Machine Learning

Model performance

To evaluate machine learning model performance on detecting or classifying rotten fruits and vegetables, you typically build a model that uses computer vision techniques (like image classification) and evaluate it using standard metrics. Here's a clear breakdown of what this involves:

🔍 Project Overview

- **Objective:** Automatically detect whether fruits or vegetables are fresh or rotten using AI/ML**.**

- **Input:** Images of fruits/vegetables (healthy and rotten).

- **Output:** A classification result – e.g., "Fresh" or "Rotten**".**

🛠️ Common Models Used

1. Convolutional Neural Networks (CNNs) – Best for image classification.

   ○ Examples: ResNet, VGG16, MobileNet, InceptionV3.

2. Transfer Learning Models – Pretrained models fine-tuned on your dataset.

3. YOLO/SSD – If object detection/localization is needed.

📊 Performance Metrics

To evaluate the model:

Metric	Description
Accuracy	% of correct predictions over total predictions.
Precision	TP / (TP + FP) – How many predicted rotten are actually rotten?
Recall	TP / (TP + FN) – How many rotten were correctly detected?
F1 Score	Harmonic mean of precision and recall.
Confusion Matrix	Shows TP, FP, FN, TN values for deeper insight.

In real-world cases, F1 Score and Recall matter most if your goal is to catch all rotten items.

Sample Results (Hypothetical)

Model	Accuracy	Precision	Recall	F1 Score
VGG16	90.5%	92.1%	88.3%	90.1%
ResNet50	93.2%	94.7%	91.6%	93.1%
MobileNetV2	89.0%	90.4%	87.2%	88.7%

📦 **Dataset Examples**

- Fruits-360 Dataset

- Kaggle Rotten Fruits Dataset

- Custom image dataset collected using mobile cameras or farms.

📉 Challenges

- Lighting and background variations

- Partial rotting (only part of the fruit is rotten)

- Different fruit types require different models or fine-tuning

---

📈 Performance Improvement Tips

- Use data augmentation: Flip, rotate, add noise.

- Apply segmentation before classification (if only part of the fruit is rotten).

- Increase dataset size and balance classes.

- Use early stopping and cross-validation during training.