# Bayu’s Data Science Notes 1: Machine learning basics

Bayu Wilson

# Outline

1. Data science vs machine learning (ML)
2. Main 2 types of ML
3. Main 2 types of ML tasks
4. Common ML algorithms
    a. Tree-based algorithms

# Background: machine learning & data science

In your own words, define data science and machine learning. How do they relate to each other?

# Background: machine learning & data science

**Machine learning (ML)**: is a field of study in AI concerned with the development and **study of statistical algorithms that can learn patterns from data and generalize these patterns to unseen data**, and thus perform tasks without explicit instructions

**Data science**: is a field of study that uses a variety of tools and methods to **extract knowledge and insights from data** (both structured and unstructured)

Similarities & differences:

- Data science is broader scope. Also includes data understanding, cleaning, and visualization. A data scientist answers business questions and may (or may not) use ML as a tool
- On the other hand, a ML engineer may focus on developing & optimizing algorithms

# Which of these would likely be a data science problem that doesn’t require machine learning? Explain. (select all)

Recall that in ML models learn patterns from data and generalize these patterns to unseen data

A. Netflix recommendations based off of user watch history, location, time of year, etc.
B. Estimating cosmological parameters from the CMB power spectrum
C. Super-resolving low resolution (don’t resolve small galaxies) simulations to predict small-scale galaxy formation
D. Running A/B testing on a website to see if changing size of “BUY NOW” button increases clicks

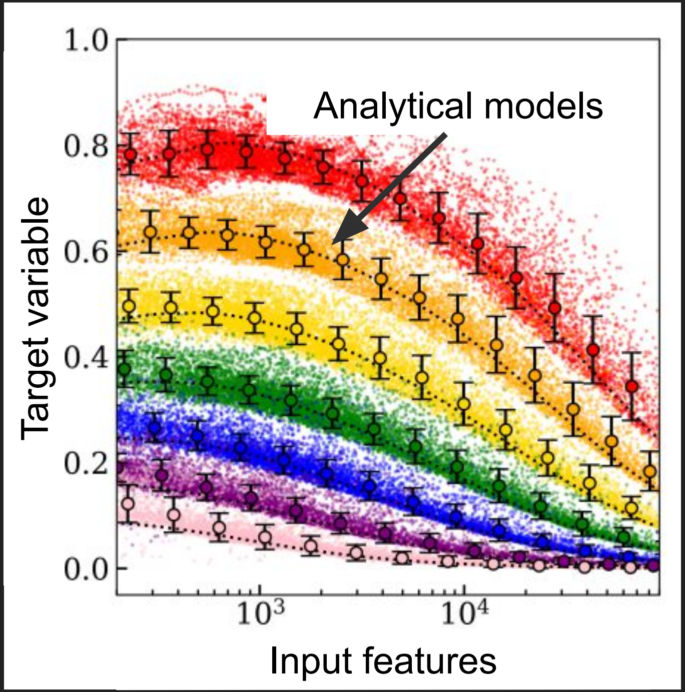
# Another example of data science without machine learning

Usually the most important (and hardest) part of a data science project is acquiring good data and truly understanding the problem you're trying to solve

In my data science astrophysics project at UCR, I first made an analytical model with physics-informed input features

No machine learning model is necessary because we **analytically predicted the mapping between the features and target**!

This is still a data science project because I extracted insights (confirmation that my model is good!) from simulation data



# Background: 2 main, basic types of machine learning

Define supervised and unsupervised learning. Give an example of each.

# Background: 2 main, basic types of machine learning

Supervised learning: The algorithm **learns from labeled data**. This means for every input, there is a corresponding output label. **The model learns the mapping from input to output.**

Unsupervised learning: The algorithm learns from unlabeled data. **The model finds patterns/groups in data without explicit guidance (no output label)**.

# Which of these are likely supervised or unsupervised problems?

A. Segmenting customers into groups with similar traits so you can show them ads best suited for them to buy stuff

B. You would like to predict cost of a car. A lot of features are redundant or inconsequential so you try to reduce the dimensionality of the data

C. Fraud detection for credit card purchases

D. Risk assessment to determine whether customers are likely to default on loans

# Background: Supervised ML tasks

Define regression and classification tasks. Give an example of each of them.

# Background: Supervised ML tasks

Regression: A task where the model learns to establish a **relationship between the input variables and a continuous numerical target variable**

Classification: A task to **predict categorical outcomes or labels based on input features**.

# Which of these are regression or classification tasks

a. Predicting if a person has a disease
b. Temperature forecasting on a farm in Kansas
c. Estimating patient recover times after surgery
d. Determining galaxy morphology using Hubble data

# Commonly used ML algorithms

## Supervised

- Classification
  - Logistic regression
  - Support vector machines (SVM)
  - k-Nearest Neighbors
  - Naive Bayes
  - Tree-based
  - Neural Networks
- Regression
  - Linear regression
  - Support Vector Regression (SVR)
  - Tree-based Regression
  - Neural Network Regression

## Unsupervised

- Clustering
  - k-Means
  - Hierarchical Clustering
  - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
  - Gaussian Mixture Models (GMM)
- Dimensionality Reduction
  - Principal Component Analysis (PCA)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)

# Commonly used ML algorithms

## Supervised

- Classification
    - Logistic regression
    - Support vector machines (SVM)
    - k-Nearest Neighbors
    - Naive Bayes
    - **Tree-based**
    - Neural Networks
- Regression
    - Linear regression
    - Support Vector Regression (SVR)
    - Tree-based Regression
    - Neural Network Regression

## Unsupervised

- Clustering
    - k-Means
    - Hierarchical Clustering
    - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
    - Gaussian Mixture Models (GMM)
- Dimensionality Reduction
    - Principal Component Analysis (PCA)
    - t-Distributed Stochastic Neighbor Embedding (t-SNE)

# Decision trees, random forests, and XGBoost: what are they and when are they used?

# Commonly used ML algorithms: Decision trees

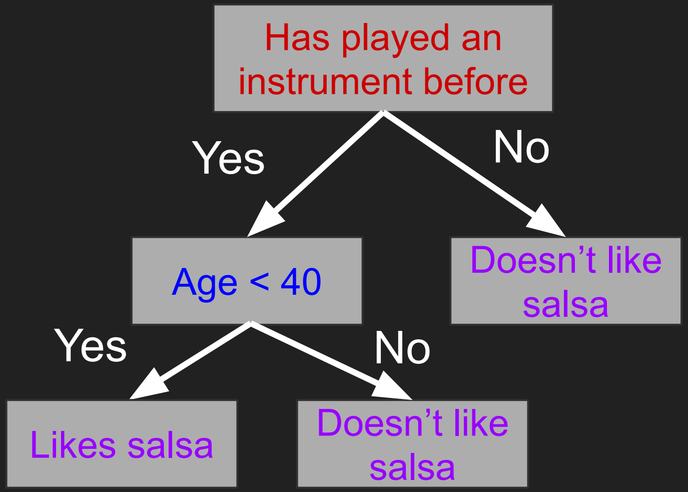
A tree learns sequence of if then questions to generate predictions. Decision trees split on the feature and corresponding split point that results in the largest information gain (i.e. minimizing impurities)

For example, classification of someone who likes salsa dancing.

- Input features: Has played instrument before and age
- Output: Binary classification of whether or not someone likes salsa

How could this work?

- Quantify impurity (how good of a split it is) for each feature. Smallest impurity will be the root. Each branch is basically chosen as the next smallest impurity



Root Nodes
Internal Nodes or Branches
Leaf Nodes

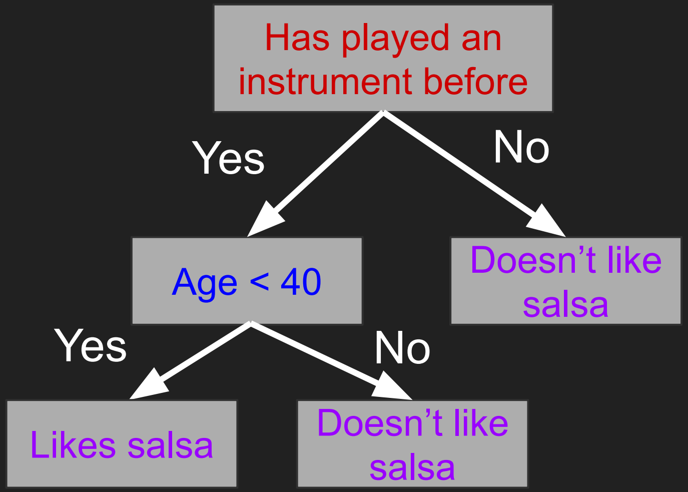
# Commonly used ML algorithms: Decision trees

Pros:

- Interpretable
- Fast
- Good for nonlinear problems

Cons:

- Prone to overfitting
- Unstable (High Variance)



Root Nodes
Internal Nodes or Branches
Leaf Nodes

# Commonly used ML algorithms: Random forests

Decision tree weaknesses, such as overfitting and high variance, can be addressed with **random forests, an ensemble method that combines multiple decision trees**. By training each tree on a *random* subset of data and features, random forests reduce overfitting and produce more stable, accurate predictions.

Uses Bagging to make decisions (bootstrapping the data and aggregating results)

Limitations

- Slow performance and large memory requirements for large datasets
- may still have high bias (underfitting) since each tree is built independently

# Commonly used ML algorithms: XGBoost

XGBoost is a more sophisticated ensemble technique based on gradient boosting, which builds decision trees sequentially, correcting errors from previous trees

*Boosting* (sequential trees) tends to lower bias while keeping variance in check, leading to models that perform better on complex data

*Gradient* boosting allows you to minimize a loss function for the tree accuracy and then choose the next model parameters using gradient descent

*eXtreme* because it is faster, more efficient, and more accurate than a most gradient boosting methods. The main advantages of XGBoost over traditional gradient boosting are its ability to handle regularization, parallel processing, automatic handling of missing values, more efficient tree-building methods, and additional features like early stopping and out-of-core processing.

# Which of these algorithms, decision tree, random forests, or XGBoost, would work best for the following scenarios?

A. Bayu wants to predict whether students will come to his salsa class given the week of the quarter, weather, and the previous week's attendance. The dataset is small, with ~100 data points.

B. A healthcare company is trying to predict whether a patient will be diagnosed with a disease based on a medium sized dataset containing medical history and symptoms. Stakeholders are looking for a quick solution that is easy to interpret but they also want to handle noisy data.

C. A financial institution is building a model to predict credit card fraud. The dataset is large, with millions of transactions, and the patterns of fraud are highly complex and nonlinear. Model performance is critical, and training time is not a major concern.

# Tutorials

https://github.com/bayu-wilson/teaching_DS/blob/main/DecisionTreeTutorial.ipynb