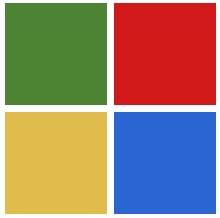
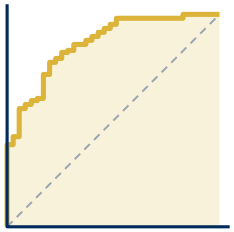
UNIT 02 · CHAPTER 11

Model Evaluation





CSE 30124 · Introduction to Artificial Intelligence
William Theisen · University of Notre Dame

Deciding on which Model

We've seen a large number of different models now, as well as a number of different use cases. **How should we decide which model to use and when?**

Models for different Tasks:

Classification:

- kNN
- Decision Trees
- Perceptrons
- Logistic Regression
- SVMs

Regression:

- kNN
- Decision Trees
- Linear Regression
- SVRs

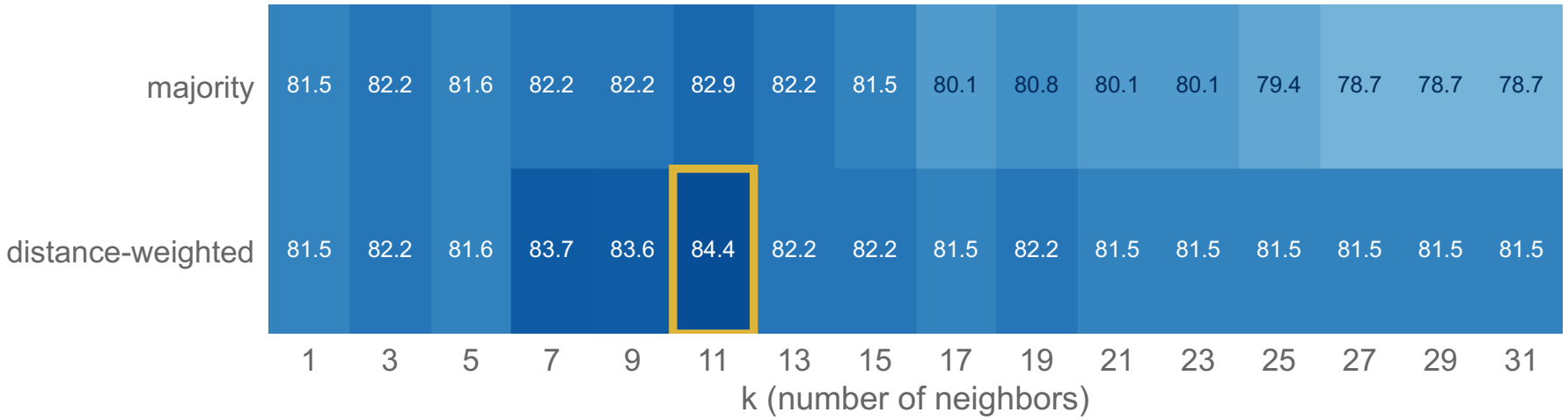
Clustering:

- kMeans
- Hierarchical

Grid Search

The simplest way to decide which model to use on a given task is to test different models and configurations on the task and compare their results

Grid Search is simply the term for **systematically testing** different combinations of



Model Evaluation

Lets suppose we were training a **classifier** to predict whether or not **Notre Dame** won a football game

classifier	features	unseen games
kNN (k = 5)	yards	81.6%
decision tree (depth 3)	yards	82.3%
logistic regression	yards	83.7%
random forest (200 trees)	15 stats	83.6%
logistic regression	15 stats	**85.8%**

How good are these **classifiers**? Thus far we've been using _____

(how close a given set of **predictions** are to their _____)

Model Evaluation

However, **using only _____ can be misleading**

classifier	features	unseen games
always guess "win"	none	**75.2%**
kNN (k = 5)	yards	81.6%
decision tree (depth 3)	yards	82.3%
logistic regression	yards	83.7%
random forest (200 trees)	15 stats	83.6%
logistic regression	15 stats	**85.8%**

Notre Dame wins 3 out of 4 of its football games, resulting in an **imbalanced dataset**. Out of our 141 samples, only 35 of them are losses. This means that we could get an accuracy of 75% by just predicting Notre Dame to win every game.

Confusion Matrices

A **confusion matrix** helps assess a **classification** model's performance by tabulating each of the four possible outcomes from a **binary prediction task**



_____ refer to **data points belonging to the target class (win)**, while

_____ refer to **data points belonging to the non-target class (loss)**

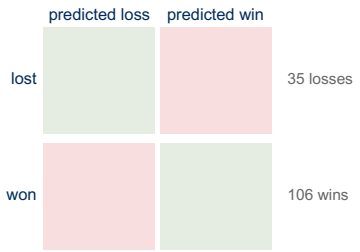
Confusion Matrices and Imbalanced Datasets

Notre Dame wins 3 games in 4, so the games worth predicting are the losses



Type 1 vs. Type 2 Errors

Type 2 errors are called "**false negatives**"



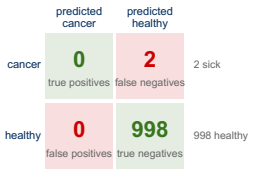
Type 1 errors are called "**false positives**"

Depending on the task at hand, we may care more about these types of errors, but only considering the overall accuracy may hide them from us.

Type 1 vs. Type 2 Errors

Consider a task in which we're performing a cancer screening, with the presence of cancer being the positive class.

If we have a dataset with 998 healthy examples, and 2 cancerous examples, we could get an overall accuracy of 99.8% by just predicting that everyone is healthy!



However, missing the 2 sick patients would be **false negatives**, and if we're trying to catch deadly diseases, making these **type 2 errors** are extremely costly!

Type 1 vs. Type 2 Errors

Type 1 errors are called "**false positives**"

- _____
- _____
- _____

Type 2 errors are called "**false negatives**"

- _____
- _____
- _____

Type 1 and Type 2 Errors in the Body

Our **immune system** is a classifier: every cell and molecule it meets is either "self" or "not self".

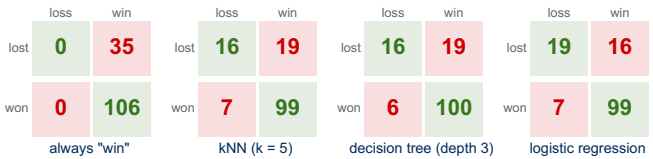
- **False positive:** attacking our own cells, which is what happens in **autoimmune diseases** like type 1 diabetes
- **False negative:** _____ a virus or bacterium, and getting sick

The body can't push both to zero, so it **trades one against the other**, the same as moving a decision threshold. ROC curves started out with radar operators, and Green and Swets (1966) brought them into psychology to measure how well people hear faint tones.

Accuracy

[11.1] accuracy

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



Accuracy:

- **Always "win":** _____
- **kNN:** _____
- **Decision Tree:** _____
- **Logistic Regression:** _____

Accuracy

Depending on what we care about, there are several other metrics we can use to evaluate our models other than accuracy.

- Accuracy
- _____
- _____
- _____

K-Fold Cross-Validation

The best way to get an accurate accuracy is to use a method called **K-Fold Cross-Validation**



In **K-Fold Cross-Validation**, we **split our dataset into K folds**, with one of the folds acting as the testing data, and the remaining folds serving as the training data.

We can then **report the average accuracy** across all K results.

Accuracy and K-Fold in sklearn

Like usual, it's very easy to run **K-fold cross-validation** and produce a **confusion matrix** in **sklearn**

```python
import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix

games = pd.read_csv("nd_football.csv")
X = games[["nd_total_yards", "opp_total_yards"]] / 100
y = 1 - games["win"]   # a loss is our positive class

model = LogisticRegression(C=1e9)

pred = cross_val_predict(model, X, y, cv=5)

print(confusion_matrix(y, pred, labels=[1, 0]))
```

OUTPUT

```
[[19 16]
 [ 7 99]]
```

Precision

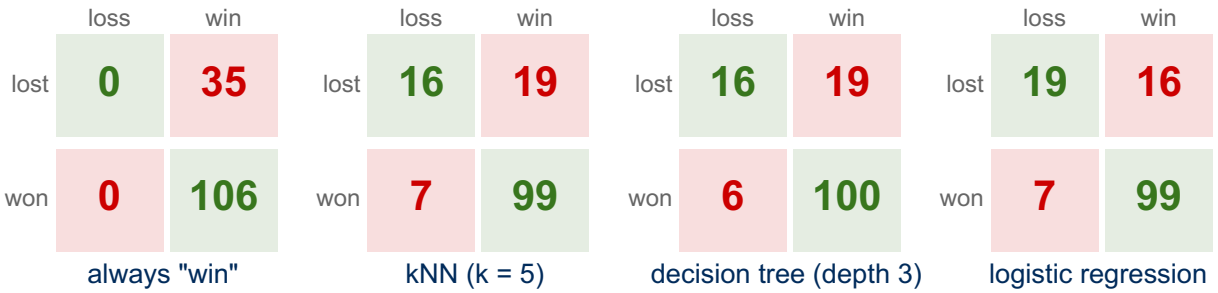
Precision is how **close the measurements** are to _____

[11.2] precision

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

We typically use **precision** when **false positives (type 1 errors) are costly**

Precision



Accuracy:

- **Always "win":** _____
- **kNN:** _____
- **Decision Tree:** _____
- **Logistic Regression:**

  _____

Precision:

- **Always "win":**

  _____
- **kNN:** _____
- **Decision Tree:** _____
- **Logistic Regression:**

  _____

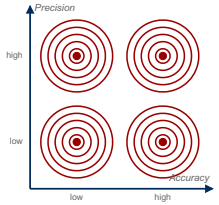
Accuracy and Precision

Accuracy:

how **close your predictions are to the true values**.

Precision:

how **close your predictions are to each other**.



Exam question: Draw shots on the four targets for low and high accuracy (left to right) and low and high precision (bottom to top).

Recall

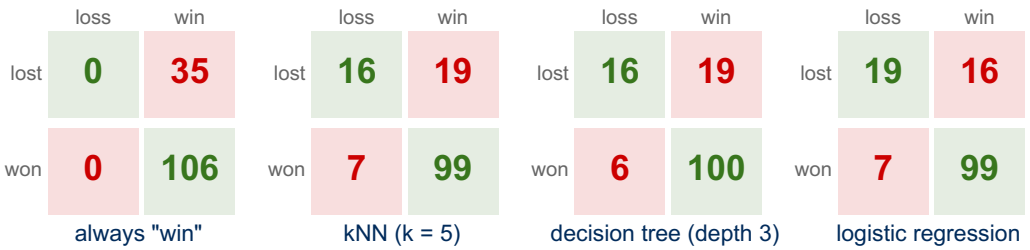
The proportion of **actual positives** that were classified _____.

[11.3] recall

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

We typically use **recall** when **false negatives (type 2 errors) are costly**

Recall



Accuracy:

- **Always "win":** _____
- **kNN:** _____
- **Tree:** _____
- **Log. reg.:** _____

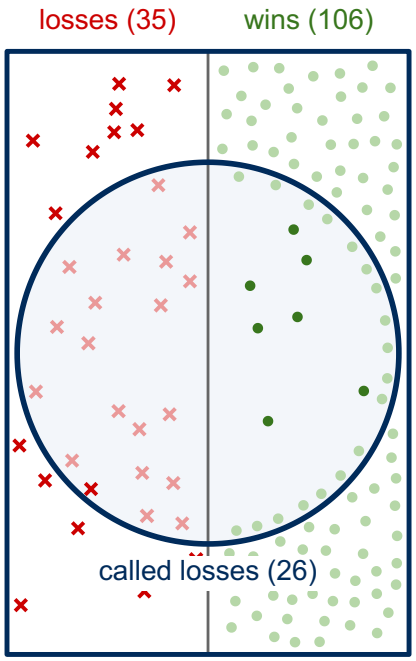
Precision:

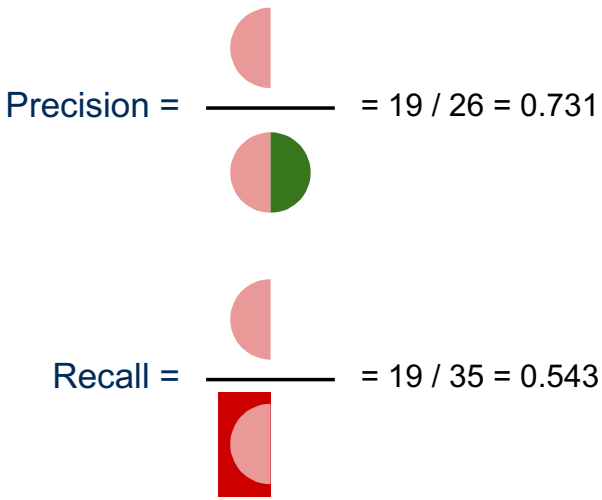
- **Always "win":** _____
- **kNN:** _____
- **Tree:** _____
- **Log. reg.:** _____

Recall:

- **Always "win":** _____
- **kNN:** _____
- **Tree:** _____
- **Log. reg.:** _____

Precision vs. Recall

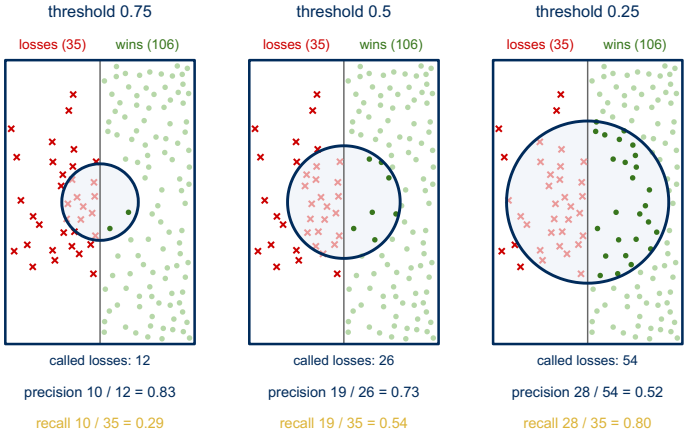




Precision only looks inside the circle, where **19** of the **26** games are losses. **Recall** only looks at the left half, where **19** of the **35** losses made it into the circle.

Precision vs. Recall

Notice what happens if the circle gets bigger. It picks up more of the missed losses, so recall goes up, but it also picks up more wins, so precision goes down. Shrink it and the opposite happens. **Usually we can trade one for the other**



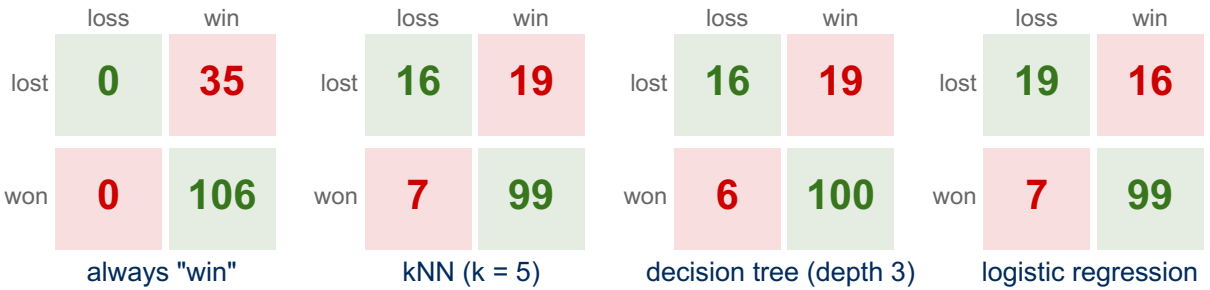
F1-Score

F1-Score is a better measure to use if we need to _____ between **Precision** and **Recall** AND there is _____

[11.4] F1-score, the harmonic mean of precision and recall

$$F_1 = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

F1-Score



F1-Score:

- **Always "win":** _
- **kNN:** _____
- **Decision Tree:** _____
- **Logistic Regression:** _____

Classification Reports

In `sklearn`, `classification_report` prints **precision**, **recall** and **F1** with each class taking a turn as the positive one:

```
import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import cross_val_predi
from sklearn.metrics import classification_report

games = pd.read_csv("nd_football.csv")
X = games[["nd_total_yards", "opp_total_yards"]] /
y = 1 - games["win"]   # a loss is our positive cla

model = LogisticRegression(C=1e9)
pred = cross_val_predict(model, X, y, cv=5)

print(classification_report(y, pred, digits=3,
                            target_names=["win", "l
```

OUTPUT

	precision	recall	f1-score
win	0.861	0.934	0.896
loss	0.731	0.543	0.623
accuracy			0.837
macro avg	0.796	0.738	0.759
weighted avg	0.829	0.837	0.828

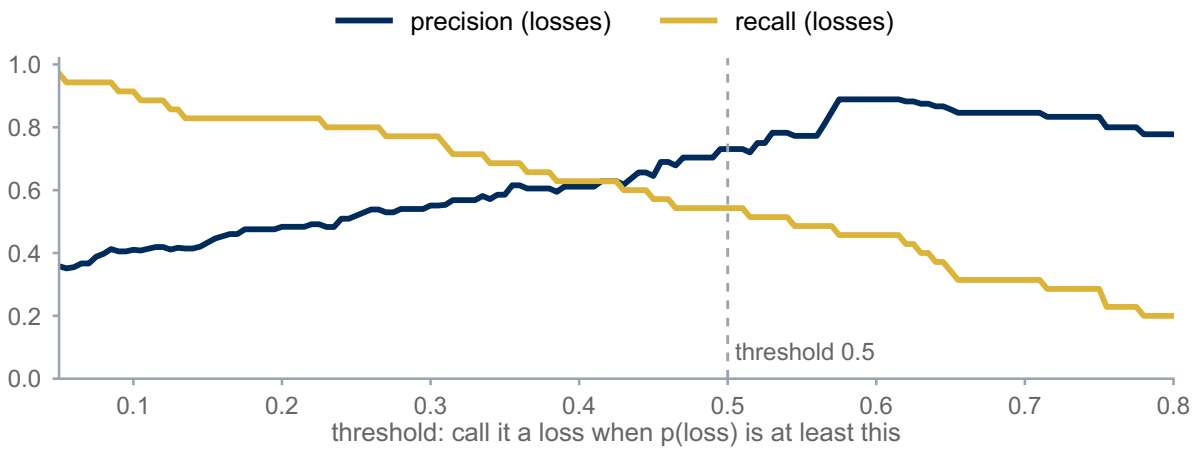
`support` is the number of true samples of each class in the data being scored.

Accuracy vs. Precision vs. Recall

- **Accuracy** - when **class distribution is balanced** and **false positives and false negatives have similar consequences**.
- **Precision** - when **false positives are costly** or undesirable
- **Recall** - when **false negatives are more critical** than false positives
- **F1** - when you need a **balance between precision and recall** and the **class distribution is imbalanced**.

Evaluating Decision Thresholds

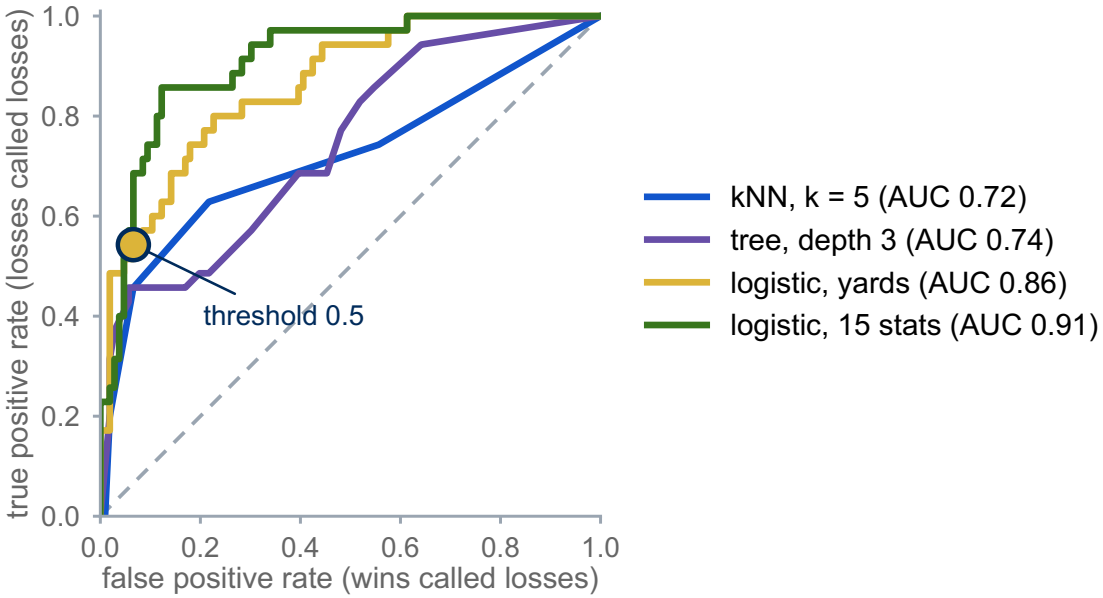
Another parameter that we could change is where the decision threshold lies, thus far for logistic regression we've just set it to 0.5



Every threshold gives a different confusion matrix, and **raising recall costs precision**.

Evaluating Decision Thresholds

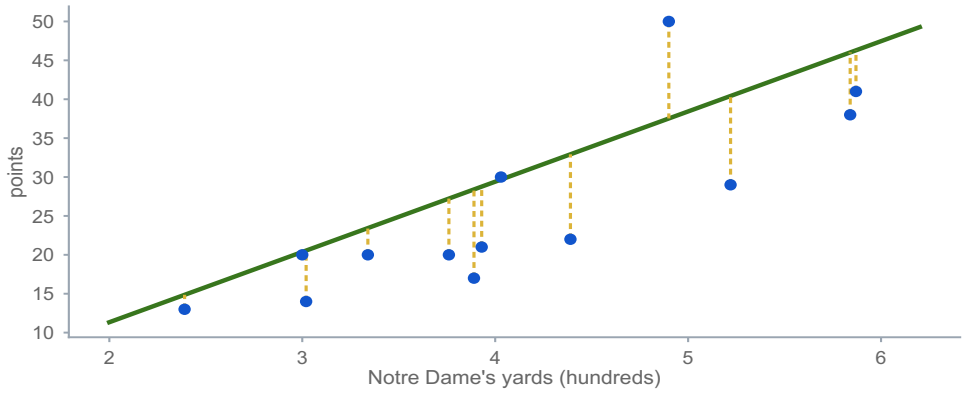
Rather than pick one threshold, we can plot all of them. This is a **receiver operating characteristic (ROC)** curve. From this we can get a single number called the **AUC (Area Under Curve)**



Evaluating Regression Tasks

Remember that for **regression tasks** we are trying to predict _____

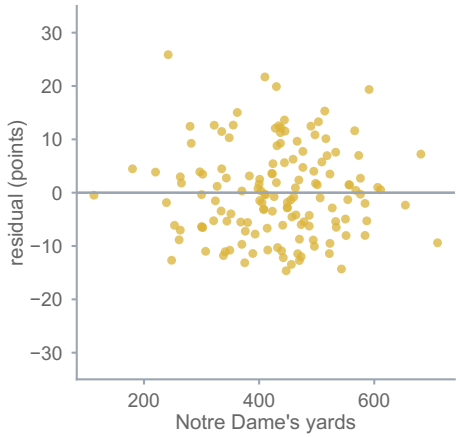
Because **regression tasks are supervised**, we have ground-truth so we can tell **how far away our predicted answers are from the ground truth**.



These gaps are called the _____ **(R)** and the goal of regression is to **minimize the residuals**.

Residuals Plot

A **residual plot** is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis.



If the points in a residual plot **are randomly dispersed** around the horizontal axis, **a linear regression model is appropriate for the data**, **otherwise, a non-linear model is more appropriate**.

Regression Tasks - R^2

R^2 indicates the **proportion of variance in the dependent variable that can be explained by the independent variable(s)** in a model.

the share of the variance the model explains

$$[11.5] \quad R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

R^2 shows **how well the regression line fits the data**, with a value ranging from **0 (poor fit) to 1 (perfect fit)**.

Mean Squared Error

Another option is, using the residuals, we can write a number of different **error functions** for **regression tasks**:

- Residual Sum of Squares (RSS)
- Mean Absolute Error (MAE)
- _____ **(MSE)**
- Mean Percentage Error (MPE)
- Mean Absolute Percentage Error (MAPE)
- Root Mean Squared Error (RMSE)

Mean Squared Error

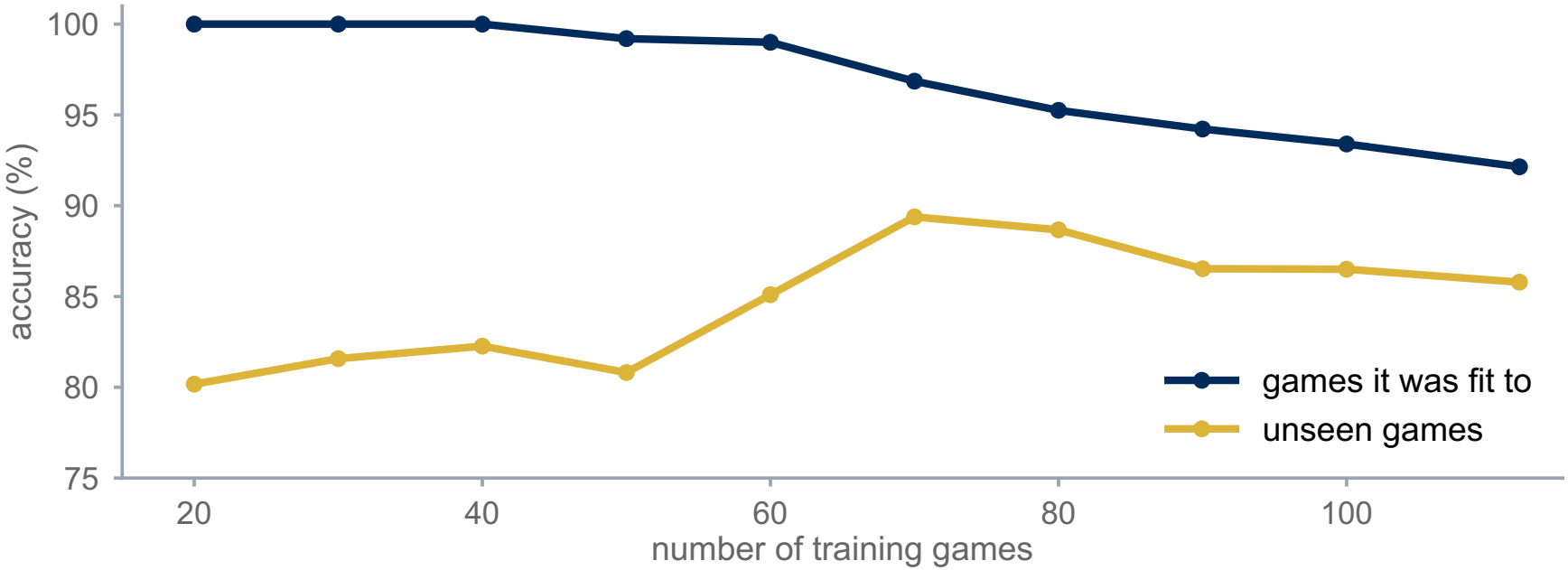
MSE measures **the average squared difference** between the estimated values and the actual value

mean squared error

[11.6] $$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

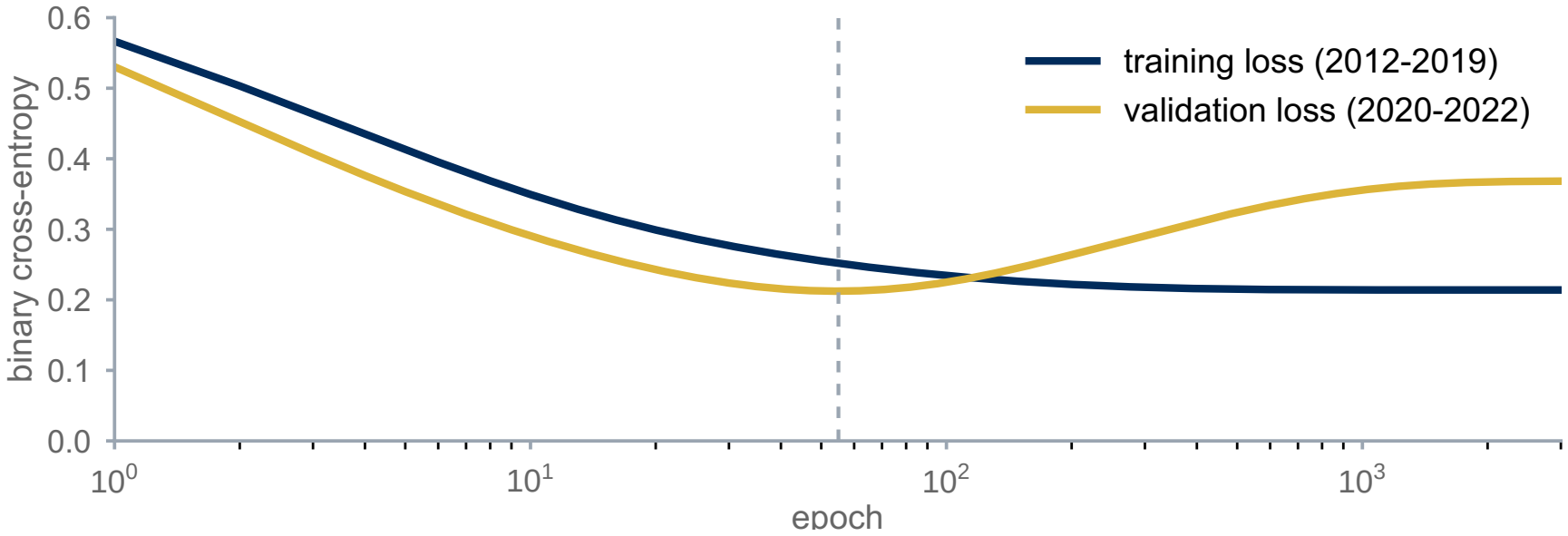
Learning Curves - Machine Learning

Learning curves are a plot of **model learning performance** over



Learning Curves - Machine Learning

For **parametric** models, we can also plot **learning curves** over _____



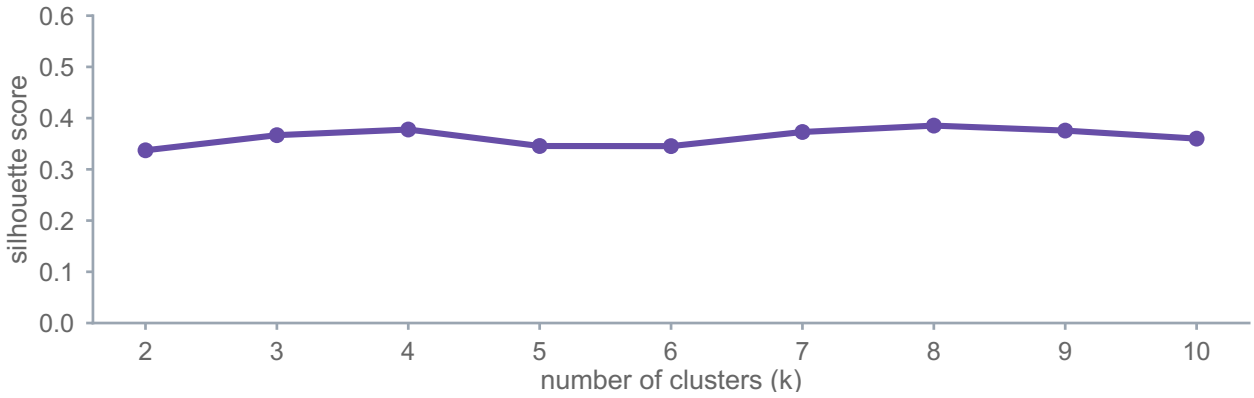
Assessing Clustering

Assessing **unsupervised learning** is difficult as there are **no** _____ against which we can compare results.

We often don't know what the right answer is!

Silhouette Score

Silhouette score is a numerical value that measures how well a **data point belongs to its cluster** and how **separated it is from other clusters**.

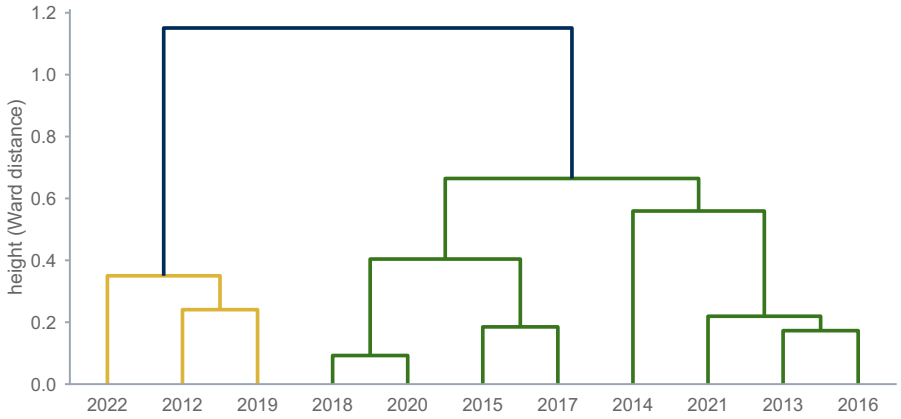


The score ranges from -1 to 1, with **higher values indicating better clustering**.

Dendrograms

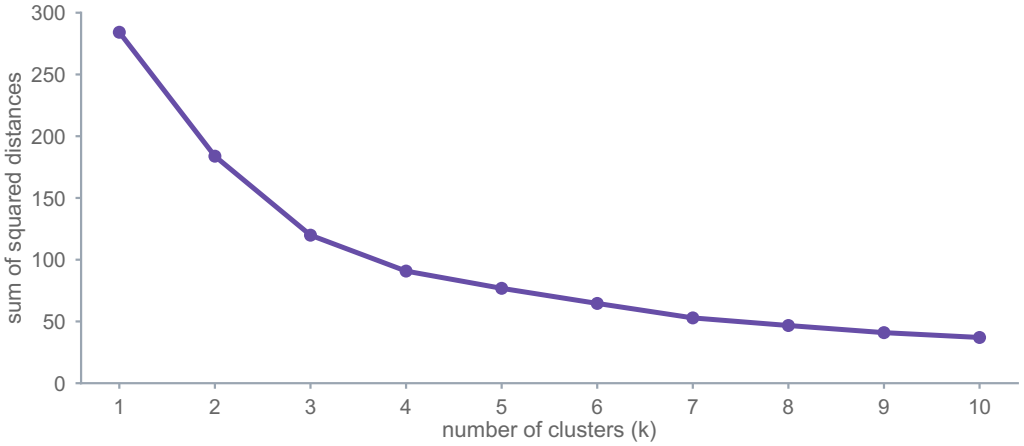
A **dendrogram** is a branching diagram that **shows the relationships** between classes

To assess a clustering, **cut the tree where the vertical lines are longest**. Here the last merge (`1.15`) is far above the one before it (`0.66`), so two clusters is the natural cut.



Elbow Plots

Previously, we have used **elbow plots** to estimate the number of _____ to use on a data set.



In **cluster analysis**, we can plot the **sum of squared distances from each point to its centroid** as a function of the number of clusters and **pick the elbow of the curve as the number of clusters to use**.