1. Install Required Libraries

```
pip install pandas scikit-learn matplotlib seaborn
```

2. Python Code Implementation

```
# Importing Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Load Data
file_path = 'Food_Manufacturing_Quality_Attributes.xlsx'  # Update path if needed
df = pd.read_excel(file_path)

# Display First 5 Rows
print(df.head())

# Data Summary
print(df.describe())

# Check for Missing Values
print(df.isnull().sum())

# Correlation Heatmap
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Quality Attributes')
plt.show()

# Defect Distribution
sns.countplot(data=df, x='Defects(count)')
plt.title('Defect Counts Distribution')
plt.show()

# Feature vs Defect Boxplots
features = ['MoistureContent(%)', 'Weight(g)', 'Color(L*)', 'Texture(N)', 'Sweetness(Brix)']
for feature in features:
    plt.figure(figsize=(6,4))
    sns.boxplot(data=df, x='Defects(count)', y=feature)
    plt.title(f'{feature} vs Defects')
    plt.show()

# Prepare Data for Modeling
X = df[features]
y = df['Defects(count)']

# Simplify target to Binary Classification: Defect(1) or No Defect(0)
y_binary = y.apply(lambda x: 1 if x > 0 else 0)
```

```
# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y_binary, test_size=0.3, random_state=42)

# Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Feature Importance
feature_importances = pd.Series(model.feature_importances_, index=features).sort_values(ascending=False)
feature_importances.plot(kind='barh')
plt.title('Feature Importance in Defect Prediction')
plt.show()
```