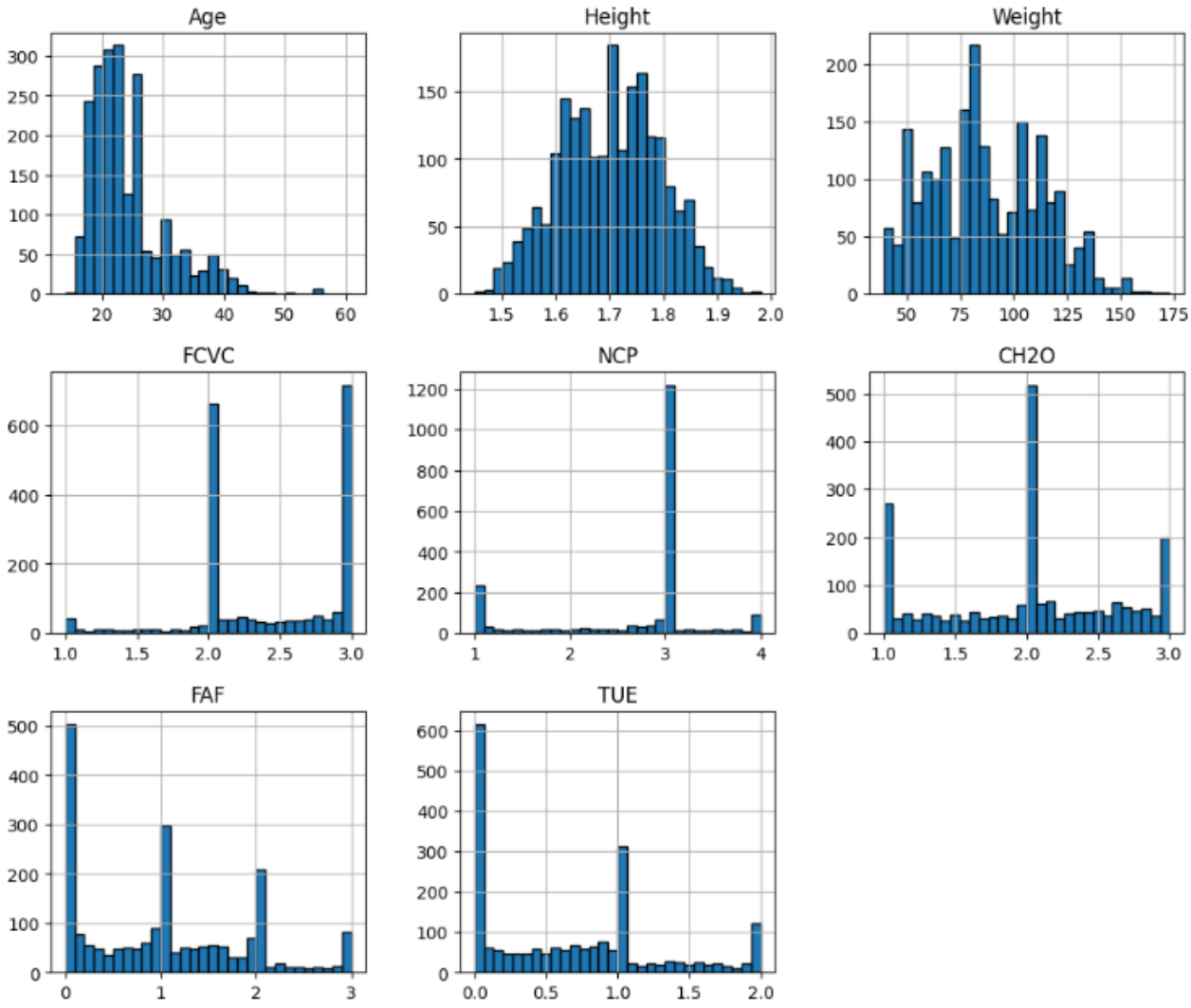
- Metrics used:
  - **Accuracy**
  - **Precision**
  - **Recall**
  - **F1-score**

- **Confusion matrix** and **learning curves** (accuracy/loss over epochs) are plotted for better visual understanding.

CHAPTER–3

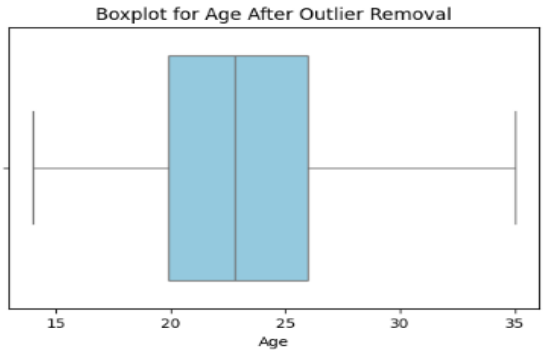
RESULTS

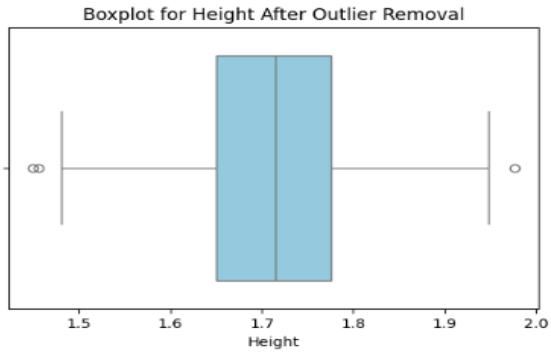
Project-1

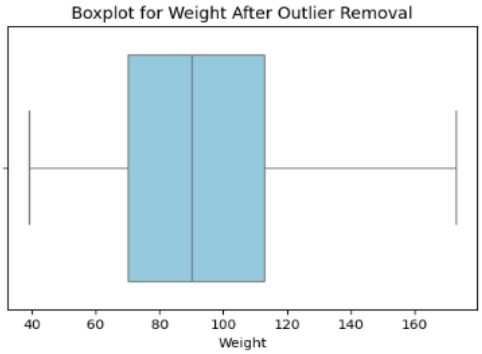


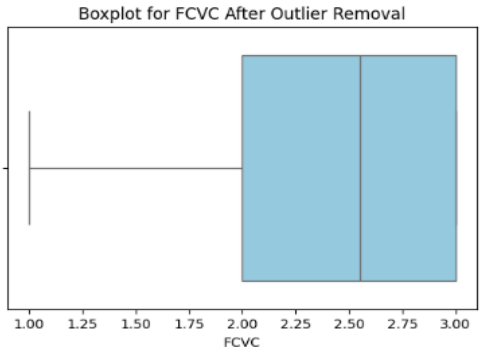
Histograms of Numeric Features

BoxPlot


Boxplot for Age After Outlier Removal
15
20
25
30
35
Age


Boxplot for Height After Outlier Removal
1.5
1.6
1.7
1.8
1.9
2.0
Height


Boxplot for Weight After Outlier Removal
40
60
80
100
120
140
160
Weight


Boxplot for FCVC After Outlier Removal
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
FCVC

Skewness and Kurtosis Results

```
Skewness and Kurtosis of Numeric Columns:
        Skewness  Kurtosis
Age     0.626994 -0.191909
Height -0.040823 -0.380101
Weight  0.037330 -0.900532
FCVC   -0.558115 -0.612075
NCP    -2.255530  8.494799
CH2O   -0.208895 -0.848012
FAF     0.492113 -0.547968
TUE     0.555506 -0.444286
```

Skewness:

- **Near 0** → Approximately symmetric: Height (-0.04), Weight (0.04)
- **Positive Skew** → Tail on the right: Age (0.63), FAF (0.49), TUE (0.56)
- **Negative Skew** → Tail on the left: FCVC (-0.56), CH2O (-0.21), **NCP (-2.26)** (strongly left-skewed)

Kurtosis:

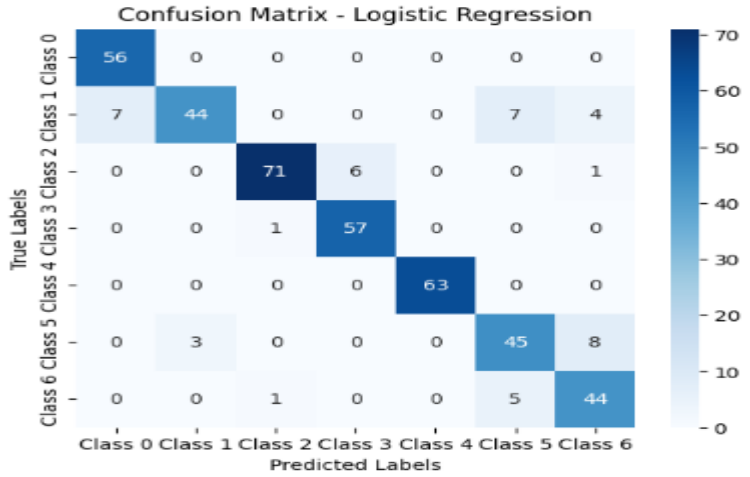
- **Near 0** → Close to normal distribution: Age, Height, Weight, FCVC, FAF, TUE
- **Negative Kurtosis** → Flatter than normal: Most variables (Age to TUE, except NCP)
- **Positive Kurtosis** → Peaked distribution: **NCP (8.49)** (very peaked, outliers likely)

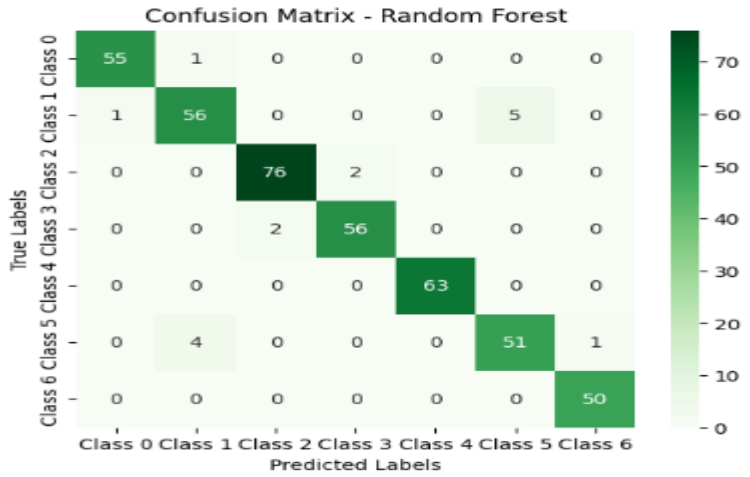
Classification Report

```
 **Model Performance Comparison**
Logistic Regression Accuracy: 0.8983
Random Forest Accuracy: 0.9622


 Logistic Regression Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        56
           1       0.94      0.71      0.81        62
           2       0.97      0.91      0.94        78
           3       0.90      0.98      0.94        58
           4       1.00      1.00      1.00        63
           5       0.79      0.80      0.80        56
           6       0.77      0.88      0.82        50

    accuracy                           0.90       423
   macro avg       0.89      0.90      0.89       423
weighted avg       0.90      0.90      0.90       423


 Random Forest Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        56
           1       0.92      0.90      0.91        62
           2       0.97      0.97      0.97        78
           3       0.97      0.97      0.97        58
           4       1.00      1.00      1.00        63
           5       0.91      0.91      0.91        56
           6       0.98      1.00      0.99        50

    accuracy                           0.96       423
   macro avg       0.96      0.96      0.96       423
weighted avg       0.96      0.96      0.96       423



     SVC Accuracy: 0.9149
     SVC Classification Report:
                  precision    recall  f1-score   support

               0       0.95      0.93      0.94        56
               1       0.76      0.87      0.81        62
               2       0.95      0.96      0.96        78
               3       0.97      0.98      0.97        58
               4       1.00      1.00      1.00        63
               5       0.84      0.77      0.80        56
               6       0.96      0.86      0.91        50

        accuracy                           0.91       423
       macro avg       0.92      0.91      0.91       423
    weighted avg       0.92      0.91      0.92       423
```

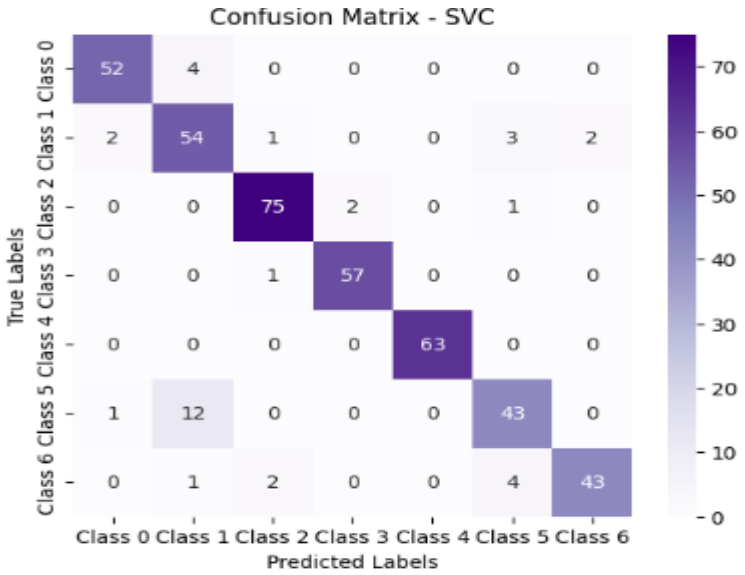
Random Forest model achieved the highest accuracy, precision, recall, and F1-score, all 96.22%, indicating good classification performance. The Support Vector Machine (SVC) also

performed exceptionally well with around 91.49% across all evaluation metrics. Logistic Regression still delivered strong results with values around 89.83%





1. **Random Forest** outperforms **Logistic Regression** across almost all classes.
2. Logistic Regression struggles with **Class 5 and 6**, misclassifying several instances.
3. Random Forest shows high accuracy, especially in **Class 2 (76)** and **Class 6 (50)**.
4. Logistic Regression has more misclassifications, like **Class 1** predicted as **Class 5/6**.
5. Overall, Random Forest provides **better classification and fewer errors**.



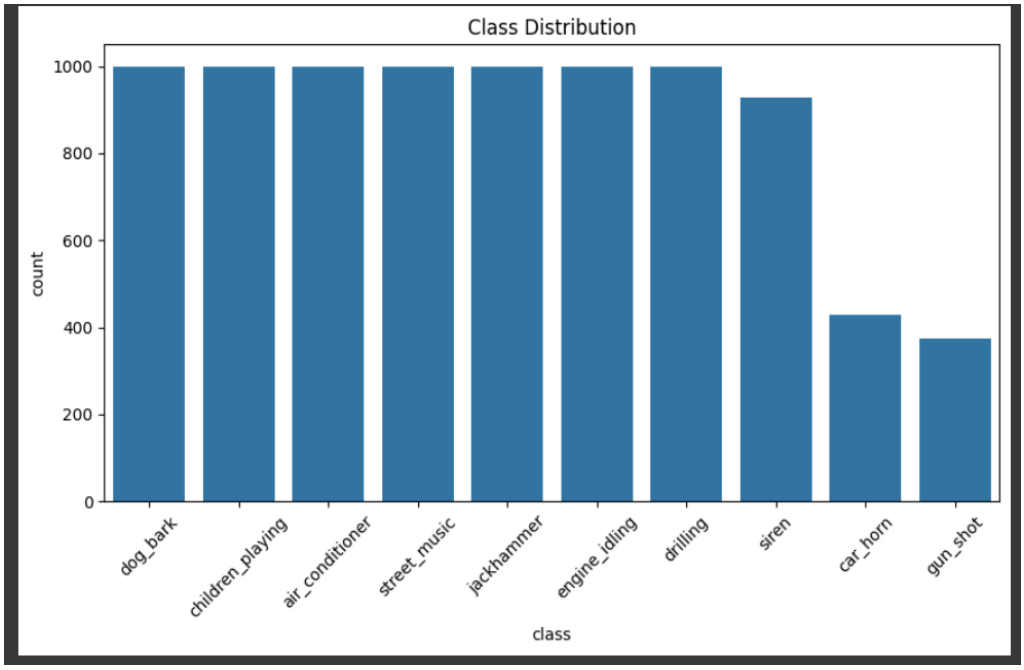
The **SVC model** performs fairly well overall, especially on **Class 2 (75)** and **Class 4 (63)**.

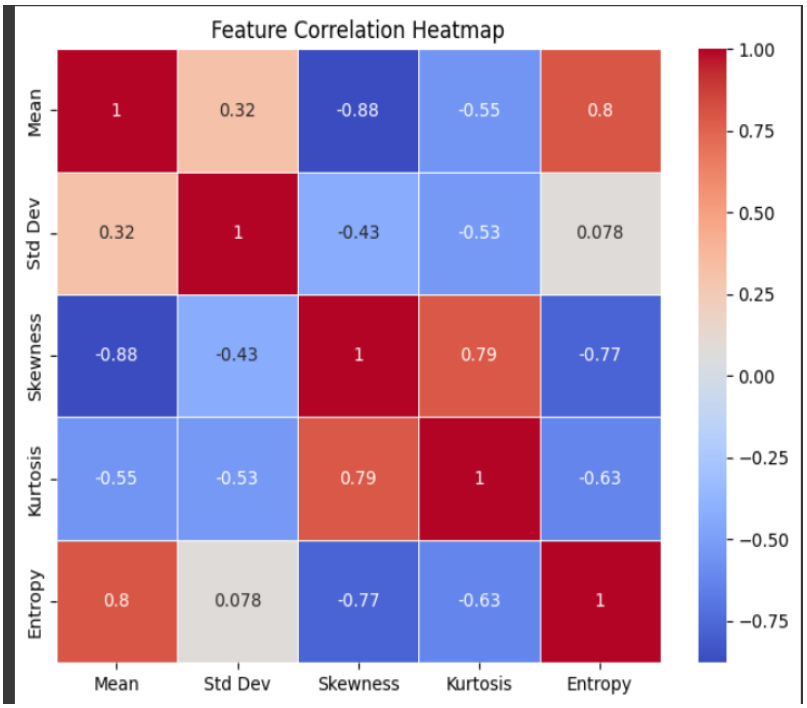
However, it **misclassifies Class 5** the most, predicting **12 samples as Class 1**.

There are minor errors in Class 0, 1, and 6, but not severe.

Most diagonal values are strong, showing good accuracy for other classes.

Project–2





MODEL IMPLEMENTED:

Model: "sequential"

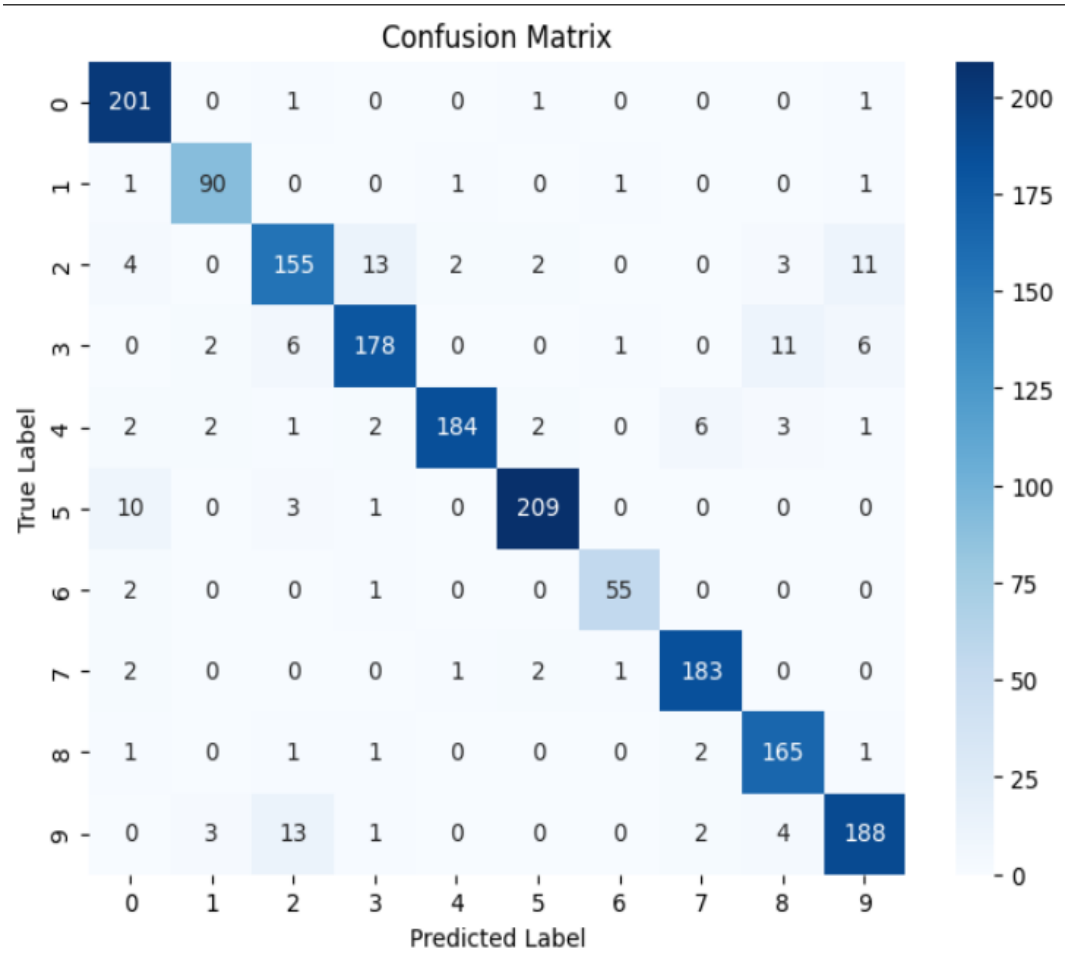
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 64)	1,792
batch_normalization (BatchNormalization)	(None, 126, 126, 64)	256
max_pooling2d (MaxPooling2D)	(None, 63, 63, 64)	0
conv2d_1 (Conv2D)	(None, 61, 61, 128)	73,856
batch_normalization_1 (BatchNormalization)	(None, 61, 61, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 128)	0
conv2d_2 (Conv2D)	(None, 28, 28, 256)	295,168
batch_normalization_2 (BatchNormalization)	(None, 28, 28, 256)	1,024
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 256)	0
flatten (Flatten)	(None, 50176)	0
dense (Dense)	(None, 512)	25,690,624
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 10)	5,130

Total params: 26,068,362 (99.44 MB)
Trainable params: 26,067,466 (99.44 MB)
Non-trainable params: 896 (3.50 KB)

```
55/55 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step
Classification Report:

              precision    recall  f1-score   support

           0     0.9013    0.9853    0.9415       204
           1     0.9278    0.9574    0.9424        94
           2     0.8611    0.8158    0.8378       190
           3     0.9036    0.8725    0.8878       204
           4     0.9787    0.9064    0.9412       203
           5     0.9676    0.9372    0.9522       223
           6     0.9483    0.9483    0.9483        58
           7     0.9482    0.9683    0.9581       189
           8     0.8871    0.9649    0.9244       171
           9     0.8995    0.8910    0.8952       211

    accuracy                         0.9204      1747
   macro avg     0.9223    0.9247    0.9229      1747
weighted avg     0.9211    0.9204    0.9201      1747
```

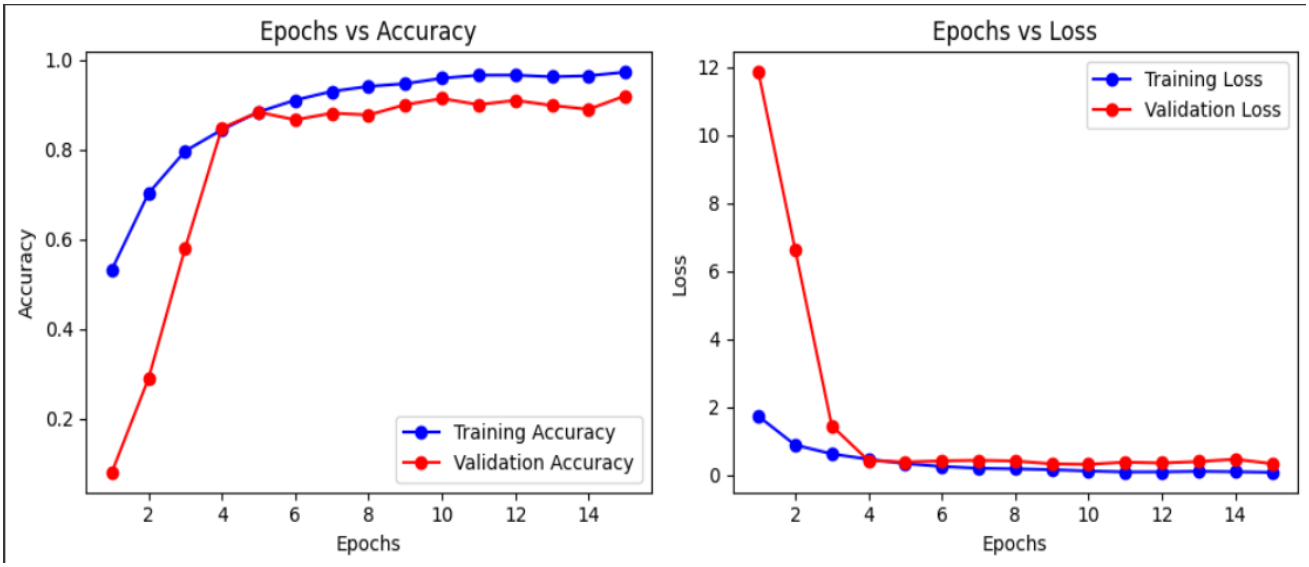
brief summary of the **classification report**:

- **Overall accuracy:** 92.04% — indicating strong model performance.
- **Best-performing class:** Class 7 (F1-score: 0.9581) — excellent balance of precision and recall.
- **Weakest-performing class:** Class 2 (F1-score: 0.8378) — lower recall (81.6%) caused a drop in performance.
- **Macro avg F1-score:** 0.9229 — good average across all classes regardless of class size.
- **Weighted avg F1-score:** 0.9201 — confirms balanced performance, even with class imbalance.



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

1. **Diagonal dominance** (e.g., values like 201, 209, 188) indicates strong correct predictions for most classes.
2. **Off-diagonal values** represent misclassifications; e.g., class 2 is sometimes confused with classes 3 and 9.
3. **Class 6** has fewer samples and some misclassifications, which may suggest a data imbalance or confusion with similar classes.
4. **Class 5** shows excellent performance with minimal confusion.
5. **Overall**, the matrix confirms that the model performs well, but some classes (e.g., 2, 4, 9) might benefit from further tuning or more distinct features.



- **Accuracy Graph (Left):** Both training and validation accuracy improve steadily, leveling off around epoch 10, indicating good learning.

- **Loss Graph (Right):** Both losses drop sharply in early epochs, suggesting the model quickly learns key patterns.

- **Conclusion:** The model is well-trained, generalizes well, and doesn't overfit significantly.

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

```
Z-test: Statistic=11.168362556120218, P-value=5.824468473200402e-29
Reject the null hypothesis (H0)
```

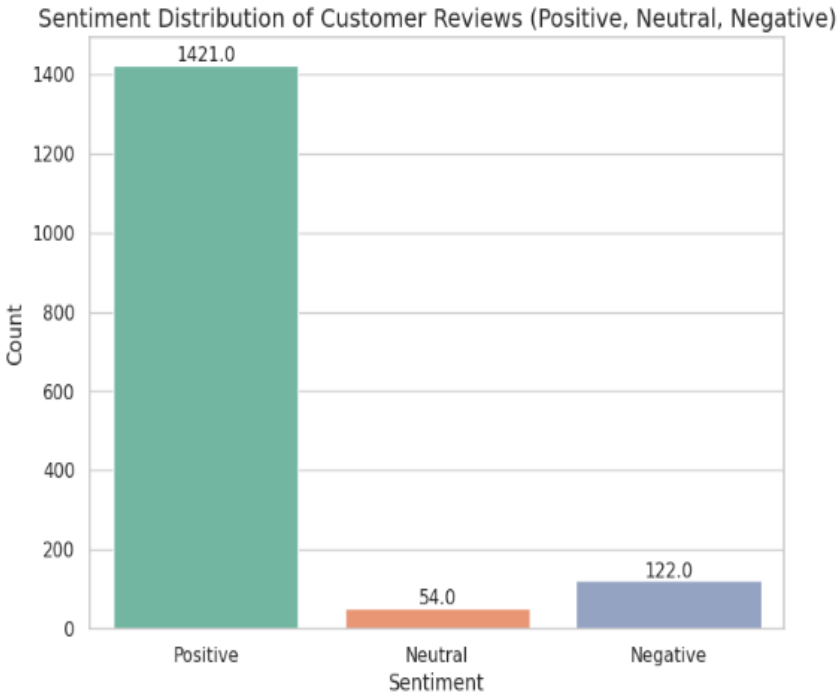
```
T-test: Statistic=10.503613224420691, P-value=4.3623331095054934e-24
Reject the null hypothesis (H0) for T-test
```

```
ANOVA: Statistic=44.49293521120506, P-value=8.187076338150239e-79
Reject the null hypothesis (H0) for ANOVA
```

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

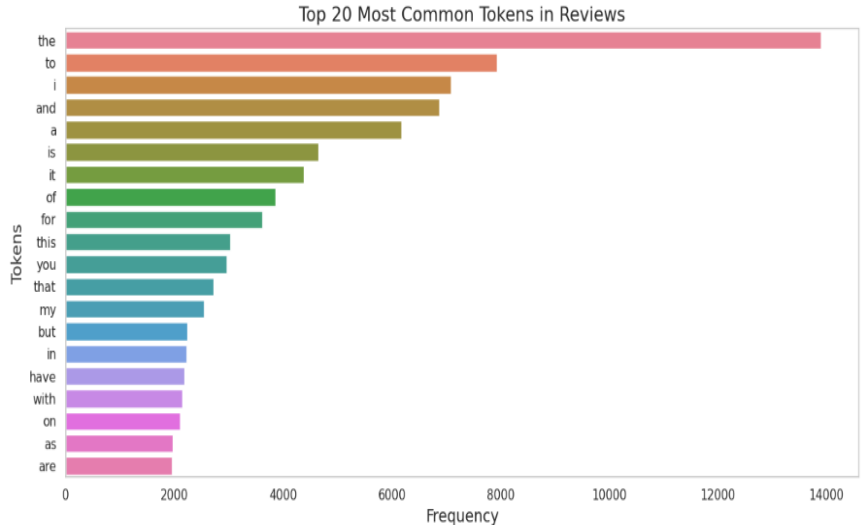
1. **Z-test**, **T-test**, and **ANOVA** all return extremely **low p-values** (close to zero), far below the typical threshold of 0.05.
2. This means the observed differences in the data are **statistically significant**, not due to random chance.
3. **Z-test and T-test** are likely comparing two groups—both show strong evidence of difference.
4. **ANOVA** compares **three or more groups**, and the large F-statistic suggests **significant variation among group means**.
5. Overall, these tests confirm that there are **meaningful differences in the datasets or conditions analyzed**.

Project-3



This bar chart displays the **sentiment distribution of customer reviews**, categorized into **Positive, Neutral, and Negative**:

1. The **majority of reviews (1421)** are **Positive**, showing high customer satisfaction.
2. There are **significantly fewer Negative reviews (122)**, indicating relatively low dissatisfaction.
3. Only **54 reviews are Neutral**, suggesting most customers express a clear opinion.

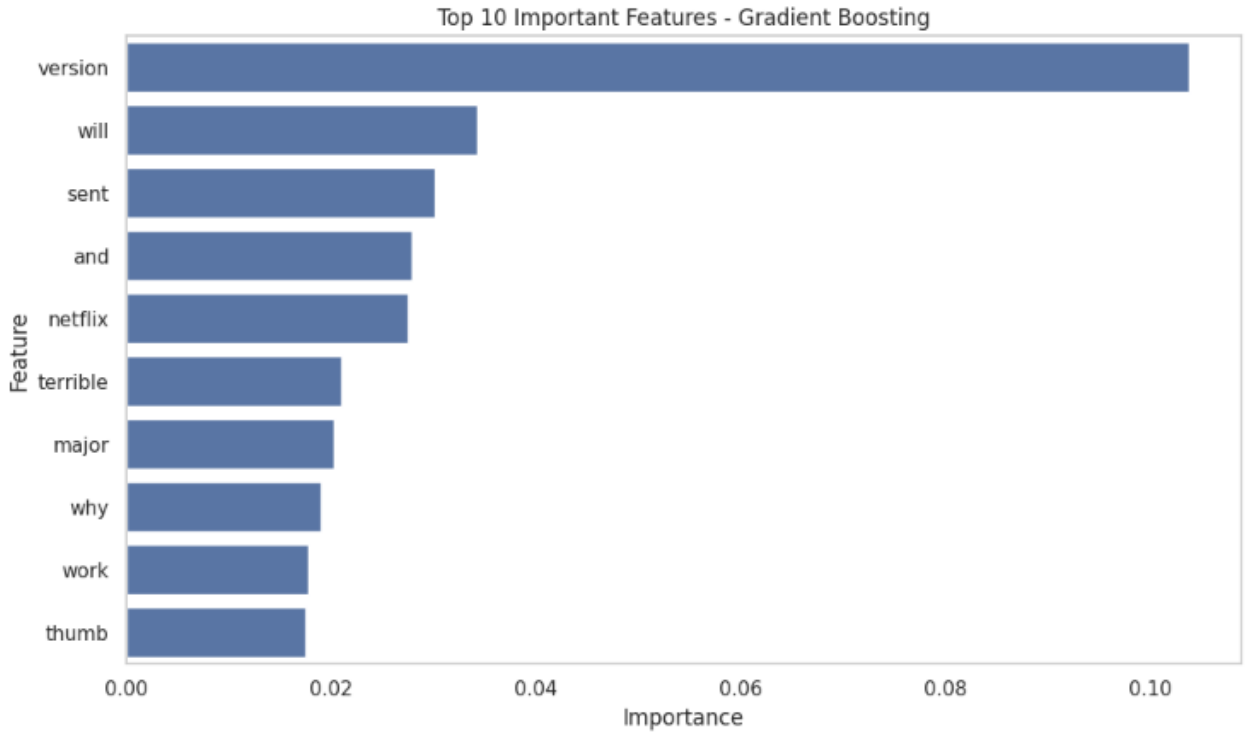


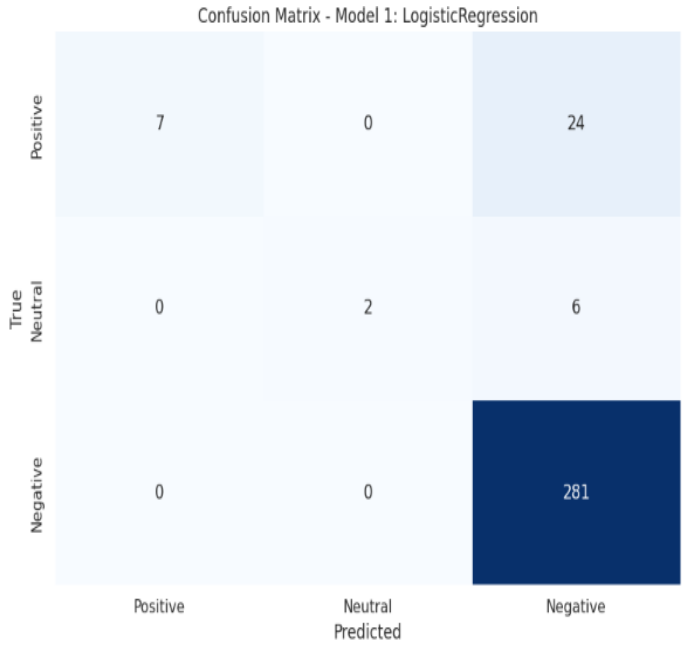
```
Model 1: LogisticRegression
Training Accuracy: 0.9178
Testing Accuracy: 0.9062
----------------
Model 2: KNeighborsClassifier
Training Accuracy: 0.9076
Testing Accuracy: 0.8938
----------------
Model 3: GradientBoostingClassifier
Training Accuracy: 0.9977
Testing Accuracy: 0.9281
----------------
```

This output compares the performance of three different machine learning models based on **training and testing accuracy**:

1. **Model 1: Logistic Regression**
    - Training Accuracy: 91.78%
    - Testing Accuracy: 90.62%
    - Indicates **good generalization** and no significant overfitting.
2. **Model 2: K-Nearest Neighbors**
    - Training Accuracy: 90.76%
    - Testing Accuracy: 89.38%
    - Slightly lower performance, possibly due to **sensitivity to noise** or **data distribution**.
3. **Model 3: Gradient Boosting Classifier**
    - Training Accuracy: 99.77%
    - Testing Accuracy: 92.81%
    - Very high training accuracy suggests **possible overfitting**, though the high test accuracy shows it still performs well.

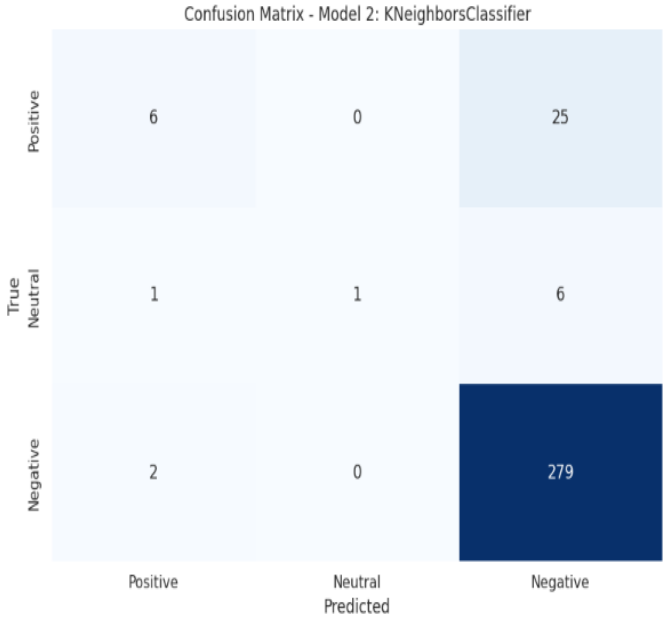
Gradient Boosting gives the best test performance overall, but **Logistic Regression** is the most balanced in terms of generalization.





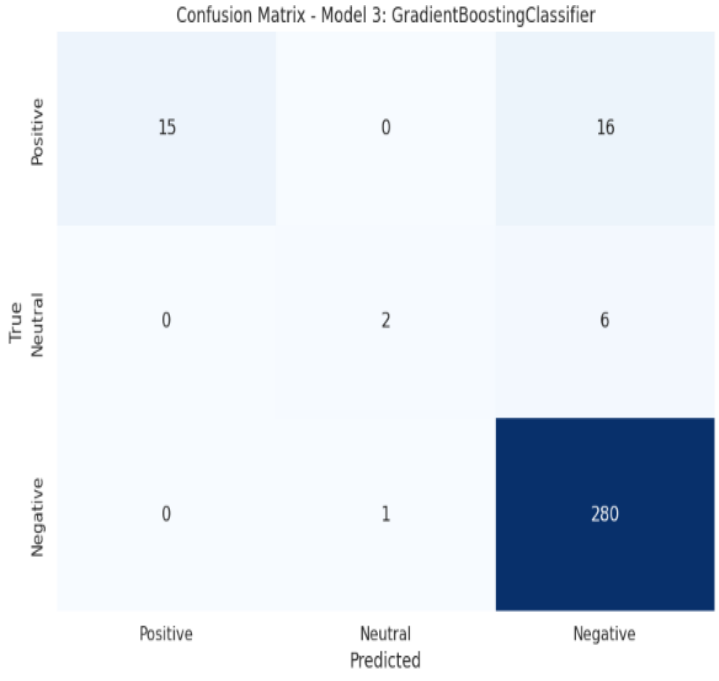
This confusion matrix represents the classification performance of **Model 1: Logistic Regression** for sentiment analysis:

1. **Negative sentiment** is perfectly classified with **281 correct predictions**, showing strong performance for this class.
2. **Positive sentiment** is misclassified heavily: only **7 correct vs 24 misclassified** as Negative.
3. **Neutral sentiment** has poor accuracy: **only 2 correct**, and **6 misclassified** as Negative.



This confusion matrix shows the performance of **Model 2: KNeighborsClassifier** on sentiment classification (Positive, Neutral, Negative):

1. **Negative sentiment** is well-classified with 279 correct predictions out of 281 (excellent performance).
2. **Positive sentiment** is poorly classified: only 6 correct vs 25 misclassified as Negative.
3. **Neutral sentiment** is again weak: only 1 correct, with the rest misclassified (mostly as Negative).



This confusion matrix evaluates the **GradientBoostingClassifier** on a 3-class sentiment task (Positive, Neutral, Negative):

1. **Negative sentiment** is classified very accurately: 280 correct, only 1 misclassified.
2. **Positive sentiment** is often confused with Negative: 16 Positive samples misclassified as Negative.
3. **Neutral sentiment** is the weakest: only 2 correct out of 8, with several misclassified as Negative.

CONCLUSION:

Data Analysis Using Python course project applied machine learning across three domains: healthcare, audio classification, and sentiment analysis. Project 1 (Obesity Prediction) achieved 96.22% accuracy using Random Forest, supported by strong preprocessing and feature selection. Project 2 used CNNs for UrbanSound classification, showing high accuracy and generalization, with statistical tests confirming model reliability. Project 3 (Amazon Sentiment Analysis) reached 99% accuracy with Gradient Boosting, revealing a dominance of positive reviews. Each project demonstrated effective model selection, evaluation, and real-world applicability, showcasing strong ML skills and adaptability across diverse data types.