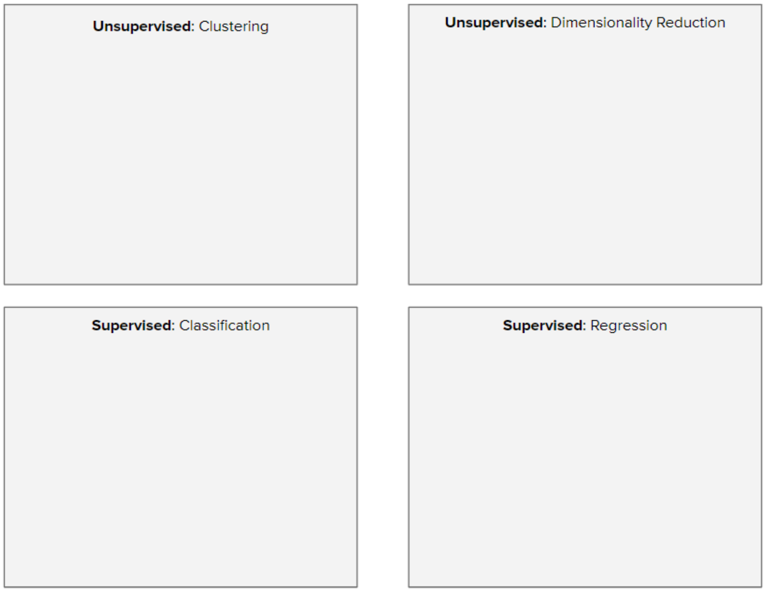
Model Selection

1. Below is a list of qualities and properties to consider when choosing an algorithm for a machine learning workflow. These can describe either what the algorithm does or its general pros and cons. A table is provided below, which has been filled in, though some entries may contain errors. To interpret the table: a ✓ indicates that the algorithm handles the property well, while an × indicates that the algorithm does not handle the property effectively. Your goal is to spend some time looking through sklearn documentation, slide decks, or other resources to verify the table. You can start by picking an algorithm at random, and verifying that row of the table. Any errors can be circled and explained below the table.

    1. Supervised: predicts an output for a target (ground truth)
    2. Unsupervised: finds patterns or structure in data with no target
    3. Regression: predicts a continuous numeric output
    4. Classification: predicts discrete class labels
    5. Clustering: groups data points based on similarity (does not need labels)
    6. Dimensionality Reduction: transforms data into a lower-dimensional representation
    7. Probabilistic: explicitly models probability distributions or outputs class probabilities
    8. Linear: assumes a linear relationship between input features and output
    9. Non-linear: can model complex, non-linear relationships
    10. Parametric: algorithm has a finite number of parameters
    11. Ensemble: combines multiple models (reduces variance)
    12. Regularization: constrains model complexity by limiting number of parameters used (reduces overfitting)
    13. Interpretable: model behavior and predictions can be easily explained or understood.
    14. Large Data: scales well computationally to large datasets.
    15. Outlier Robust: relatively insensitive to extreme values or noisy observations.
    16. Missing Data: handles missing values natively without explicit imputation.
    17. High Variance: flexible and prone to overfitting (unless constrained or regularized)
    18. High Bias: too simple relative to the data complexity and prone to underfitting.

Algorithm	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Hierarchical	×	✓			✓									×	×		✓	
DBSCAN	×	✓			✓									×	✓			
k-means	×	✓			✓			✓		✓				✓	×			
k-modes	×	✓			✓			✓		✓				✓				
k-medians	×	✓			✓			✓		✓				✓	✓			
GMM	×	✓			✓		✓		✓	✓		✓		×	×		✓	
Decision Tree	✓	×	✓	✓					✓			✓	✓	×		✓	✓	
Linear Regression	✓	×	✓					✓		✓		✓	✓	✓	×		×	✓
Logistic Regression	✓	×		✓			✓	✓		✓		✓	✓	✓	×		×	✓
Linear SVM	✓	×		✓				✓		✓		✓		✓	×		×	✓
Kernel SVM	✓	×		✓					✓	×		✓		×	×		✓	×
Naive Bayes	✓	×		✓			✓	✓		✓			✓	✓			×	✓
Random Forest	✓	×	✓	✓					✓		✓			✓	✓		×	✓
Neural Network	✓	×	✓	✓			✓		✓	×		✓		✓	×		✓	×
XGBoost	✓	×	✓	✓			✓		✓		✓	✓		✓	✓		×	✓
PCA	×	✓				✓		✓		✓		✓	✓	✓				
SVD	×	✓				✓		✓		✓		✓	✓	✓				

2. After filling in the table, write down where that algorithm lives. Note some supervised learning algorithms can be both classification and regression.



Unsupervised: Clustering	**Unsupervised**: Dimensionality Reduction
Supervised: Classification	**Supervised**: Regression

3. Choose one of the supervised boxes, and then make a flow chart below on how you would decide on which algorithm to choose. You will find the table helpful in coming up with questions for the chart (reference the unsupervised examples if needed).

Model Workflow

4. List out the five steps of an ML workflow:

   -
   -
   -
   -
   -

   Let's say you want to run three models on the iris dataset that uses petal width and length to predict species type.

5. Is this supervised or unsupervised? Is this classification, regression, or clustering?

6. Pick three algorithms that you would like to try on this dataset:

   -
   -
   -

7. If we wanted to do cross-fold validation, when would we do this?

8. If we wanted to tune each model, when would we do this?

9. For each algorithm you've selected, write down at least one hyperparameter you could tune if you were implementing this using sklearn. For this question, you can look back over slide decks or use sklearn documentation.

   -
   -
   -