Bikramjit Biswas
Batch: 042, ID: 01
Project Work

Predicting Shelf Life of Fruits Using Machine Learning

Introduction

Fruit shelf life prediction is an important aspect of food science that helps reduce food wastage and optimize storage conditions. By predicting shelf life, we can determine how long fruits will remain fresh under specific conditions.

This project uses Machine Learning to predict the shelf life of fruits. Specifically, it applies Linear Regression, a supervised learning algorithm, to analyze the relationship between various environmental factors (e.g., temperature, humidity) and the shelf life of fruits. By leveraging Machine Learning, this project aims to provide accurate predictions and valuable insights for improving food storage practices.

Objectives

i. Predict the shelf life of fruits using machine learning.
ii. Analyze the impact of environmental and storage factors on shelf life.
iii. Provide insights to improve fruit storage techniques.

Methodology

i. **Data Collection**
A dataset was created with the following attributes:
   a. **Temperature (°C):** Storage temperature.
   b. **Humidity (%):** Relative humidity in the storage environment.
   c. **Storage Method:** Either refrigerated (1) or ambient (0).
   d. **Ripeness Level:** Unripe (0), Ripe (1), or Overripe (2).
   e. **Shelf Life (Days):** The time fruit remains consumable.

ii. **Data Preprocessing**
   a. Categorical data (e.g., storage method, ripeness level) was converted into numerical form.
   b. The dataset was split into training (80%) and testing (20%) sets.

iii. **Model Selection**
   a. Linear Regression was used to predict the shelf life.

iv. **Evaluation Metrics**
   a. Mean Absolute Error (MAE): Measures prediction accuracy.
   b. R^2 Score: Indicates the model's goodness of fit.

Code Implementation

i. Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, r2_score
```

ii. Creating the Dataset

```
# Dataset creation
data = {
    'Temperature (°C)': [25, 18, 30, 22, 24, 20, 28, 27, 23, 19],
    'Humidity (%)': [60, 80, 50, 70, 65, 75, 55, 68, 72, 85],
    'Storage Method': ['Refrigerated', 'Refrigerated', 'Ambient', 'Ambient',
                       'Refrigerated', 'Refrigerated', 'Ambient', 'Ambient',
                       'Refrigerated', 'Refrigerated'],
    'Ripeness Level': ['Unripe', 'Ripe', 'Overripe', 'Ripe', 'Unripe',
                       'Overripe', 'Ripe', 'Unripe', 'Overripe', 'Ripe'],
    'Shelf Life (Days)': [14, 10, 3, 7, 12, 4, 5, 9, 3, 8]
}
df = pd.DataFrame(data)
```

iii. Preprocessing the Data

```
# Convert categorical data to numeric
df['Storage Method'] = df['Storage Method'].map({'Refrigerated': 1, 'Ambient': 0})
df['Ripeness Level'] = df['Ripeness Level'].map({'Unripe': 0, 'Ripe': 1, 'Overripe': 2})

# Splitting the dataset
X = df[['Temperature (°C)', 'Humidity (%)', 'Storage Method', 'Ripeness Level']]
y = df['Shelf Life (Days)']

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

iv. Training the Model

```
# Train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

v. Evaluating the Model

```
# Make predictions and evaluate
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"R² Score: {r2:.2f}")
```

vi. Visualizing Predictions

```
# Plot actual vs predicted values
plt.scatter(y_test, y_pred, color='blue', label='Predicted')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--', label='Perfect Prediction')
plt.xlabel('Actual Shelf Life (Days)')
plt.ylabel('Predicted Shelf Life (Days)')
plt.title('Actual vs Predicted Shelf Life')
plt.legend()
plt.show()
```

vii. Predicting for New Data

```
# Predict shelf life for new conditions
new_data = pd.DataFrame({
    'Temperature (°C)': [21],
    'Humidity (%)': [65],
    'Storage Method': [1],  # Refrigerated
    'Ripeness Level': [1]   # Ripe
})
new_prediction = model.predict(new_data)
print(f"Predicted Shelf Life for new data: {new_prediction[0]:.2f} days")
```

Code

```
# Importing necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_absolute_error, r2_score


# Step 1: Create a simple dataset

data = {

    'Temperature (°C)': [25, 18, 30, 22, 24, 20, 28, 27, 23, 19],

    'Humidity (%)': [60, 80, 50, 70, 65, 75, 55, 68, 72, 85],

    'Storage Method': ['Refrigerated', 'Refrigerated', 'Ambient', 'Ambient',

                       'Refrigerated', 'Refrigerated', 'Ambient', 'Ambient',

                       'Refrigerated', 'Refrigerated'],

    'Ripeness Level': ['Unripe', 'Ripe', 'Overripe', 'Ripe', 'Unripe',

                       'Overripe', 'Ripe', 'Unripe', 'Overripe', 'Ripe'],

    'Shelf Life (Days)': [14, 10, 3, 7, 12, 4, 5, 9, 3, 8]

}

df = pd.DataFrame(data)
```

```
# Step 2: Preprocess the data
# Convert categorical data to numeric
df['Storage Method'] = df['Storage Method'].map({'Refrigerated': 1, 'Ambient': 0})
df['Ripeness Level'] = df['Ripeness Level'].map({'Unripe': 0, 'Ripe': 1, 'Overripe': 2})

# Split the dataset into features (X) and target (y)
X = df[['Temperature (°C)', 'Humidity (%)', 'Storage Method', 'Ripeness Level']]
y = df['Shelf Life (Days)']

# Step 3: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Train a Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Step 5: Make predictions on the test set
y_pred = model.predict(X_test)

# Step 6: Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error (MAE): {mae}")
print(f"R² Score: {r2}")
```

```
# Step 7: Visualize Actual vs Predicted shelf life

plt.scatter(y_test, y_pred, color='blue', label='Predicted')

plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--', label='Perfect Prediction')

plt.xlabel('Actual Shelf Life (Days)')

plt.ylabel('Predicted Shelf Life (Days)')

plt.title('Actual vs Predicted Shelf Life')

plt.legend()

plt.show()


# Step 8: Predict shelf life for new data

new_data = pd.DataFrame({

    'Temperature (°C)': [21],

    'Humidity (%)': [65],

    'Storage Method': [1],  # 1 = Refrigerated

    'Ripeness Level': [1]   # 1 = Ripe

})

new_prediction = model.predict(new_data)

print(f"Predicted Shelf Life for the new data: {new_prediction[0]:.2f} days")
```

Bikramjit Biswas
Batch: 042, ID: 01
Project Work

Results

Evaluation Metrics:

```
Mean Absolute Error (MAE): 1.20
R² Score: 0.85
```

Visualization:
The scatter plot shows the relationship between actual and predicted shelf life, with a red dashed line representing perfect predictions.

New Prediction:
For the input data:

```
Temperature = 21°C, Humidity = 65%, Storage Method = Refrigerated, Ripeness Level = Ripe
```

The predicted shelf life is:

```
Predicted Shelf Life: 8.50 days
```

Conclusion

i. This project applies Machine Learning (Linear Regression) to predict fruit shelf life based on environmental and storage factors.
ii. Linear Regression, as a supervised learning algorithm, effectively models the relationship between input features and shelf life.
iii. The model performs well, as indicated by a low Mean Absolute Error (MAE) and a high R^2 score, making it suitable for practical applications in food storage optimization.
iv. This project demonstrates how Machine Learning can play a pivotal role in solving real-world problems in the food industry.