

Lecture 3: Classification and Regression- Supervised Learning

Sinuo Wu

Course: AI for Business Applications (AI3000)

EXPERT INSIGHT

Artificial Intelligence with Python

Your complete guide to building
intelligent apps using Python 3.x

Second Edition

Alberto Artasanchez
Prateek Joshi

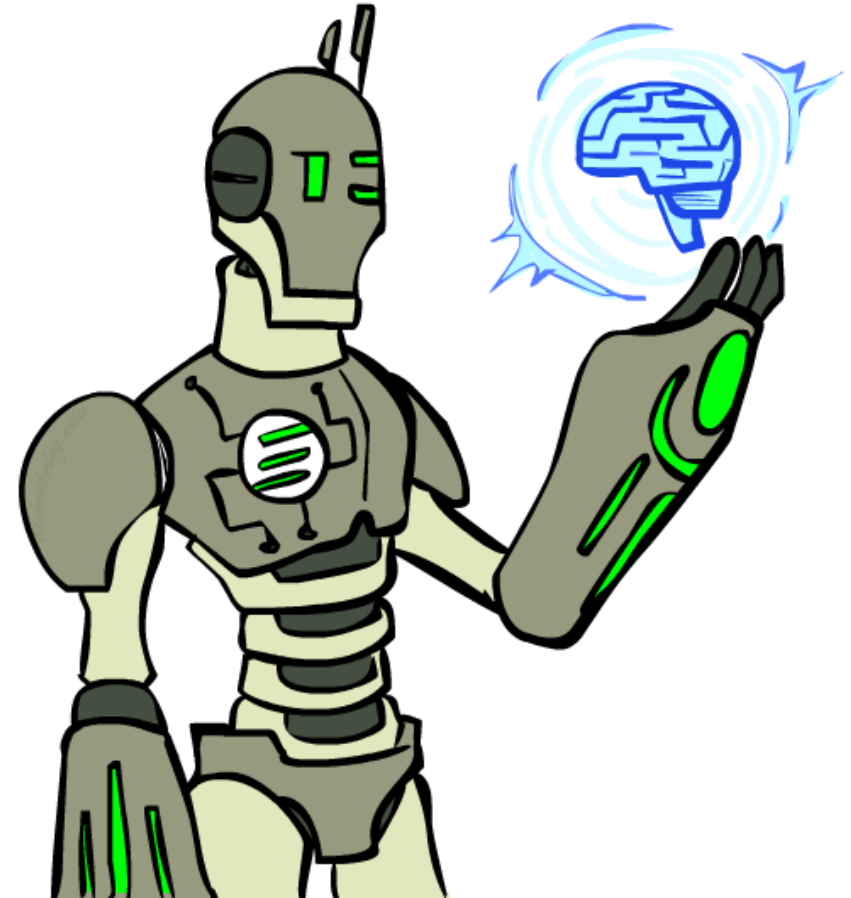
Packt

Quiz

- Enter Room number or Scan the QR code

Today

- Supervised vs Unsupervised Learning
- Supervised learning
 - The k-nearest neighbors (KNN)
 - Naïve Bayes
 - Support Vector Machines (SVM)
 - Decision Trees
- Confusion Matrix



Supervised vs Unsupervised Learning

Supervised vs Unsupervised Learning

- Supervised = Task Driven (regression/classification)

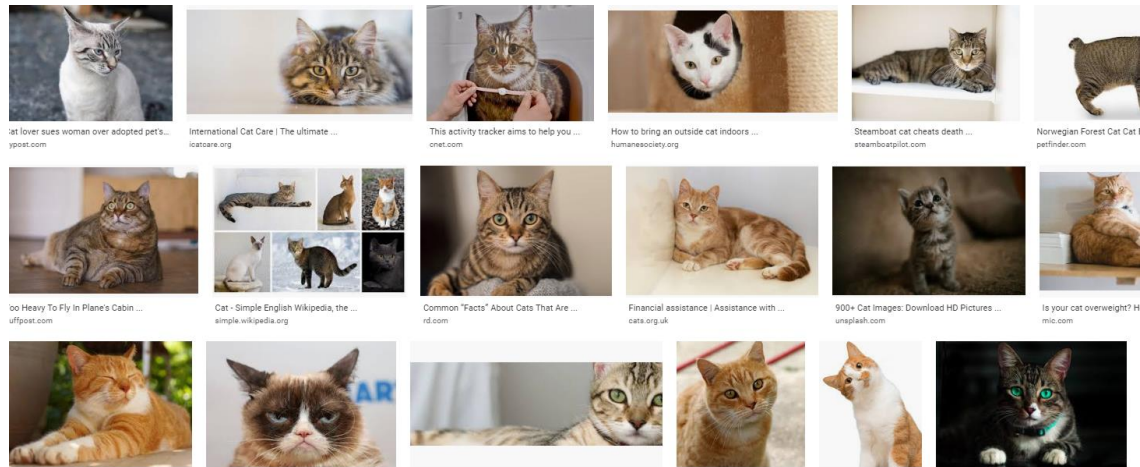
Supervised learning refers to the process of building a machine learning model that is based on **labeled training data**.

- Unsupervised = Data driven (clustering)

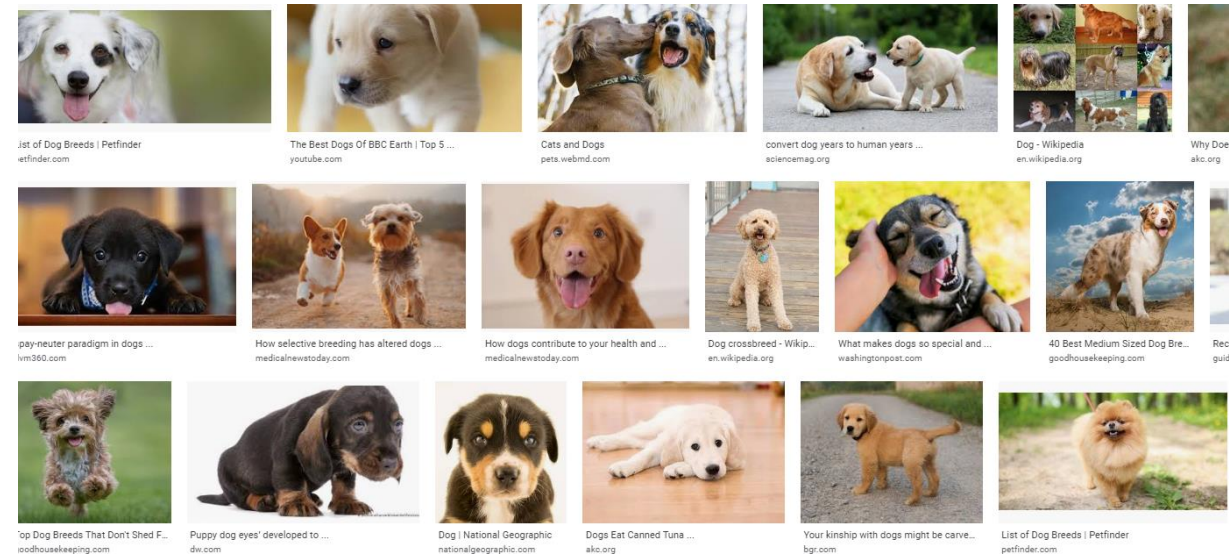
Unsupervised learning refers to the process of building a machine learning model **without relying on labeled training data**.

Supervised learning

Cats



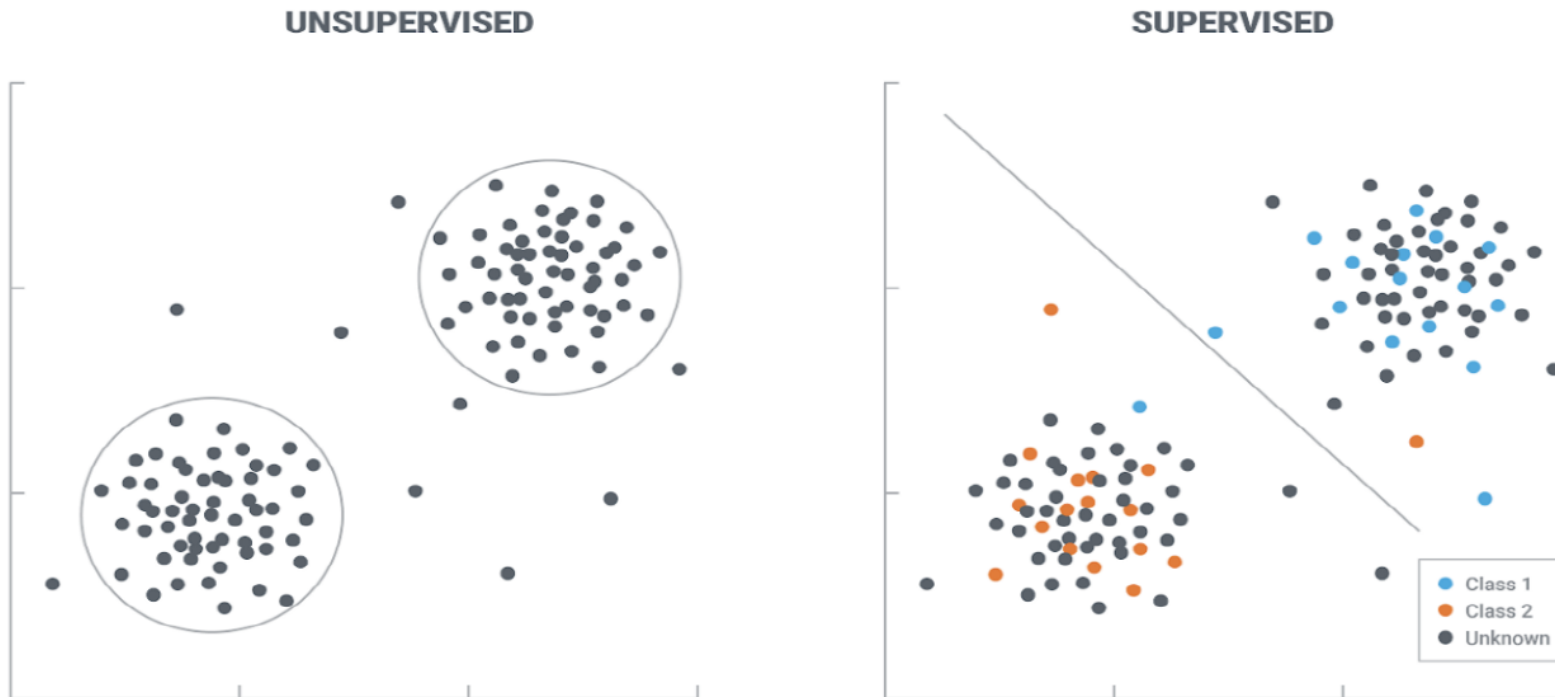
Dogs



Unsupervised learning

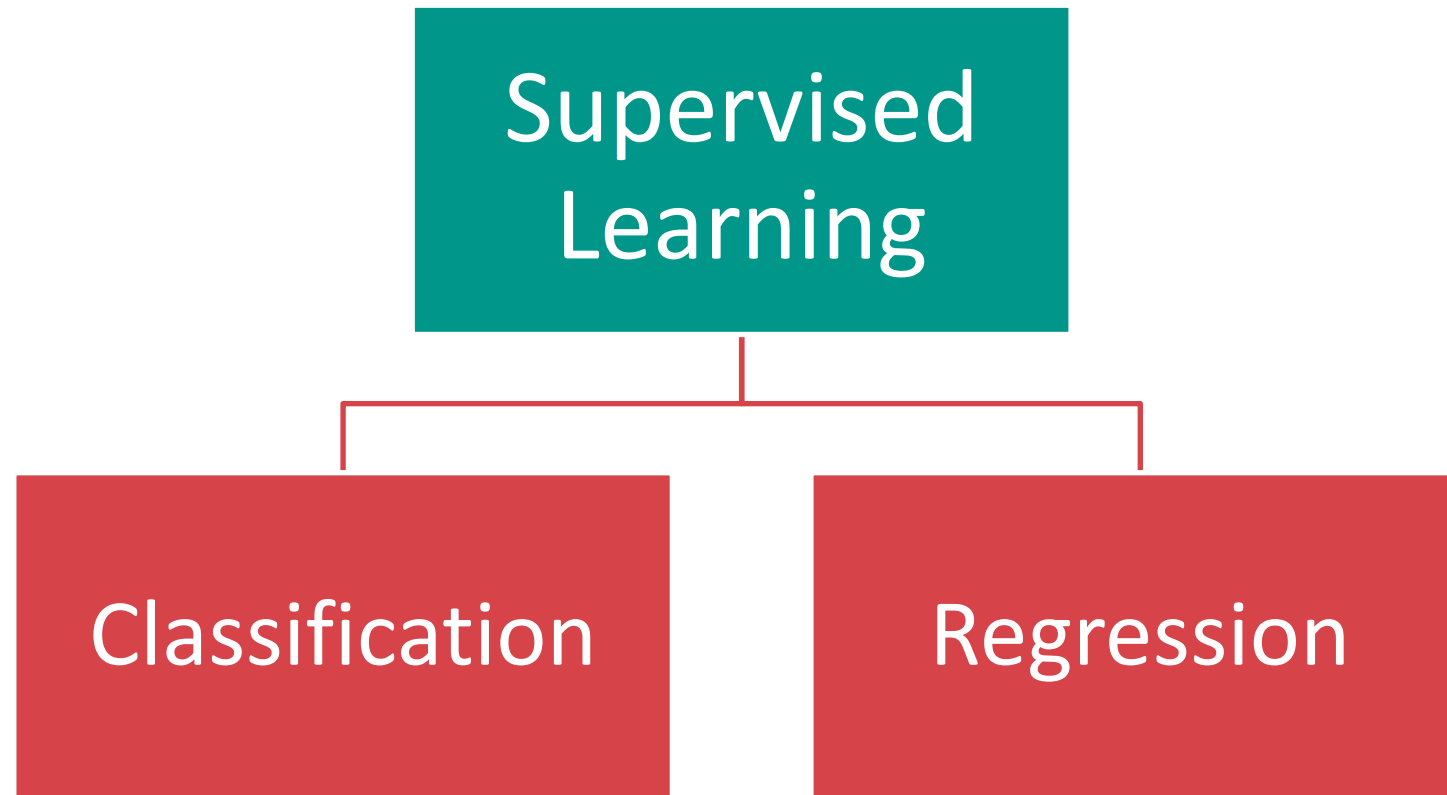


Supervised vs unsupervised



Supervised Learning

Supervised Learning



Classification

Classification: This is a type of problem where we predict the *categorical response* value where the data can be separated into specific “**classes**” (ex: we predict one of the values in a set of values). classification solves the problem of identifying the category to which a new data point belongs.

Some examples are :

- Is this mail spam or not?
- Will it rain today or not?
- What type of animal it is?

Basically ‘Yes/No’ type questions called **binary classification**.

Regression

- ***Regression:*** This is a type of problem where we need to predict the *continuous-response* value

Some examples are:

- what is the price of house in a specific city?
- what is the value of the stock?
- how many total runs can be on board in a cricket game?

The k -nearest neighbors

KNN

- Used to classify data based on closest or neighboring training examples in the given region.
- Widely applied for classification
- For a new input, the K nearest neighbors are calculated and the majority among the neighboring data decides the classification for the new input.
- A sample has **K** most similar samples in the feature space. If most of these samples belong to a certain category, then the sample also belongs to this category.

KNN

$$\sqrt{(a1 - b1)^2 + (a2 - b2)^2 + (a3 - b3)^2}$$

Name	Fight scenes	Kiss scenes	Type of the film
California man	3	104	Romance movie
He is not really into dues	2	100	Romance movie
Beautiful woman	1	81	Romance movie
Kevin Long blade	101	10	Action Movie
Robo Slayer 3000	99	5	Action Movie
Amped	98	2	Action Movie
?	18	90	Unknown

Name	Distance to the unknown film
California man	20.5
He is not really into dues	18.7
Beautiful woman	19.2
Kevin Long blade	115.3
Robo Slayer 3000	117.4
Amped	118.9

ML Steps

- Step 1: Load the data
- Step 2: Preprocess the data (e.g. handling missing value, drop unnecessary columns)
- Step 4: Split the data into training and testing sets
- Step 5: Standardize the features
- Step 6: Train and predict using **k-Nearest Neighbors**
- Step 6: Evaluate the model
- Step 7: Conduct and adjust

Split data

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

knn_classifier = KNeighborsClassifier(n_neighbors=3)
knn_classifier.fit(X_train, y_train)

y_pred = knn_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

Case Study

Ever wonder what it's like to work at Facebook? Facebook and Kaggle are launching a machine learning engineering competition for 2016. Trail blaze your way to the top of the leaderboard to earn an opportunity at interviewing for one of the 10+ open roles as a software engineer, working on world class machine learning problems.






The goal of this competition is to predict which place a person would like to check in to. For the purposes of this competition, Facebook created an artificial world consisting of more than 100,000 places located in a 10 km by 10 km square. For a given set of coordinates, your task is to return a ranked list of the most likely places. Data was fabricated to resemble location signals coming from mobile devices, giving you a flavor of what it takes to work with real data complicated by inaccurate and noisy values. Inconsistent and erroneous location data can disrupt experience for services like Facebook Check In.

<https://www.kaggle.com/competitions/facebook-v-predicting-check-ins>

Data Explorer

857.5 MB

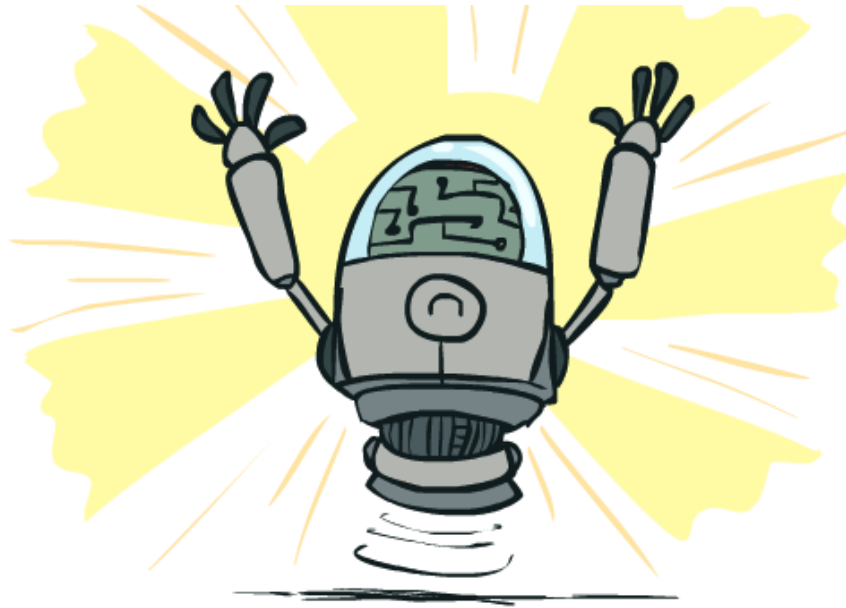
-  sample_submission.csv.zip
-  test.csv.zip
-  train.csv.zip

Download from Canvas

Have a Break!

Practice

TRY THE COMPETITION



Report

One representative from each group report for:

1. What is your results
2. Explain your results



Example

- Coding

Naive Bayes

Naïve Bayes

- **Naïve Bayes** is a technique used to build classifiers using Bayes Theorem. Bayes Theorem describes the **probability** of an event occurring based on different conditions that are related to this event.

$$P(A | B) = \frac{P(B | A) \cdot P(A)}{P(B)}$$

A, B = events

$P(A|B)$ = probability of A given B is true

$P(B|A)$ = probability of B given A is true

$P(A), P(B)$ = the independent probabilities of A and B

Applications of Naive Bayes

- Real time predictions
- Multiclass prediction
- Text classification, spam filtering, sentiment analysis.
- Recommendation systems

Strength: Stable, Fast, Relative high accuracy

Weakness: When A is not independent of B...

Case of Naive Bayes

Category

'sci.med': Represents the category related to discussions about medical topics.'

soc.religion.christian': Represents the category related to discussions about Christianity and religious topics.

'comp.graphics': Represents the category related to discussions about computer graphics and related topics.

'rec.sport.baseball': Represents the category related to discussions about baseball and related topics.

sci.crypt
sci.electronics
sci.med
sci.space
soc.religion.christian
talk.politics.guns
talk.politics.mideast
talk.politics.misc
talk.religion.misc

alt.atheism
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey

```
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

# Load the 20 Newsgroups dataset
categories = ['sci.med', 'soc.religion.christian', 'comp.graphics', 'rec.sport.baseball']
data = fetch_20newsgroups(categories=categories)

# Extract the features (text) and target variable (labels)
X = data.data
y = data.target

# Vectorize the text data
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Multinomial Naive Bayes classifier
clf = MultinomialNB()

# Train the classifier
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Print the predicted labels
print("Predicted labels:", y_pred)

# Calculate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```


Decision Trees

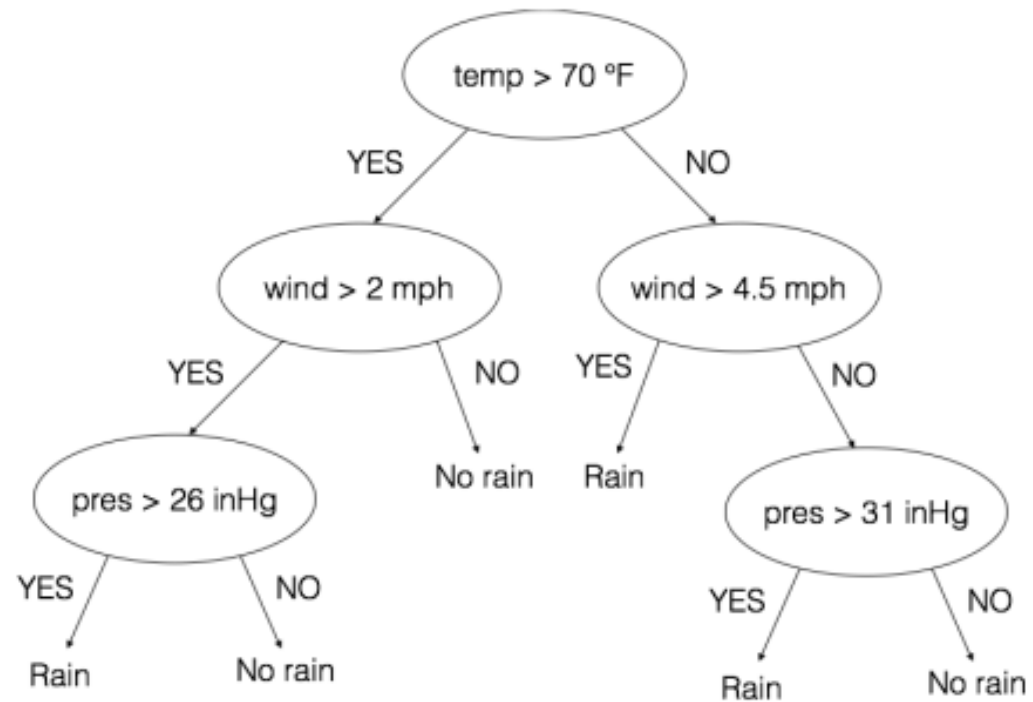
Decision Trees

- A **decision tree** is a way to partition a dataset into distinct branches.
- The branches or partitions are then traversed to make simple decisions.
Decision trees are produced by training algorithms, which identify how to split the data in an optimal way.

$$H(D) = - \sum_{k=1}^K \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

$$g(D, A) = H(D) - H(D|A)$$

Decision Trees



How to make efficient decisions?

Decision Trees

```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

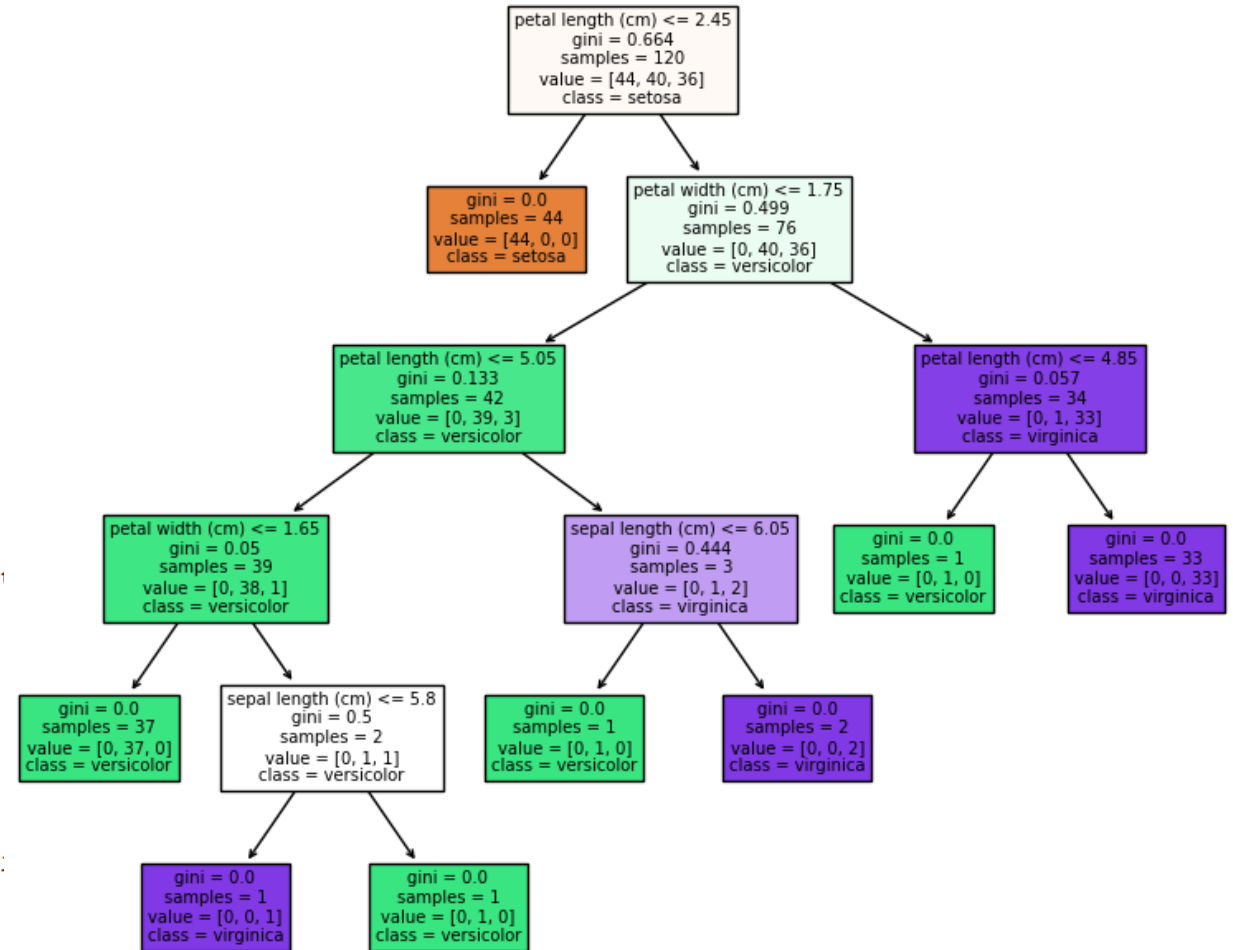
# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s

# Create a Decision Tree classifier
dt = DecisionTreeClassifier()

# Train the classifier on the training data
dt.fit(X_train, y_train)

# Visualize the decision tree
plt.figure(figsize=(10, 8))
plot_tree(dt, feature_names=iris.feature_names, class_names=iris.target_names, fi
plt.show()
```



Decision Trees

- Strength: Clear at each steps
- Weakness: If the data is too complex, it will be difficult to build the tree.
(At this situation we can try random forest)

Case Study 2

- Survive from the Titanic

Steps?

What features are important?

How to build the model with decision tree?

Use your model to predict if Rose would survive

How bout Jack?



Random Forests

What are random forests

- An **ensemble learning method**.
 - Building multiple models and combining them for better results
 - Individual models might become biased or overfit the training data, ensemble learning method reduce the overall risks.
- Individual models are constructed using decision trees.
- For both classification and regression tasks.
- Combines the predictions of multiple decision trees to improve the accuracy and robustness of the model.

Example

- Try case study 2 with random forests

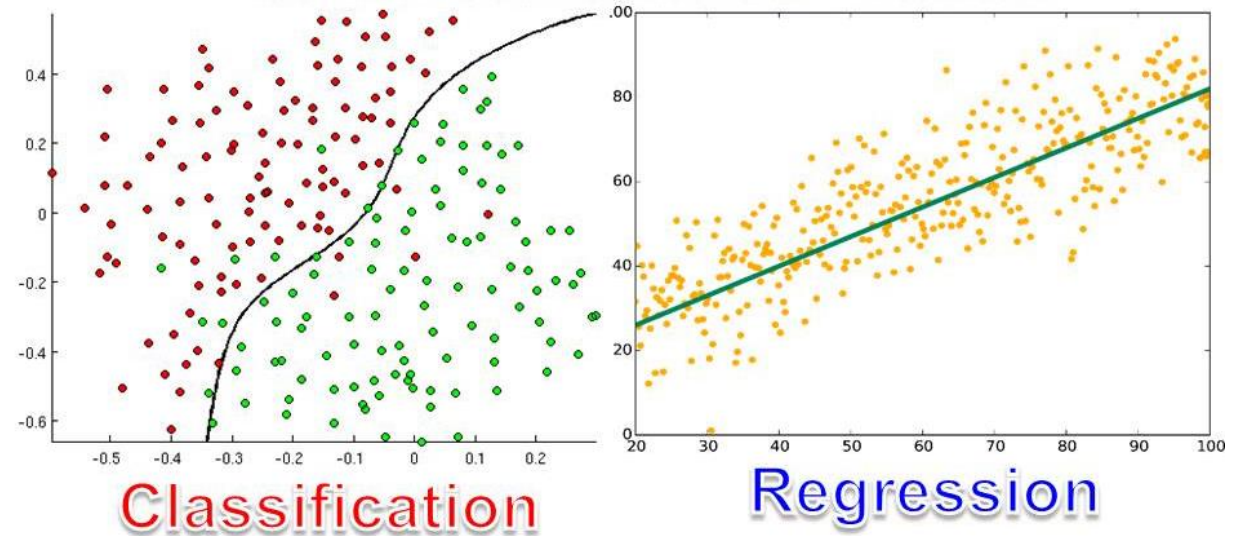
Have a Break!

Regression

Classification VS Regression

Classification> The objective is to assign each of the input vector to one of a given number of discrete categories. An algorithm that implements classification is known as a classifier.

Regression> The objective is to find the relationship among the input variables. Regression analysis helps in understanding how the dependent variable changes with respect to the independent variables. An algorithm that implements regression is known as a regressor.



What is regression

- The process of estimating the relationship between input and output variables. (Statistical model)
 - **Linear regression:** The relationship between a dependent variable (also known as the target) and one or more independent variables (often referred to as features or predictors) is linear.
 - **Logistic regression:** Yes or No tasks
- Frequently used for the prediction of prices, economics and so on.

Logistic Regression

Example – Cancer Prediction

1. Load the dataset
2. Data processing (Define data into features and target)
3. Split data into train and test sets
4. Standardize the data using “StandardScaler”
5. Create and train a logistic regression model
6. Make predictions in the test set
7. Accuracy
8. Save the trained model

7. Attribute Information: (class attribute has been moved to last column)

#	Attribute	Domain

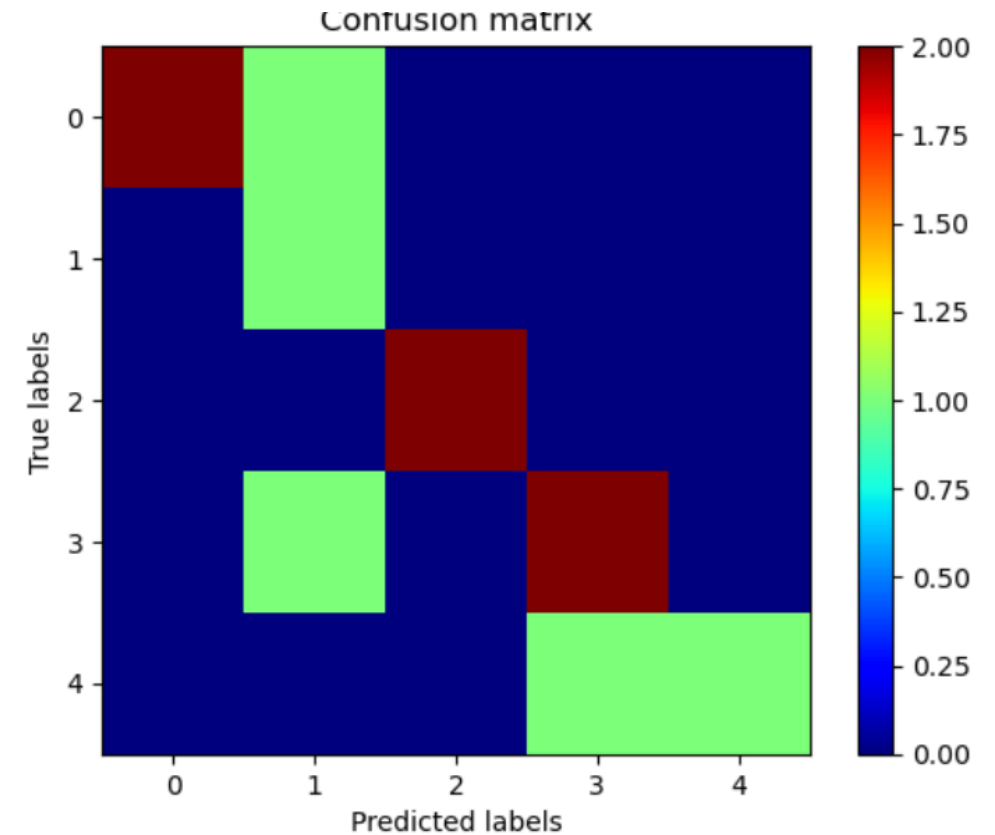
1.	Sample code number	id number
2.	Clump Thickness	1 - 10
3.	Uniformity of Cell Size	1 - 10
4.	Uniformity of Cell Shape	1 - 10
5.	Marginal Adhesion	1 - 10
6.	Single Epithelial Cell Size	1 - 10
7.	Bare Nuclei	1 - 10
8.	Bland Chromatin	1 - 10
9.	Normal Nucleoli	1 - 10
10.	Mitoses	1 - 10
11.	Class:	(2 for benign, 4 for malignant)



Confusion Matrix

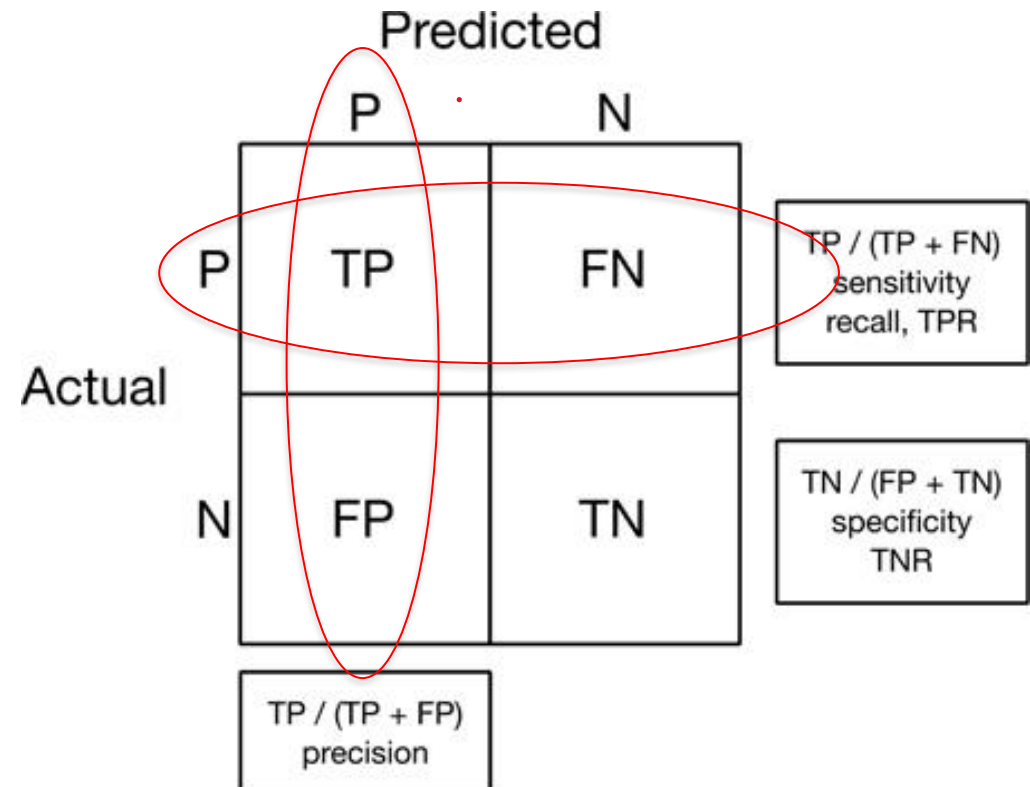
Confusion Matrix

- A figure or table used to describe the performance of a classifier.
- Extracted from the test dataset for which the ground truth is known.
- Each class is compared with every other class and see how many samples are misclassified



Confusion Matrix

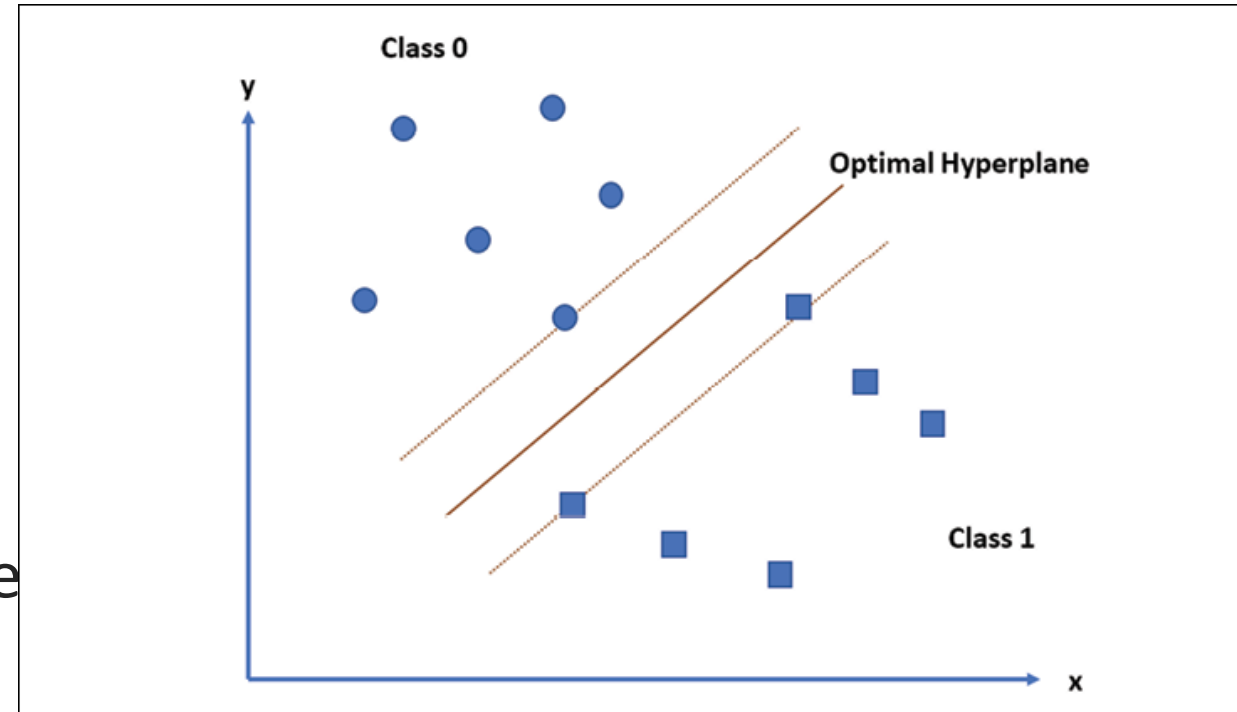
	Predict malignant (Positive)	Predict benign (Negative)
Person has malignant	True (predicted) Positive	False (predicted) Negative
Person has benign	False (predicted) Positive	True (predicted) Negative



Support Vector Regressor

Support Vector Machine

- A classifier that is defined using a separating hyperplane between the classes.
- This hyperplane is the N-dimensional version of a line.
- Given labeled training data and a binary classification problem, the SVM finds the optimal hyperplane that separates the training data into two classes.



Guess the Algorithm: 2

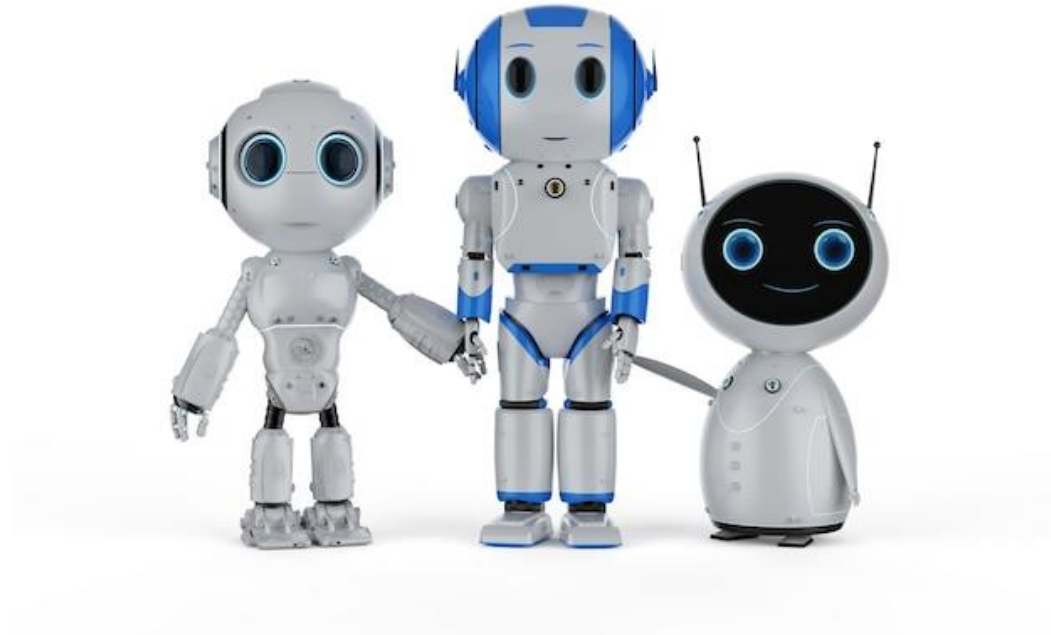
- A consumer electronics company created a model to decide whether a loudspeaker was good enough for a professional musician. Evaluating a musical instrument loudspeaker involves subjective judgement about whether it generates a “good” sound. Only engineers with years of experience can reliably make that decision, and then only after repeated listening to a single loudspeaker and comparing the sounds it produces with those produced by a reference speaker.
- With 60 speakers and more than 50 features, which algorithm worked best?

SVM

https://www.mathworks.com/company/newsletters/articles/applying-machine-learning-techniques-to-classify-musical-instrument-loudspeakers.html?s_tid=mldl_celarticle_but21&

- Selection started by examining the features of their data and used a combination of domain knowledge and preprocessing techniques to reduce the feature size to 27. They tried **logistic regression** and **an ANN**, but found they achieved the best results with a support vector machine—this was likely impacted by the size of the dataset.
- “There is value in applying simpler algorithms such as linear logistic regression, even if they end up performing poorly compared to SVM, neural networks, or ensemble methods. With more advanced machine learning techniques, it was difficult to break down and interpret the results. In this respect, I learned more about ideal performance from the simpler algorithms.”

Group Up!



Work on your assignment!

Guidance

<https://archive.ics.uci.edu/>

<https://www.kaggle.com/datasets>

<https://registry.opendata.aws/>

Find your data!

Reflection

ASSIGNMENT PROCESS

