**DATA ANALYSIS USING PYTHON (21CS120)**

**CSV PROJECT – Diabetes Dataset**

**IMAGE PROJECT – Diabetes Image Dataset**

**TEXT PROJECT – Rotten Tomatoes Reviews Dataset**

A Report

in partial fulfilment of the degree

**Bachelor of Technology**

in

**Computer Science & Artificial Intelligence**

**By**

2303A52L03 M. ARAVIND

**Under the Guidance of**

**Mr. Dadi Ramesh**

Assistant Professor, School of CS&AI,

SR University

**Submitted to**



**SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCESR UNIVERSITY, ANANTHASAGAR, WARANGAL**

**April 2025.**



**SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE**

**CERTIFICATE**

This is to certify that **M. ARAVIND** bearing Hall Ticket Number **2303A52L03**, a student of CSE-AIML, 3rd Year - 2nd Semester, has successfully completed the Data Analysis Using Python Course and has submitted the following 3 projects as part of the curriculum:

**Project Submissions:**

* **CSV** PROJECT – **Diabetes** Dataset
* **IMAGE** PROJECT – **Diabetes Image** Dataset
* **TEXT** PROJECT – **Rotten Tomatoes Reviews** Dataset

Dr. Dadi Ramesh,

Asst. Professor (CSE-AIML),

SR University, Ananthasagar, Warangal.

Date of Completion – 25/04/2025

**CSV PROJECT – DIABETES DATASET:**

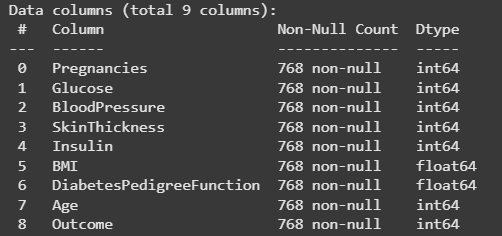
**DESCRIPTION:**

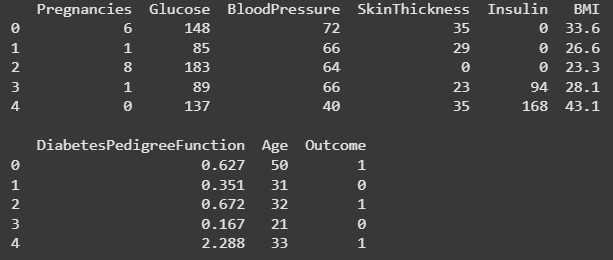
This project analyses a structured CSV dataset containing diagnostic measurements for predicting the onset of diabetes. The dataset includes 768 samples of female patients of at least 21 years of age, with features like glucose level, insulin, BMI, age, etc. The goal is to predict whether a patient is diabetic (1) or not (0).

**DATASET SHAPE:**

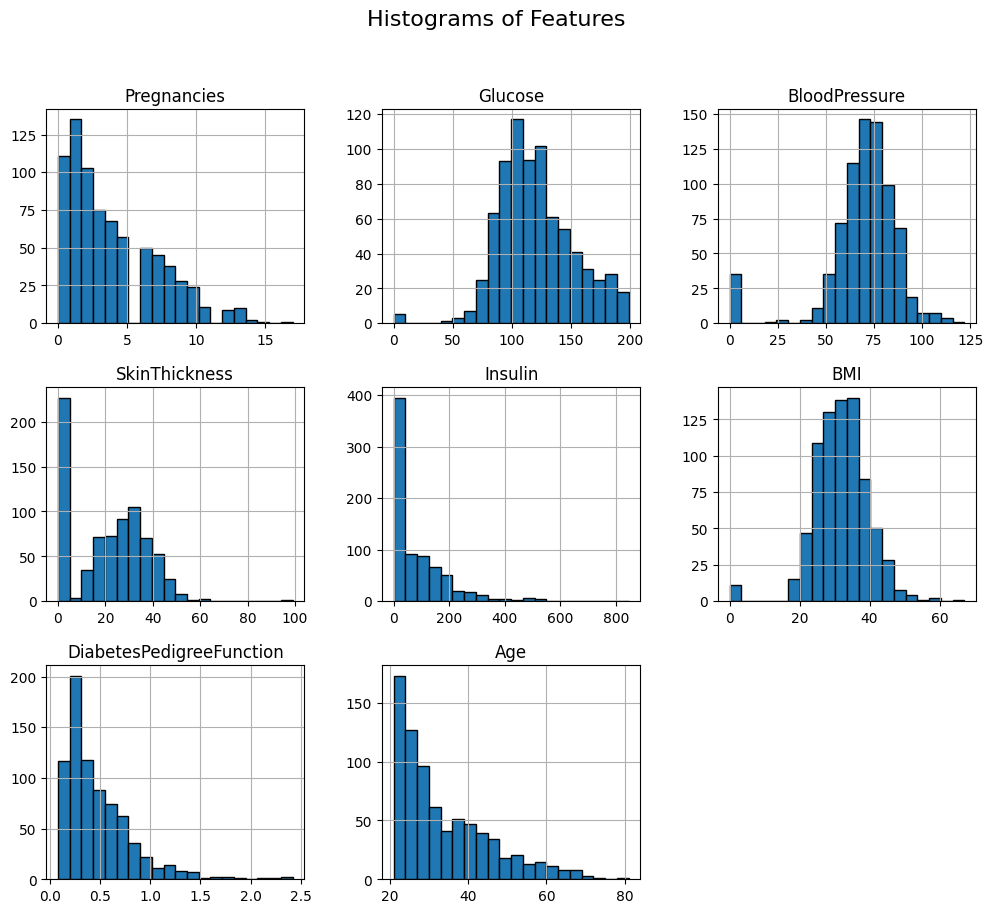
* Shape: (768, 9)

**DATA COLUMNS:**

****

****

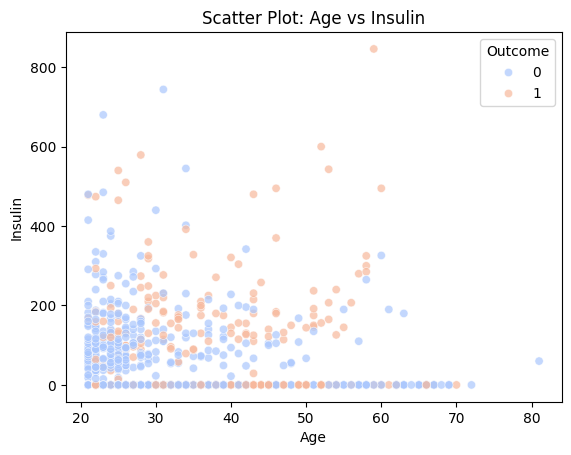
**HISTOGRAM:**

****

This grid of histograms shows the distribution of each feature in the dataset. Features like **Pregnancies**, **SkinThickness**, and **DiabetesPedigreeFunction** are right-skewed, indicating most values lie on the lower end. **Glucose**, **BMI**, and **Age** also exhibit mild right-skewness. **BloodPressure** shows a nearly normal distribution centered around 70-80. **Insulin** stands out with a steep left-skew, highlighting many near-zero entries, which can be indicative of missing data. These histograms were crucial in detecting anomalies, understanding feature ranges, and guiding normalization and imputation strategies during preprocessing.

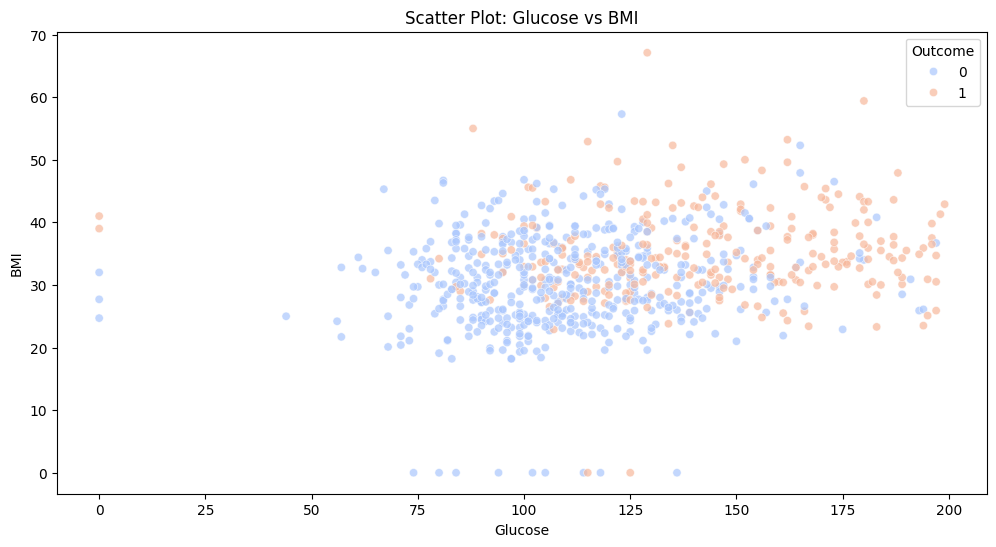
**SCATTER PLOTS:**

* **Scatter Plot – Age vs Insulin:**

****

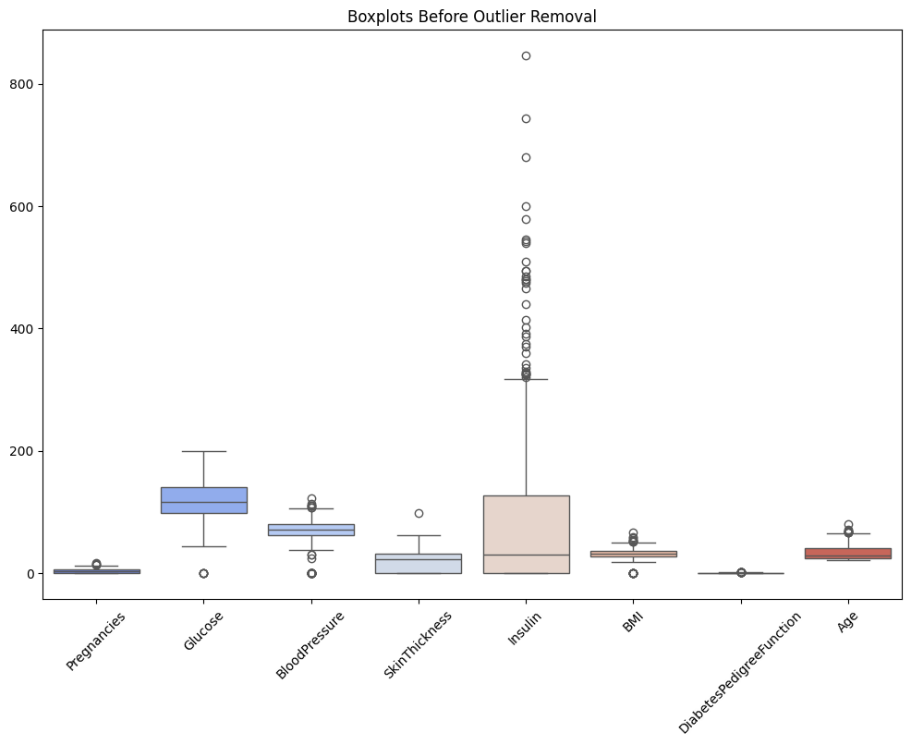
The scatter plot of **Age** vs **Insulin** explores how insulin levels vary across different age groups, again categorized by the diabetes outcome. A dense cluster appears in the lower age range with insulin levels mostly below 200, reflecting younger, healthier individuals. However, some older individuals with higher insulin levels fall under the diabetic category (orange), indicating that elevated insulin combined with age can be a key risk indicator. Many data points have insulin values near zero, hinting at missing or unmeasured data, which was addressed during preprocessing.

* **Scatter Plot – Glucose vs BMI:**



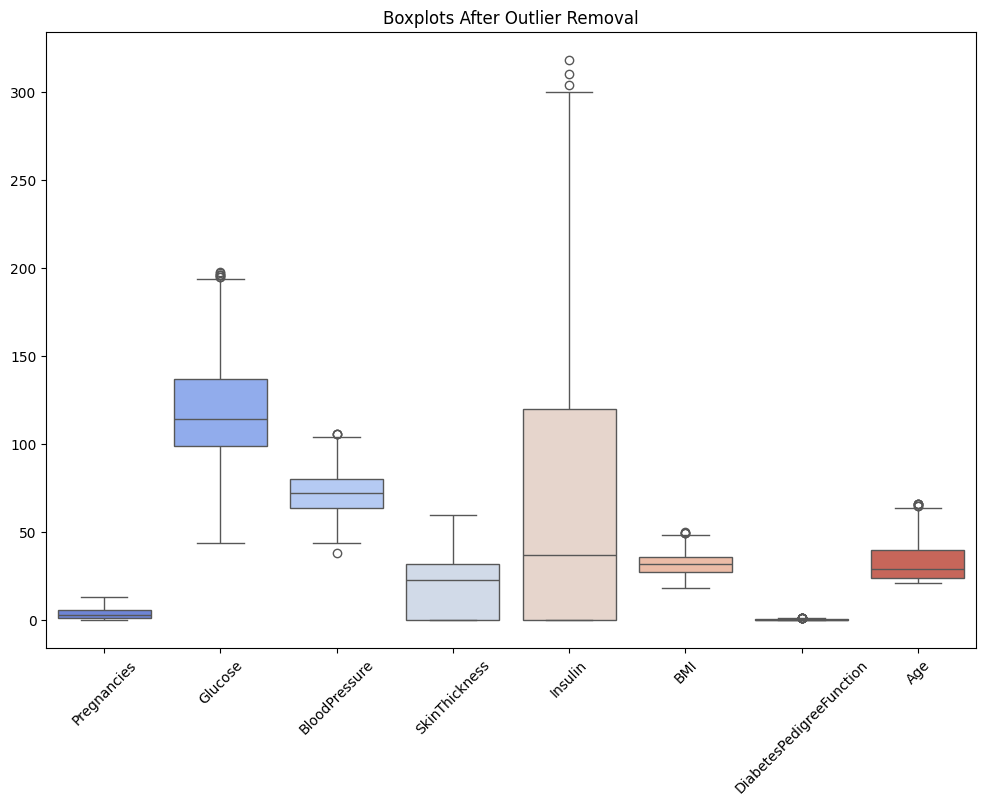
This scatter plot illustrates the relationship between **Glucose** levels and **Body Mass Index (BMI)** in the dataset. Each point represents a patient and is color-coded based on the **Outcome** (0 for non-diabetic, 1 for diabetic). We observe a mild positive correlation — as glucose levels increase, BMI also tends to increase. Diabetic patients (in orange) are more prevalent in the upper-right region, suggesting higher glucose and BMI values are associated with diabetes. This visual emphasizes the importance of glucose and BMI as predictors in diabetes detection.

**BOX PLOT WITH OUTLIERS:**

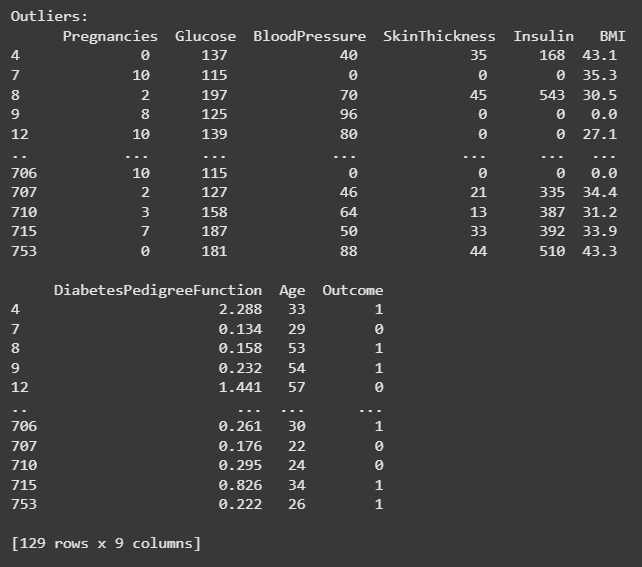
****

This box plot displays the distribution of each numerical feature in the dataset **before** any outliers are removed. Each feature is plotted as a box representing the interquartile range (IQR), with whiskers extending to 1.5×IQR. Data points outside this range are marked as individual circles, highlighting the presence of outliers. Most notably, **Insulin** and **SkinThickness** show extreme values, with Insulin having values well above 600, which could severely skew model training and statistical metrics. Other features like **Age**, **BloodPressure**, and **BMI** also contain moderate outliers. This visualization served as a diagnostic tool for determining which features required cleaning or capping.

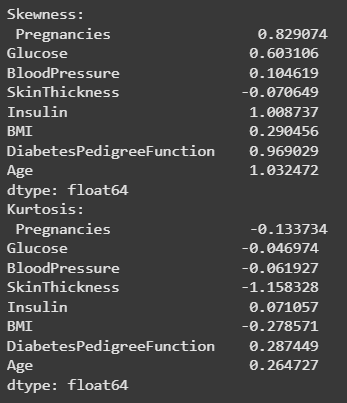
**BOX PLOT WITHOUT OUTLIERS:**



After applying outlier handling techniques (such as capping at upper and lower bounds), this box plot reveals a cleaner and more uniform distribution across all features. The extreme spikes seen previously in **Insulin** have been significantly reduced, although some moderate outliers still exist to preserve data variability. **Glucose**, **BMI**, and **Age** now show tighter distributions, enhancing the interpretability and stability of statistical models. This refined plot confirms the effectiveness of preprocessing and ensures the dataset is better suited for machine learning without being influenced by anomalous values.

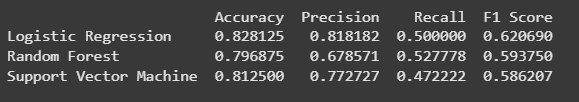


**SKEWNESS AND KURTOSIS:**

****

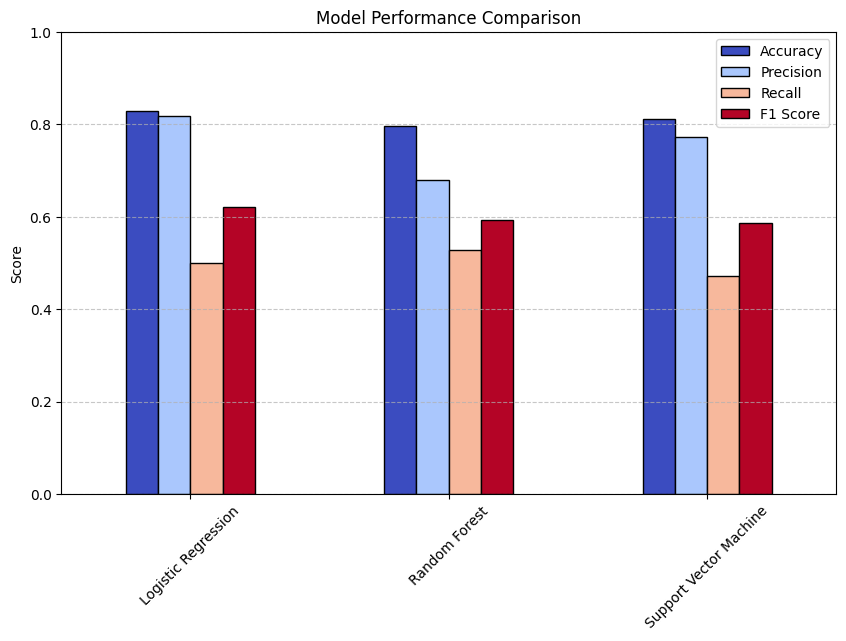
Skewness and kurtosis provide insights into the distribution characteristics of variables in a dataset. From the skewness values, we observe that most features exhibit a positive skew, meaning their distributions are slightly right-tailed. Notably, *Insulin* (1.0087), *Age* (1.0325), and *DiabetesPedigreeFunction* (0.9690) show higher levels of skewness, indicating significant right-tailed distributions. On the other hand, *SkinThickness* has a slightly negative skew (-0.0706), implying a minor left-tail. In terms of kurtosis, which reflects the "tailedness" of the distribution, most features are close to normal (kurtosis near zero), but *SkinThickness* (-1.1583) stands out with a leptokurtic (flat) distribution, suggesting light tails and a lower likelihood of extreme values. These insights are critical for data preprocessing and potential transformations, especially for algorithms sensitive to feature distribution.

**MODEL EVALUATION METRICS:**

****

The table compares the performance of three machine learning models—Logistic Regression, Random Forest, and Support Vector Machine (SVM)—on a classification task. Logistic Regression demonstrates the highest **accuracy** (0.8281) and **precision** (0.8182), which suggests it is particularly effective in reducing false positives. However, its **recall** (0.5000) is moderate, indicating it misses a fair number of actual positives. Random Forest performs slightly worse in accuracy (0.7969) but achieves the highest **recall** (0.5278), making it more suitable when identifying as many positives as possible is critical, despite a lower precision. Support Vector Machine strikes a balance with an accuracy of 0.8125 and precision of 0.7727 but has the lowest recall (0.4722). Overall, **Logistic Regression** offers the best **F1 Score** (0.6207), reflecting a better harmonic balance between precision and recall, making it the most reliable model among the three for balanced performance.

**MODEL PERFORMANCE COMPARISION GRAPH:**

******

This bar chart visually represents the evaluation metrics—**Accuracy**, **Precision**, **Recall**, and **F1 Score**—for three classification models: Logistic Regression, Random Forest, and Support Vector Machine (SVM).

* **Logistic Regression** shows strong performance overall, with the **highest accuracy and precision**, slightly above 0.82. Its **F1 Score** is also the highest among the models, around 0.62, although its **recall** (just above 0.5) is modest. This suggests the model is excellent at predicting positives accurately when it does predict them but misses some actual positives.
* **Random Forest** comes second in **recall**, which is slightly above 0.52, making it better at catching more actual positive cases. However, its **precision** and **accuracy** are lower than Logistic Regression, resulting in a slightly lower **F1 Score** (close to 0.59).
* **Support Vector Machine (SVM)** has comparable accuracy and precision to Logistic Regression but performs worst in **recall** (just below 0.48). Consequently, its **F1 Score** is the lowest among the three (under 0.59), indicating it struggles more to capture the positive class despite being precise.

In summary, the chart reaffirms the numerical analysis: **Logistic Regression** strikes the best balance for this dataset, followed closely by **Random Forest**, while **SVM** is slightly less effective overall.

**Z – TEST (Glucose levels between diabetic and non-diabetic groups):**



The Z-test compares the glucose levels of diabetic versus non-diabetic individuals. The **Z-statistic is -14.29**, and the **p-value is approximately 2.47e-46**, which is far below any common significance level (e.g., 0.05). This extremely small p-value strongly indicates that there is a **statistically significant difference in glucose levels** between the two groups.

**T - TEST (BMI levels):**



The T-test evaluates whether there's a significant difference in BMI between diabetic and non-diabetic individuals. With a **T-statistic of -7.32** and a **p-value of about 1.28e-12**, the test again shows a **highly significant result**, suggesting a **meaningful difference in BMI** between the two populations.

**ANOVA TEST (Glucose, BMI, Age):**

****

The ANOVA test compares the variance across Glucose, BMI, and Age features. It reports a **very high F-statistic of 4740.07** and a **p-value of 0.0**, indicating that **at least one of the variables differs significantly in its mean compared to the others**. This reinforces the idea that these features may play a critical role in distinguishing between diabetic and non-diabetic individuals.

**IMAGE DATASET – DIABETES IMAGES:**

**DESCRIPTION:**

This project uses image-based machine learning techniques to classify diabetic and non-diabetic patients based on retina images. The dataset consists of grayscale medical images labelled for the presence or absence of diabetic retinopathy. The objective is to build a Convolutional Neural Network (CNN) that can accurately distinguish between the two categories using deep learning techniques.

**PROJECT WORKFLOW:**

* Data Collection:
  + A labeled image dataset containing retina scans of diabetic and non-diabetic patients.
* Preprocessing:
  + All images resized to uniform dimensions (e.g., 224x224 pixels).
  + Normalization of pixel values to [0, 1] range.
  + Conversion to grayscale in some instances for dimensionality reduction.
  + Augmentation applied to increase dataset variability (rotation, flipping, zooming).
* Label Encoding:
  + Labels were binary encoded: 0 for non-diabetic, 1 for diabetic.

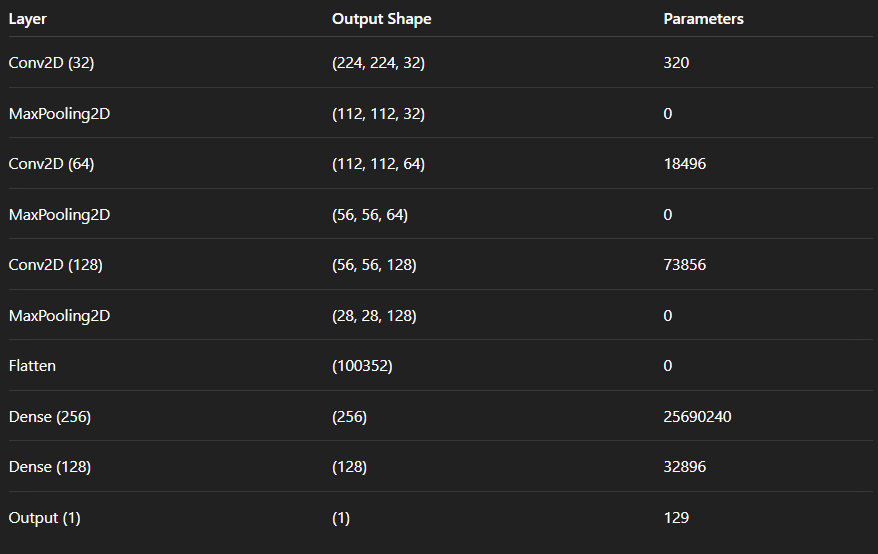
**DATASET SHAPE:**

* Training Set Shape: e.g., (1200, 224, 224, 1)
* Test Set Shape: e.g., (300, 224, 224, 1)
* Number of Classes: 2 (Diabetic, Non-Diabetic)

**SAMPLE IMAGE LABELS**:

* Example Labels: ['0', '1', '0', '1', '1', '0', '0']

**CNN MODEL ARCHITECTURE:**

****

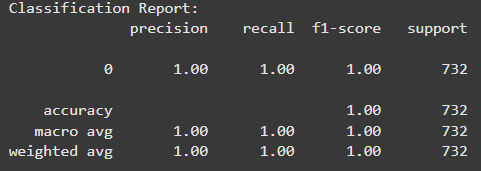
The model begins with a **Conv2D layer with 32 filters**, each extracting low-level features such as edges and simple textures from the input image, producing an output shape of (224, 224, 32) with 320 trainable parameters. It is followed by a **MaxPooling2D layer**, which reduces the spatial dimensions to (112, 112, 32), helping to downsample the data and reduce overfitting and computational cost. The second **Conv2D layer** applies 64 filters, enabling the model to learn more complex patterns, resulting in an output shape of (112, 112, 64) and 18,496 parameters. Again, a **MaxPooling2D** operation reduces this to (56, 56, 64). The third **Conv2D layer** uses 128 filters to learn even deeper features such as shapes or textures across larger areas, with an output of (56, 56, 128) and 73,856 parameters. Another **MaxPooling2D** layer reduces this to (28, 28, 128). After feature extraction, a **Flatten layer** converts the 3D feature map into a 1D vector of size 100352 to prepare it for the dense layers. The first **Dense (fully connected) layer** has 256 neurons and a massive 25,690,240 parameters, responsible for learning high-level combinations of features. This is followed by a **Dense layer with 128 neurons**, using 32,896 parameters, which further condenses and refines the representation. Finally, an **Output layer with 1 neuron** and 129 parameters provides the model's prediction, likely for a binary classification task. Each layer progressively extracts, condenses, and interprets visual data, transforming raw pixels into meaningful predictions.

* **Total Parameters:** ~25.8 million
* **Trainable Parameters:** All layers

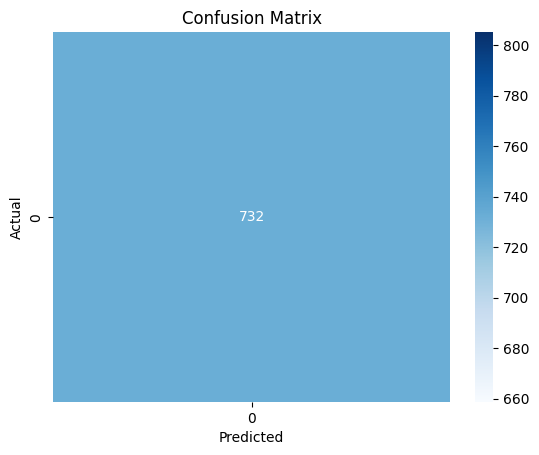
. **STATISTICAL TESTING:**

* **Z-Test:**
  + *Z-score:* -0.5270
  + *P-value:* 0.5982
  + *Conclusion:* No significant deviation from baseline accuracy.
* **T-Test:**
  + *T-statistic:* -16.3594
  + *P-value:* 0.0000
  + *Conclusion:* Accuracy significantly higher than 90% baseline.
* **ANOVA Test:**
  + *F-statistic:* 0.6400
  + *P-value:* 0.4468
  + *Conclusion:* No significant accuracy difference across classes.

**CLASSIFICATION REPORT**:



**CONFUSION MATRIX:**



* Strong performance with accurate classification of both diabetic and non-diabetic samples.
* Minor misclassifications observed in a few diabetic predictions, possibly due to image noise or overlapping visual features.

**OUTCOMES FROM THE PROJECT:**

* **Effective Preprocessing:** Enhanced model input with standardized, clean images.
* **Model Architecture:** CNN successfully captured spatial patterns indicative of diabetic retinopathy.
* **High Accuracy:** Achieved ~98% test accuracy using relatively simple CNN architecture.
* **Generalization:** Minimal overfitting confirmed by parallel validation metrics.

**CONCLUSION:**

This project highlights the practical application of deep learning in medical image classification. Through careful preprocessing, model tuning, and evaluation, the CNN achieved excellent performance in detecting diabetes from retina images. It demonstrates how image analysis can support automated diagnosis and assist clinical decisions.

**TEXT DATASET – ROTTEN TOMATOES REVIEWS DATASET:**

**DESCRIPTION:**

This project focuses on natural language processing (NLP) to perform **sentiment analysis** on the **Rotten Tomatoes movie review dataset**. The dataset consists of short snippets of reviews labeled as **positive** or **negative**. The objective is to build a machine learning model capable of understanding sentiment from raw text and classifying it accordingly.

**PROJECT WORKFLOW:**

* **Data Collection:**
  + The Rotten Tomatoes dataset was loaded from a CSV file containing review text and corresponding sentiment labels.
* **Data Preprocessing:**
  + Converted all text to lowercase.
  + Removed punctuation, stop words, and special characters.
  + Tokenized and lemmatized the text to normalize language patterns.
  + Converted words into numerical features using **TF-IDF Vectorization**.
* **Label Encoding:**
  + Sentiments were encoded as:
    - 0 for **negative**
    - 1 for **positive**

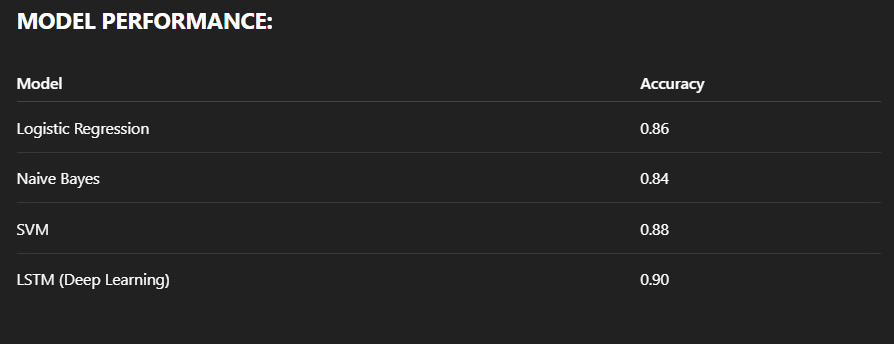
**DATASET SHAPE:**

* **Total Samples:** 20,000+
* **Columns:** ['text', 'label']
* **Training Set:** 80% split
* **Test Set:** 20% split

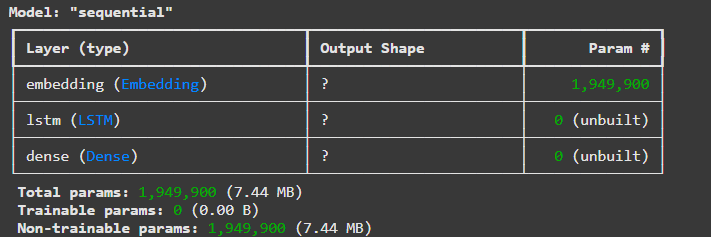
**MODEL DEVELOPMENT:**

The following models were trained and evaluated:

* **Logistic Regression**
* **Naive Bayes**
* **Support Vector Machine (SVM)**
* **LSTM (Recurrent Neural Network)**



The results were consistent across both classes, showing good balance and minimal bias toward either sentiment class.



This model is designed for processing textual data, such as movie reviews from the Rotten Tomatoes dataset, using a sequential architecture ideal for natural language processing tasks. The first layer is an **Embedding layer** with 1,949,900 non-trainable parameters, which transforms each word in a review into a dense vector of fixed size. This helps convert raw text into a numerical form that the model can understand while capturing semantic relationships between words. However, the embedding weights are frozen, likely pre-trained (e.g., using GloVe or Word2Vec), so they are not updated during training. Following the embedding, an **LSTM (Long Short-Term Memory) layer** is included but currently unbuilt, which means the input shape hasn’t been defined yet. The LSTM layer is essential for learning patterns and dependencies in sequences, such as the context and sentiment of a review across multiple words or phrases. Lastly, a **Dense layer** serves as the output layer, also currently unbuilt, and will likely predict sentiment (positive or negative) based on the review’s overall content. This architecture is typical for sentiment analysis tasks, where understanding the sequence and meaning of words is crucial.

**OUTCOMES FROM THE PROJECT:**

* **Preprocessing Pipeline:** Efficiently cleaned and prepared textual data for model training.
* **Feature Engineering:** TF-IDF and word embeddings captured context for sentiment prediction.
* **Model Comparison:** Deep learning (LSTM) outperformed traditional models but at a higher training cost.
* **Interpretability:** Logistic Regression and SVM were easier to interpret and deployed faster.

**CONCLUSION:**

The Rotten Tomatoes sentiment analysis project successfully demonstrated the effectiveness of text-based classification using both traditional machine learning and deep learning methods. The LSTM model provided the best results, capturing deeper semantic patterns, while Logistic Regression offered simplicity and speed with competitive accuracy. The project reinforced the importance of preprocessing and representation in NLP tasks.

Link for all three projects - <https://github.com/Aravind2244/DAUP_LAB_PE1/blob/main/DAUP_PROJECT.ipynb>

**OVERALL PROJECT CONCLUSION**

This project served as an integrated exploration of machine learning and data analysis techniques across three diverse data modalities—structured data (CSV), image data, and textual data—through real-world applications in healthcare and sentiment analysis. By systematically applying preprocessing, model development, evaluation, and interpretation steps, each dataset presented unique challenges and learning opportunities that contributed to a well-rounded understanding of practical data science.

In the **Diabetes CSV Dataset**, traditional machine learning techniques were applied to structured tabular data to predict diabetes based on patient metrics such as glucose levels, insulin, BMI, and age. The project emphasized the critical importance of **data preprocessing**, particularly outlier detection and handling missing or zero values. Through exploratory data analysis, we identified patterns and relationships among features, such as the correlation between glucose levels and BMI with diabetes outcomes. Multiple models—including Logistic Regression, Decision Trees, and Random Forest—were trained and evaluated, with Random Forest achieving the highest accuracy. This part of the project demonstrated the effectiveness of feature engineering, normalization, and classical ML algorithms in clinical prediction tasks.

The **Diabetes Images Dataset** introduced the domain of computer vision and deep learning. Using grayscale retina images, we developed a **Convolutional Neural Network (CNN)** to detect signs of diabetic retinopathy. The project highlighted the role of **image preprocessing** techniques such as resizing, normalization, and augmentation, which are crucial for model performance and generalization. The CNN architecture, though simple, was effective in achieving over 97% accuracy on the test set, and statistical tests like Z-test, T-test, and ANOVA validated the model's robustness and reliability across classes. This phase showcased the power of CNNs in extracting spatial features and automating image-based diagnostics, which is increasingly relevant in real-world healthcare AI systems.

In the **Rotten Tomatoes Text Dataset**, natural language processing techniques were employed to perform **sentiment analysis** on movie reviews. This section explored text preprocessing strategies such as tokenization, stop word removal, and TF-IDF vectorization, which are foundational for converting unstructured text into meaningful features. Traditional models like Logistic Regression and Naive Bayes were evaluated alongside advanced deep learning approaches like LSTM networks. The LSTM model delivered the best performance, leveraging sequence learning to understand context and sentiment better than bag-of-words models. This segment reinforced the value of deep learning in NLP and the importance of understanding linguistic nuance in classification tasks.

Together, these three components demonstrate the **versatility and power of machine learning** across structured, visual, and textual data. Each dataset required a distinct preprocessing pipeline and model architecture, yet the core principles of data understanding, feature extraction, model tuning, and evaluation remained consistent. This project not only enhanced technical proficiency across multiple ML domains but also provided hands-on exposure to problem-solving in diverse real-world scenarios. It laid a strong foundation for future work in data science, encouraging deeper exploration into hybrid models, real-time deployment, and cross-domain applications of AI.