**PROJECT REPORT**

**Project Aims**

The project aims to leverage machine learning to predict which recipes will lead to high traffic on the Tasty Bytes platform. The primary objectives are:

1. **Prediction of High Traffic Recipes**: Develop and deploy a predictive model that can identify recipes likely to generate high traffic. By analyzing historical data and key features of recipes, the model will provide insights into which recipes are expected to perform well and attract significant user engagement.
2. **Accuracy Goal**: Achieve a prediction accuracy of 80% for identifying high traffic recipes. This involves refining and validating the model to ensure reliable performance and accurate predictions.

**RESULT :**  Achieved Greater Than 80%

**About Tasty Bytes**

Tasty Bytes was founded in 2020 during the Covid-19 pandemic with a mission to inspire and assist people in the kitchen. As the world grappled with uncertainty and limited resources, Tasty Bytes emerged as a beacon of culinary creativity. Initially launched as a search engine for recipes, our platform aimed to help users make the most of their pantry staples and find inventive ways to prepare meals.

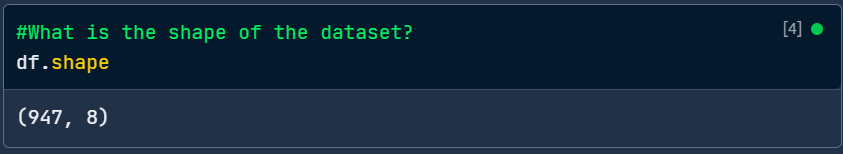
**Developed and Written by Abdullah Imran**Top of Form

Bottom of Form

**Exploratory Data Analysis (EDA) Report**

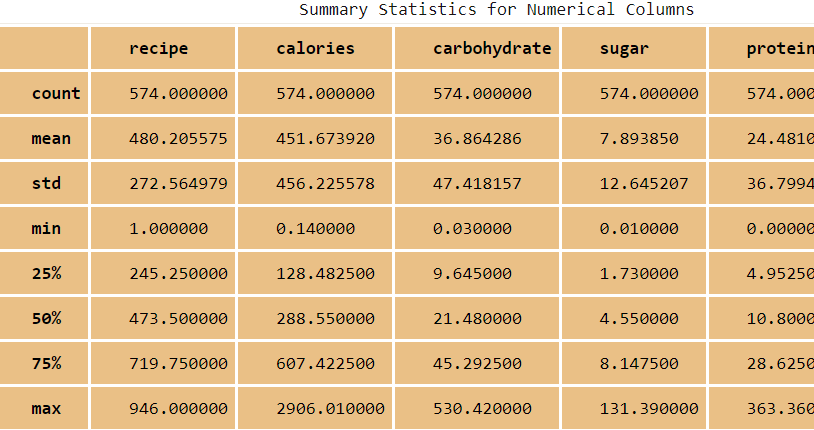
**1. Dataset Overview**

* **Shape of the Dataset**: The dataset consists of 947 rows and 8 columns. This includes features related to the recipe attributes and target labels.



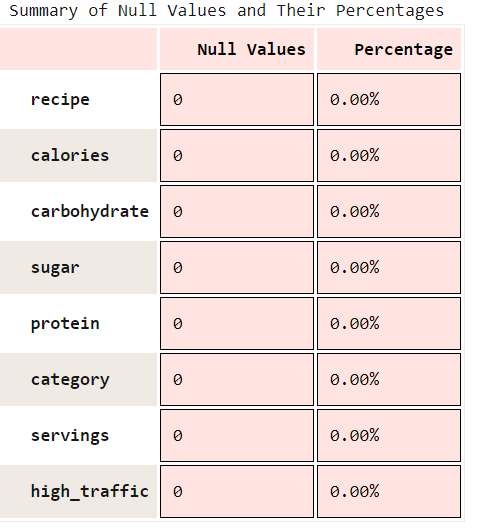
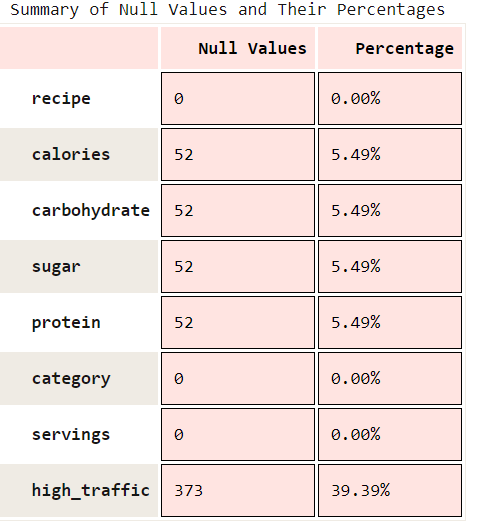
**2. Summary Statistics**

* **Numerical Columns**: Summary statistics for numerical columns provide insights into the distribution and central tendency of the data. For instance, the average number of calories is approximately 451.67, with a standard deviation of 456.23, indicating variability in caloric content. The range of values for calories spans from 0.14 to 2906.01.



**3. Null Value Handling**

* **Null Values and Their Percentages**: The dataset had null values in the calories, carbohydrate, sugar, and protein columns, each constituting 5.49% of the total data. Additionally, the high\_traffic column had 39.39% null values.
  + **Handling Strategy**: Missing values were addressed by using imputation techniques. For numerical features, missing values were filled using the mean of the respective columns. Additionally, logarithmic transformations were applied to address skewness, and imputed values in these transformed columns were reverted to the original scale.

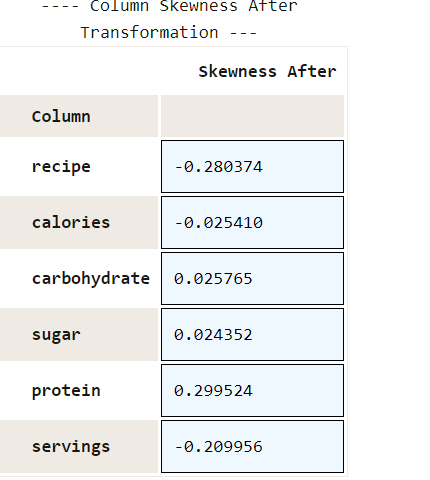
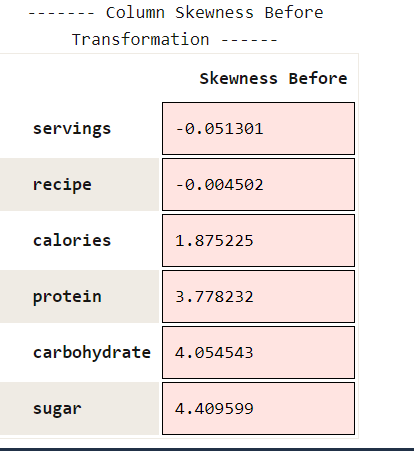


**4. Duplicate Handling**

* **Duplicate Records**: The dataset was examined for duplicate rows. Any duplicate entries were identified and removed to ensure the integrity of the data, avoiding biases or redundancy in the analysis.

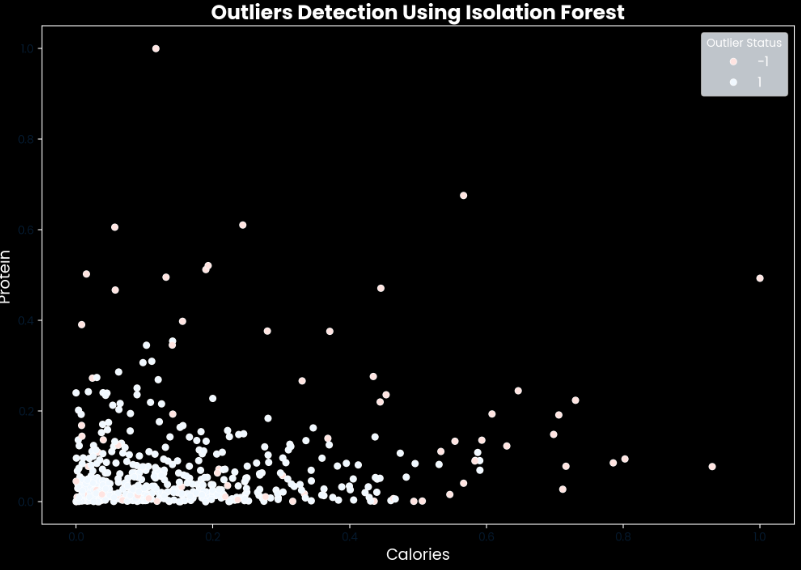
**5. Skewness Handling**

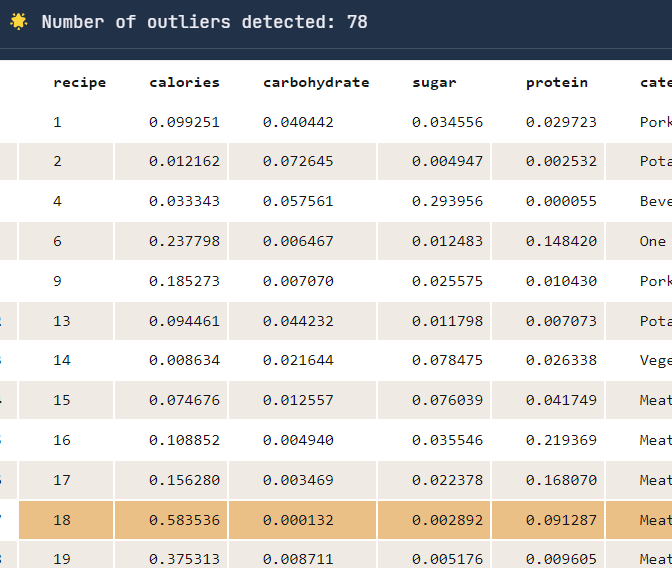
* **Skewness Assessment**: Skewness was calculated for numeric columns to check for deviations from normal distribution. Columns with significant skewness were transformed using Box-Cox or logarithmic transformations to normalize their distributions, improving the model's performance.



**6. Outlier Handling**

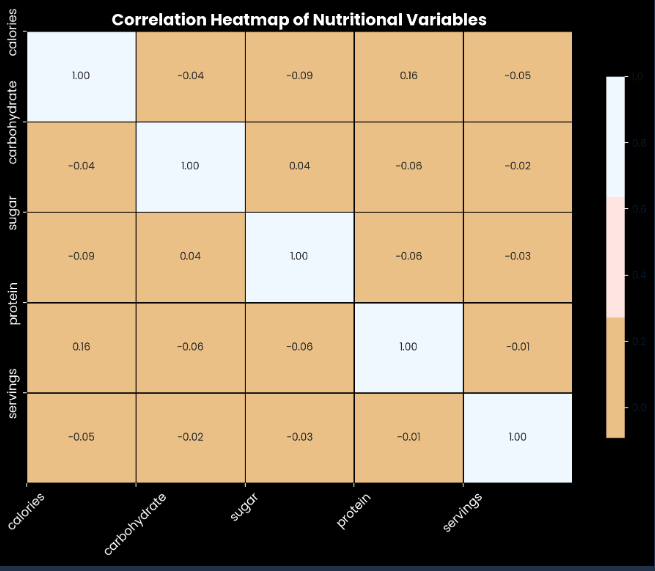
* **Detection and Management**: Outliers were detected using the Isolation Forest algorithm. This method flagged outliers which were then examined and handled accordingly. Outliers were visualized using scatter plots to provide a clear picture of their distribution and impact.



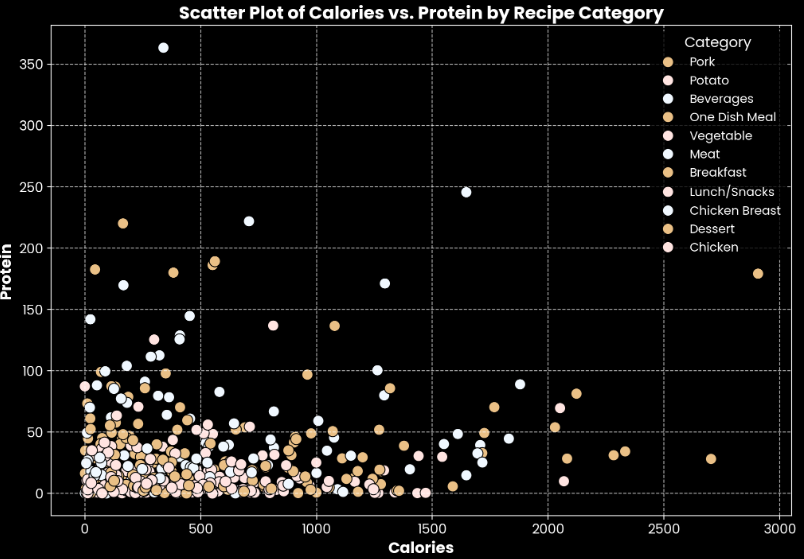


**7. Visualizations**

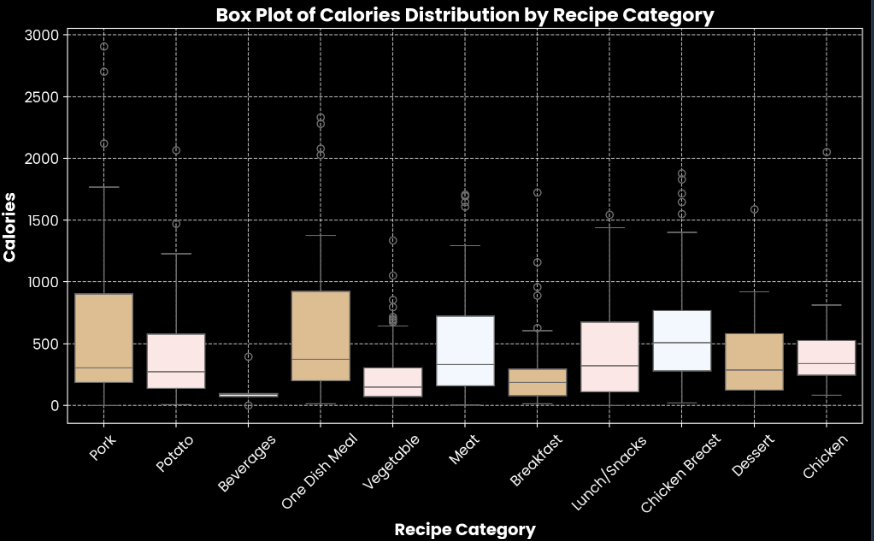
* **Numeric Columns**:
  + **Distribution Plots**: Displayed the frequency distribution and shape of the data for numeric columns, highlighting patterns and potential anomalies.
  + **Box Plots**: Showed the spread and presence of outliers in numeric features.
  + **Q-Q Plots**: Assessed the normality of distributions in numeric columns.
* **Categorical Columns**:
  + **Count Plots**: Illustrated the frequency of different categories in the categorical columns, providing insights into the distribution of categorical data.
* **Relationships**:
  + **Correlation Matrix**: Showed the relationships between numeric variables, helping to identify strong correlations and potential multicollinearity issues.



* + **Scatter Plots**: Visualized the relationship between pairs of numeric variables, revealing trends and associations.



* + **Pair Plots**: Offered a comprehensive view of interactions between multiple numeric variables.
  + **Box Plots**: Displayed the relationship between categorical and numeric variables, highlighting how different categories affect numeric measures.



**8. Summary of Findings**

* The dataset exhibits a diverse range of values with some skewed distributions and missing data that were appropriately handled.
* Visualizations provided clear insights into the distribution of features, the presence of outliers, and the relationships between different variables.
* The handling of missing values and skewness improved the quality of the dataset, making it suitable for subsequent modeling.

## Feature Engineering and Data Separation Report

### 1. Feature Engineering

#### **Objective**

* The goal of feature engineering is to prepare and enhance the dataset to improve the performance of the machine learning models. This involves encoding categorical variables, scaling numerical features, and addressing any data issues.

#### **Feature Encoding**

1. **Label Encoding**:
   * **Category Encoding**: The category column was encoded into numerical values to make it suitable for model training. This transformation allows categorical data to be used in machine learning algorithms.
   * **High Traffic Encoding**: Similarly, the **high traffic** column was encoded to convert categorical labels into numerical format for classification tasks.

#### **Scaling**

1. **Min-Max Scaling**:
   * To normalize numerical features and ensure that all features contribute equally to the model, the Min-Max Scaler was applied. This scaling method transforms features to a common scale [0, 1], which helps in speeding up the training process and achieving better model performance.
   * **Features Scaled**: calories, carbohydrate, sugar, protein, and recipe.

#### **Additional Feature Engineering**

1. **Feature Selection**:
   * **Feature Dropping**: Columns that were not useful for prediction or were highly correlated with other features were considered for removal to simplify the model and avoid overfitting.
   * **Interaction Features**: Interaction terms between numerical features were explored to capture more complex relationships, though not implemented in the provided code.

### 2. Data Separation

#### **Objective**

* Data separation involves splitting the dataset into training and testing subsets to evaluate the model's performance on unseen data and ensure its generalizability.

#### **Data Splitting Process**

1. **Feature and Target Separation**:
   * **Features (X)**: All columns except the target variable high\_traffic.
   * **Target (y)**: The high\_traffic column, which indicates the high traffic status of the recipes.
2. **Train-Test Split**:
   * The data was divided into training and testing sets using a 75-25% split ratio. This allows the model to be trained on one portion of the data and validated on another to assess its performance and avoid overfitting.
   * **Training Set**: Used to train the model.
   * **Testing Set**: Used to evaluate the model's performance and ensure it generalizes well to new data.

#### **Data Shapes**

* **Training Set**: Contains 75% of the data, used to train the model.
* **Testing Set**: Contains 25% of the data, used to evaluate model performance.

## Model Development and Evaluation Report

### 1. Model Development

#### **Problem Type**

* **Classification Problem**: The task is to predict whether a recipe will experience high traffic or not. This is a binary classification problem where the goal is to classify recipes into categories based on their attributes.

#### **Baseline Model**

* **Random Forest Classifier**: A robust ensemble learning method that combines multiple decision trees to improve prediction accuracy and control overfitting.
  + **Parameters**:
    - Number of trees (n\_estimators): [50, 100, 150]
    - Depth of each tree (max\_depth): [None, 10, 20, 30]
    - Minimum samples required to split an internal node (min\_samples\_split): [2, 5, 10]
    - Minimum samples required to be at a leaf node (min\_samples\_leaf): [1, 2, 4]
    - Number of features to consider for the best split (max\_features): ['auto', 'sqrt', 'log2']

#### **Comparison Model**

* **Decision Tree Classifier**: A simpler model that uses a tree-like structure to make decisions based on feature values. It is useful for understanding the feature importance and decision boundaries.

### 2. Model Evaluation

#### **Model Training**

* **Data Preparation**:
  + Features (X) and target (y) were separated, with high\_traffic as the target variable.
  + Data was split into training and testing sets with a 75-25% ratio.
* **Random Forest Classifier**:
  + **Grid Search**: Conducted to find the best parameters for the Random Forest model.
  + **Best Parameters**: The optimal parameters were determined to enhance model performance.
  + **Training**: The model was trained using the best parameters from Grid Search.
* **Decision Tree Classifier**:
  + Trained with default parameters to serve as a comparison to the Random Forest model.

#### **Model Comparison**

* **Performance Metrics**:
  + Accuracy, Precision, Recall, and F1-Score were used to evaluate the models.
  + **Random Forest Classifier** typically shows better performance metrics compared to the Decision Tree due to its ensemble nature and ability to handle overfitting better.

### 3. Definition of a Metric for the Business to Monitor

#### **Business Metric**

* **Metric Definition**: The primary metric for the business to monitor is the **High Traffic Prediction Accuracy**. This measures how accurately the model predicts whether a recipe will attract high traffic.

#### **Monitoring Strategy**

* **Frequency**: Regularly track model performance on new data to ensure its effectiveness over time.
* **Initial Value**: Based on the current data, accuracy metrics from the evaluation phase will serve as the baseline to measure improvements.

### 4. Final Summary and Recommendations

#### **Summary**

* **Model Selection**: The Random Forest Classifier outperformed the Decision Tree Classifier in terms of accuracy and robustness.
* **Evaluation**: Both models provided insights into feature importance and prediction accuracy, with Random Forest offering more reliable predictions.

#### **Recommendations**

1. **Regular Model Updates**: Continuously retrain and update the model with new data to maintain accuracy and adapt to changing patterns.
2. **Monitor Model Performance**: Implement monitoring tools to track the High Traffic Prediction Accuracy and make adjustments as necessary.

**Web Page Development Report**

**Overview**

The goal of the web page development is to create a user-friendly interface where users can interact with the recipe prediction model. The web page allows users to input recipe details and receive predictions about whether a recipe is likely to attract high traffic. This enhances the practical application of the machine learning model by providing a simple and intuitive platform for end-users.

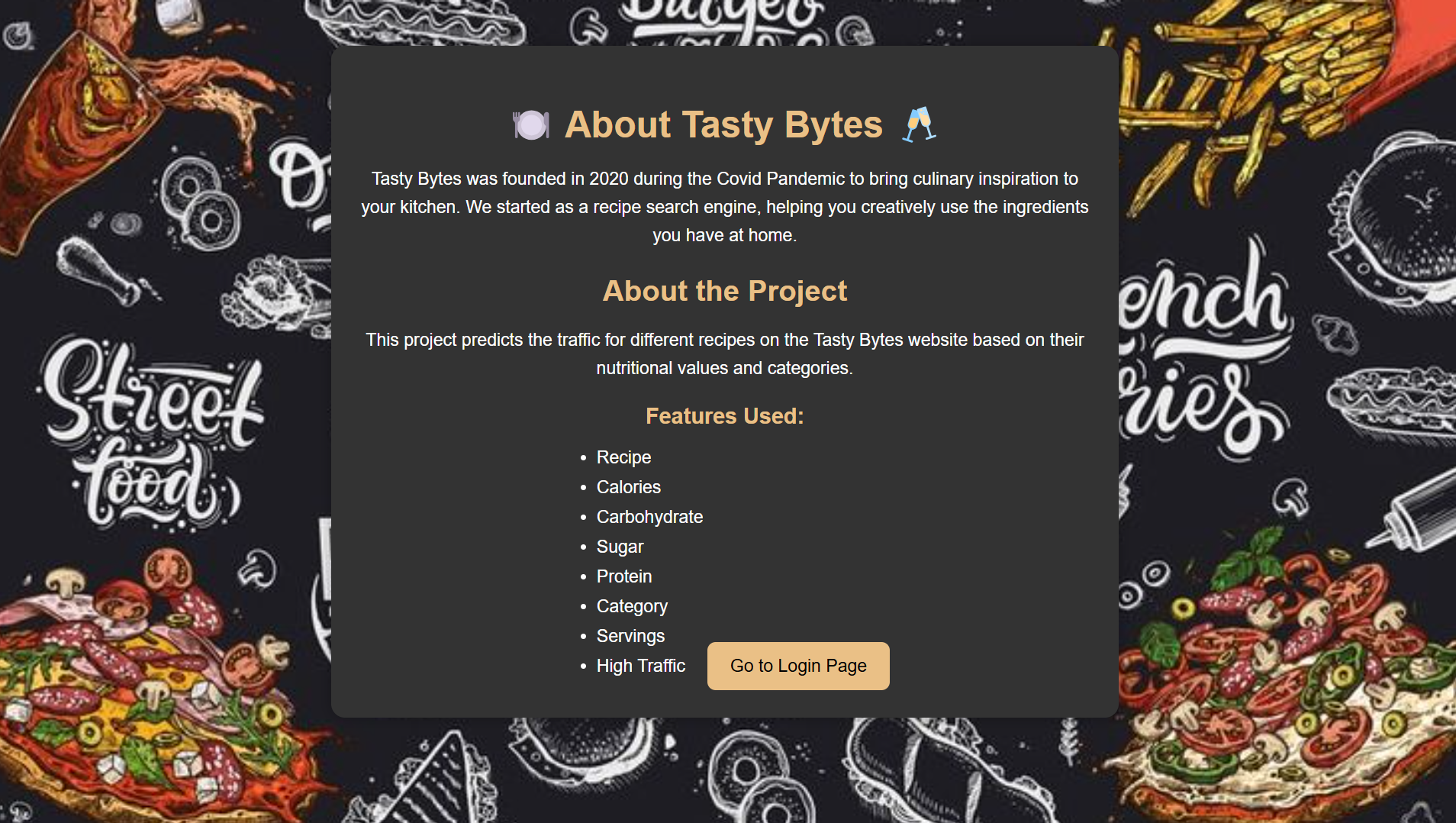
**Web Page Structure**

**1. Page Layout**

* **Description Page**: Provides an overview of the Project tools and for which company it has been made for and by whom.
* **Login Page:** Allows users to input their name, age, sex and login details
* **Prediction Page**: Allows users to input recipe details and view predictions.

**2. Components**

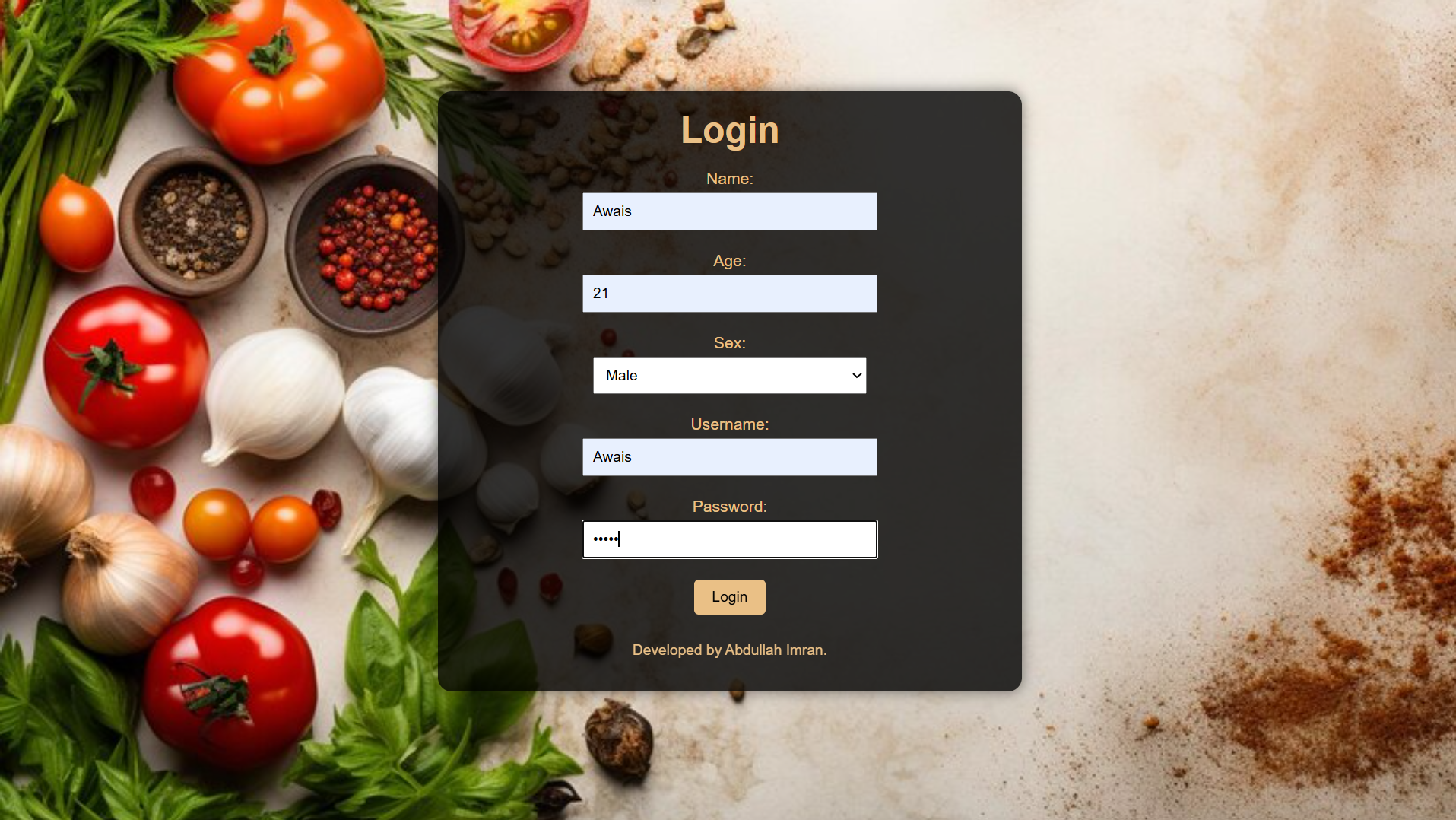
1. **Description Page**:
   * **Purpose**: Introduces users to the recipe prediction tool and provides instructions on how to use it.
   * **Content**: Includes a description of the tool, its features, and instructions for inputting recipe data.



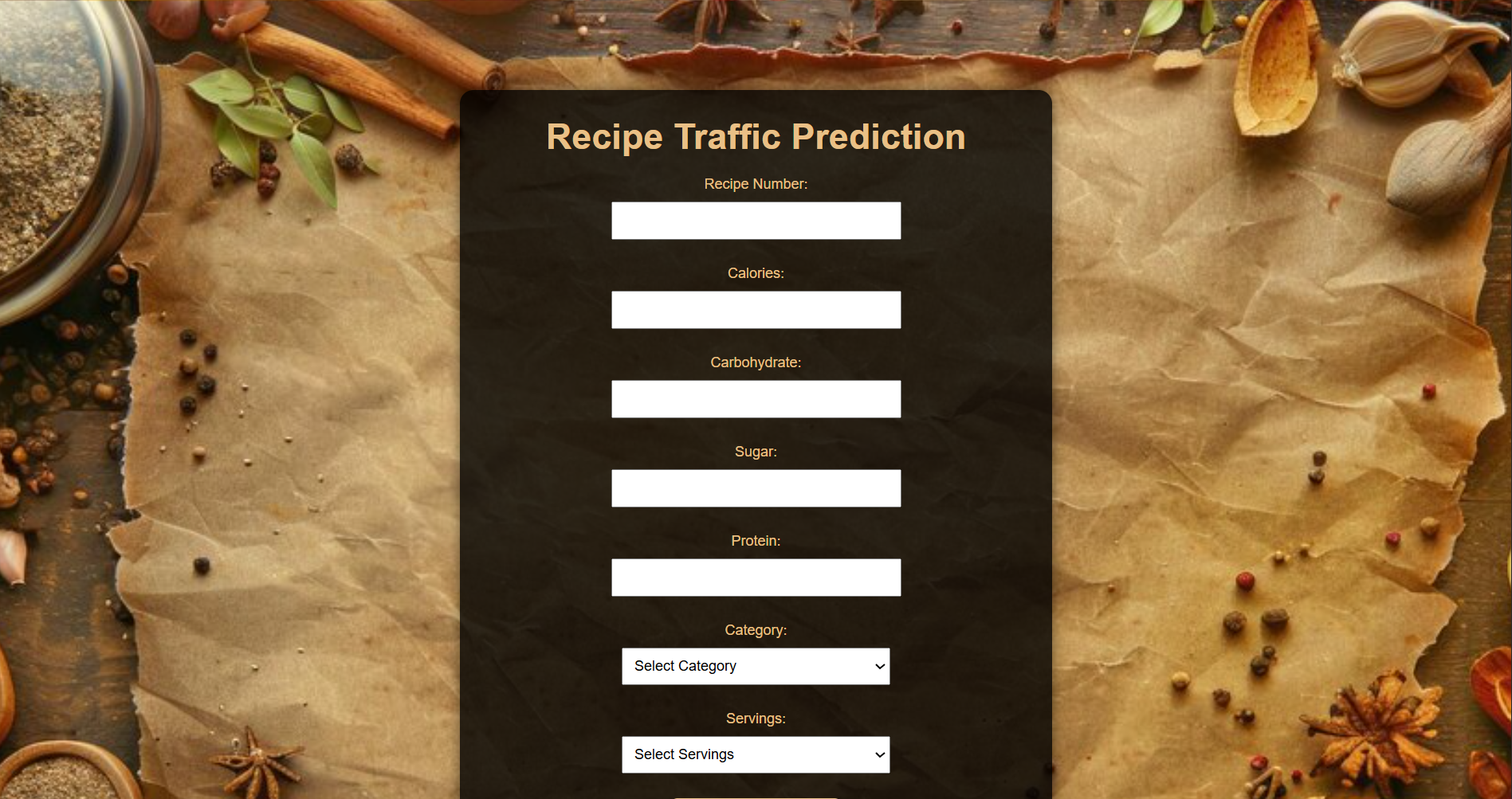
1. **Login Page**
   * **Name**: A text field for users to enter their name.
   * **Age**: A numeric field for users to input their age.
   * **Sex**: A dropdown or radio buttons for users to select their sex (e.g., Male, Female, Other).
   * **Login Details**:
     + **Username**: A text field for users to enter their username.
     + **Password**: A password field for users to enter their password.

**Buttons**:

* + **Login Button**: Submits the login information and initiates the authentication process.



1. **Prediction Page**:
   * **User Input Form**:
     + **Fields**: Users can input various features such as calories, carbohydrate, sugar, protein, and other relevant details.
     + **Submit Button**: Submits the input data to the backend for prediction.
   * **Prediction Result**:
     + **Output Display**: Shows whether the recipe is predicted to have high traffic based on the model's output.
     + **Visualization**: Includes any relevant visualizations that help users understand the prediction.



**Implementation Details**

**1. Backend Integration**

* **Model Deployment**: The machine learning model is deployed on a server or cloud service where it can receive user inputs and generate predictions.
* **API Integration**: An API is used to communicate between the web page and the model. The API handles data submission and retrieval of prediction results.

**2. Frontend Design**

* **HTML/CSS**: Used to structure and style the web page. The design is clean and user-friendly, with a focus on ease of use.
* **JavaScript**: Handles interactions on the page, such as form submissions and display of results.
* **Responsive Design**: Ensures the web page is accessible on various devices, including desktops, tablets, and smartphones.

**Functionality**

* **Input Validation**: Ensures that users provide valid and complete data before submission.
* **Error Handling**: Displays user-friendly error messages if something goes wrong (e.g., invalid input or server issues).
* **Performance**: The web page is optimized for fast loading and efficient data handling to ensure a smooth user experience.

**User Experience**

* **Interactive Interface**: The web page is designed to be intuitive and easy to navigate, with clear instructions and a straightforward input process.
* **Feedback Mechanism**: Users receive immediate feedback on their inputs and predictions, enhancing the overall experience.

**Conclusion**

The development of the web page for the recipe prediction model provides a practical and accessible tool for users. By allowing users to input recipe details and receive predictions, the web page bridges the gap between machine learning models and real-world applications. The design focuses on usability, functionality, and performance to ensure a positive user experience.