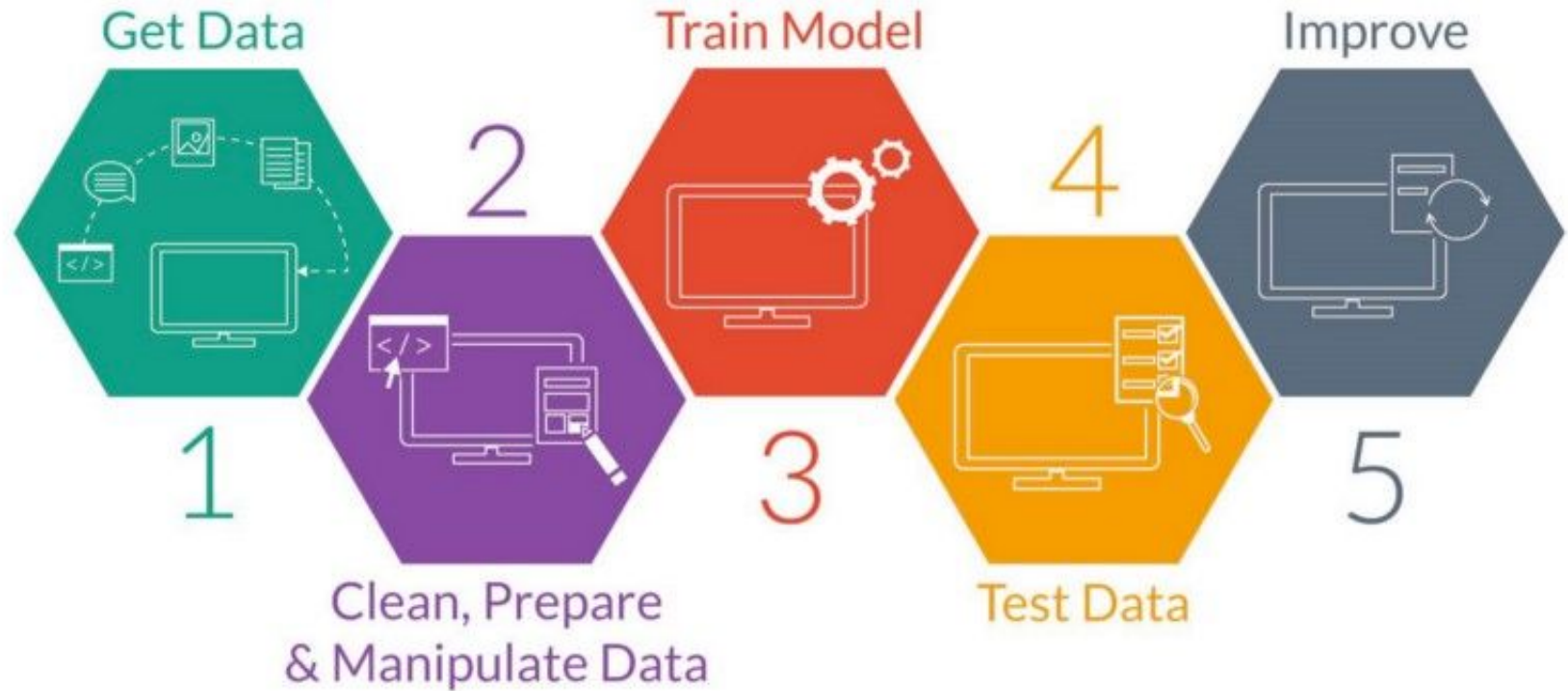


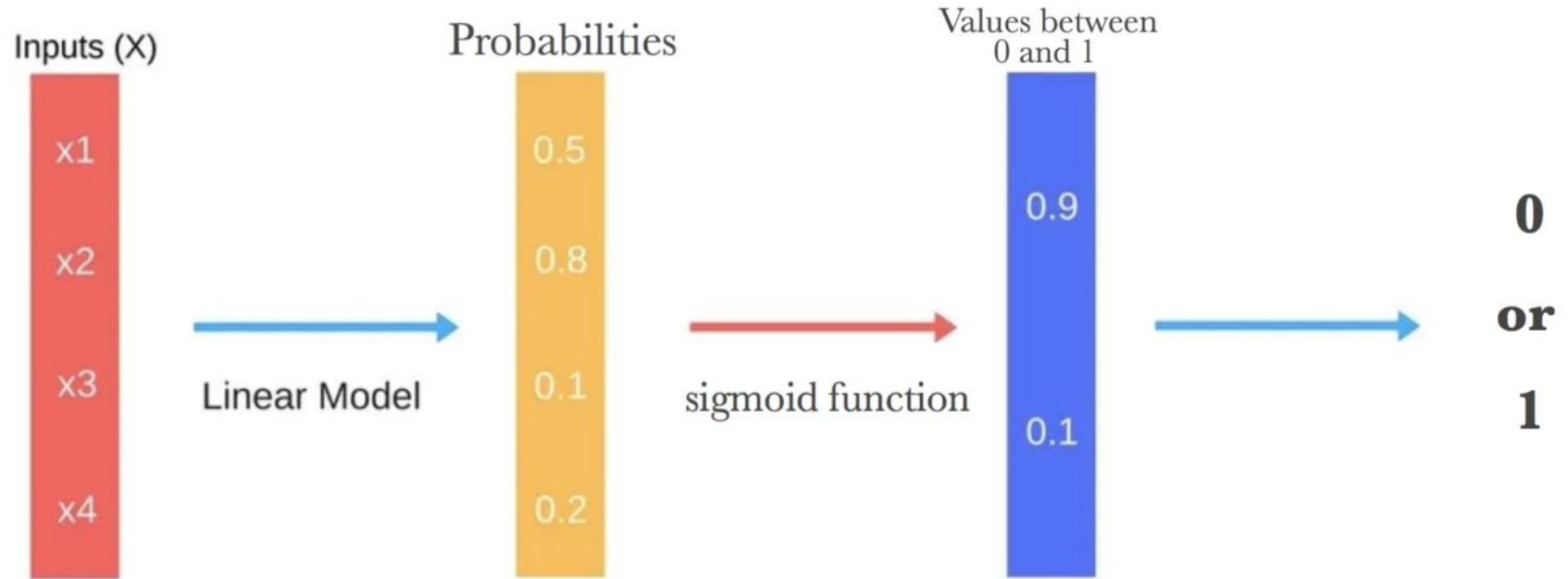
A Brief Tour of Classification Algorithms

Vanessa Rivera Quinones

ML Pipeline



Logistic Regression

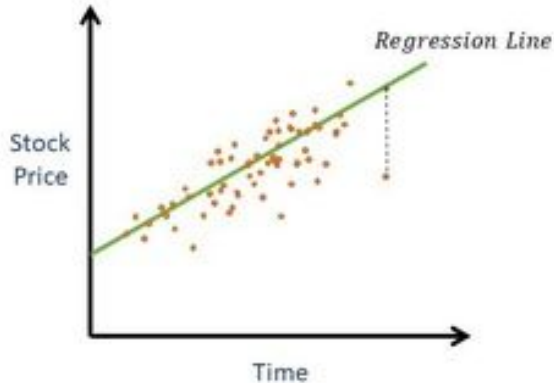


Logistic Regression

Linear Regression Vs Logistic Regression

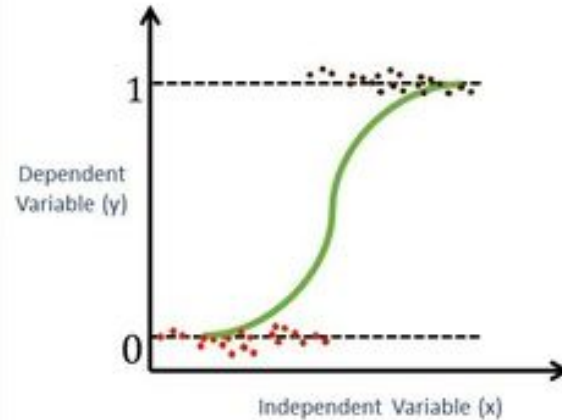
Linear Regression

- Aim is to predict continuous valued output.
- Output value can be any possible integer number.



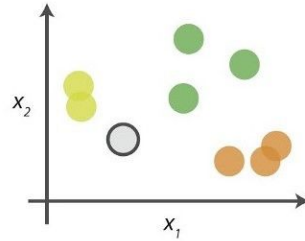
Logistic Regression

- Aim is to predict the label for input data.
- Output is categorical (Binary) i.e. 0/1, True/False, etc.



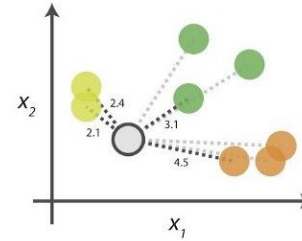
KNN

0. Look at the data











Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances









Start by calculating the distances between the grey point and all other points.

2. Find neighbours

Point Distance		
 ... 	2.1	→ 1st NN
 ... 	2.4	→ 2nd NN
 ... 	3.1	→ 3rd NN
 ... 	4.5	→ 4th NN

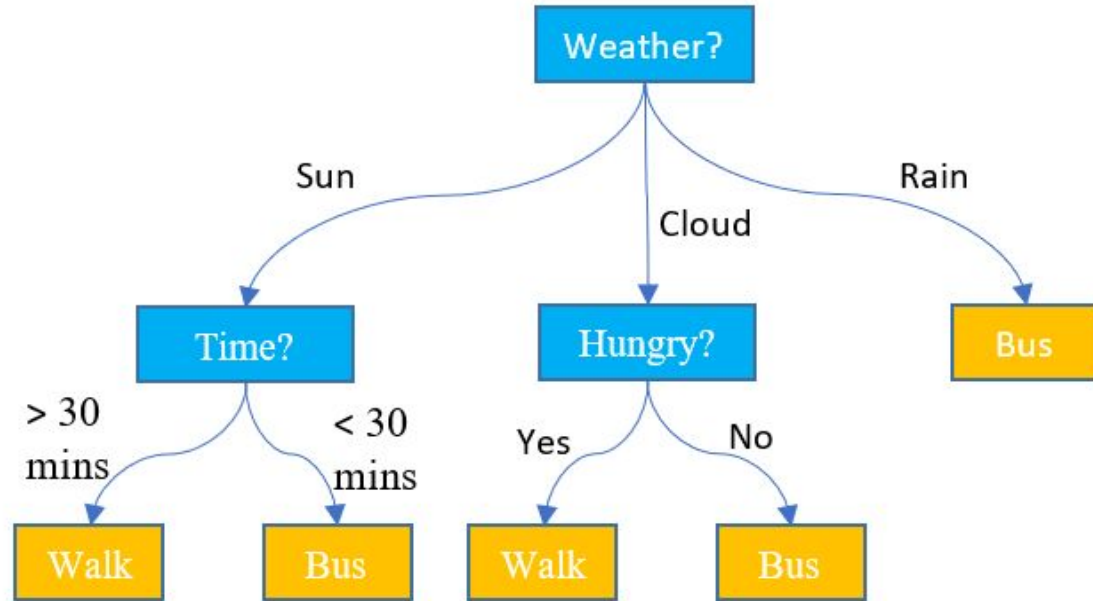
Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

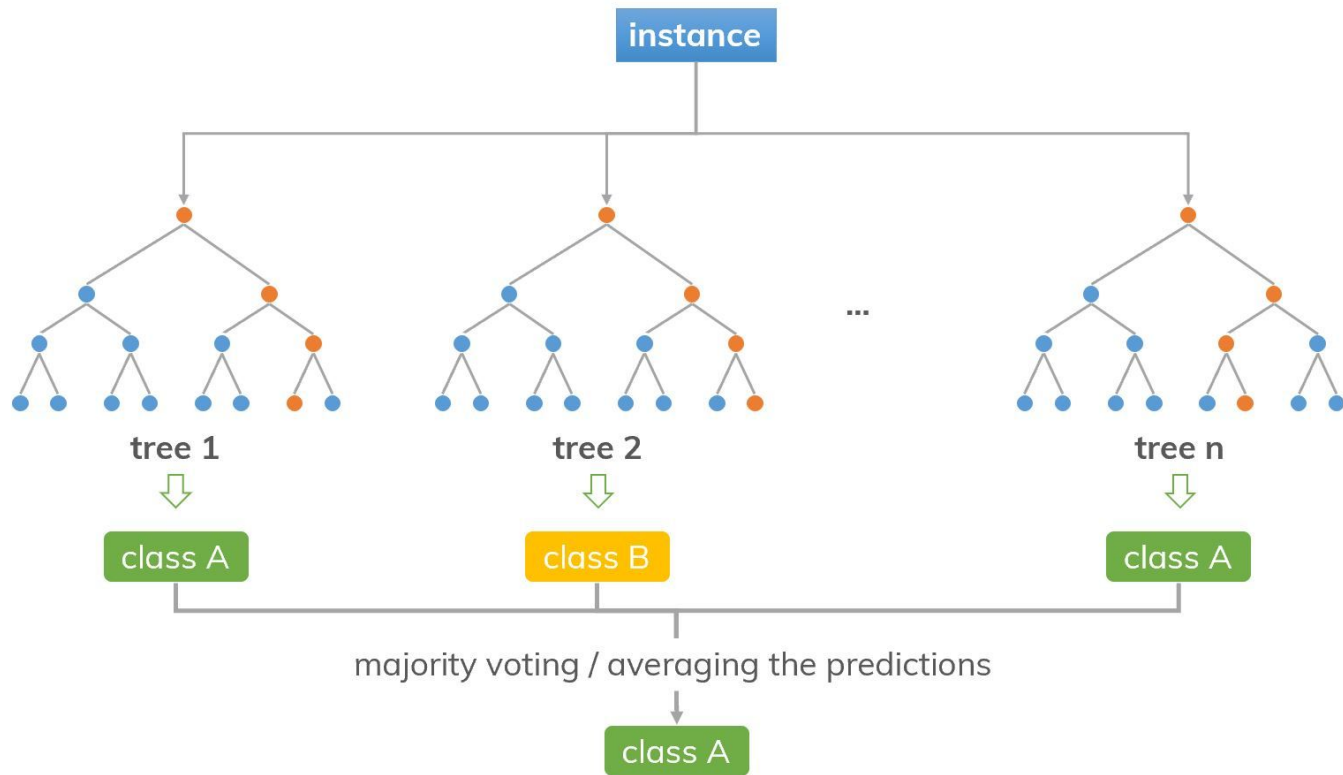
Class	# of votes	
	2	→ Class  wins the vote! Point  is therefore predicted to be of class  .
	1	
	1	

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the $k=3$ nearest neighbours.

Decision Trees



Random Forest



Metrics

Metric	Measures	In Scikit-learn
Precision	How many selected are relevant?	<code>from sklearn.metrics import precision_score</code>
Recall	How many relevant were selected?	<code>from sklearn.metrics import recall_score</code>
F1	Weighted average of precision & recall	<code>from sklearn.metrics import f1_score</code>
Confusion Matrix	True positives, true negatives, false positives, false negatives	<code>from sklearn.metrics import confusion_matrix</code>
ROC	True positive rate vs. false positive rate, as classification threshold varies	<code>from sklearn.metrics import roc</code>
AUC	Aggregate accuracy, as classification threshold varies	<code>from sklearn.metrics import auc</code>

Metrics

	Predicted label class 1	Predicted label class 2
True label class 1	correct true positive for class 1	wrong false positive for class 2
True label class 2	wrong false positive for class 1	correct true positive for class 2

$$\text{accuracy} = \frac{\text{orange} + \text{blue}}{\text{orange} + \text{yellow} + \text{blue} + \text{green}}$$

class 1 precision	$= \frac{\text{orange}}{\text{orange} + \text{yellow}}$	class 1 recall	$= \frac{\text{orange}}{\text{orange} + \text{green}}$
class 2 precision	$= \frac{\text{blue}}{\text{blue} + \text{green}}$	class 2 recall	$= \frac{\text{blue}}{\text{blue} + \text{yellow}}$

Metrics in Context

Bad Loan = 1

Good Loan = 0



Cost of **FN** > Cost of **FP**

Accuracy: Out of the total prediction made, how many did we predict correctly?

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

$$\text{Accuracy} = (559+22)/(559+22+33+0) = 95\%$$

Actual

Predict	Actual	
	Bad Loan (1)	Good Loan (0)
Bad Loan (1)	✓ TP - 559 👍	✗ FP - 0 👍
Good Loan (0)	✗ FN - 33 👎	✓ TN - 22 👎

Precision: Out of the loan that is predicted as a bad loan, how many did we classify correctly?

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

$$\text{Precision} = 559/(559+0) = 100\%$$

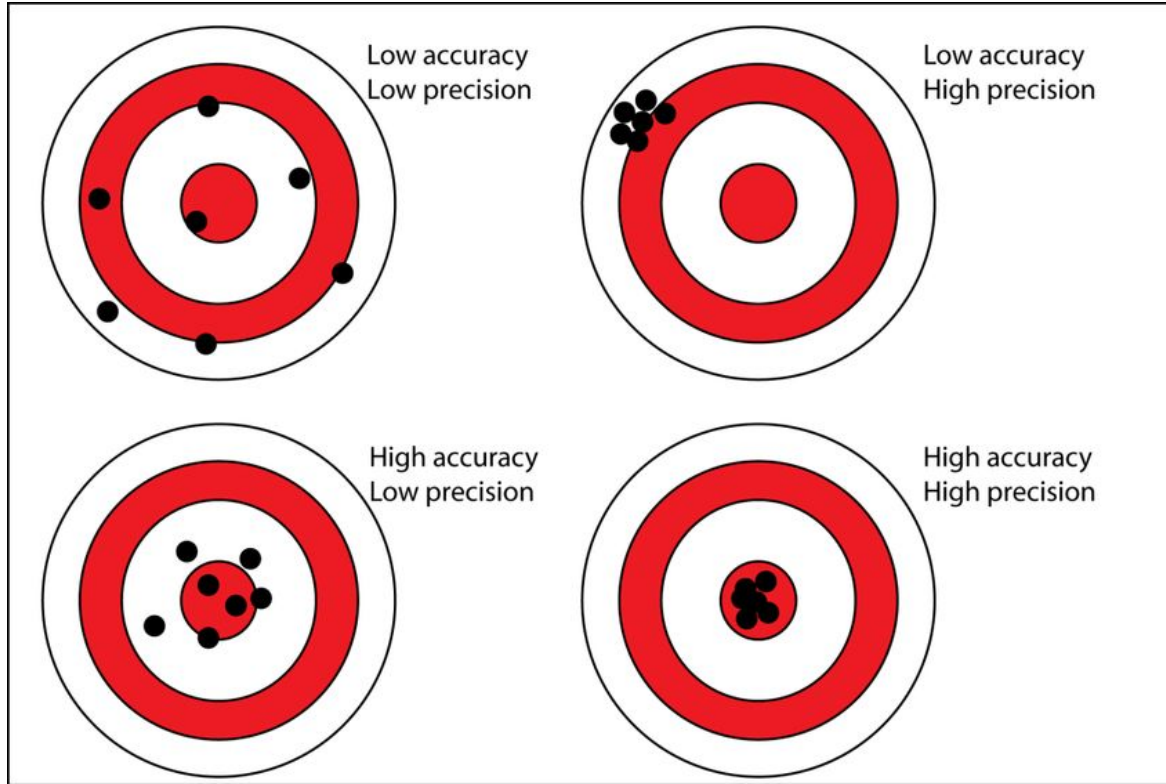
Recall: Out of the **actual** bad loan, how many did we correctly predict as a bad loan?

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

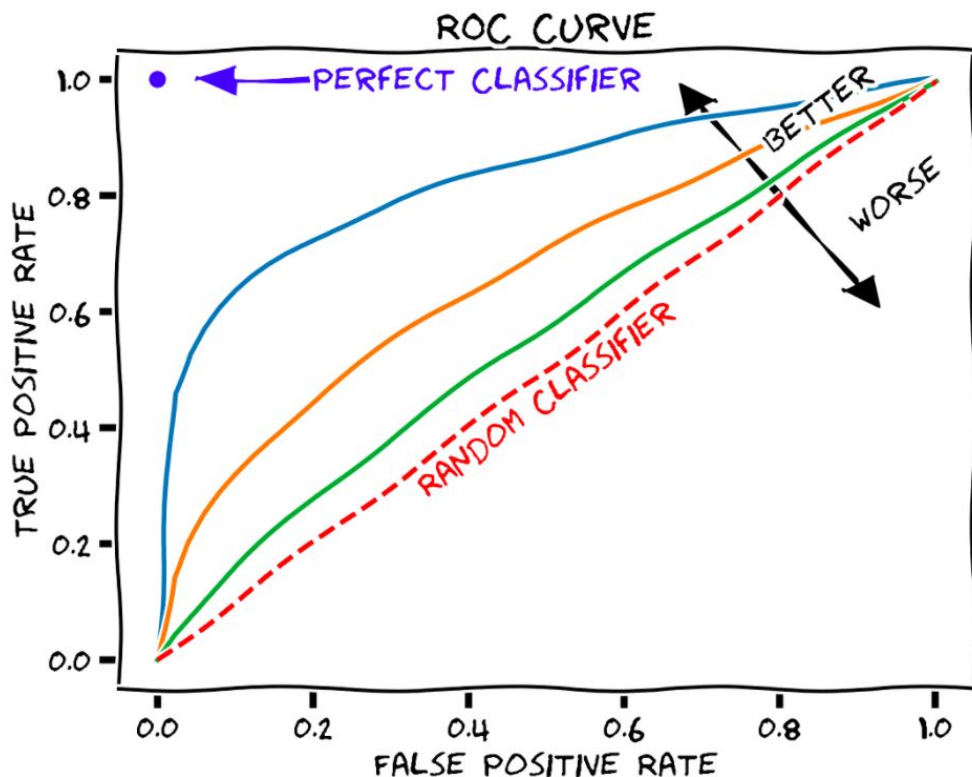
Instead of using **accuracy**, we should evaluate **recall**. If we can decrease **FN**, the recall will increase.

$$\text{Recall} = 559/(559+33) = 94.5\%$$

Precision vs. Accuracy



ROC



- A perfect classifier has 0 false positives (0.0 rate) and correctly predicts all true positives (1.0 rate)
- If you flip a coin (random classifier), your rates will be roughly the same.

Interesting Resources

- <https://www.kaggle.com/>
- <https://www.drivendata.org/>

