# **Evaluation Metrics**

## **Introduction to Evaluation Metrics in Machine Learning**

* Goal: Evaluation metrics measure the effectiveness of a machine learning model in making accurate predictions.
* Importance: Selecting the right metric is essential for understanding a model’s strengths and limitations and for tuning it for better performance.
* Types: Metrics vary based on the type of problem: classification, regression, or clustering.

## **Evaluation Metrics for Classification Models**

Classification models predict categorical labels (e.g., spam vs. not spam, positive vs. negative sentiment).

### **Accuracy**

* Definition: The proportion of correctly predicted instances out of the total instances.



* Example: In a dataset of 100 emails, if 90 are classified correctly as spam/not spam, accuracy = 90%.
* Limitation: Not reliable for imbalanced datasets, as it ignores class distribution.

### **Precision, Recall, and F1-Score**

* Precision: The ratio of true positive predictions to all positive predictions (how many selected items are relevant).

A black text on a white background

Description automatically generated

* + Example: If a model identifies 30 emails as spam, but only 20 are actually spam, precision = 20/30 = 66.7%.
* Recall: The ratio of true positive predictions to all actual positives (how many relevant items were selected).

A black text on a white background

Description automatically generated

* + Example: If there are 50 actual spam emails, but the model only identifies 20 as spam, recall = 20/50 = 40%.
* F1-Score: The harmonic mean of precision and recall, balancing both metrics.

A black text with black letters

Description automatically generated with medium confidence

* + Example: If precision is 66.7% and recall is 40%, F1-score is 50.4%.

For classification metrics, we’ll use scikit-learn, a popular library for machine learning in Python.

### **Example Code for Accuracy, Precision, Recall, and F1-Score**

A screenshot of a computer program

Description automatically generated

### **Confusion Matrix**

* Definition: A table summarizing the number of true positives, false positives, true negatives, and false negatives.
* Layout:

A close-up of words

Description automatically generated

* Example:
  + In a binary classification task, a confusion matrix could look like:

A number with black numbers

Description automatically generated with medium confidence

* + This means there are 50 true positives, 10 false positives, 5 false negatives, and 100 true negatives.

A confusion matrix is a useful tool for visualizing classification performance.

### **Example Code for Confusion Matrix**

from sklearn.metrics import confusion\_matrix

import seaborn as sns

import matplotlib.pyplot as plt

# Generate confusion matrix

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

# Plot the confusion matrix

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=["Class 0", "Class 1"], yticklabels=["Class 0", "Class 1"])

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix")

plt.show()

### **ROC Curve and AUC (Area Under Curve)**

* ROC Curve (Receiver Operating Characteristic): Plots the true positive rate (sensitivity) against the false positive rate for different thresholds.
* AUC: The area under the ROC curve; higher AUC values indicate better model performance.
* Example: An AUC of 0.9 suggests the model has a 90% chance of distinguishing between positive and negative classes.

A screenshot of a computer program

Description automatically generated

## **Evaluation Metrics for Regression Models**

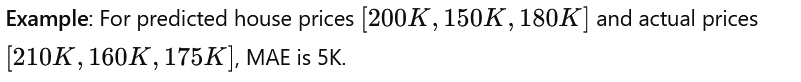
Regression models predict continuous values (e.g., predicting house prices or temperatures).

### **Mean Absolute Error (MAE)**

* Definition: The average absolute difference between predicted and actual values.

A math equation with numbers and symbols

Description automatically generated with medium confidence



### **Mean Squared Error (MSE)**

* Definition: The average squared difference between predicted and actual values, penalizing larger errors.

A number and symbol with numbers

Description automatically generated with medium confidence

* Example: If the MSE is 25K, the model has a squared average error of 25K.

### **Root Mean Squared Error (RMSE)**

* Definition: The square root of MSE, providing an error in the same units as the target.

A black text on a white background

Description automatically generated

* Example: RMSE of 5 means the model's predictions are on average 5 units off from the true values.

### **R-Squared (R²)**

* Definition: Measures how well the regression model explains the variance in the data, ranging from 0 to 1.

A number and a equal sign

Description automatically generated with medium confidence

* Example: An R² of 0.85 indicates that the model explains 85% of the variance in the data.

A screenshot of a computer program

Description automatically generated

## **Evaluation Metrics for Clustering Models**

Clustering models group similar instances without predefined labels.

### **Silhouette Score**

* Definition: Measures how similar each point is to its own cluster compared to other clusters, ranging from -1 to 1.
* Example: A score close to 1 indicates well-separated clusters, while a score near 0 indicates overlapping clusters.

### **Dunn Index**

* Definition: Ratio of the minimum inter-cluster distance to the maximum intra-cluster distance, assessing compactness and separation.
* Example: A higher Dunn Index indicates well-separated and compact clusters.

### **Inertia (Within-Cluster Sum of Squares)**

* Definition: Measures the compactness of clusters; lower inertia indicates tighter clusters.
* Example: Inertia = 300 in a K-Means model suggests better clustering than inertia = 500.

A screenshot of a computer program

Description automatically generated

# **Choosing the Right Metric**

* Classification: Use precision/recall for imbalanced data, accuracy for balanced data.
* Regression: RMSE or MAE for understanding prediction accuracy.
* Clustering: Silhouette score or Dunn Index for separation and compactness.
* AUC: Best for evaluating model performance on binary classification tasks with imbalanced data.