Week 1: Introduction to Machine Learning

Welcome to Week 1: Introduction to Machine Learning! In this module, we'll take our first steps in machine learning, exploring its foundational principles and the algorithms that drive the field of data science. By the end of the week, you'll understand the various types of problems that machine learning can address, the most important techniques appropriate to each, and the ways that results may be evaluated. We'll also introduce the idea of fairness and how biased data can compromise your data science projects.

Learning Objectives

At the end of this week, you should be able to:

- Describe the basic types of machine learning algorithms and the problems they address
- Explain the importance of creating ML models that generalize to unseen data
- Compare prediction and inference in obtaining results
- Explain the issues of accuracy and interpretability in evaluating the results obtained
- Explain the difference between reducible and irreducible errors and their sources
- Summarize the bias-variance tradeoff in evaluating models
- Identify how biased data can compromise your results and lead to unfair outcomes

Key Terms

- Machine Learning (ML): A field of AI where algorithms learn from data to make predictions or decisions without explicit programming
- Training Set: A dataset used to train a machine learning model by adjusting its parameters to minimize error
- Testing Set: A dataset used to evaluate the performance of a trained model and estimate its generalization to new, unseen data
- Supervised Learning: A type of ML where models learn from labeled data to predict outcomes
- Unsupervised Learning: A type of ML where models find patterns or structures in unlabeled data
- Regression: A supervised ML technique used to predict real-valued outcomes
- Classification: A supervised ML technique for categorizing data into distinct classes
- Clustering: An unsupervised ML technique that groups similar data points based on their features
- Dimensionality Reduction: A ML technique that combines the information from multiple input variables into fewer dimensions, retaining the most important patterns

and structure in the data
- Prediction: Using a model to calculate outcomes based on new data not included in the training set
- Inference: Drawing conclusions about relationships between variables in the data
- Generalization: The ability of a machine learning model to perform well on new, unseen data by effectively capturing underlying patterns during training rather than memorizing the specific details of the training set
- Reducible Error: The part of a model's error that can be minimized by improving the model
- Irreducible Error: The portion of the error that cannot be eliminated, caused by inherent noise or randomness in the data regardless of the model used
- Bias (of a model): The error introduced by using a model that is too simple to capture the complexity of the underlying data patterns
- Variance: The error introduced because the model is too complex and sensitive to fluctuations in the training data, capturing noise along with the underlying pattern
- Underfitting: A situation where the model is not complex enough, resulting in high bias
- Overfitting: A situation where the model is too complex, resulting in high variance
- Accuracy: Used formally to refer to the percentage of correct predictions for classification models; used informally to refer to how well a model performs
- Interpretability: The ability for a human to understand how a machine learning model makes decisions
- Bias (of a dataset): Errors or distortions in the data that result in unfair or unequal outcomes, often reflecting societal inequalities or imbalanced data collection

Fairness: Ensuring models do not produce biased or discriminatory outcomes

Introduction to Machine Learning

Machine learning (ML) is a discipline that enables computers to learn from data without explicit programming. It allows systems to incrementally improve their performance by finding patterns in data rather than following hardcoded rules.

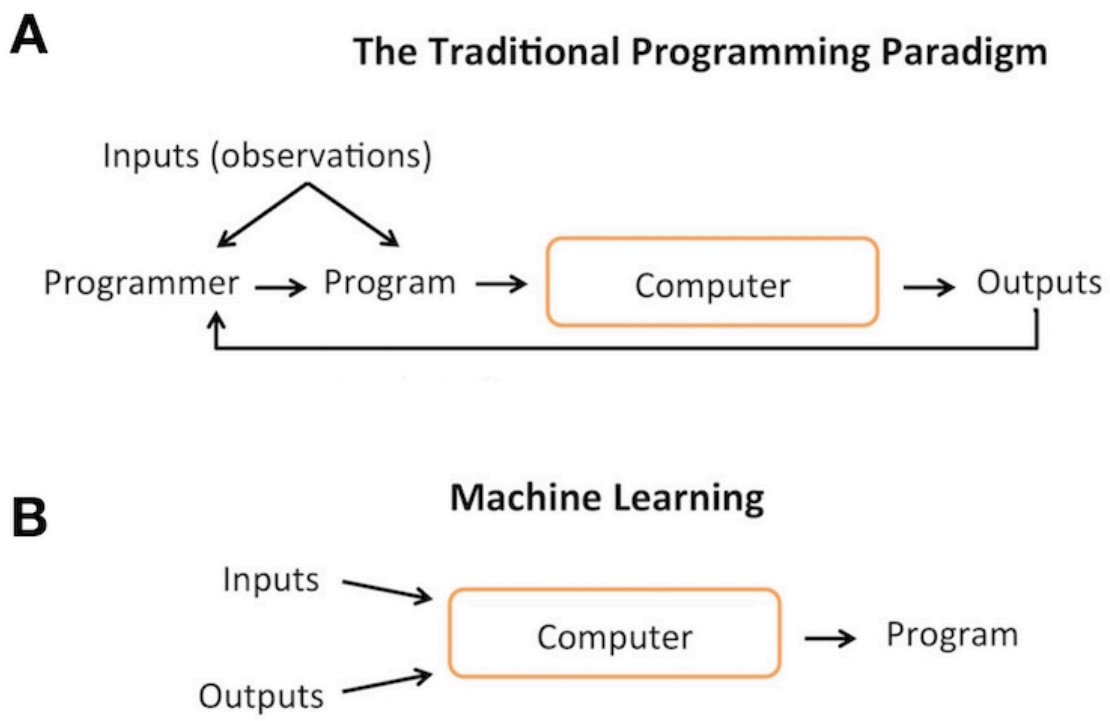


Figure 1: Machine Learning vs Traditional Programming

Notice how we've flipped the script in this diagram: the computer still takes inputs, but instead of loading a program and producing outputs, we load the expected outputs and produce a program (called a **model** in ML).

Before we drill down into the details, let's establish some notation following our textbook Introduction to Statistical Learning (with applications in Python):

he machine learning is to find a model $\hat{f}$ with hat on top approximating an idealized perfect function f which takes inputs (variously called predictors, independent variables, or features) X and produces an output (a response or dependent variable) plus some randomness e that we can not model: $Y \quad = \quad f(X) + e$.

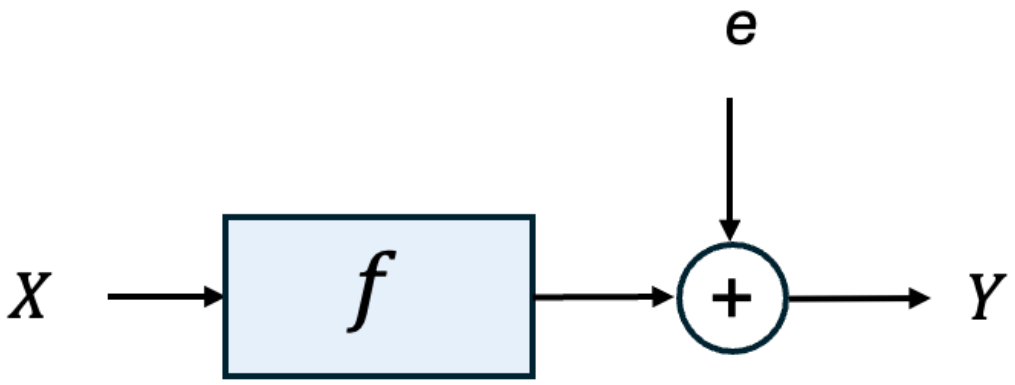


Figure 2: Basic Form of the ML Problem

The key distinction in this framework depends on whether we have a response variable Y in advance. Y plays the role of a **supervisor** of our learning task, leading to two major categories of ML techniques.

Supervised Learning

- The dataset consists of inputs and corresponding outputs:

$$(X_1, Y_1), (X_2, Y_2), \ldots, (X_p, Y_p)$$

Where each X_k is a column vector and Y_k a scalar or tuple, then there are two possible goals:

First we may wish to develp a model $\widehat{f}$ which predicts yeah Y_k with the smallest possible e. In this case, we call the dataset a **training set**. In particular, we wish to find a model that **generalizes** to new data (a **testing set**) from the same domain.

Unsupervised Learning

- When the dataset has not matching outputs,

$$(X_1, X_2, \ldots, X_p)$$

Where each X_k is again a column of vectors. Our goal is to develop a model $\hat{f}$ which predicts a meaningful Y_k with the smallest possible e, where Y_k represents underlying patterns in the data.

In the next two lessons, we shall explore these two types of ML in more detail.

1.2 Lesson: Supervised Learning

Prediction

Although our dataset has both X and Y, we may have new data points X' and without corresponding responses Y'. Therefore, our goal is to use the estimated function f(X) to predict Y' for new observations. This is why prediction is a key goal of supervised learning—it allows us to estimate future or unknown outcomes based on the information in the dataset.

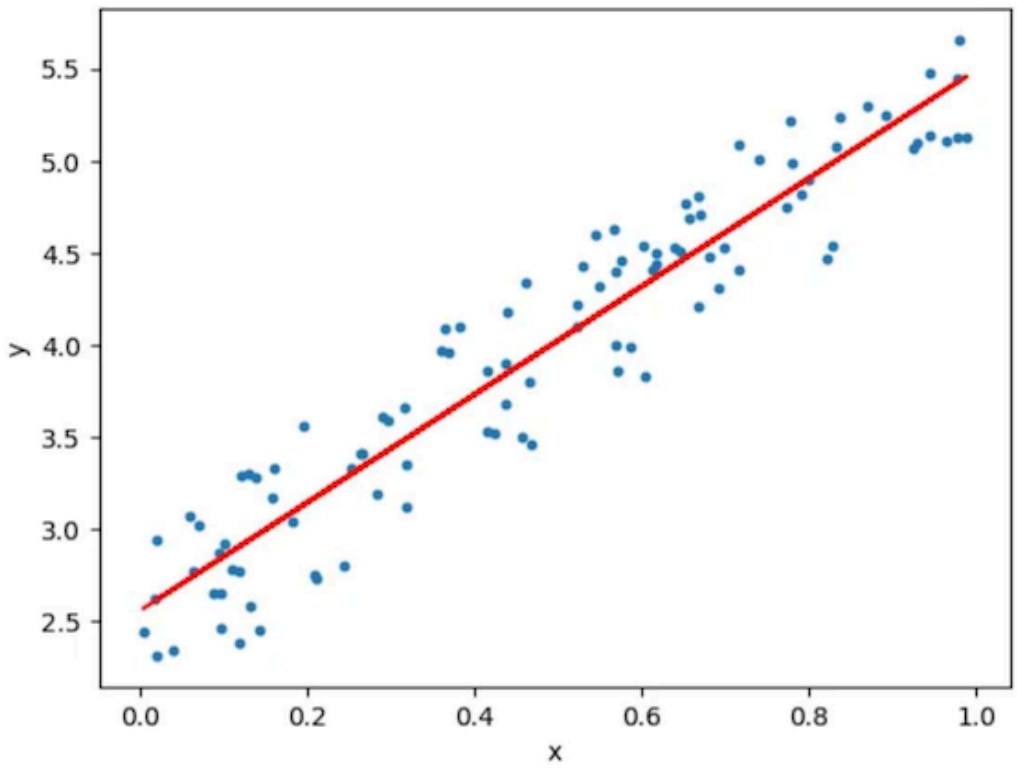
Inference

In other cases, we might be interested in understanding the underlying relationship between the predictor X and the response Y. For example, we may want to determine which predictors are the most influential and how they affect the outcome. In this context, f cannot be treated as a "black box" since we need to interpret the form and influence of each variable.

Techniques we will study in Supervised Learning in an upcoming week include the following.

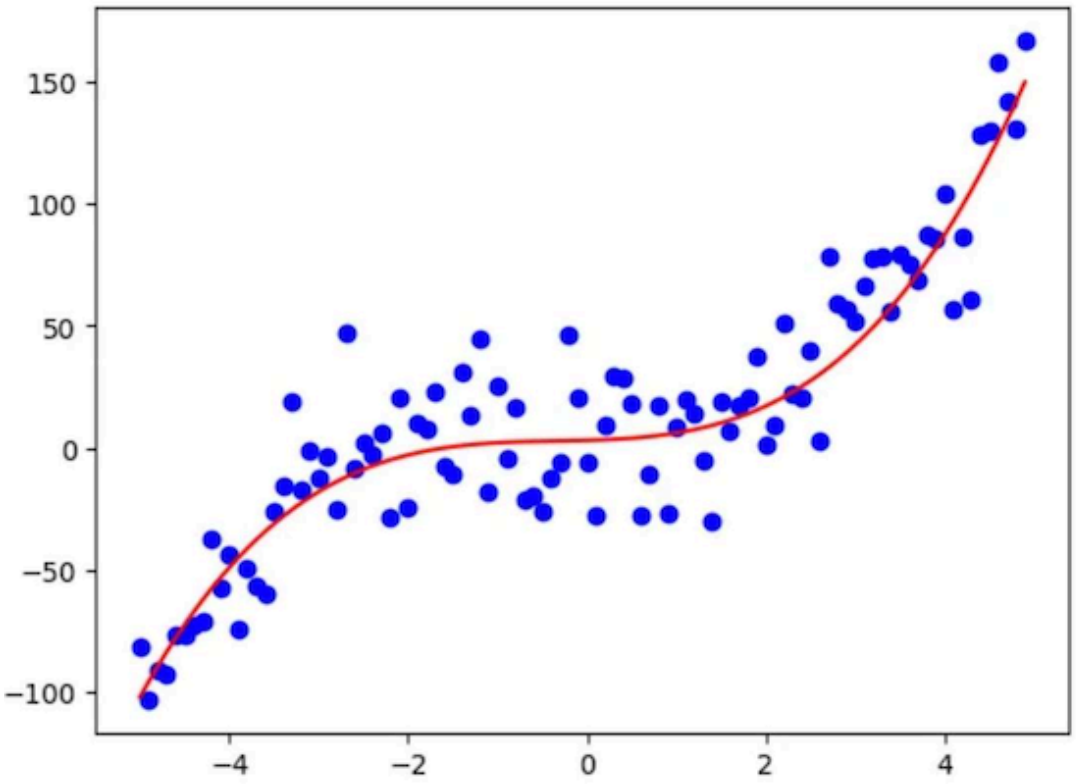
Linear Regression

Linear Regression models this relationship using a straight line, assuming a linear relationship between input and output.



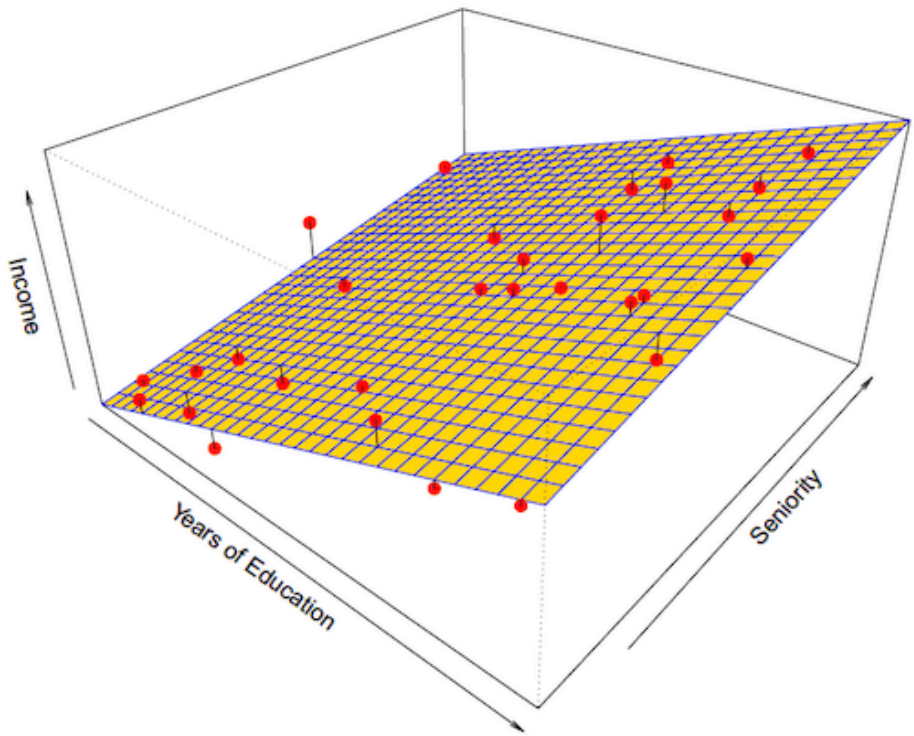
Polynomial Regression

When the underlying pattern in the data is non-linear, Polynomial Regression extends this idea by fitting a curve using higher-degree polynomials to capture more complex, non-linear relationships in the data.



Multiple Regression

In Multiple Regression, we apply linear or polynomial regression techniques to higher-dimensional datasets.



Example of Regression : House Price Prediction

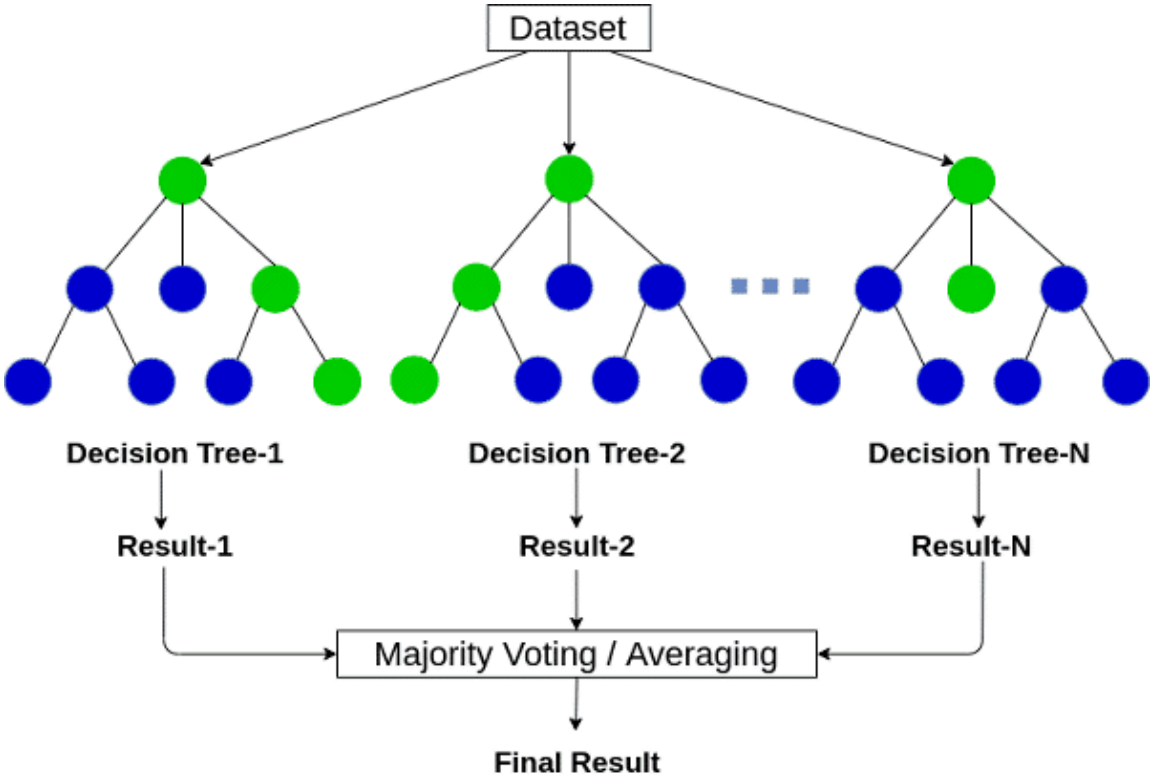
Suppose a developer wants to predict the price of new houses he wants to build based on a dataset that includes:

- X = (# of bedrooms, size in square feet, neighborhood, age in years).
- Y = price of the house in millions of dollars

Example We aim to estimate a function f that models the relationship between X and Y. Once we have trained our model, we can use it to predict the price for new houses based on their features. The dataset would have a 4D X and scalar Y. However, it is often useful to consider the relationship between a single feature and the output. Here is a regression of size in square feet vs. price of the house in millions of dollars.

Decision Trees

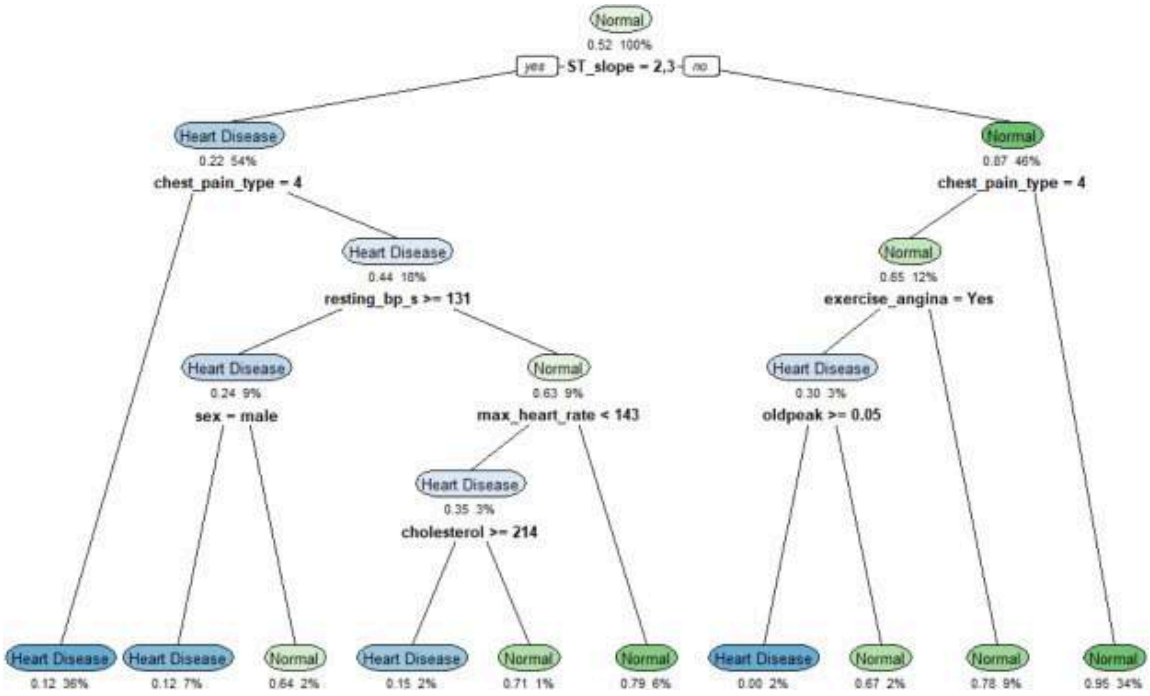
Decision trees partition data into smaller groups, making decisions based on the feature that best separates the data at each split. Random forests are collections of decision trees that combine their predictions to improve accuracy and reduce overfitting.



Example of Decision Trees: Medical Diagnosis

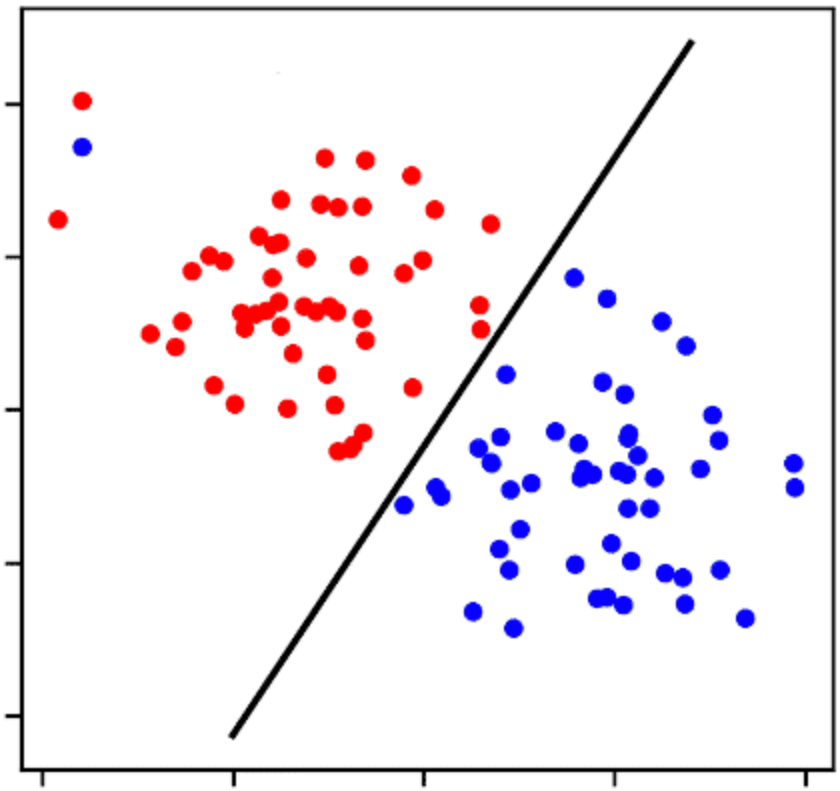
Decision trees can diagnose diseases based on patient symptoms, medical history, and test results. Such a model could predict whether a patient has a particular type of cancer by analyzing features such as tumor size, patient age, genetic markers, and other health factors.

The tree's structure allows doctors to interpret the decision-making process, providing explanations alongside predictions, which is valuable in clinical settings where the doctor ultimately makes the treatment decision.



Classification

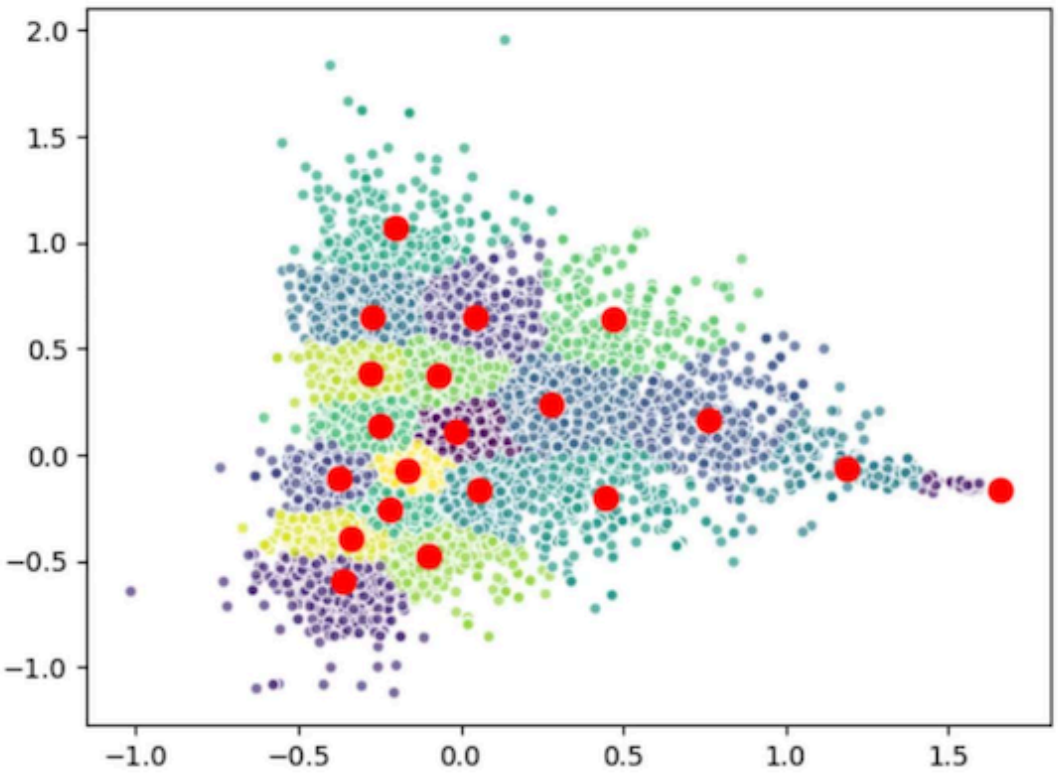
Classification is a type of supervised learning used to predict a categorical outcome by assigning input data to predefined classes or labels. It aims to learn decision boundaries that separate the classes, enabling the model to categorize new data with minimum error.



Example of Text Classification: The Kaggle 20 Newsgroups Dataset

Text classification is a common technique in the field of Natural Language Processing. Such systems can detect spam (versus useful “ham”), route customer complaints to the appropriate department, filter news articles, detect hate speech in social media, or help researchers retrieve similar documents from a large text database.

The Kaggle 20 Newsgroups Dataset(opens in a new tab) is a collection of about 20,000 newsgroup documents classified into 20 categories. Here is a K-Means classification model output that uses GloVe text embeddings as features.



Summary

Currently, much of the focus in machine learning is on supervised learning, as it allows us to achieve both accurate predictions for future observations and meaningful inferences about data relationships, which humans can hopefully interpret. Ideally, we strive not only to identify correlations but also to uncover causal relationships within the data.

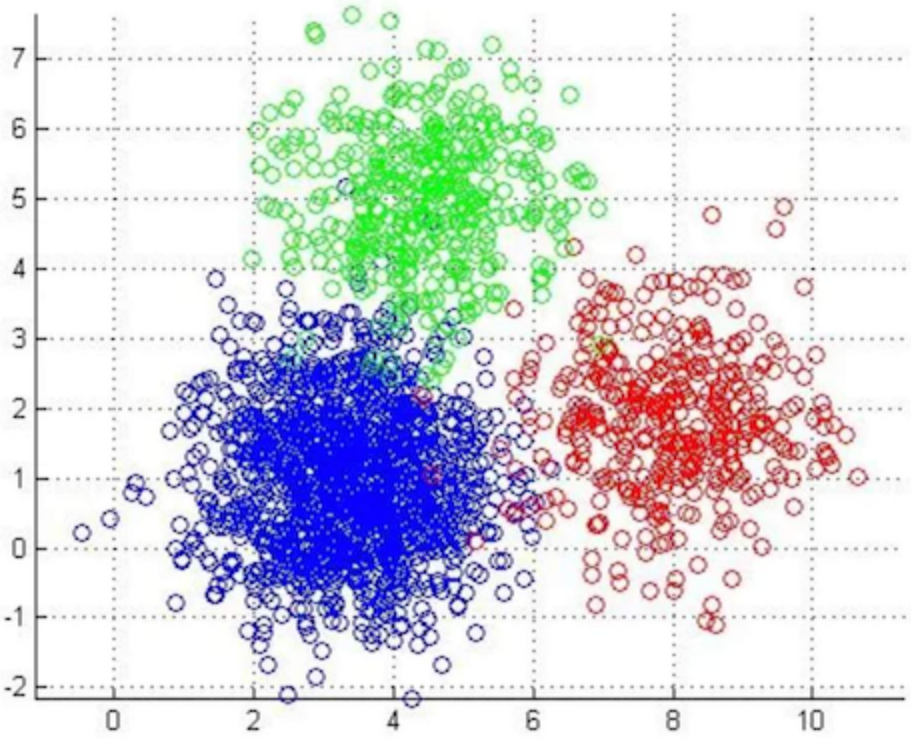
In this module, our emphasis will be on supervised learning. Next week, we will start our exploration with linear regression, which predicts a quantitative response by making certain simplifying assumptions, such that the relationship between X and Y is essentially linear.

Lesson 1.3: Unsupervised Learning

In situations where the response variable Y is not part of our dataset, our goal is to explore and discover patterns or structures within the data X. This is inherently exploratory, meaning we seek to identify relationships, groupings, or dimensionality reductions without a specific outcome variable to predict. The focus is on understanding the structure of the data rather than directly making predictions.

Clustering

Clustering is a technique used to group data points into clusters based on their similarity. The goal is to find natural groupings within the data so that points within the same cluster are more similar to each other than to those in other clusters. The response variable Y returns a cluster identifier for each X.

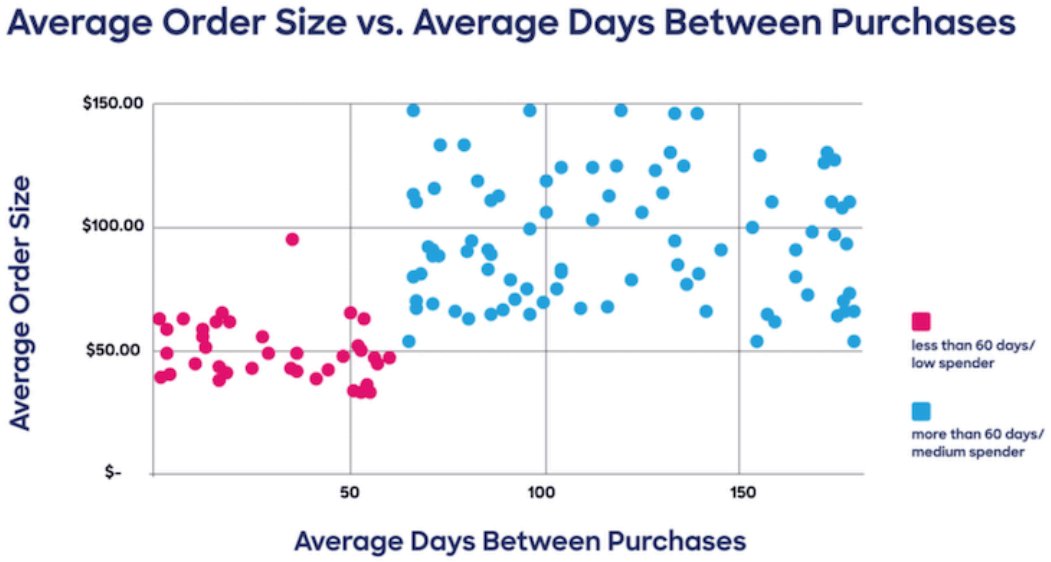


Clustering Example: Customer Segmentation in Marketing

Scenario: A company wants to segment its customers based on purchasing behavior, demographics, and preferences.

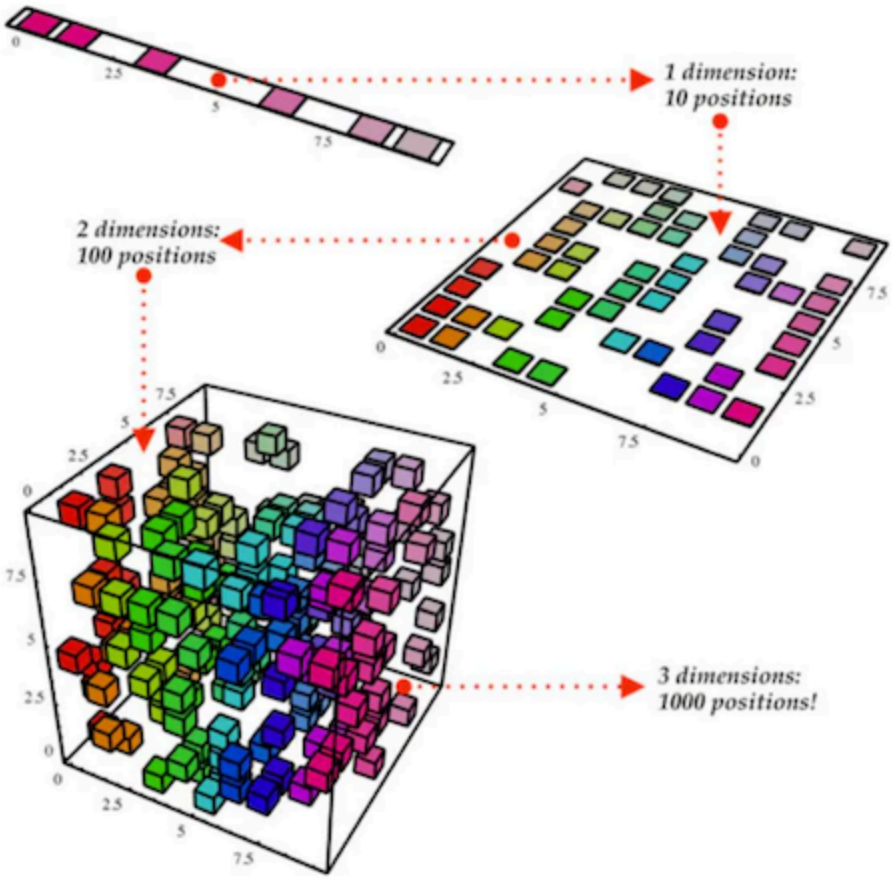
In this case, there is no specific response variable Y to predict. Instead, we only have predictors such as customer age, size of purchases, frequency of purchases, types of products bought, and average spending.

The goal is to find natural groupings or clusters in the data that help the company understand different types of customers. This can allow them to tailor marketing campaigns for each group, even though we are not predicting a specific outcome.



Dimensionality Reduction

Dimensionality Reduction reduces the number of features (dimensions) in a dataset while retaining as much important information as possible. It is particularly useful when dealing with high-dimensional data, as it can help simplify models, speed up computations, and make visualization possible. The technique aims to capture the underlying structure in the data while discarding noise or less informative features.

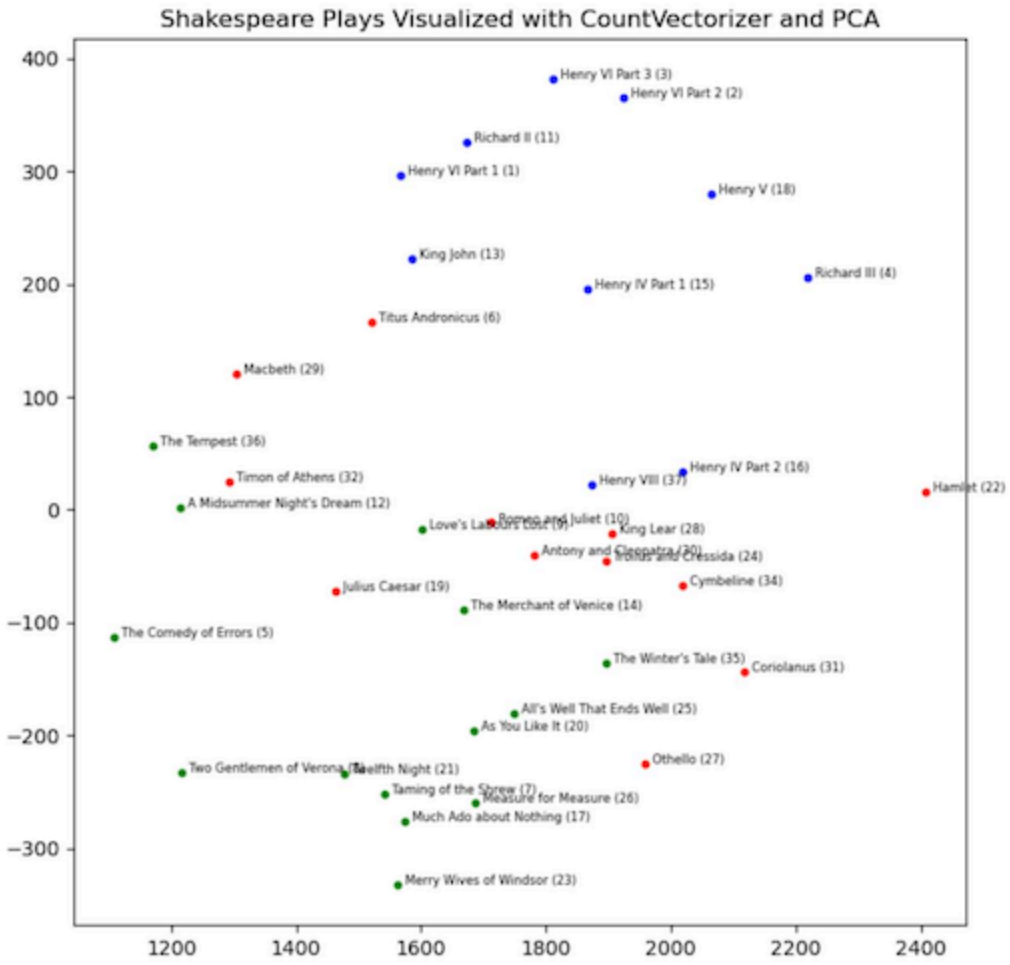


Dimensionality Reduction Example: Stylometric Analysis of Shakespeare's Plays using PCA

Scenario: Stylometry is a method of linguistic analysis that quantifies various features of texts—such as word frequency, sentence length, grammatical structures, and word patterns—to compare different writing styles objectively. By applying these methods, we can discover stylistic traits that might not be obvious through traditional literary analysis, giving us a data-driven perspective on the text to augment more traditional approaches.

In the scatterplot below, we applied stylometric analysis to Shakespeare's plays using a "bag of words" model, which counts the frequency of the 22,000 most common words across all of Shakespeare's works. Using Principal Components Analysis (PCA), this 22,000-dimensional dataset was reduced to just 2 dimensions to allow us to visualize the relationships between the plays more easily. Different genres are indicated by color: comedies in green, histories in blue, and tragedies in red. The numbers in parentheses show the order in which the plays were written.

A rough grouping by genre, suggesting stylistic similarities, can be observed, with a few intriguing exceptions. For instance, in the figure below that visualizes groupings of Shakespeare plays, "Othello" groups more closely with the comedies, while "The Tempest" appears closer to the tragedies. Interestingly, "Hamlet" appears isolated, perhaps highlighting its complexity and thematic uniqueness.



Summary

Although this module primarily emphasizes supervised learning techniques, we will briefly consider unsupervised techniques for clustering and dimensionality reduction in Week 12.

Think About It:

- Clustering techniques often rely on the distance between points and try to group “close” points into the same clusters. Is there a simpler way to cluster the customer segmentation scatterplot given above?
- Consider the scatterplot of the Shakespeare dataset. If the genres were not considered, how would you decide on the number of clusters, and how well would they correspond to the actual genres?
- Models for large, high-dimensional datasets are computationally expensive to create, and reducing the number of dimensions before training a model seems sensible. Can you see any drawbacks in this approach when evaluating the results?

1.4 Lesson: How do we evaluate ML Models?

We shall now consider the interplay of several important notions in the design and evaluation of models, which will concern us throughout our study of ML. We are primarily interested in the issues inherent in supervised learning, although similar characterizations can be made for unsupervised learning.

Accuracy vs Interpretability

- **Accuracy** is a quantitative measure of how often a machine learning model correctly predicts or classifies outcomes. According to scikit learn's Metrics documentation, there is a bewildering variety of precise metrics available, but in the simplest case, it is simply the number of correct results divided by the number of data instances. We can measure the model's accuracy on the training set, but the more important metric is the accuracy on the test set, as this measures how well the model generalizes.
- **Interpretability** is the extent to which a machine learning model's decisions can be understood by humans. It allows users to trace and explain how inputs lead to particular outputs or predictions.

Although one of these is precise and quantifiable and the other is not, the tension between these two is a fundamental challenge in ML. Highly accurate models, such as deep neural networks, often act as 'black boxes,' making their decision-making process difficult to understand and explain. While these complex models can achieve remarkable performance on a wide range of tasks, their lack of transparency limits their applicability in fields where understanding the rationale behind predictions is crucial, such as healthcare or finance.

On the other hand, simpler models such as linear regression or decision trees are more interpretable, allowing us to trace how features contribute to predictions, but they may sacrifice some accuracy in favor of transparency.

Balancing accuracy and interpretability involves choosing the right model for the context and ensuring that both performance and the ability to explain the results meet the project's goals.

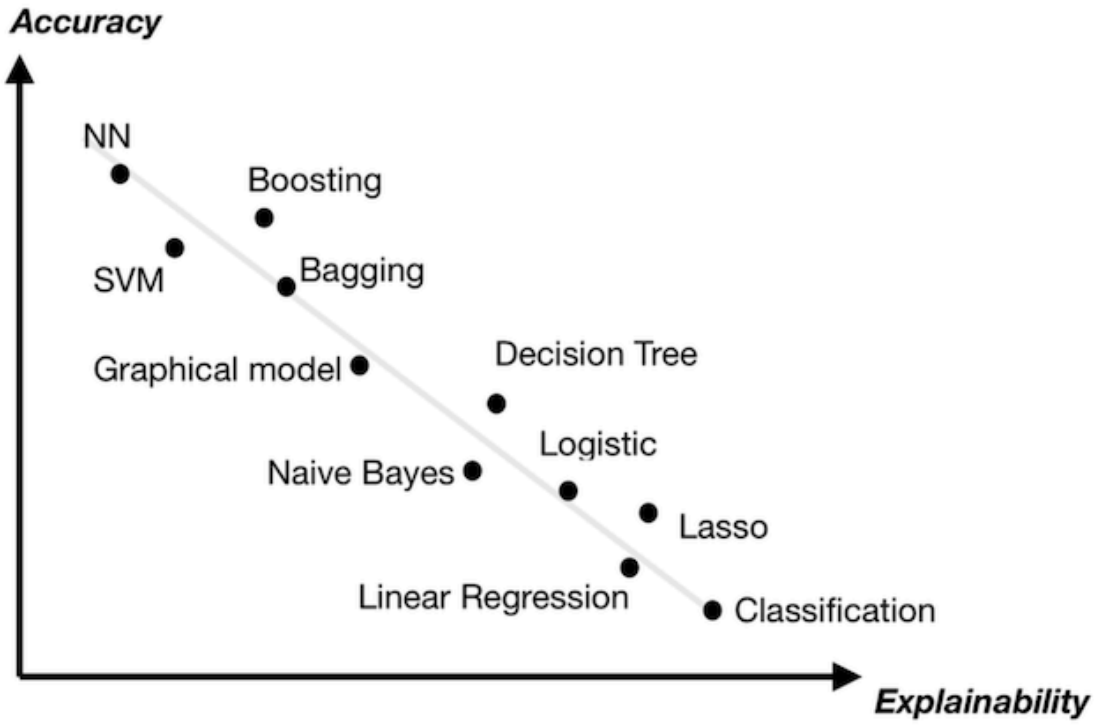


Figure 1: The Accuracy-Interpretability Tradeoff in ML Models

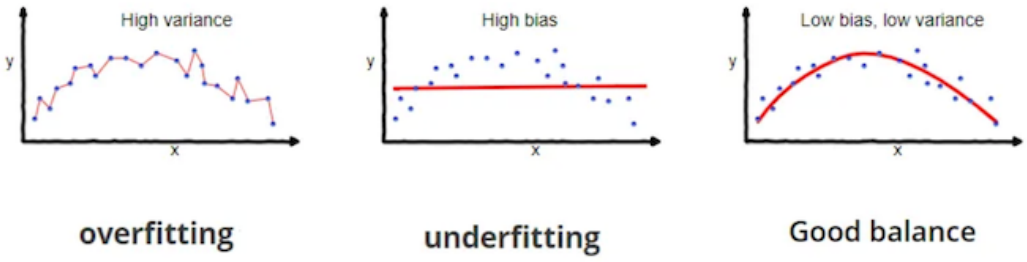
Two Types of Error and the Bias-Variance Tradeoff

As we shall explore in more detail in week 3, it is useful when developing models to separate the error term $e# into two components:

- **Irreducible Error** is due to inherent noise or randomness in the data that is meaningless in our goal of finding the relationship between X and Y. This can be caused by measurement errors, missing features, or simply unpredictable variations in the data.
- **Reducible Error** is the component of error that can be minimized by improving the model. It occurs when the model does not adequately capture the underlying patterns in the data—often due to poor model selection, insufficient training, or incorrect parameter tuning.
- The **bias-variance tradeoff** helps explain different sources of **reducible** error in machine learning models. It is especially useful when thinking about why models do not gneeralize well to new data.
- **Variance** is the component of reducible error produced by a too-complex model. This high variance causes the models to fit the details of the trainign data too closely, making it less able to generalize to new data. It "misses the forest for the trees."
- **Bias**, on the other hand, refers to the compoent of reducible error that a too-simple model produces. High-bias models do not fit the training data well enough, so they

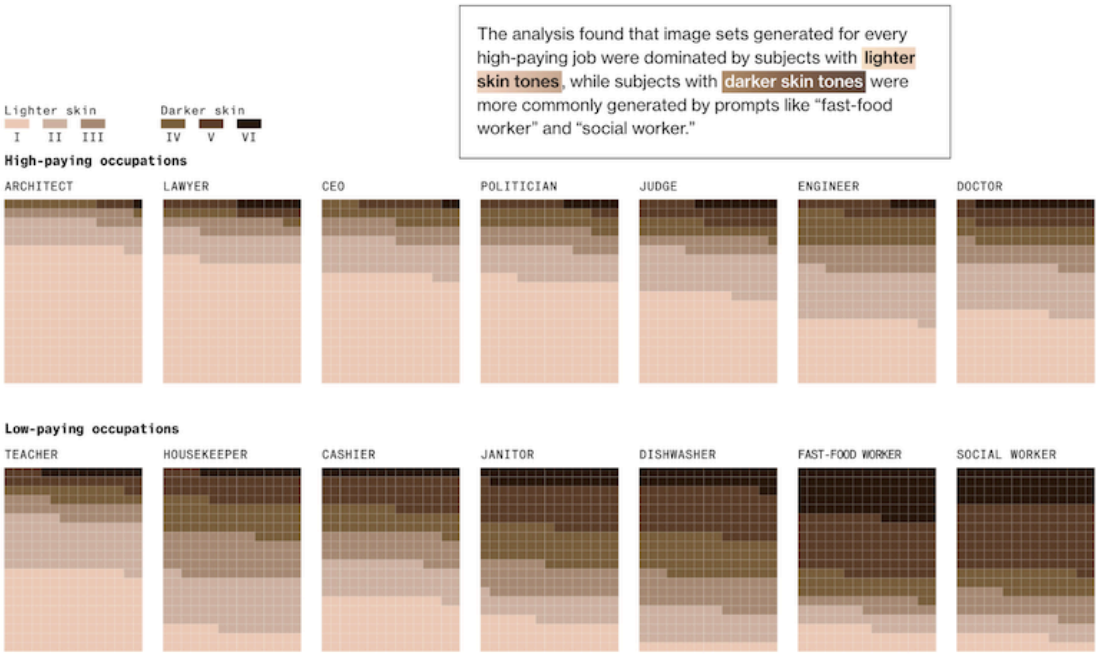
will have systematic errors when applied to a test test. THery're not even in the right forest!

The **Bias-Variance tradeoff** is about minimizing the two sources of reducible error. A model with high bias **underfits** because it fails to learn the underlying relationships, while a model with high variance **overfits** by being too sensitive to the training data. The goal is to balance these errors, reducing overall reducible errors while recognizing that irreducible errors cannot be eliminated. Balancing high variance and high bias is key to creating models that generalize well. Unlike the qualitative terms “underfitting” and “overfitting,” the bias-variance tradeoff can be quantified precisely using statistical techniques (covered in Week 3).



Biased Data and the Fairness Conundrum

In machine learning, biased datasets can lead to unfair outcomes, especially when training data does not represent all groups equitably. This can result in models that amplify existing inequalities or exhibit discriminatory behavior, as seen in hiring algorithms that favor male candidates due to biased historical data or facial recognition systems that perform poorly on individuals with darker skin tones. Such biases also affect areas like lending, where algorithms can unjustly deny loans to marginalized groups. Addressing this 'fairness conundrum' involves understanding sources of bias, recognizing their impact, and employing techniques such as re-sampling, algorithmic adjustments, or post-processing methods to mitigate unfairness. But definitive solutions are few, and generative AI charges ahead without them.



Think About It

- Is there any hope that humans will create unbiased algorithms if they are biased?
- What are the ethical implications of deploying face-recognition systems exhibiting racial bias, and how might these implications vary depending on the context?