Supervised Machine Learning

- Supervised Machine Learning Algorithms
- To evaluate a supervised machine learning model, we need to be careful about:
  - Accuracy:
    - the success of a classifier in correctly classifying data
  - Speed:
    - the computational cost of a classifier.
  - Durability:
    - Being durable to noise or erroneous data
  - Scalability:
    - The ability of a classifier to remain effective even on large data sets.

Evaluation Metrics

- **Supervised Machine Learning**
  - **Classification**
    - Accuracy
    - Precision
    - Recall
    - F1 score
  - **Regression**
    - MSE
    - RMSE
    - MAE

Classification

- **Classification model**
  - Predicts class labels given input data
  - In Binary classification, there are two possible output classes
    - 0 or 1, True or False, Positive or Negative, Yes or No, etc.
  - Ex. Spam email detection
    - Spam or not spam

Classifier Performance

- In a classifier, the error rate is calculated on the test data
- To find the error rate:
  - Correct Predictions: if the class was guessed correctly
  - Error: if the class was guessed incorrectly
  - Error rate: the number of errors / the total number of test samples
- Classifier accuracy is based on correct predictions

Accuracy

- Accuracy: Ratio between the number of correct predictions and the total number of predictions

$$\text{accuracy} = \frac{\text{\# of correct predictions}}{\text{\# of total data points}}$$

- Ex: Suppose we have test data with 300 data points for predicting whether the patient has cancer:
  - Out of 300 test results, 100 tests are positive (has cancer), 200 tests are negative (doesn't have cancer)
  - Our model declares 80 out of 100 positives as positive correctly and 195 out of 200 negatives as negative correctly
  - Our model's accuracy: (80+195)/(100+200)=91.67%

Confusion Matrix

- Shows a more detailed breakdown of correct and incorrect classification of each class
- The notation of the confusion matrix is as follows:

Actual value

Predicted value	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

	Actually Positive	Actually Negative
Predicted Positive	80 (TP)	5 (FP)
Predicted Negative	20 (FN)	195 (TN)

Confusion Matrix (cont.)

- We can use the confusion matrix to calculate accuracy and error rate

	Actually Positive	Actually Negative
Predicted Positive	80 (TP)	5 (FP)
Predicted Negative	20 (FN)	195 (TN)

- Accuracy: $\frac{TP + TN}{TP + FP + FN + TN}$ = (80+195)/300 = 91.67%

Error Rate: $\frac{FP + FN}{TP + FP + FN + TN}$ = (5+20)/300 = 8.33%

Per-Class Accuracy

- Per-Class Accuracy: Average per-class accuracy
- In our previous example:
  - The model predicted 80 out of 100 positives as positive correctly (80% accuracy for positive class)
  - The model predicted 195 out of 200 negatives as negative correctly (97.5% accuracy for negative class)

    Per-Class Accuracy = (80+97.5)/2 = 88.75%
- Why important?
  - Class with more examples than others will dominate the statistics of accuracy, hence producing a distorted picture
  - Also, it can show different scenario when there are different numbers of examples per class

Precision

- Precision measures the proportion of true positives (i.e., correct positive predictions) among all positive predictions made by the model. It is calculated as:

$$\text{Precision:} \quad \frac{TP}{TP + FP}$$

- A high precision indicates that the model is making very few false positive predictions, meaning that when it predicts a positive result, it is very likely to be correct.
- Why important?
  - Precision helps to evaluate the model's ability to correctly identify true positive instances and avoid false positives.
  - For example, in medical diagnosis, precision is crucial because false positive predictions can lead to unnecessary medical interventions or treatments, causing harm to the patient and wasting resources. Therefore, a high precision model is desirable in such situations to minimize the number of false positives.

Recall

- Recall, on the other hand, measures the proportion of true positives among all actual positive instances in the data. It is calculated as:

$$\text{Recall:} \quad \frac{TP}{TP + FN}$$

- A high recall indicates that the model is capturing most of the positive instances in the data, meaning that when a positive instance is present, it is very likely to be correctly identified by the model.
- Why important?
  - Recall helps to evaluate the model's ability to capture all positive instances in the data and avoid false negatives.
  - For example, in spam email detection, recall is crucial because false negative predictions can result in spam messages being delivered to the user's inbox, causing annoyance and missing important emails. Therefore, a high recall model is desirable in such situations to minimize the number of false negatives.

Precision & Recall

- In our example, our precision and recall scores:

	Actually Positive	Actually Negative
Predicted Positive	80 (TP)	5 (FP)
Predicted Negative	20 (FN)	195 (TN)

$$\text{Precision: } \frac{TP}{TP + FP} = 80/(80+5) = 0.94$$

$$\text{Recall: } \frac{TP}{TP + FN} = 80/(80+20) = 0.80$$

F-measure (F1 score)

- F1 score is a harmonic mean of precision and recall, which is another commonly used metric for evaluating the performance of binary classification models. It provides a single score that combines both precision and recall, and is useful when both metrics are equally important.

$$\text{F1 score: } \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}}$$

- Like precision and recall, F1 score ranges from 0 to 1, with a higher score indicating better performance. A perfect classifier would have an F1 score of 1.
- F1 score is useful in situations where there is an imbalanced class distribution, where one class may have many more instances than the other. In such situations, accuracy can be a misleading metric, as it can be high simply by predicting the majority class.
- F1 score takes into account both precision and recall, which are less affected by imbalanced class distribution, and provides a more balanced evaluation of the model's performance.

F1 score (cont.)

- In our example, F1 score:

	Actually Positive	Actually Negative
Predicted Positive	80 (TP)	5 (FP)
Predicted Negative	20 (FN)	195 (TN)

$$\text{Precision: } \frac{TP}{TP + FP} = 80/(80+5) = 0.94$$

$$\text{Recall: } \frac{TP}{TP + FN} = 80/(80+20) = 0.80$$

$$\text{F1 score: } \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}}$$

$$= \frac{2*0.94*0.80}{0.94 + 0.80} = .864 \text{ F1 score}$$

Classifier Performance

- Assume we have the following two classifiers. Which one is better?

Classifier A	
TP=25	FP=25
FN=25	TN=25

Classifier B	
TP=50	FP=0
FN=25	TN=25

- Classifier A

Accuracy = (25 + 25) / (25 + 25 + 25 + 25)

Error Rate = (25 + 25) / (25 + 25 + 25 +25)

- Classifier B

Accuracy = (50 + 25) / (50 + 25 + 25 + 0)

Error Rate = (25) / (50 + 25 + 25 + 0)

Classifier Performance

- **Which classifier is better?**

Classifier A	
TP=45	FP=5
FN=5	TN=45

Classifier B	
TP=47	FP=7
FN=3	TN=43

Classifier C	
TP=40	FP=0
FN=10	TN=50

Each one of them has 90% accuracy.

For classifier A:

Precision = TP / (TP + FP) = 45 / (45 + 5) = 0.9

Recall = TP / (TP + FN) = 45 / (45 + 5) = 0.9

F1 score = 2 * (precision * recall) / (precision + recall) = 2 * (0.9 * 0.9) / (0.9 + 0.9) = 0.9

For classifier B:

Precision = TP / (TP + FP) = 47 / (47 + 7) = 0.87

Recall = TP / (TP + FN) = 47 / (47 + 3) = 0.94

F1 score = 2 * (precision * recall) / (precision + recall) = 2 * (0.87 * 0.94) / (0.87 + 0.94)= 0.905

For classifier C:

Precision = TP / (TP + FP) = 40 / (40 + 0) = 1.0

Recall = TP / (TP + FN) = 40 / (40 + 10) = 0.8

F1 score = 2 * (precision * recall) / (precision + recall) = 2 * (1.0 * 0.8) / (1.0 + 0.8) = 0.89

Validation Methods

- In machine learning, validation refers to the process of evaluating the performance of a trained model on a dataset that is separate from the dataset used for training.
- The goal of validation is to estimate the model's ability to generalize to new, unseen data.

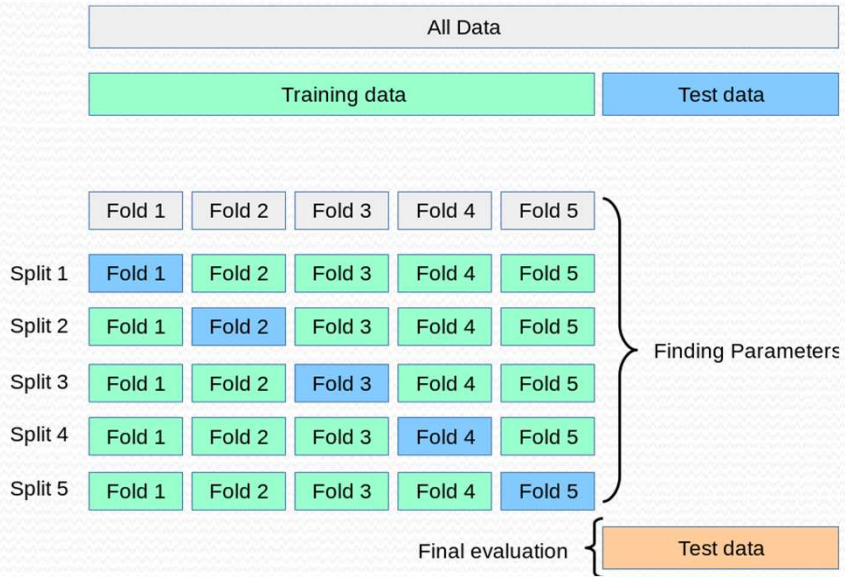
Holdout validation:

- The dataset is split into two parts: a training set and a test (validation) set.
- The model is trained on the training set and evaluated on the test set.
- The split between training and validation sets is typically around 70/30 or 80/20.

Cross-Validation

K-Fold Cross-Validation:

1. The dataset is split into k-folds, where k is a pre-defined number.
2. The model is trained on k-1 folds and evaluated on the remaining fold.
3. This process is repeated k times, with each fold used once as a validation set.
4. The final performance of the model is the average performance across all k folds.

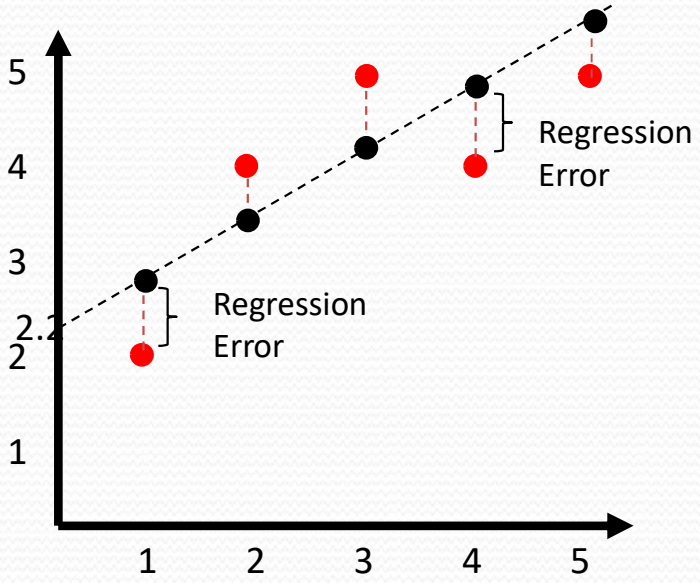


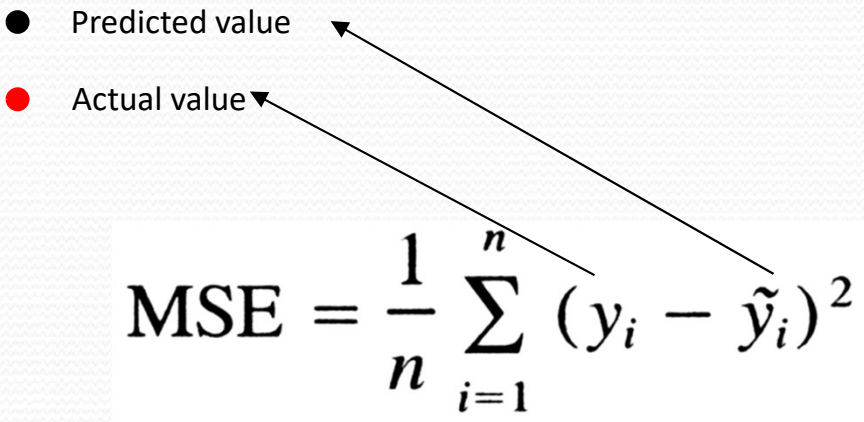
Regression Performance

- Regression is a supervised learning task that involves predicting a continuous numerical output variable based on one or more input variables.
- We cannot use accuracy, F1 score to evaluate regression models
- To evaluate the performance of a regression model, several evaluation metrics are used. The most common regression evaluation techniques:
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)

Mean Square Error (MSE)

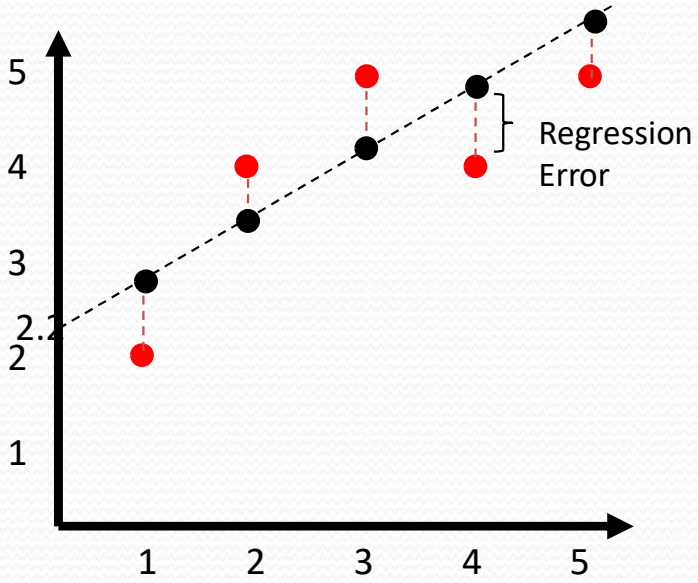
- **Mean Squared Error (MSE):** This is the most commonly used regression evaluation metric. It measures the average squared difference between the predicted and actual values of the target variable. The lower the MSE value, the better the model.





A lower MSE indicates a better fit of the model to the data, with a value of 0 indicating a perfect fit.

Mean Square Error (MSE)



$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

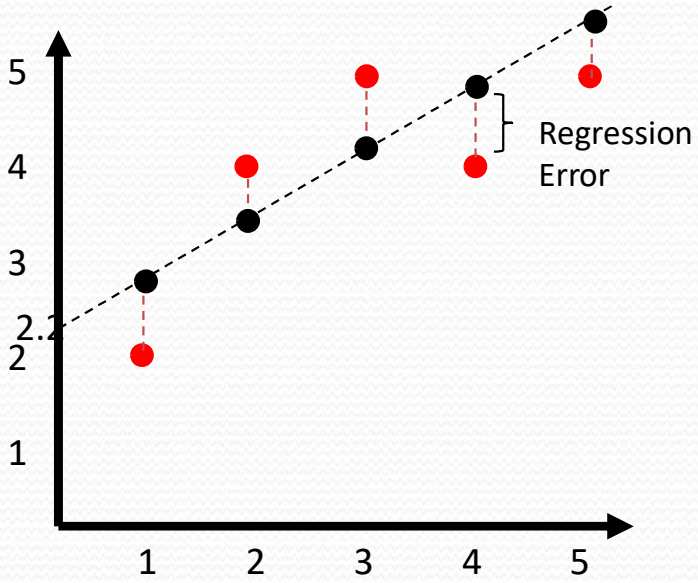
x	Actual Value (y)	Predicted value ($\hat{y}$)	$y-\hat{y}$	$(y-\hat{y})^2$
1	2	2.8	-0.8	0.64
2	4	3.4	0.6	0.36
3	5	4	1	1
4	4	4.6	-0.6	0.36
5	5	5.2	-0.2	0.04

sum= 2.4

MSE= 2.4/5 = 0.48

Root Mean Square Error (RMSE)

- **Root Mean Squared Error (RMSE):** This is the square root of the MSE. It measures the average difference between the predicted and actual values of the target variable. The lower the RMSE value, the better the model.



$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2}$$

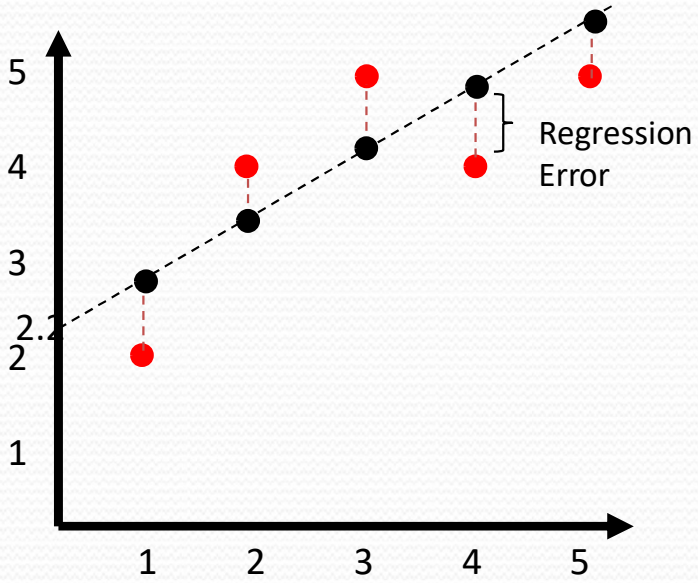
x	Actual Value (y)	Predicted value ($\hat{y}$)	$y-\hat{y}$	$(y-\hat{y})^2$
1	2	2.8	-0.8	0.64
2	4	3.4	0.6	0.36
3	5	4	1	1
4	4	4.6	-0.6	0.36
5	5	5.2	-0.2	0.04

sum= 2.4

RMSE= $\sqrt{(2.4/5)}$ = 0.69

Mean Absolute Error (MAE)

- **Mean Absolute Error (MAE):** This measures the average absolute difference between the predicted and actual values of the target variable. It is less sensitive to outliers compared to MSE. The lower the MAE value, the better the model.



$$\text{RMSE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \tilde{y}_i|$$

x	Actual Value (y)	Predicted value (ŷ)	|y-ŷ|
1	2	2.8	0.8
2	4	3.4	0.6
3	5	4	1
4	4	4.6	0.6
5	5	5.2	0.2

sum = 3.2

MAE = 3.2/5 = 1.6