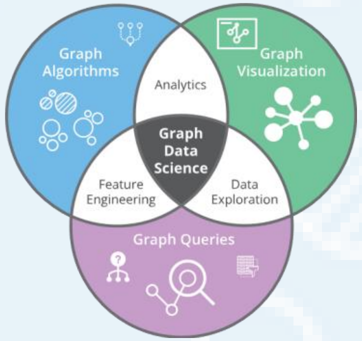


Habib University
shaping futures

CS343 Graph Data Science

Model Evaluation

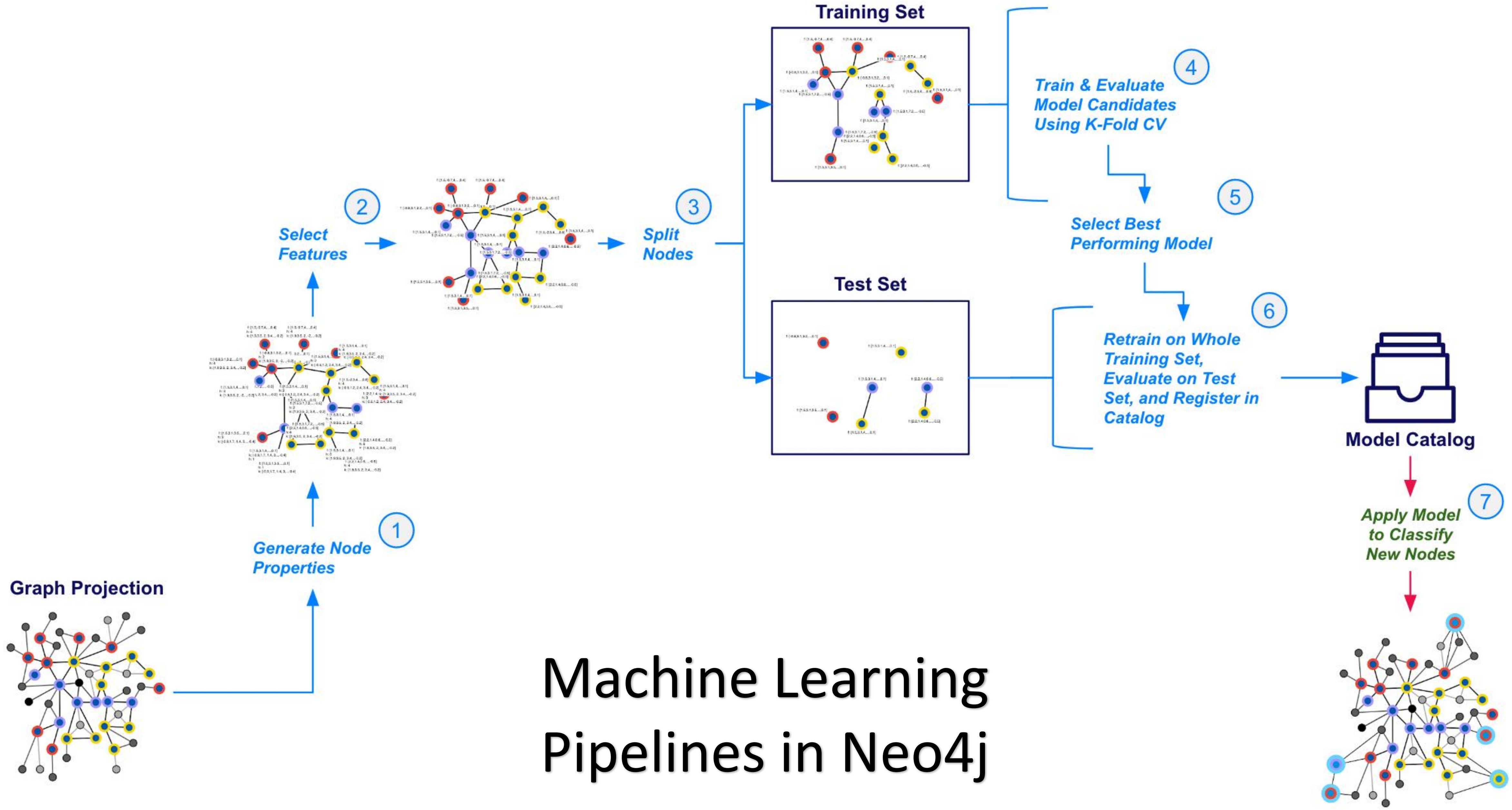
Muhammad Qasim Pasta
qasim.pasta@sse.habib.edu.pk

Slides are intended to be filled during the lectures. Certain details are intentionally omitted for in-class discussions. These slides are not meant to be used as reading material.

Workflow of Machine Learning

- **Problem Definition:** What are we trying to predict?
- **Data Collection & Preprocessing:** Creating projections.
- **Feature Engineering:** Defining embedding, calculating centralities etc.
- **Model Selection:** Choosing an appropriate algorithm (e.g., knn, Random Forest)
- **Splitting Data into Train/Test Sets:** To evaluate performance.
- **Training the Model:** Learning from the training data.
- **Evaluating Performance:** How well the model is learning.
- **Hyperparameter Tuning & Optimization:** Improving the model.
- **Final Deployment:** Using the model in real-world applications.

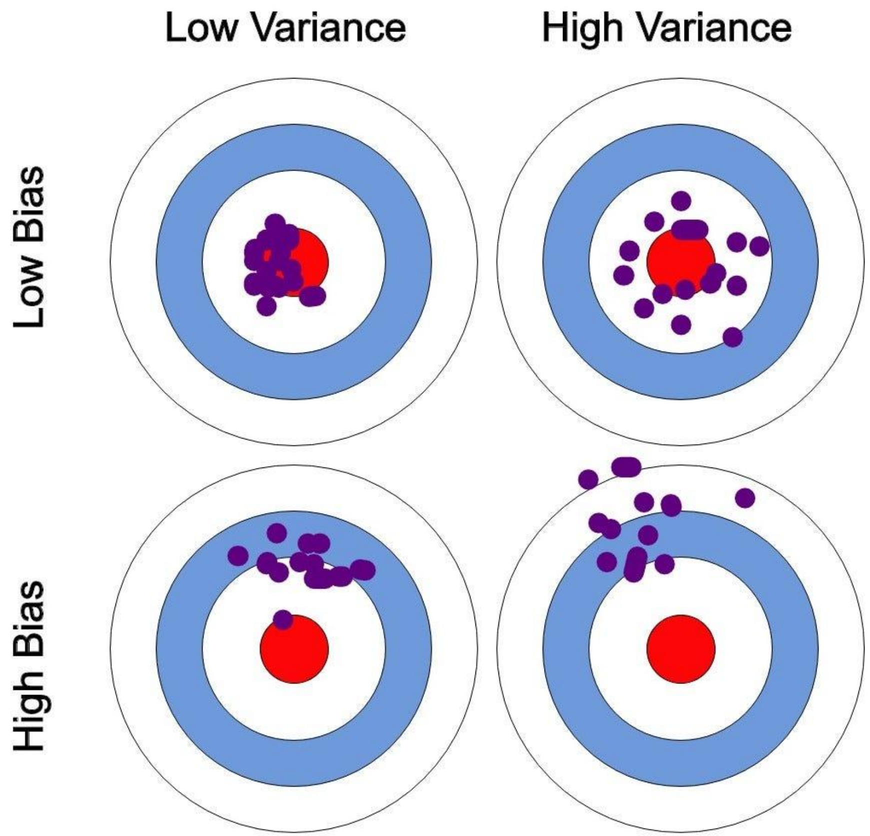
Machine Learning Pipelines in Neo4j



Why splitting data?

- A student memorizing past exam questions vs. understanding concepts.
- What if model memorizes patterns instead of learning general rules
- Ensure the model generalizes to new, unseen data.

- Bias: inability to capture the true relationship
- Variance: the difference between training and testing



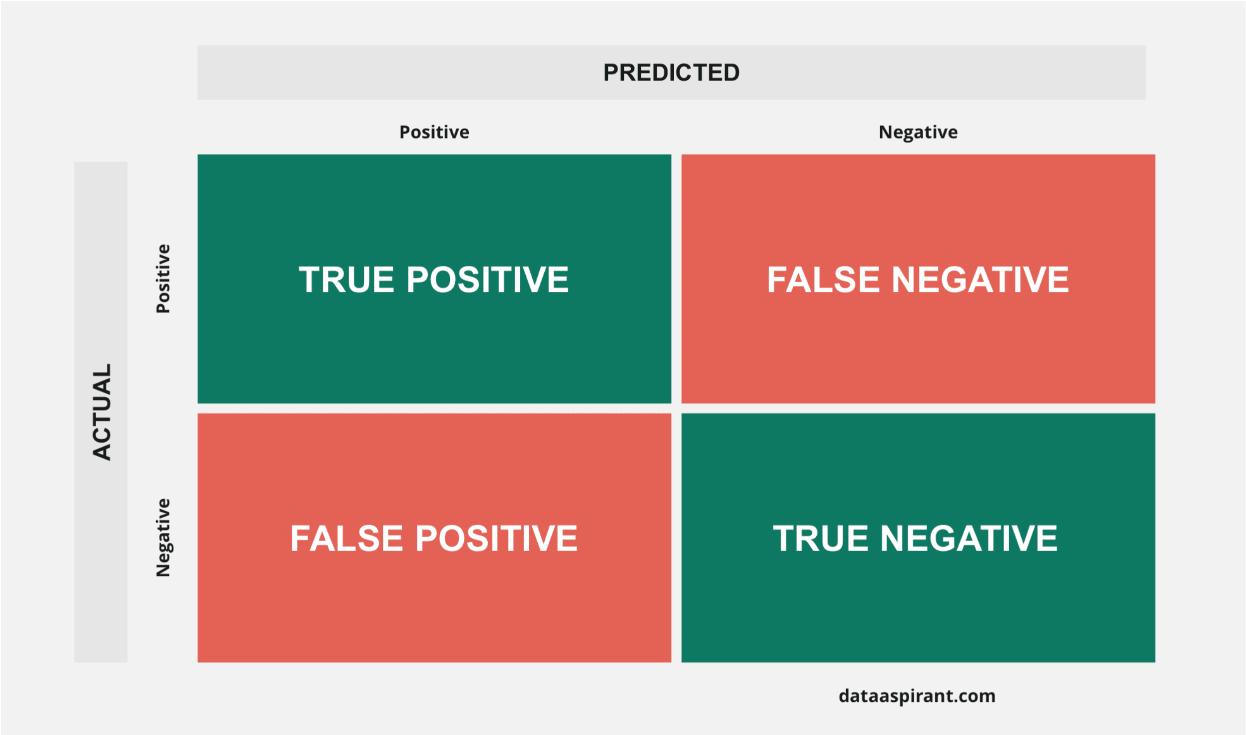
Model Evaluation

- A machine learning model must generalize well to unseen data.
- Without proper evaluation, we risk overfitting (too specific to training data) or underfitting (too simple to learn patterns).

- Helps assess the performance of a model before deployment.
- Avoids overfitting or underfitting.
- Ensures generalization to unseen data.

Confusion Matrix

- Summarizes the performance of a classification model.
- True Positives (TP):
  – Correctly predicted positive cases.
- False Positives (FP):
  – Incorrectly predicted as positive
- False Negatives (FN):
  – Incorrectly predicted as negative
- True Negatives (TN):
  – Correctly predicted negative cases.



Confusion Matrix: Example

Spam Detection:

- TP: Correctly classified spam emails.
- FP: Normal emails incorrectly classified as spam.
- FN: Spam emails classified as normal.
- TN: Normal emails correctly classified.

n=200	Predicted to be: SPAM	Predicted to be: NOT SPAM
Actually is : SPAM	120	30
Actually is : NOT SPAM	10	40

Accuracy

- Measures overall correct predictions.

- Formula: $\frac{(TP + TN)}{(TP + TN + FP + FN)}$

- Calculate:

n=200	Predicted to be: SPAM	Predicted to be: NOT SPAM
Actually is : SPAM	120	30
Actually is : NOT SPAM	10	40

Sensitivity

- aka Recall or True Positive Rate
- Measures how well the model captures actual positives.
- Important when false negatives are costly (e.g., dise

- Formula: $\frac{TP}{(TP + FN)}$

- 120 / (120+10) =0.92

n=200	Predicted to be: SPAM	Predicted to be: NOT SPAM
Actually is : SPAM	120	30
Actually is : NOT SPAM	10	40

Specificity

- Measures how well the model identifies negatives.
- Formula = TN / (TN + FP)
- Calculate:

n=200	Predicted to be: SPAM	Predicted to be: NOT SPAM
Actually is : SPAM	120	30
Actually is : NOT SPAM	10	40

Precision

- Measures how many precited positive are actually correct
- Among the positive predictions made, how many were actually correct?
- When false positives are costly (e.g., predicting someone has a disease when they don’t).
- Formula: $\frac{(TP\ )}{(TP + FP)}$
- Calculate:

n=200	Predicted to be: SPAM	Predicted to be: NOT SPAM
Actually is : SPAM	120	30
Actually is : NOT SPAM	10	40

F1 Score

- A balance between precision and recall.
- When it matters: When we need to weigh both false equally.
- Formula: $2 * \frac{(Precision * Recall)}{(Precision + Recall)}$
  - Precision: $\frac{(TP)}{(TP + FP)}$
  - Recall: $\frac{TP}{(TP + FN)}$
- Calculate:

n=200	Predicted to be: SPAM	Predicted to be: NOT SPAM
Actually is : SPAM	120	30
Actually is : NOT SPAM	10	40

Comparison

Metric	When to Use	Strengths	Weaknesses
Accuracy	Balanced datasets where false positives & false negatives matter equally.	Simple to interpret.	Misleading in imbalanced datasets.
Sensitivity (Recall)	When false negatives are costly (e.g., medical diagnosis, fraud detection).	Ensures important cases are not missed.	Can be high even if there are many false positives.
Specificity	When false positives are costly (e.g., spam detection, legal cases).	Good for ruling out false alarms.	May ignore false negatives.
Precision	When false positives are costly (e.g., recommending medical treatment, sending marketing emails).	Ensures reliable positive predictions.	Can be low if there are many false negatives.
F1-score	When both false positives & false negatives matter.	Balances both recall & precision.	Doesn't account for true negatives.

Cross-Validation

- A resampling technique used to assess model performance
- Reduces variance compared to a single train-test split.
- Ensures the model is not biased toward a particular subset of data.
- Uses the entire dataset for training and testing, reducing bias and variance.

- **k-Fold Cross-Validation**
  - dataset is divided into k parts (folds).
  - Model is trained on k-1 folds and tested on the remaining fold.
  - The process repeats k times, averaging the scores.

Reference:

- https://neo4j.com/docs/graph-data-science/current/machine-learning/node-property-prediction/noderegression-pipelines/config/