Week 4 flow

- Supervised learning and unsupervised learning crash course 1

Supervised and unsupervised learning crash course I

IOAI Training and Selection Programme 2025

Apr 12, 2025 Sat

Recap

Classification vs regression

Regression = predict real-valued numbers

Diabetes dataset from scikit-learn

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019908	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068330	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005671	-0.045599	-0.034194	-0.032356	-0.002592	0.002864	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022692	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031991	-0.046641	135.0

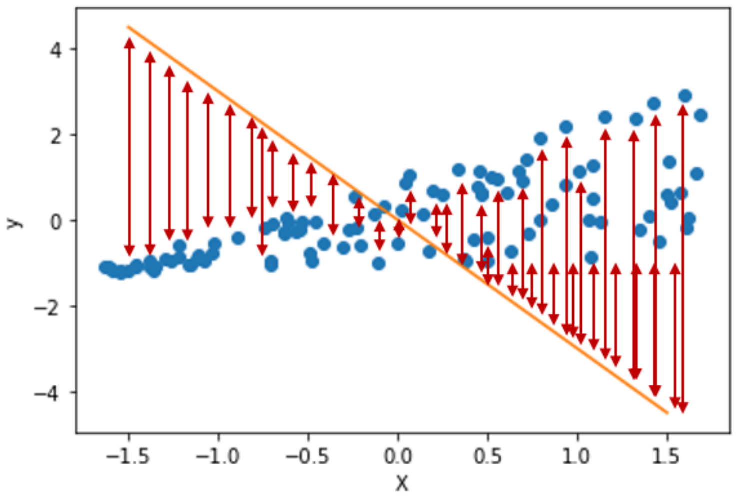
Iris dataset from scikit-learn

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
...	...	...	...	...	...
95	5.7	3.0	4.2	1.2	1
96	5.7	2.9	4.2	1.3	1
97	6.2	2.9	4.3	1.3	1

Classification = predict discrete classes

Regression and classification are supervised learning tasks

Mean squared error (MSE)



Calc
pred

Sma
loss

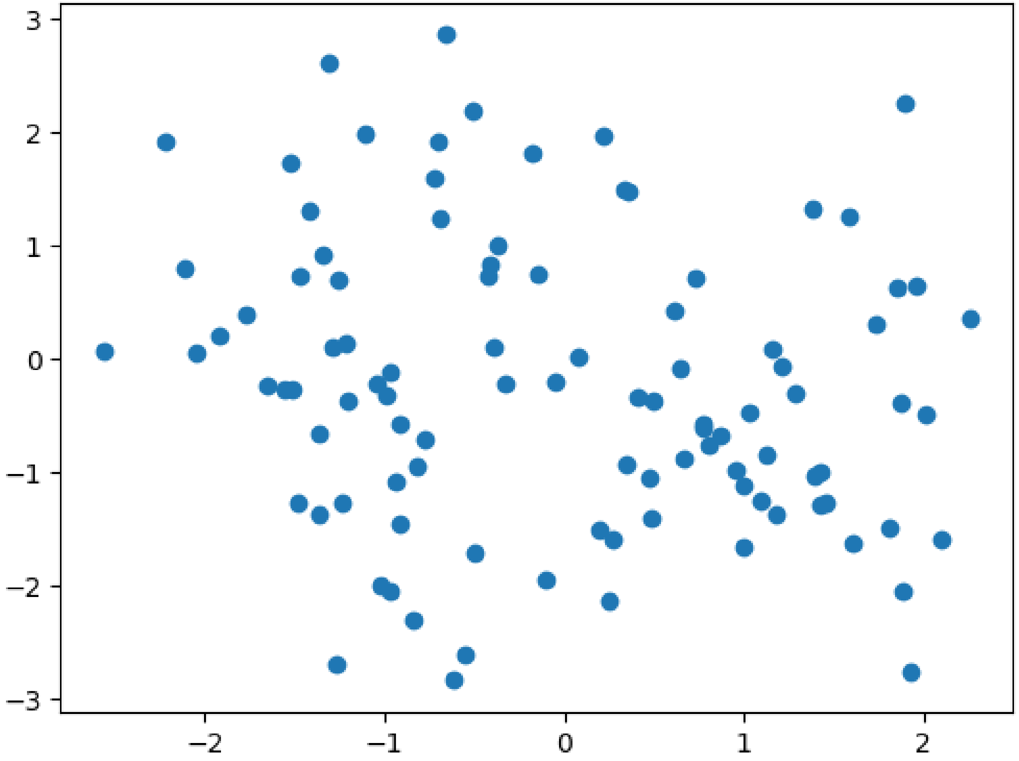
y_{apple}	1	p_{apple}	0.8	$\hat{y}_{apple}$	1
y_{banana}	0	p_{banana}	0.1	$\hat{y}_{banana}$	0
y_{orange}	0	p_{orange}	0.1	$\hat{y}_{orange}$	0

$$
\begin{aligned}
Xent &= -(y_a \ln(p_a) + y_b \ln(p_b) + y_o \ln(p_o)) \\
&= -(1 \times \ln(0.8) + 0 \times \ln(0.1) + 0 \times \ln(0.1)) \\
&= -\ln(0.8) \\
&= 0.223
\end{aligned}
$$

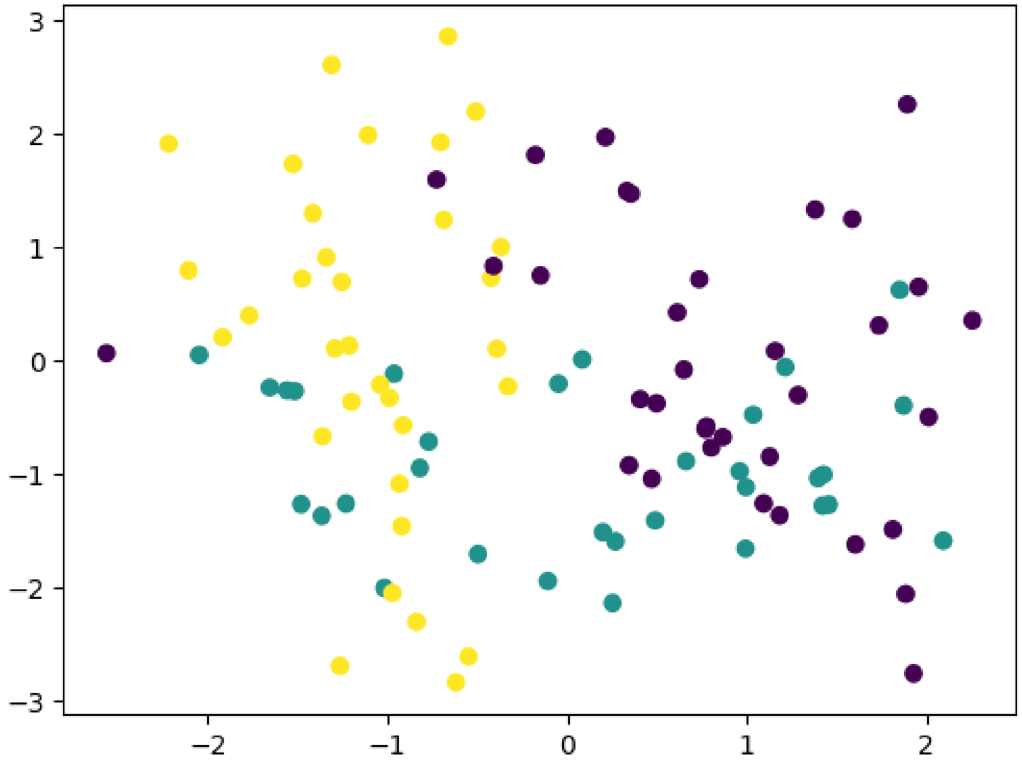
$-\log(p_{true\ class})$

Supervised = guided by labels

Group these datapoints

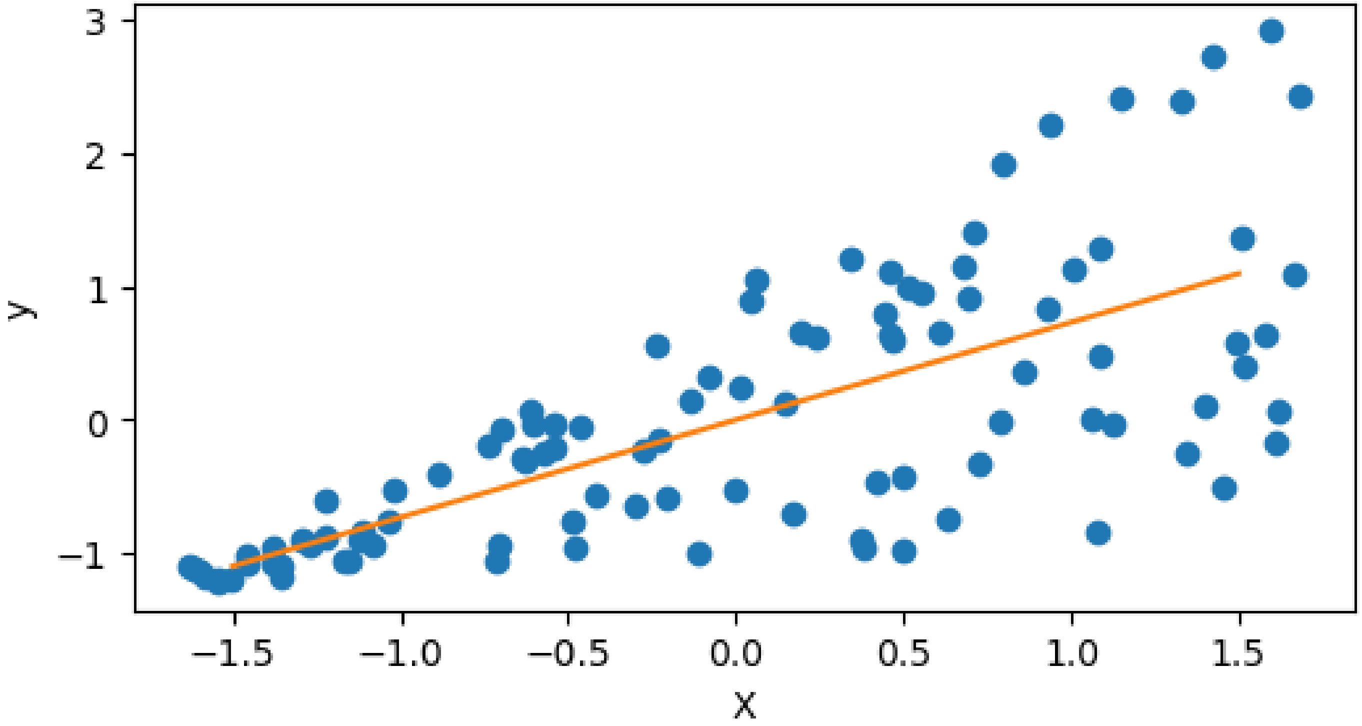


Group these datapoints (cont)



Linear Regression

$$\hat{y} = a_1 x_1 + \cdots + a_n x_n + b$$



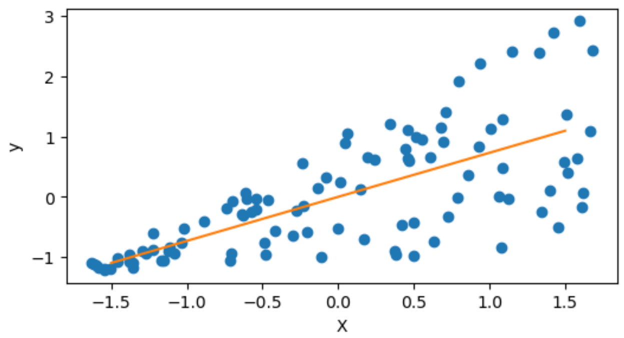
- MSE loss
- Use from sklearn.linear_model

Linear Regression with extra sauce

- L1 regularization = MSE loss + $\lambda \sum |a_i|$
- L2 regularization = MSE loss + $\lambda \sum a^2$
- Tons of variation in sklearn, go explore bah

Logistic regression

Linear regression



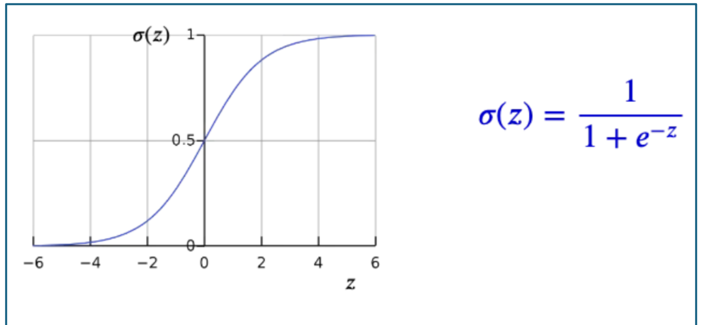
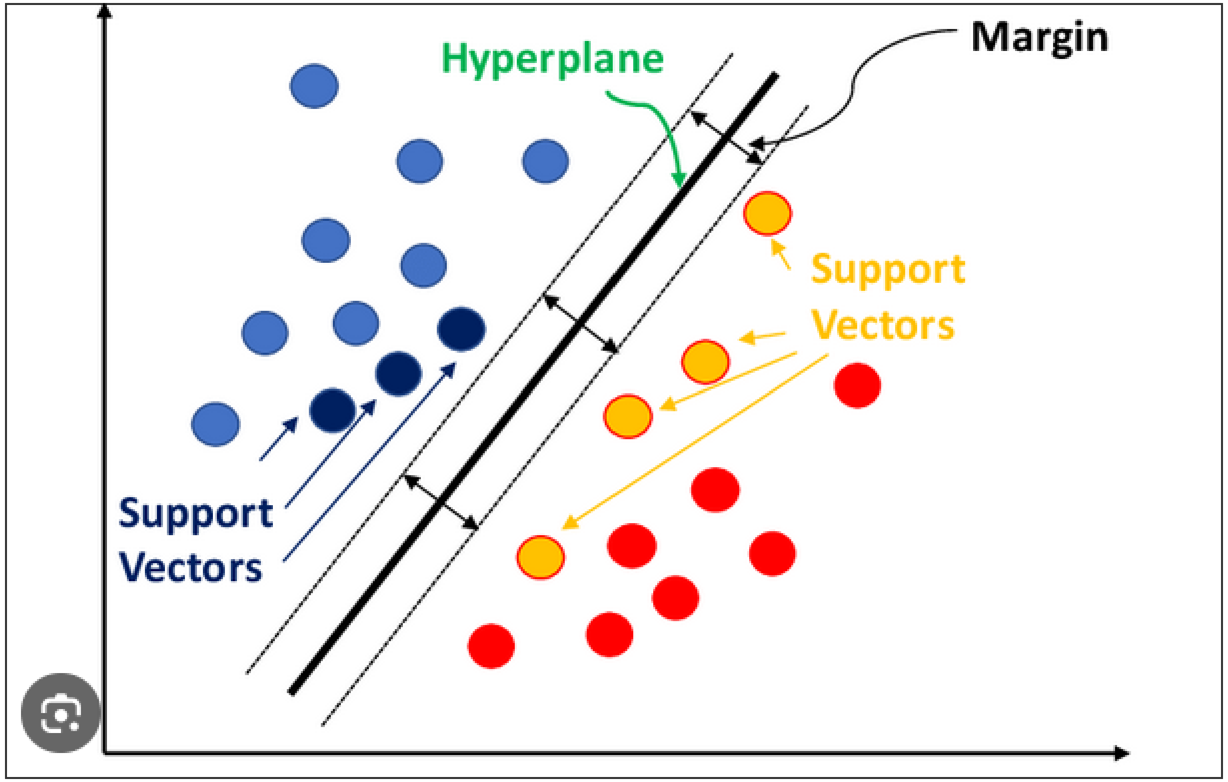


Fig 4. Output sigmoid/logistic node

Logistic function

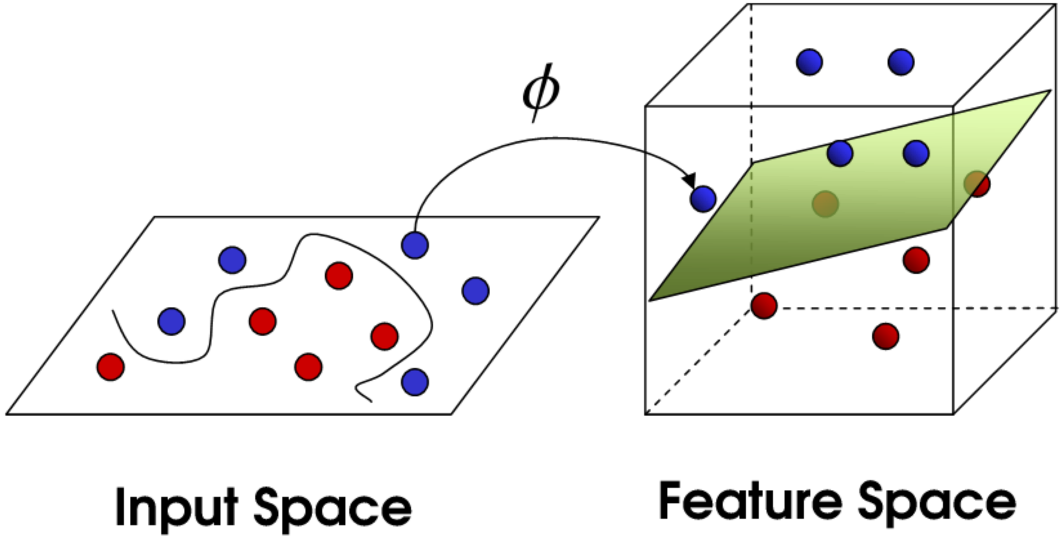
- Linear model
- Uses logistic loss (binary xent)
- Now a classifier!
- Use from sklearn.linear_model

Support Vector Machines



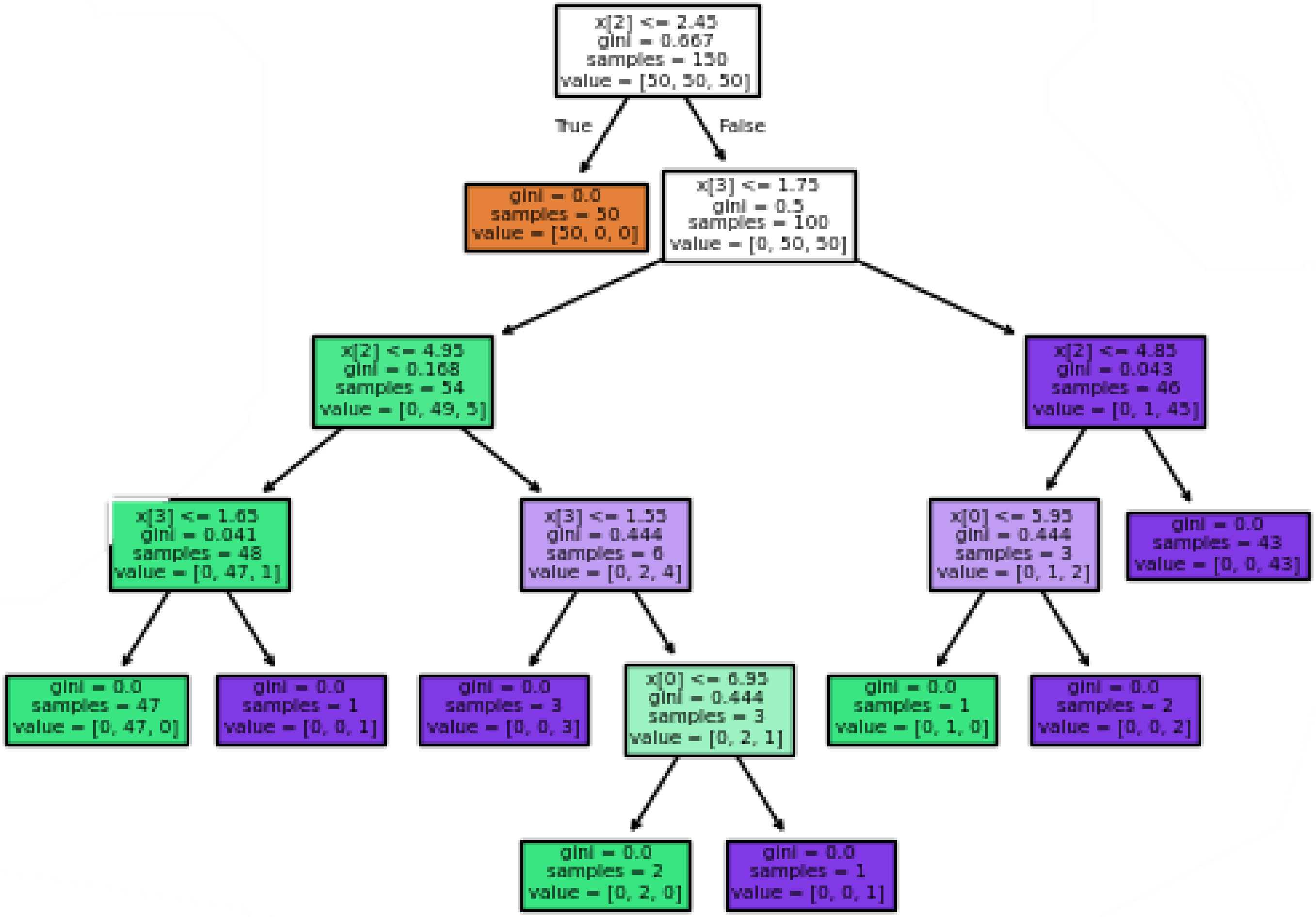
- Aims to find max margin by minimizing hinge loss
- Non-parametric by nature
- Commonly used for classification but works on regression
- Can model non-linear behaviour through kernel methods

Support Vector Machines (cont)



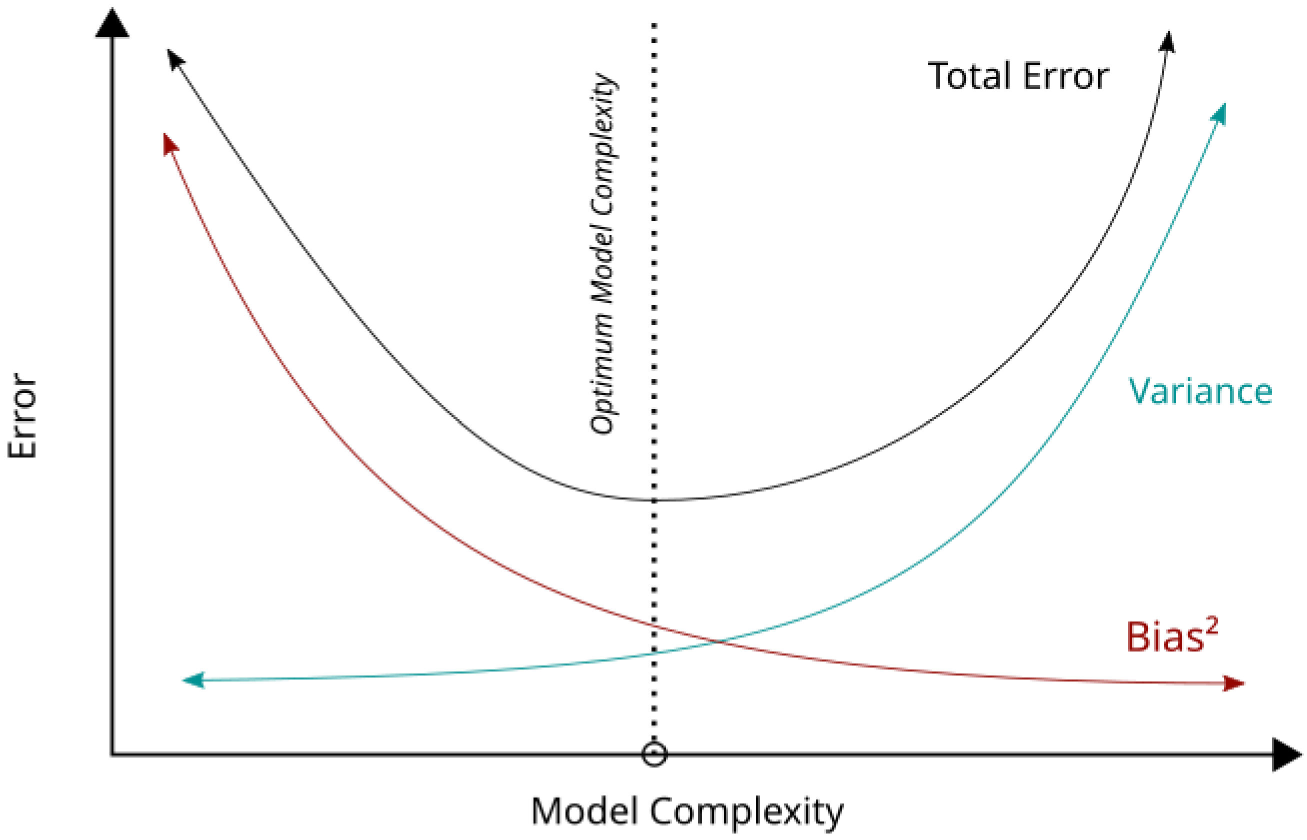
- Common kernels: radial basis function kernel, polynomial kernel
- Effectively has built-in feature engineering
- Was very powerful back in the day for embedding input features into a higher dimension
- Use from sklearn.svm

Decision Trees



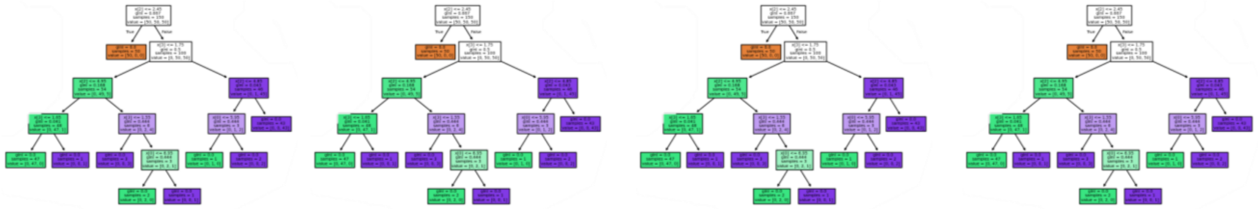
- Literally a bunch of “if-else”
- Decision trees have low bias, which is a useful property when ensembling
- Don’t use this on its own!

Bias vs variance tradeoff



- High bias = model underfit
- High variance = model overfit, too sensitive to small changes to values in the dataset
- Decision trees have low bias (are weak learners) and have high variance

Random Forest

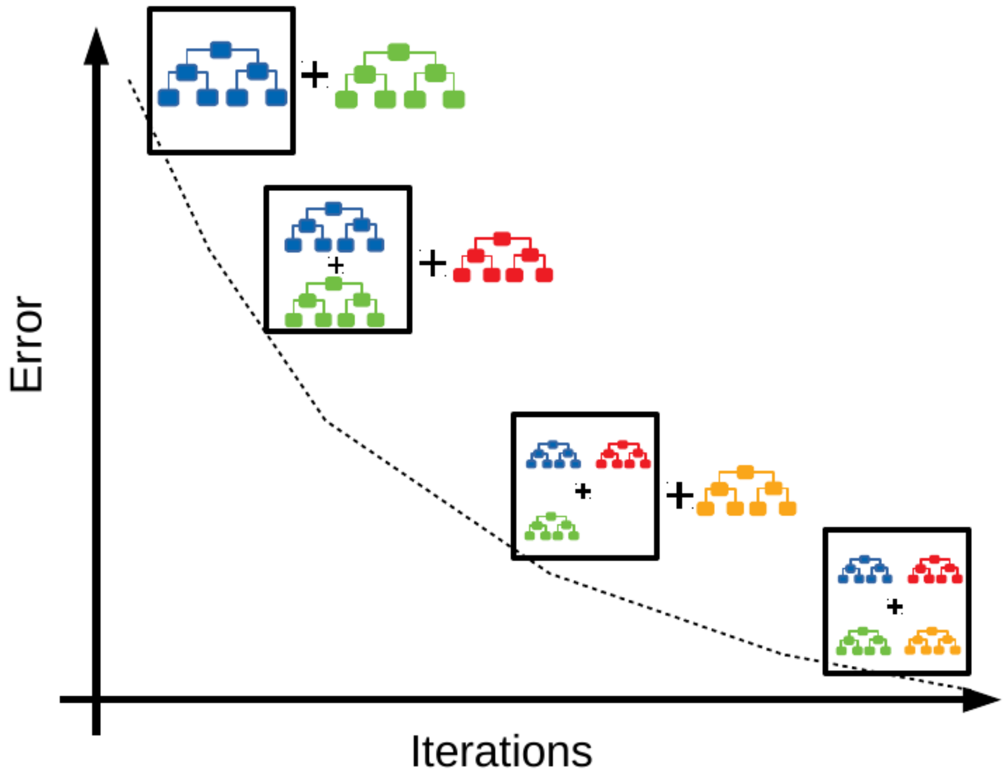


Apple Orange Apple Banana

Apple

- Ensemble trees trained on different samples of data (bootstrap sampling)
- Trees only consider random subsets of features
- Majority vote for classification
- Average predictions for regression
- Non-parametric
- Use from sklearn.ensemble

Gradient Boosted Decision Trees

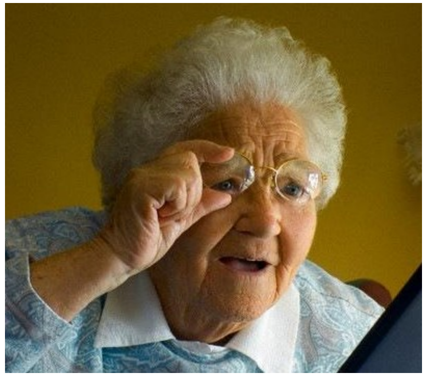
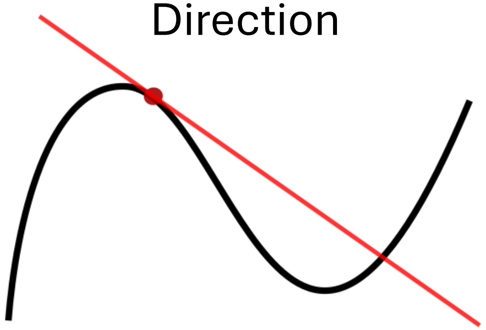


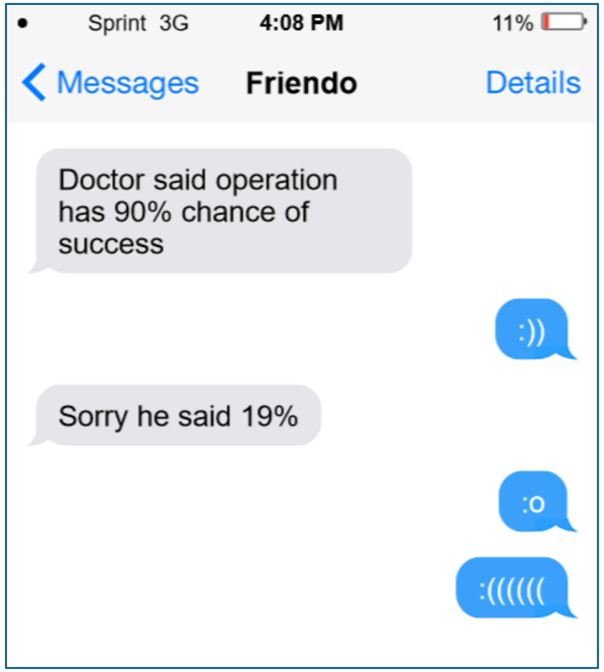
Imagine if your models keep making mistakes like underpredict. Can I just make another model that predicts the mistake, so that I can add a correction factor?

.. yes

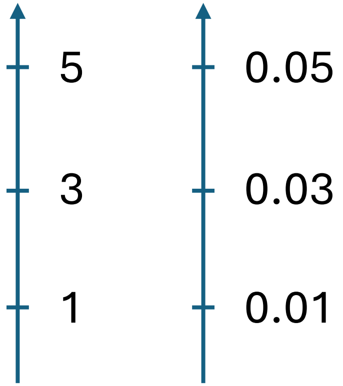
- Use from lightgbm or xgboost

Difference between metrics and loss functions

	Meaningful to …	Purpose	Most important
Metrics		To be an interpretable measure of model performance	Magnitude
Loss functions		To train networks	

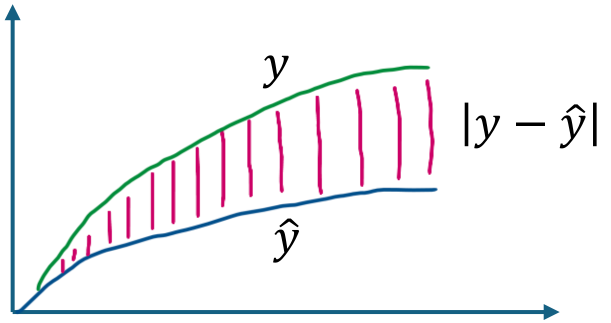


Which y-axis is right?



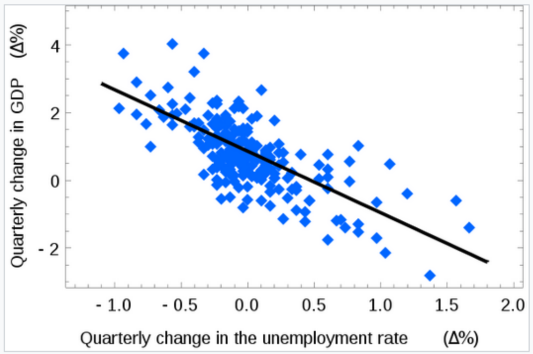
Doesn't matter that much

Common metrics for regression



- Avg → **Mean Absolute Error**
- Square then avg → **Mean Squared Error**
- Divide by y, then avg → **Mean Absolute Percentage Error**
- Square then avg, then sqrt() → **Root Mean Squared Error**

Why? Get back in same units
e.g. The predicted yield of snake eggs has RMSE of 6.7 eggs



Ordinary least squares regression of Okun's law. Since the regression line does not miss any of the points by very much, the R^2 of the regression is relatively high.

R^2, coefficient of determination*

In statistics, the **coefficient of determination**, denoted R^2 or r^2 and pronounced "R squared", is the proportion of the variation in the dependent variable that is predictable from the independent variable(s).

https://en.wikipedia.org/wiki/Coefficient_of_determination

*R^2 tends to have slightly conflicting explanations online, because of how the term R^2 is used across different fields.
See https://towardsdatascience.com/explaining-negative-r-squared-17894ca26321

Common metrics for classification

Context: Sorting edible mushrooms gold coins from poisonous mushrooms chocolate wrapped in gold paper

Accuracy How many edible mushrooms gold coins are predicted as edible? gold coins
How many poisonous mushrooms chocolate coins are predicted as poisonous? chocolate coins

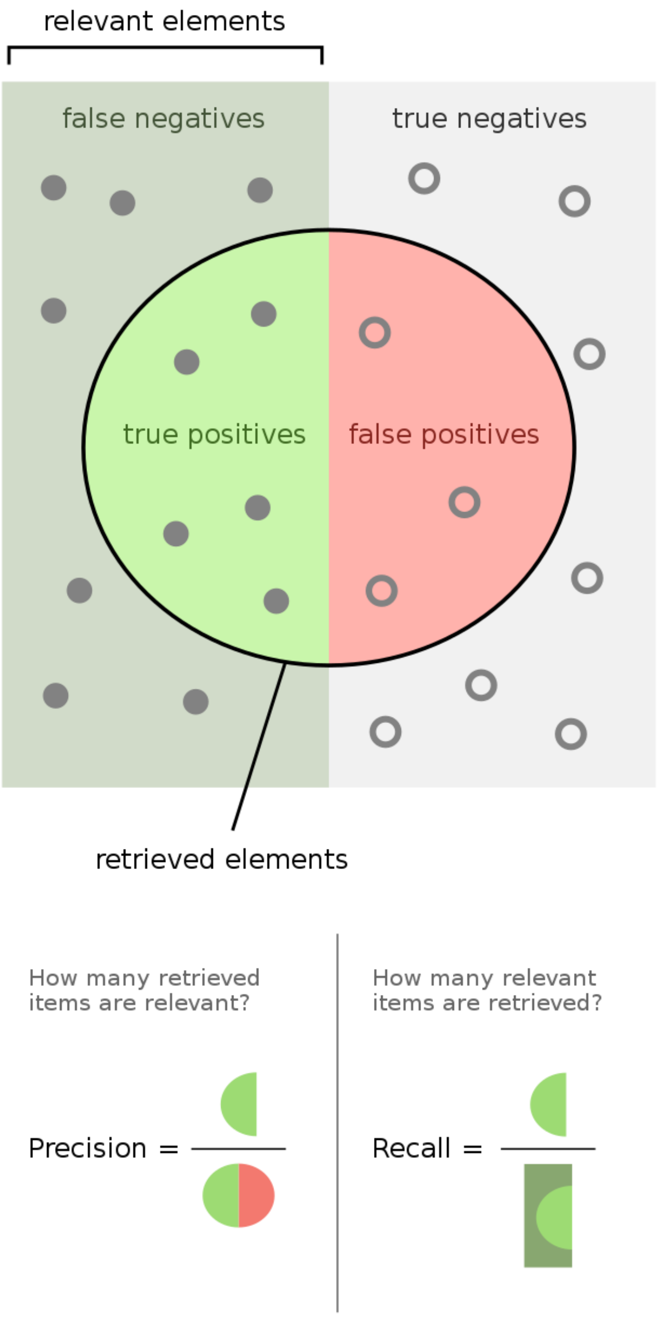
Precision How many of the edible mushrooms gold coins I picked, actually are edible mushrooms gold coins?

Recall How many of the actual edible mushrooms gold coins did I manage to get?

F1-score

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Different contexts require different metrics!



https://en.wikipedia.org/wiki/Precision_and_recall

Common metrics for classification (cont)

Confusion matrix

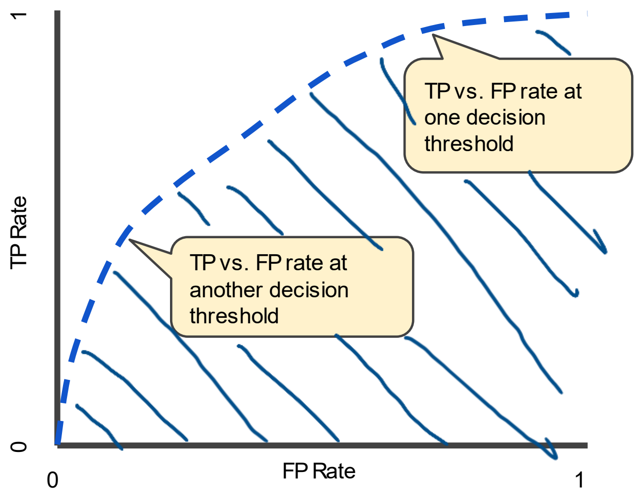
		Predicted condition	
	Total population $= \mathrm{P} + \mathrm{N}$	**Predicted Positive (PP)**	**Predicted Negative (PN)**
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation
	Negative (N) [d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]

https://en.wikipedia.org/wiki/Precision_and_recall

Area Under Receiver Operating Characteristic Curve

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

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



https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Regression and classification are two sides of the same coin

- Hope you realize there isn’t a super clear line between regression vs classification
- There isn’t a dichotomy where everything is either regression or classification
- I skipped a lot of material, go read the book: https://github.com/harvard-ml-courses/cs181-textbook/blob/master/Textbook.pdf