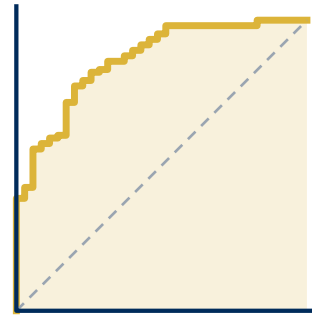




UNIT 02 · CHAPTER 11

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:

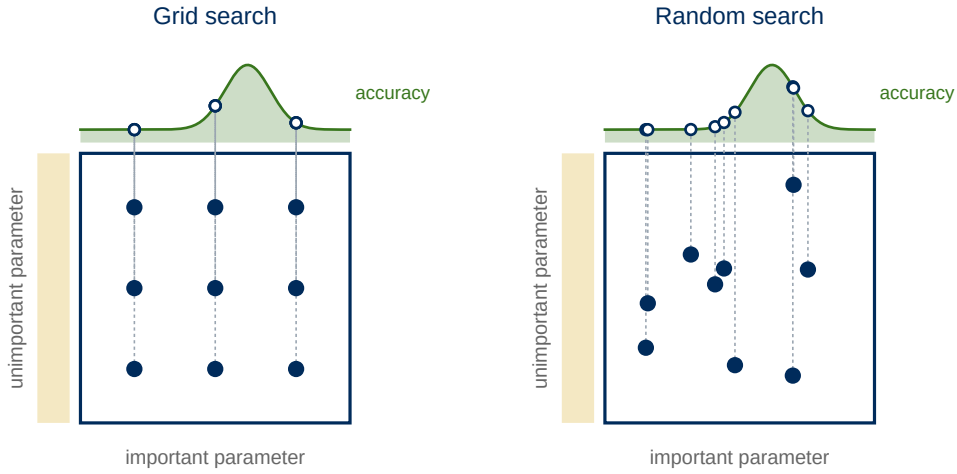
- kNN
- Decision Trees
- Linear Regression
- SVRs

Clustering:

- kMeans
- Hierarchical (Agglomerative)

Grid Search

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



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

Model Evaluation

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

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%**

How good are these **classifiers**? Thus far we've been using _____
(how close a given set of **predictions** are to their _____)

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

Confusion Matrices

A **confusion matrix** helps assess a **classification** model's performance by **comparing predicted values against actual values**

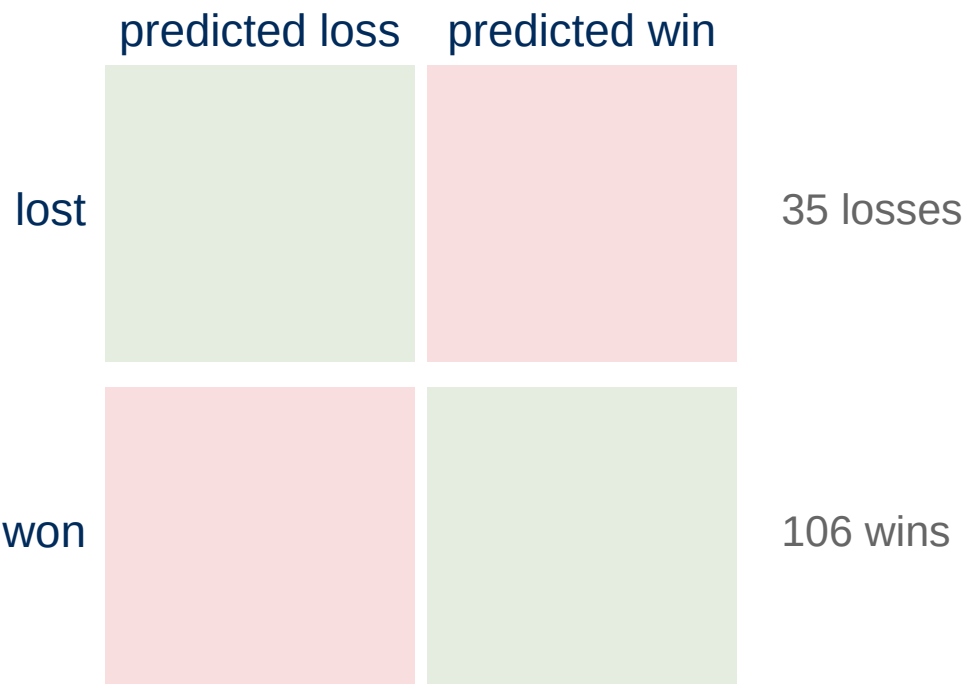
	predicted loss	predicted win	
lost	**19** true positives	**16** false negatives	35 losses
won	**7** false positives	**99** true negatives	106 wins

_____ refer to **data points belonging to the target class**, while _____ refer to **data points belonging to the non-target class**

Notre Dame wins **3** games in **4**, so the games worth predicting are the losses: **a loss is our**

_____

Type 1 vs. Type 2 Errors



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

- _____
- _____
- _____

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

- _____
- _____
- _____

Accuracy

[11.1] accuracy

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

	loss	win
lost	0	35
won	0	106

always "win"

	loss	win
lost	16	19
won	7	99

kNN (k = 5)

	loss	win
lost	16	19
won	6	100

decision tree (depth 3)

	loss	win
lost	19	16
won	7	99

logistic regression

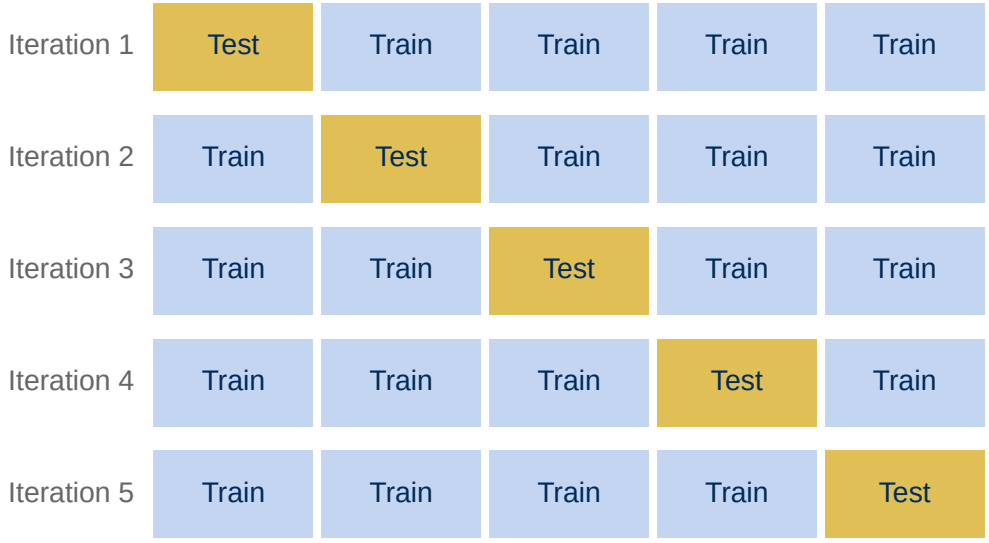
Taken to the extreme, picture a test that has to find `2` sick patients among `1,000`. If it finds one of them and calls everybody else healthy, it's `99.9%` accurate and has still missed half the people it was built to find.

Classification Model Evaluation Metrics

- 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`**

```
1  import pandas as pd
2
3  from sklearn.linear_model import LogisticRegression
4
5  from sklearn.model_selection import cross_val_predict
6  from sklearn.metrics import confusion_matrix
7
8  games = pd.read_csv("nd_football.csv")
9  X = games[["nd_total_yards", "opp_total_yards"]] / 100
10 y = 1 - games["win"]   # a loss is our positive class
11
12 model = LogisticRegression(C=1e9)
13
14 pred = cross_val_predict(model, X, y, cv=5)
15
16 print(confusion_matrix(y, pred, labels=[1, 0]))
```

OUTPUT

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

`labels=[1, 0]` puts the losses first, the same way round as ours (`sklearn` would put the `0`s first).

Precision

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

[11.2] precision

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

	loss	win
lost	0	35
won	0	106

always "win"

	loss	win
lost	16	19
won	7	99

kNN (k = 5)

	loss	win
lost	16	19
won	6	100

decision tree (depth 3)

	loss	win
lost	19	16
won	7	99

logistic regression

Accuracy:

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

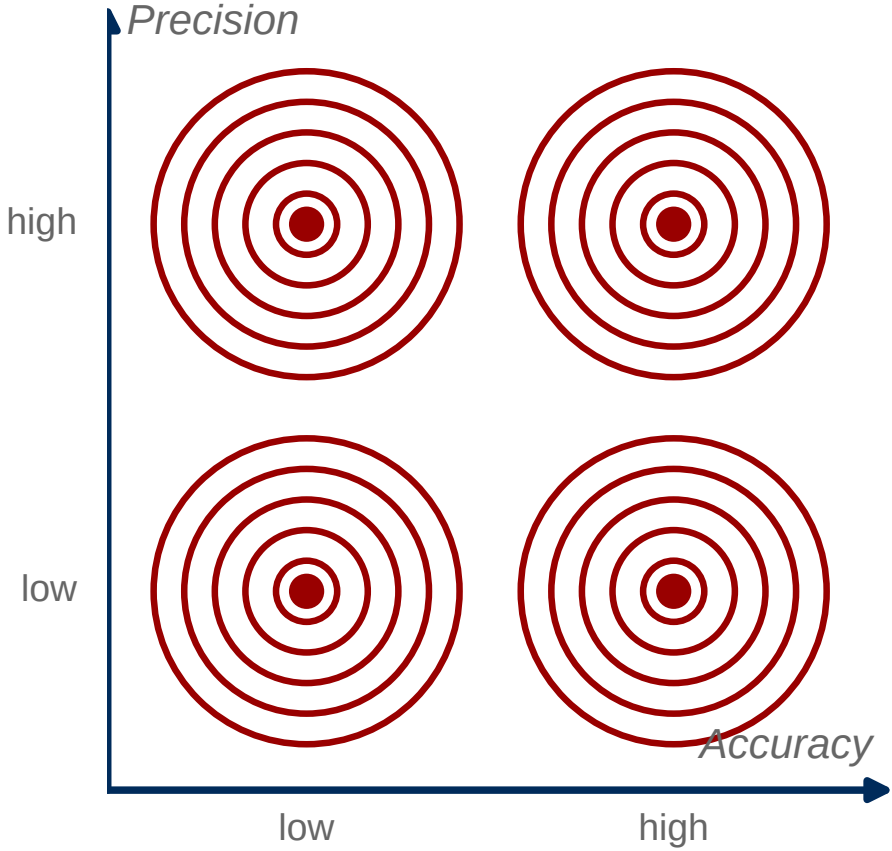
Precision:

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

Accuracy and Precision

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

Precision is 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

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

always "win"

	loss	win
lost	0	35
won	0	106

kNN (k = 5)

	loss	win
lost	16	19
won	7	99

decision tree (depth 3)

	loss	win
lost	16	19
won	6	100

logistic regression

	loss	win
lost	19	16
won	7	99

Accuracy:

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

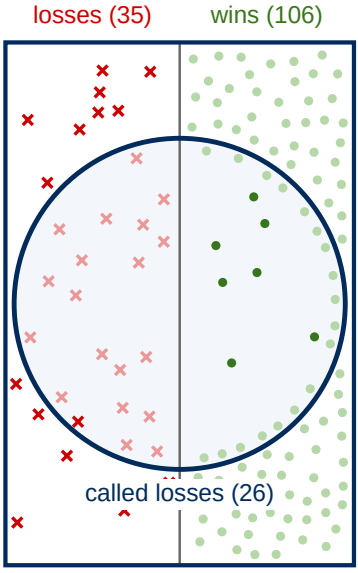
Precision:

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

Recall:

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

Precision vs. Recall



Precision = $\frac{\text{losses called}}{\text{games called}}$ = 19 / 26 = 0.731

Recall = $\frac{\text{losses called}}{\text{losses}}$ = 19 / 35 = 0.543

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.

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

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

	loss	win
lost	0	35
won	0	106

always "win"

	loss	win
lost	16	19
won	7	99

kNN (k = 5)

	loss	win
lost	16	19
won	6	100

decision tree (depth 3)

	loss	win
lost	19	16
won	7	99

logistic regression

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_predict
from sklearn.metrics import classification_report

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(classification_report(y, pred, digits=3,
                            target_names=["win", "loss"]))
```

OUTPUT

```
              precision    recall  f1-score   support

         win      0.861     0.934     0.896       106
        loss      0.731     0.543     0.623        35

    accuracy                          0.837       141
   macro avg      0.796     0.738     0.759       141
weighted avg      0.829     0.837     0.828       141
```

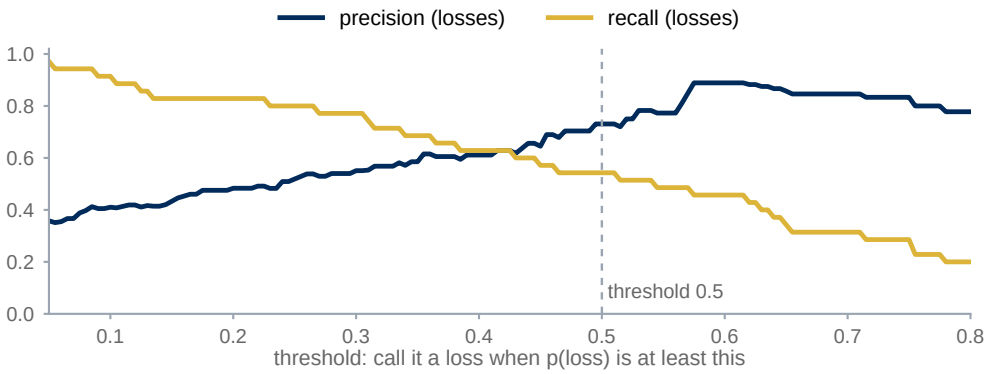
`support` is the number of true samples of each class in the data being scored.

Accuracy vs. Precision vs. Recall

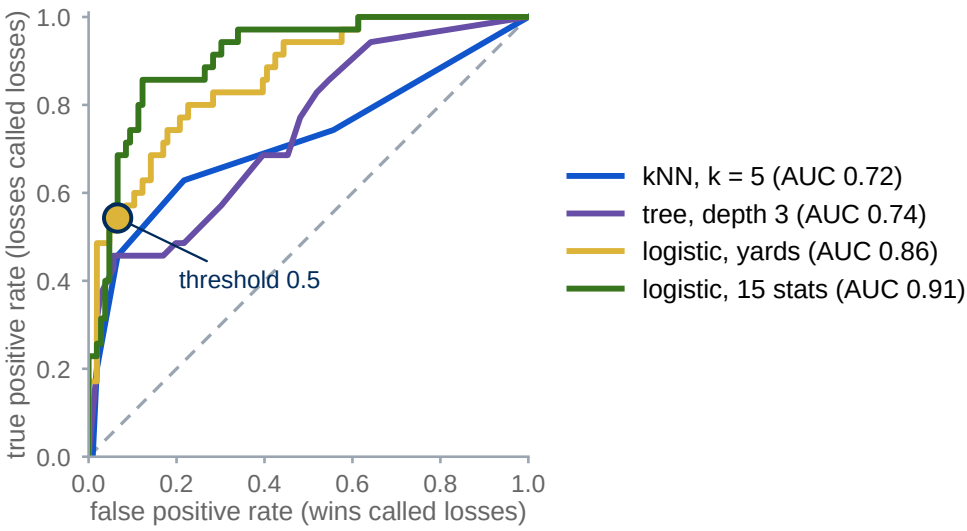
- **Accuracy** - when **class distribution is balanced** and **false positives and false negatives have similar consequences**.
- **Precision** - when **false positives are costly** or undesirable
- **Recall** - when **false negatives are more critical** than false positives
- **F1** - when you need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

Evaluating Decision Thresholds

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

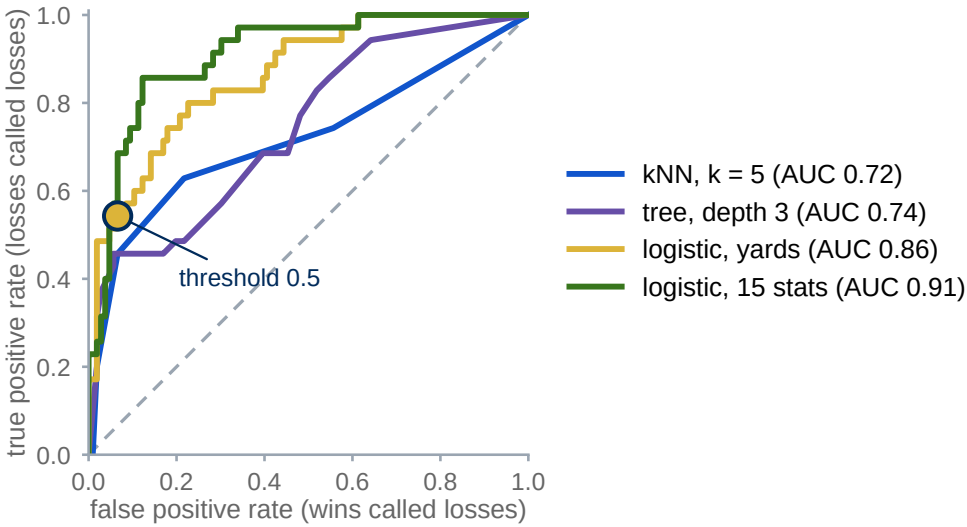


Rather than pick one threshold, we can draw all of them. This is a **receiver operating characteristic (ROC) curve**



ROC and AUC

AUC (area under the **ROC** curve) measures **the area lying below the entire ROC curve**.



To compare **classifiers** we'd like one number instead of a curve. An **AUC** of `1` is a perfect classifier and `0.5` is randomly guessing.

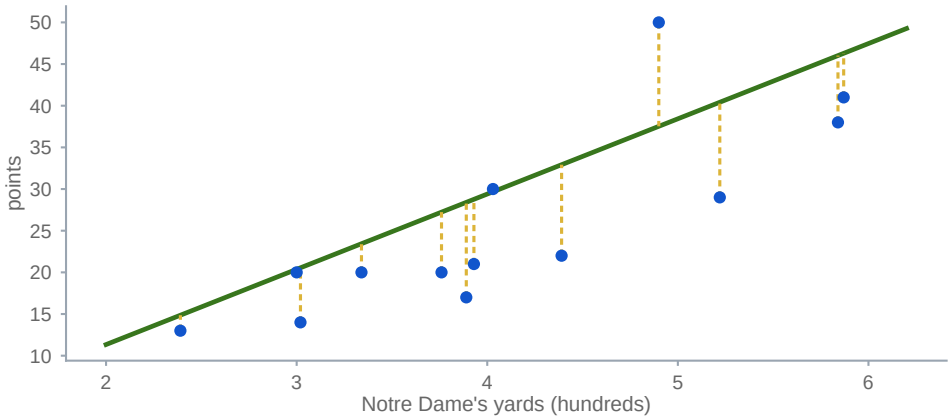
Logistic regression has an **AUC** of `0.86`, while **kNN** has an **AUC** of `0.72`

ROC is a way of visualizing our results while **AUC provides a single number** we could use to compare directly against other classifiers.

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**.

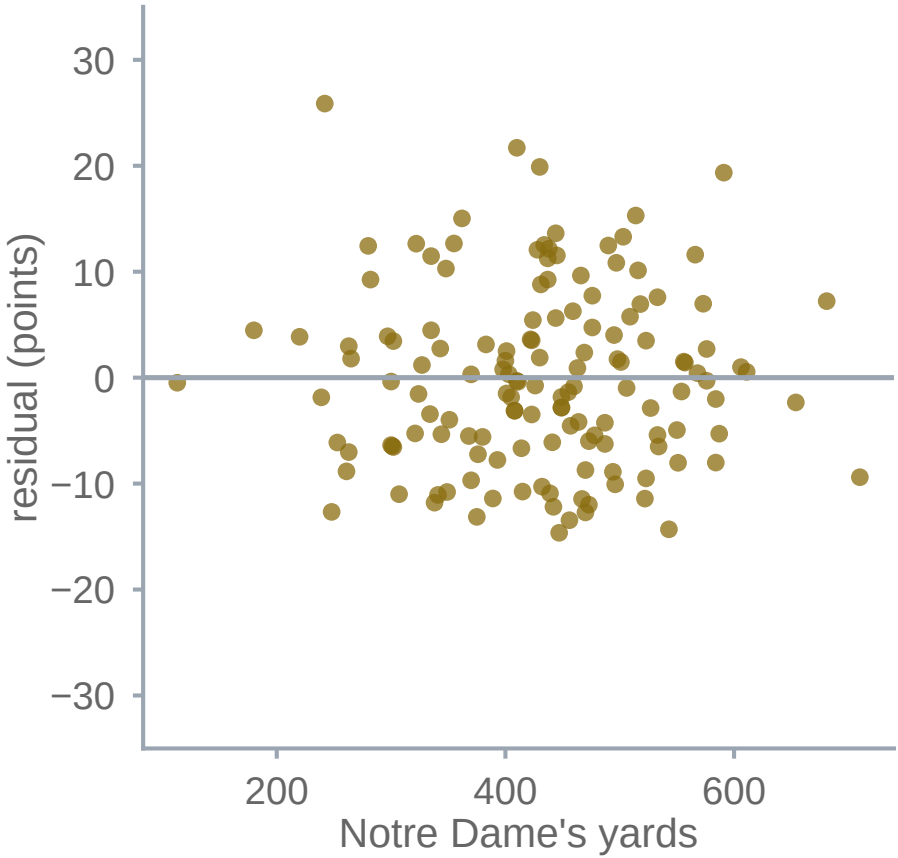


These gaps are called the _____ **(R)**

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

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

the share of the variance the model explains

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

Mean Squared Error

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)

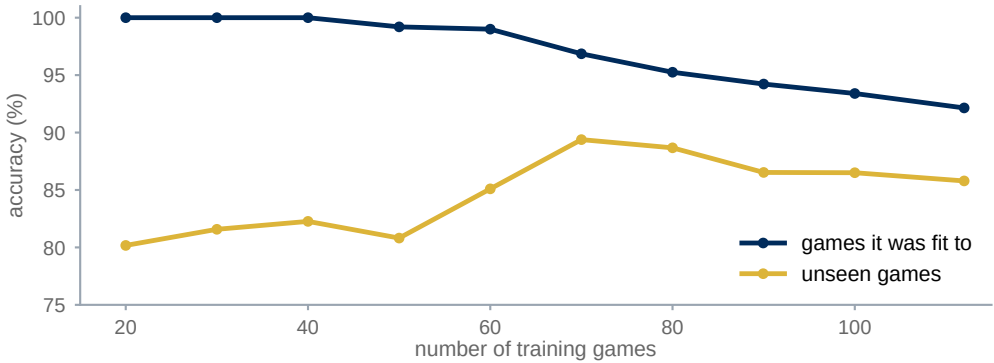
MSE measures **the average squared difference** between the estimated values and the actual value

mean squared error

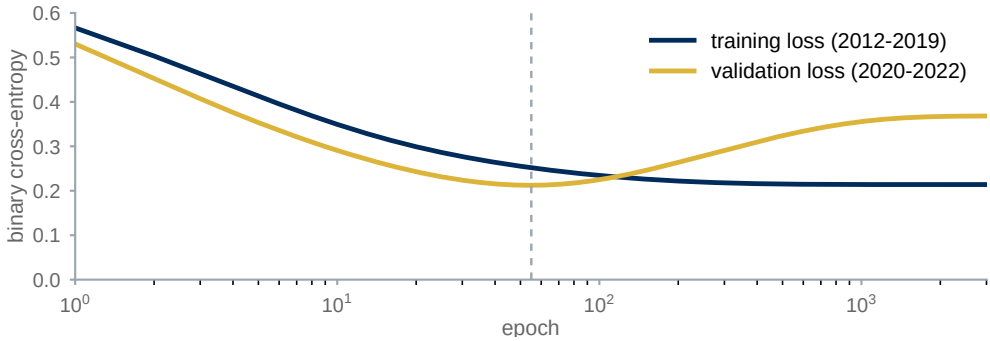
[11.6] $$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Learning Curves - Machine Learning

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



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



Assessing Clustering

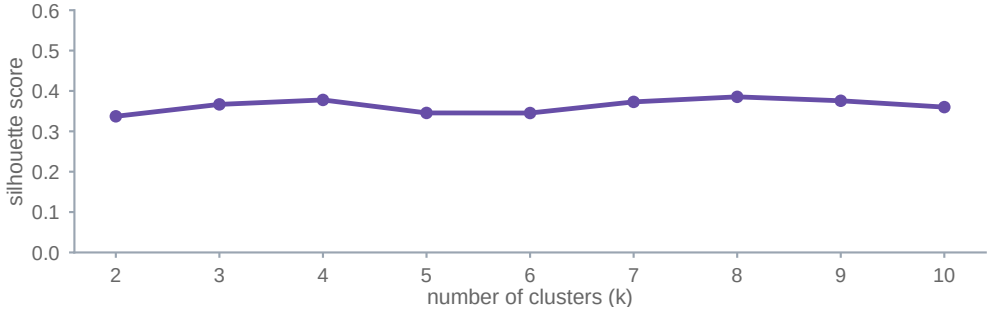
Assessing **unsupervised learning** is difficult as there are **no** _____ against which we can compare results.

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

Silhouette Score

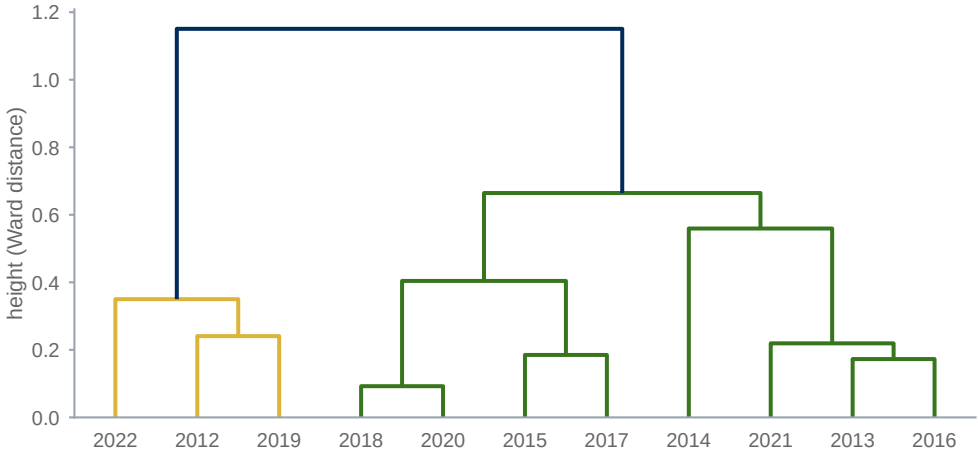
Silhouette score is a numerical value that measures how well a **data point belongs to its cluster** and how **separated it is from other clusters**.

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



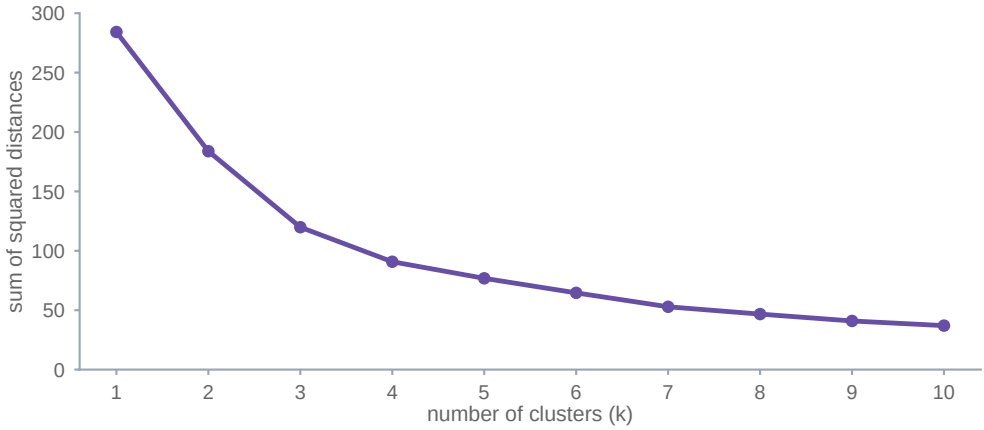
Dendrograms

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



Elbow Plots

Previously, we have used **elbow plots** to estimate the number of **principal components** 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**.