



PPgEEC

Machine Learning

Fundamentals of ML and Decision Trees

01

What is Machine Learning?

02

Machine Learning Types

03

Main Challenges of Machine Learning

Variables, pipelines, controlling chaos, data segregation,
bias vs variance

"A reproducible Pipeline is all you need"

04

Decision Tree

Introduction, Mathematical Foundations

05

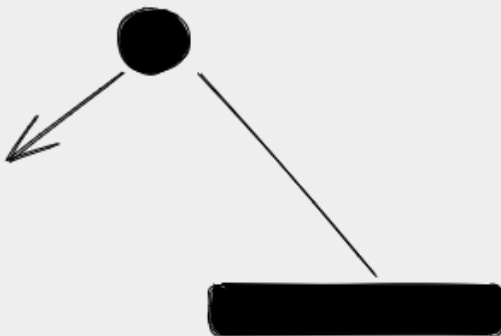
Evaluation Metrics

Best practices, threshold and ranking metrics

06

Case study

What is Machine Learning?

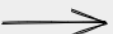


```
if (ball.collide(brick)){  
    removeBrick();  
    ball.dx = 1.1*(ball.dx);  
    ball.dy = -1*(ball.dy);  
}
```

Rules



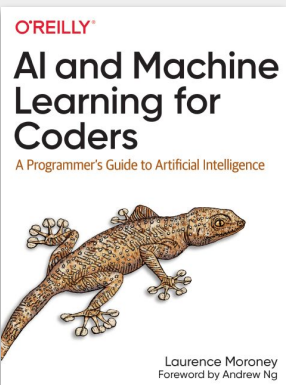
Data



Traditional
Programming



Answers



Limitations of traditional programming

<activity detection>



```
if (speed < 4){  
    status = WALKING;  
}
```



```
if (speed < 4){  
    status = WALKING;  
} else {  
    status = RUNNING;  
}
```



```
if (speed < 4){  
    status = WALKING;  
} else if (speed < 12) {  
    status = RUNNING;  
} else {  
    status = BIKING;  
}
```



// ????



From coding to ML

<gathering and label data>



```

010111100001110101
111010101010111000
111010101010101010
000000000111001111
  
```

Label = WALKING



```

010111100011101110
000111010101011011
000111101010111000
000000000000001111
  
```

Label = RUNNING



```

111101110010101011
110101110101011011
111110101110010101
000111111001101111
  
```

Label = BIKING

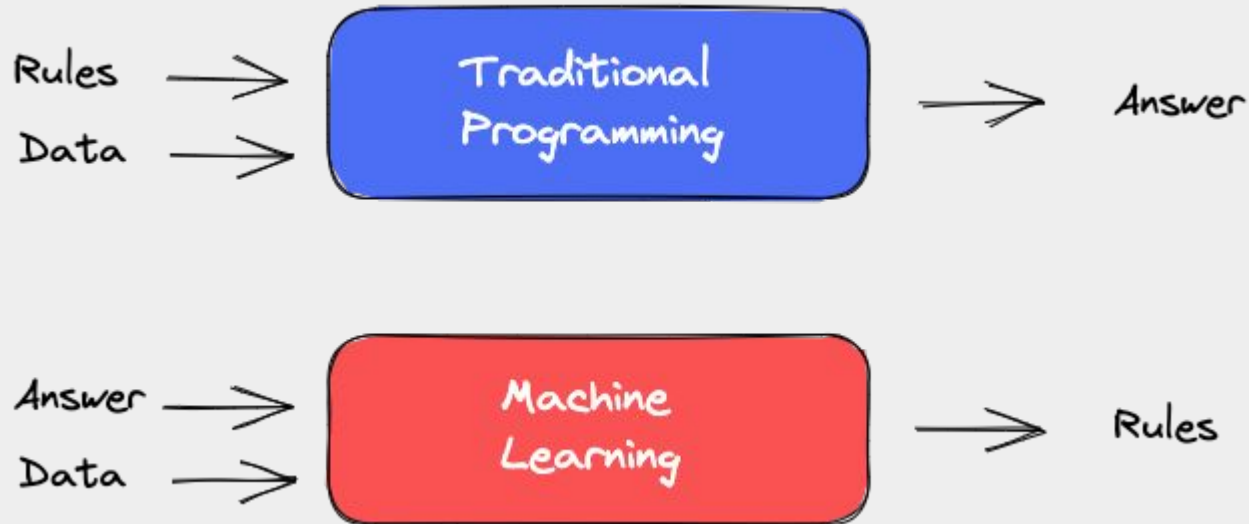


```

100000000010101011
111111111000011001
000000011100111101
111111111100000001
  
```

Label = GOLFING

From programming to learning



What is Machine Learning?

Machine Learning (ML): a subset of AI that often uses statistical techniques to give machines the ability to "learn" from data without being explicitly given the instructions for how to do so. This process is known as "training" a "model" using a learning "algorithm" that progressively improves models performance on a specific task.

Computer Vision

 Semantic Segmentation 203 benchmarks 2300 papers with code	 Image Classification 279 benchmarks 1989 papers with code	 Object Detection 264 benchmarks 1737 papers with code	 Image Generation 169 benchmarks 771 papers with code	 Denoising 100 benchmarks 739 papers with code
---	---	---	--	---

Natural Language Processing

 Language Modelling 27 benchmarks 1513 papers with code	 Machine Translation 73 benchmarks 1366 papers with code	 Question Answering 103 benchmarks 1307 papers with code	 Sentiment Analysis 69 benchmarks 836 papers with code	 Text Generation 84 benchmarks 649 papers with code
--	---	---	---	--

Medical

 Medical Image Segmentation 86 benchmarks 244 papers with code	 Drug Discovery 14 benchmarks 151 papers with code	 Lesion Segmentation 6 benchmarks 104 papers with code	 Brain Tumor Segmentation 10 benchmarks 69 papers with code	 COVID-19 Diagnosis 4 benchmarks 59 papers with code
---	---	---	--	---




Graphs

 Link Prediction 69 benchmarks 463 papers with code	 Node Classification 77 benchmarks 370 papers with code	 Graph Embedding 2 benchmarks 252 papers with code	 Graph Classification 54 benchmarks 209 papers with code	 Community Detection 11 benchmarks 156 papers with code
--	--	---	---	--




Time Series

 Time Series 2 benchmarks 1127 papers with code	 EEG 8 benchmarks 177 papers with code	 Imputation 10 benchmarks 160 papers with code
--	---	---




Speech

 Speech Recognition 121 benchmarks 575 papers with code	 Speech Synthesis 3 benchmarks 142 papers with code	 Dialogue Generation 10 benchmarks 108 papers with code
--	--	--

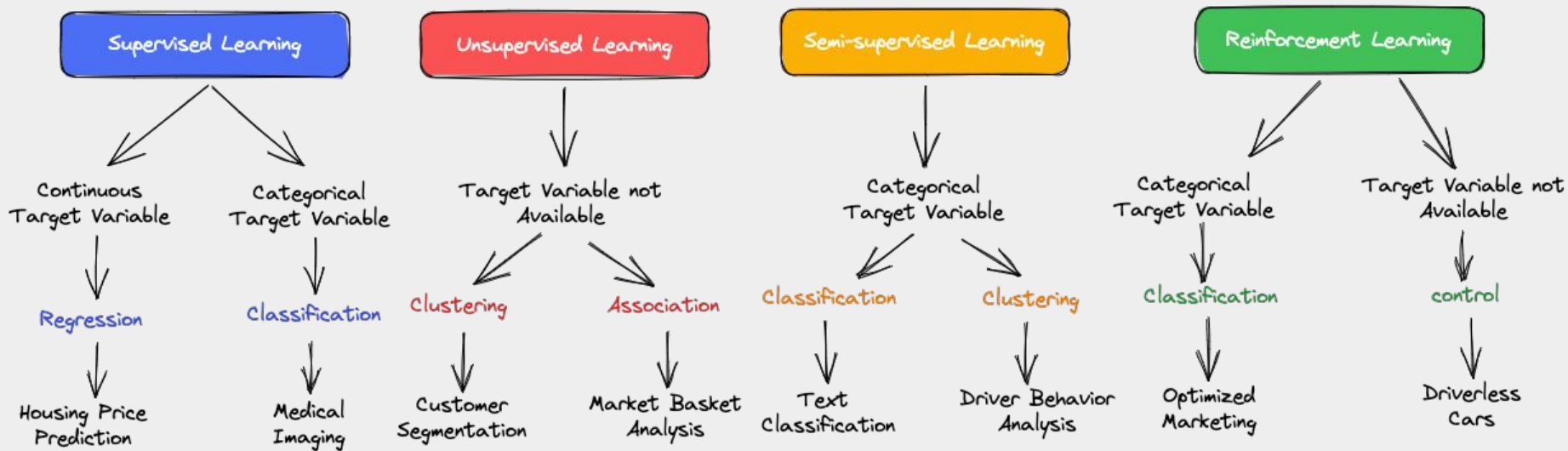
Playing Games

 Continuous Control 76 benchmarks 242 papers with code	 Atari Games 65 benchmarks 213 papers with code	 OpenAI Gym 9 benchmarks 112 papers with code
---	--	--

Music

 Music Generation 60 papers with code	 Music Information Retrieval 55 papers with code	 Music Source Separation 3 benchmarks 31 papers with code
---	--	--

Machine Learning Types



Supervised Learning

Classification Problem

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
Yes	Yes	205	Yes
No	Yes	180	Yes
Yes	No	210	Yes
Yes	Yes	167	Yes
No	Yes	156	No
No	Yes	125	No
Yes	No	168	No
Yes	Yes	172	No

Chest Pain	Blocked Arteries	Patient Weight
Yes	No	200

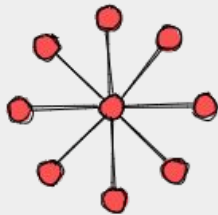
Supervised Learning

Regression Problem

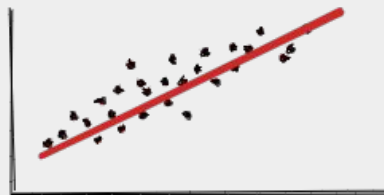
Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Height (m)	Favorite Color	Gender
1.83	Yellow	Male

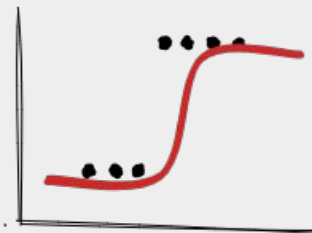
K-Nearest Neighbors (KNN)



Linear Regression



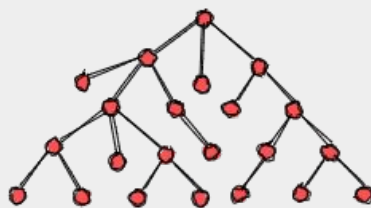
Logistic Regression



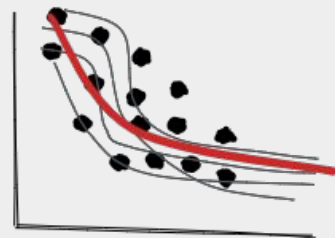
Support Vector Machines (SVM)



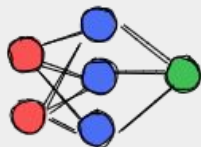
Decision Trees



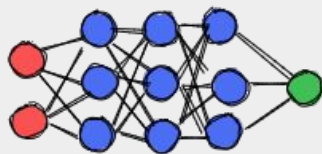
Ensemble



Neural Networks



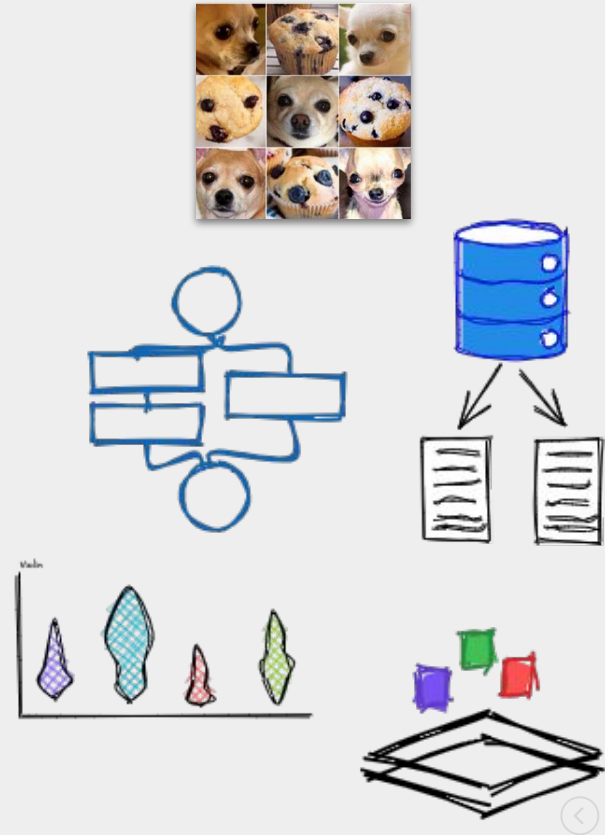
Deep Learning



Bosting

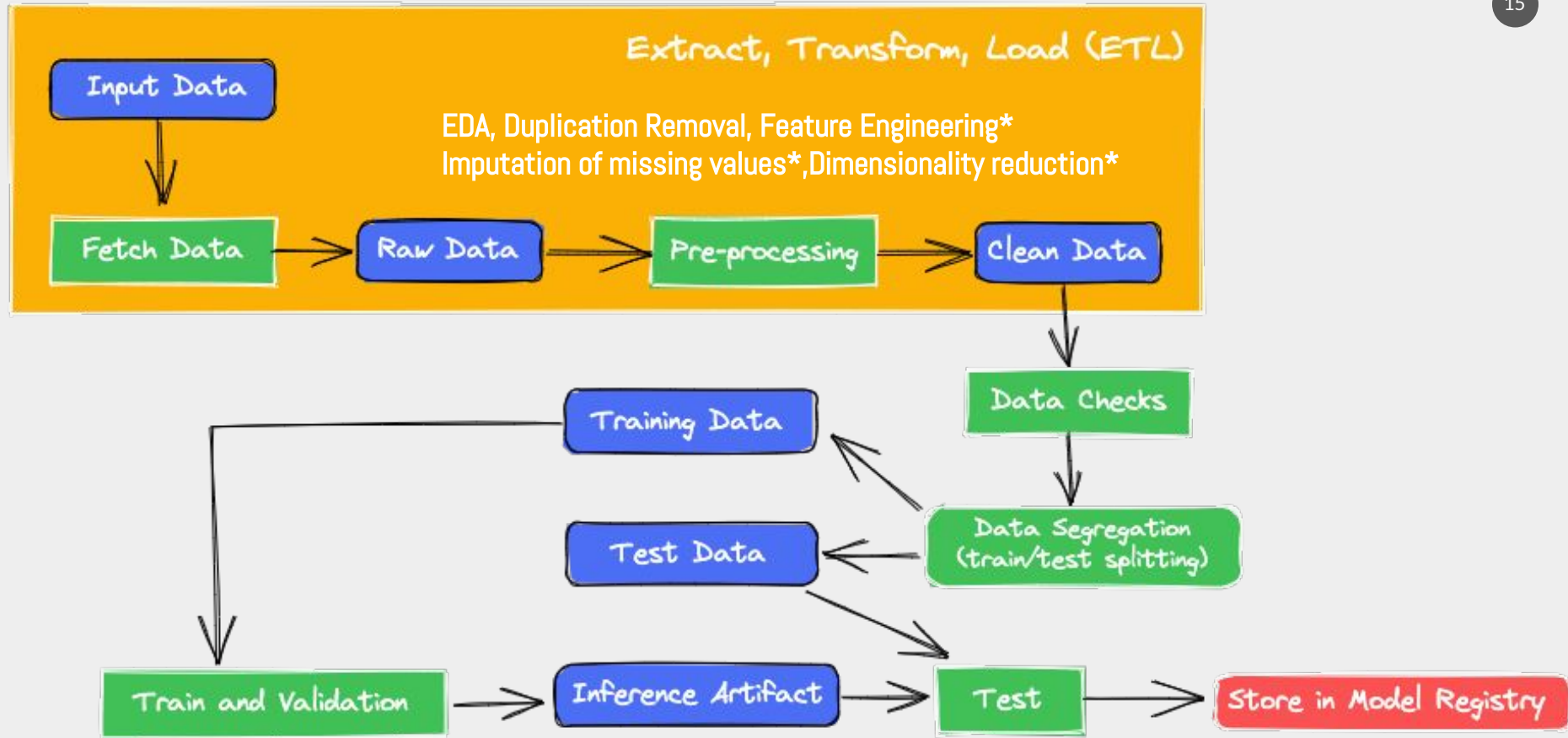


Main Challenges Of Machine Learning

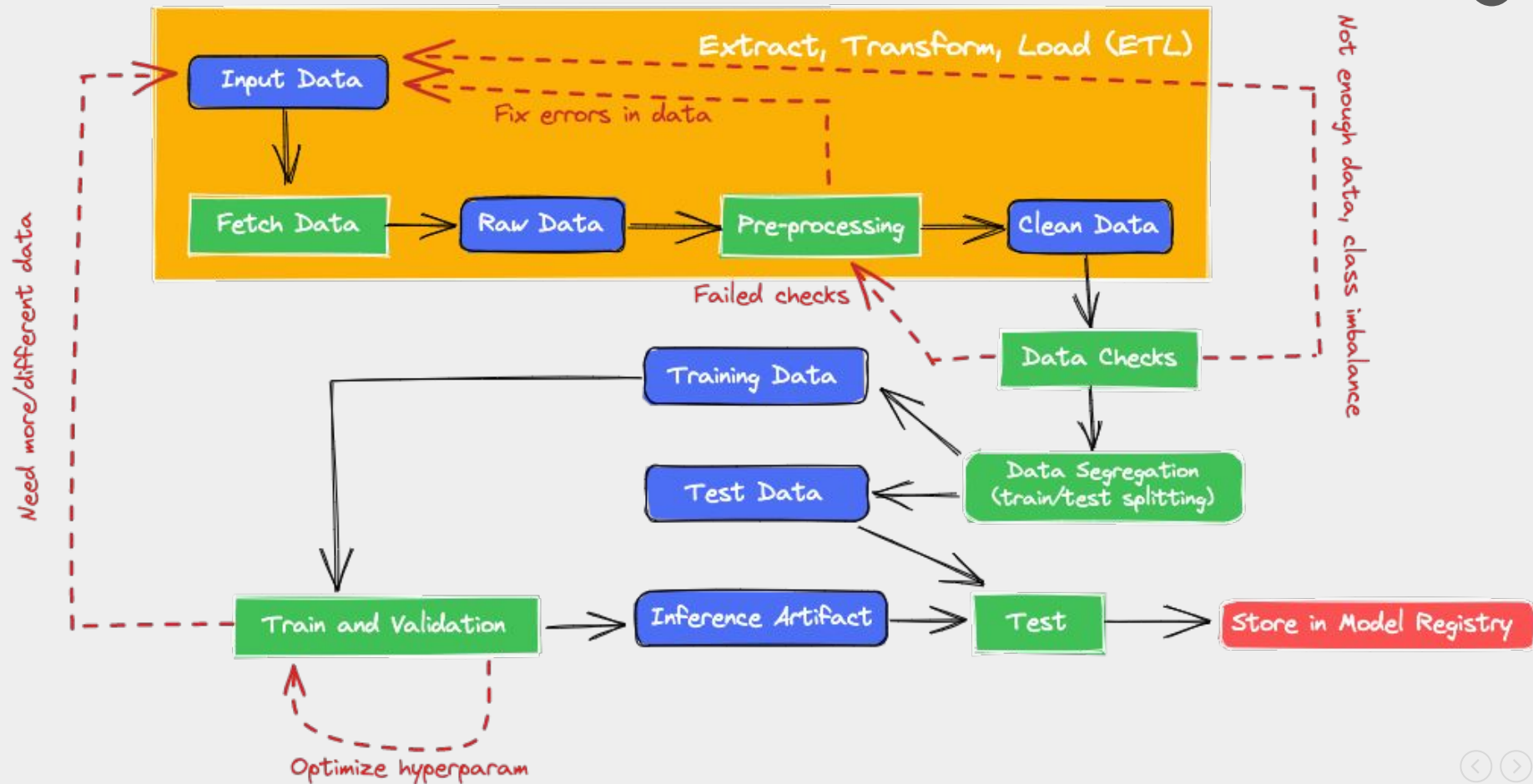


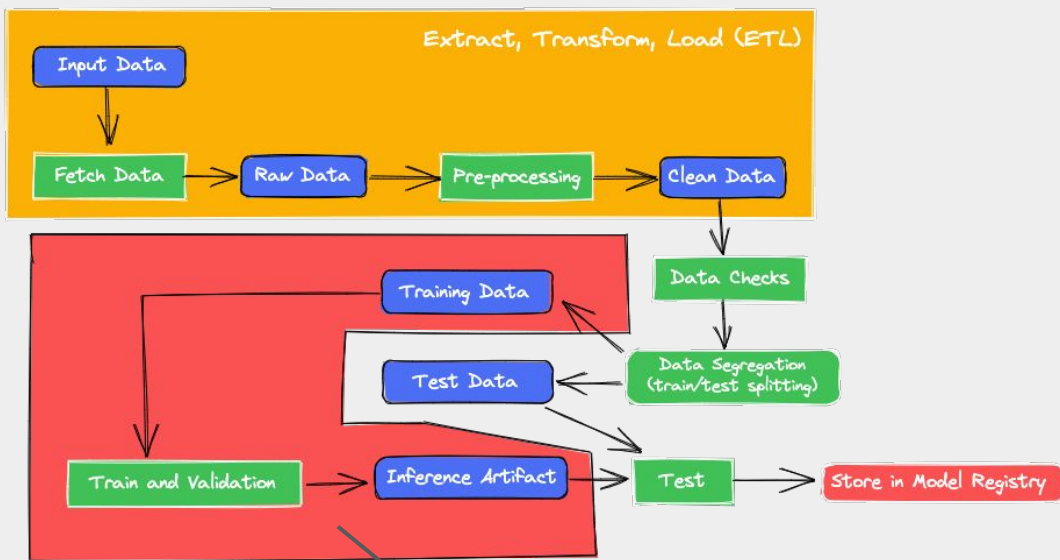
Titanic: Machine Learning from Disaster

Survived	Pclass	Name	Sex	Age	Ticket	Cabin	Embarked
0	3	Braund, Mr. Owen	Male	22	A/5 21171	NaN	S
1	1	Cummings, Mrs John	Female	38	PC 17599	C85	C
1	3	Heikkinen, Ms Laina	Female	26	STON/O2	NaN	S
1	1	Futrelle, Mrs Jacques	Female	35	113803	C123	S
0	3	Allen, Mr. William	Male	35	373450	NaN	S

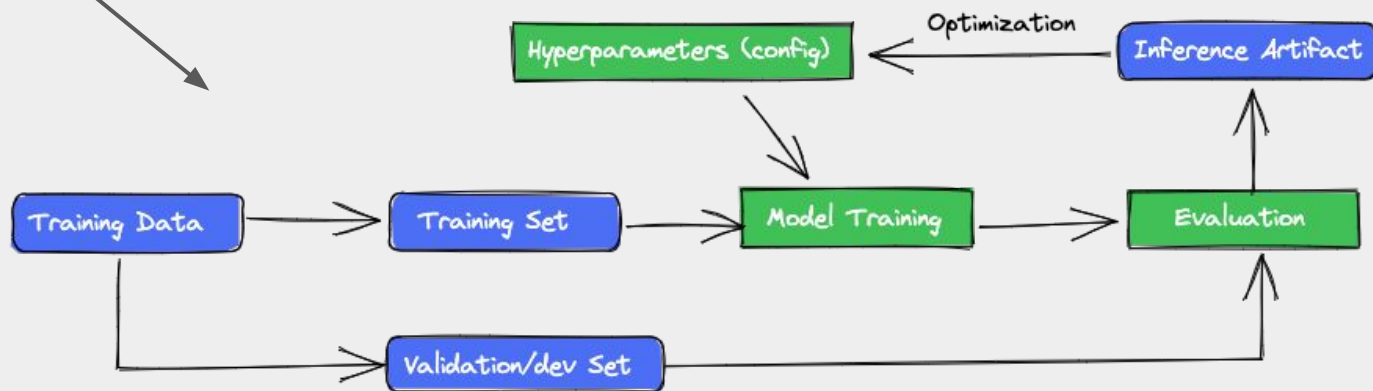


Feature Store, Categorical encoding missing values imputation, Dimensionality Reduction





Train and Evaluate



Controlled Chaos



Assume you are going to iterate A LOT



Nothing is lost
You learn something with every experiment



Give yourself time within the project deadlines



Perfection is the enemy of good
Be clear on your objective and stop once you reach it



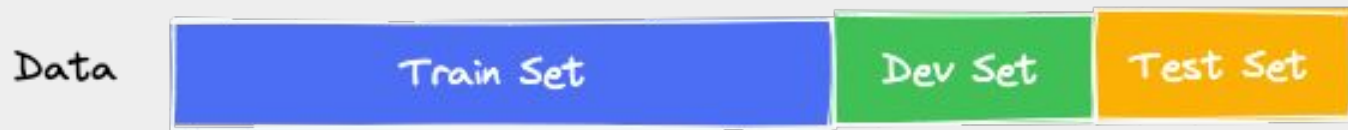
Be systematic
Normally, change one thing at the time



Nothing is fixed
data, code and hyperparameters

Train - Dev - Test Sets

Making good choices in how you set up your training, development, and test sets can make a huge difference in helping you quickly find a good high performance neural network.



Previous ML era

- 70/30
- 60/20/20

Big Data era

- 98/1/1
- 99.5/0.25/0.25
- 99.5/0.4/0.1

Holdout
Cross-Validation
Validation
Development

Mismatched train/test distribution

Scenario: say you are building a cat-image classifier application that determines if an image is of a cat or not. The application is intended for users in rural areas who can take pictures of animals by their mobile devices for the application to classify the animals for them.

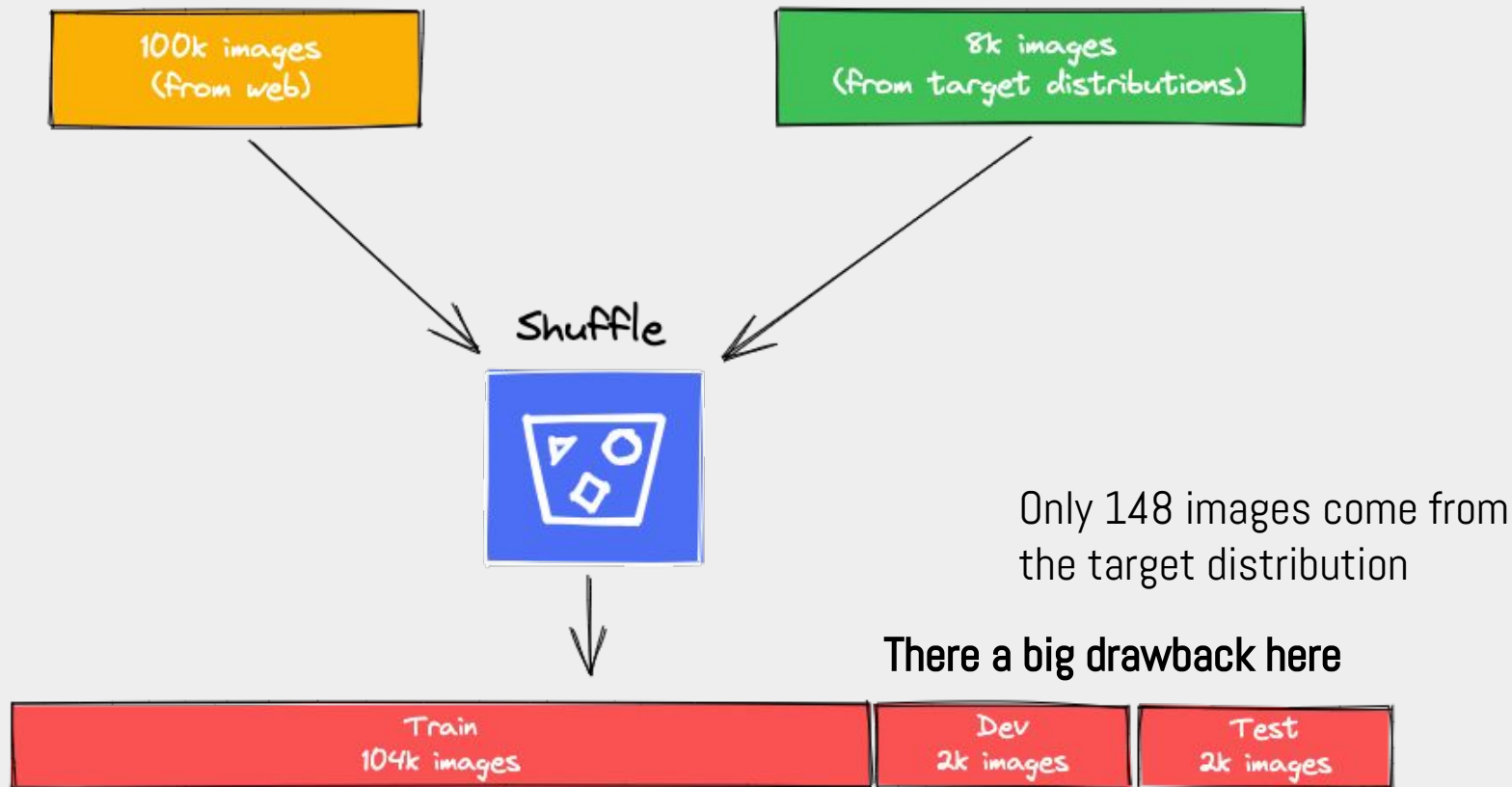


Scraped from Web Pages
100k images

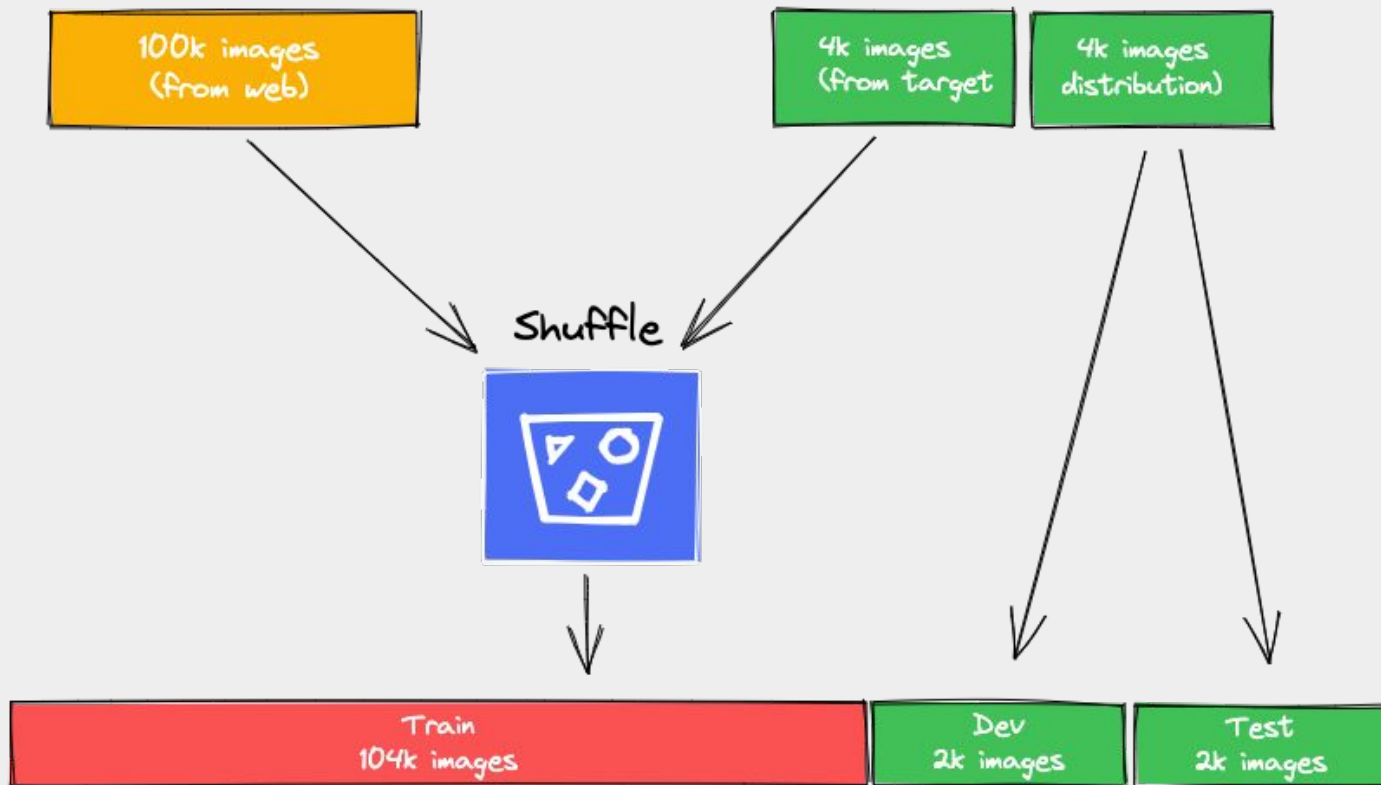


Collected from Mobile Devices
<<target distribution>>
8k images

A possible option: shuffling the data



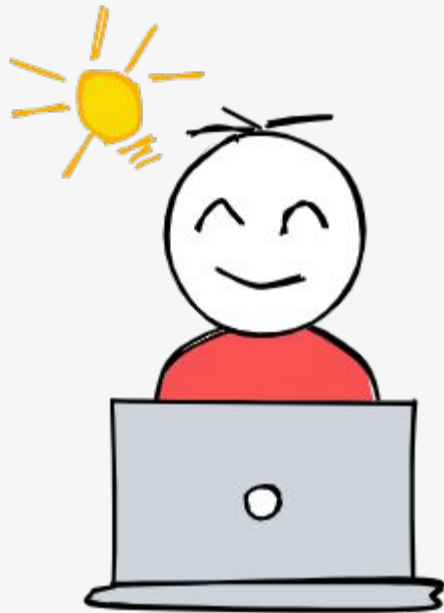
A better option



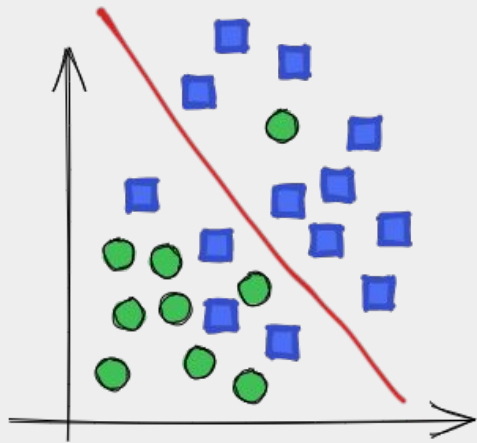
Rule of the thumb

>> make sure that the dev and test sets come from the same distribution

Not having a test set might be okay. (Only dev set)

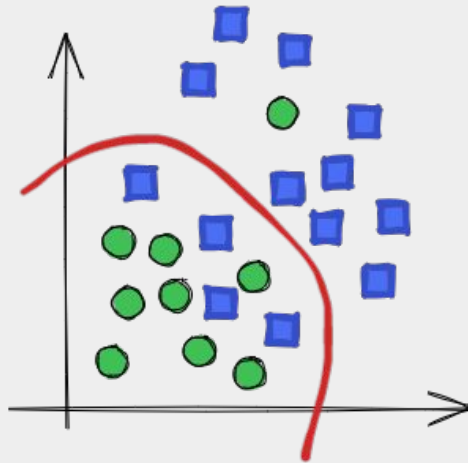


Bias vs Variance

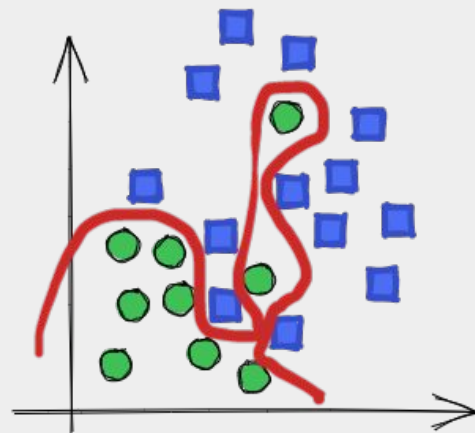


High Bias

Underfitting



Just Right



High Variance

Overfitting

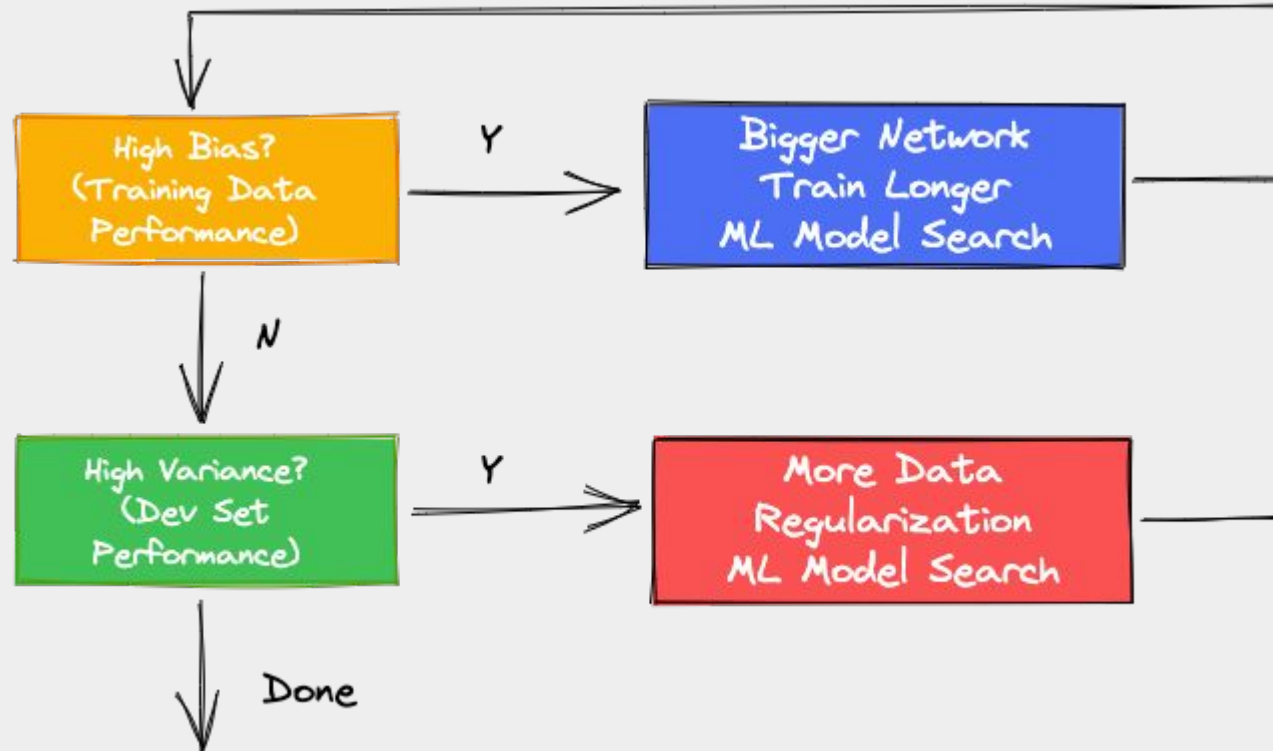
Bias vs Variance

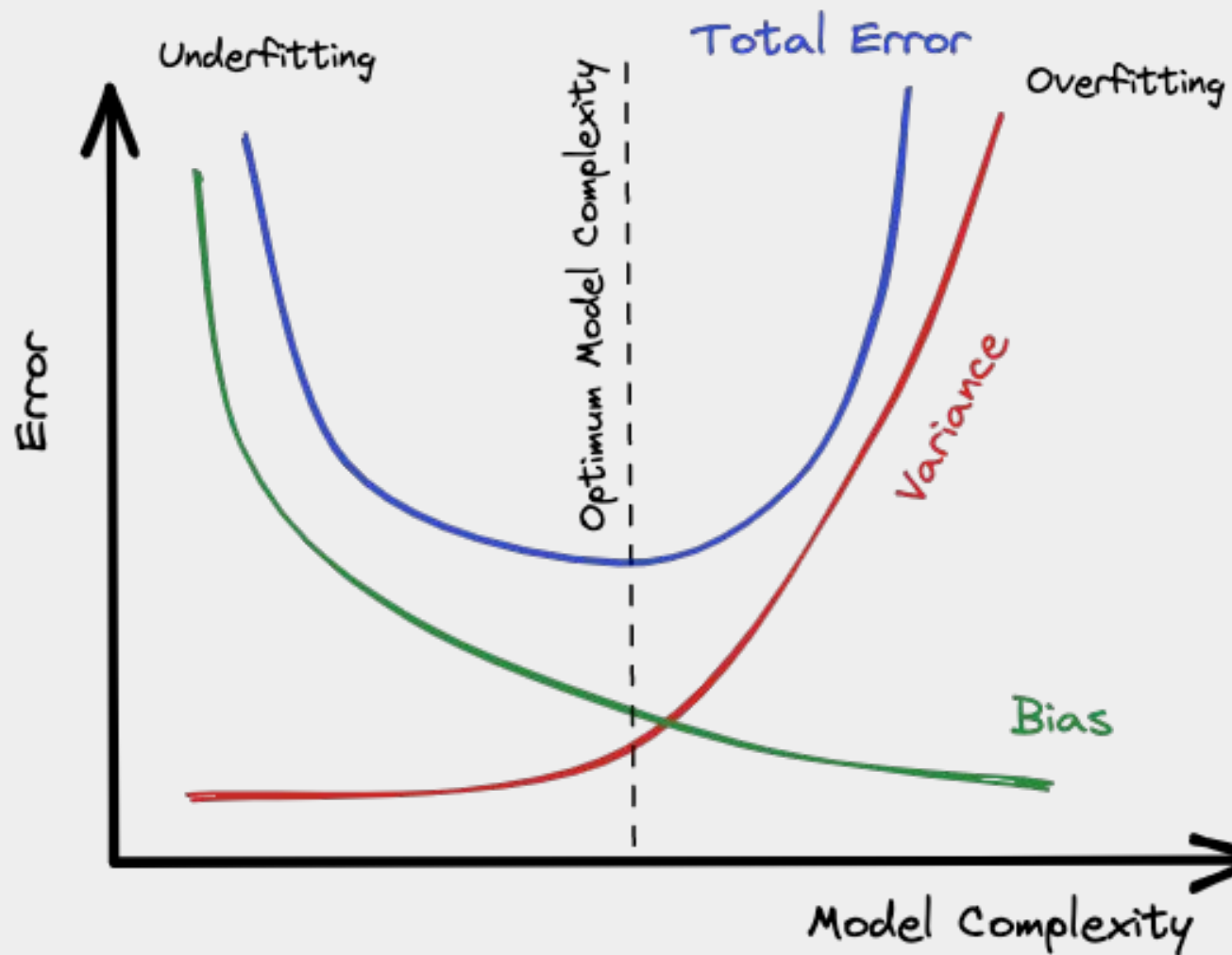
Cat Classification



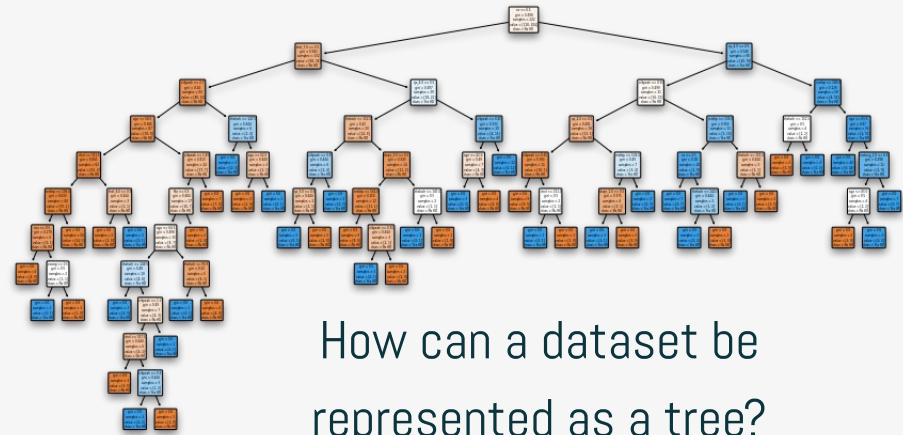
	Scenario #01	Scenario #02	Scenario #03	Scenario #04
Train Set Error	1%	15%	15%	0.5%
Dev Set Error	16%	16%	30%	1%
	Low Bias High Variance	High Bias Low Variance	High Bias High Variance	Low Bias Low Variance

Basic Recipe for Machine Learning





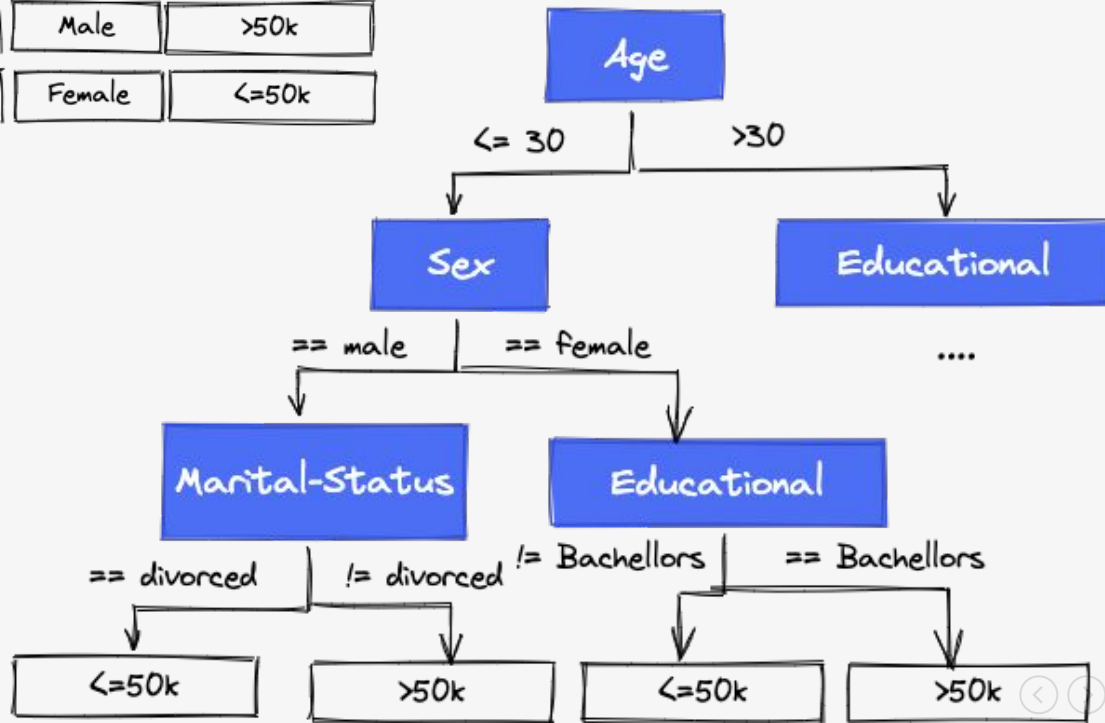
Decision Trees



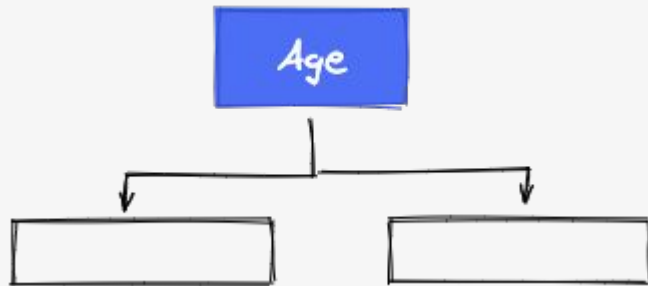
How can a dataset be represented as a tree?

Age	Marital-Status	Educational	Sex	High-Income
28	Never-Married	Bachelors	Female	$\leq 50k$
46	Never-Married	Assoc-acdm	Female	$\leq 50k$
35	Married-civ-spouse	Some-college	Male	$\leq 50k$
27	Married-civ-spouse	Bachelors	Male	$> 50k$
59	Divorced	Some-college	Female	$\leq 50k$

Decision Tree (classification)



How can we split the tree?



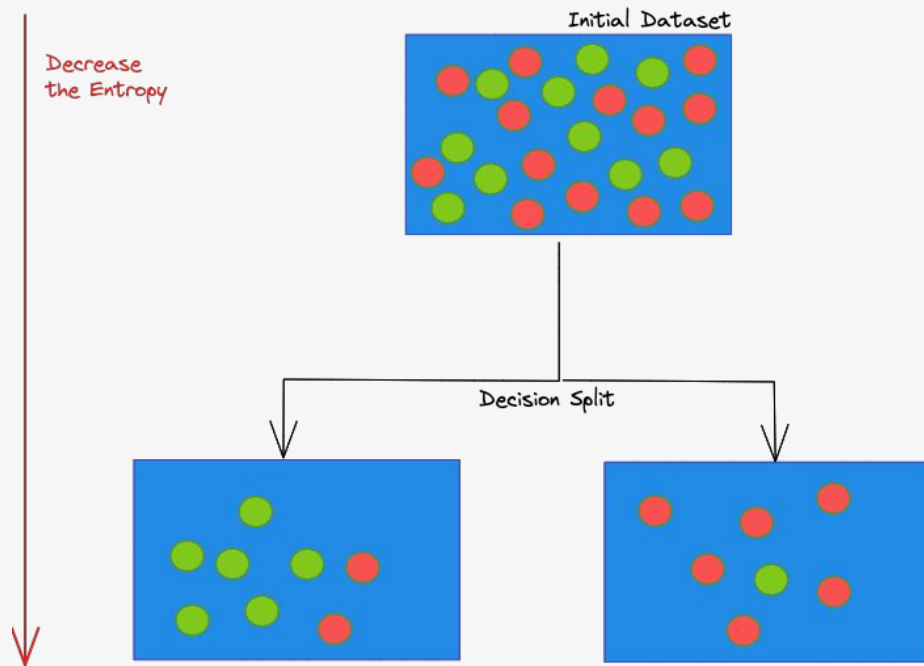
Algorithm used in Decision Trees

1. ID3 (Entropy)
2. Gini Index
3. Chi-Square
4. Reduction in Variance
 - a. C4.5, pruning
5. ...



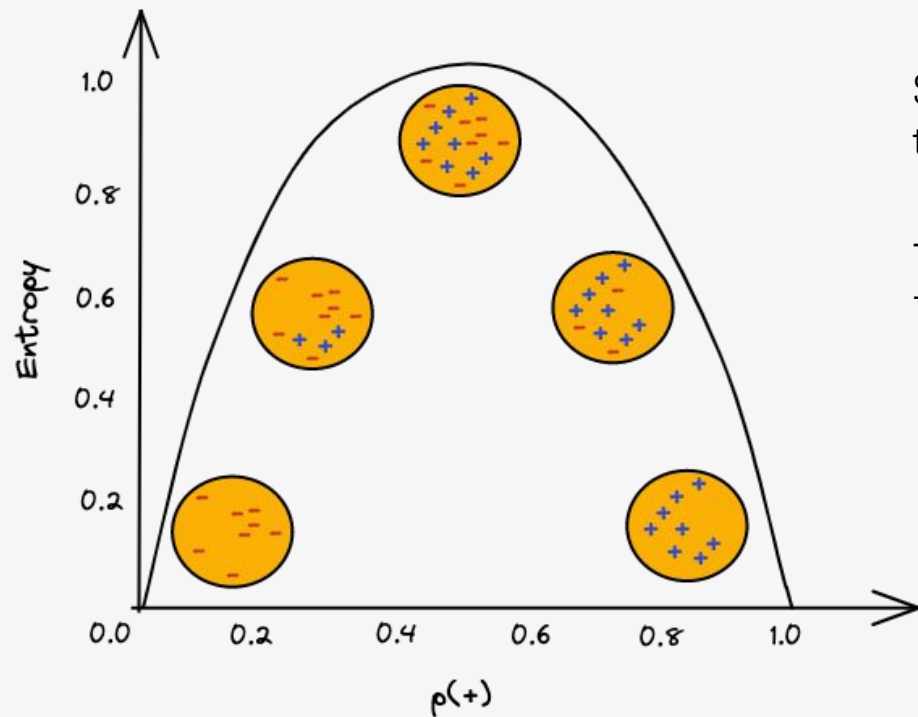
Entropy is an indicator of how messy your data is.

Why Entropy in Decision Trees?



- The goal is to tidy the data.
- You try to separate your data and group the samples together in the classes they belong to.
- You maximize the purity of the groups as much as possible each time you create a new node of the tree
- Of course, at the end of the tree, you want to have a clear answer.

Mathematical Definition of Entropy



Suppose a set of N items, these items fall into two categories:

+ gain $> 50k$ (k)

- gain $\leq 50k$ (m)

$$p = \frac{k}{N}, q = \frac{m}{N}$$

$$Entropy = -p \log p - q \log q$$

Generalization

Feature X

$$E(X) = - \sum_{i=1}^c P(x_i) \log_b P(x_i)$$

$P(x_i)$ is the fraction of examples in a given class i

$\leq 50k.$	17288
$> 50k.$	5487

```
from scipy.stats import entropy
entropy(df_train.high_income.value_counts(), base=2)
0.7965702796015677
```

Entropy using the frequency table of two attributes

		High Income		
		<= 50k	> 50k	
Age	<=37	7206	3883	11089 (48%)
	>37	10082	1604	11686 (52%)

```
cross = pd.crosstab(
    df_train.age <= df_train.age.median(),
    df_train.high_income)
```

$$E(T | X) = \sum_{c \in X} \frac{|X_c|}{|X|} E(T | X_c)$$

```
0.486894 * entropy(cross.iloc[0], base=2) \
+ 0.513106 * entropy(cross.iloc[1], base=2)
0.7509335429830957
```

Information Gain

$$IG(T, X) = E(T) - E(T|X)$$

Information Gain from X on T

The information gain is based on the **decrease in entropy after a dataset is split** on an attribute.

Constructing a decision tree is all about finding attribute that returns the **highest information gain** (i.e., the most homogeneous branches).

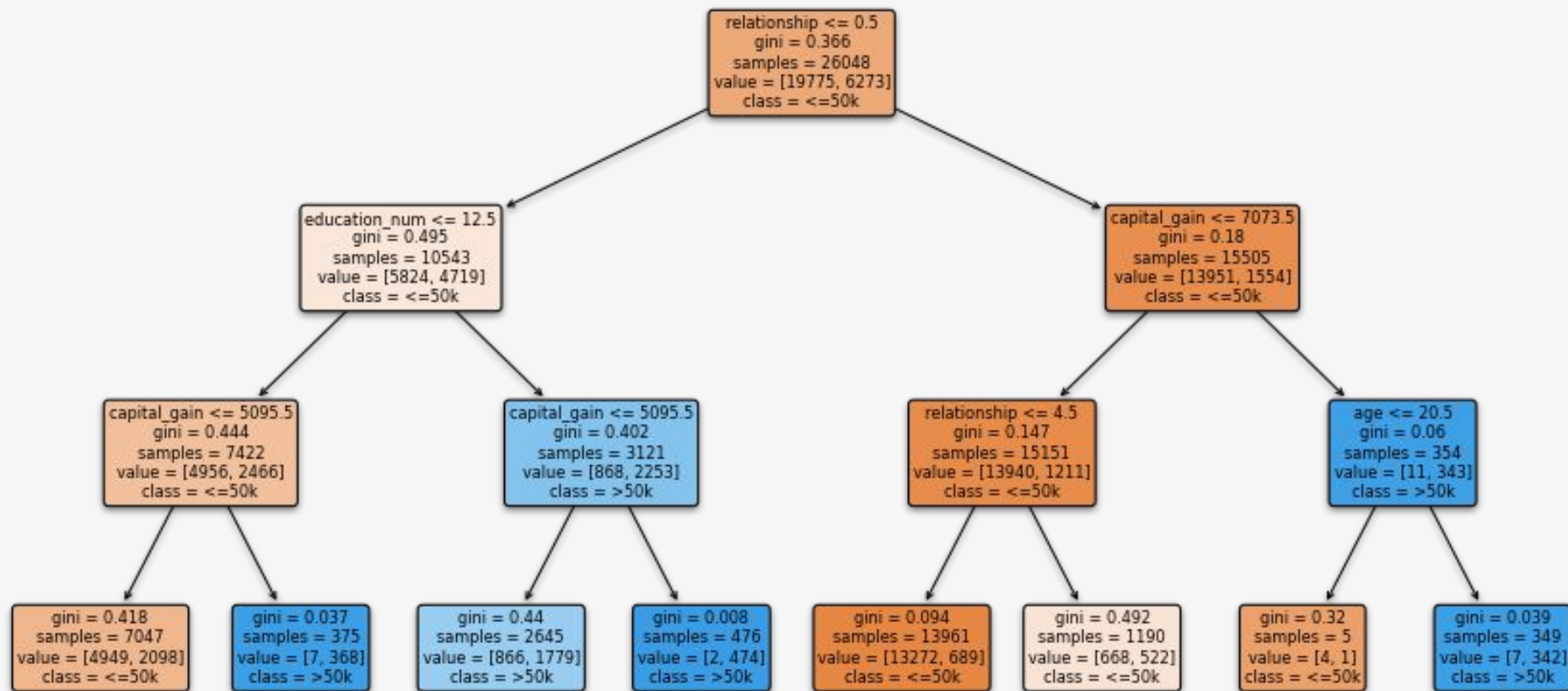
$$\text{Gini}(x) = 1 - \sum_{i=1}^c P(x_i)^2$$

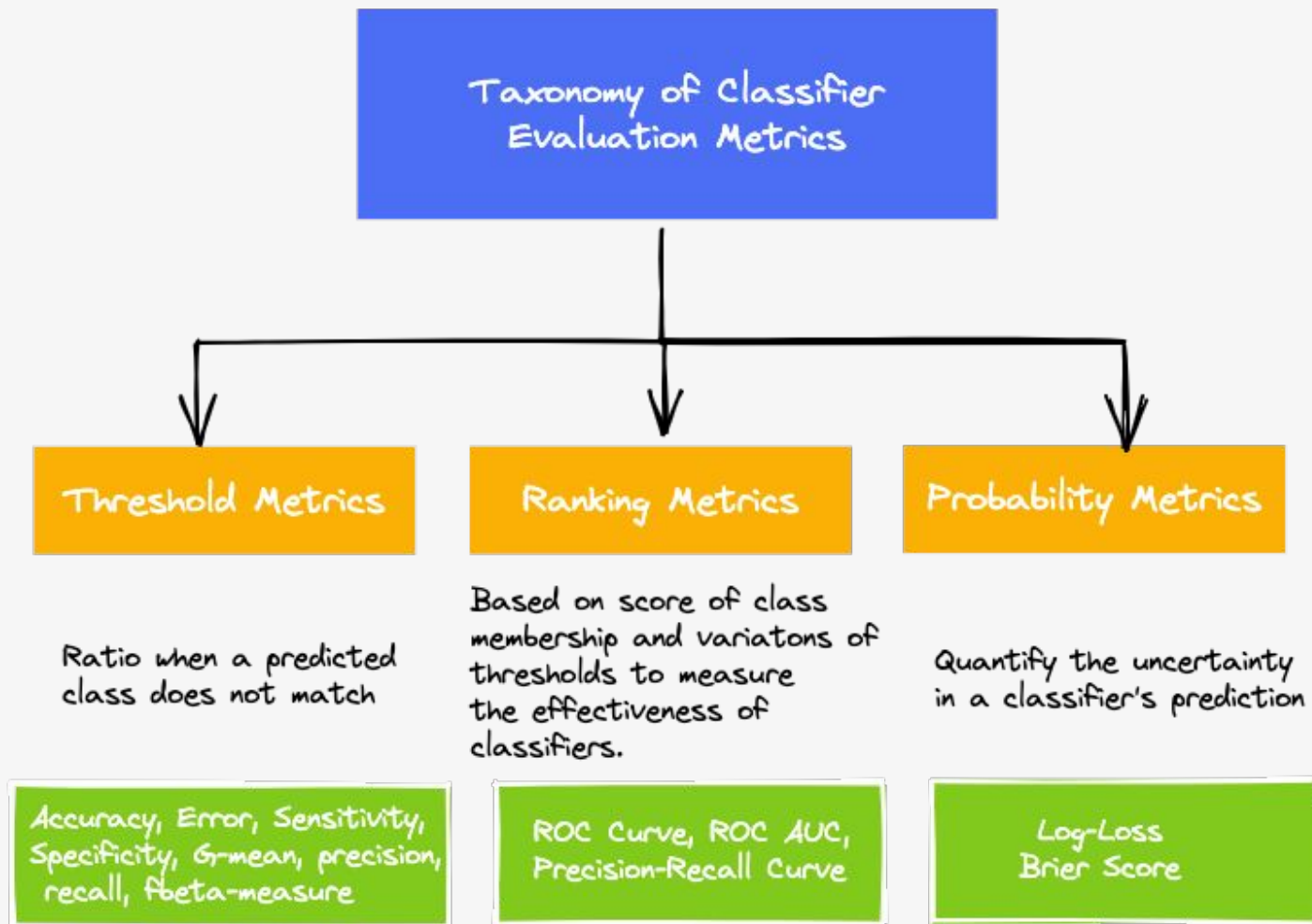
$$\text{Entropy}(x) = - \sum_{i=1}^c P(x_i) \log_b P(x_i)$$

Gini index or Entropy is the criterion for calculating **Information Gain**. Both of them are measures of impurity of a node.

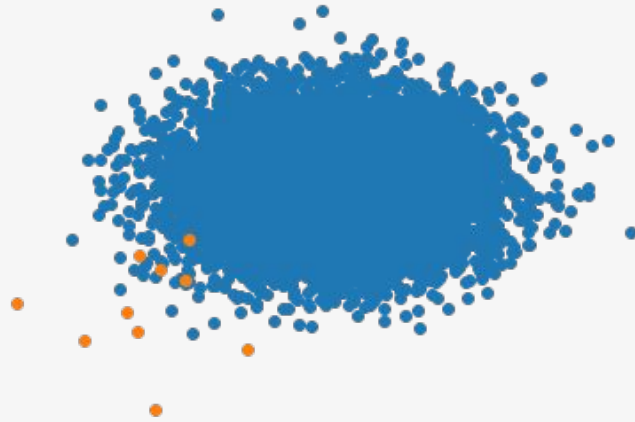
```
from sklearn.tree import plot_tree
```

38





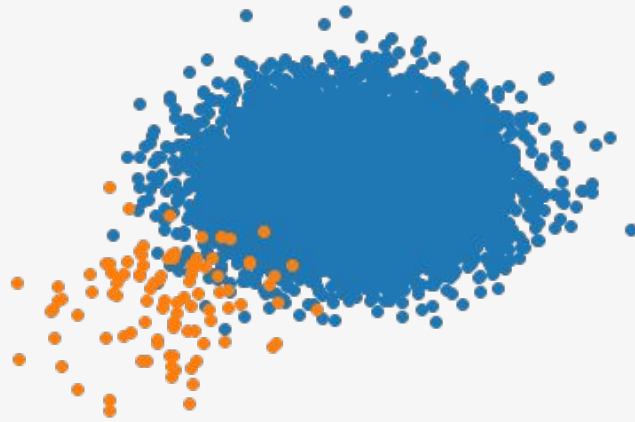
Imbalanced 1:1000



Imbalanced 1:10



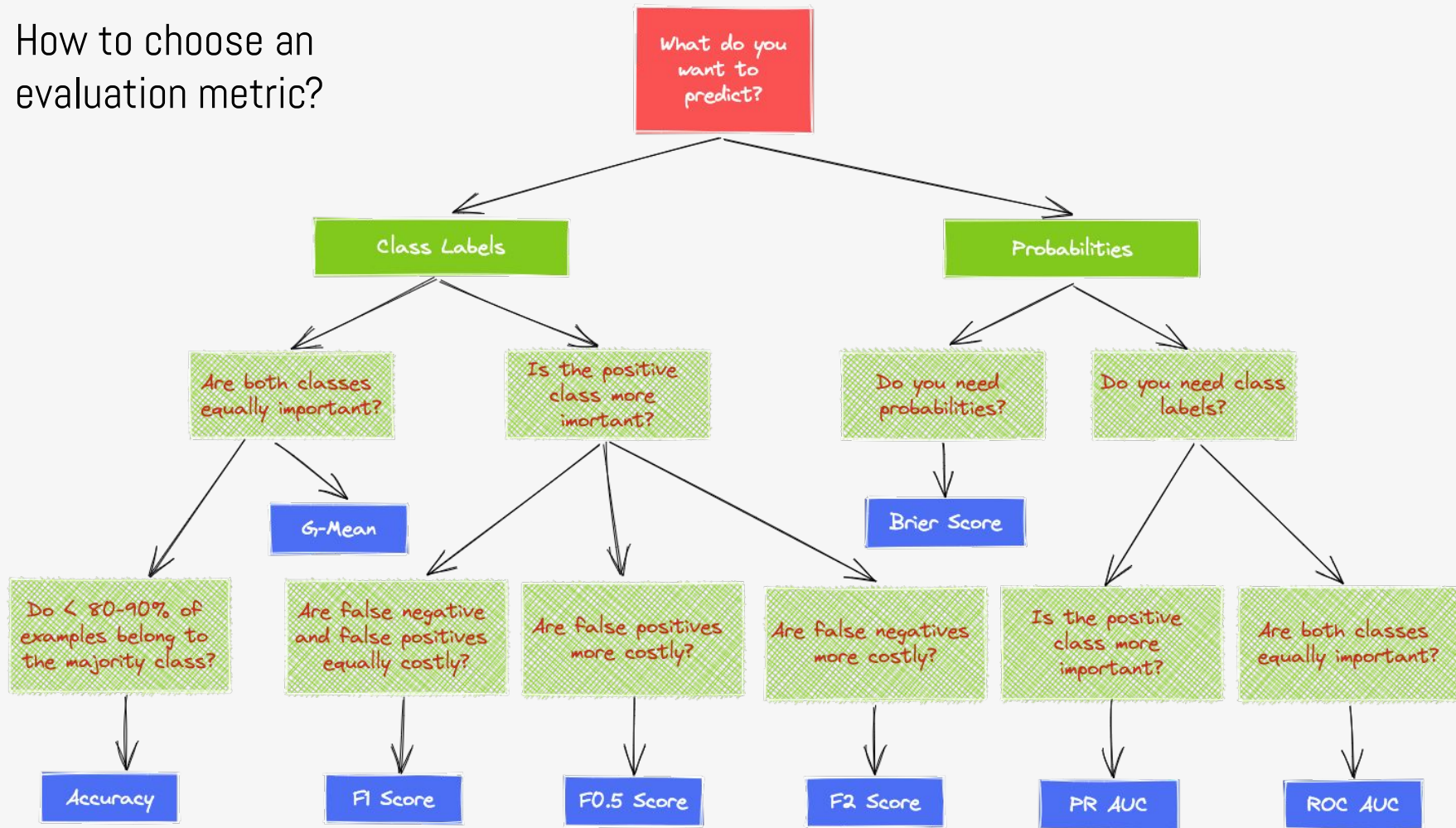
Imbalanced 1:100



Imbalanced 1:2











How to choose an evaluation metric?



Confusion Matrix

Expected









		Positive class (1)		Negative class (0)	
Predicted	Negative class (0)	True Positive (TP)		False Positive (FP)	
		Predicted	Expected	Predicted	Expected
					
Predicted	Positive class (1)	False Negative (FN)		True Negative (TN)	
		Predicted	Expected	Predicted	Expected
					

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{Error} = 1 - \text{Accuracy}$$

Confusion Matrix

Expected

		Positive class (1)		Negative class (0)	
Predicted	Negative class (0)	True Positive (TP)		False Positive (FP)	
		Predicted	Expected	Predicted	Expected
					
Predicted	Positive class (1)	False Negative (FN)		True Negative (TN)	
		Predicted	Expected	Predicted	Expected
					

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{FP + TN}$$

$$\text{G-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$$

Confusion Matrix

Expected

Positive class (1)

Negative class (0)

Predicted
Positive class (1)
Negative class (0)

True Positive (TP)

Predicted



Expected



False Positive (FP)

Predicted



Expected



False Negative (FN)

Predicted



Expected



True Negative (TN)

Predicted



Expected



$$\text{Precision} = \frac{TP}{TP + FP}$$

(positive predictive value - PPV)

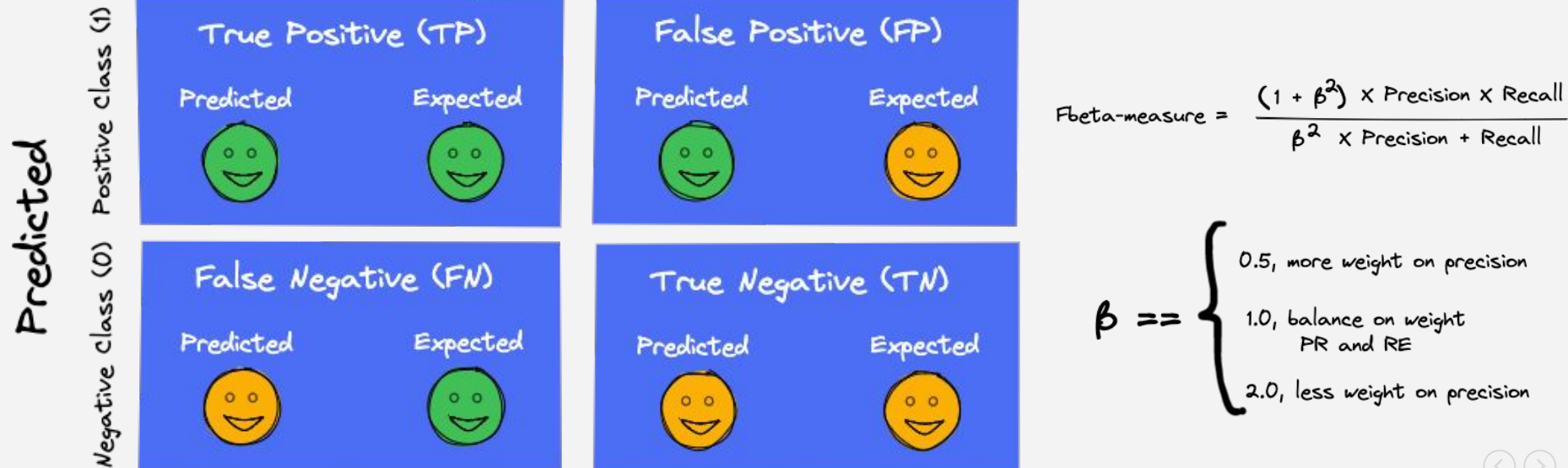
$$\text{Precision} = \frac{TN}{TN + FN}$$

(negative predictive value - NPV)

$$\text{Recall} = \frac{TP}{TP + FN}$$

Confusion Matrix

Expected

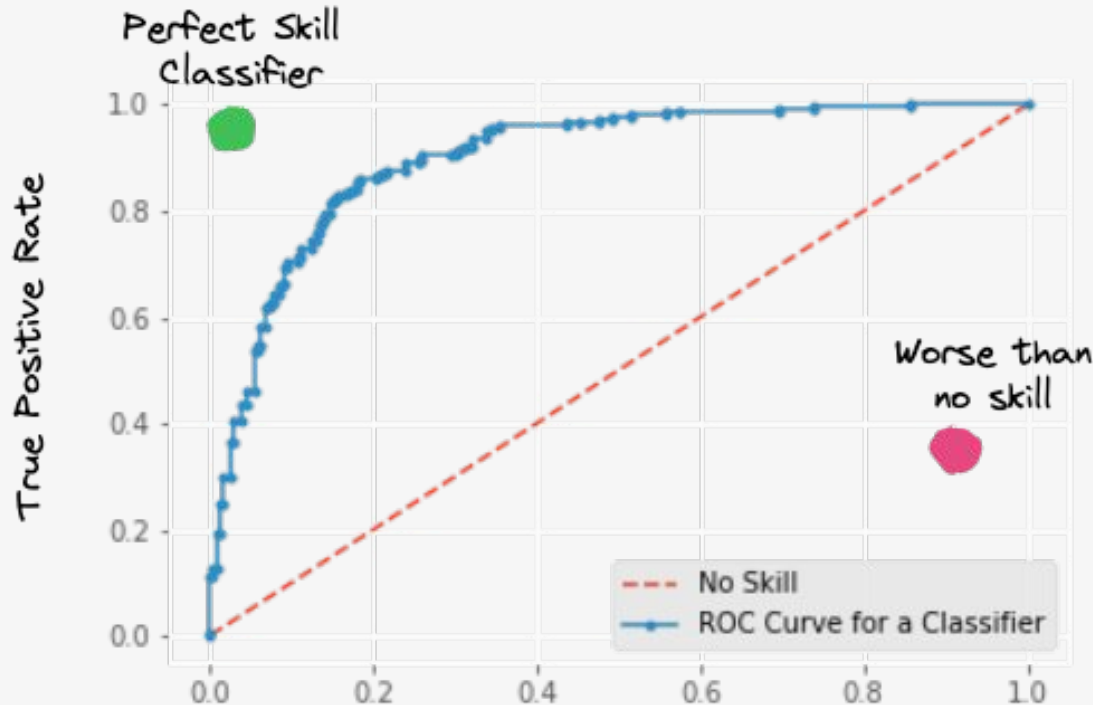


Rank metrics are more concerned with evaluating classifiers based on **how effective** they are at separating classes.

These metrics require that a **classifier predicts a score** or a probability of class membership. From this score, **different thresholds** can be applied to **test the effectiveness of classifiers**. Those models that maintain a good score across a range of thresholds will have good class separation and will be ranked higher.

Receiver Operating Characteristic (ROC)

$$TPR = \frac{TP}{TP + FN}$$



Expected

Positive class (1)

Negative class (0)

Predicted

Positive class (1)	True Positive (TP)	
	Predicted	Expected
Negative class (0)	False Positive (FP)	
	Predicted	Expected
Positive class (1)	False Negative (FN)	
	Predicted	Expected
Negative class (0)	True Negative (TN)	
	Predicted	Expected

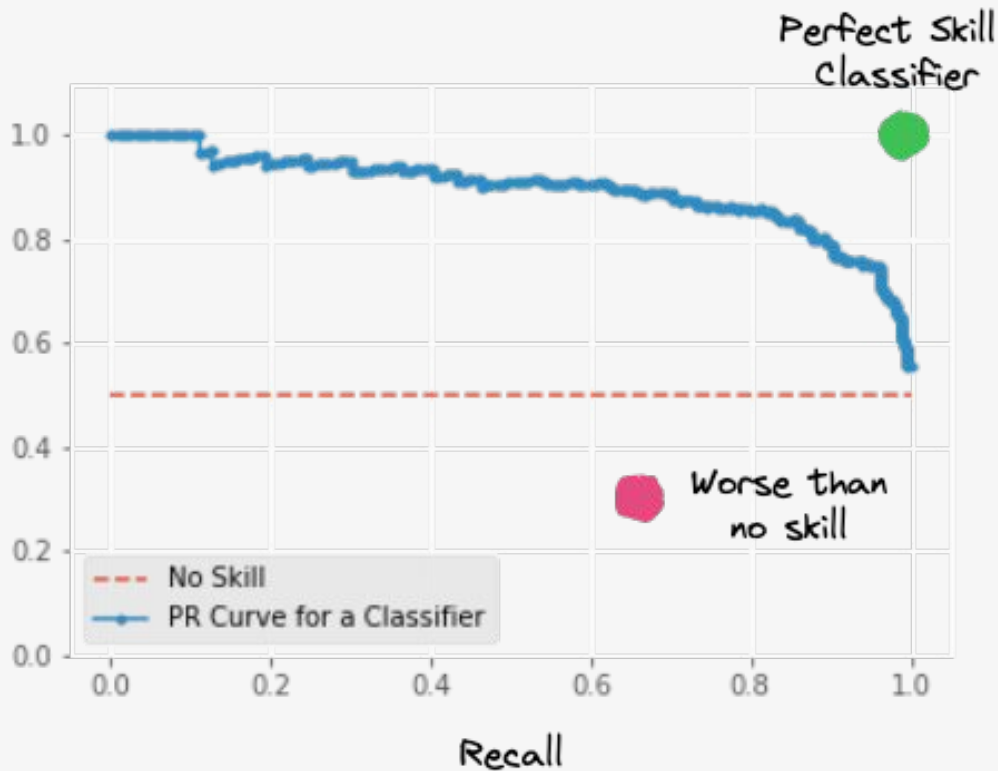
False Positive Rate

$$FPR = \frac{FP}{FP + TN}$$

Precision-Recall (PR) Curve

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision



Expected

Positive class (1)

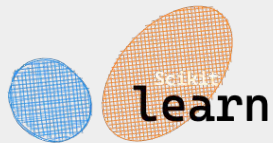
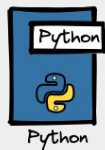
Negative class (0)

Predicted	True Positive (TP)		False Positive (FP)	
	Predicted	Expected	Predicted	Expected
Negative class (0)	False Negative (FN)		True Negative (TN)	
	Predicted	Expected	Predicted	Expected

$$\text{Recall} = \frac{TP}{TP + FN}$$



Hands ON



Optional





Adult Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Predict whether income exceeds \$50K/yr based on census data. Also known as "Census Income" dataset.



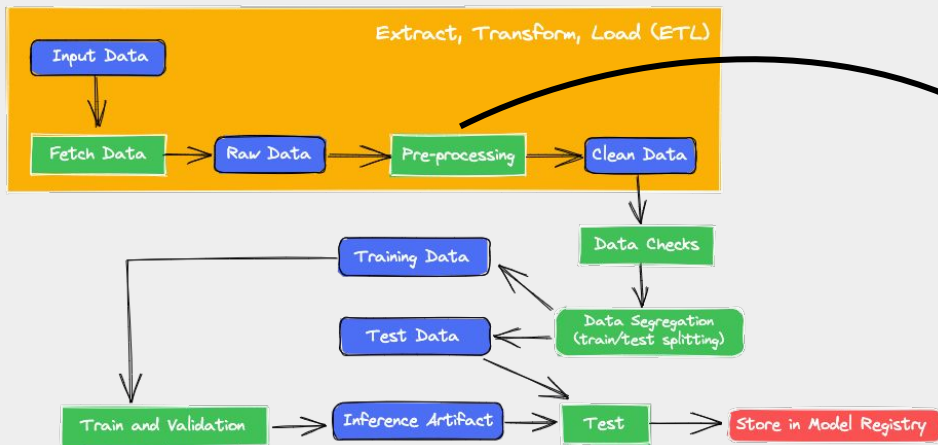
Data Set Characteristics:	Multivariate	Number of Instances:	48842	Area:	Social
Attribute Characteristics:	Categorical, Integer	Number of Attributes:	14	Date Donated	1996-05-01
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	2437279

Source:

Donor:

Ronny Kohavi and Barry Becker
Data Mining and Visualization
Silicon Graphics.
e-mail: ronnyk '@' live.com for questions.

Exploratory Data Analysis



dataprep

D-TALE

LUX

Sweetviz

PANDAS
PROFILING

bokeh

plotly

Altair

Interaction Visualization

matplotlib

Static Visualization