Lecture 2: Terminology, Baselines, Decision Trees

Firas Moosvi (Slides adapted from Varada Kolhatkar)

Announcements

- Things due this week
  - Homework 1
  - There are some assert statements in the notebook you can use to check your work! If they fail, you didn't get the exercise correct
- Homework 2 (hw2) has been released
  - There is some autograding in this homework.
- You can find the tentative due dates for all deliverables here.
- Please monitor Ed Discussion (especially pinned posts and instructor posts) for announcements.
- I'll assume that you've watched the pre-lecture videos.

Participation marks (10%)

- You will have weekly “Learning Logs” where you will reflect on the material covered for the week
- The first Learning Log will be released soon!

Suggested Workflow for working with Jupyter Notebooks

🎯 Learning Outcomes

By the end of this lesson, you will be able to:

- Define key machine learning terminology:
  features, targets, predictions, training, error, classification vs. regression, supervised vs. unsupervised learning, hyperparameters vs. parameters, baselines, decision boundaries
- Build a simple machine learning model in **scikit-learn**, explaining the `fit`–`predict` workflow and evaluating performance with the `score` method
- Describe at a high level how decision trees are trained (fitting) and how they make predictions
- Implement and visualize decision trees in scikit-learn using `DecisionTreeClassifier` and `DecisionTreeRegressor`

Recap: What is ML?

- ML uses data to build models that find patterns, make predictions, or generate content.
- It helps computers learn from data to make decisions.
- No one model works for every situation.

Class Participation using Agora

- Visit: https://agora.students.cs.ubc.ca
- Login with your UBC CWL
- Use enrol code: canvas

Clicker 2.1: ML or not

Select all of the following statements which are suitable problems for machine learning.

- a. Identifying objects within digital images, such as facial recognition in security systems or categorizing images based on content.
- b. Determining if individuals meet the necessary criteria for government or financial services based on strict guidelines.
- c. Identifying unusual patterns that may indicate fraudulent transactions in banking and finance.
- d. Automatically analyzing images from MRIs, CT scans, or X-rays to detect abnormalities like tumors or fractures.
- e. Addressing mental health issues where human empathy, understanding, and adaptability are key.

Therapists using ChatGPT secretly 😔

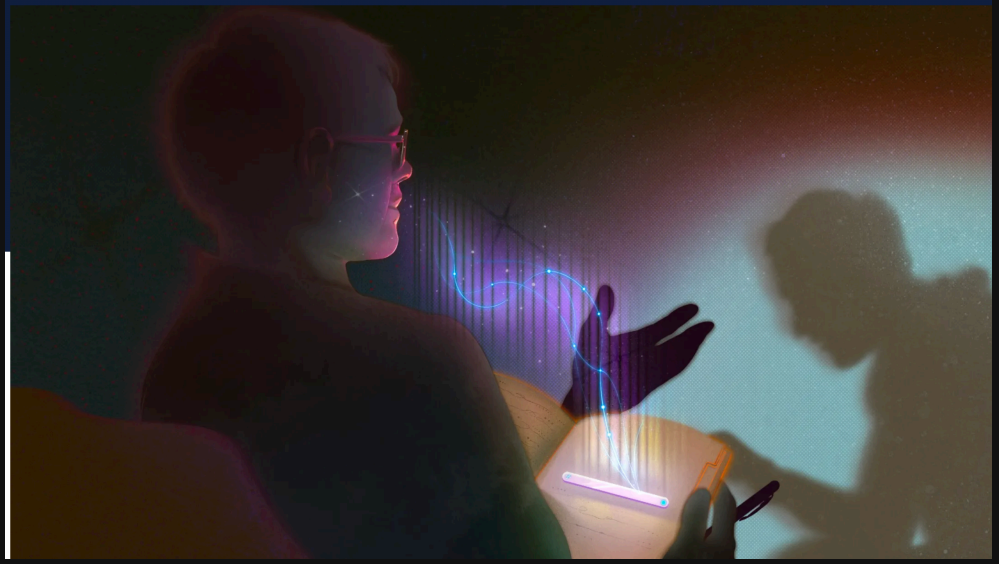
ARTIFICIAL INTELLIGENCE

Therapists are secretly using ChatGPT. Clients are triggered.

Some therapists are using AI during therapy sessions. They're risking their clients' trust and privacy in the process.

By Laurie Clarke

September 2, 2025

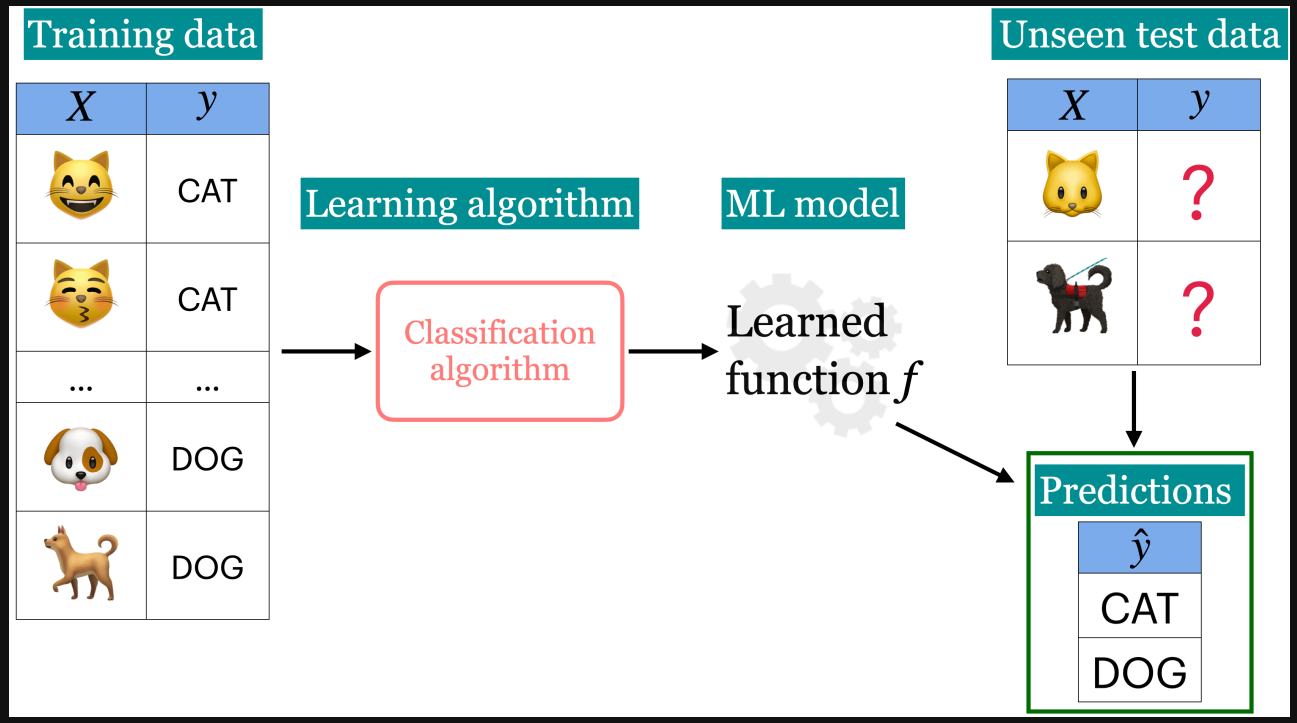


Recap: When is ML suitable?

- ML excels when the problem involve identifying complex patterns or relationships in large datasets that are difficult for humans to discern manually.
- Rule-based systems are suitable where clear and deterministic rules can be defined. Good for structured decision making.
- Human experts are good with problems which require deep contextual understanding, ethical judgment, creative input, or emotional intelligence.

Recap: Supervised learning

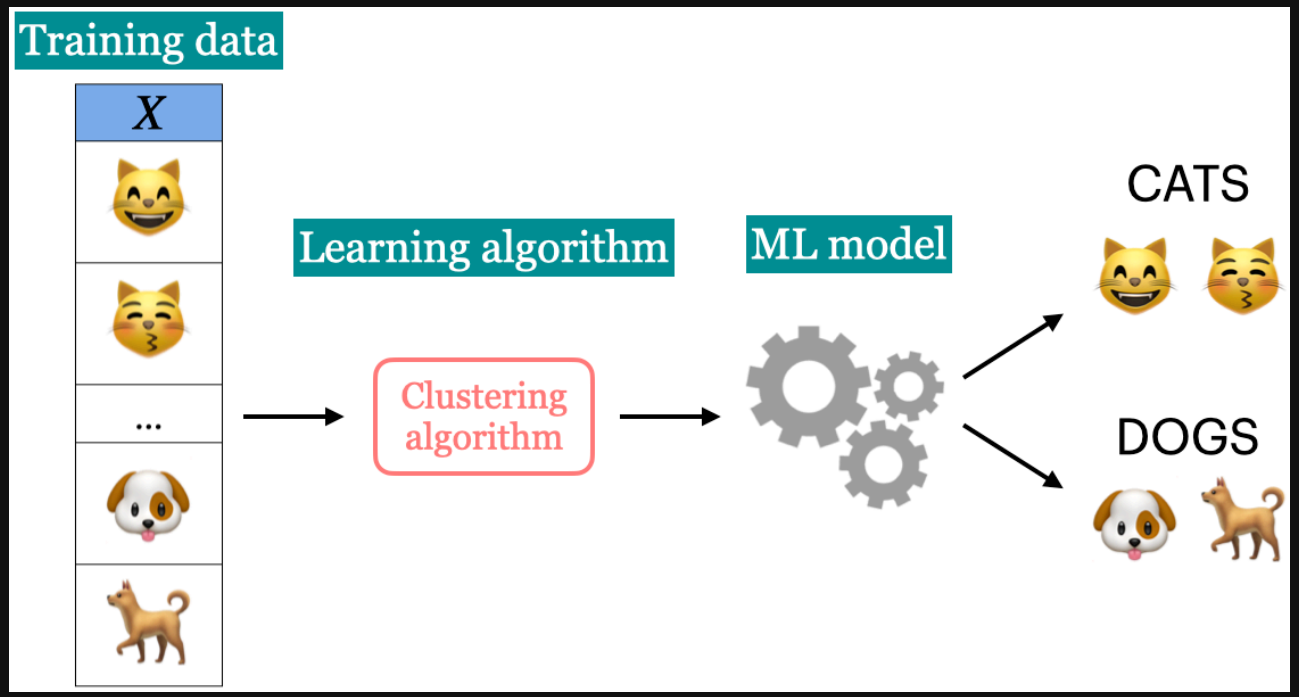
- We wish to find a model function f that relates X to y.
- We use the model function to predict targets of new examples.



In the first part of this course, we'll focus on supervised machine learning.

Unsupervised learning

- Training data consists of observations X without any corresponding targets.
- Unsupervised learning could be used to group similar things together in X or to find underlying structure in the data.



Clicker 2.2: Supervised vs unsupervised

Participate using Agora (code: canvas)

Select all of the following statements which are examples of supervised machine learning

- a. Finding groups of similar properties in a real estate data set.
- b. Predicting whether someone will have a heart attack or not on the basis of demographic, diet, and clinical measurement.
- c. Grouping articles on different topics from different news sources (something like the Google News app).
- d. Detecting credit card fraud based on examples of fraudulent and non-fraudulent transactions.
- e. Given some measure of employee performance, identify the key factors which are likely to influence their performance.

Clicker 2.3: Classification vs. Regression

Participate using Agora (code: canvas)

Select all of the following statements which are examples of regression problems

- a. Predicting the price of a house based on features such as number of bedrooms and the year built.
- b. Predicting if a house will sell or not based on features like the price of the house, number of rooms, etc.
- c. Predicting percentage grade in CPSC 330 based on past grades.
- d. Predicting whether you should bicycle tomorrow or not based on the weather forecast.
- e. Predicting appropriate thermostat temperature based on the wind speed and the number of people in a room.

Today's focus

- ML Terminology
- Using sklearn to build a simple supervised ML model
- Intuition of Decision Trees

Framework

- There are many frameworks to do do machine learning.
- We'll mainly be using `scikit-learn` framework.

Running example

Imagine you're in the fortunate situation where, after graduating, you have a few job offers and need to decide which one to choose. You want to pick the job that will likely make you the happiest. To help with your decision, you collect data from like-minded people.

- Can you think of relevant features for this problem?

Toy job happinees dataset

Here are the first few rows of a toy dataset.

```
1 toy_happiness_df = pd.read_csv(DATA_DIR + 'toy_job_happiness.csv')
2 toy_happiness_df
```

	supportive_colleagues	salary	free_coffee	boss_vegan	happy?
0	0	70000	0	1	Unhappy
1	1	60000	0	0	Unhappy
2	1	80000	1	0	Happy
3	1	110000	0	1	Happy
4	1	120000	1	0	Happy
5	1	150000	1	1	Happy
6	0	150000	1	0	Unhappy

Terminology

Features, target, example

Terminology | Data

- What are the **features** X?
  - features = inputs = predictors = explanatory variables = regressors = independent variables = covariates
- What's the target y?
  - target = output = outcome = response variable = dependent variable = labels
- What is an example?

Classification vs. Regression

- Is this a **classification** problem or a **regression** problem?

	supportive_colleagues	salary	free_coffee	boss_vegan	happy?
0	0	70000	0	1	Unhappy
1	1	60000	0	0	Unhappy
2	1	80000	1	0	Happy
3	1	110000	0	1	Happy
4	1	120000	1	0	Happy
5	1	150000	1	1	Happy
6	0	150000	1	0	Unhappy

(Optional) Inference vs. Prediction

- **Inference** asks: *Why does something happen?*
- Goal: **understand** and **quantify** the relationship between variables
- Often involves estimating model parameters and testing hypotheses
- Example: *Which factors influence happiness, and by how much?*

- **Prediction** asks: *What will happen?*
- Goal: **accurately predict the target** without needing to fully explain the relationships
- Example: *Will you be happy in a particular job?*

Of course these goals are related, and in many situations we need both.

Training

- In supervised ML, the goal is to learn a function that maps input features (X) to a target (y).
- The relationship between X and y is often complex, making it difficult to define mathematically.
- We use algorithms to approximate this complex relationship between X and y.
- **Training** is the process of applying an algorithm to learn the best function (or model) that maps X to y.
- In this course, I'll help you develop an intuition for how these models work and demonstrate how to use them in a machine learning pipeline.

Error and accuracy

- Machine learning models are **not perfect**—they will make mistakes.
- To judge whether a model is **useful**, we need to track its performance.
- For classification problems, the most common (and default in `sklearn`) metric is **accuracy**:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of examples}}$$

Separating X and y

- In order to train a model we need to separate X and y from the dataframe.

```
X = toy_happiness_df.drop(columns=["happy?"]) # Extract the feature set by removing the target column "happy
y = toy_happiness_df["happy?"] # Extract the target variable "happy?"
```

Baseline

- Let's try a simplest algorithm of predicting the most popular target!

```
1 from sklearn.dummy import DummyClassifier
2 model = DummyClassifier(strategy="most_frequent") # Initialize the DummyClassifier to always predict the mos
3 model.fit(X, y) # Train the model on the feature set X and target variable y
4 toy_happiness_df['dummy_predictions'] = model.predict(X) # Add the predicted values as a new column in the d
5 toy_happiness_df
```

	supportive_colleagues	salary	free_coffee	boss_vegan	happy?	d
0	0	70000	0	1	Unhappy	H
1	1	60000	0	0	Unhappy	H
2	1	80000	1	0	Happy	H
3	1	110000	0	1	Happy	H
4	1	120000	1	0	Happy	H
5	1	150000	1	1	Happy	H
6	0	150000	1	0	Unhappy	H

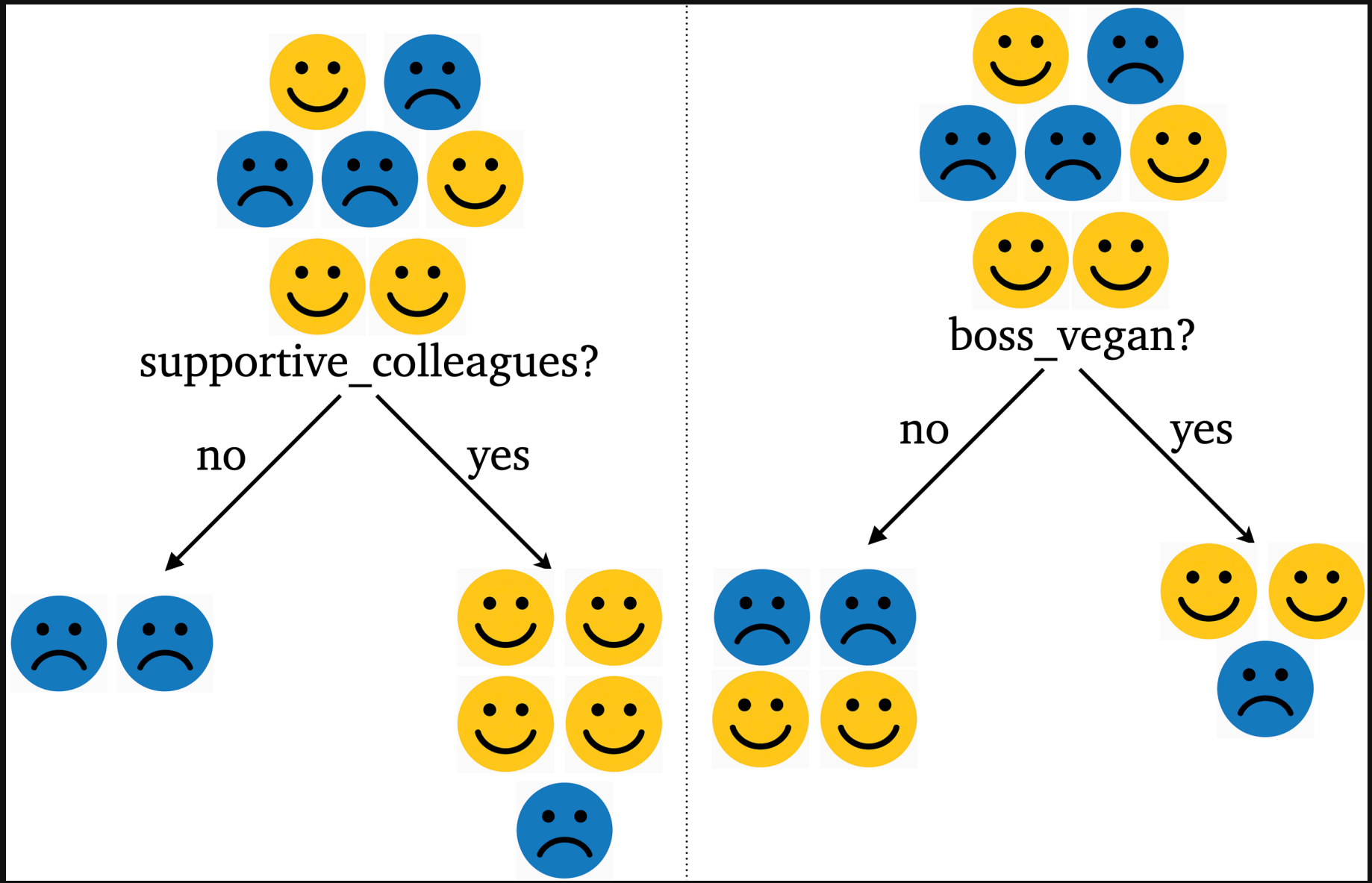
Decision trees intuition

- One intuitive way to build a model is by asking a series of **yes/no questions**, forming a tree.
- Which question would help you best separate the **happy** and **unhappy** examples?

	supportive_colleagues	salary	free_coffee	boss_vegan	happy?
0	0	70000	0	1	Unhappy
1	1	60000	0	0	Unhappy
2	1	80000	1	0	Happy
3	1	110000	0	1	Happy

Which question is more effective?

Visual | Data


supportive_colleagues?
no
yes
boss_vegan?
no
yes

What are we trying to learn?

- We want to learn which questions to ask and in what order.
- How many possible questions could we ask with these features?

	supportive_colleagues	salary	free_coffee	boss_vegan	happy?
0	0	70000	0	1	Unhappy
1	1	60000	0	0	Unhappy
2	1	80000	1	0	Happy
3	1	110000	0	1	Happy
4	1	120000	1	0	Happy
5	1	150000	1	1	Happy
6	0	150000	1	0	Unhappy

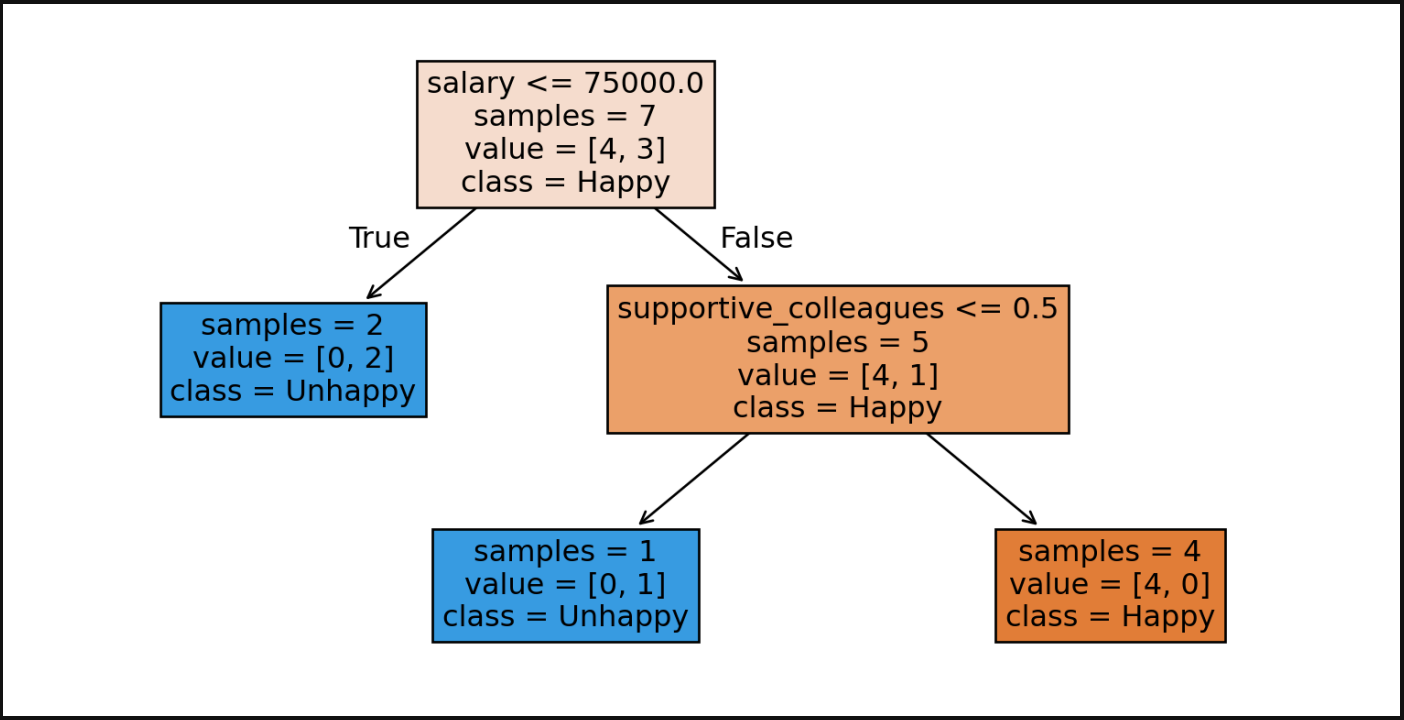
Decision tree Training (high level)

- Training a decision tree is a search process: we look for the “best” tree among many possible ones.
- There are different algorithms for learning trees. Check this out.
- At each step, we evaluate candidate questions using measures such as:
  - Information gain
  - Gini index
- The goal is to split the data into groups with greater certainty (more homogeneous outcomes).

Decision tree with sklearn

Let's train a simple decision tree on our toy dataset using sklearn

```
1 from sklearn.tree import DecisionTreeClassifier # import the classifier
2 from sklearn.tree import plot_tree
3
4 model = DecisionTreeClassifier(max_depth=2, random_state=1) # Create a class object
5 model.fit(X, y)
6 plot_tree(model, filled=True, feature_names = X.columns, class_names=["Happy", "Unhappy"], impurity = False,
```



Prediction

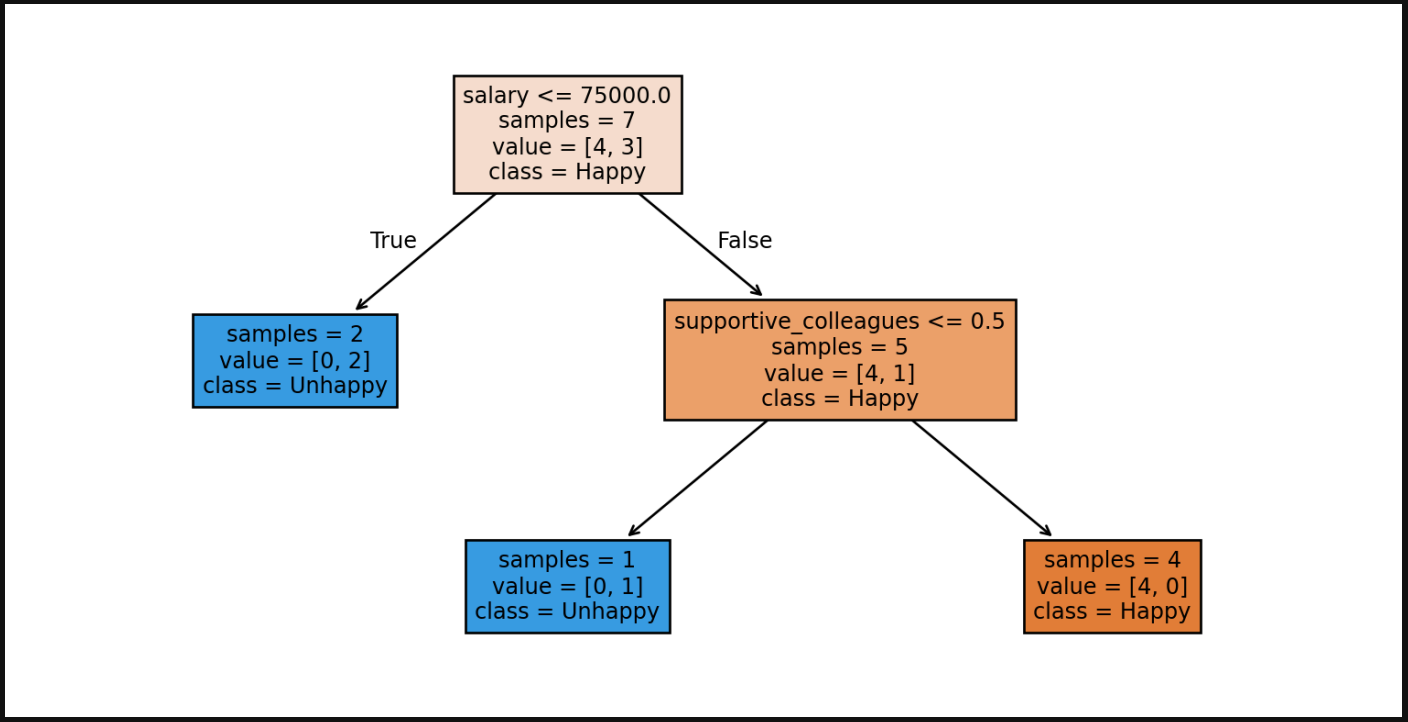
- Given a new example, how does a decision tree predict the class of this example?
- What would be the prediction for the example below using the tree above?
  - supportive_colleagues = 1, salary = 60000, coffee_machine = 0, vegan_boss = 1,

Prediction with sklearn

- What would be the prediction for the example below using the tree above?
  - supportive_colleagues = 1, salary = 60000, free_coffee = 0, vegan_boss = 1,

```
1 test_example = [[1, 60000, 0, 1]]
2 print("Model prediction: ", model.predict(test_example))
3 plot_tree(model, filled=True, feature_names = X.columns, class_names = ["Happy", "Unhappy"], impurity = Fals
```

```
Model prediction:  ['Unhappy']
```



Parameters vs. Hyperparameters

- Parameters
  - The questions (features and thresholds) used to split the data at each node.
  - Example: salary <= 75000 at the root node
- Hyperparameters
  - Settings that control tree growth, like `max_depth`, which limits how deep the tree can go.

Decision boundary

- A decision boundary is the **line, curve, or surface** that separates classes.
- Points on one side ⟶ Model predicts Class Happy
- Points on the other side ⟶ Model predicts Class Unhappy

Decision boundary with `max_depth=1`

Decision boundary with `max_depth=2`

Clicker 2.4: Baselines and Decision trees

iParticipate using Agora (code: canvas)

Select all of the following statements which are TRUE.

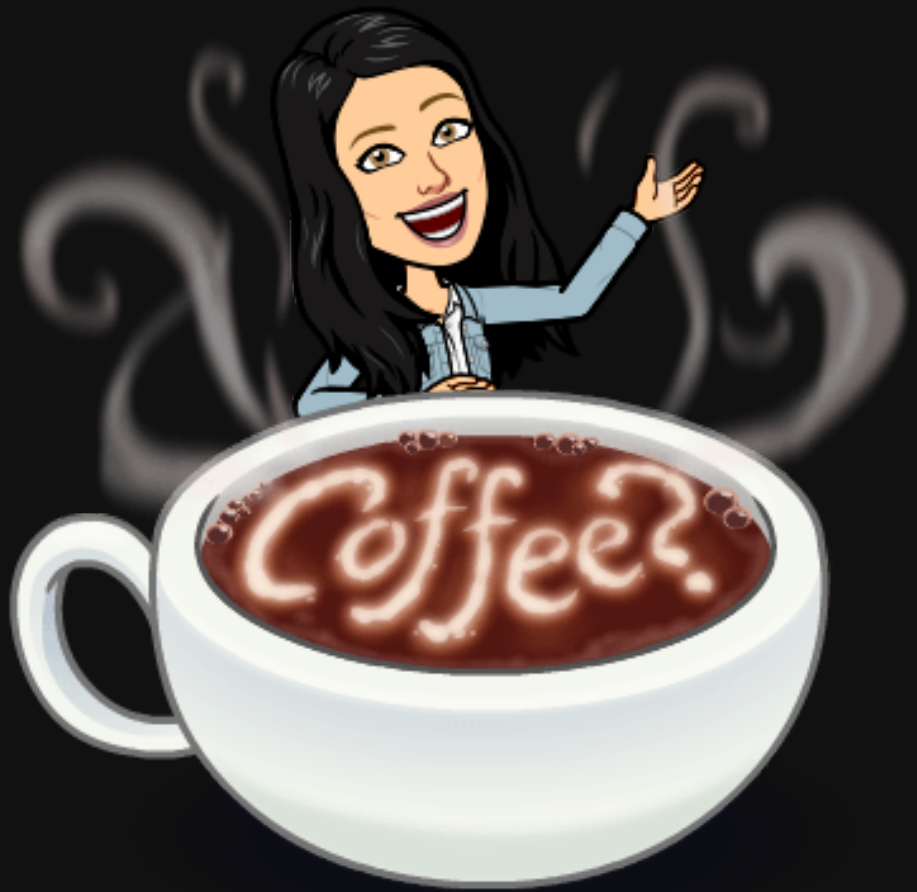
- a. Change in features (i.e., binarizing features above) would change DummyClassifier predictions.
- b. predict takes only X as argument whereas fit and score take both X and y as arguments.
- c. For the decision tree algorithm to work, the feature values must be binary.
- d. The prediction in a decision tree works by routing the example from the root to the leaf.

Summary

- Terminology
- `sklearn` basic steps
- Decision tree intuition

Break

Let’s take a break!



Group Work: Class Demo & Live Coding

In some of the classes, we will do a bit of live coding to get your used to practical machine learning. You are **highly encouraged** to follow along - we won't usually finish *everything* in the demo, but it should be a significant portion that you can finish off after class.

For this demo, each student should click this link to create a new repo in their accounts, then clone that repo locally to follow along with the demo from today.