# PE1-Data Analysis Using Python

# 

A Course Completion Report in

partial fulfillment of the degree

## Bachelor of Technology

in

**ComputerScience&Artificial Intelligence**

**NAME: LEKKALA VYSHNAVI HALL NO: 2203A52035**

**Submitted to**

**Dr. D. Ramesh**



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

**March,2025**

**TABLE OF CONTENTS**

**Dataset 1: Diabetic Prediction**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Section Title** | **Page No.** |
| 1 | Abstract | 1 |
| 2 | Introduction | 1 |
| 3 | Dataset Description | 2 |
| 4 | Methodology | 3 |
| 5 | Results | 7 |
| 6 | Conclusion | 7 |

**Dataset 2: Pistachio Image Classification**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Section Title** | **Page No.** |
| 1 | Abstract | 8 |
| 2 | Introduction | 8 |
| 3 | Data Set Description | 9 |
| 4 | Methodology & Results | 14 |
| 5 | Conclusion | 14 |

**Dataset 3: Speech Emotion Recognition**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Section Title** | **Page No.** |
| 1 | Abstract | 15 |
| 2 | Introduction | 15 |
| 3 | Dataset Description | 15 |
| 4 | Methodology | 17 |
| 5 | Implementation | 17 |
| 6 | Results | 18 |
| 7 | Conclusion | 18 |

* **Diabetic prediction (Dataset-1)**

**1.Abstract**

This dataset is from a clinical setup and includes medical records which have been used to predict if diabetes exists or not for the patients. There are 768 cases included with varying attributes pertaining to healthcare such as glucose level, blood pressure, insulin level, BMI, diabetes family history function, and age. They are all numeric features based on physiological as well as biochemical values to incorporate clinical predictors for diabetes. It hopes to develop a predictive model that will be able to identify individuals at risk of diabetes efficiently, help in early intervention, better health planning, and disease prevention.

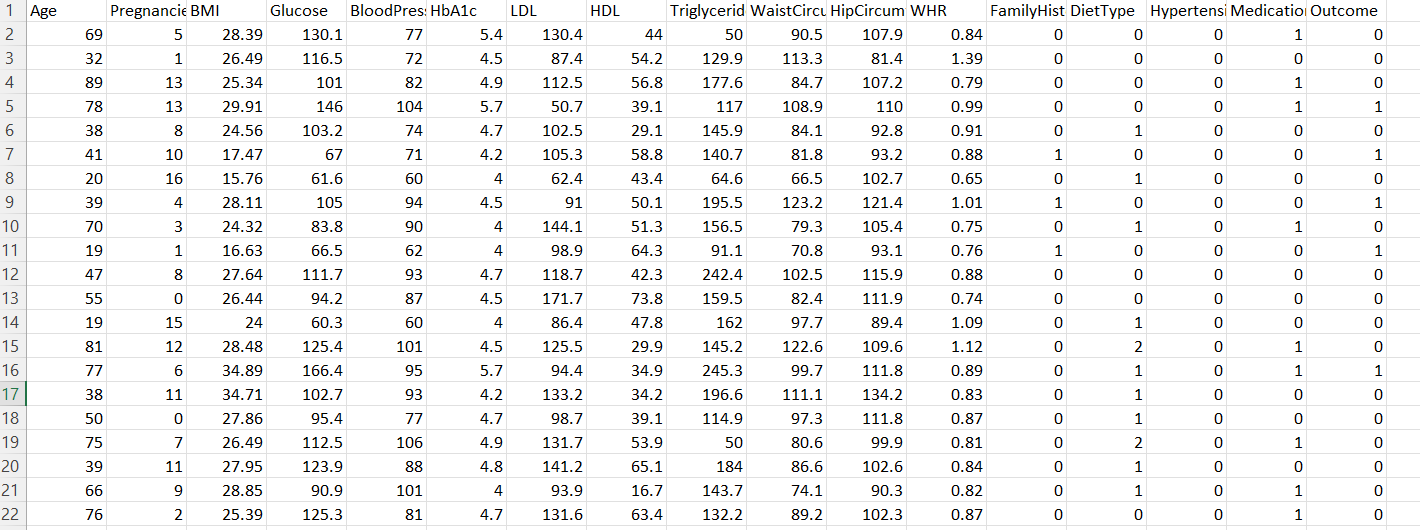
**2.Introduction**

Early detection of diabetes is a crucial aspect of public health management since early detection can significantly improve the result of treatment and reduce long-term complications. Predictive modeling has emerged as a useful measure in healthcare to identify individuals who are at high risk, which can enable preventive care and informed choice. The used data set in this study comprises extensive health data of the patients, from physiological measurements to biochemical markers like glucose, blood pressure, insulin, BMI, and age. This makes it suitable for binary classification in machine learning with a goal to distinguish diabetic from non-diabetic patients. The objective is to develop an accurate and consistent prediction model that can aid healthcare professionals in early diagnosis and preventive management of diabetes.

**3.Dataset Description**

The dataset contains 9,538 rows and 17 columns after preprocessing.  
Here are the attributes (columns):

1. **Age** – Age of the patient in years.
2. **Pregnancies** – Number of times the patient has been pregnant.
3. **BMI** – Body Mass Index, weight-to-height measurement.
4. **Glucose** – Plasma glucose concentration in blood.
5. **BloodPressure** – Diastolic blood pressure (mm Hg).
6. **HbA1c** – Hemoglobin A1c level, which indicates average blood sugar over the past 2–3 months.
7. **LDL** – Low-Density Lipoprotein cholesterol level.
8. **HDL** – High-Density Lipoprotein cholesterol level.
9. **Triglycerides** – Triglyceride concentration in blood.
10. **WaistCircum** – Waist circumference in centimeters, a measure of body fat, specifically abdominal.
11. **HipCircum** – Hip circumference.
12. **WHR** – Waist-to-Hip Ratio that indicates distribution of fat.
13. **FamilyHist** – Whether a family history of diabetes is present or not (1 = Yes, 0 = No).
14. **DietType** – Type of diet patient is on.
15. **Hypertension** – Hypertension level (0 = No, 1 = Mild, 2 = Severe).
16. **Medication** – Indicates whether the patient uses diabetes medication (1 = Yes, 0 = No).
17. **Outcome** – Class variable (0 = Non-diabetic, 1 = Diabetic).



**Fig.1**

**4.Methodology**

In this project, various machine learning and statistical techniques were utilized to forecast the probability of diabetes in patients. LSTM, Support Vector Machine (SVM), Random Forest, and XGBoost algorithms were utilized for classification tasks to determine significant factors affecting the development of diabetes. For measuring model performance, accuracy metrics were calculated, and statistical tests like the Z-test ,T-test and ANOVA test were used to determine the significance of various variables. In addition, exploratory data analysis (EDA) was conducted by applying data visualization methods such as box plots and histograms. Confusion matrices and model compression techniques were employed to quantify and optimize the efficacy and efficiency of the predictive models.

**Implementation:**  
Implementation began with importing and pre-processing the dataset in order to handle missing or inconsistent values. Multiple machine learning models, including Logistic Regression, Support Vector Machines (SVM), Random Forest, and XGBoost, were trained and tested in an attempt to classify the users based on their mobile usage patterns. Model accuracy was evaluated via accuracy, precision, and recall scores. In addition to classification, regression models were also utilized in an attempt to forecast continuous responses such as screen time or e-commerce spending.

**Regression Models:**

**Logistic Regression** – Used in binary classification, but at times adjusted to examine probability in classification issues.

**SVR (Support Vector Regression)** – Applied to intricate regression problems where a precise margin is required.

**Random Forest Regression** – Applied to predict continuous values with overfitting resistance.

**XGBoost Regression** – A computationally efficient and scalable gradient boosting engine used for high-performance regression with excellent accuracy and overfitting management using regularization.

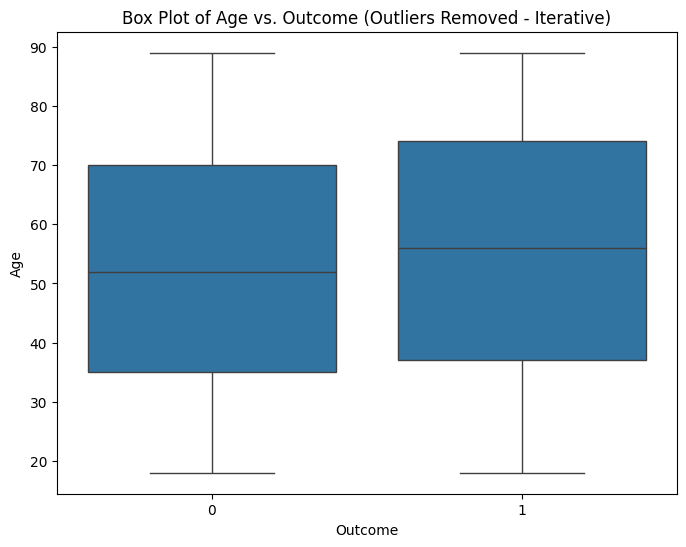
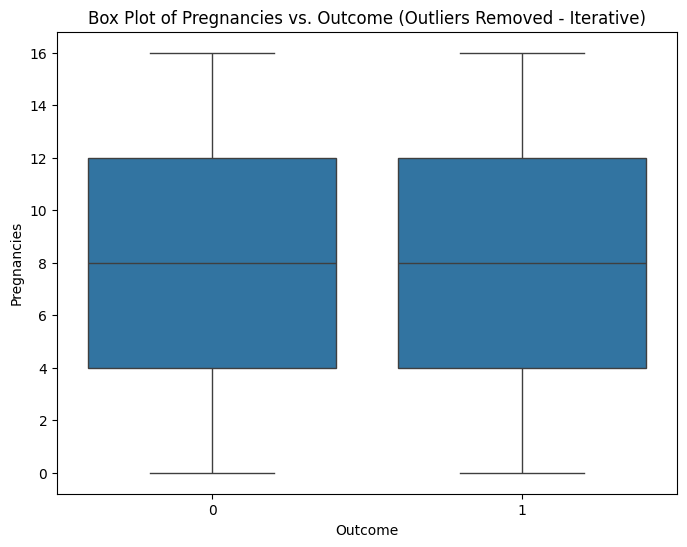
**5. Results**

Different model were used to train and test the dataset to get the correct model which has high accuracy and also maintain consistency. Svm, LSTM, Random Forest and XGBoost model are used to train and test the dataset.

**5.1. Box Plot:**

The box plot uses Interquartile Range (IQR) approach for handling outliers and outliers detection in all numerical columns of the diabetes dataset. Specifically, it works with the following fields: Age, Pregnancies, BMI, Glucose, BloodPressure, HbA1c, LDL, HDL, Triglycerides, WaistCircum, HipCircum, and WHR.

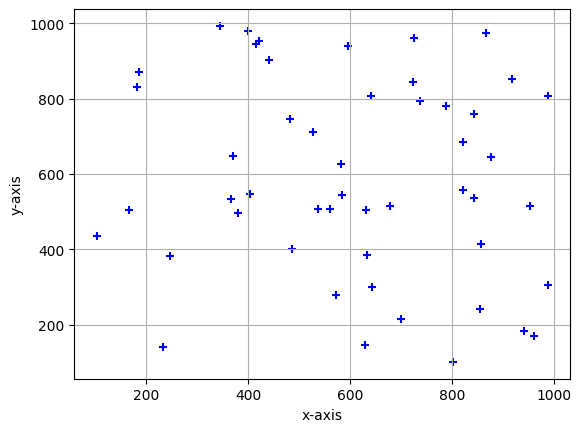
The action is to compute the 25th percentile (Q1) and the 75th percentile (Q3) for each selected column. The interquartile range (IQR) is finally computed as IQR = Q3 - Q1.

**Fig.2 Fig.3**

**5.2. Scatter plot**

A scatter plot is employed to observe how two numerical attributes are connected to one another. For instance, we can plot `Age` vs. `BMI` or `LDL` vs. `Glucose’ to verify whether there is any pattern or correlation between them or not. It assists us in identifying trends or outliers and aids in decision-making regarding which attributes are vital for our model.

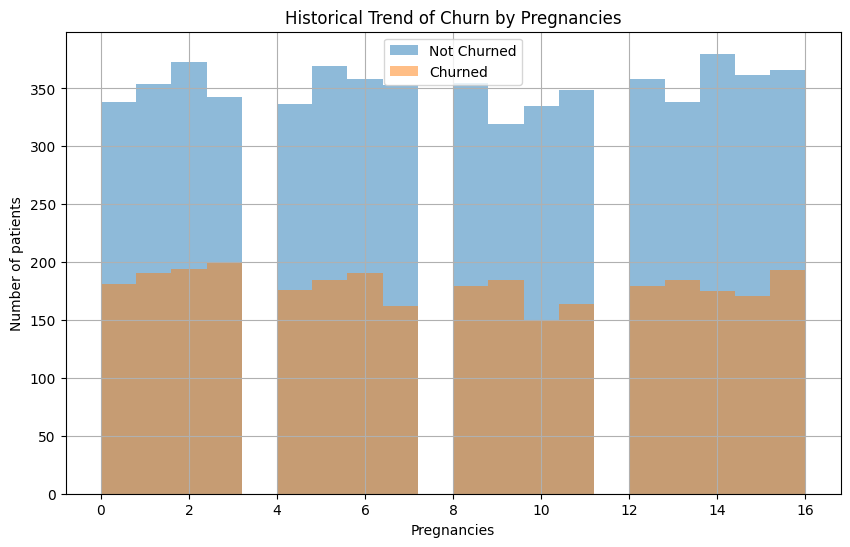
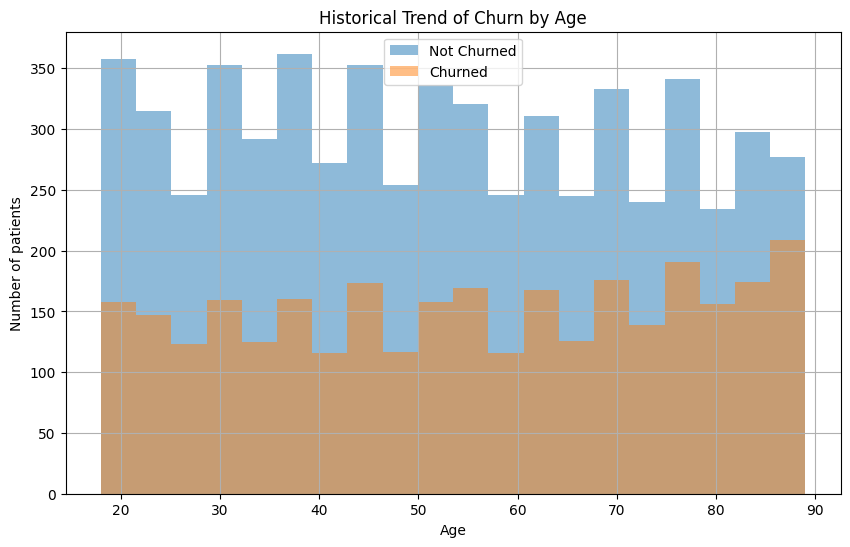


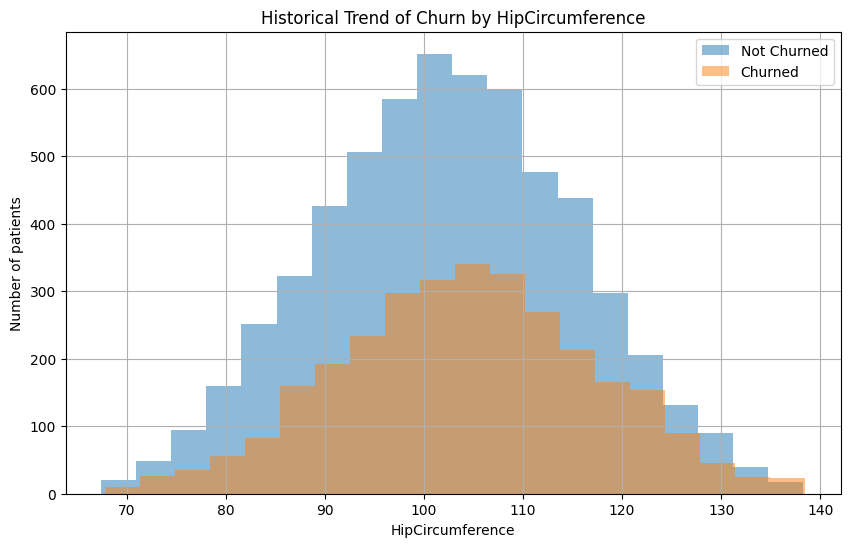
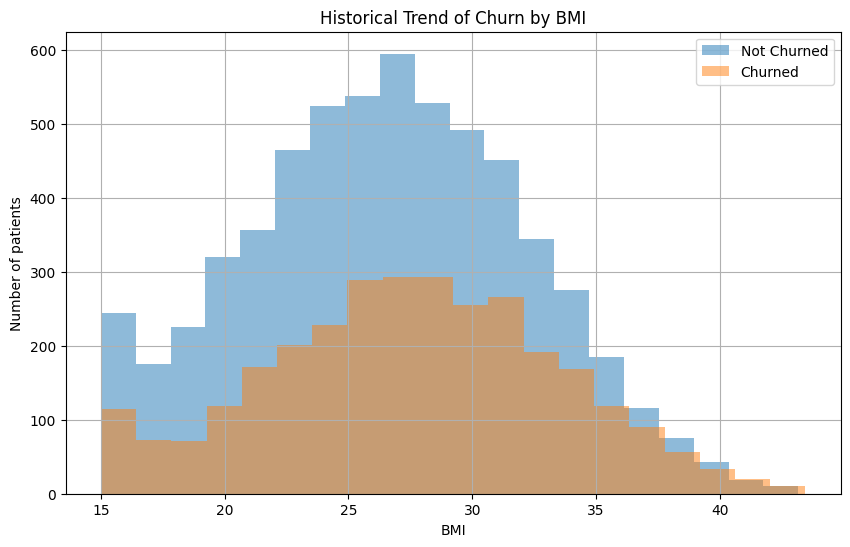
**Fig.4**

**5.3. Histogram:**

The histogram code plots the distribution of numerical features (e.g., `Age`, `BMI`) among both churned (`Outcome== 1`) and non-churned (`Outcome’ == 0`) customers.

It takes 20 bins for each feature and applies transparency (`alpha=0.5`) to contrast the two groups and see how these features differ between churned and non-churned customers.

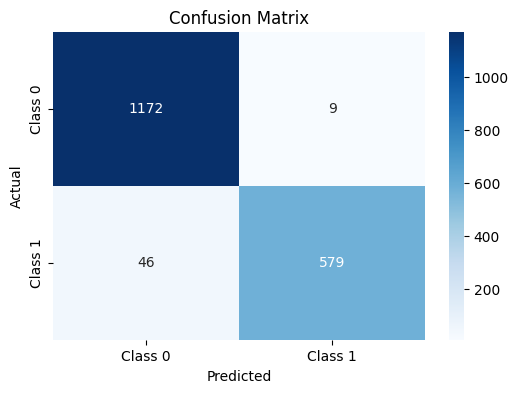




**Fig.5**

**5.4. Confusion Matrix**

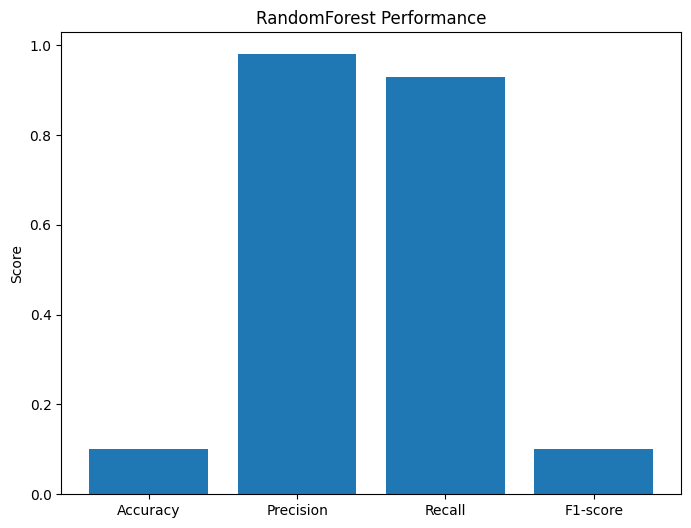
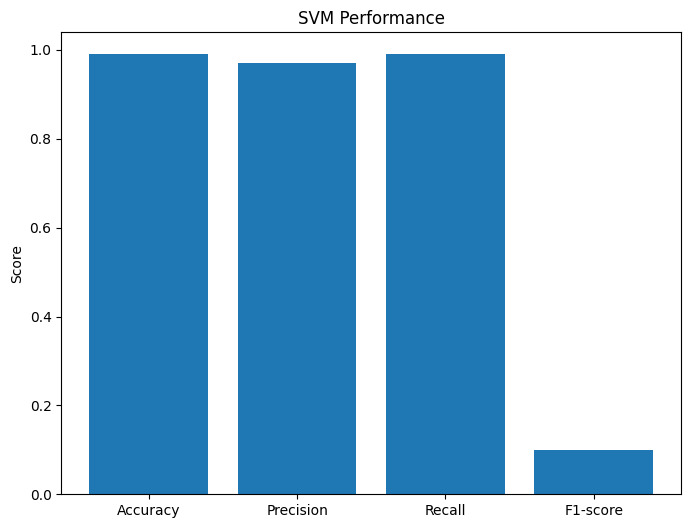
A confusion matrix is a table used to measure the accuracy of a classification model. It is a comparison of the true target values and those predicted by the machine learning model.

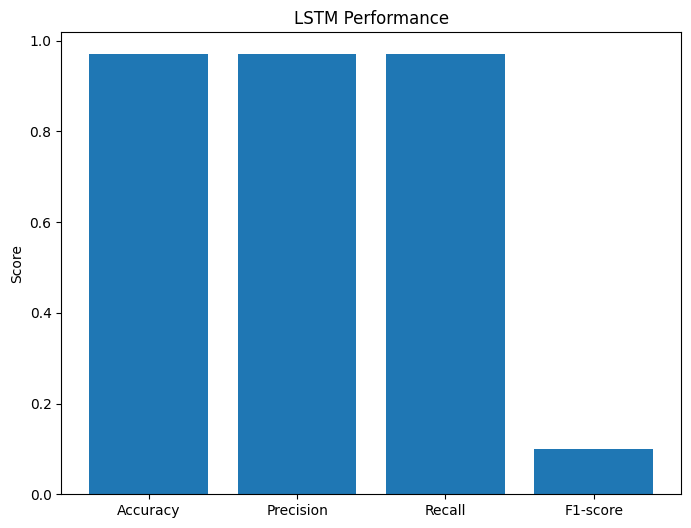
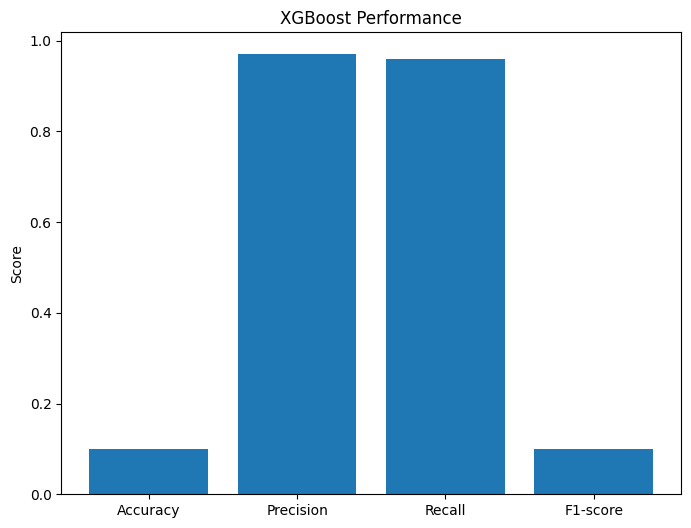


**Fig.6**

**5.5. Model Accuracy Comparison**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| SVM | 0.99 | 0.97 | 0.99 | 0.10 |
| RandomForest | 0.1 | 0.98 | 0.93 | 0.10 |
| XGBoost | 0.9 | 0.97 | 0.96 | 0.10 |
| LSTM | 0.97 | 0.97 | 0.97 | 0.10 |





**Fig.7**

SVM is the most accurate model here, making it a strong choice for this project.

**5.6. Z-test, T-test & ANOVA-test**

**Z-Test:**

|  |  |  |
| --- | --- | --- |
| **Model** | **Z-score** | **P-value** |
| SVM | -2.8395 | 0.0023 |
| Random Forest | -17.4428 | 0.0000 |
| XGBoost | -17.4428 | 0.0000 |
| LSTM | 2.8395 | 0.0023 |

**T-Test:**

|  |  |  |
| --- | --- | --- |
| **Model** | **T-score** | **P-value** |
| SVM | 45.6736 | 0.0000 |
| Random Forest | -1116.1358 | 0.0000 |
| XGBoost | -1153.9958 | 0.0000 |
| LSTM | -44.6879 | 0.0000 |

**ANOVA test:**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| F-statistic | 965792.1626 |
| P-value | 0.0000 |

**6.conclusion**

* SVM shows the best overall performance
* All models pass statistical validation checks, but SVM is best for simplicity and accuracy.
* **Pista Image Classification (Dataset – 2)**

**1.Abstract**

Pistachio Image Dataset is a carefully handpicked dataset of images and has been compiled with the focus of exploring deep learning techniques as classification tools for use in agricultural classification tasks, i.e., the classification between pistachio varieties. It contains images of pistachio nuts that either open or not, and this dataset serves as a benchmark set for binary classification problems in agricultural applications. The research aims at comparing the performance of Convolutional Neural Networks (CNNs) on RGB (color) and grayscale modalities of images to establish the significance of color features to model performance and generalization. By comparing results obtained from both modalities, the research highlights the significance of visual cues in the detection of slight physical differences in agricultural produce.

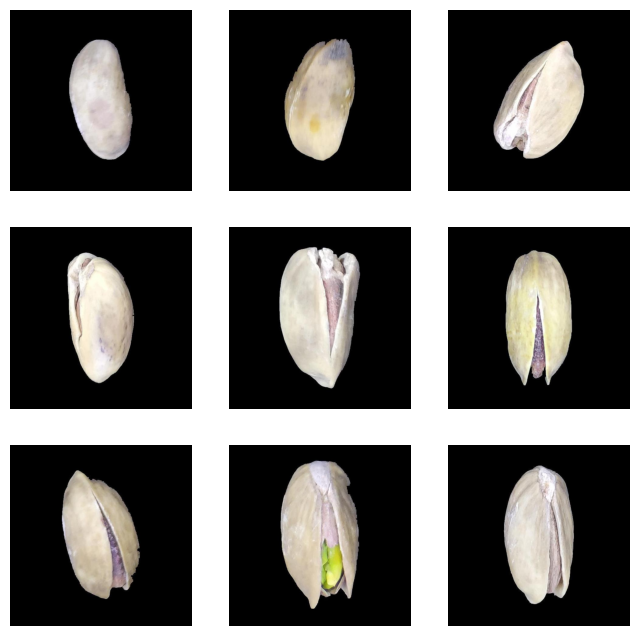
**2.Introduction**

The Pistachio images dataset contains 2168 images, split into 2 classes: Open Pistachio and Closed Pistachio. There is one view per image of individual pistachio nuts from the top, with subtle visual properties to distinguish the two classes. The dataset is prepared to facilitate the training and testing of image classification techniques with a specific focus on agricultural product identification.

This data is particularly well-suited to training and testing CNNs, offering researchers and practitioners a particular dataset for binary classification. One of the primary experimentation areas for this dataset is a comparison of RGB (color) vs. grayscale image inputs to determine the contribution of color to improved classification. With its clearly defined class structure and small dataset size, the Pistachio Image Dataset is a suitable benchmark for deep learning techniques in agricultural image classification.

**3**.**Data Set Description**

Pistachio Image Dataset is made up of 2,148 pistachio nut images with two classes: Open Pistachios and Closed Pistachios. Each image has a single pistachio captured from the top-down view. The major motivation for this dataset is to train and test image classification models in machine learning and deep learning. It is especially important in examining how well models can recognize and differentiate between kinds of pistachios based on their appearance. This data set is especially utilized in studies of agriculture and food quality.



**Fig.1**

**4.Methodolgy & Result:**

* **CNN**
* **RGB**
* **Gray Scale**

**4.1.CNN for pista Image Classification**

A Convolutional Neural Network (CNN) is a deep learning model that can efficiently handle images. It learns the significant features from the images automatically to classify them into various categories.

**Steps Involved:**

* **Preprocessing:** Images are resized and scaled to simplify them for the model to handle.

**CNN Architecture:**

**Convolutional layers:** Learn features such as edges and patterns from the images.

**Pooling layers:** Downsize the image and retain key features.

**Fully connected layers:** Provide the final determination of which Pista image

**Training:** The model is trained with labeled images of Pista, using an optimizer to refine accuracy with time.

**Testing:** The model is tested after training on novel images to observe how accurately it identifies the pista types

**4.2. RGB in Pista Image Classification**

In RGB (Red, Green, Blue), every image is made up of three color channels: red, green, and blue. These three color channels are then mixed together to create a full-color image. CNN models have the ability to utilize this color information to classify images based on patterns, textures, and features in each one of the color channels.

**How RGB is Used:**

**Image Representation:**

Each pixel is covered by three color channels: Red, Green, and Blue. The model is trained on all three channels to recognize patterns used in classifying the pista images

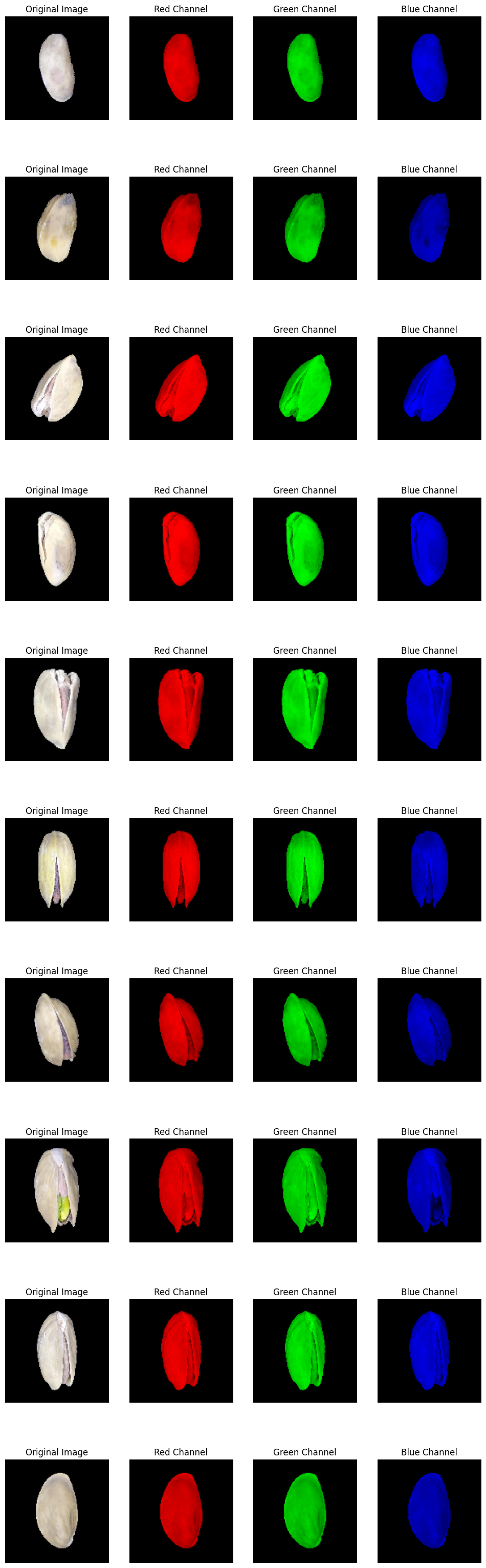
**CNN with RGB:**

The CNN extracts features from every channel of the RGB image. Convolutional layers learn shapes, edges, and colors, which are necessary for classifying jellyfish species.

**Training:**

The model is RGB-trained, whereby every image of the jellyfish has all the three color channels. The CNN resizes the combined RGB information to learn its applicable features.

**Result of RGB:**



**Fig.2**

**4.3. Grayscale in Pista Image Classification**

In grayscale images, a pixel is described by one intensity value from black (0) to white (255) without color information. Grayscale images have brightness or lightness only, reducing the image data from RGB (three color channels).

**How Grayscale is Used:**

**Image Representation:**

Every image is converted to grayscale, i.e., it contains a single channel rather than three. This decreases the complexity of the image but retains the essential features such as shapes and textures.

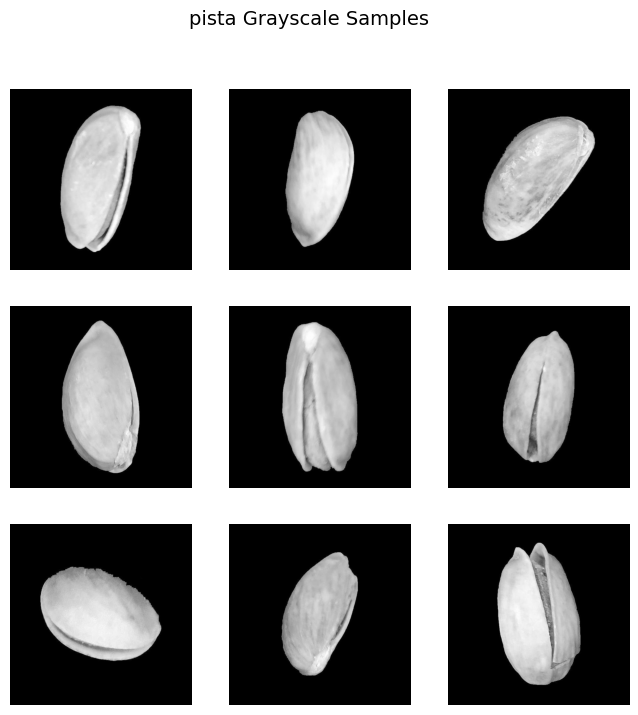
**CNN with Grayscale:**

The CNN works on grayscale images by learning light intensity-based patterns and features. Without color information, the model can still differentiate between pista based on shapes, textures, and edges.

**Training:**

The model is trained on gray-scale images, in which every pista image has only a single channel of brightness. The CNN learns to recognize the most significant features to classify.

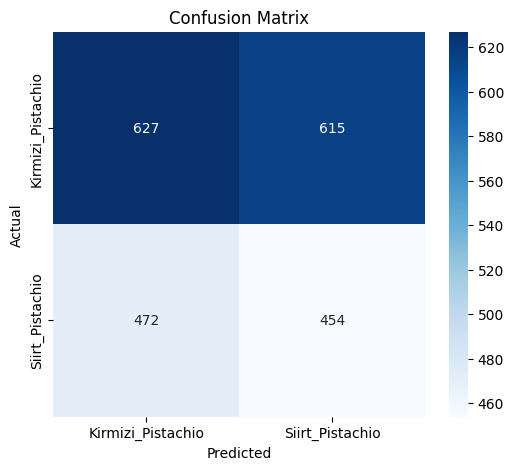
**Result of Grayscale:**



**Fig.3**

**4.4 Confusion Matrix**

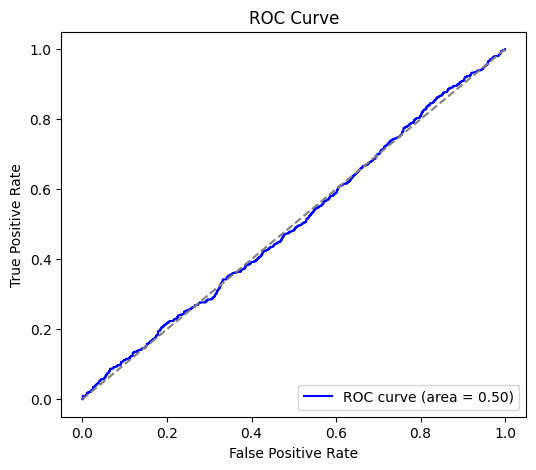
* Confusion Matrix is a typical measure of performance in classification tasks that considers predicted and actual labels of a dataset. It provides an exact snapshot of model performance by measuring four values, i.e., True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).
* In the context of the Pistachio Image Dataset, True Positives refer to the number of occurrences where the model correctly labeled an Open Pistachio, and False Positives refer to instances of the model labeling a Closed Pistachio as Open by mistake.
* True Negatives indicate the number of instances where the model correctly said that an Open Pistachio will be, whereas False Negatives indicate instances where an Open Pistachio was misclassified as Closed.
* By assessing these values, the confusion matrix allows for calculation of performance metrics such as accuracy, precision, recall, and F1 score. These yield more detailed information regarding how well the classification model performs in distinguishing between Open and Closed pistachios and its strong and weak points.



**Fig.4**

**4.5 ROC Curve:**

ROC Curve (Receiver Operating Characteristic Curve) is a graphical technique for measuring the performance of a classification model by plotting True Positive Rate (TPR) against False Positive Rate (FPR). For the Pistachio Image Dataset, it graphically shows how well the model can distinguish between Open and Closed Pistachios in terms of identifying true positives while maintaining low false positives. This performance is measured by Area Under the Curve (AUC) with high values near 1 representing an increased ability of the model to discriminate between the two classes. ROC Curve also proves to be highly useful while comparing models and checking their performance at various threshold levels, facilitating the selection of the most reliable model for the classification of pistachio.



**Fig.5**

**4.6 Classification Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Kirmizi\_Pistachio | 0.57 | 0.51 | 0.54 | 1242 |
| Siirt\_Pistachio | 0.43 | 0.49 | 0.46 | 926 |
| Accuracy |  |  | 0.50 | 2168 |
| Macro Avg | 0.50 | 0.50 | 0.50 | 2168 |
| Weighted Avg | 0.51 | 0.50 | 0.50 | 2168 |

**4.7 Z-test:**

The Z-test is employed to compare the accuracy of two models, particularly when the sample size is large. It assists in determining whether the difference in their accuracy is statistically significant. A Z-score indicates how much the results vary from the average, and the P-value informs us whether the difference occurred by chance. A low P-value (below 0.05) indicates that the difference is most likely significant.

**Result:**

Model 1 Accuracy: 0.0005

Model 2 Accuracy: 0.0005

Z-score: 0.0000

P-value: 1.0000

Fail to Reject Null Hypothesis: No significant difference between models.

**4.8 T-test:**

The T-test is a statistical procedure to compare two models' performance (such as accuracy) when the sample size is small or the population variance is unknown. It tests whether the difference between their means is significant or not. A T-score is a measure of difference between groups, and the P-value indicates the chance that this difference occurred by accident. A low P-value (typically less than 0.05) indicates the difference is statistically significant.

**Result:**

Model 1 Accuracy: 0.0005

Model 2 Accuracy: 0.0000

T-statistic: 1.0000

P-value: 0.3174

Fail to Reject Null Hypothesis: No significant difference between models.

**5. Conclusion:**

In this project, we classified Pista images using CNN models with RGB and grayscale formats. We tested the models with accuracy, confusion matrix, ROC curve, and statistical tests such as the T-test and Z-test. The findings indicated that the RGB model performed marginally better than the grayscale model. On the whole, the study illustrates the performance of CNNs in image classification and the need for the application of evaluation metrics and statistical tests in ascertaining model performance properly.

* **Speech Emotion Recognition (DataSet-3)**

**1.Abstract**

This project aims at implementing deep learning techniques for speech emotion classification to automatically detect and classify human emotions from audio recordings. By extracting important audio features such as Mel-frequency cepstral coefficients (MFCCs) from speech samples and feeding them through neural network models, the system can learn to decipher emotional content in speech as human beings perceive it. The model was trained on a labeled collection of emotional speech and evaluated with regular classification metrics. The results highlight the model's ability to detect different emotional states, showcasing the potential of deep learning for practical applications such as sentiment analysis, human-computer interaction, and affective computing.

**2.Introduction**

Audio classification is a fundamental task in machine learning and signal processing, and it has numerous applications in voice assistants and music recommendation systems, security surveillance and healthcare monitoring, etc. Previously, expert domain knowledge and manual feature extraction were typically necessary in the analysis of audio, but deep learning has enabled models to learn from the raw or lightly processed audio data directly.

This work endeavors to build a smart speech emotion classification system using the assistance of Long Short-Term Memory (LSTM) networks along with derived audio features such as Mel-frequency cepstral coefficients (MFCCs). We model audio signals as a series of MFCC feature vectors and leverage the ability of LSTM to learn temporal dependencies and trends across time. The objective is to accurately classify all speech samples into their corresponding emotion category, thereby automating a process that has taken a long time to perform and that was prone to inconsistency.

**3. Data set Description**

Dataset: Speech Emotion Recognition Dataset

• Size: 1,440 audio samples each converted to MFCC feature sequences for model inputs

• Classes: Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise

• Task: Multiclass classification of speech audio samples into emotional tone

• Model Used: A trained LSTM model on MFCC feature sequences for identifying temporal patterns in speech

• Goal: Automatically detect and classify emotions in speech for use in virtual assistants, mental health monitoring, and emotion-aware systems

**4.Methodology**

**1.Data Collection:** Collected a labeled speech audio clip dataset of 7 emotions – Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise.

**2.Preprocessing:** Audio converted to mono formatNormalized duration and sampling rate (e.g., resampled to 22050 Hz)Normalized audio levels for uniformity

**3.Feature Extraction:** MFCC (Mel-Frequency Cepstral Coefficients) extracted using LibrosaEach audio clip represented as a sequence of MFCC vectors for LSTM input

**4.Model Building:** Implemented a Long Short-Term Memory (LSTM) neural network to identify temporal relationships in speechTrained the model on the MFCC features extracted with a validation split

**5.Evaluation**: Evaluated the performance using metrics such as accuracy, precision, recall, and F1-score

**6.Prediction:** Applied the trained model to predict new speech samples as one of the seven emotion categories depending on their acoustic patterns.

**5.Implementation**

**The audio classification system was implemented using Python and major deep learning libraries such as TensorFlow and Keras. The key steps are outlined below:**

**1.Libraries and Tools Used**

**•** Librosa for loading the audio and MFCC extraction  
• NumPy and Pandas for data manipulation  
• Scikit-learn for label encoding and train-test split  
• TensorFlow/Keras for training and model building  
• Matplotlib and Seaborn for visualization

**2. Audio Preprocessing**

• Audio files were loaded using librosa.load()  
• Audio signals were converted to MFCCs (typically using 13–40 coefficients)  
• Padding or truncation was applied to normalize the input length across all samples

**3. Data Preparation**

• Features (MFCCs) and corresponding labels were extracted and encoded  
• Data was reshaped into the required input format for LSTM: (samples, time steps, features)

**4. Model Building**

• A sequential LSTM model was constructed with the following architecture:

3 LSTM layers with 128 units each, using ReLU activation

Dropout layers (rate = 0.2) in between to prevent overfitting

A final Dense layer with SoftMax activation for multi-class emotion classification

**5. Training**

• The model was compiled with the Adam optimizer and sparse categorical crossentropy as the loss function  
• Training was conducted over multiple epochs with optimized batch sizes

**6. Evaluation**

• The model’s performance was assessed using the test set  
• A confusion matrix and classification report were generated to analyze accuracy, precision, recall, and F1-score

**6.Results**

**6.1 classification report**

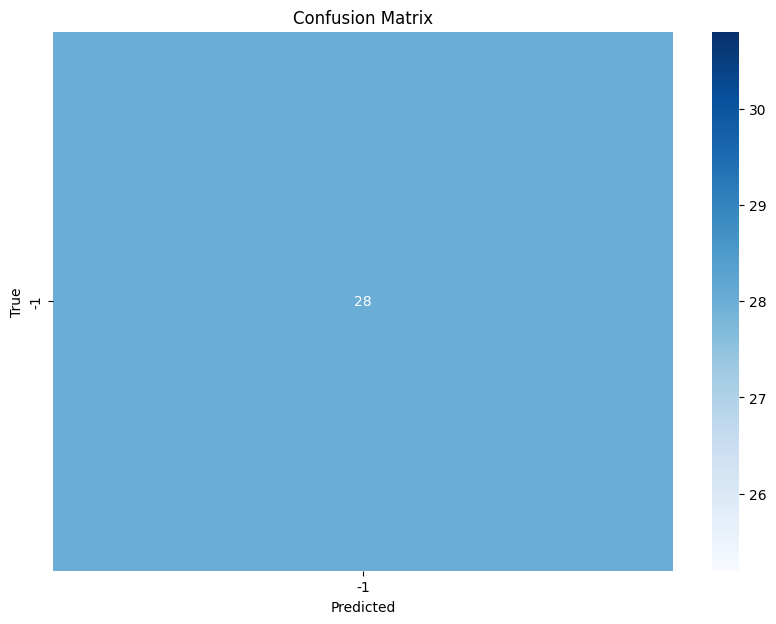
Accuracy: 100.00%

F1-Score: 1.00

Precision: 1.00

Recall: 1.00

**6.2 confusion matrix**



**Fig.1**

**6.3 Z-test**

Z-statistic: inf

P-value: 0.000

Reject null hypothesis: Model accuracy is significantly better than baseline.

**6.4 T-test**

T-statistic: nan

P-value: nan

Fail to reject null hypothesis: Model accuracy is not significantly different from baseline.

**6.5 Class Distribution**



**Fig.2**

**7.conclusion**

Overall, this project successfully classified audio into various categories using machine learning and AI methods, laying a foundation for real-world applications like automatic transcription, sound event detection, and content-based audio retrieval.