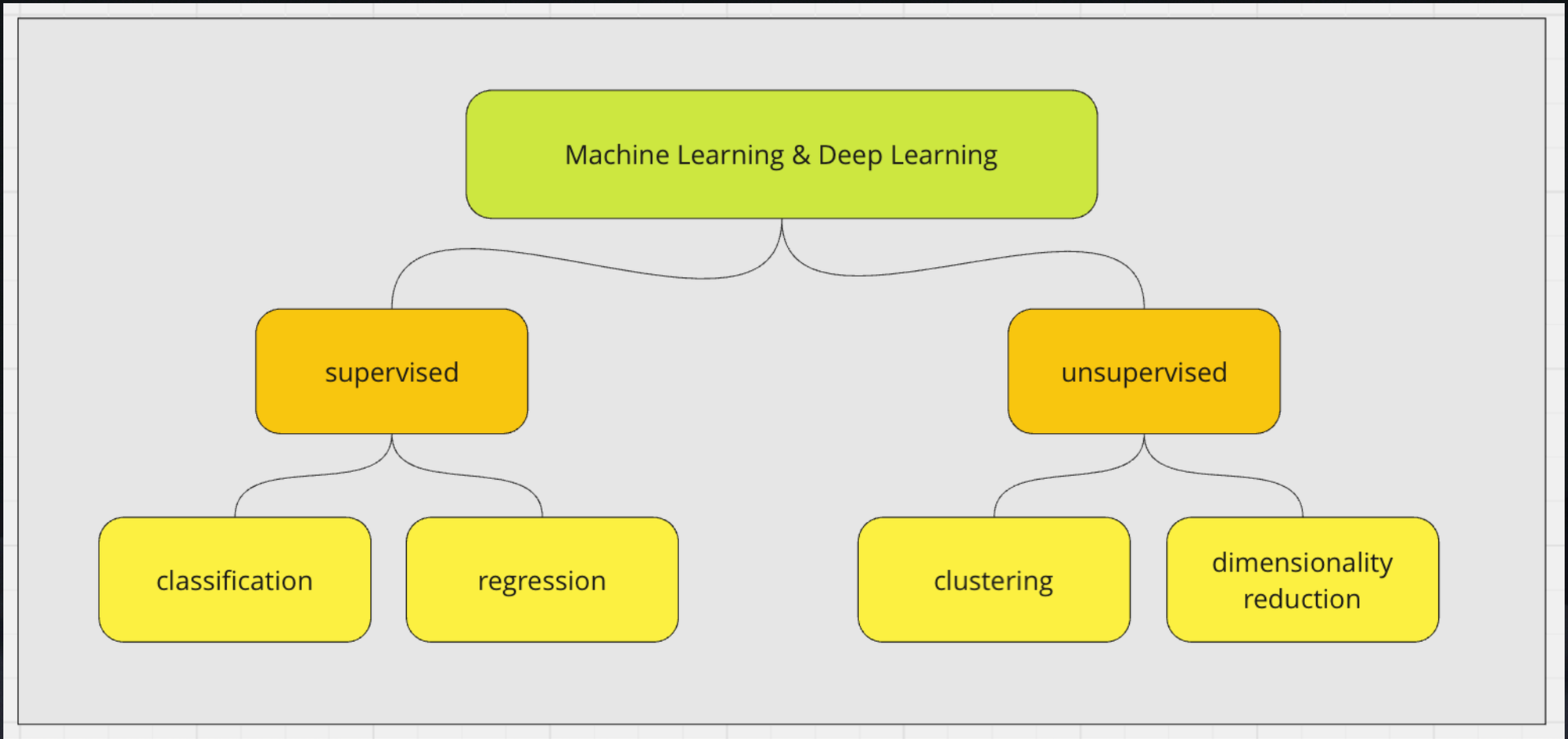
Fundamentals of Machine Learning

Kevin E Dean

OverView

- Introduction
- Supervised Learning (specific output)
  - Linear Regression, Ridge / Lasso Regression, K-Nearest Neighbor Algorithm, Decision Tree (random forest, adaboost, xgboost)
- Unsupervised Learning (No specific output)
  - K-Means Clustering, Silhouette Clustering, Density-Based Scan

Introduction



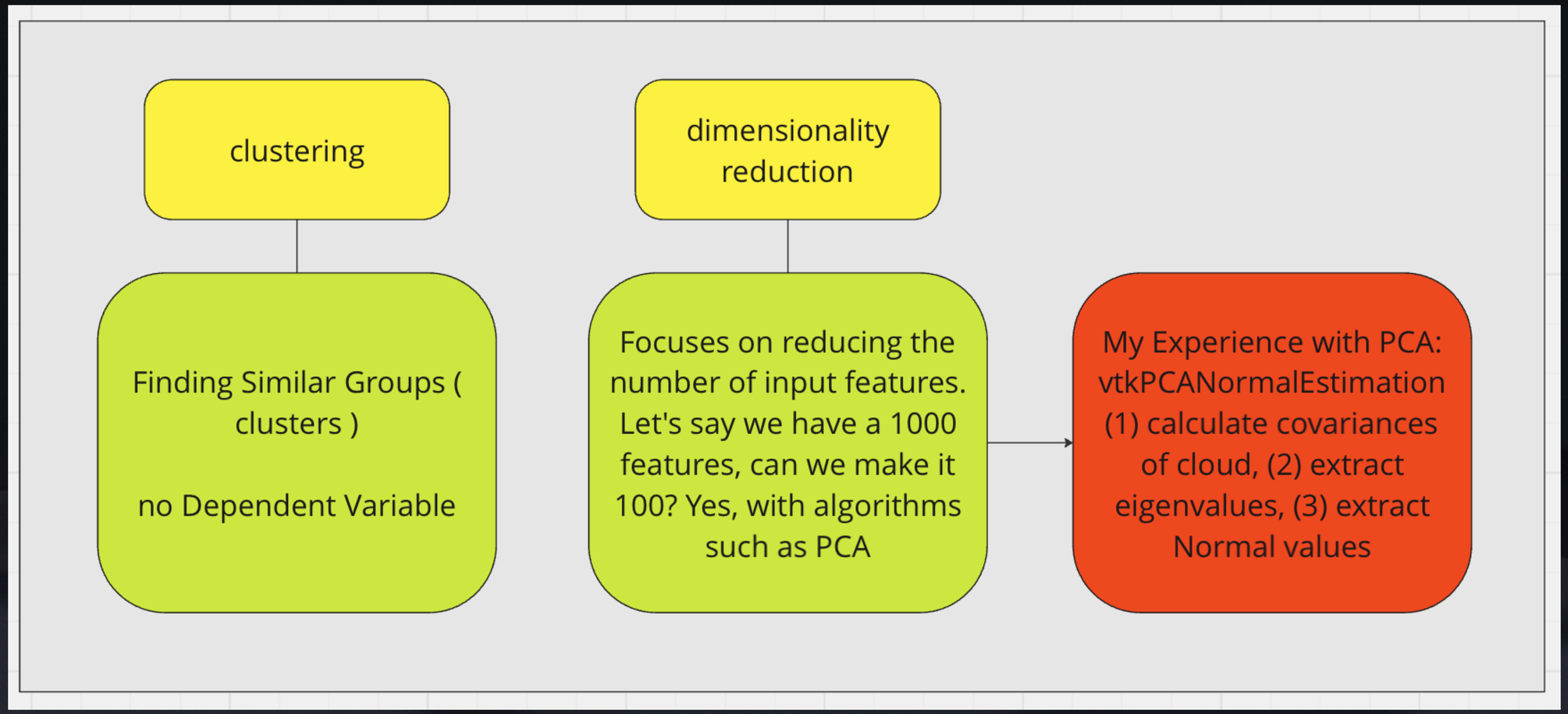
Introduction

What problems do Supervised learning algorithms solve?

- Regression
  - Output has a continuous variable
- Classification
  - Output has a fixed number of categories (binary or multi class)

Introduction

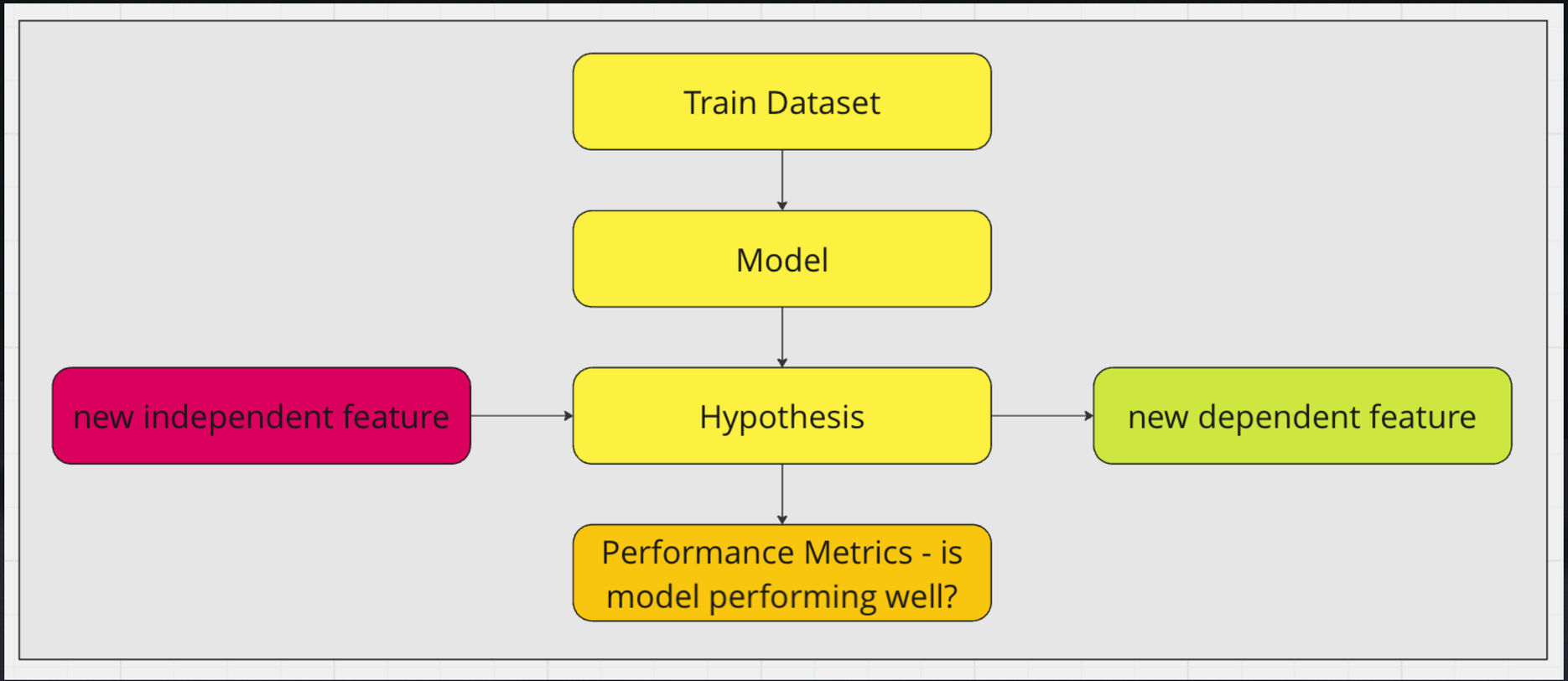
What problems do Unsupervised learning algorithms solve?



Supervised Learning

Linear Regression

- We try to create a model with a training dataset and essentially perform hypothesis testing.



Supervised Learning

Linear Regression

- Trying to find the best fit line to the data, if you think about it in [x, y], it is the predicted point with respect to y.
- Y is a linear function of x
  - Y = mx + c, however there are several notations for this representation of a line
- Create the best fit line when / how?
  - Calculate the distance between the truth (data points I have) and predicted is how we measure the best fit.

Supervised Learning

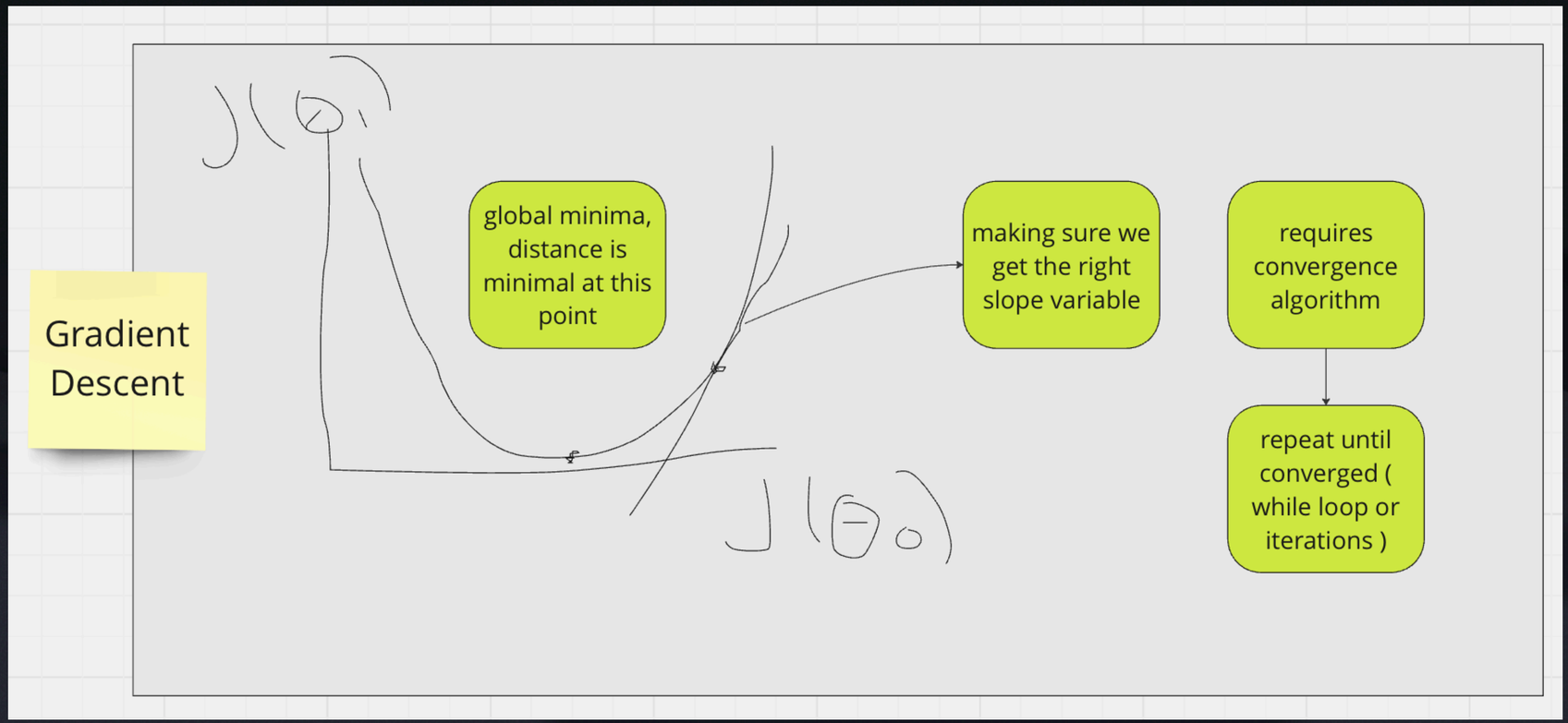
Linear Regression

- Imagine we have 2 Dimensional data, and we try to fit several lines to this data and choose the best fit.
  - Through many iterations
    - Start at a point
    - Go towards finding the best fit line with cost function (squared error function)
      - Distance formula explaining the relationship between the predicted and the truth
      - Distance during the summation should be minimal

Supervised Learning

Linear Regression

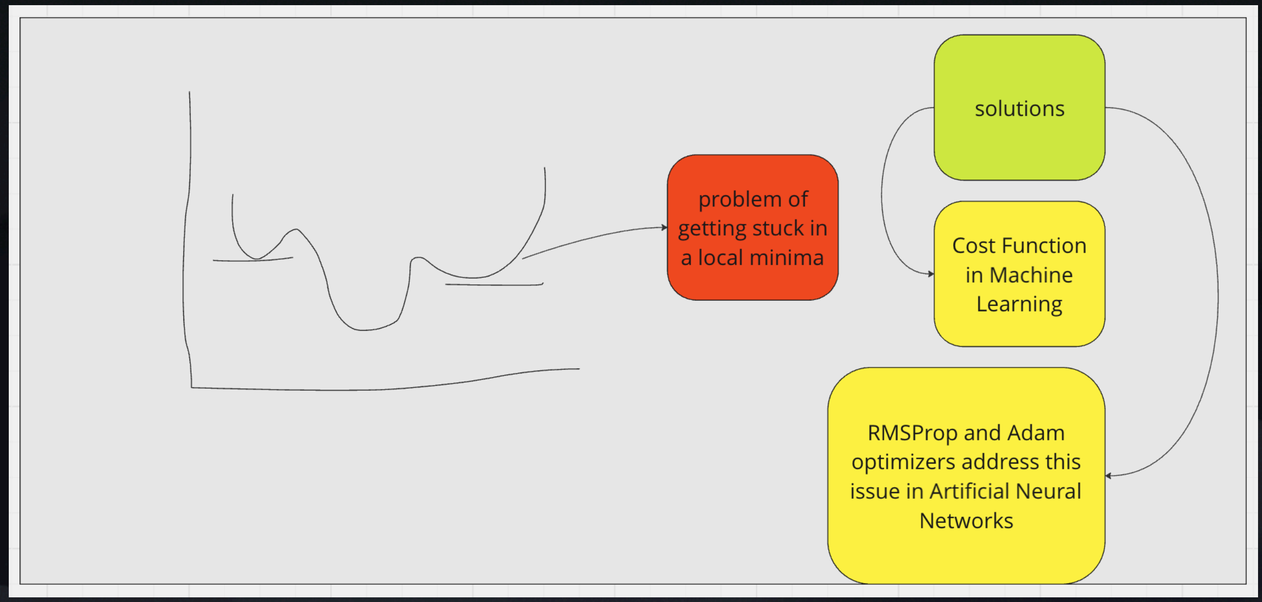
- Purpose is to take the derivative of the cost function to find the slope to utilize and find the global minima (will return the best coefficients for the algorithm). - AKA Gradient Descent.



Supervised Learning

Linear Regression

- At what speed will we adjust the weight (referred to as the Learning Rate)
- Local Minima Problem



Supervised Learning

Linear Regression - Performance Metrics

- R squared and adjusted R squared
  - This is how we verify our model with respect to “linear” regression
  - Refers to how well a model performs based upon it’s sum of residuals (difference between predicted and actual) and sum of total (difference between data and the mean)
  - If adding another feature (that has NO correlation to the dependent feature), and R squared value increases, we can look at adjusted R squared which takes into account the number of predictors and the number of data points.

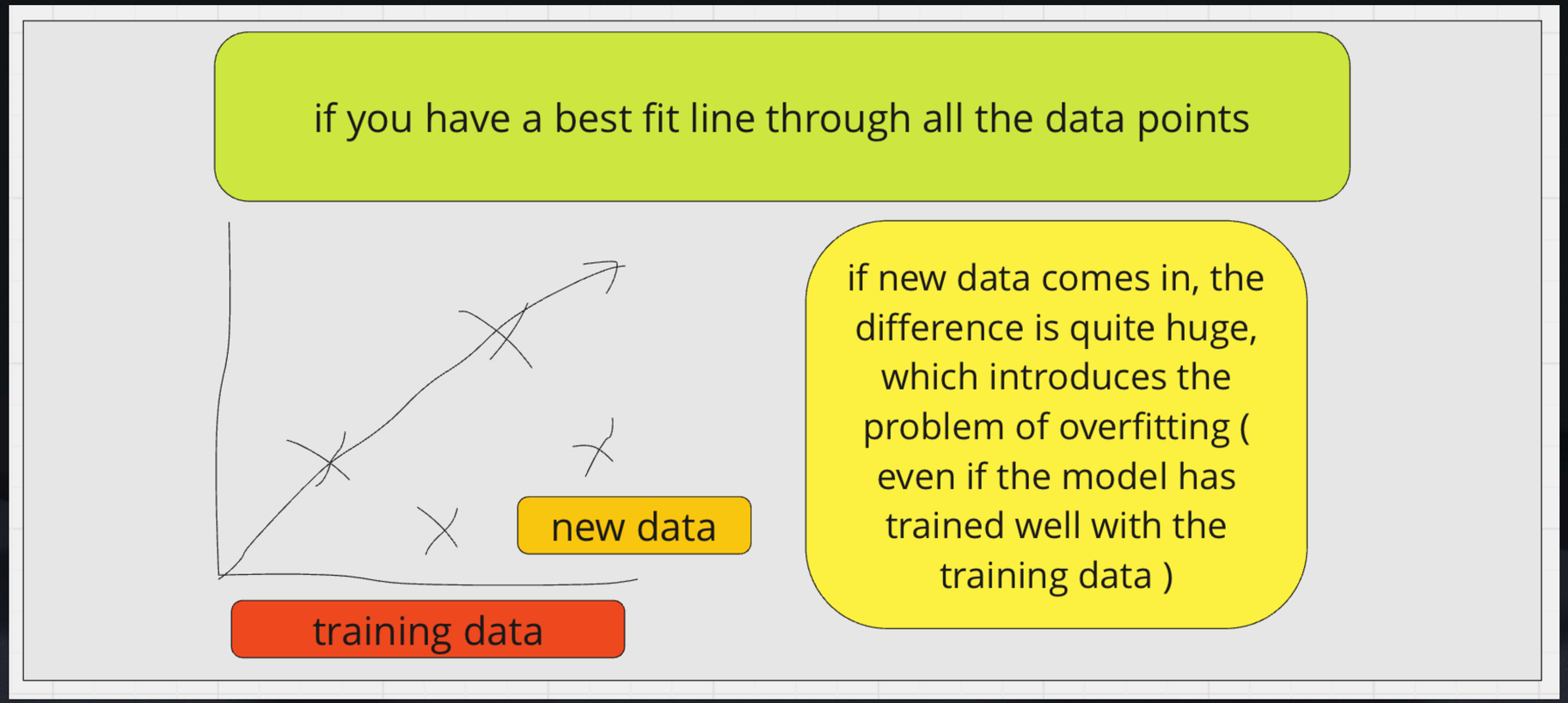
Supervised Learning

Non Linear Regression Validate Performance

- Step 1: Determine whether the regression line fits your data. ...
- Step 2: Examine the relationship between the predictors and the response. ...
- Step 3: Determine how well the model fits your data. ...
- Step 4: Determine whether your model meets the assumptions of the analysis.

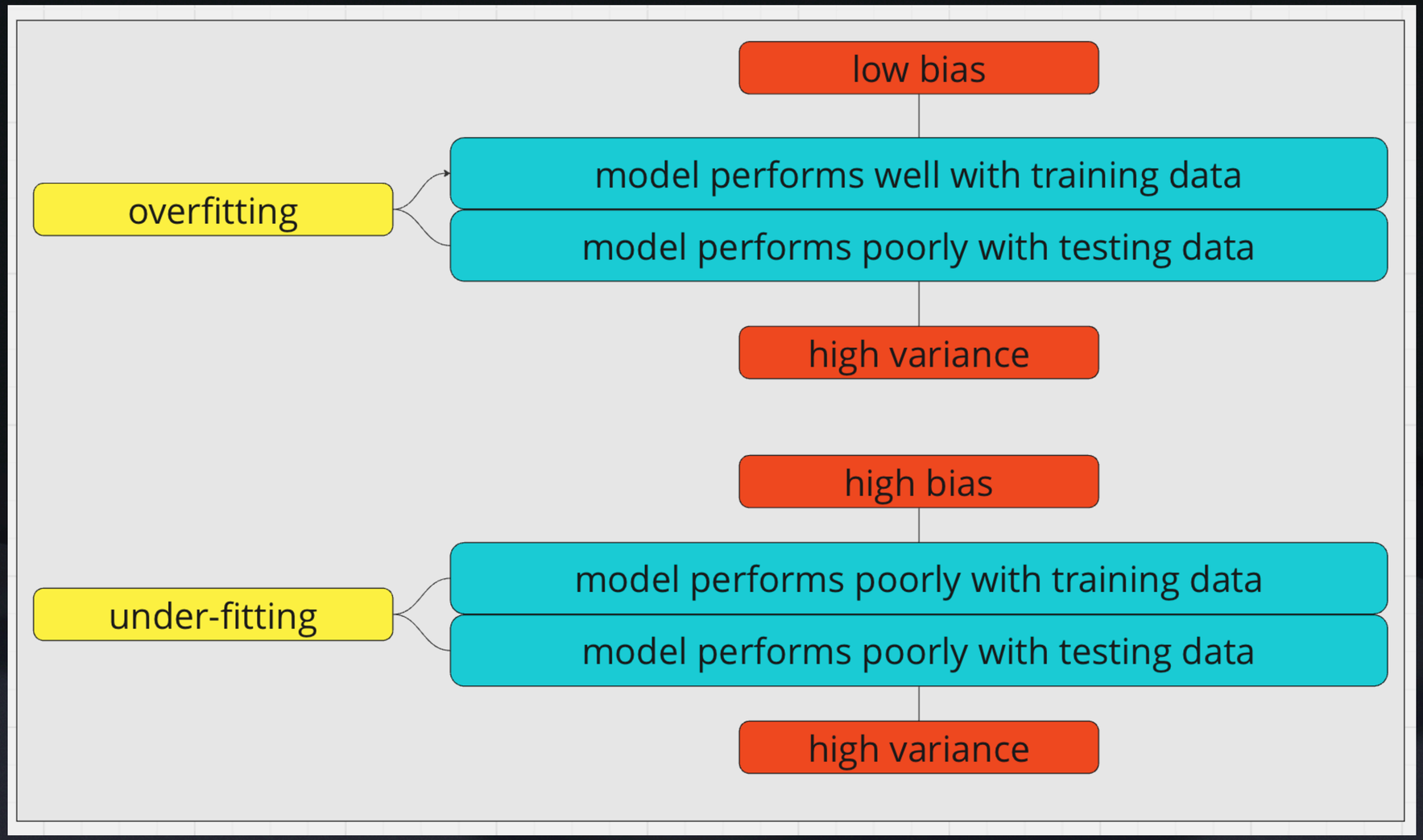
Supervised Learning

Ridge and Lasso Regression



Supervised Learning

Ridge and Lasso Regression



Supervised Learning

Ridge and Lasso Regression

- Low Bias and Low Variance will give us our generalized model
- Bias
  - Influence on the training dataset
- Variance
  - Accounts for the difference or fluctuations of the model

Supervised Learning

Ridge (L2 Normalization) Regression

- Overfitting -> cost function -> 0
- To prevent overfitting
  - Add hyper parameter (lambda) * slope squared to the cost function to keep the cost from falling to 0
  - Repeat until convergence
- Another way I’ve read to think about this is imagine the points in a dataset are getting some distance closer to the data mean.

Supervised Learning

Lasso (L1 Normalization) Regression

- Overfitting -> cost function -> 0
- To prevent overfitting
  - Add hyper parameter (lambda) * the mode of the slope to the cost function
- Features x1, x2, ... xN
  - Whatever features are not playing an amazing role the coefficient value, the slope, will be very small; just like the entire feature is neglected (or set to zero)
- Prevents overfitting, and allows feature selection
- Cross Validation - change the lambda number

Supervised Learning

Assumptions of Linear Regression

- Follows a Normal / Gaussian Distribution —> model will get trained well
- Standardization { SCALING DATA }
- Linearity (linearly separable)
- Multicollinearity
  - x1, x2, x3 | y
    - Let’s say how correlated are x2 and x3; also if it is highly correlated to the dependent feature, it is not necessary to use both features (hence we can drop one)

Supervised Learning

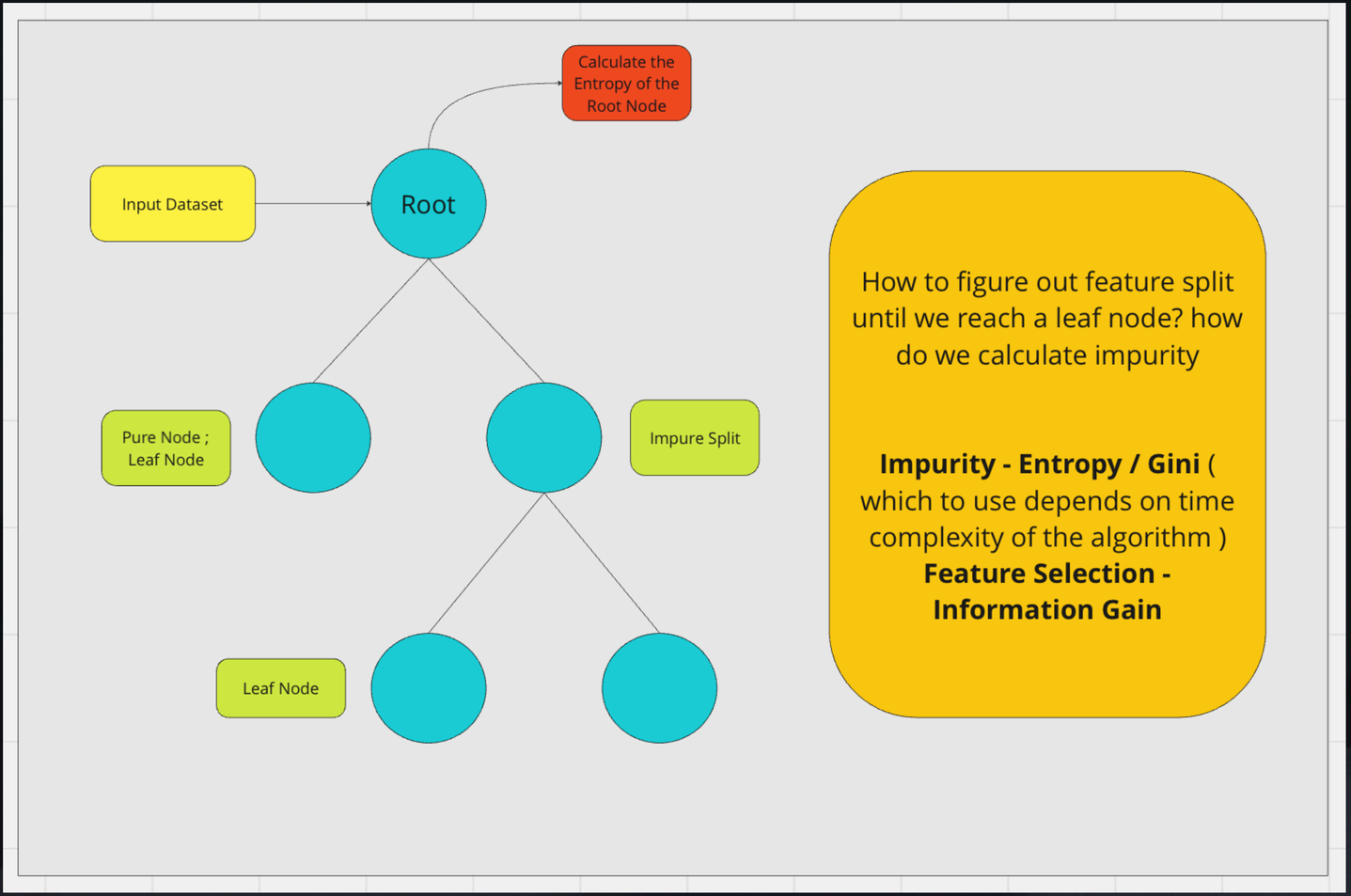
K-Nearest Neighbor Algorithm

- Requires standardization
- Uses proximity, that’s why is is a non-parametric supervised learning technique
- Can be used to solve classification and regression
- Classification
  - After grouping, when presented with new data the algorithm will come to a conclusion for which group the new data belongs to either by the Manhattan or Euclidean distance formula.
- Regression
  - Calculates the average of all points within K (hyperparameter)

Supervised Learning

Decision Tree

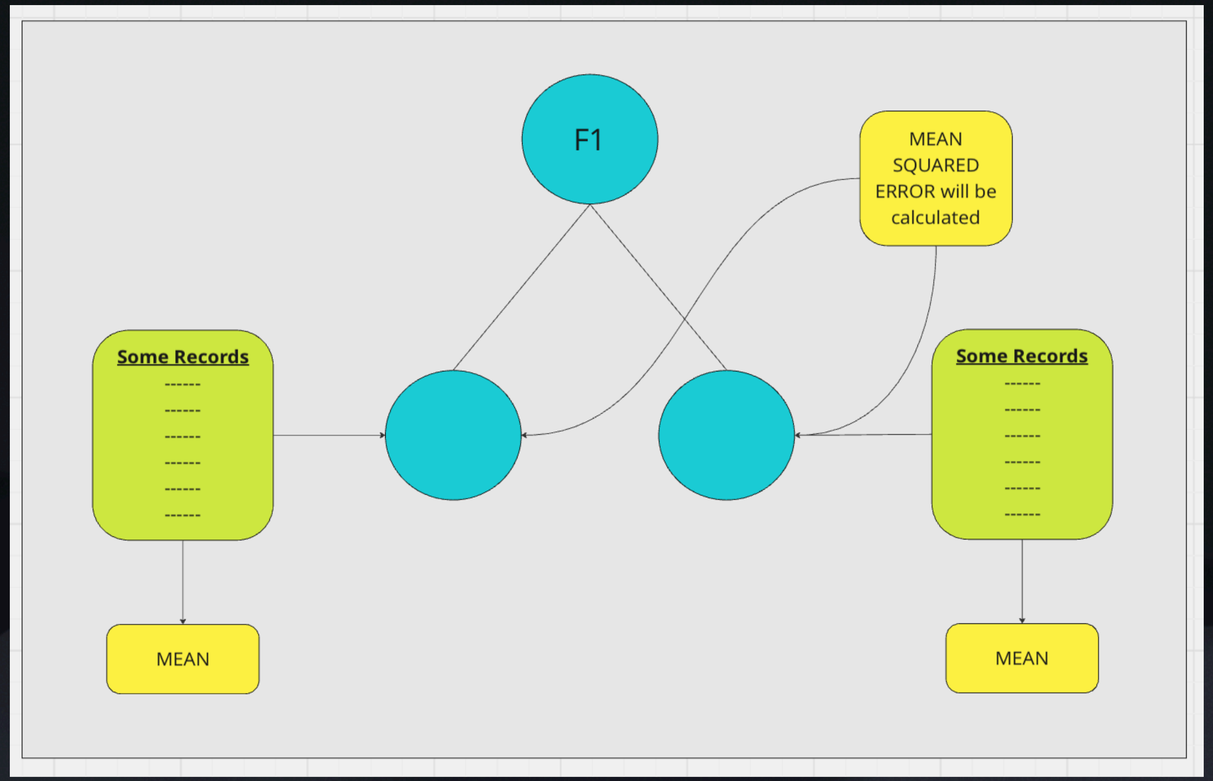
- Does NOT require standardization
- Used to handle regression and classification
- Imagine some nested if else condition



Supervised Learning

Decision Tree Regressor

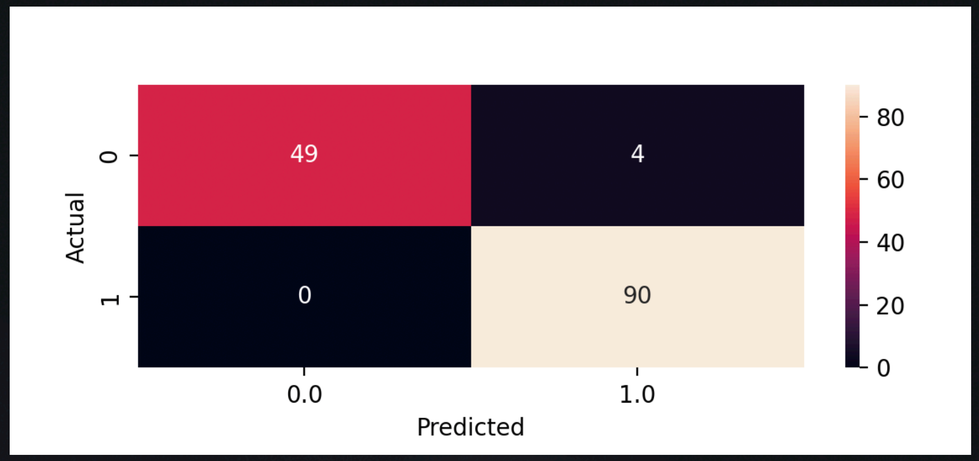
- Output is continuous
- How do I split the feature when getting selected for node in the tree?
  - Entire mean will be calculated of the subsection of data
- Cost function is mean squared error, mean absolute error, maybe a combination of the two



Supervised Learning

XGBoost, AdaBoost, Random Forest

- Xgboost Relies upon Gradient Descent
- Sequential binary trees
- Calculate similarity per node
- Calculate information gain per node



- https://www.kaggle.com/code/cesartrevisan/using-xgboost-to-predict-breast-cancer
  - Also utilized AdaBoost (sequential “stumps” boosting weak learners into a strong learner technique that focuses on wrong answers getting more penalty), and random forest (simultaneously chunked, or bagging, data decision tree aggregator processing weak learners as a strong learner) to see how the algorithms would perform. According to the confusion matrix results, AdaBoost performed the best with about 20 estimators as a hyper parameter regarding the sklearn algorithm(s).

Supervised Learning

Decision Tree

- Hyperparameters
  - Levels or depth of the tree, number of nodes
- Decision Tree —> Overfitting
  - Post pruning : after tree is created; see if we can cut levels of the tree
  - Pre pruning : utilizing hyperparameters to create the tree (GridSearchCV sklearn)
  - Bagging, Boosting
    - Used to prevent overfitting of decision trees
    - Different ways of combining weak learners to create a strong learner

Unsupervised Learning

K-Means Clustering

- K —> number of centroids
- How do we determine the number of centroids (K)?
  - Try with different K values
  - Initialize K number of centroids randomly
  - Compute the average to update the centroids
  - Elbow Method, Silhouette
    - Within cluster sum of squares over iterations that user determines
    - Find nearest cluster, and calculate the summation of all the distances between 1 point in cluster to every point in the other (repeat for every point, and take the average distance : -1 to 1)

Unsupervised Learning

Density-Based Scan

- Helps with outliers in situations where K-Means cannot do as well
- Hyperparameter(s)
  - Min points - minimum number of points to determine cluster
  - Epsilon - radius of search
- If search radius has Min points within epsilon, becomes a core point in a cluster
- If only 1 point within epsilon, becomes a border point
- If points lie outside the cluster epsilon, becomes and outlier