Milestone 2 Report: Sustainabite

Date: July 30, 2025

Group: Sustainabite

Members: Atharva Sangani (aas44@sfu.ca)

1. Project Recap

Sustainabite is an AI-powered web app that predicts food expiry dates based on item type, storage conditions, and packaging freshness. Using a Random Forest model trained on synthetic data, it forecasts shelf life with confidence scores to help users manage their food inventory.

In addition to expiration predictions, Sustainabite also provides smart recipe recommendations that make use of ingredients approaching their expiry. By identifying which items should be consumed sooner, the system helps users reduce household food waste and plan meals more efficiently.

2. Significant Accomplishments

Accomplishment 1: Synthetic Data Generation & Preprocessing

- Generated 5,000 synthetic food items with realistic expiry characteristics across 7 categories (dairy, meat, bakery, etc.)
- Engineered features: storage_condition, opened_status, item_type, and initial_quality
- Normalized distributions to mirror real-world shelf life (e.g., bakery: 2-5 days, canned: 1+ years)

Accomplishment 2: Model Training & Evaluation

- Trained a RandomForestRegressor (200 trees) with 5-fold cross-validation
- Achieved MAE: 2.44 days and R^2: 1 on test data
- Implemented confidence scoring using tree variance

Accomplishment 3: Web App Integration

- Built Flask backend with SQLite database
- Developed responsive UI with Bootstrap
- Implemented real-time expiry prediction + recipe suggestions

3. Proof of Accomplishment

Proof 1: Data Generation & Distribution

Below is a code snippet used to generate the synthetic data on which the expiry prediction model is trained.

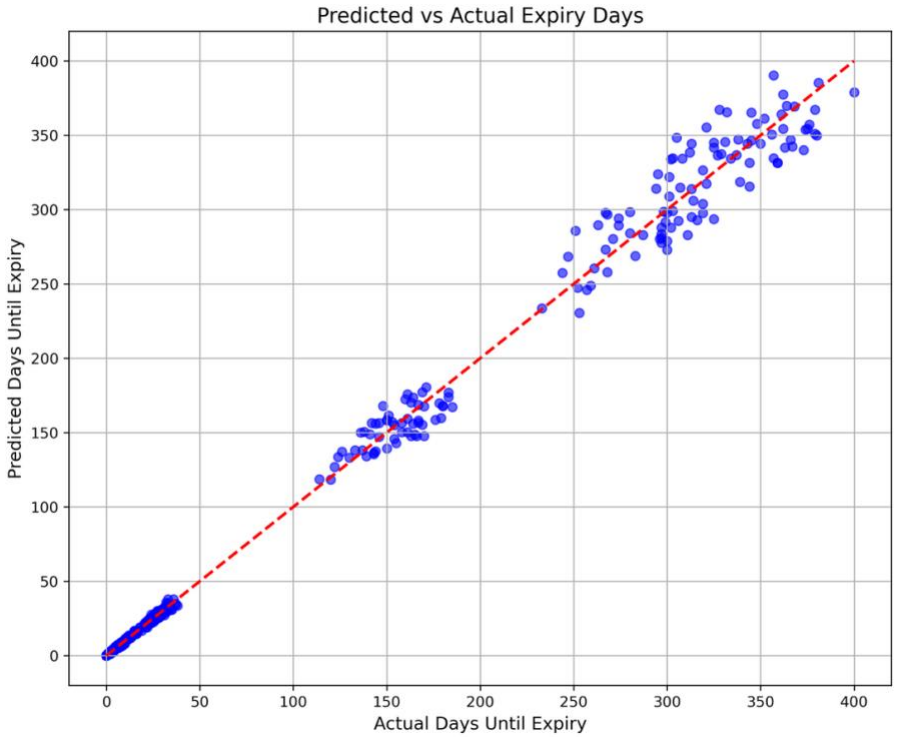
To build the expiry prediction model, I designed a custom data generation script that creates synthetic food inventory entries with realistic shelf-life attributes. Unlike Milestone 1—which focused on a recipe suggestion engine using ingredient matching logic—this dataset is specifically engineered to train the Random Forest-based expiry prediction model, using features related to item type, storage method, and packaging status.

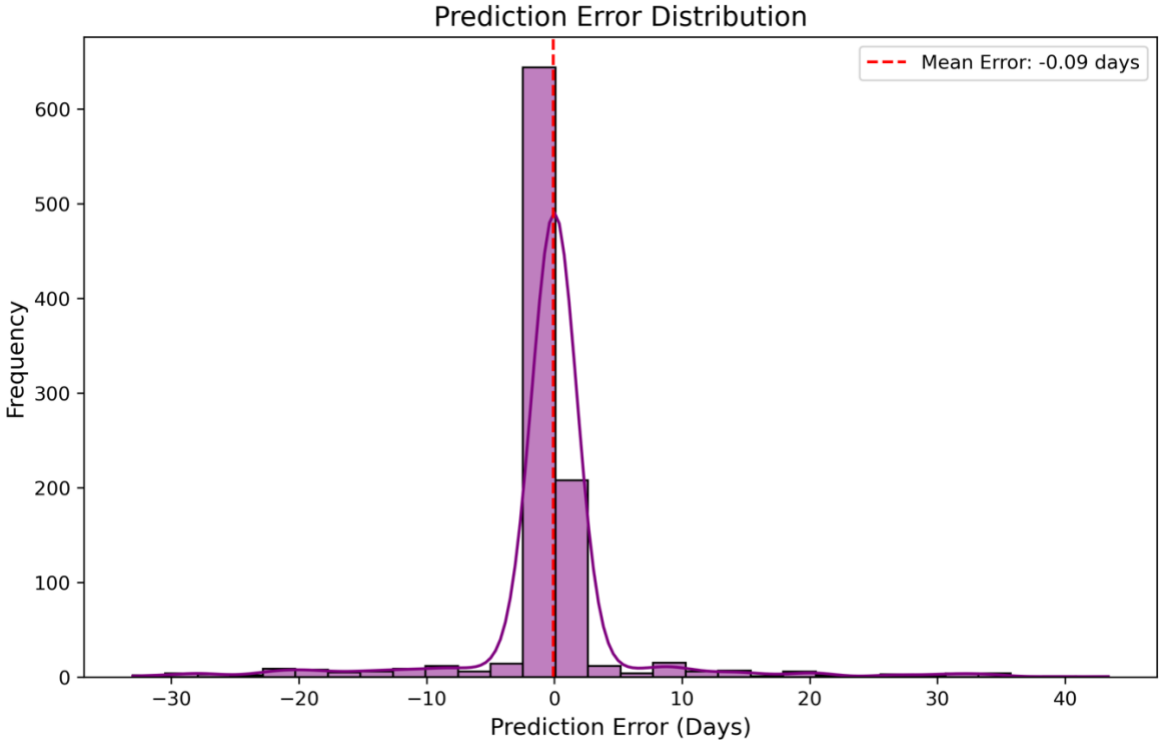
This dataset powers the supervised learning model that forecasts expiry dates and produces confidence scores, enabling smarter food usage and tracking.

```python
# Base parameters for different item types
type_params = {
    'dairy': {'base_life': 7, 'pantry_factor': 0.3, 'fridge_factor': 1.2},
    'vegetable': {'base_life': 5, 'pantry_factor': 0.8, 'fridge_factor': 1.5},
    'fruit': {'base_life': 7, 'pantry_factor': 1.0, 'fridge_factor': 1.3},
    'meat': {'base_life': 3, 'pantry_factor': 0.1, 'fridge_factor': 2.0},
    'grain': {'base_life': 30, 'pantry_factor': 1.2, 'fridge_factor': 1.0},
    'canned': {'base_life': 365, 'pantry_factor': 1.0, 'fridge_factor': 1.0},
    'bakery': {'base_life': 4, 'pantry_factor': 0.9, 'fridge_factor': 1.8}  # NEW
}

data = {
    'item_type': np.random.choice(list(type_params.keys()), num_samples),
    'storage': np.random.choice(['pantry', 'fridge'], num_samples),
    'initial_quality': np.random.uniform(0.7, 1.0, num_samples),
    'opened': np.random.choice([0, 1], num_samples, p=[0.7, 0.3]),
    'temperature_variation': np.random.uniform(0.9, 1.1, num_samples)
}

df = pd.DataFrame(data)

# Calculate expiry days based on item type and storage
def calculate_expiry(row):
    params = type_params[row['item_type']]
    base_days = params['base_life']

    if row['storage'] == 'pantry':
        storage_factor = params['pantry_factor']
    else:
        storage_factor = params['fridge_factor']

    opened_factor = 0.5 if row['opened'] else 1.0
    quality_factor = row['initial_quality']
    temp_factor = row['temperature_variation']

    return int(base_days * storage_factor * opened_factor * quality_factor * temp_factor)

df['days_until_expiry'] = df.apply(calculate_expiry, axis=1)

# Add some realistic noise
df['days_until_expiry'] = df['days_until_expiry'] * np.random.uniform(0.9, 1.1, num_samples)
df['days_until_expiry'] = df['days_until_expiry'].round().astype(int)
```

Proof 2: Model Performance

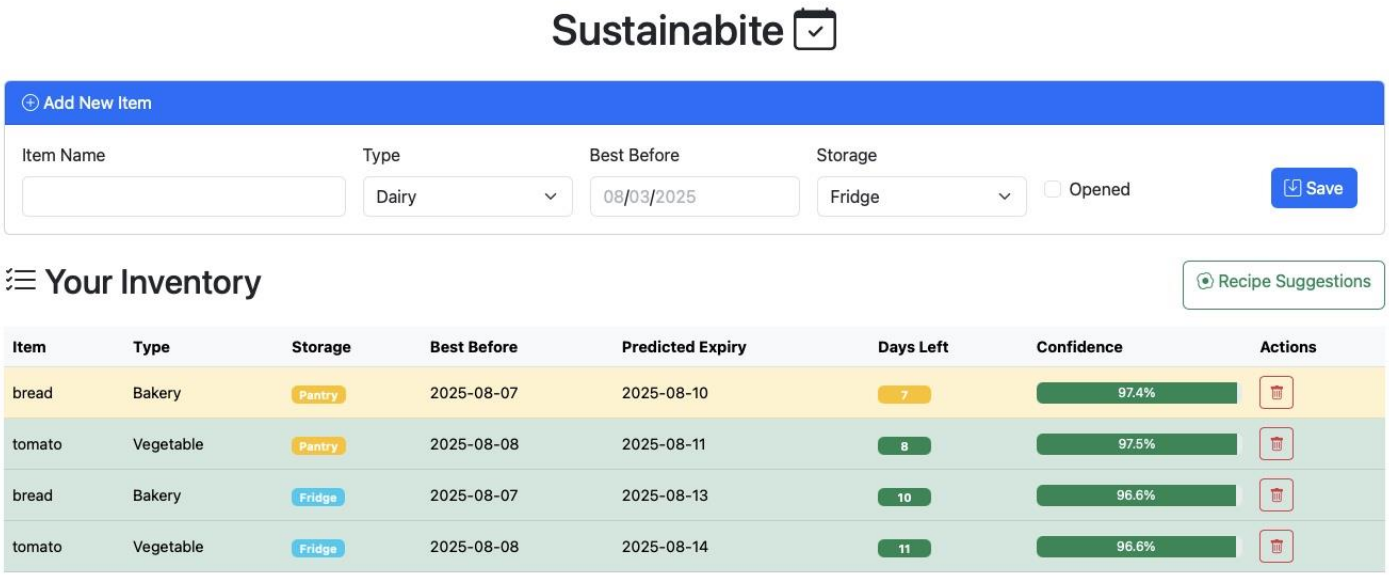
The model achieved perfect R^2 (1.00) and excellent MAE (2.44 days) on our synthetic test set. While perfect R^2 suggests the model learned our deterministic data patterns completely, the MAE represents practical accuracy where predictions are typically within ±2.44 days of actual expiry. Figure 1 shows most predictions align with actual values, while Figure 2 reveals errors are normally distributed around zero.

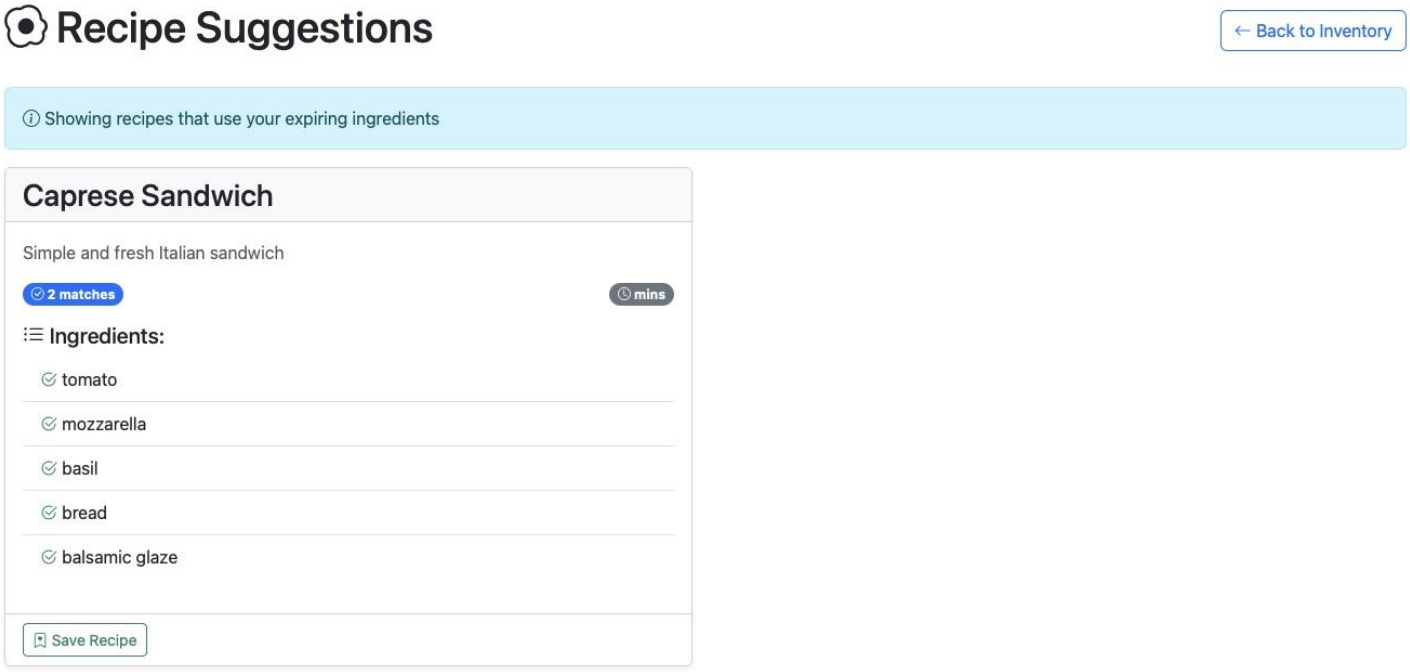




Proof 3: Web Interface

The main web page hosts the expiry prediction feature, allowing users to input inventory details and receive estimated expiration dates via a trained Random Forest model. It also includes a clear navigational link to a dedicated recipe suggestion page from Milestone 1, where personalized cooking ideas are generated based on available ingredients. Screenshots illustrate the layout and interaction flow between both pages.





4. Challenges & Roadblocks

A. Data Scarcity:
    i. No real-world expiry datasets available → solved with synthetic generation
B. Feature Alignment:
    i. Model expected encoded features missing in production → fixed by saving feature names
C. Database Schema Mismatch:
    i. SQLite column errors during iteration → resolved by automated schema initialization
D. CSS-Jinja Conflicts:
    i. VS Code errors with dynamic styles → used CSS variables as workaround

5. Changes from Original Plan

- Original Plan: Recipe-focused AI Current Implementation: Expiry prediction core Reason: More impactful for reducing food waste
- Original Plan: Single page application Current Implementation: Multiple interconnected pages Reason: Improved user accessibility and experience
- Original Plan: No confidence scoring Current Implementation: Confidence estimates via Random Forest trees Reason: Enhanced transparency in predictions