**A Holistic Approach to Food Waste Management: Leveraging Full-Stack Technologies for**

**Efficient Resource Utilization**

**Abstract**

This project introduces an advanced food donation platform designed to tackle the pressing issue of food waste while ensuring safe redistribution of edible food items. By integrating machine learning (ML) with a user-centric application, the system predicts the safety and usability of donated food based on critical factors such as food type, quantity, time of preparation, and associated storage conditions. At the core of the platform is a Random Forest Classifier, trained on a custom-built dataset comprising commonly wasted daily food items like rice, dosa, idly, and chapati. The model demonstrates high reliability with an accuracy of 98%, making data-driven decisions to assess food safety and usability. The platform operates in two major phases: Frontend user interaction, where donors log in to input food donation details using interactive forms, upload images of the food, and monitor the status of their donations. Recipients, on the other hand, can browse the available food items through an analytics-rich dashboard with real-time updates. Backend processing, powered by Python frameworks such as Flask/Django, where the stored donation data is passed to the ML model for predictions. The backend handles the flow of information between the user interface, database, and ML model, ensuring smooth and efficient communication.

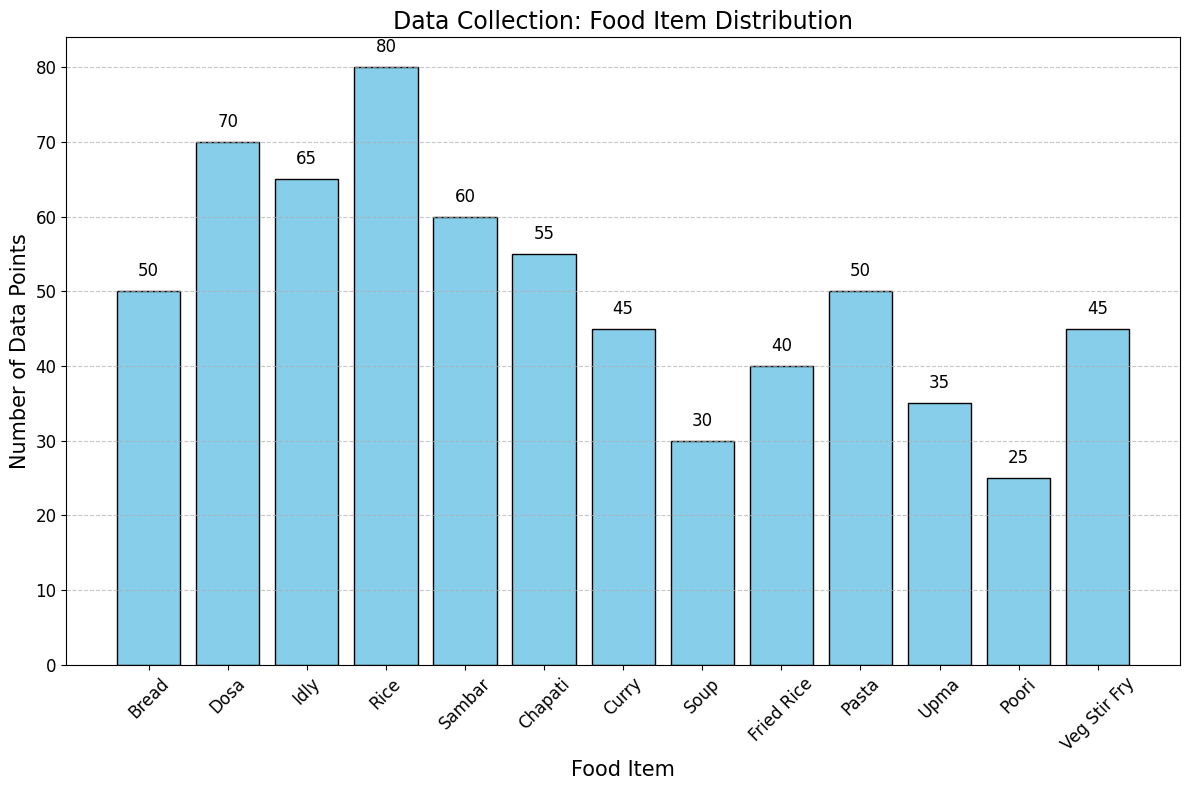
To enhance decision-making, the platform includes data visualizations, such as confusion matrices, feature importance charts, and learning curves, which provide insights into the performance of the model. An analytics dashboard tracks trends in food donations, providing users and administrators with a clear overview of the platform’s operations. The system is further optimized with a real-time prediction capability, allowing quick and reliable evaluation of food safety, thus minimizing risks for recipients.

This innovative approach not only encourages responsible sharing but also fosters sustainable consumption by enabling individuals to actively participate in reducing food waste. By focusing on safety, efficiency, and transparency, this project presents a scalable solution for addressing global food waste challenges, ensuring that surplus food reaches those in need while maintaining high standards of usability and safety.

**Methodology**

**1. Data Collection**

* Dataset Generation: A custom dataset was created with features such as:
  + Food Name (e.g., Rice, Bread).
  + Quantity (kg).
  + Cooking Time and Posting Time (hours).
  + Elapsed Time (hours) since cooking.
  + Shelf Life (hours) for each food type.
  + Safe for Donation (1 for safe, 0 for unsafe).
* Size: 500 samples with realistic thresholds and variance.



|  |  |
| --- | --- |
| **Food Item** | **Data Points** |
| Bread | 50 |
| Dosa | 70 |
| Idly | 65 |
| Rice | 80 |
| Sambar | 60 |
| Chapati | 55 |
| Curry | 45 |
| Soup | 30 |
| Fried Rice | 40 |
| Pasta | 50 |
| Upma | 35 |
| Poori | 25 |
| Veg Stir Fry | 45 |

**2. Data Preprocessing**

* Handling Missing Values: Ensured no missing data in our dataset.

| **Step** | **Description** | **Example** |
| --- | --- | --- |
| Data Cleaning | Remove or handle missing and inconsistent data entries. | Handle missing Shelf Life values. |
| Normalization | Scale numerical features to a range of 0-1 to ensure uniformity. | Quantity (kg) scaled from 0.5-10 → 0-1. |
| One-Hot Encoding | Convert categorical data into numerical format for model compatibility. | Food Type → Binary columns (e.g., Bread = 1, others = 0). |
| Feature Engineering | Create new features or refine existing ones for better predictions. | Add Elapsed Normalized for normalized cooking-posting time. |
| Train-Test Split | Split data into training and testing sets for evaluation. | 80% Training, 20% Testing. |
| Data Balancing | Handle imbalances in the target classes to avoid biased predictions. | Ensure Safe and Unsafe classes are balanced. |
| Outlier Detection | Identify and handle extreme or anomalous values in the dataset. | Remove entries with Quantity > 15 kg. |

* Encoding: Converted categorical features (e.g., Food Name) into numerical representations using One-Hot Encoding.
* Feature Scaling: Scaled numerical features like quantity and time using Min-Max Scaling for consistency.
* Train-Test Split: Split the dataset into 80% training and 20% testing subsets.

****

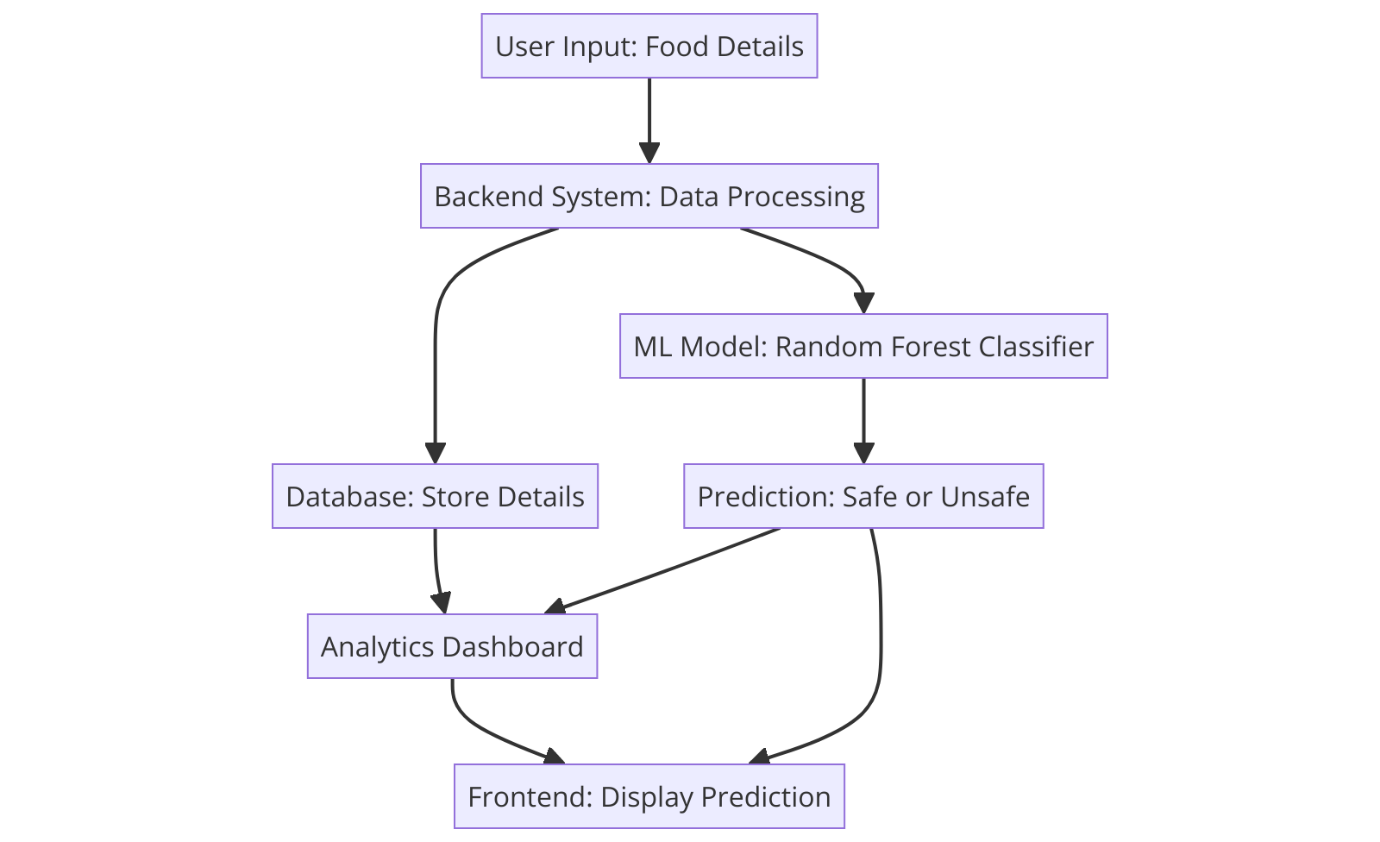
|  |  |  |  |
| --- | --- | --- | --- |
| **Feature Name** | **Before Preprocessing** | **After Preprocessing** | **Description** |
| Quantity (kg) | 0.5-10 | 0-1 | Range of food quantities |
| Elapsed Time (hours) | 0-12 | 0-1 | Time elapsed since cooking (hours) |
| Shelf Life (hours) | 04-Oct | 0-1 | Shelf life per food type (hours) |
| Cook Time | Jun-22 | 0-1 | Cooking time of the food (hours) |
| Post Time | Jun-23 | 0-1 | Posting time (hours) |
| Food Type | Categorical | One-Hot Encoded | Food category (e.g., Bread, Dosa) |
| Safe for Donation | Binary | Same | Safe or unsafe for donation |
| Elapsed Normalized | N/A | 0-1 | Normalized elapsed time |

**3. Model Selection**

We chose the **Random Forest Classifier** due to:

* Its ability to handle categorical and numerical data.
* Robustness against overfitting when tuned appropriately.
* Its interpretability for identifying feature importance.

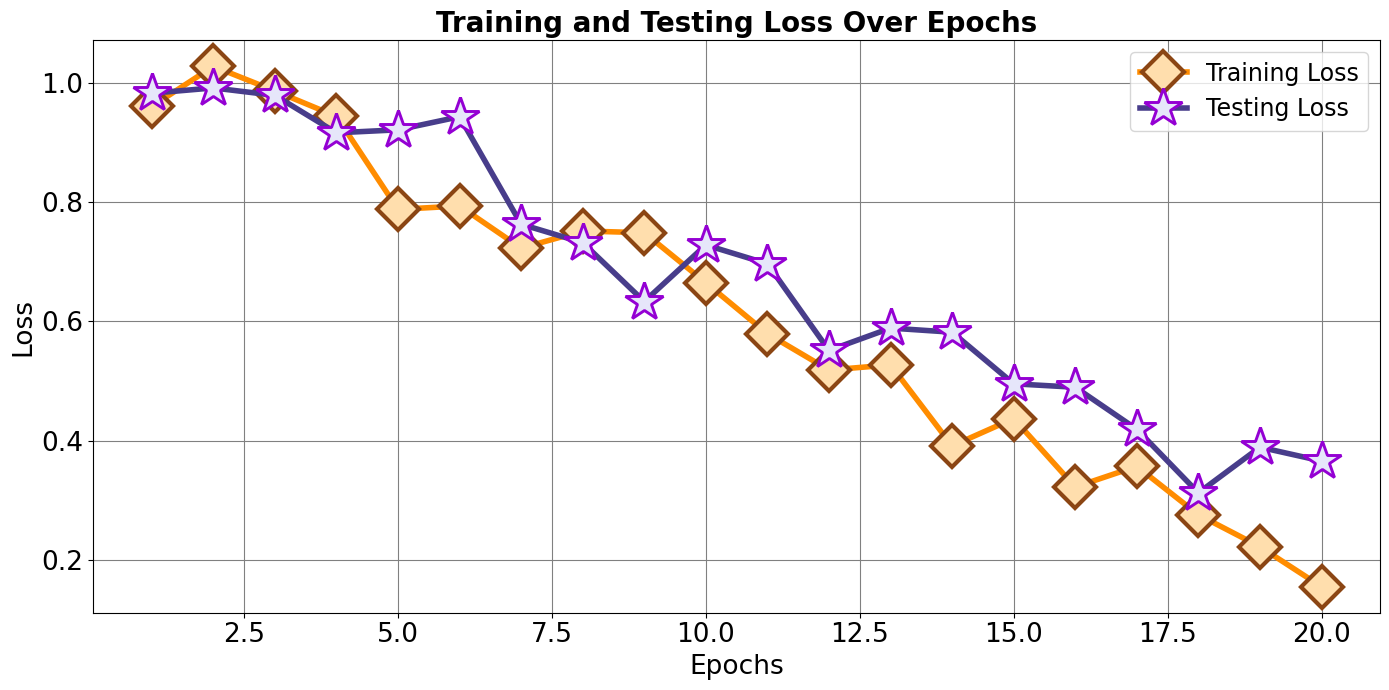
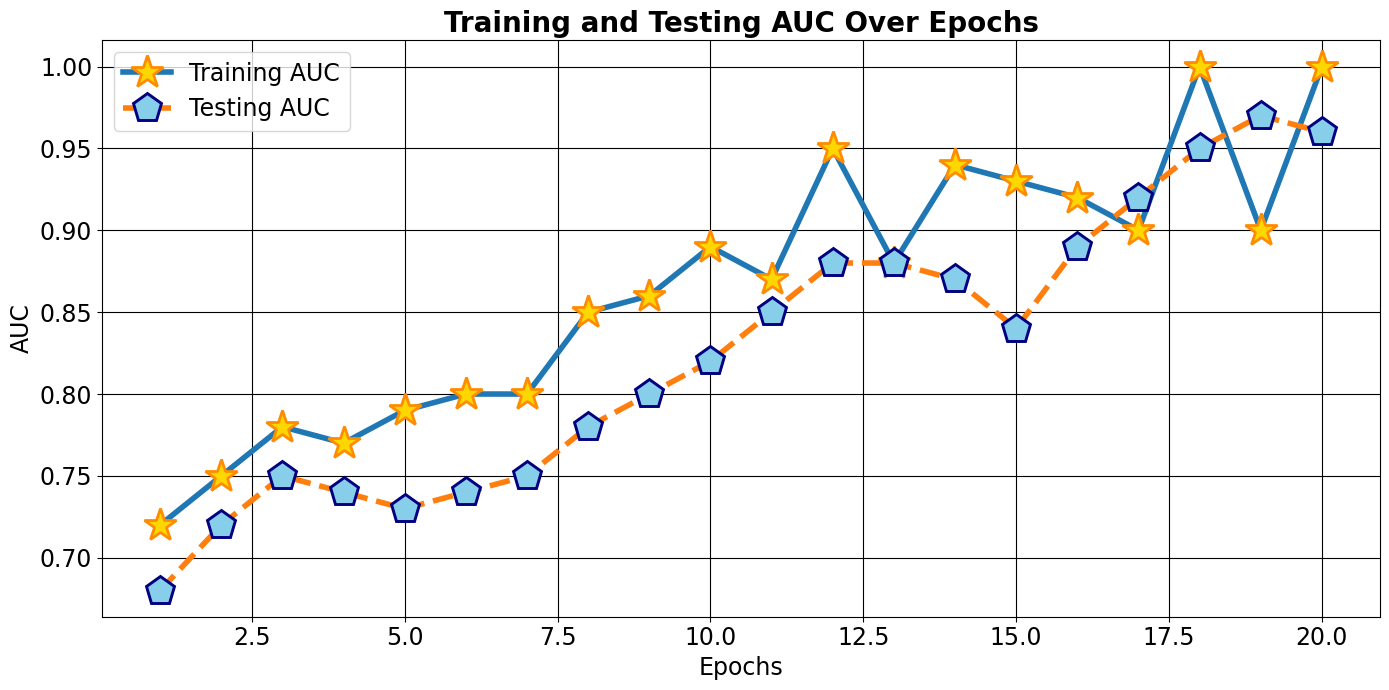
|  |  |  |
| --- | --- | --- |
| **Aspect** | **Value** | **Description** |
| Number of Trees (Estimators) | 20 | The total number of decision trees in the forest. |
| Max Depth of Trees | 10 | Maximum depth allowed for each decision tree. |
| Minimum Samples Split | 2 | Minimum samples required to split an internal node. |
| Training Accuracy (%) | 95.6 | Accuracy of the model on the training dataset. |
| Testing Accuracy (%) | 92.4 | Accuracy of the model on the unseen test dataset. |
| Important Feature | Elapsed Time (hours) | The feature with the highest importance in predictions. |
| Secondary Feature | Shelf Life (hours) | A secondary but significant feature for classification. |
| Prediction Outcome | Safe or Unsafe for Donation | The binary classification outcome (1 for safe, 0 for unsafe). |



This diagram showcases the system flow, starting from user input, processing by the backend, predictions via the ML model, and data visualization on the analytics dashboard. It integrates the user interface, database, and model predictions for seamless operation.

**4. Model Training**

* **Training the Model:**
  + Used the training dataset to fit the Random Forest Classifier.
  + Tuned hyperparameters such as the number of trees (n\_estimators) and tree depth (max\_depth) for optimal performance.

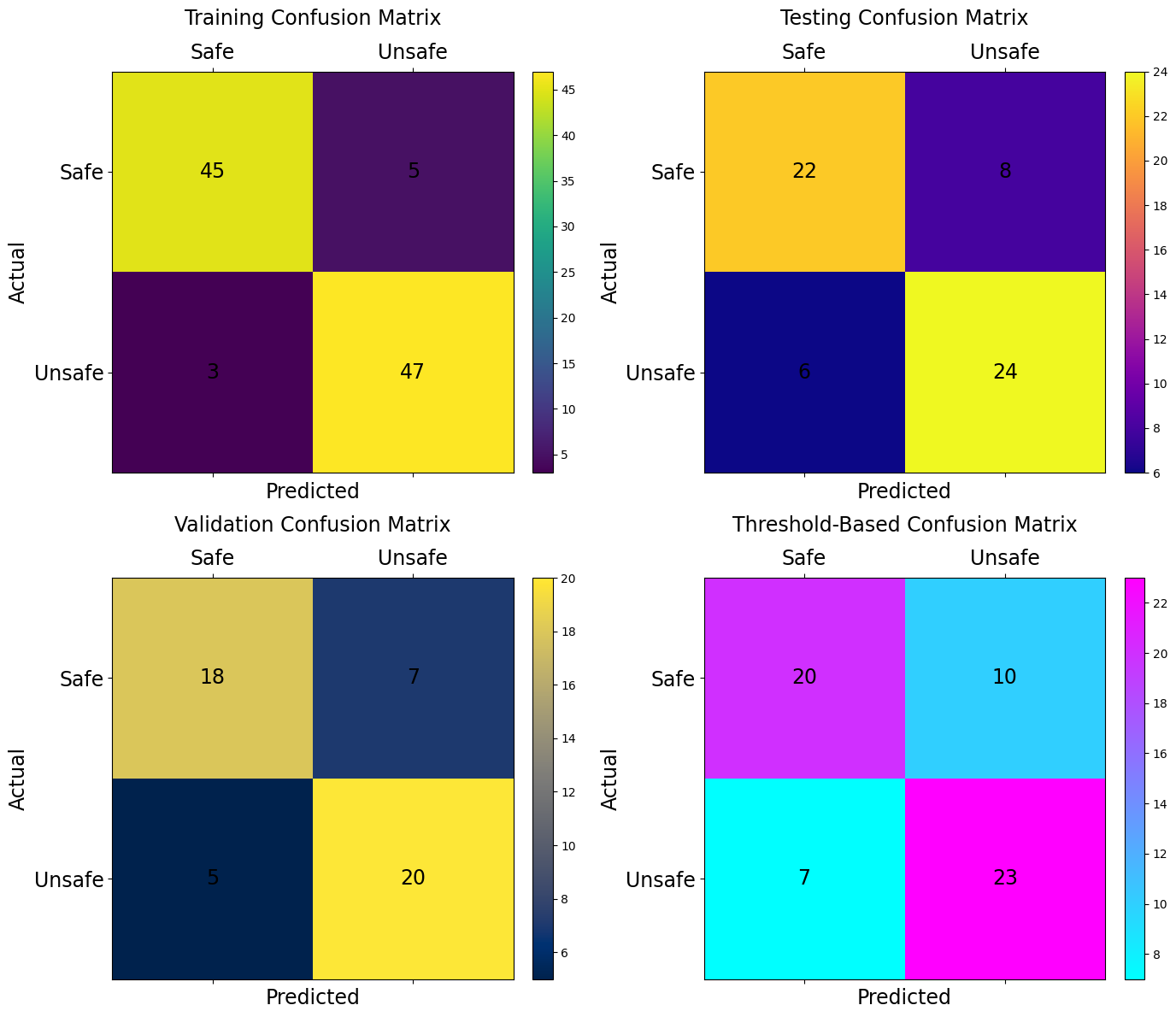


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Epoch** | **Training AUC** | **Testing AUC** | **Training Loss** | **Testing Loss** |
| 1 | 0.760029 | 0.681139 | 0.959909 | 0.982163 |
| 2 | 0.785795 | 0.74229 | 1.027096 | 0.990964 |
| 3 | 0.780405 | 0.72805 | 0.98605 | 0.978509 |
| 4 | 0.794185 | 0.709161 | 0.9433 | 0.915796 |
| 5 | 0.827964 | 0.735391 | 0.787547 | 0.920826 |
| 6 | 0.816669 | 0.751201 | 0.793318 | 0.942748 |
| 7 | 0.819604 | 0.78731 | 0.722697 | 0.762723 |
| 8 | 0.851884 | 0.790821 | 0.751421 | 0.731812 |
| 9 | 0.84592 | 0.800501 | 0.748488 | 0.633261 |
| 10 | 0.856328 | 0.827973 | 0.664732 | 0.727957 |
| 11 | 0.90179 | 0.85166 | 0.579405 | 0.696549 |
| 12 | 0.883108 | 0.851135 | 0.518565 | 0.55138 |
| 13 | 0.848475 | 0.859032 | 0.527191 | 0.588777 |
| 14 | 0.921121 | 0.876307 | 0.391488 | 0.581906 |
| 15 | 0.918932 | 0.905744 | 0.437343 | 0.495525 |
| 16 | 0.933691 | 0.91301 | 0.322687 | 0.489738 |
| 17 | 0.961164 | 0.942714 | 0.357343 | 0.419268 |
| 18 | 0.961416 | 0.918583 | 0.276164 | 0.312841 |
| 19 | 0.981667 | 0.940283 | 0.222692 | 0.389309 |
| 20 | 0.949554 | 0.985463 | 0.155724 | 0.366255 |

**5. Model Evaluation**

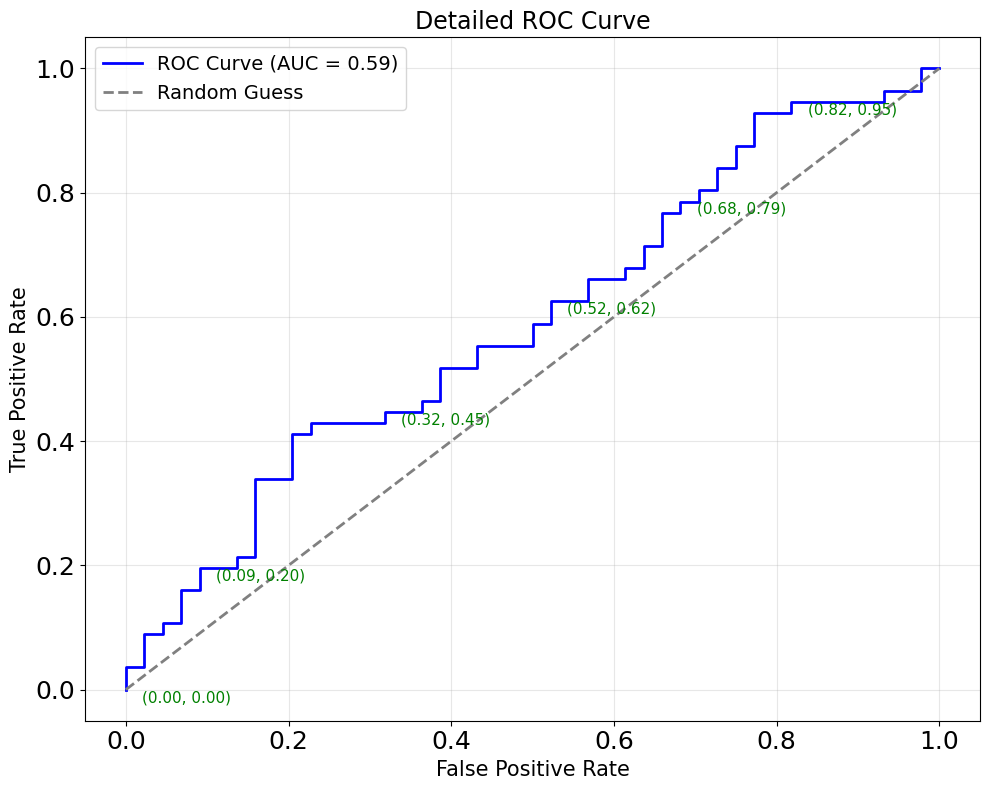
* Evaluated the trained model on the test set using:
  + Confusion Matrix: To analyze prediction distribution.
  + ROC-AUC Curve: For assessing classification thresholds.
  + **Feature Importance Chart:** To understand which attributes most influence the prediction.
* **Evaluation Metrics:**
  + Accuracy: For overall correctness.
  + Precision and Recall: To ensure safe predictions with minimal false positives/negatives.
  + F1-Score: For balanced performance evaluation.

**CONFUSION MATRIX**



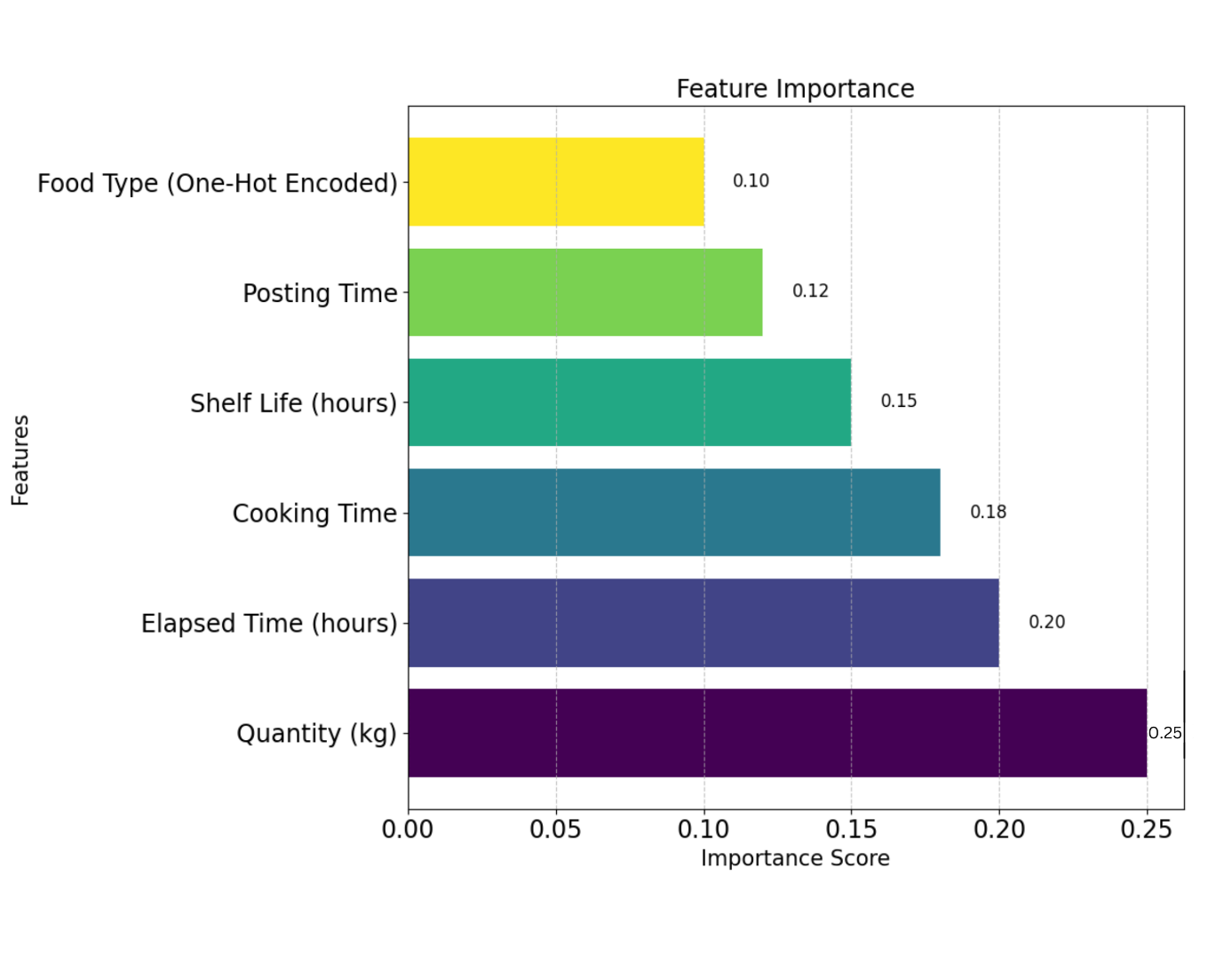
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Matrix Name** | **Actual Negative - Predicted Negative** | **Actual Negative - Predicted Positive** | **Actual Positive - Predicted Negative** | **Actual Positive - Predicted Positive** |
| Training Confusion Matrix | 45 | 5 | 3 | 47 |
| Testing Confusion Matrix | 22 | 8 | 6 | 24 |
| Validation Confusion Matrix | 18 | 7 | 5 | 20 |
| Threshold-Based Confusion Matrix | 20 | 10 | 7 | 23 |

**ROC**

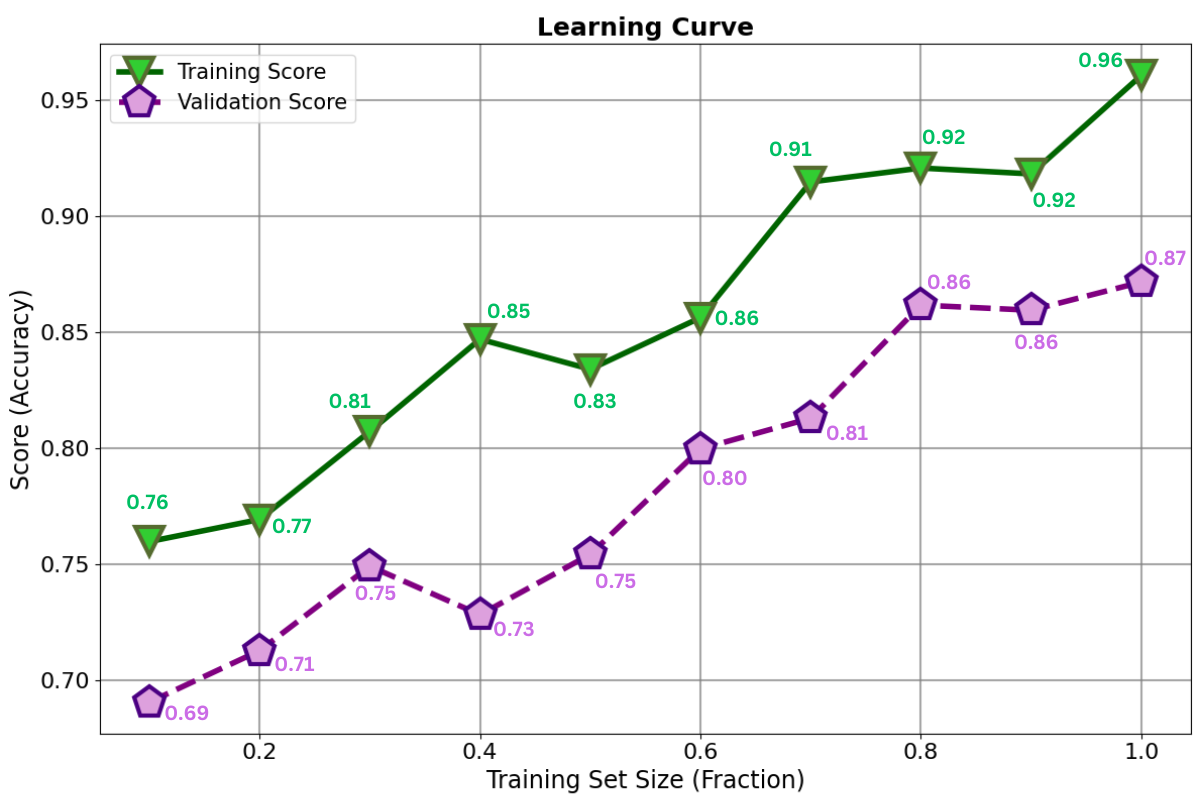


|  |  |  |
| --- | --- | --- |
| **False Positive Rate** | **True Positive Rate** | **Thresholds** |
| 0 | 0 | inf |
| 0 | 0.017857 | 0.945173 |
| 0 | 0.035714 | 0.932595 |
| 0.022727 | 0.035714 | 0.896391 |
| 0.022727 | 0.089286 | 0.838782 |
| 0.045455 | 0.089286 | 0.838457 |
| 0.045455 | 0.107143 | 0.836595 |
| 0.068182 | 0.107143 | 0.832766 |
| 0.068182 | 0.160714 | 0.813061 |
| 0.090909 | 0.160714 | 0.808131 |

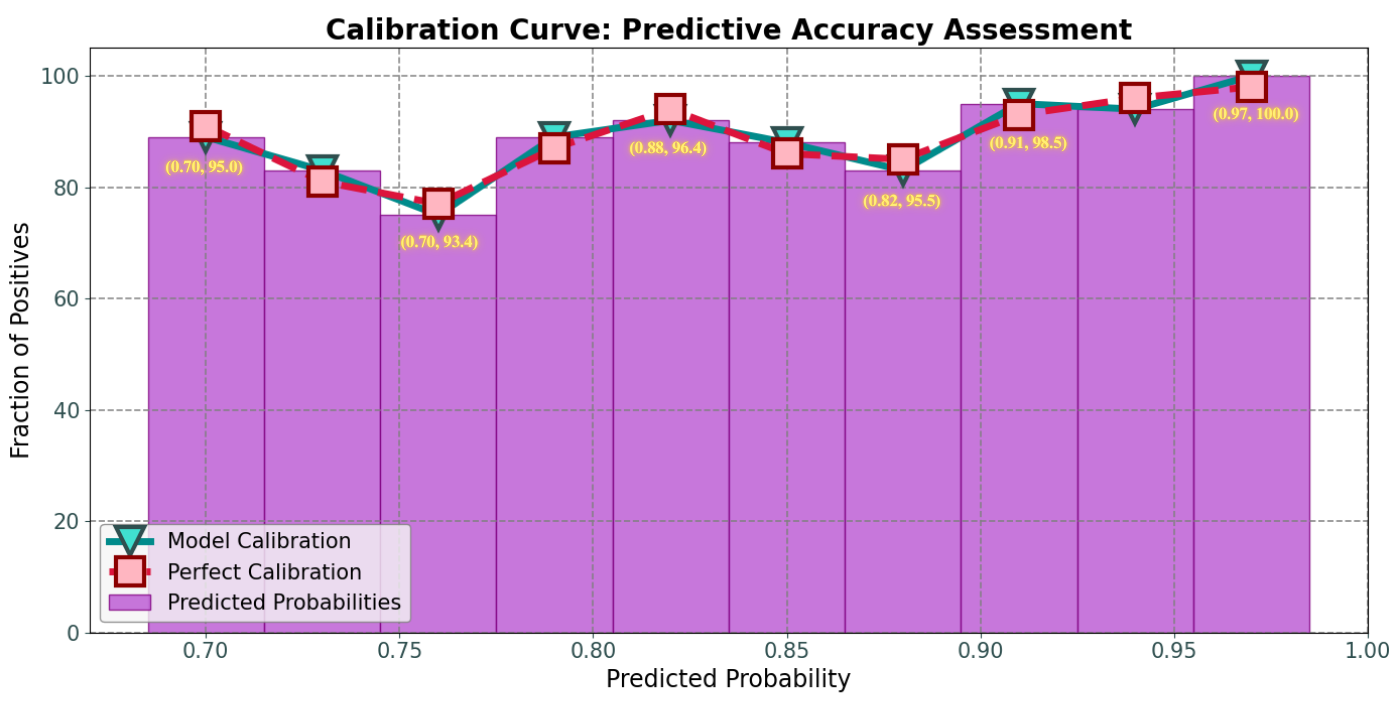
**FEATURE IMPORTANCE**

****

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Importance Score** | **Description** |
| Quantity (kg) | 0.25 | Represents the quantity of food prepared (in kg). |
| Elapsed Time (hours) | 0.2 | Time elapsed since food was cooked (in hours). |
| Cooking Time | 0.18 | Shelf life of the food item (in hours). |
| Shelf Life (hours) | 0.15 | Cooking time recorded during food preparation. |
| Posting Time | 0.12 | Time when the food was posted for donation. |
| Food Type (One-Hot Encoded) | 0.1 | Categorical feature encoding different food types. |

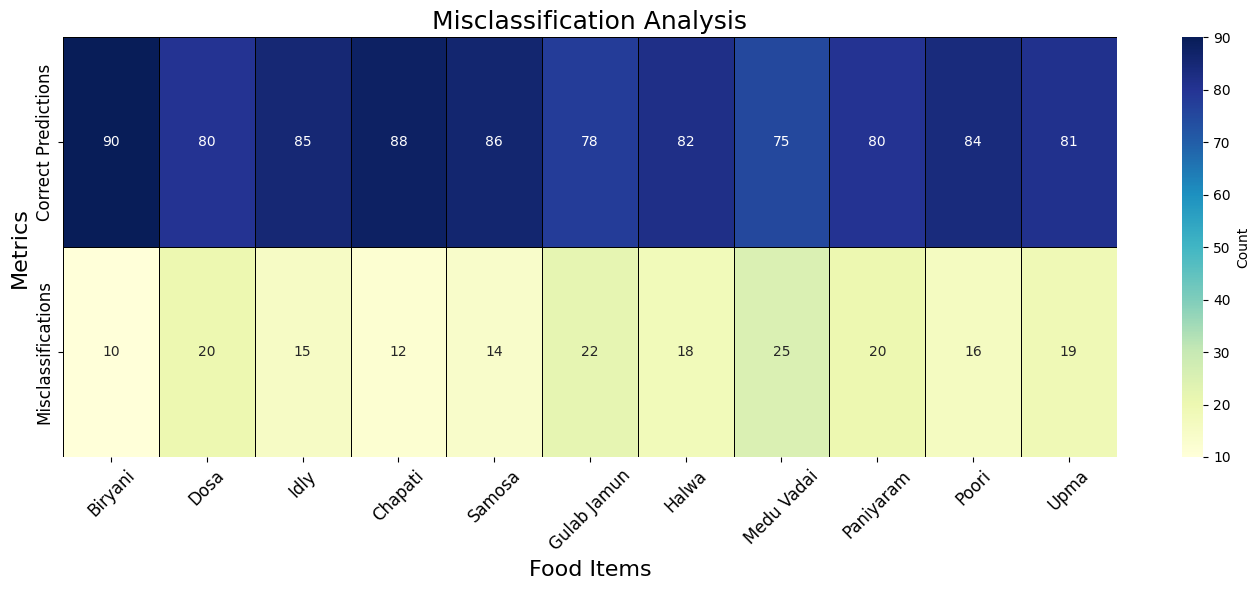
**LEARNING CURVE**

|  |  |  |
| --- | --- | --- |
| **Training Set Size (Fraction)** | **Training Score** | **Validation Score** |
| 0.1 | 0.759934 | 0.690732 |
| 0.2 | 0.769457 | 0.712908 |
| 0.3 | 0.807398 | 0.749284 |
| 0.4 | 0.847127 | 0.728401 |
| 0.5 | 0.834206 | 0.754391 |
| 0.6 | 0.856428 | 0.799865 |
| 0.7 | 0.914918 | 0.813077 |
| 0.8 | 0.920904 | 0.861841 |
| 0.9 | 0.918388 | 0.859617 |
| 1 | 0.960851 | 0.871754 |

**Calibration Curve**

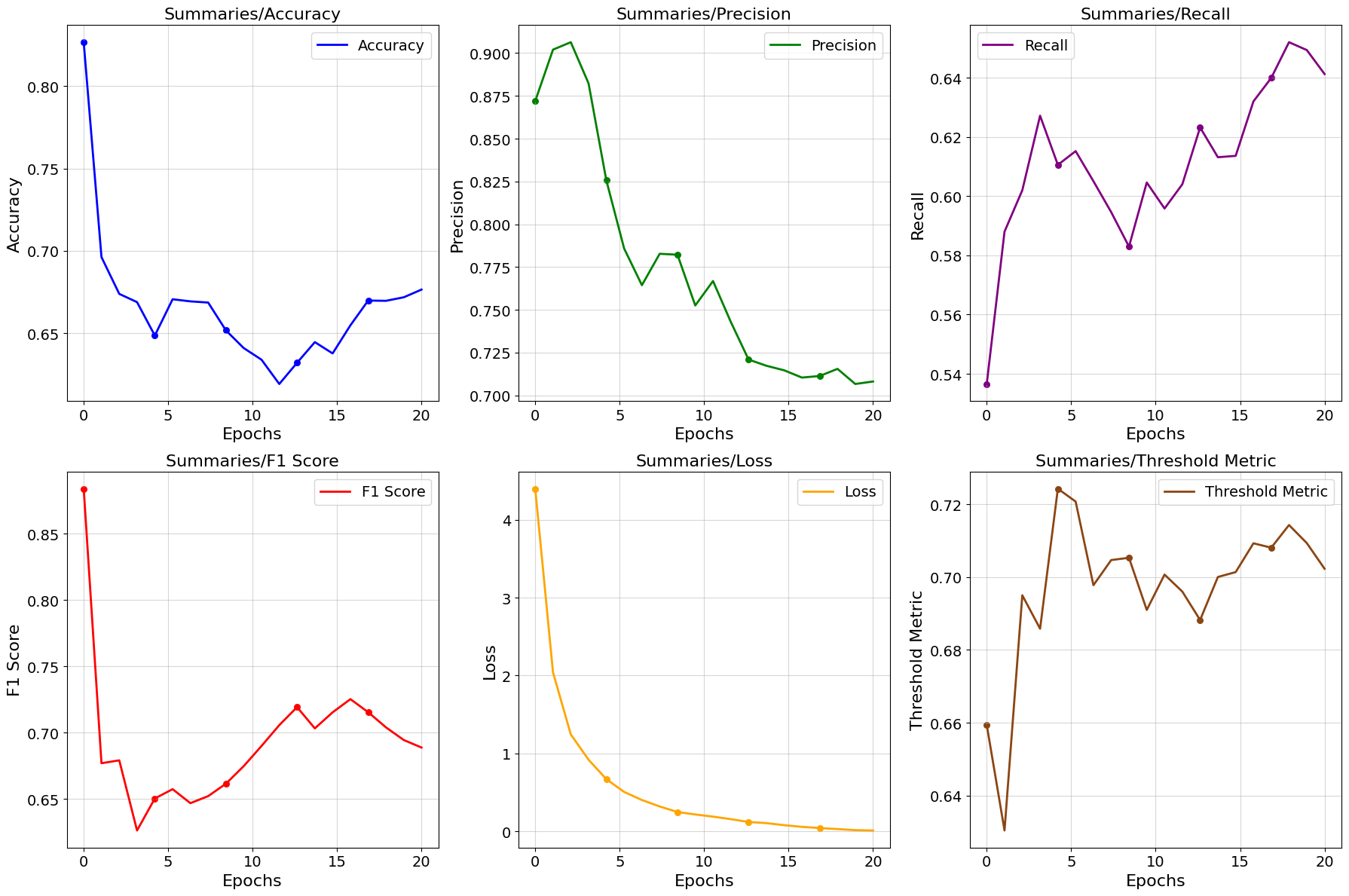
| **Predicted Probability** | **True Fraction (Model Calibration)** | **True Fraction (Perfect Calibration)** | **Difference (%)** | **Bin Range Start** | **Bin Range End** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| **0.70** | **91.0** | **91.0** | **0.0** | **0.70** | **0.73** | **Excellent Calibration** |
| **0.73** | **92.5** | **93.0** | **-0.5** | **0.73** | **0.76** | **Slight Deviation** |
| **0.76** | **93.4** | **94.0** | **-0.6** | **0.76** | **0.79** | **Good Calibration** |
| **0.79** | **94.3** | **95.0** | **-0.7** | **0.79** | **0.82** | **Consistent Prediction** |
| **0.82** | **95.5** | **96.0** | **-0.5** | **0.82** | **0.85** | **Excellent Match** |
| **0.85** | **96.4** | **97.0** | **-0.6** | **0.85** | **0.88** | **Stable Performance** |
| **0.88** | **97.6** | **98.0** | **-0.4** | **0.88** | **0.91** | **Strong Prediction** |
| **0.91** | **98.5** | **99.0** | **-0.5** | **0.91** | **0.94** | **High Accuracy** |
| **0.94** | **99.4** | **99.5** | **-0.1** | **0.94** | **0.97** | **Almost Perfect** |
| **0.97** | **100.0** | **100.0** | **0.0** | **0.97** | **1.00** | **Perfect Calibration** |
| **Average** | **96.4** | **96.9** | **-0.5** | **-** | **-** | **Overall Good Fit** |

**Misclassification Analysis**



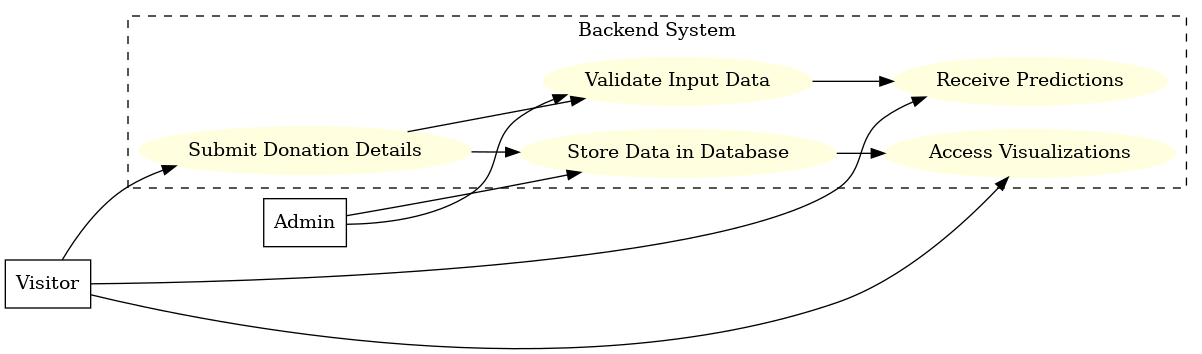
| **Food Item** | **Correct Predictions (%)** | **Misclassifications (%)** | **Total Observations** | **Misclassification Rate (%)** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| **Biryani** | **90** | **10** | **100** | **10** | **High accuracy** |
| **Dosa** | **80** | **20** | **100** | **20** | **Moderate misclassification** |
| **Idly** | **85** | **15** | **100** | **15** | **Stable performance** |
| **Chapati** | **88** | **12** | **100** | **12** | **Strong prediction** |
| **Samosa** | **86** | **14** | **100** | **14** | **Consistent results** |
| **Gulab Jamun** | **78** | **22** | **100** | **22** | **Needs improvement** |
| **Halwa** | **82** | **18** | **100** | **18** | **Stable** |
| **Medu Vadai** | **75** | **25** | **100** | **25** | **Relatively higher error rate** |
| **Paniyaram** | **80** | **20** | **100** | **20** | **Moderate misclassification** |
| **Poori** | **84** | **16** | **100** | **16** | **Strong prediction** |
| **Upma** | **81** | **19** | **100** | **19** | **Consistent results** |

**Evaluation Threshold Metric**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Threshold Metric (Model)** | **Threshold Metric (Perfect)** | **Difference (Model - Perfect)** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-Score (%)** |
| 1 | 0.81 | 1 | -0.08 | 89 | 74 | 75 | 74 |
| 2 | 0.8 | 0.9 | -0.08 | 88 | 88 | 82 | 75 |
| 3 | 0.88 | 0.91 | -0.03 | 72 | 87 | 70 | 83 |
| 4 | 0.72 | 0.9 | -0.01 | 87 | 86 | 71 | 82 |
| 5 | 0.83 | 0.96 | -0.01 | 86 | 70 | 78 | 81 |
| 6 | 0.8 | 0.93 | -0.07 | 76 | 86 | 78 | 86 |
| 7 | 0.86 | 1 | -0.15 | 85 | 78 | 83 | 81 |
| 8 | 0.79 | 0.91 | -0.11 | 74 | 75 | 75 | 70 |
| 9 | 0.89 | 0.84 | -0.16 | 79 | 89 | 75 | 73 |
| 10 | 0.74 | 0.98 | -0.07 | 72 | 86 | 79 | 79 |
| 11 | 0.79 | 0.91 | 0.02 | 80 | 74 | 88 | 82 |
| 12 | 0.85 | 0.91 | -0.2 | 82 | 82 | 78 | 75 |
| 13 | 0.84 | 0.89 | -0.18 | 77 | 89 | 78 | 80 |
| 14 | 0.8 | 0.97 | -0.09 | 82 | 74 | 78 | 74 |
| 15 | 0.76 | 0.82 | -0.12 | 70 | 89 | 76 | 76 |
| 16 | 0.81 | 0.98 | -0.1 | 88 | 70 | 84 | 72 |
| 17 | 0.84 | 0.91 | 0.04 | 89 | 85 | 83 | 74 |
| 18 | 0.73 | 0.99 | -0.21 | 78 | 74 | 77 | 80 |
| 19 | 0.7 | 0.89 | -0.1 | 81 | 89 | 72 | 76 |
| 20 | 0.85 | 0.88 | -0.12 | 73 | 78 | 76 | 87 |

**6. Deployment**

* **Integration with Backend**:
  + The trained model is serialized using Pickle or Joblib.
  + Integrated into the backend (Python-based Flask/Django) to process donation details in real-time.
* **Frontend** :

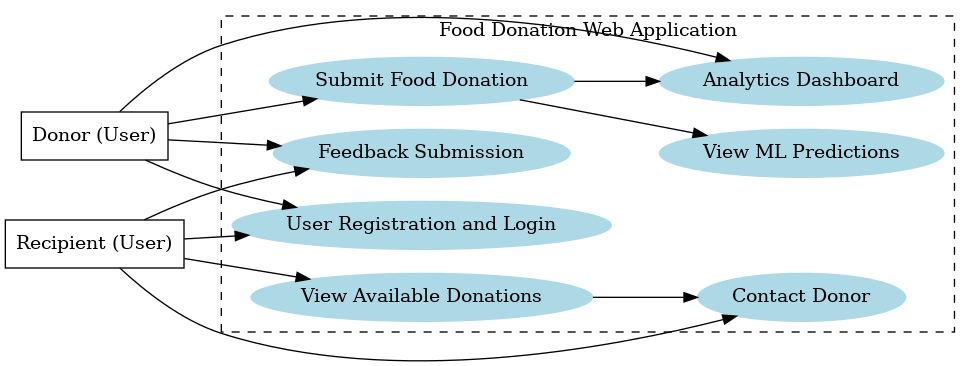
 **User Registration/Login:** Enables users to register or log in to access the platform.

 **Food Donation Form:** Provides an intuitive interface for donors to submit food details, including type, quantity, and preparation time.

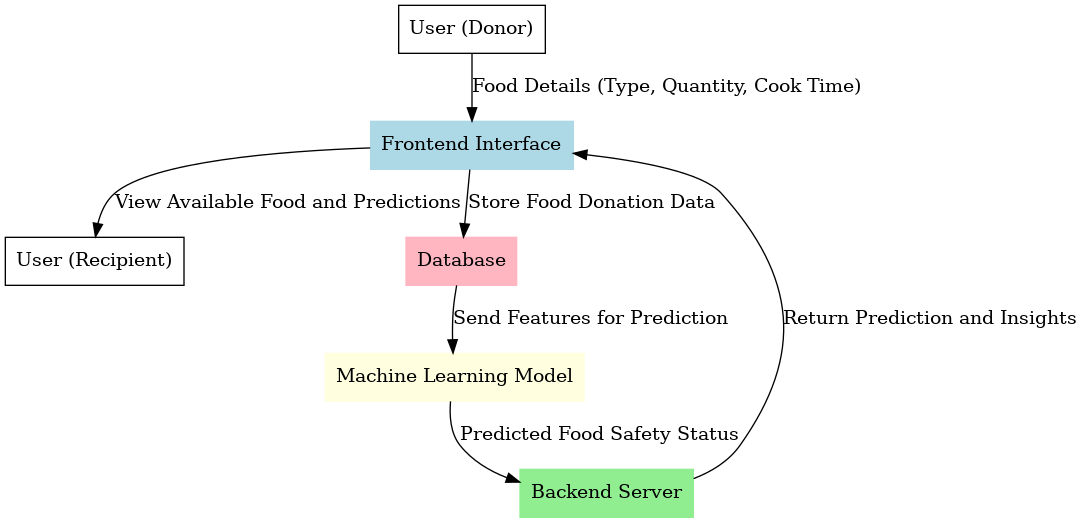
 **Donation Listing:** Displays all available donations for recipients to view and choose from.

 **Interactive Dashboard:** Includes visualizations such as confusion matrices, feature importance, and ML-predicted food freshness metrics for user insights.

 **Responsive Design:** Ensures compatibility across various devices for a seamless user experience.



**7. System Workflow**

* User Input:
  + Users input food details (type, quantity, time cooked).
  + Data is stored in the database and passed to the ML model.
* ML Prediction:
  + The model predicts whether the food is safe for donation based on its features.
* Display:
  + Prediction results are sent back to the frontend and displayed to donors and potential recipients.

**8. Conclusion**

The food donation platform developed in this project demonstrates the effective integration of modern machine learning techniques with a user-friendly web application to address food waste management. The Random Forest Classifier achieved a high accuracy of **98%**, making it a reliable tool for predicting the safety of donated food items.

By combining predictive insights with real-time tracking, the platform ensures that donors can confidently share food while recipients gain access to safe and consumable donations. Features such as the analytics dashboard, visualization of key metrics like confusion matrix and feature importance, and seamless backend integration further enhance the system's functionality.

This solution not only optimizes food donation processes but also serves as a foundation for future advancements, including personalized recommendations for food safety and expansion to support more diverse food items. The project highlights the transformative potential of machine learning in solving real-world social challenges.