Lecture 2: Terminology, Baselines, Decision Trees

Varada Kolhatkar

Announcements

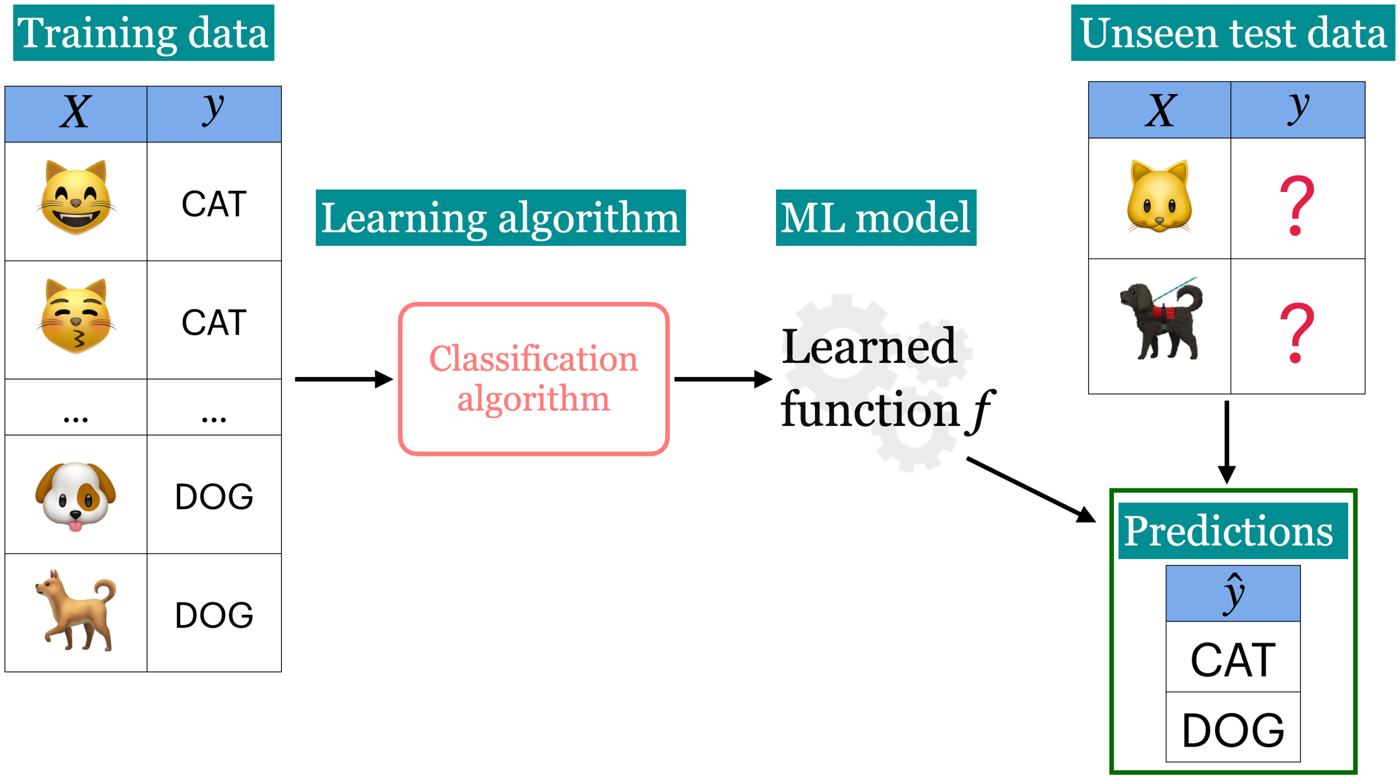
- Things due this week
  - Homework 1 (hw1): Due Sept 10 11:59pm
- Homework 2 (hw2) has been released (Due: Sept 16, 11:59pm)
  - There is some autograding in this homework.
- You can find the tentative due dates for all deliverables here.
- Please monitor Piazza (especially pinned posts and instructor posts) for announcements.
- I’ll assume that you’ve watched the pre-lecture videos.

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.

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.



Training data

X	y
😸	CAT
😽	CAT
...	...
🐶	DOG
🐕	DOG

Unseen test data

X	y
🐱	?
🐕‍🦺	?

Predictions

$\hat{y}$
CAT
DOG

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

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. Here are the first few rows of this toy dataset.

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

	supportive_colleagues	salary	free_coffee	boss_veg
0	0	70000	0	1
1	1	60000	0	0
2	1	80000	1	0

	supportive_colleagues	salary	free_coffee	boss_veg
3	1	110000	0	1
4	1	120000	1	0
5	1	150000	1	1
6	0	150000	1	0

Terminology

Features, target, example

- 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
- Can you think of other relevant features for this problem?
- What is an example?

Classification vs. Regression

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

	supportive_colleagues	salary	free_coffee	boss_veg
0	0	70000	0	1
1	1	60000	0	0
2	1	80000	1	0
3	1	110000	0	1
4	1	120000	1	0
5	1	150000	1	1
6	0	150000	1	0

Prediction vs. Inference

- **Inference** is using the model to understand the relationship between the features and the target
  - Why certain factors influence happiness?
- **Prediction** is using the model to predict the target value for new examples based on learned patterns.
- 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.

Separating *X* and *y*

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

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

Baseline

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

```python
from sklearn.dummy import DummyClassifier
model = DummyClassifier(strategy="most_frequent") # Initialize the DummyCla
model.fit(X, y) # Train the model on the feature set X and target variable
toy_happiness_df['dummy_predictions'] = model.predict(X) # Add the predicte
toy_happiness_df
```

	supportive_colleagues	salary	free_coffee	boss_veg
0	0	70000	0	1
1	1	60000	0	0
2	1	80000	1	0
3	1	110000	0	1

	supportive_colleagues	salary	free_coffee	boss_veg
4	1	120000	1	0
5	1	150000	1	1
6	0	150000	1	0

Decision trees

Intuition

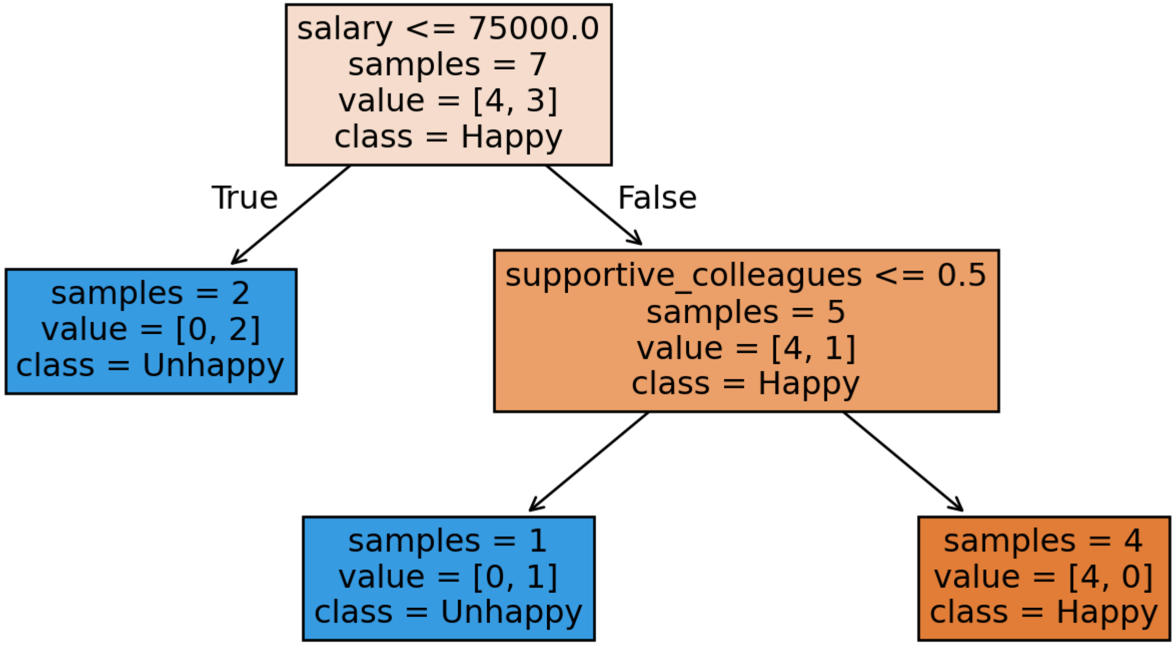
- Decision trees find the “best” way to split data to make predictions.
- Each split is based on a question, like ‘Are the colleagues supportive?’
- The goal is to group data by similar outcomes at each step.
- Now, let’s see a decision tree using sklearn.

Decision tree with sklearn

Let’s train a simple decision tree on our toy dataset.

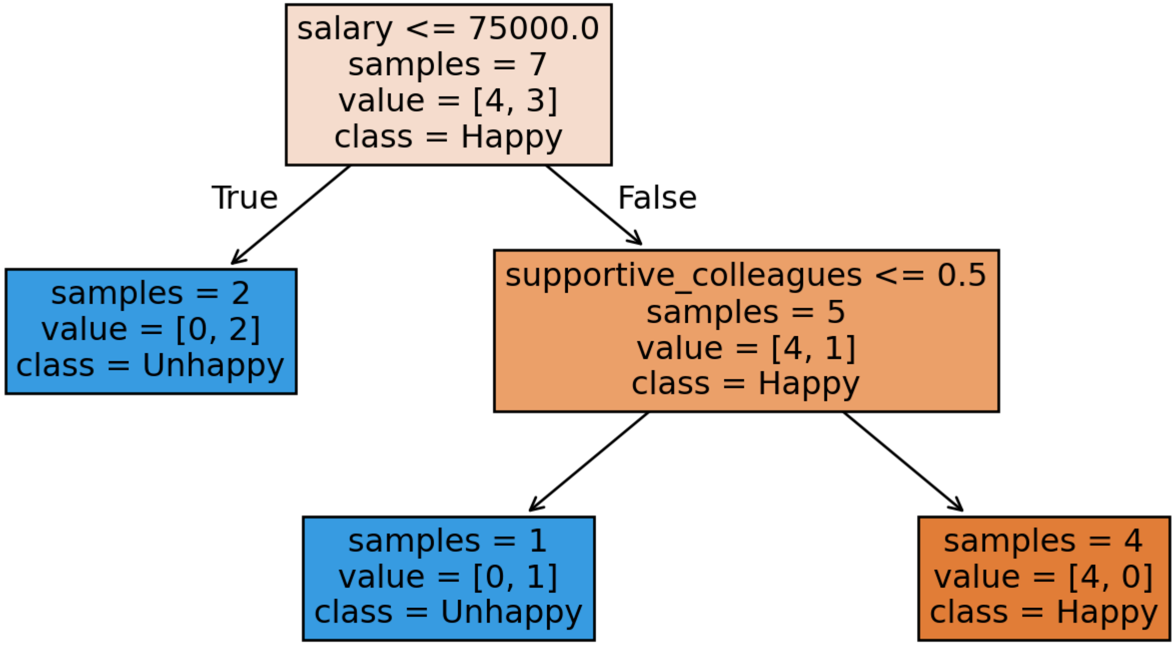
```python
from sklearn.tree import DecisionTreeClassifier # import the classifier
from sklearn.tree import plot_tree

model = DecisionTreeClassifier(max_depth=2, random_state=1) # Create a clas
model.fit(X, y)
plot_tree(model, filled=True, feature_names = X.columns, class_names=["Happ
```



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,

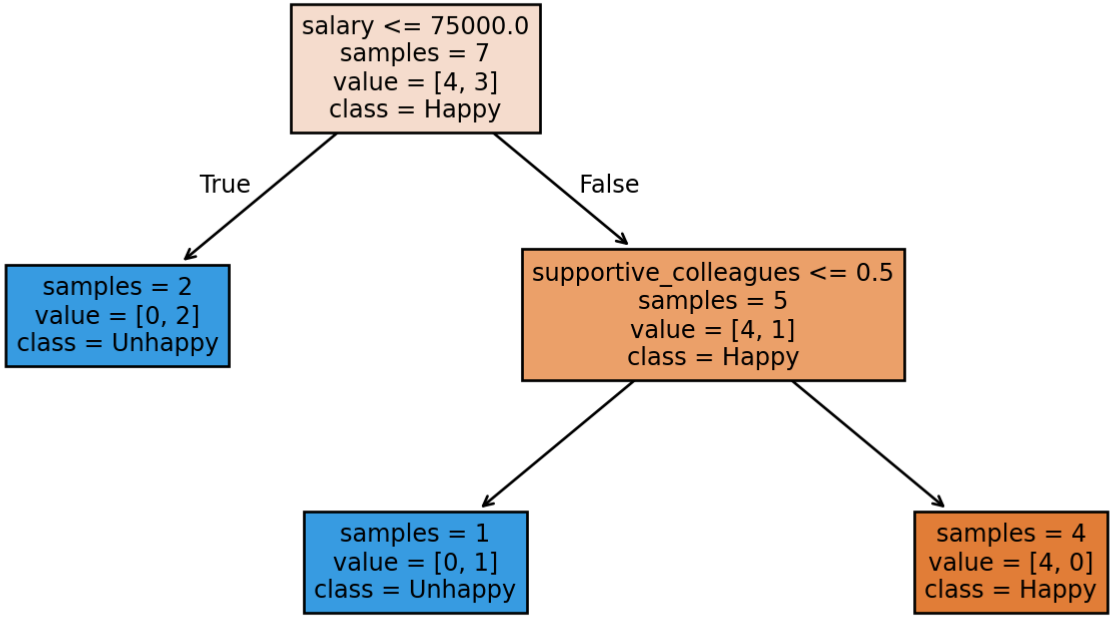

salary <= 75000.0
samples = 7
value = [4, 3]
class = Happy
True
False
samples = 2
value = [0, 2]
class = Unhappy
supportive_colleagues <= 0.5
samples = 5
value = [4, 1]
class = Happy
samples = 1
value = [0, 1]
class = Unhappy
samples = 4
value = [4, 0]
class = Happy

Prediction with `sklearn`

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

```python
test_example = [[1, 60000, 0, 1]]
print("Model prediction: ", model.predict(test_example))
plot_tree(model, filled=True, feature_names = X.columns, class_names = ["Ha
```

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


salary <= 75000.0
samples = 7
value = [4, 3]
class = Happy
True
False
samples = 2
value = [0, 2]
class = Unhappy
supportive_colleagues <= 0.5
samples = 5
value = [4, 1]
class = Happy
samples = 1
value = [0, 1]
class = Unhappy
samples = 4
value = [4, 0]
class = Happy

Training (high level)

- How many possible questions could we ask in this context?

	supportive_colleagues	salary	free_coffee	boss_veg
0	0	70000	0	1
1	1	60000	0	0
2	1	80000	1	0
3	1	110000	0	1
4	1	120000	1	0
5	1	150000	1	1
6	0	150000	1	0

Training (high level)

- Decision tree learning is a search process to find the “best” tree among many possible ones.
- We evaluate questions using measures like **information gain** or the **Gini index** to find the most effective split.
- At each step, we aim to split the data into groups with more certainty in their outcomes.

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 with `max_depth=1`

Decision boundary with `max_depth=2`

iClicker 2.2: Supervised vs unsupervised

Clicker cloud join link: https://join.iclicker.com/VYFJ

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.

iClicker 2.3: Classification vs. Regression

Clicker cloud join link: https://join.iclicker.com/VYFJ

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.

iClicker 2.5: Baselines and Decision trees

iClicker cloud join link: https://join.iclicker.com/VYFJ

Select all of the following statements which are TRUE.

HW2 Worksheet portion