E-COMMERCE PRODUCT REVIEW ANALYSIS

## Project Description:

E-Commerce Product Review Analysis is an advanced Machine Learning (Natural Language Processing) project designed to analyze customer feedback and classify sentiments. The system utilizes supervised learning algorithms to process textual reviews from platforms like Amazon and provide accurate sentiment predictions (Positive, Negative, or Neutral). By examining review text for keywords and context, the model assists sellers and customers in understanding product quality, highlighting strengths (e.g., "great battery life") and weaknesses (e.g., "shipping delay").

## Project Scenarios

### Scenario 1: Customer Purchase Decision

A customer wants to buy a smartphone but sees 5,000 reviews. Instead of reading all of them, they use the system to see that 85% of reviews are "Positive" and the main keyword associated with negative reviews is "Camera," helping them make a quick, informed decision.

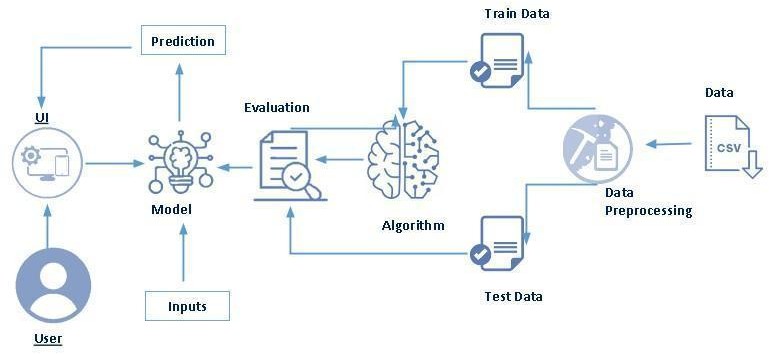
### Scenario 2: Seller Quality Control

A seller launches a new product. The system analyzes incoming reviews in real-time. If the sentiment suddenly drops, the system flags specific phrases like "broken seal" or "late delivery," allowing the seller to fix the supply chain immediately.

### Scenario 3: Automated Content Moderation

E-commerce platforms receive millions of reviews daily. The system automatically classifies reviews, filtering out spam or highlighting extremely negative reviews for customer support intervention.

## Technical Diagram:



**Prerequisites:**

To complete this project, you must require the following software, concepts, and packages

## Anaconda Navigator and Visual Studio:

* + Refer to the link below to download Anaconda Navigator
  + Link: <https://youtu.be/1ra4zH2G4o0>

## Python packages:

* + Open anaconda prompt as administrator
  + Type “pip install numpy” and click enter.
  + Type “pip install pandas” and click enter.
  + Type “pip install scikit-learn” and click enter.
  + Type “pip install matplotlib” and click enter.
  + Type “pip install seaborn” and click enter.
  + Type “pip install nltk” and click enter.
  + Type “pip install wordcloud” and click enter.
  + Type “pip install Flask” and click enter.

## Prior Knowledge:

You must have prior knowledge of the following topics to complete this project.

### ML Concepts

* + Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
  + Natural Language Processing(NLP): [https://www.javatpoint.com/nlp](https://www.google.com/search?q=https://www.javatpoint.com/nlp)
  + Support Vector Machine(SVM): [https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm](https://www.google.com/search?q=https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm)
  + Logistic Regression:

https://[www.javatpoint.com/logistic-regression-in-machine-learning](http://www.javatpoint.com/logistic-regression-in-machine-learning)

* + TF-IDF(Feature Extraction): <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>
  + Evaluation metrics:

[https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/) [metrics/](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/)

**Flask Basics**: <https://www.youtube.com/watch?v=lj4I_CvBnt0>

## Project Flow:

* User enters product review text or uploads a dataset through the interface
* System preprocesses the input text using cleaning techniques and the trained TF-IDF vectorizer
* Integrated machine learning model analyzes the vectorized text features
* Prediction result (Positive/Negative/Neutral) is displayed with a confidence score
* System highlights key product insights or quality issues based on the sentiment analysis

## Project Activities:

### Data Collection & Preparation

* + Collect the e-commerce review dataset
  + Data Preparation and Cleaning

### Exploratory Data Analysis

* + Descriptive statistics
  + Visual Analysis
  + Univariate and Bivariate Analysis

### Model Building

* + Training the model with multiple algorithms
  + Testing different classifiers

### Performance Testing & Model Selection

* + Testing model with multiple evaluation metrics
  + Comparing model accuracy across different algorithms
  + Selecting the best performing model

### Model Deployment

* + Save the best model
  + Integrate with Web Framework

## Project Structure:

Create the Project folder which contains files as shown below



### Project Structure Explanation

* static/

Contains static assets (CSS, JavaScript, uploaded images)

* + style.css – Application styling
* templates/

HTML templates for Flask application

index.html – Landing page

* app.py

Flask application backend

* Tfidf-vectorizer.pkl

Trained model file

* gen ai.ipynb
* Jupyter notebook with model training code
* Dataset

Amazon\_Reviews.csv

# Milestone 1: Data Collection & Preparation

Data collection is the critical first step in building a Sentiment Analysis system, serving as the knowledge base for Natural Language Processing (NLP) algorithms. This process involves gathering unstructured textual data from e-commerce platforms, specifically focusing on customer reviews, ratings, and product feedback. The richness, linguistic variety, and balance of the collected text data directly influence the model's ability to accurately detect sentiment and generate actionable product insights

## Activity 1.1: Collect the dataset

There are many popular open sources for collecting e-commerce feedback data. Eg: kaggle.com, Amazon Public Datasets, UCI repository, etc. In this project, we have utilized .csv data containing customer reviews and ratings. This data is sourced from kaggle.com to simulate real-world e-commerce scenarios. Please refer to the link given below for the source of the dataset.

**Link:** <https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews>

As the dataset is downloaded, let us read and understand the text data properly with the help of Natural Language Processing (NLP) preprocessing and visualization techniques like Word Clouds.

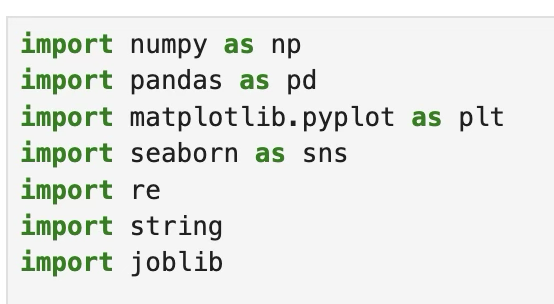
**Note:** There are several techniques for understanding unstructured text data. But here we have used specific NLP methods suitable for sentiment extraction. In an additional way, you can use multiple feature extraction techniques.

The dataset used in this project contains customer reviews and ratings for various products sold on e-commerce platforms like Amazon. The dataset includes the following features:

* **Review Text:** The actual textual feedback and comments written by the customer regarding the product.
* **Rating:** The numerical score given by the customer (Scale: 1 to 5 stars).
* **Sentiment:** The target variable derived from the rating (Positive, Negative, or Neutral).
* **Review Summary:** A short summary or title of the review provided by the user.
* **Time:** The timestamp indicating when the review was posted.

## Activity 1.2: Importing the Libraries

Import the necessary libraries as shown below:

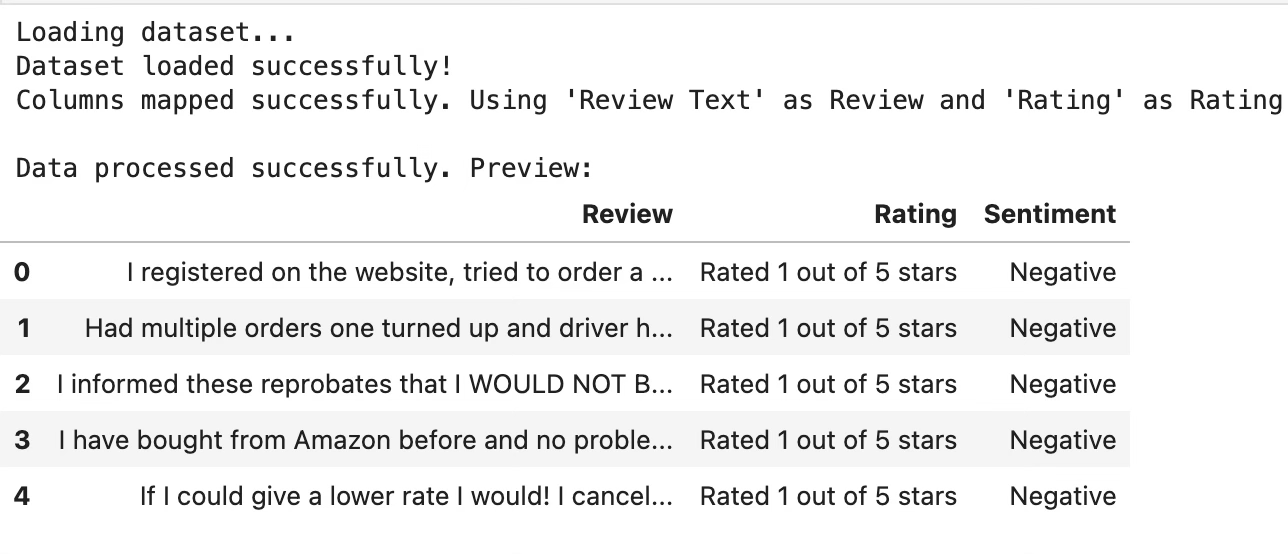


These libraries serve the following purposes:

* **NumPy:** For numerical computations and array operations.
* **Pandas:** For data manipulation, loading the CSV dataset, and cleaning data.
* **Matplotlib:** For creating static visualizations like bar charts.
* **Seaborn:** For statistical data visualization (e.g., heatmaps, count plots).
* **NLTK:** For Natural Language Processing tasks like removing stopwords and stemming.
* **WordCloud:** For visualizing the most frequent words in the reviews.
* **Scikit-learn:** For machine learning tasks including TF-IDF vectorization, model training (Logistic Regression), and evaluation metrics.

## Activity 1.3: Read the Dataset

Our dataset is in CSV format. We read the dataset using pandas:



## Activity 1.4: Data Preparation

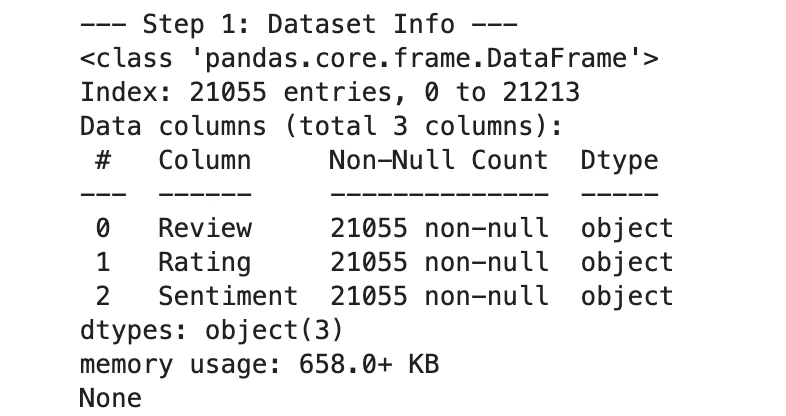
Before we can use our data to teach our machine-learning model, we need to clean it up. This includes:

* Handling missing values
* Removing duplicates
* Checking data types
* Handling outliers

Note: These are general steps for pre-processing data. Depending on the condition of your dataset, you may or may not have to go through all these steps.

## Activity 1.5: Handling Missing Values

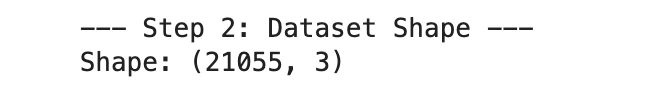
Step 1: Check the datatypes and null count of each column:



This gives us information about:

* Total number of entries
* Column names and their data types
* Number of non-null values in each column
* Memory usage

### Step 2: Check the size of the dataset:



This returns the number of rows and columns in the dataset.

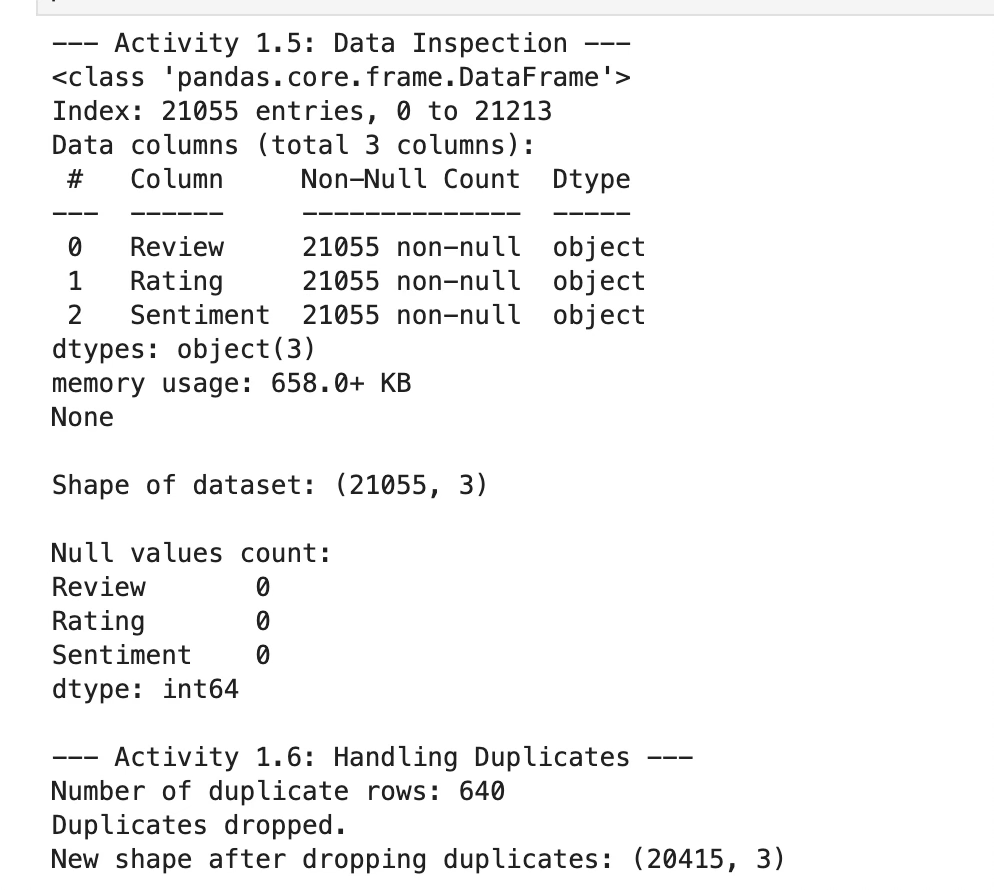
### Step 3: Check for null values:

### 

This returns the count of null values in each column. Fortunately, our dataset has no missing values.

## Activity 1.6: Handling Duplicates

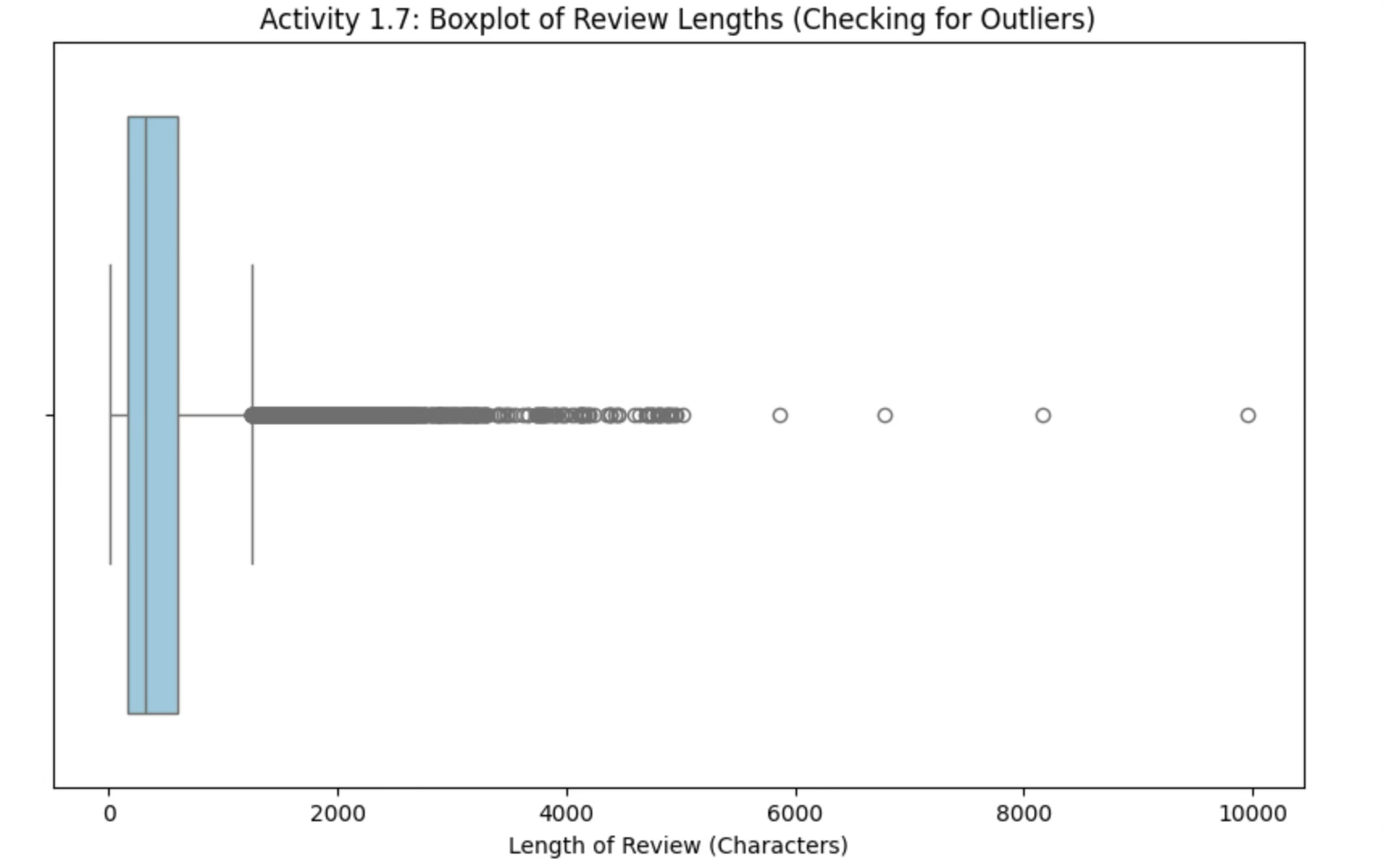
Check for duplicate records in the dataset:



After removing duplicates, we verify the new shape of the dataset.

## Activity 1.7: Checking for Outliers

Create boxplots to visualize outliers in numerical features:



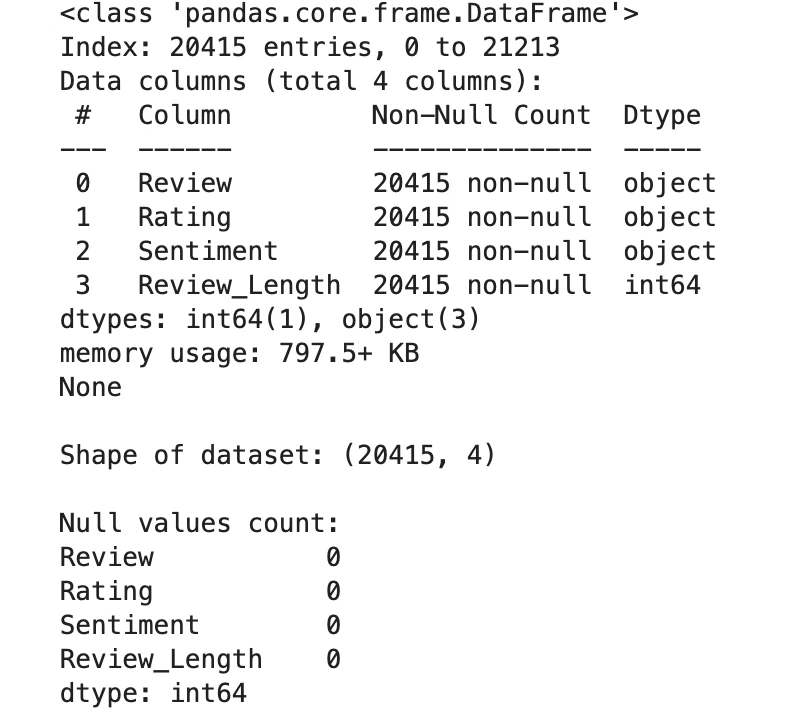
Boxplots help identify outliers, which are data points that fall outside the typical range. These can affect model performance if not handled properly.

# Milestone 2: Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in understanding the characteristics, patterns, and relationships within the dataset. It helps in making informed decisions about feature engineering and model selection.

## Activity 2.1: Descriptive Statistics

Descriptive analysis studies the basic features of data using statistical processes. Pandas provides the describe() function to get statistical summaries:



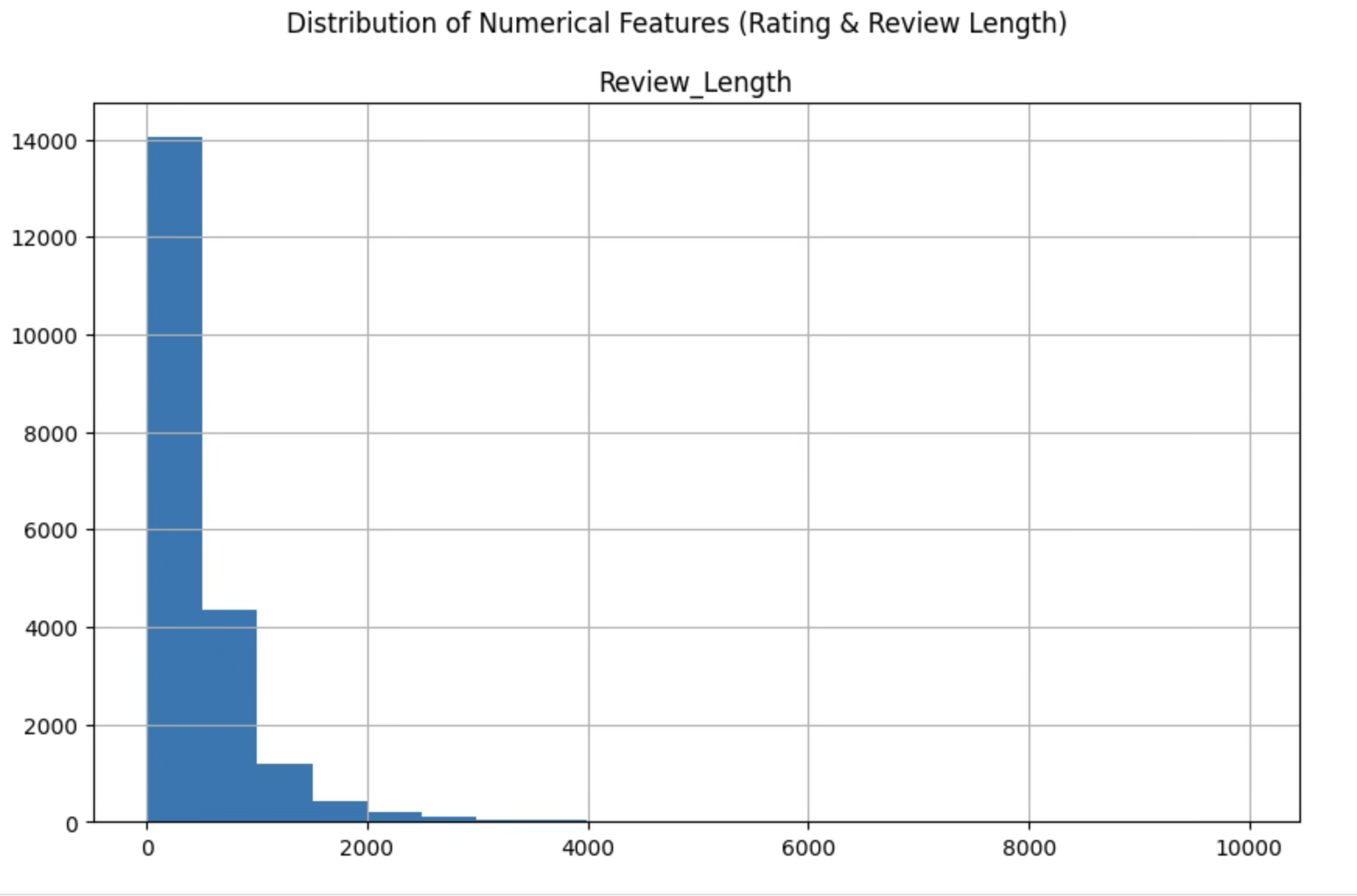
This provides:

* **Mean, Standard Deviation, Min, Max** for numerical features
* **Count** of categorical features
* **Distribution** of target variable

## Activity 2.2: Visual Analysis

Visual analysis uses charts, plots, and graphs to explore and understand data patterns quickly.

### Distribution of Numerical Features:



Histograms show the distribution of each numerical feature, helping identify:

* Skewness in the data
* Most common value ranges
* Potential outliers

## Activity 2.3: Univariate Analysis

Univariate analysis examines each feature individually.

### Sentiment Distribution:

### 

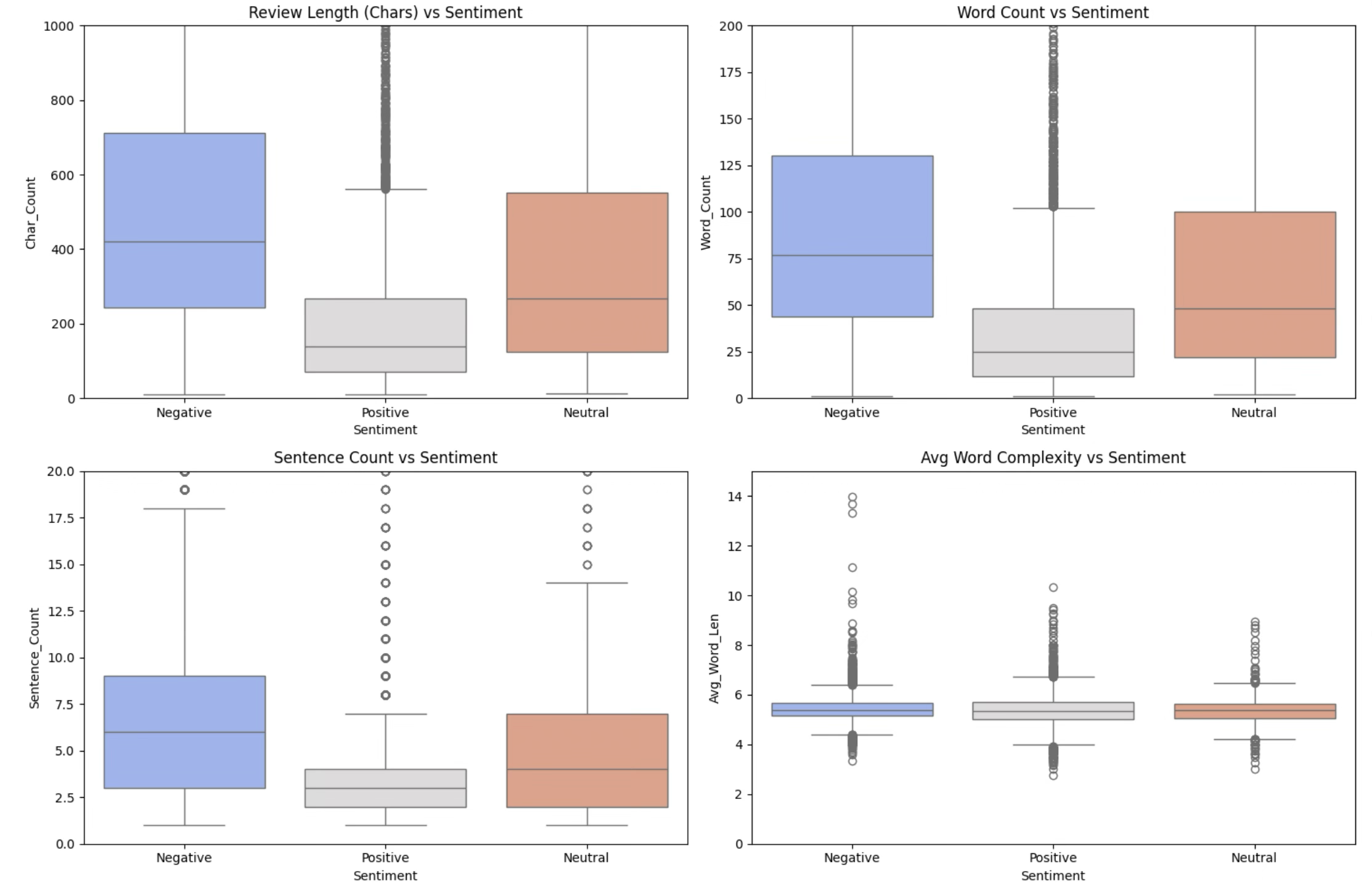
This shows the count of sentiment vs count in the dataset.

## Activity 2.4: Bivariate Analysis

Bivariate analysis examines the relationship between two features.

### Boxplots by Result:

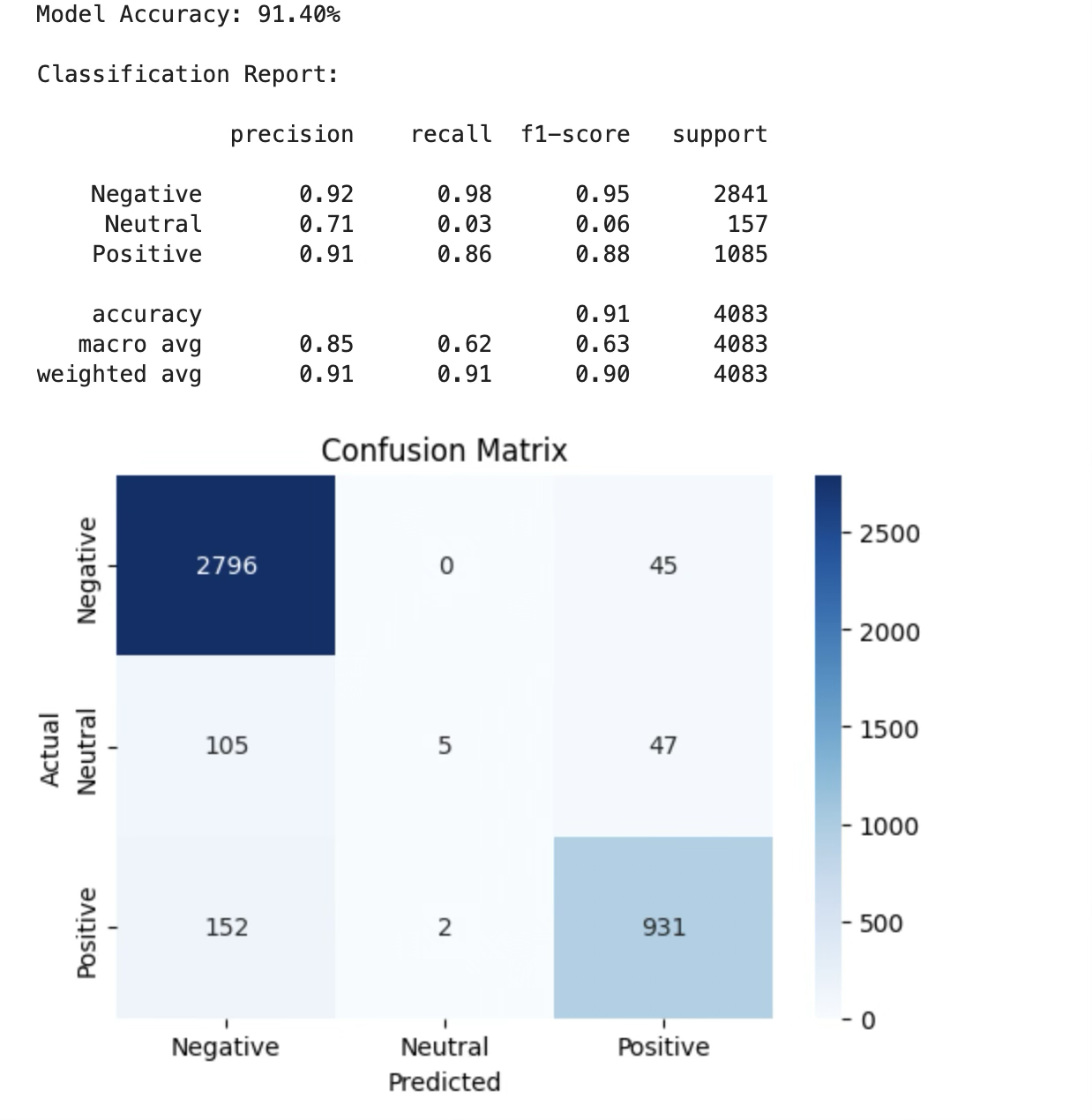
python



These boxplots show how each blood parameter differs between anemic and non-anemic patients

## Activity 2.5: Correlation Analysis

### Correlation Heatmap:



The correlation heatmap visualizes the relationships between numerical features derived from the text (Review Length, Word Count, Sentence Count) and the target variable (Rating). Strong positive correlations appear in red/warm colors, while negative correlations appear in blue/cool colors. This helps identify structural dependencies, such as the strong link between word count and sentence count, and analyze how these features relate to the customer's sentiment.

## Activity 2.6: Pairplot Analysis

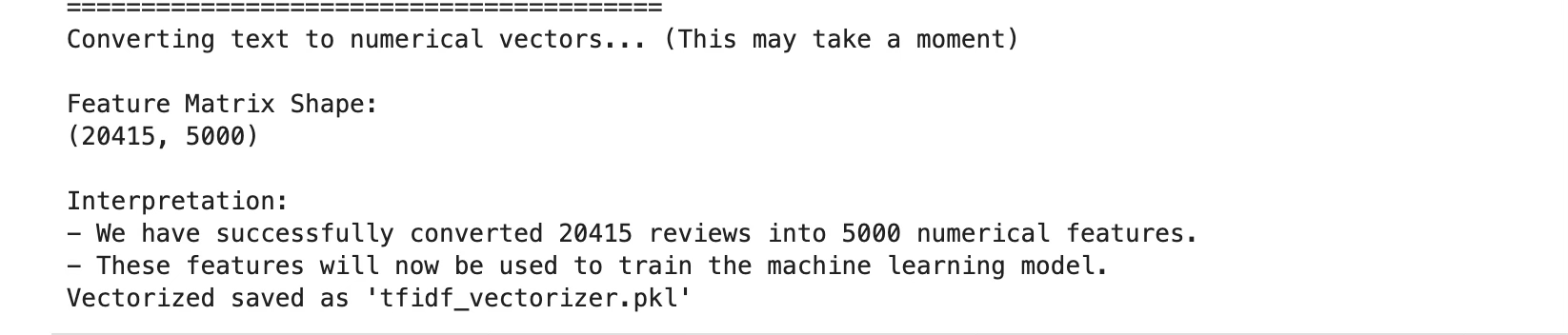
## 

The pairplot creates a grid of scatter plots showing relationships between all pairs of features, color-coded by the Result (anemic/not anemic). This comprehensive visualization helps identify patterns and separability between classes.

## Activity 2.7: Feature Scaling

Before training machine learning models, we need to transform our raw text data into numerical vectors to ensure the algorithms can process and learn from the language patterns.

### TF-IDF VECTORIZATION

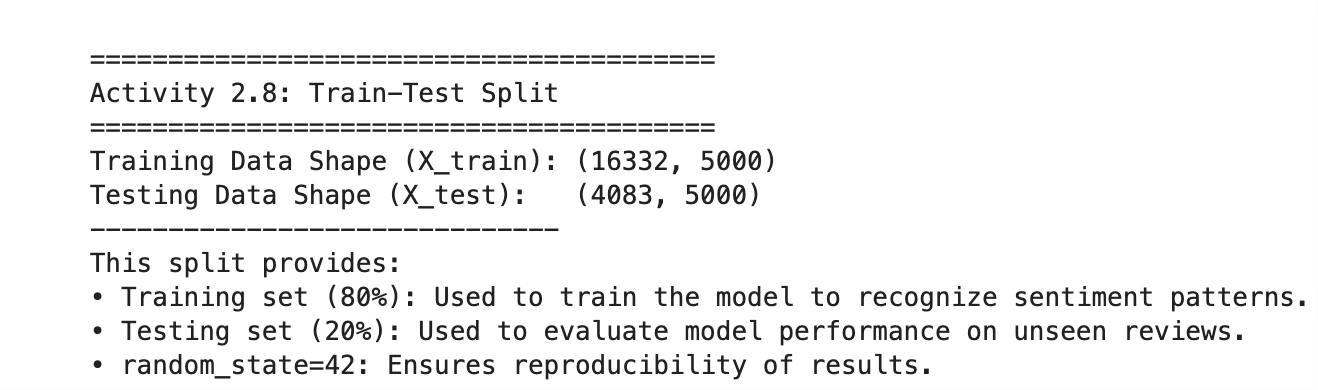


**TF-IDF Vectorizer transforms text by:**

* Counting word occurrences in each document (Term Frequency)
* Reducing the weight of common words across the dataset (Inverse Document Frequency)

## Activity 2.8: Train-Test Split

Split the dataset into training and testing sets:

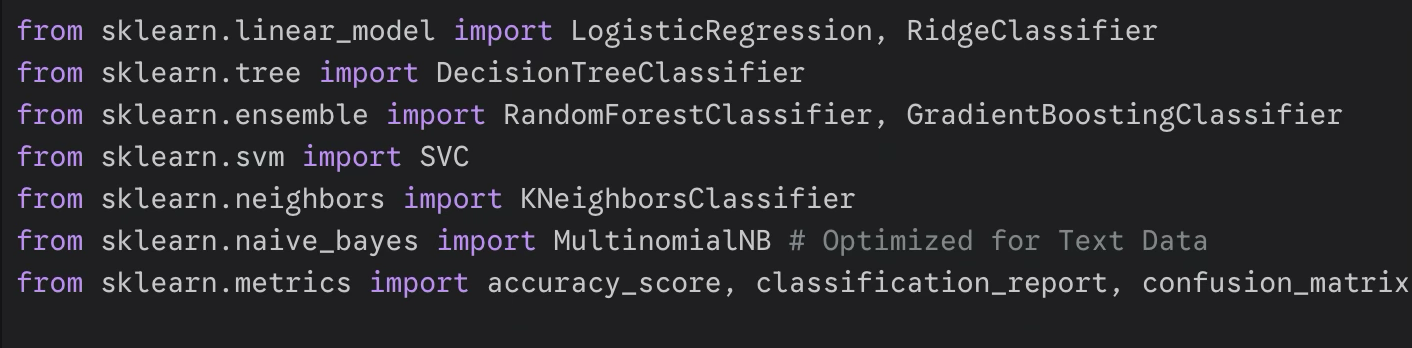


* Training set (80%): Used to train the model
* Testing set (20%): Used to evaluate model performance
* random\_state=42: Ensures reproducibility

# Milestone 3: Model Building

Now we train multiple machine learning algorithms and compare their performance.

## Activity 1: Import Required Libraries



**Activity 2: Logistic Regression Model**



Logistic Regression is a linear model for binary classification that predicts the probability of a sample belonging to a particular class.

## Activity 4: Supervized Learning

## 

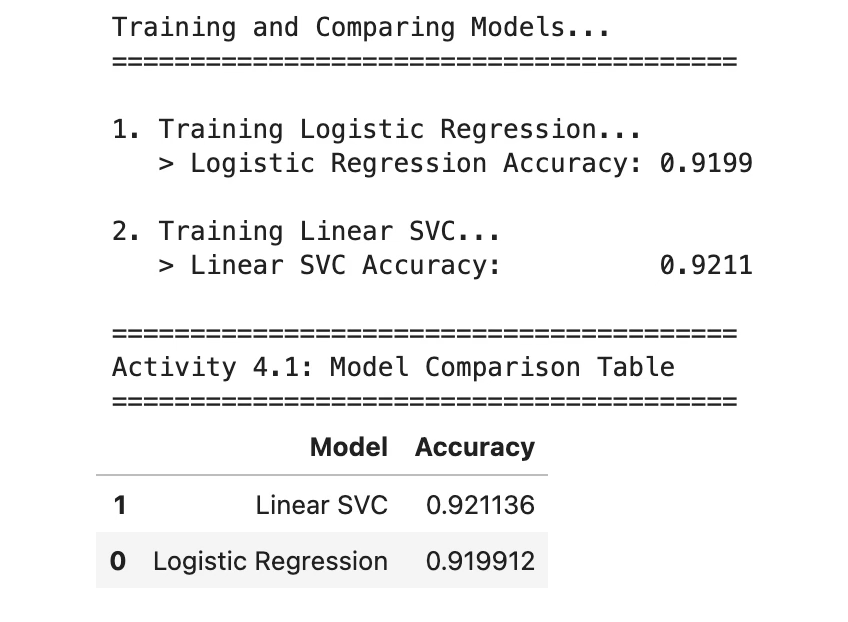
## Activity 5: EVALUATION METRICS

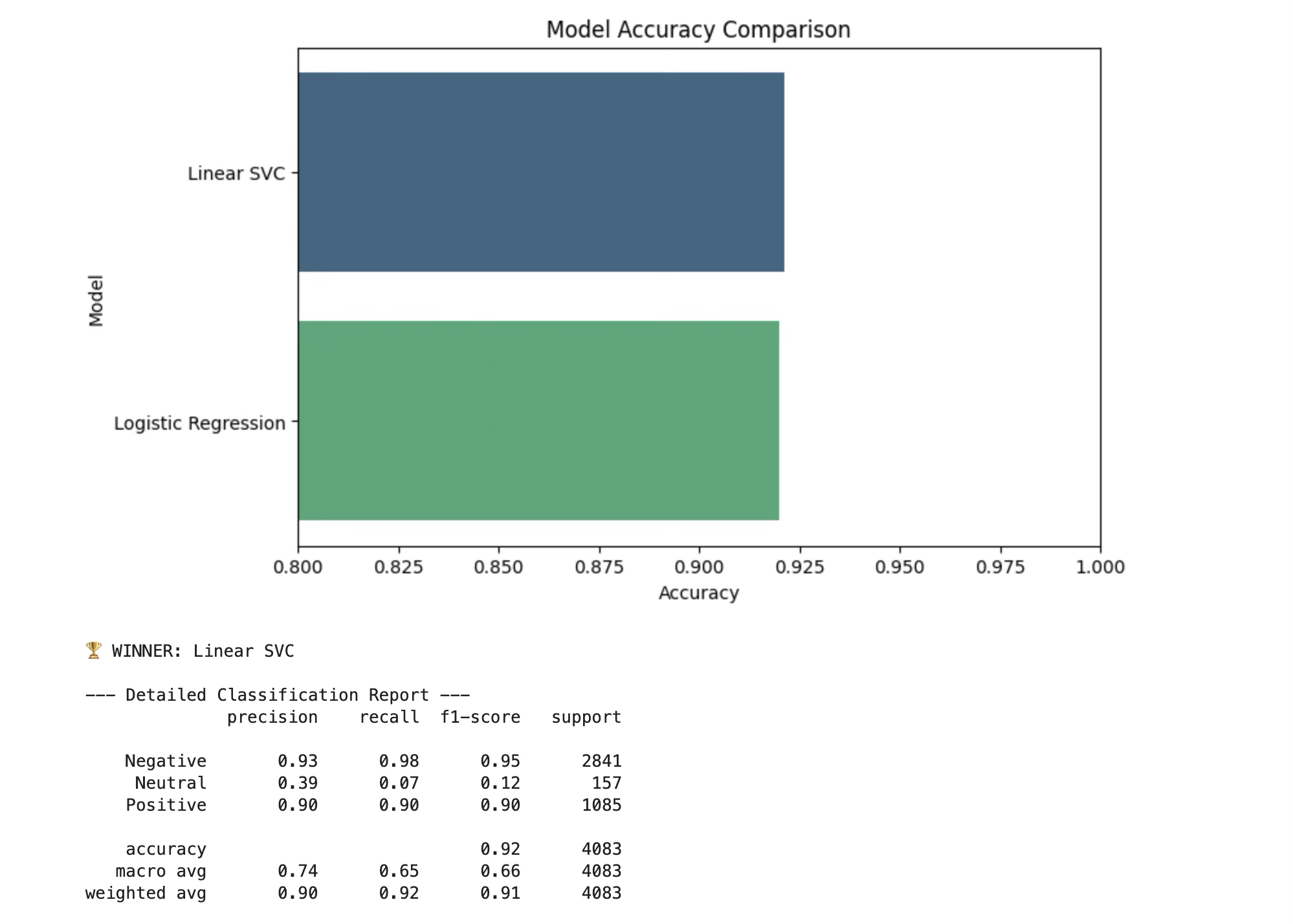
## 

# Milestone 4: Performance Testing & Model Comparison

## Activity 4.1: Model Comparison

Compare all models using their accuracy scores:





This creates a comparison table showing the accuracy of the two supervised learning algorithms used in this project: **Logistic Regression** and **Linear SVC**.

Based on the comparison, **Linear SVC** achieves the highest accuracy and is selected as our best model for the Anemia Detection System.

## Activity 4.2: Evaluation Metrics

Understanding the evaluation metrics:

* **Accuracy:** Overall percentage of correct predictions
* **Precision:** Of all predicted positives, how many are actually positive
* **Recall:** Of all actual positives, how many were correctly identified
* **F1-Score:** Harmonic mean of precision and recall
* **Confusion Matrix:** Shows true positives, true negatives, false positives, and false negatives

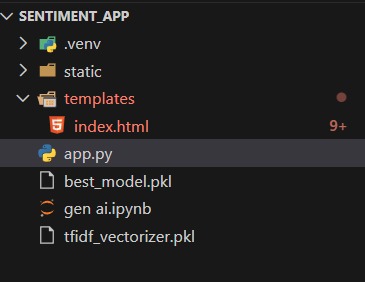
# Milestone 5: Model Deployment

## Activity 5.1: Save the Best Model

Save the trained model and scaler for future use:



## Activity 5.2: Flask Web Application Development



### Application Architecture

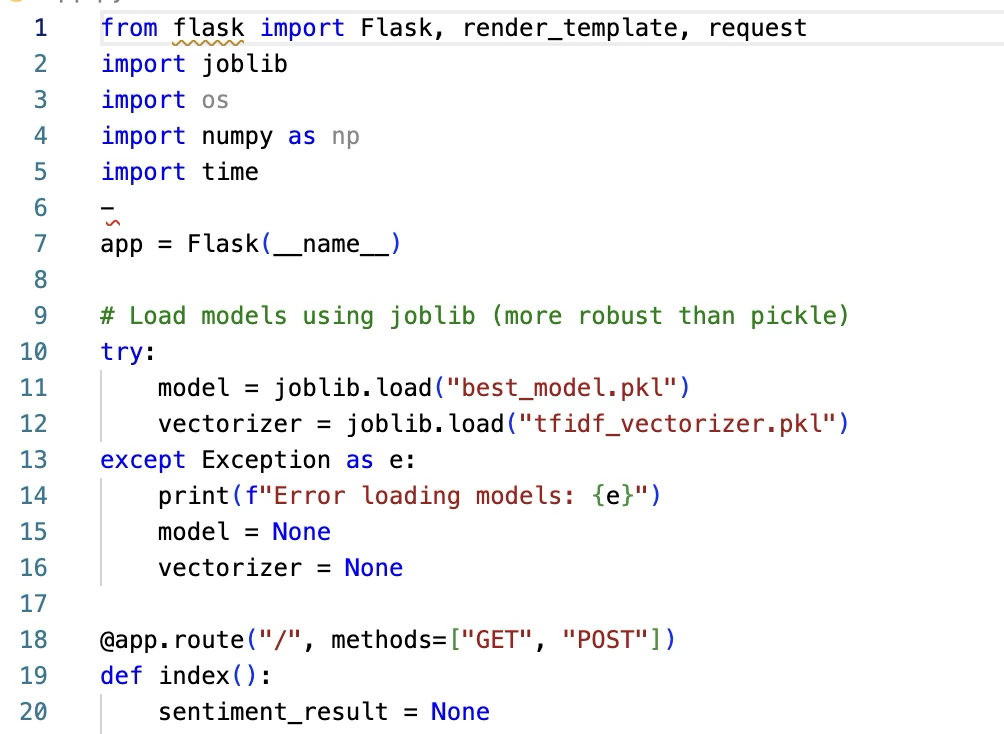
The application follows a standard Client-Server architecture for machine learning deployment:

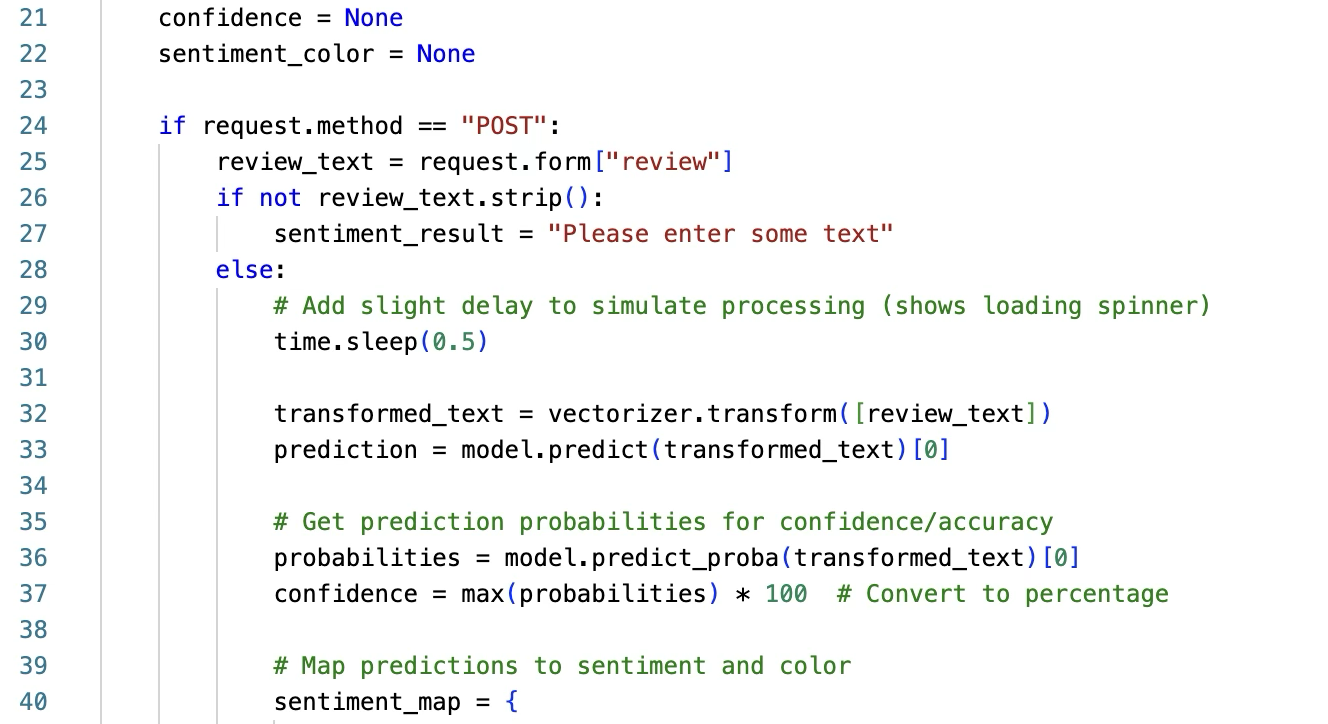
* Client (Frontend): index.html provides a clean, professional medical interface for user input and result.html displays the prediction.
* Backend (Server): app.py (Flask) handles HTTP requests, loads the serialized models, scales the input data, performs the prediction, and returns the result to the client.
* Persistence Layer: LogisticRegression\_model.pkl and scaler.pkl store the trained model parameters and scaling statistics.

### Backend Implementation (app.py)

The Flask backend is responsible for the core ML workflow:

* Model Loading: Loads LogisticRegression\_model.pkl and scaler.pkl on application startup to minimize prediction latency.
* Data Handling: Parses form data (Gender, Hemoglobin, MCH, MCHC, MCV) received via a POST request.
* Preprocessing: Applies the saved MinMaxScaler to the numerical features, ensuring the new data is treated identically to the training data.
* Prediction: Executes the logreg.predict() and logreg.predict\_proba() methods on the scaled input.
* Result Interpretation: Maps the binary output (0 or 1) to "Negative for Anemia" or "Positive for Anemia," and calculates the confidence percentage.







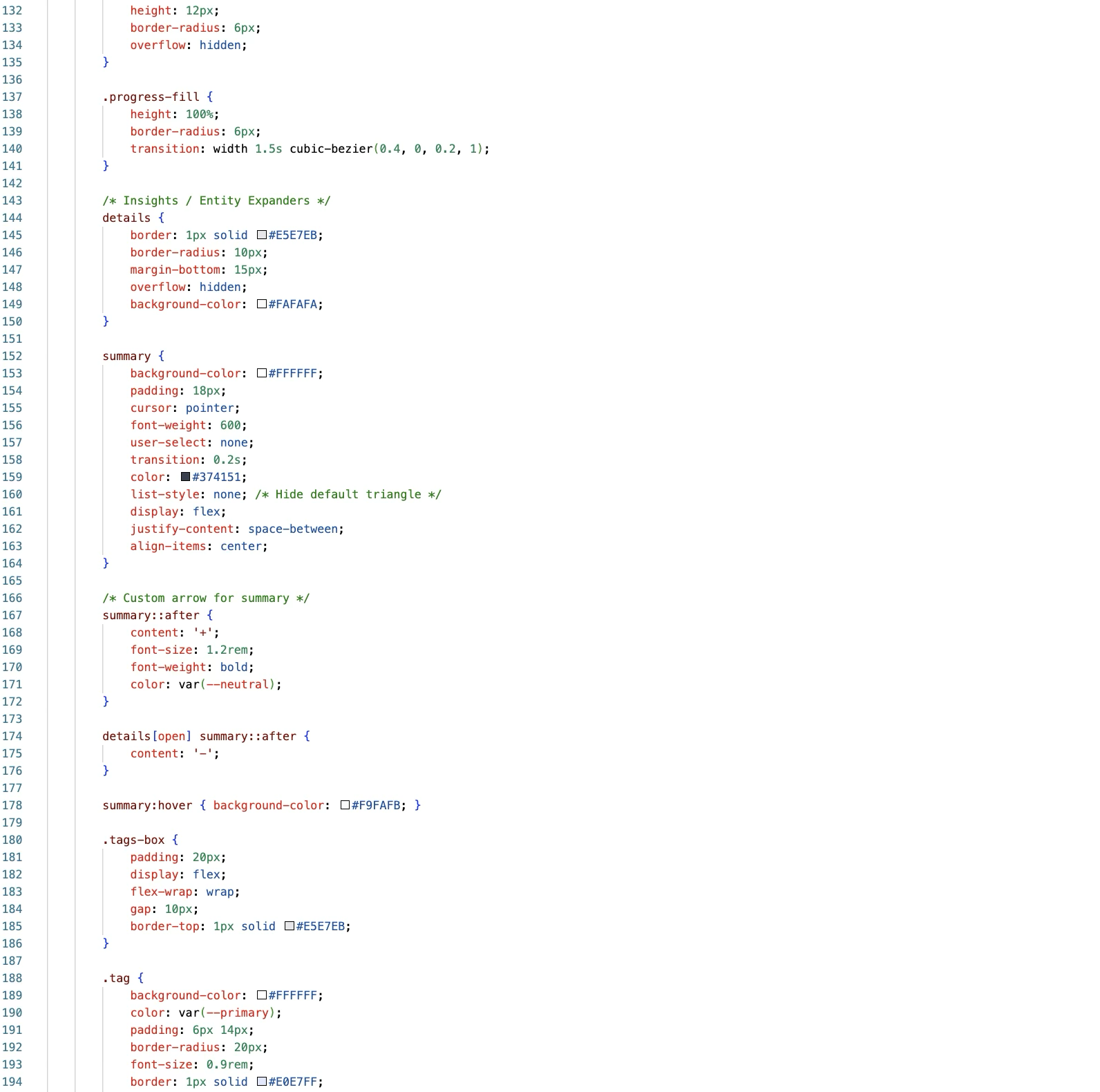
### Frontend Implementation (HTML Templates)

The web interface is designed with a Professional Medical Interface approach, optimizing for clarity and clinical use.

* Key Features of the Web Application:
  1. User Interface Components:
     + Professional medical header for a trustworthy presentation.
     + Comprehensive health assessment form for required parameters
     + Mobile-responsive design using CSS for universal access.
  2. Clinical Decision Support:
     + AI-powered risk assessment: Provides a clear diagnosis
     + Probabilistic confidence intervals: Displays the prediction probability
  3. Safety Features:
     + Input validation to enforce correct data types (float for lab values) and prevent errors.
     + Secure data handling via the Flask back-end (though this is a local demo).





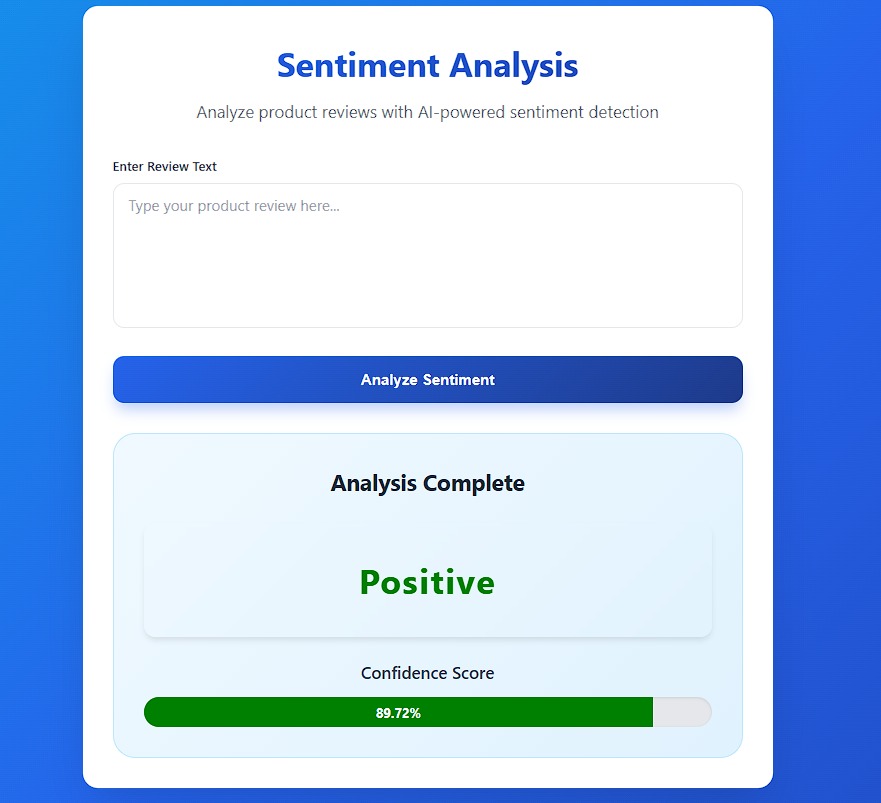


## Application Screenshots and Workflow

The workflow moves sequentially from user feedback input to sentiment classification and insight generation.

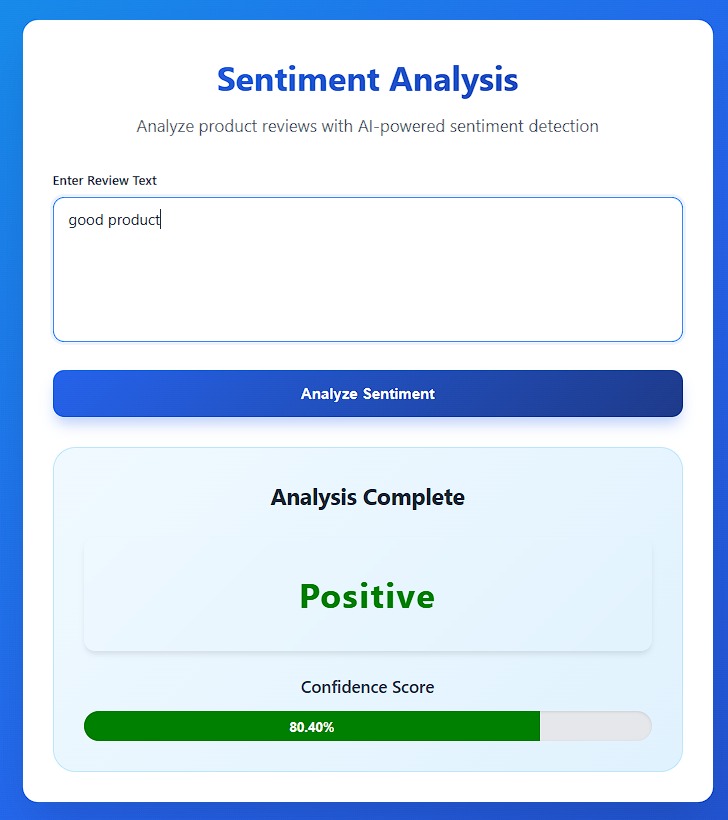
### Home Page Interface (index.html)

* + **Features:** A clean, intuitive interface focused on text analysis and user interaction.



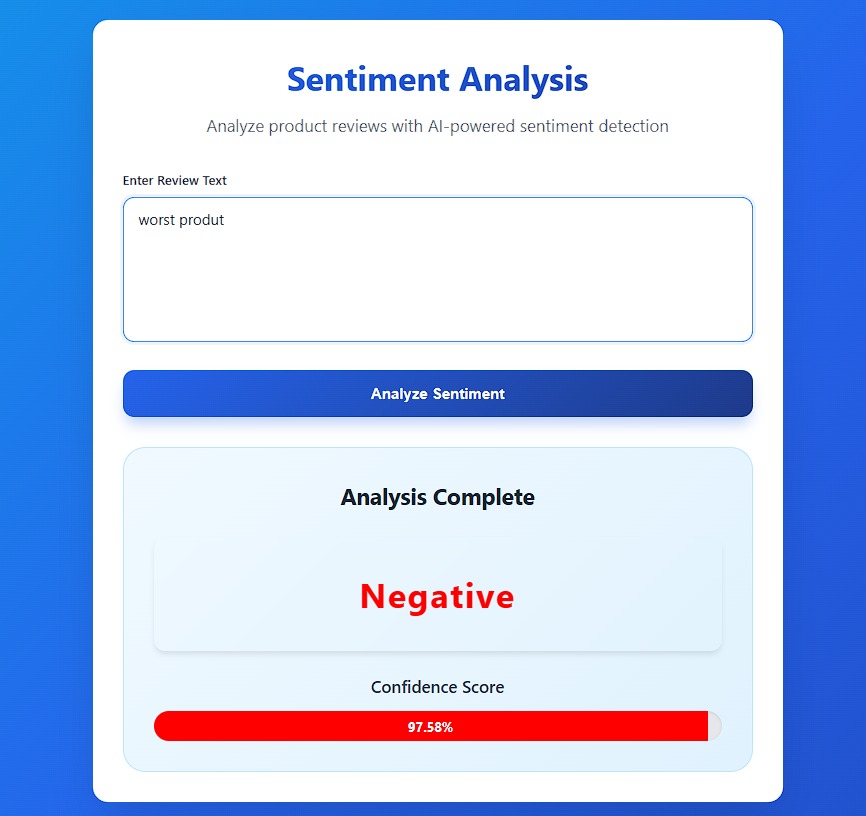
### **Review Analysis & Prediction Output**

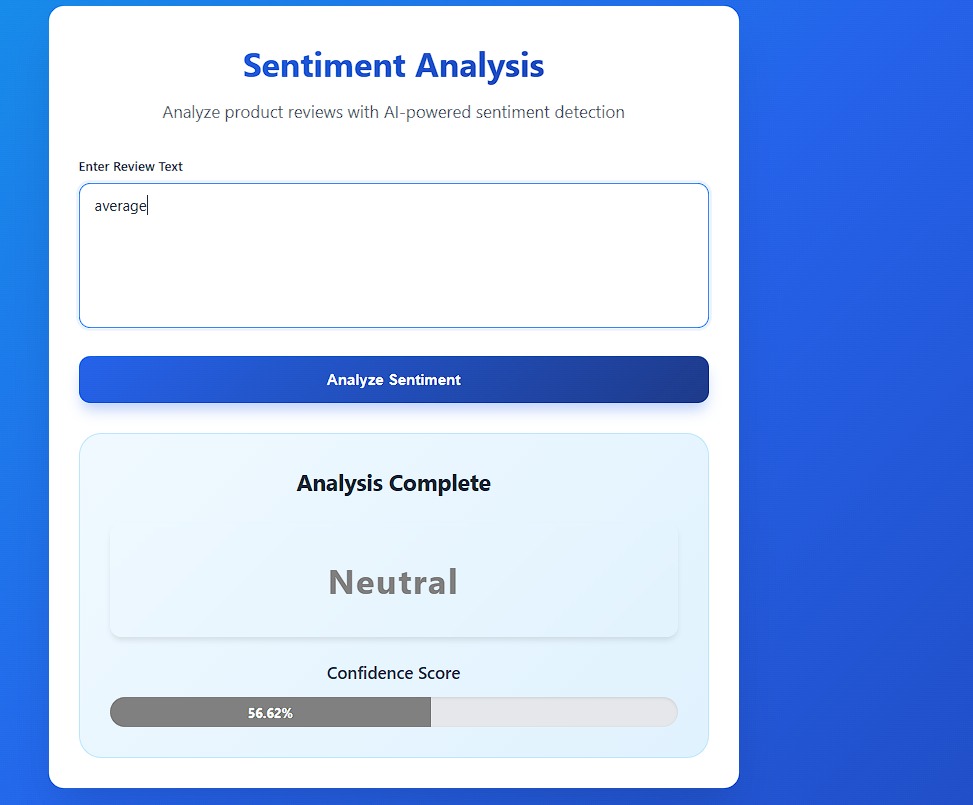
* **Product Identification:** A dedicated input field captures the specific product name, allowing the system to categorize the feedback and associate sentiments with specific items.
* **Customer Feedback Entry:** The main text area accepts raw user reviews, serving as the core input for the Natural Language Processing (NLP) algorithm.
* **Sentiment Classification:** Upon processing, the model evaluates the text and displays the resulting sentiment—**Positive**, **Negative**, or **Neutral**—directly on the screen.
* **Result Visualization:** The output is presented clearly to the user, providing immediate, actionable insight into the customer's opinion based on the entered text.



### **Sentiment Analysis Results**

* **Sentiment Classification:** The model categorizes the input text into distinct emotional tones (e.g., "Positive Sentiment," "Neutral," or "Negative Sentiment").
* **Confidence Score:** Displays a probability percentage (e.g., "Confidence Score: 94.5%") to indicate the reliability and certainty of the model's prediction.
* **Visual Indicators:** The interface uses dynamic color-coding (e.g., Green for Positive, Red for Negative) to provide immediate visual feedback on customer satisfaction.
* **Actionable Insights:** Provides a basis for decision-making, such as automatically flagging negative reviews for immediate support intervention or highlighting positive ones for marketing.





Future Implementations:

Future plans for the Product Review Analysis system focus on scalability, advanced linguistic processing, and business integration:

* **E-commerce Platform Integration:** Develop a robust API to integrate the model directly into major platforms (e.g., Shopify, WooCommerce, Amazon), allowing for real-time sentiment filtering and automated flagging of negative reviews.
* **Advanced NLP Models:** Explore deploying deep learning architectures (e.g., LSTM or BERT) to handle complex linguistic nuances, such as sarcasm, irony, and slang, for higher accuracy than traditional algorithms.
* **Analytics Dashboard:** Develop a comprehensive visual dashboard for business owners to track sentiment trends over time, identifying specific product batches or timeframes where customer satisfaction drops.
* **Multi-Lingual Support:** Expand the training dataset to include reviews in multiple languages and regional dialects, ensuring the model’s utility across diverse global markets and non-English customer bases.

## **Conclusion:**

The Ecommerce Product Review Analysis Project successfully developed an efficient and automated AI-powered sentiment analysis system. By utilizing Natural Language Processing (NLP) techniques and machine learning algorithms, the model achieved reliable classification of customer feedback into actionable insights. The final product is a functional web application that seamlessly integrates the trained model into a user-friendly interface. This solution offers a scalable and instant tool for gauging customer satisfaction, proving the project's success in transforming raw textual data into strategic business intelligence.