



Artificial Intelligence and Machine Learning (ACCA6011)

Machine learning algorithms

Dr. Seena Joseph

Today's Plan

1. What is a machine learning model?
2. How a model learns: training, testing and predicting
3. Types of learning: supervised, unsupervised and reinforcement
4. Classification vs regression
5. Measuring performance: the confusion matrix

Four short Python activities along the way. How individual algorithms (e.g. linear regression, SVM) work comes in later lectures.

Warm-up: Where Did You Meet ML Today?

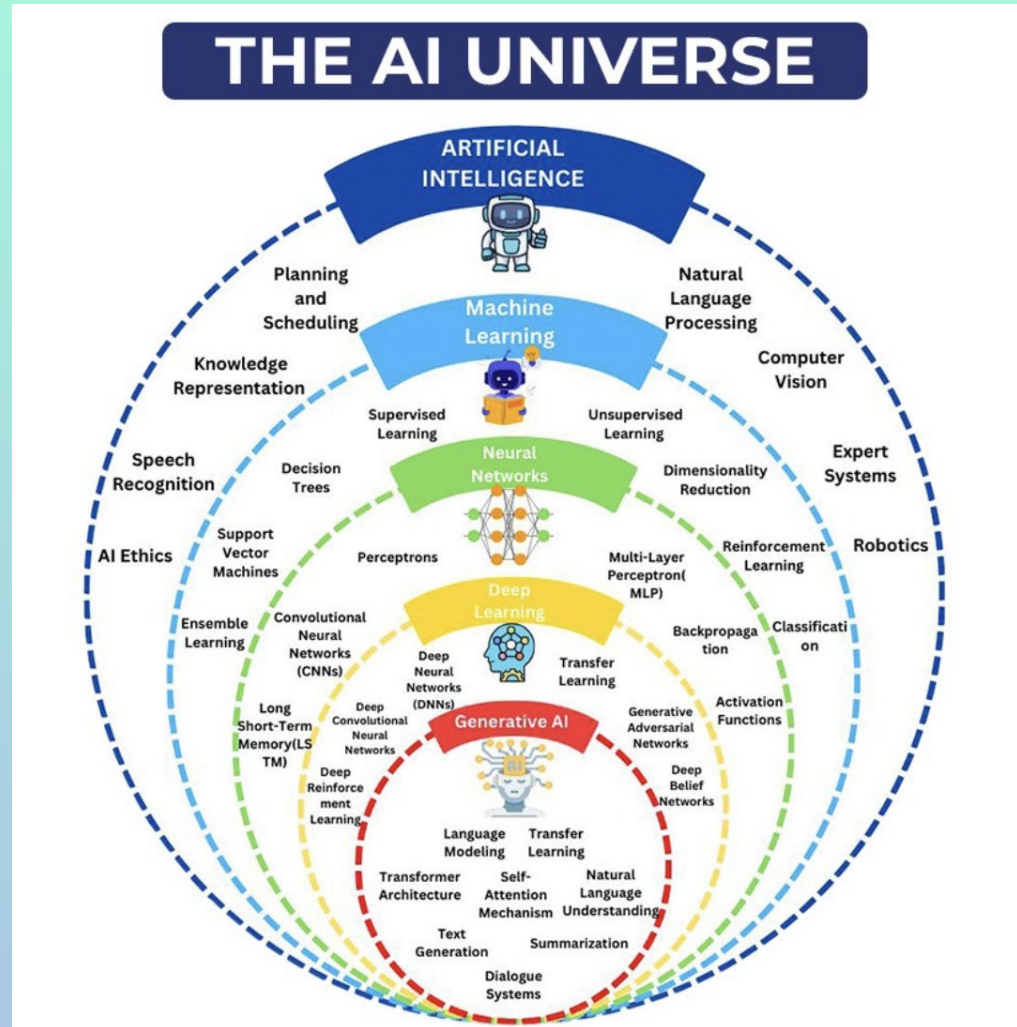
Poll

Which of these have you used today? Vote for all that apply.

- Face unlock on your phone
- “Recommended for you” on Netflix, Spotify or YouTube
- An email moved to your spam folder
- Arrival time in Google Maps
- Matchmaking in an online game
- Predictive text on your keyboard

Keep your answers - we will sort them into classification and regression later.

Artificial Intelligence



What is a Machine Learning Model?

A model is a function learned from data. It takes inputs (features) and produces an output (a prediction).

Instead of writing the rules ourselves, we give the computer examples and it learns the rules.

	We give the computer	The computer produces
Traditional programming	Data + rules	Answers
Machine learning	Data + answers (examples)	Rules (the model)

Once trained, the model makes predictions on new data it has never seen.

Features, Labels and Instances

player_id	session_minutes	ping_ms	character_class	returns_next_week
P001	33	27	Mage	Yes
P002	47	27	Archer	Yes
P003	25	26	Warrior	No

- **Instance (sample):** one row; here, one player
- **Features (X):** the inputs the model learns from
- **Label/target (y):** the answer we want to predict

Which columns are features and which is the label? Should `player_id` be a feature?

How a Model Learns: Train, Test, Predict

1. **Split** the data , usually about 80% for training and 20% for testing
2. **Train (fit)**: the model learns patterns from the training data
3. **Test (predict)**: the model predicts labels for test data it has never seen
4. **Evaluate**: compare its predictions with the true labels
5. **Use**: make predictions on brand-new data

Like practising on past papers (training), then sitting an exam with new questions (testing).

```
model.fit(X_train, y_train)      # train
predictions = model.predict(X_test)  # test
```

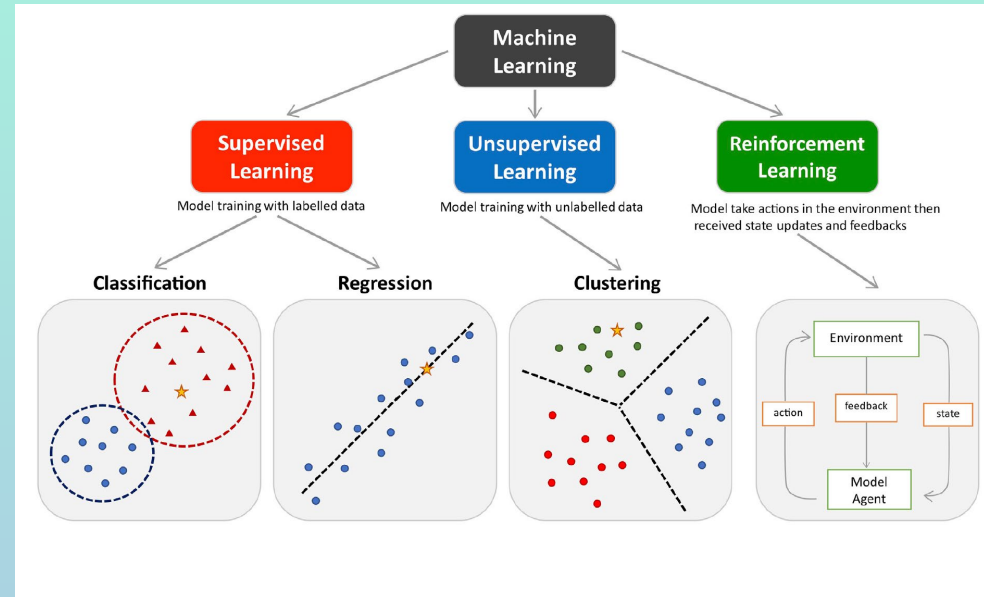
Underfitting and Overfitting

	Underfitting	Good fit	Overfitting
What happens	Too simple : misses the pattern	Learns the real pattern	Memorises the training data, including noise
Training score	Low	High	Very high (often 100%)
Test score	Low	High	Noticeably lower
Exam analogy	Didn't revise enough	Understood the topic	Memorised past-paper answers

A model is only useful if it works on data it has not seen – so always judge it on the test set.

Machine Learning algorithm

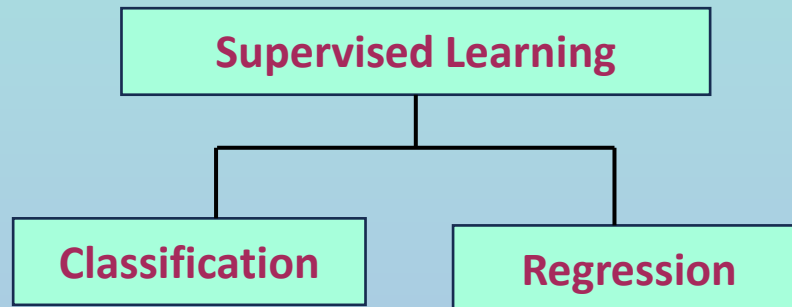
- Supervised learning
- Unsupervised learning
- Reinforcement learning



Supervised Learning

Supervised learning algorithm is trained on a labelled dataset

- Labelled data-> Model training-> Prediction (**Classification and Regression**)
- Classification – prediction of output of discrete values [predicting Classes]
- Regression – prediction of output of continuous values [**Predicting Values**]



Supervised Learning

Classification and Prediction

Regression



price: 124 032\$

Classification



category: cookie

Supervised Learning

Classification

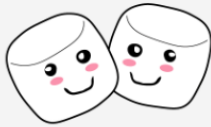
Training set



weight: 62
width: 14
is_cookie: 1



weight: 45
width: 12
is_cookie: 1



weight: 12
width: 6
is_cookie: 0

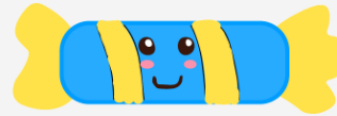


weight: 59
width: 12.5
is_cookie: 1



weight: 16
width: 5 } features
is_cookie: 0 } target

New instance

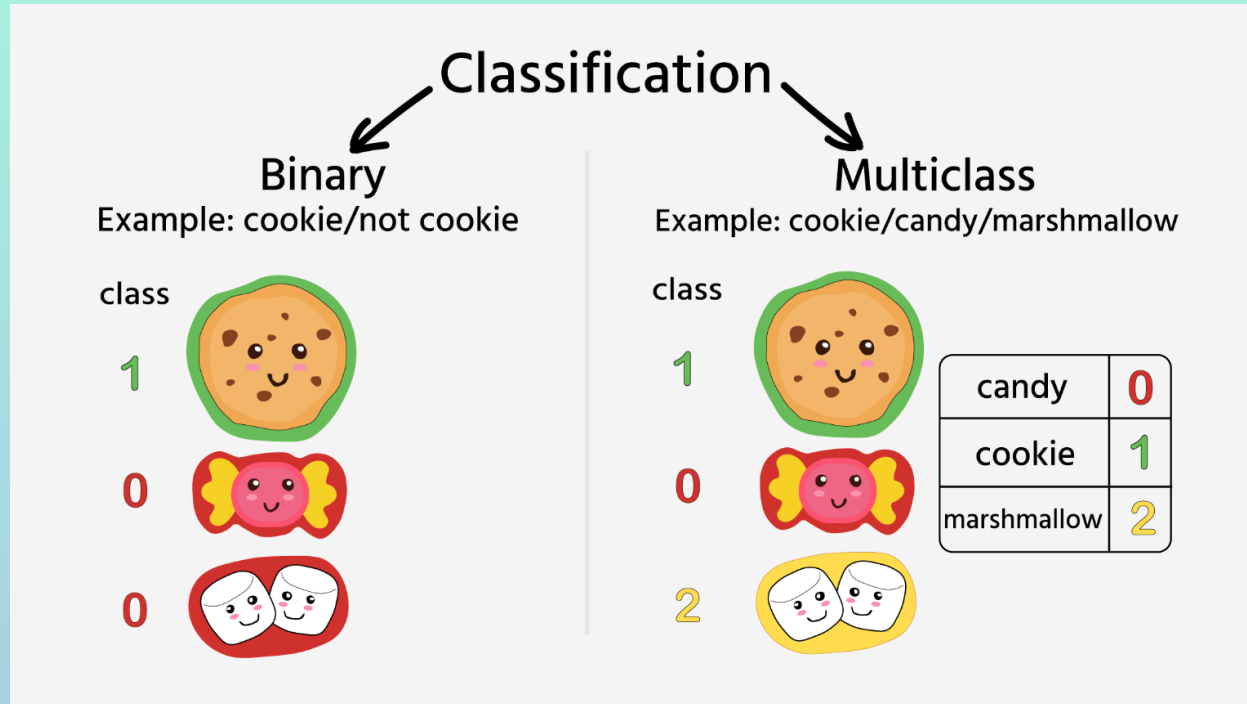


weight: 26
width: 9
is_cookie: ???

need to predict

Supervised Learning

Classification – Binary or Multiclass

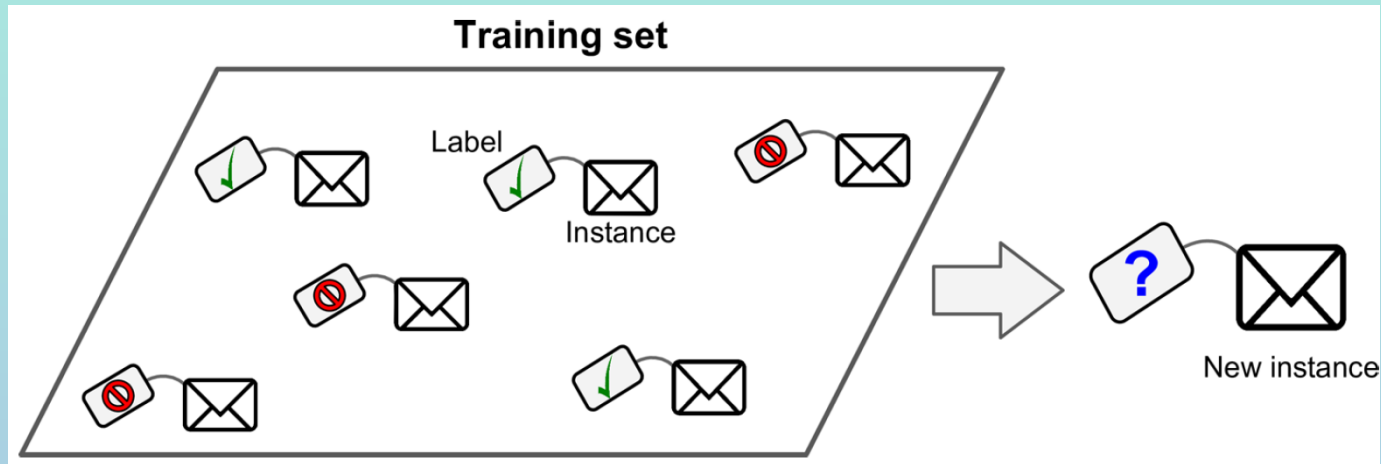


Supervised Learning

The training set contains the desired solution.

Spam filter is a good example for supervised classification.

It is trained with many example emails with their classes (Spam or Ham) to learn how to classify new emails.



Is this binary or multiclass ?

Classification or Regression?

Activity

Is the model predicting a category or a number?

1. Will this customer cancel their subscription? (Yes / No)
2. How many minutes will this player play tomorrow?
3. Which digit (0–9) is in this handwritten image?
4. What will the temperature be at noon tomorrow?
5. Is this network connection normal or an attack?
6. What price will this second-hand laptop sell for?

Bonus: which of the classification questions are binary and which are multiclass?

Supervised Learning

Example for supervised learning algorithms

- Linear Regression
- Logistic Regression
- Support Vector Machine
- K-Nearest Neighbor
- Decision Tree
- Random Forest
- Naïve Bayes
- Neural Networks

Python Activity 1: Explore the Data

```
[12]: from sklearn.datasets import load_breast_cancer
X, y = load_breast_cancer(return_X_y=True, as_frame=True)
print(X.shape)
print(y.value_counts())
```

```
(569, 30)
target
1    357
0    212
Name: count, dtype: int64
```

```
[18]: X.head()
```

```
[18]:
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	compa
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	184.60	2019.0	0.1622	
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	158.80	1956.0	0.1238	
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50	1709.0	0.1444	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	98.87	567.7	0.2098	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	152.20	1575.0	0.1374	

5 rows × 30 columns

Python activity

Dataset: 569 tumours, 30 measurements each. Label: 0 = malignant, 1 = benign.

1. How many instances and features are there?
2. Is this classification or regression? Binary or multiclass?
3. Is the data balanced between the two classes?

Python Activity 2: Train and Test a Model

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)          # the model learns

print(model.score(X_train, y_train)) # Train
print(model.score(X_test, y_test))   # how accurate on new data?

1.0
0.9473684210526315
```

Python activity

Today the model is a “black box”. We only train it, test it and compare.

1. How many rows are in the training and test sets?
2. Compare training and test accuracy. Any sign of overfitting?
3. Try this with another model:
from sklearn.neighbors import KNeighborsClassifier

model =
KNeighborsClassifier()

Unsupervised Learning

- The purpose is to either transform the data or discover pattern without the need of a set of explicit labeled data.
- The training data is unlabeled

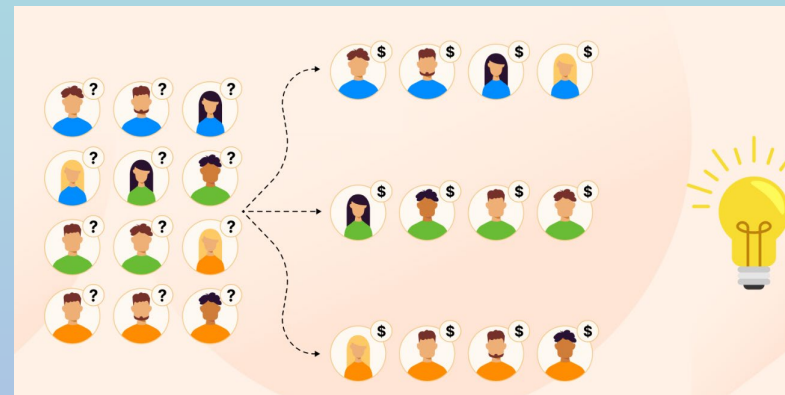
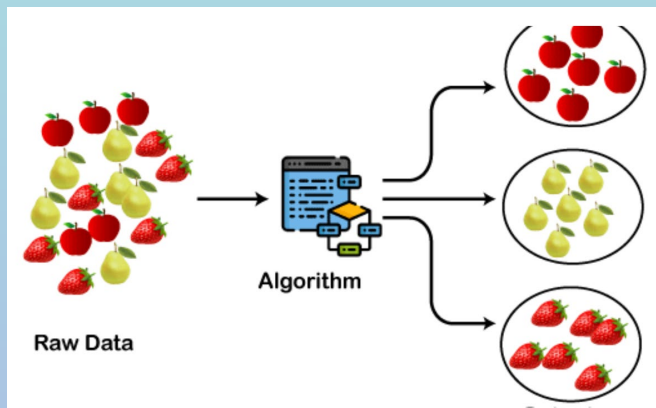
Common Unsupervised Learning algorithm

- Clustering
- Dimensionality Reduction
- Association rule learning

Unsupervised Learning

Clustering : Group similar instance into clusters.

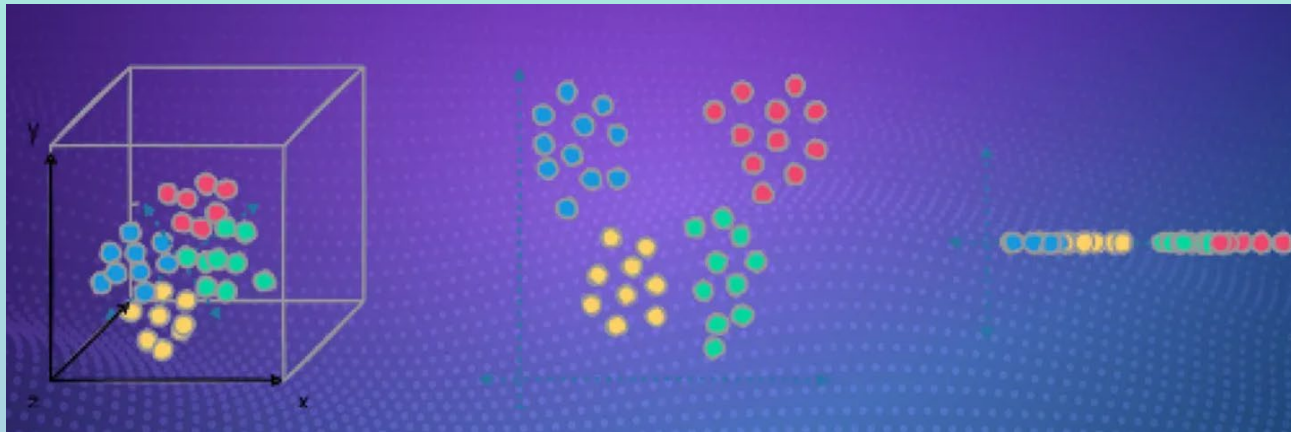
- Examples: K Means, Fuzzy C Means, DBSCAN, MeanShift, Spectral Clustering etc.
- Applications
 - Customer segmentation – cluster customers based on the activity on the website or their purchases
 - For data analysis – easy to analyse the data in each cluster separately



Unsupervised Learning

Dimensionality reduction : Reducing the number of dimensions. Reduce the number of features/dimensions while keeping the essential information.

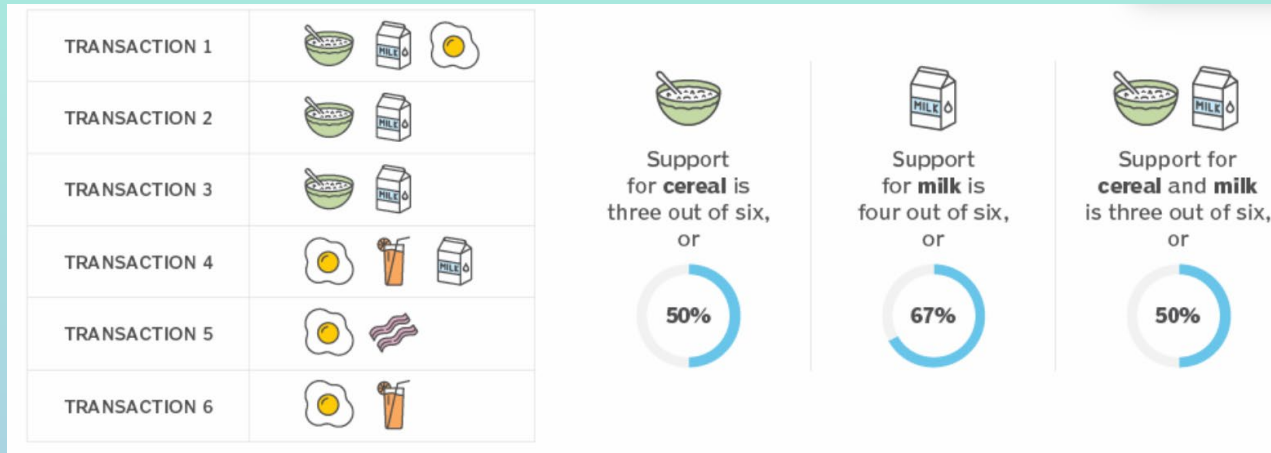
- Example: Principal Component Analysis (PCA)



Unsupervised Learning

- Association rule learning : Discover interesting relationship variables in a dataset.

Example : Apriori, Eclat



Reinforcement Learning

- An agent learns by trial and error: it takes actions in an environment and receives rewards or penalties.
- There are no labelled answers. Over time it learns which actions lead to the most reward.

Term	Meaning	Racing-game example
Agent	The learner that makes decisions	The AI car
Environment	The world the agent acts in	The race track
Action	What the agent can do	Steer, brake, accelerate
Reward	Feedback after an action	+ ve points for speed, – ve points for crashing

Examples: AlphaGo learning Go by playing itself, robots learning to walk, game characters learning to race.

Which Type of Learning?

Activity

Pairs, 2 minutes. Decide which type of learning each one uses.

1. Predict whether a bank transaction is fraudulent, using past transactions labelled fraud / not fraud
2. Group players into play styles from their in-game behaviour (no labels)
3. A game character learns to win races by trial and error, earning points for good moves
4. Predict a house price from its size, location and number of bedrooms
5. Find products that are often bought together in a supermarket
6. A robot vacuum learns the best route around a room

Tip: ask “does the training data contain the correct answers?”

ML Model - Performance Measures

Confusion Matrix

A confusion matrix is a table used to evaluate the performance of a classification model. It shows how well the model's predicted labels match the actual (true) labels.

A binary classification problem [YES/NO]

Is this transaction Fraudulent ?

		<u>Predicted</u> Fraud	
		Y	N
<u>Is Actual</u> Fraud	Y	+ / + TP	- / + FN
	N	+ / - FP	- / - TN

ML Model - Performance Measures

Confusion Matrix

		<u>Predicted</u> Fraud	
		Y	N
<u>Is Actual Fraud</u>	Y	+ / + TP	- / + FN
	N	+ / - FP	- / - TN

Case 1 [True Positive]: A transaction was fraud, and the classifier correctly labelled it as fraud.

Case 2 [False Positive]: A transaction was not fraud and the classifier incorrectly labelled it as fraud. [Type I Error]

Case 3 [False Negative]: A transaction was fraud, and the classifier incorrectly labelled it as not fraud. [Type II Error]

Case 4 [True Negative]: A transaction was not fraud and the classifier correctly labelled it as not fraud.

ML Model - Performance Measures

Precision

		Predicted Fraud	
		Y	N
Is Actual Fraud	Y	2 TP	4 FN
	N	8 FP	86 TN

Precision – What proportion of the positive identification was actually correct ?

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

$$\text{Example, Precision} = \frac{2}{2+8} = \frac{2}{10} = 0.2$$

What percentage of the time an instance labelled as positive (predicted as positive) was actually positive ?

FP should be low for a high Precision.

ML Model - Performance Measures

Recall

		<u>Predicted</u> Fraud	
		Y	N
<u>Is Actual</u> Fraud	Y	2 TP	4 FN
	N	8 FP	86 TN

Recall – what proportion of the actual positives were identified correctly ?

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

$$\text{Example, Recall} = \frac{2}{2+4} = \frac{2}{6} = 0.3$$

How **sensitive** the classifier is the indicators of possible fraud?

FN should be low for a High Recall.

ML Model - Performance Measures

F1 Score – the harmonic mean of Precision and Recall

Low Recall (Miss Some Fraud)
High Precision (Few False Positive)

High Recall (Catch Most Fraud)
Low Precision (Many False Positive)



Conservative
Flag fewer transaction

Aggressive
Flag more transaction

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

ML Model - Performance Measures

Accuracy - it is the ratio of the correct predictions made by a classification model out of the total predictions. It provides an overall idea of how often the model is correct.

The correct prediction is sum of TP and TN.

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

$$= \frac{2+86}{2+8+86+4} = .88 \quad (88\%)$$

	Y	N
Y	2 TP	4 FN
N	8 FP	86 TN

Is This Cheat Detector Any Good?

Activity

An anti-cheat system checked 1,000 players in an online game.

	Predicted cheat	Predicted not cheat
Actually cheating	18 (TP)	2 (FN)
Not cheating	30 (FP)	950 (TN)

Calculate:

- Accuracy
- Precision
- Recall
- F1 score

Accuracy looks excellent. Is the system good? Who is affected by the 30 false positives, and what would you change?

ML Model - Performance Measures

Activity

1. "Imagine you're developing a test to detect a rare but dangerous disease. Would it be more important for this test to have high precision or high recall? Why?"
2. For an email spam filter, would you prefer high precision or high recall, and why?"
3. "Imagine you are developing an automated hiring system that shortlists candidates. Would it be more beneficial to prioritize high precision or high recall?"
4. "For a credit card fraud detection system, would high recall or high precision be more important, and why?"

Python Activity 4: The Confusion Matrix

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

```
y_pred = model.predict(X_test)
```

```
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[40  3]
 [ 3 68]]
```

Python activity

Rows = actual class, columns = predicted class. Class 0 = malignant, class 1 = benign.

1. How many malignant tumours were predicted as benign? Why is this the most dangerous error?
2. Find the precision and recall for class 0 in the report.
3. For this test, which matters more: precision or recall?

Extra reading



[Embracing the AI Universe: Insights and Innovation - Lucid Analytics](#)

[User Guide — scikit-learn 1.9.1 documentation](#)

Reference

Géron, Aurélien. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems*. Third edition. Sebastapol, CA: O'Reilly Media, Inc., 2023. Print.