CEDAR 2023 Student Day Tutorial

Doğacan Su Öztürk

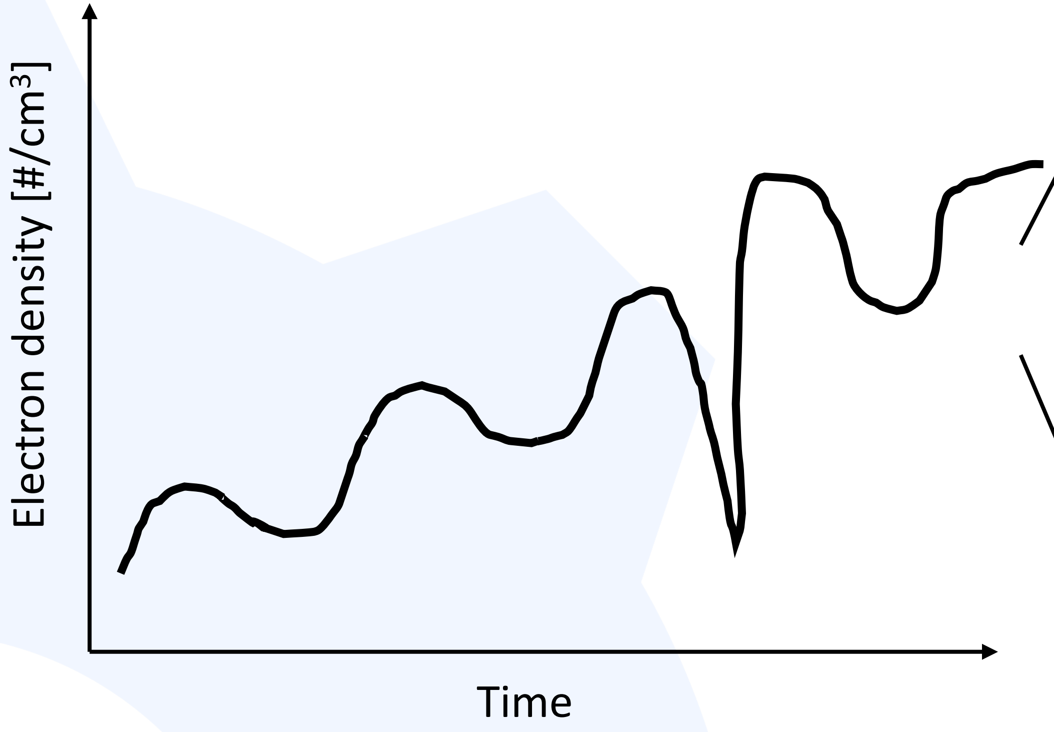
Contact: dsozturk@alaska.edu

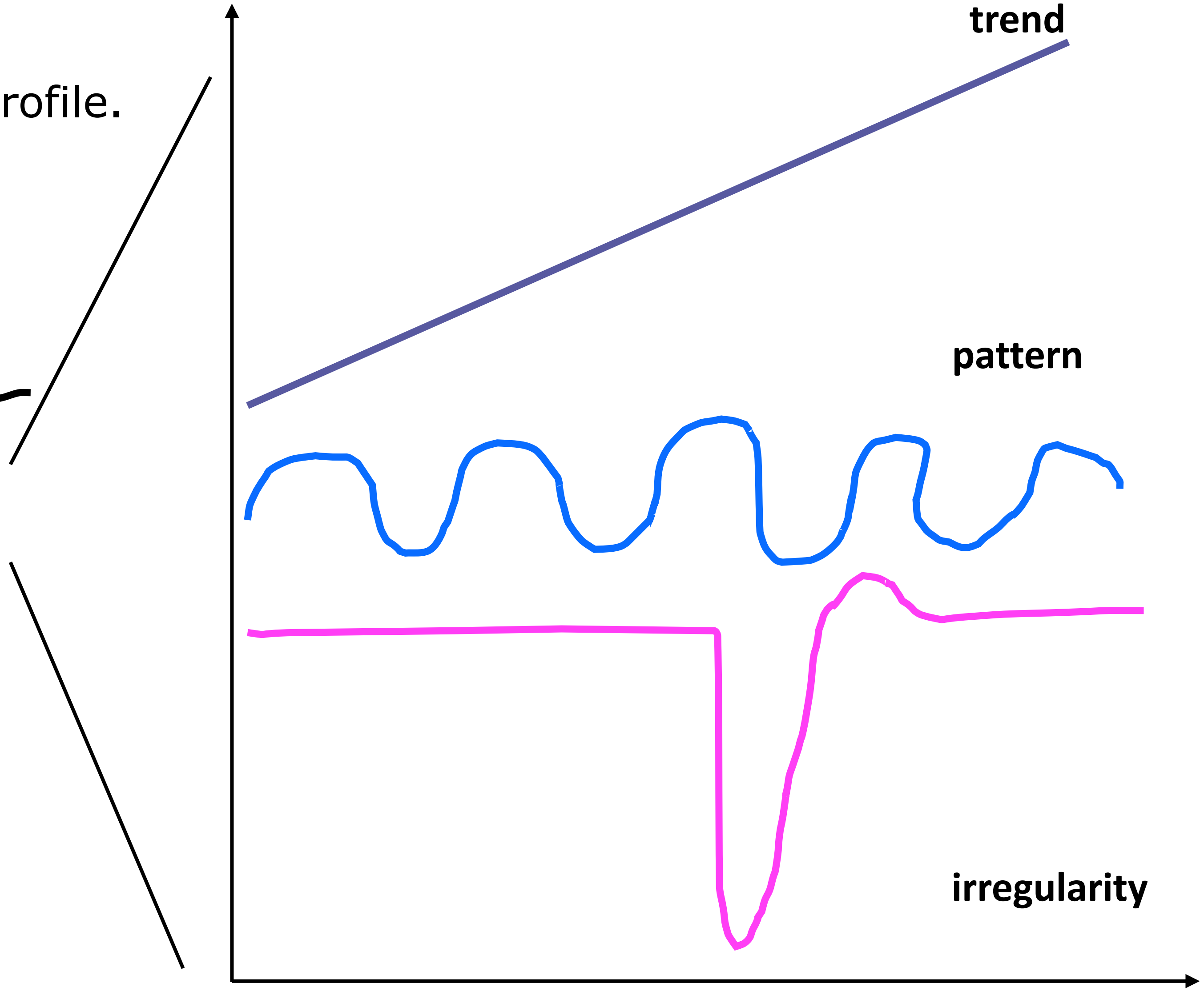
Machine Learning

Image Credit: Visme

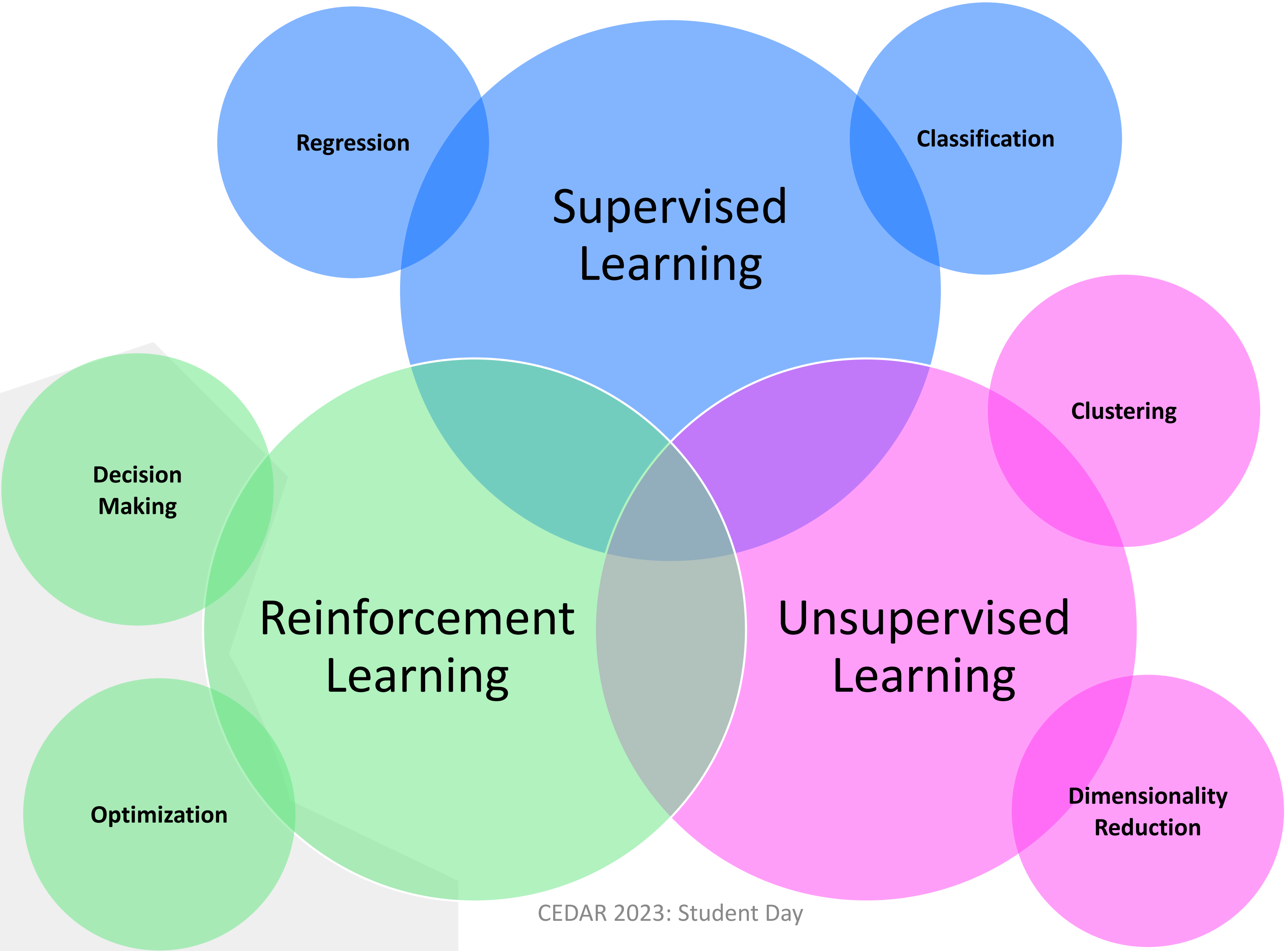
Machine Learning aims to understand information.

Let's say we have an electron density profile.
What can we infer from the data?



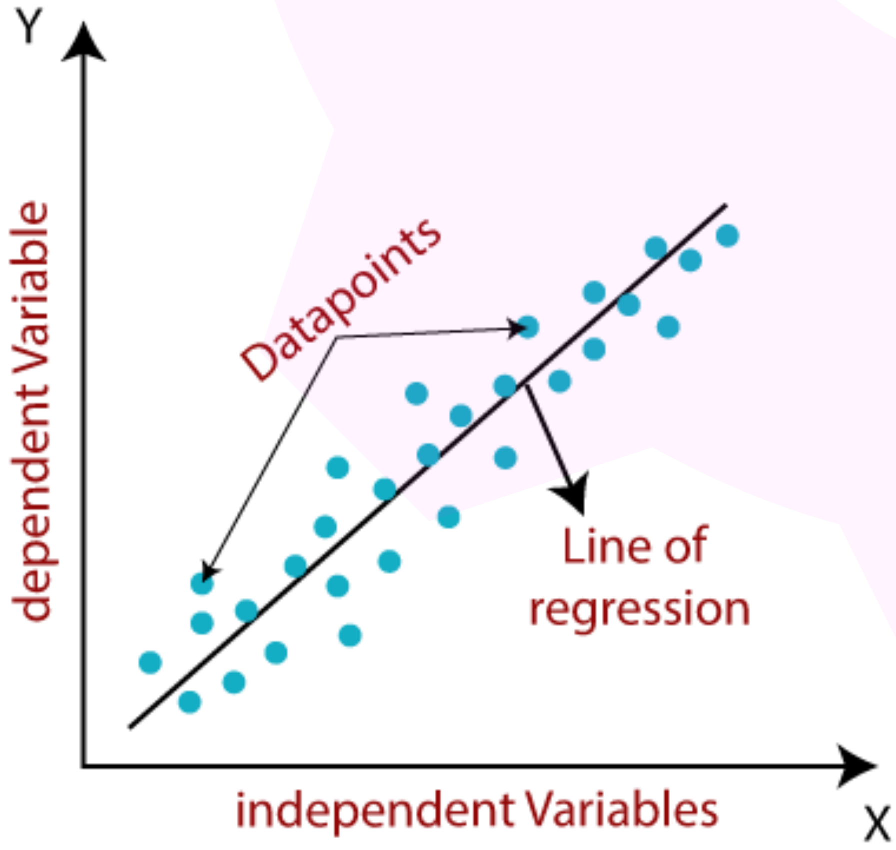


Machine Learning builds upon traditional statistical techniques.



What is regression?

- Regression is the basic relationships pertaining to the data/signal.
- Regression aims to find the most simple relationship between a data and independent variables.
    1. Linear Regression
    2. Polynomial Regression
    3. Logistic Regression
- Regression could be used for interpolation and extrapolation.
- Polynomial regression can be further regularized using weights for coefficients (Ridge, LASSO, etc.)



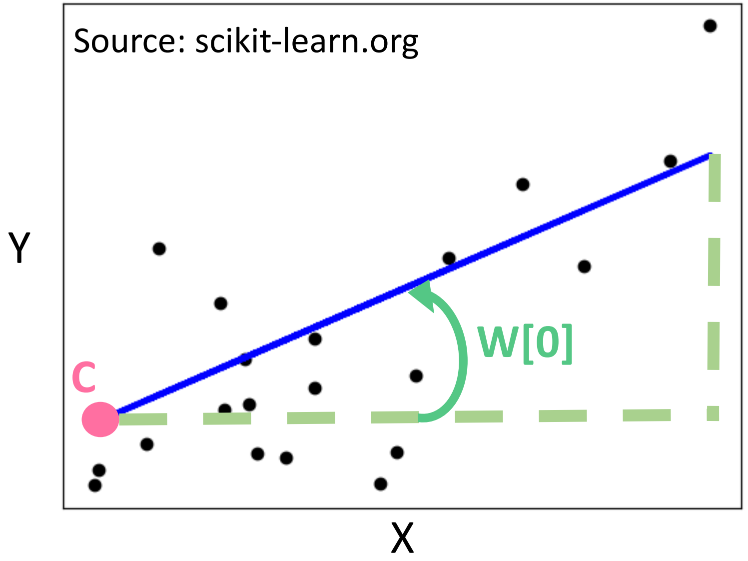
💡 Interpolation (data gaps) and Extrapolation (predicting/forecasting)

Supervised Learning Basics: Linear Regression

Linear regression finds the parameters, weight and constant, to minimize mean squared errors between predictions and actual values.

$$\hat{y} = w[0]x[0] + w[1]x[1] + \cdots + w[n]x[n] + c$$

Here the $\hat{y}$ denotes the prediction model makes, w values are the weights, x are the input parameters (feature columns), and c is a constant.



C: Constant or intercept

W[0]: Slope or first order weight

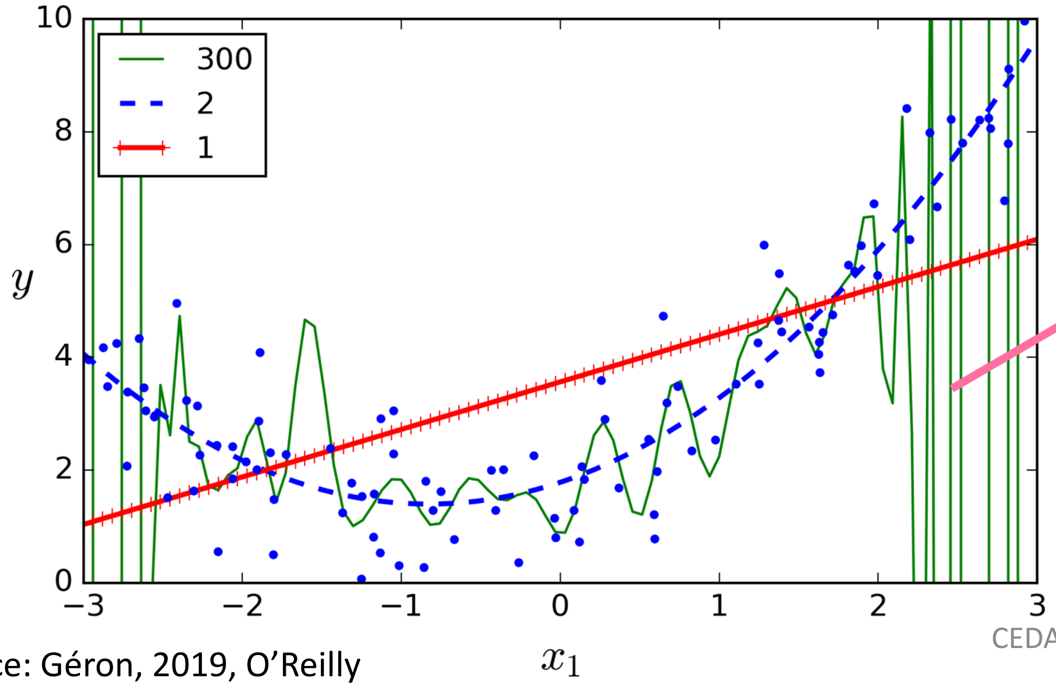
Objective/cost function: $minimize\ \|y - Xw\|_2^2$

Supervised Learning Basics: Polynomial Regression

The linear representation can be improved by adding polynomial features to the fitting function.

$$\hat{y} = w_1 x + w_2 x^2 + c$$

More polynomial terms can be added to increase the order of regression.



Source: Géron, 2019, O'Reilly

But it is not always a good thing.

💡 Regression can be used in various ways: trend detection and forecasting, baseline fitting/removal, data calibration.

Evaluating Regression: Error

Most commonly used error descriptors:

- Mean absolute error

$$\text{mae } error = \frac{1}{N}\sum_{i=1}^{N}|actual\ values\ - predictions|$$

- Mean absolute percentage error

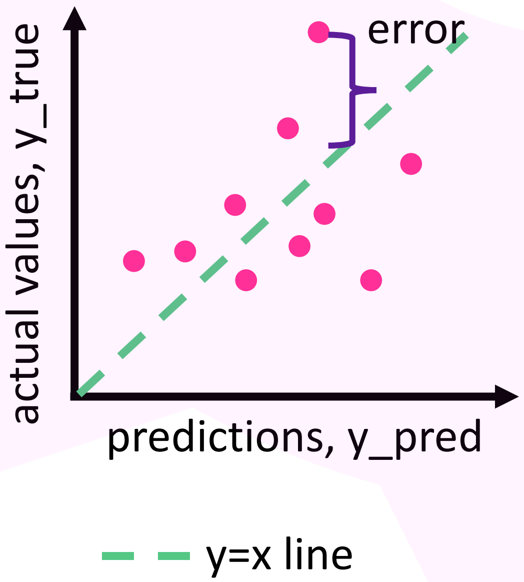
$$\text{mape } error = \frac{100\ \%}{N}\sum_{i=1}^{N}\left|\frac{actual\ values\ - predictions}{actual\ values}\right|$$

- Mean squared error

$$\text{mse } error = \frac{1}{N}\sum_{i=1}^{N}(actual\ values\ - predictions)^2$$

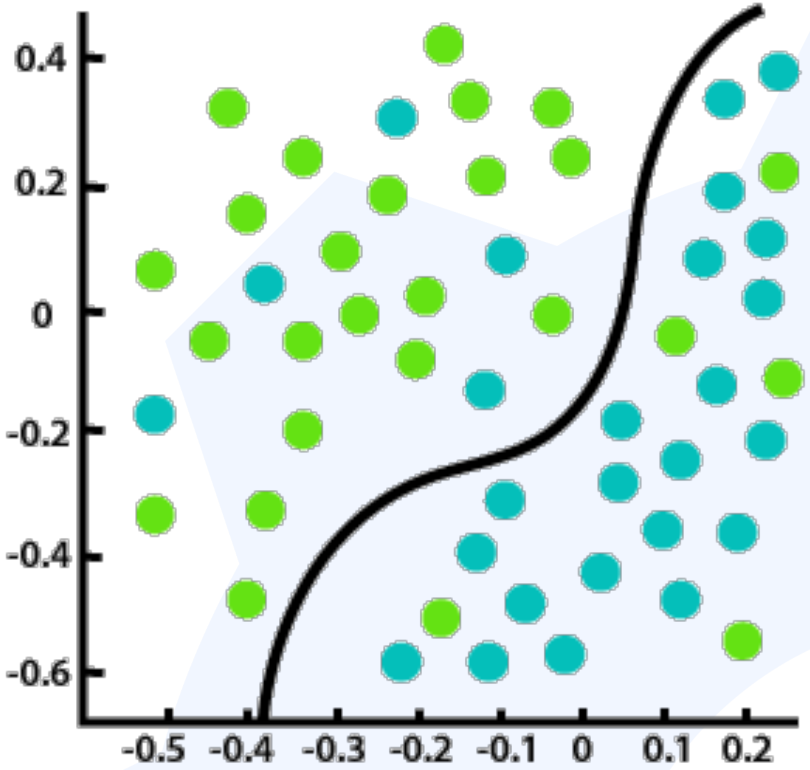
- Root mean squared error

$$\text{rmse } error = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(actual\ values\ - predictions)^2}$$



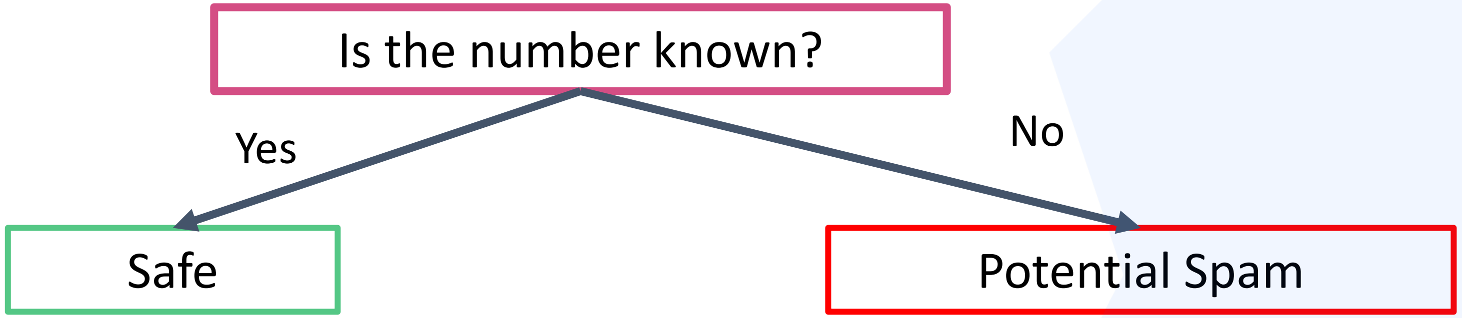
What is classification?

- Classification is one of the most common supervised learning techniques.
- It can be simplified as “predicting classes/categories”.
- A classification application consists of the following:
    - Classifier: Algorithm chosen for the task
    - Classification model: The model that predicts the class
    - Feature: Descriptors of the data set leading to distinct classes
    - Binary classification: Classification task with two outcomes
    - Multi-class Classification: Each sample belongs to only one class
    - Multi-label Classification: A sample can be assigned to a set of classes
    - Target: The class
    - Evaluation: Evaluation of the model’s prediction capability
- When you are training your machine learning algorithms you have to pick whether the problem at hand is a classification or regression problem. It is not as easy as it sounds.



Supervised Learning Basics: Binary Classification

- Classification with only 2 classes is called binary classification.
- Can you think of some examples?
- What have we learnt so far to help with Binary Classification?



💡 Binary classification can be used for event/anomaly detection applications.

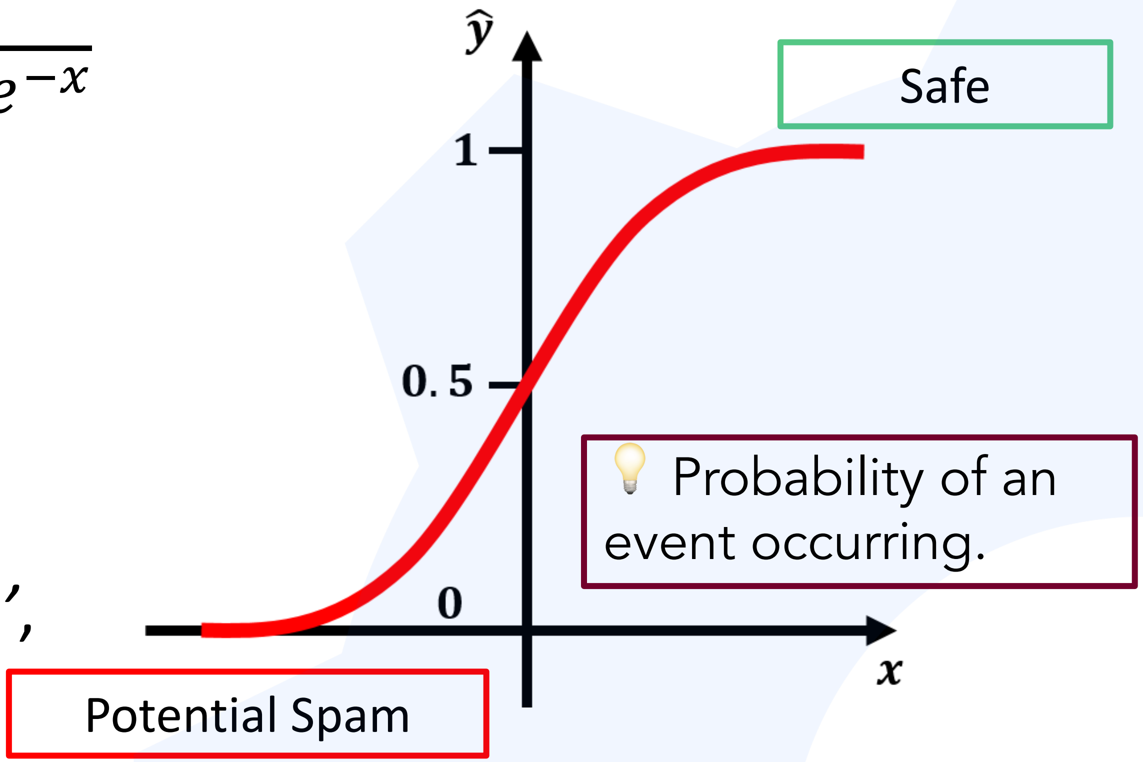
Supervised Learning Basics: Logarithmic Classification

Logistic regression is commonly used to provide a probability (pass/fail, win/lose).
It is the binary classification analog of linear regression.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

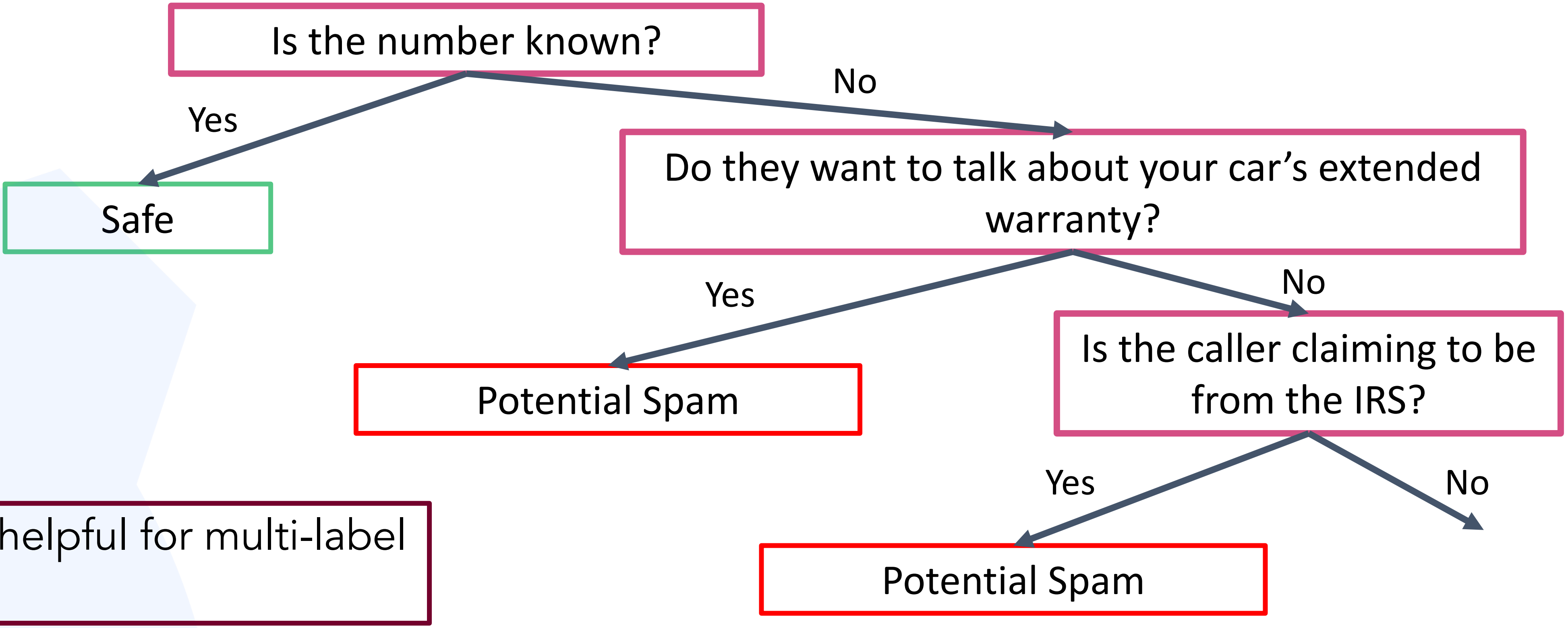
from sklearn.linear_model import LogisticRegression

class sklearn.linear_model.LogisticRegression(
*penalty='l2', *, dual=False, tol=0.0001, C=1.0,*
fit_intercept=True, intercept_scaling=1,
class_weight=None, random_state=None,
solver='lbfgs', max_iter=100, multi_class='auto',
verbose=0, warm_start=False, n_jobs=None,
l1_ratio=None)



Supervised Learning Basics: Decision Trees

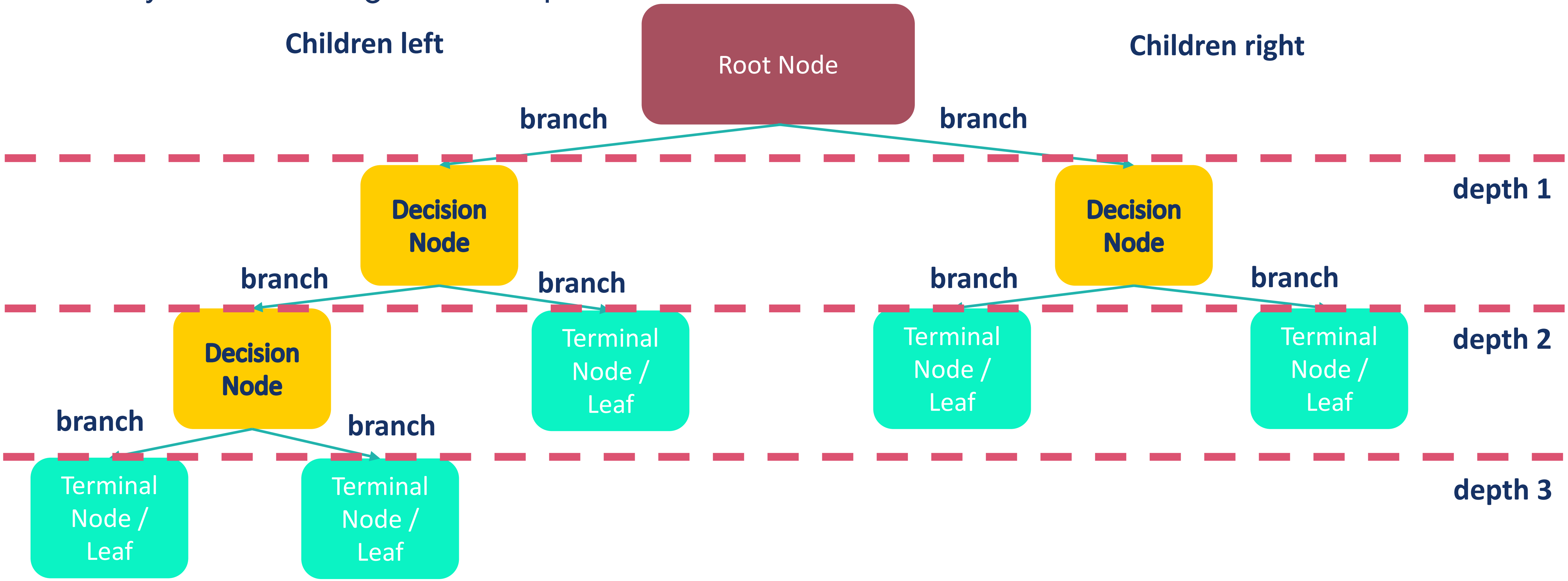
Decision trees could help process more complex cases.



💡 Decision trees are helpful for multi-label categorization.

Decision Trees in Depth

- Decision trees are widely used for both Regression and Classification.
- They learn through if/else questions to arrive at a decision.



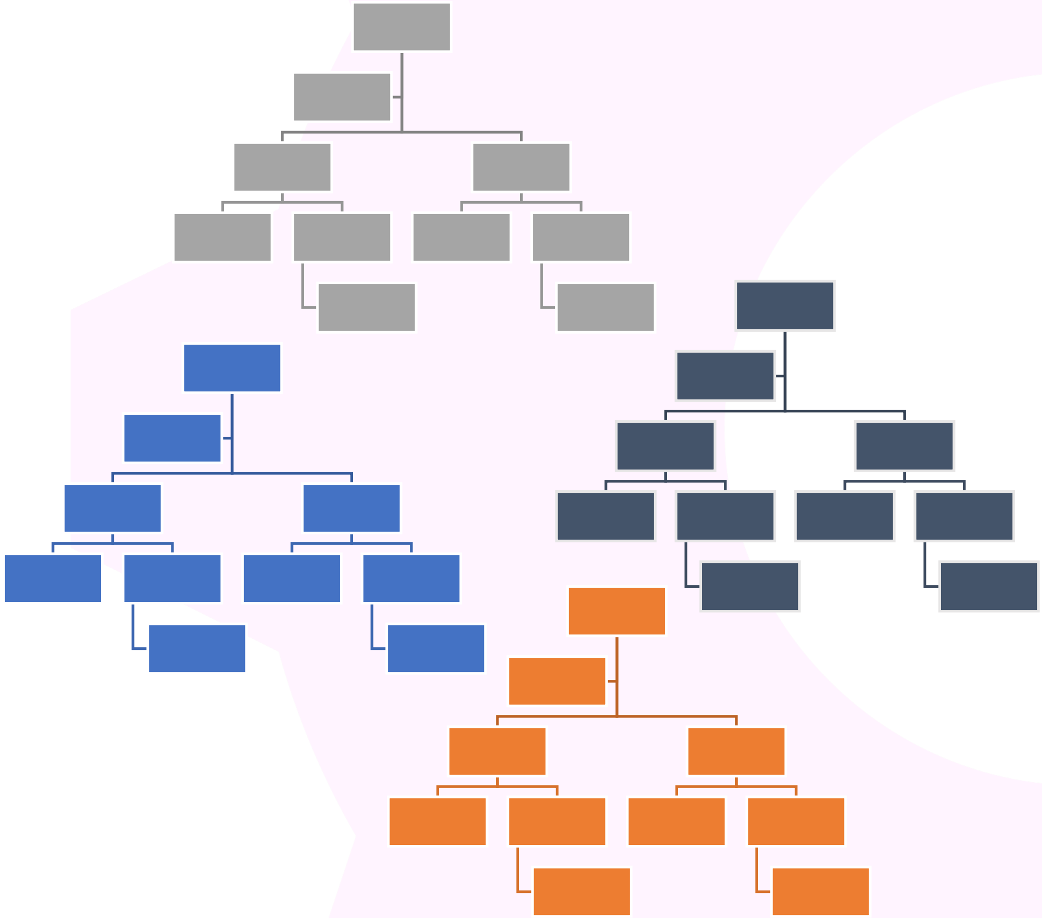
Decision Trees in Depth

- Decision trees are widely used for both Regression and Classification.
- They learn through if/else questions to arrive at a decision.
- Each node has a certain amount of sample in it.
- Decision trees are:
  - Very likely to overfit.
  - Very likely to be limited by the range of the data.
- Decision trees can predict both Regression and Classification.
- **from sklearn.tree import** DecisionTreeRegressor
  class sklearn.tree.DecisionTreeRegressor(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, ccp_alpha=0.0*)
- **from sklearn.tree import** DecisionTreeClassifier
  class sklearn.tree.DecisionTreeClassifier(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0*)

https://scikit-learn.org/stable/modules/tree.html

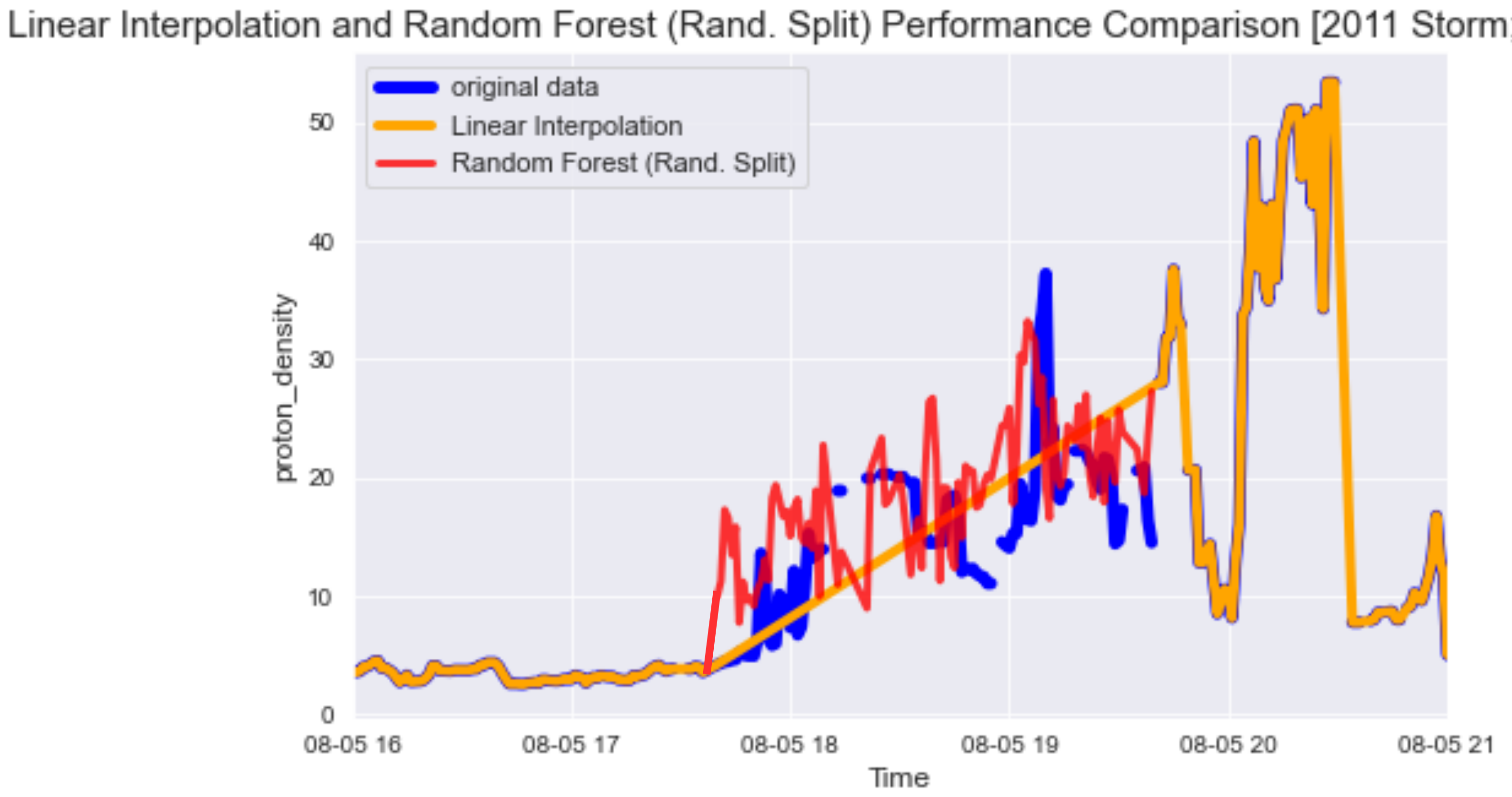
Ensemble Methods: A random forest

- Another way to overcome overfitting is growing a forest of random decision trees with constraints.
- Random forest trains multiple decision trees in parallel and ensembles their predictions into a single decision tree.
- Each decision tree is trained with a random subset of observations.
- Random forests can be further constrained.
- **from sklearn.ensemble import** RandomForestRegressor
  class sklearn.ensemble.RandomForestRegressor(*n_estimators=100*, ***, *criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features='auto'*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *bootstrap=True*, *oob_score=False*, *n_jobs=None*, *random_state=None*, *verbose=0*, *warm_start=False*, *ccp_alpha=0.0*, *max_samples=None*)



Data gap prediction with Random Forest Models

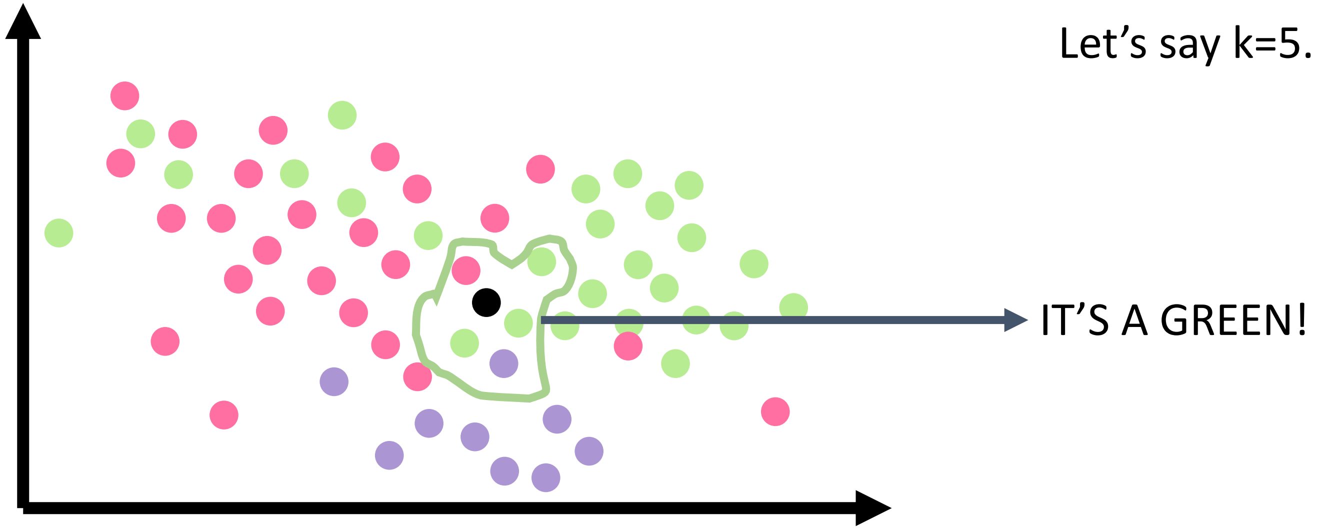
RF models are known to overfit, however they achieve a higher performance than linear interpolation.



Jasmine Kobayashi, SWRI Boulder

Supervised Learning Basics: k-Nearest Neighbours

- Depending on the k number of neighbouring point labels from training data, kNN algorithms predicts a label for the point.



- **from sklearn.neighbors import** KNeighborsClassifier

  class sklearn.neighbors.KNeighborsClassifier(*n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs*)

Evaluating Classification: Confusion Matrix

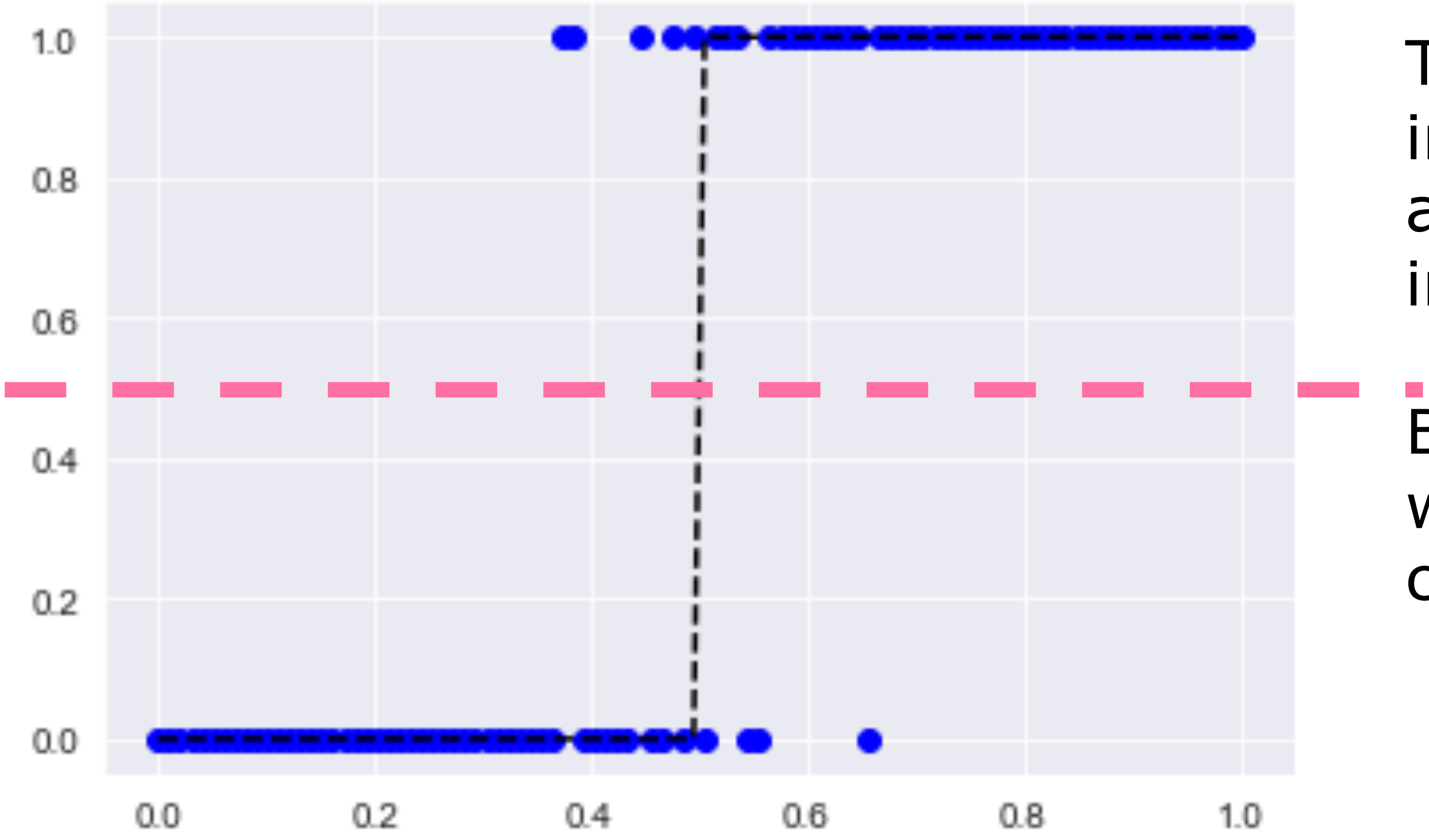
- There are four possible outcome of predicting classes.
  - True positive - Predict 1 when the actual class is 1.
  - False positive - Predict 1 when the actual class is 0.
  - True negative - Predict 0 when the actual class is 0.
  - False negative - Predict 0 when the actual class is 1.
- Accuracy metric alone can not account for the false positive and negative.

TN	**FP**
FN	**TP**

- **from sklearn.metrics import** confusion_matrix

  sklearn.metrics.confusion_matrix(*y_true*, *y_pred*, *, *labels=None*, *sample_weight=None*, *normalize=None*)

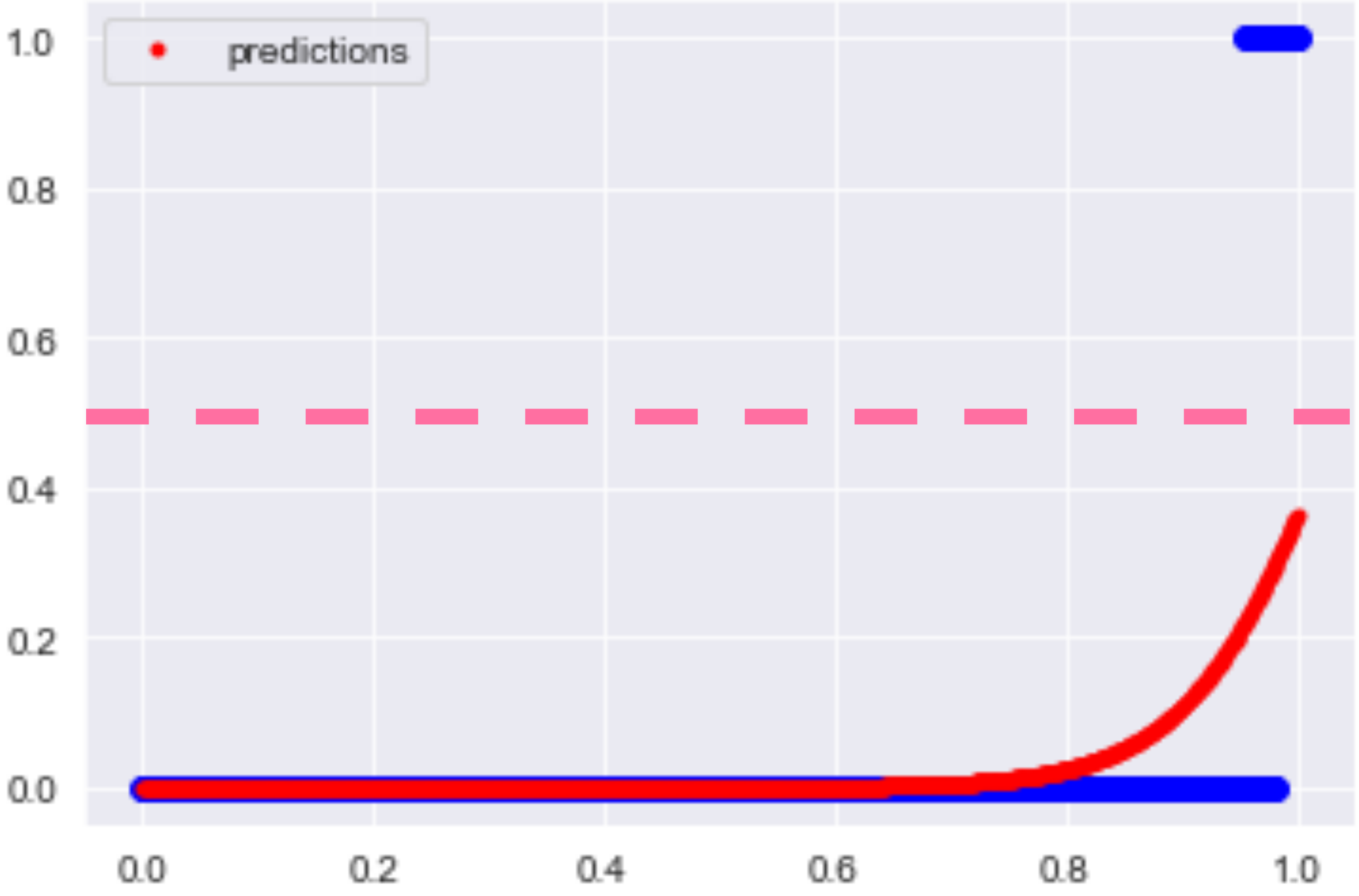
How the decisions are made: Prediction Threshold



Threshold is 0.5 by default in binary classification and also across sklearn implementations.

But what happens when we have an imbalanced class?

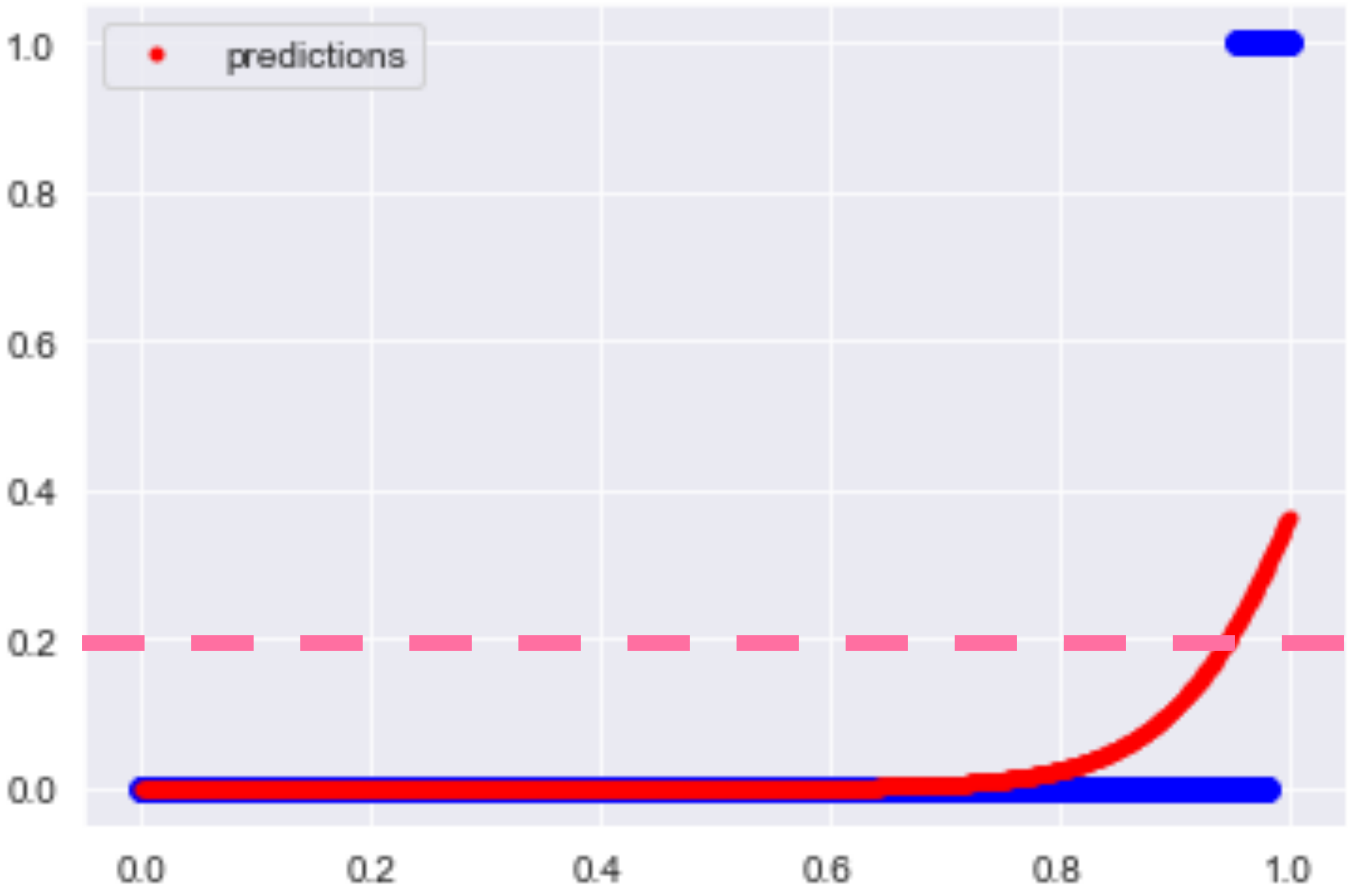
How the decisions are made: Prediction Threshold



- What if we have an unevenly distributed classification problem?
- How will the threshold affect the results?

95	**0**
5	0

How the decisions are made: Prediction Threshold



- What if we have an unevenly distributed classification problem?
- How will the threshold affect the results?

95	**0**
1	4

Evaluating Classification: Precision-recall

- Precision is defined as TP/(TP+FP).
- Recall is defined as TP/(TP+FN).
- Precision-recall is a measure of the "success" of prediction. Here, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

from sklearn.metrics import precision_recall_curve

sklearn.metrics.precision_recall_curve(*y_true*, *probas_pred*, ***, *pos_label=None*, *sample_weight=None*)

from sklearn.metrics import PrecisionRecallDisplay

class sklearn.metrics.PrecisionRecallDisplay(precision, recall, *, average_precision=None, estimator_name=None, pos_label=None)[source]

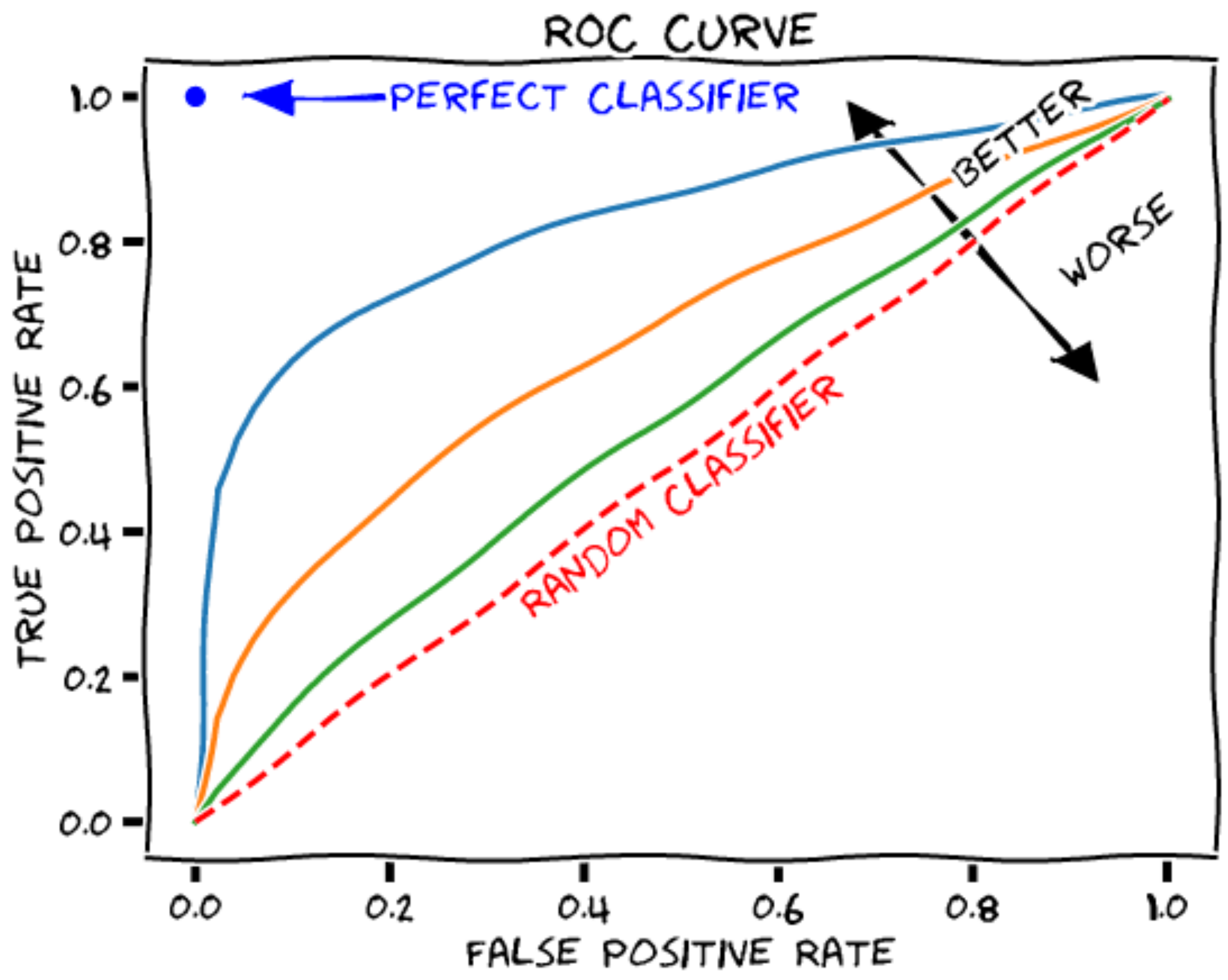
Evaluating Classification: ROC curve

- Receiver Operating Characteristic Curve is the graph that shows the performance of a binary classifier. TP vs FP curve at various threshold settings.
- AUROC: area under ROC curve is a reliable metric for binary classification. It shows the probability that a random positive class observation ranks higher than a random negative class observation.

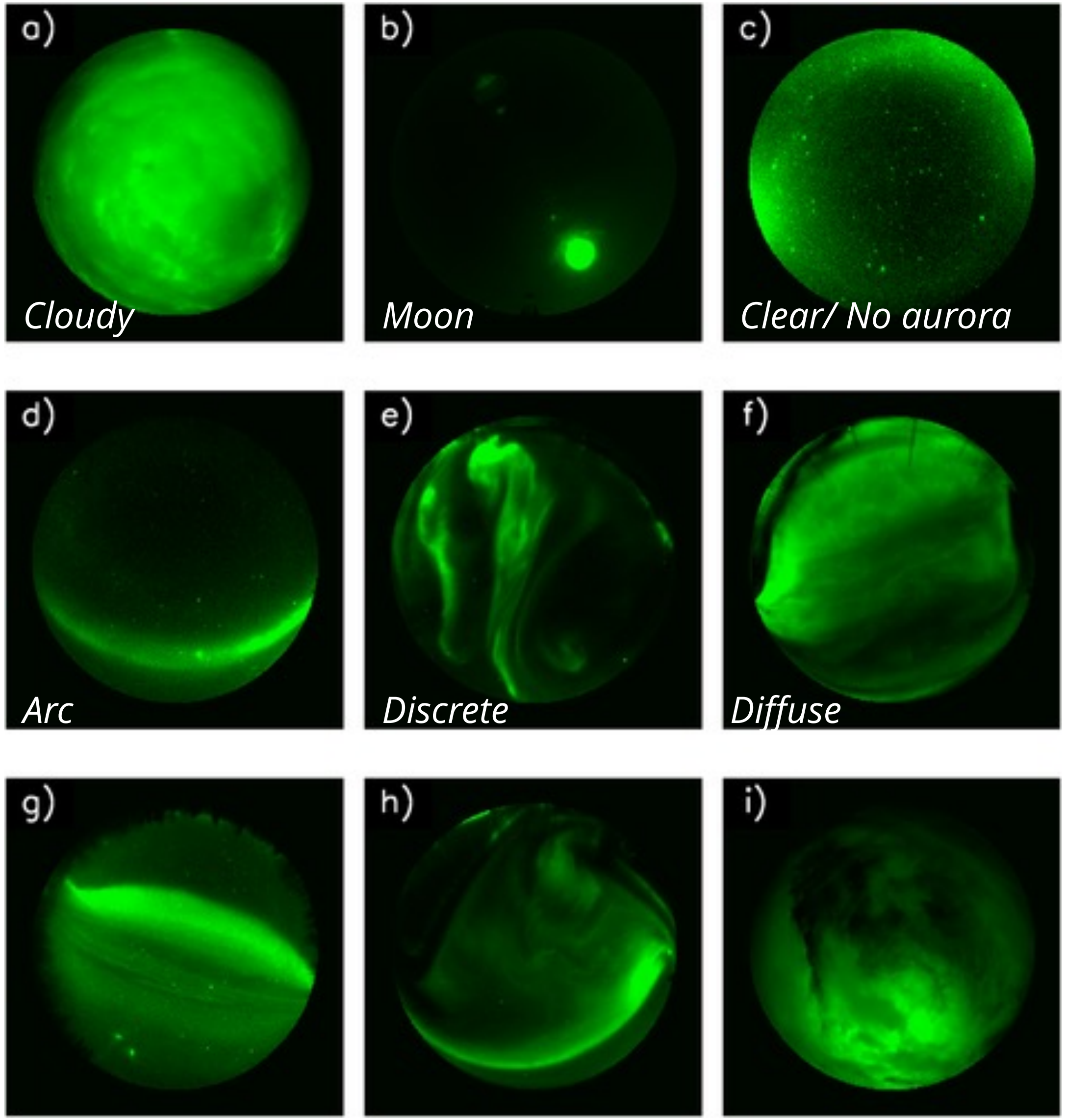
from sklearn.metrics import roc_curve, roc_auc_score

sklearn.metrics.roc_curve(*y_true*, *y_score*, ***, *pos_label=None*, *sample_weight=None*, *drop_intermediate=True*)

sklearn.metrics.roc_auc_score(*y_true*, *y_score*, ***, *average='macro'*, *sample_weight=None*, *max_fpr=None*, *multi_class='raise'*, *labels=None*)



Predicting different classes of aurora

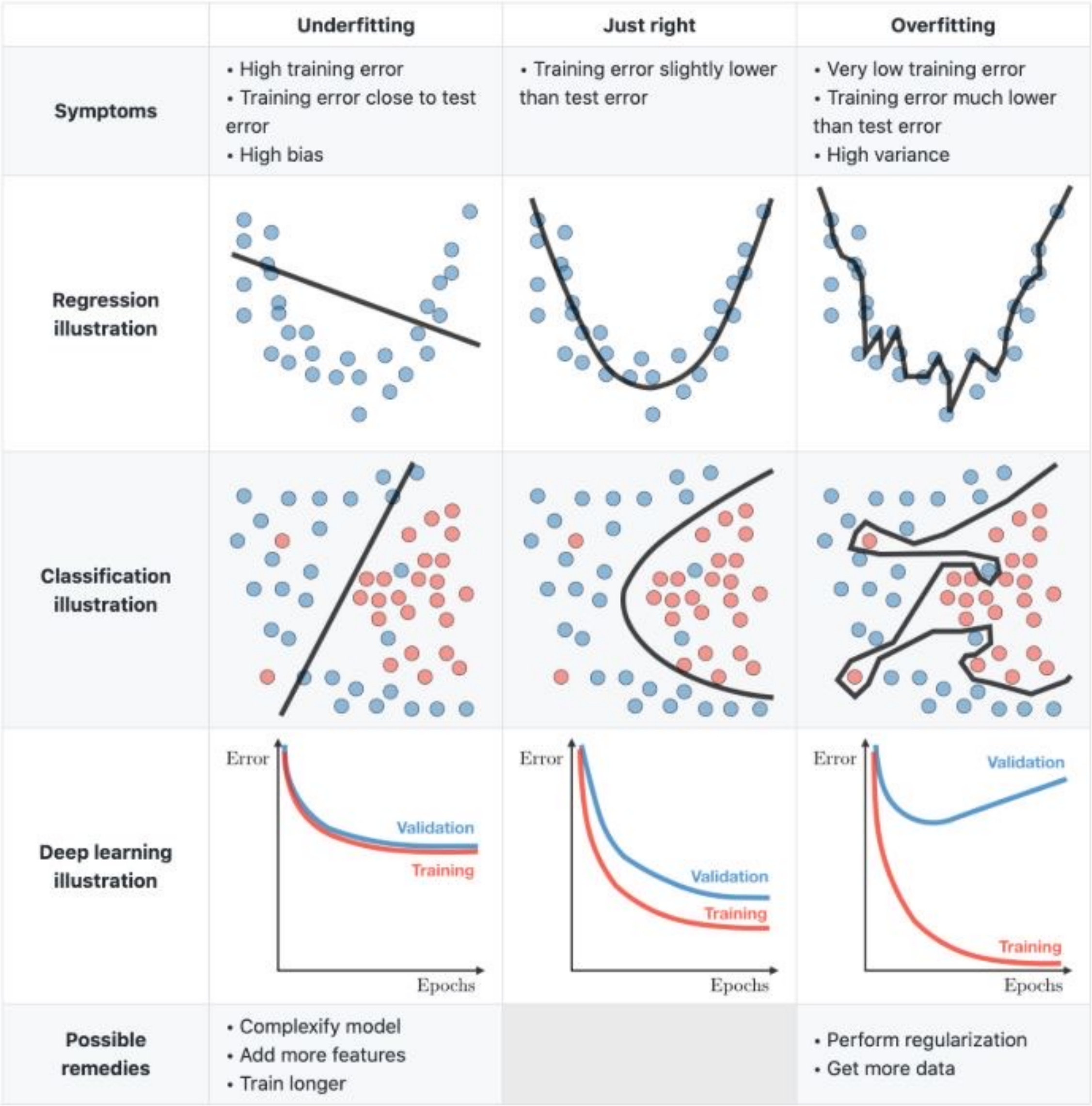


The OATH data set consists of 5824 images labelled with the Ridge classifier and lays the ground work furthering automation of aurora classification.

- Enables statistical analysis of large auroral data sets.
- Various ML techniques (i.e. fully supervised) require labelled data sets.
- Could be adapted to different meso-scale auroral feature detection.

Similar classification work done by Shang et al., 2023; Sado et al., 2022; Guo et al., 2022; Yang et al., 2019 obtained high performance results.

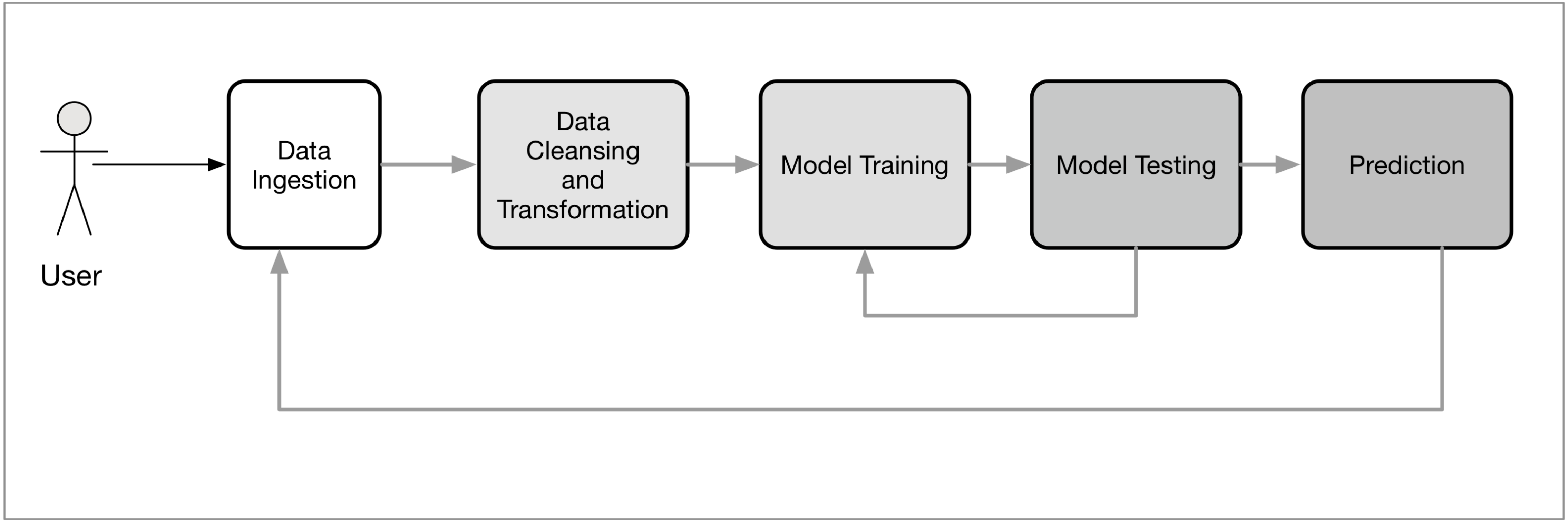
Evaluating Accuracy of Models: Underfitting-Overfitting



Conditions	Underfitting	Overfitting
Model parameters	Increase model complexity	Decrease model complexity
Regularization	Decrease regularization	Increase regularization
Data Quality	Improve data cleaning	Avoid downsampling
Feature set	Increase features	Decrease features
Data set	MORE DATA ALWAYS HELPS	

A Machine Learning Application Framework

A machine learning application consists of the following steps:



8 Main Steps for The Machine Learning Process

1. Frame the problem and look at the big picture.
2. Get the data.
3. Perform exploratory analysis.
4. Prepare the data for the ML applications.
5. Explore different models and shortlist the best ones.
6. Fine-tune the models and combine them into a solution.
7. Present your solution.
8. Launch, monitor, and maintain your system.

The 8 ML Principles from the Institute for Ethical AI & Machine Learning

1. Human augmentation

I commit to assess the impact of incorrect predictions and, when reasonable, design systems with human-in-the-loop review processes

2. Bias evaluation

I commit to continuously develop processes that allow me to understand, document and monitor bias in development and production.

3. Explainability by justification

I commit to develop tools and processes to continuously improve transparency and explainability of machine learning systems where reasonable.

4. Reproducible operations

I commit to develop the infrastructure required to enable for a reasonable level of reproducibility across the operations of ML systems.

5. Displacement strategy

I commit to identify and document relevant information so that business change processes can be developed to mitigate the impact

6. Practical accuracy

I commit to develop processes to ensure my accuracy and cost metric functions are aligned to the domain-specific applications.

7. Trust by privacy

I commit to build and communicate processes that protect and handle data with stakeholders that may interact with the system directly

8. Data risk awareness

I commit to develop and improve reasonable processes and infrastructure to ensure data and model security are being taken into

https://ethical.institute/principles.html

Resources for an Ethical ML Framework

Quick links to sections in this page

Explaining predictions & models	Privacy preserving ML	Model & data versioning
Model Training Orchestration	Model Serving and Monitoring	Neural Architecture Search
Reproducible Notebooks	Visualisation frameworks	Industry-strength NLP
Data pipelines & ETL	Data Labelling	Metadata Management
Functions as a service	Computation distribution	Model serialisation
Optimized computation frameworks	Data Stream Processing	Outlier and Anomaly Detection
Feature engineering	Feature Stores	Adversarial Robustness
Commercial Platforms	Data Storage Optimization	

https://github.com/EthicalML/awesome-production-machine-learning#model-and-data-versioning

Contact: dsozturk@alaska.edu

Thank you!