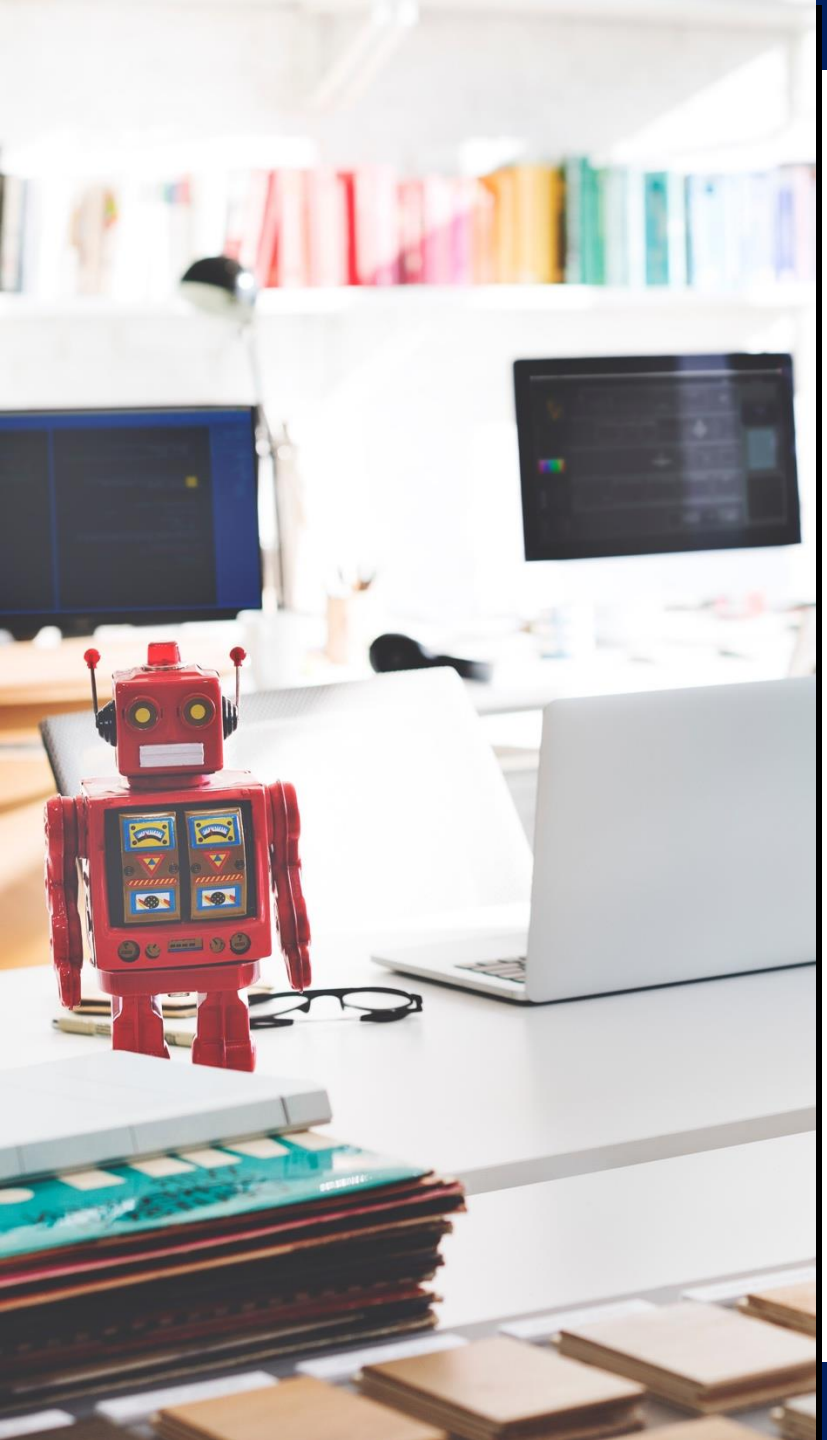
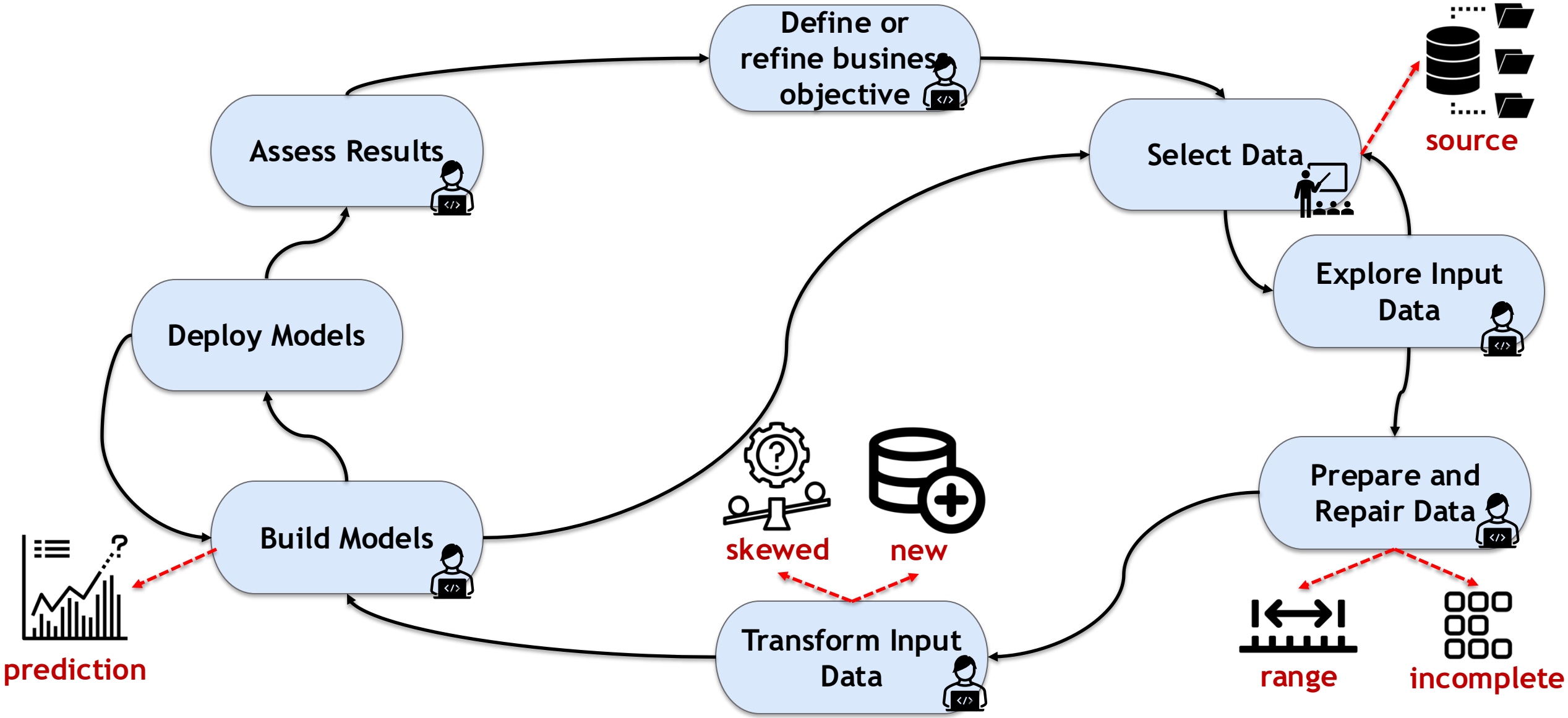


Bias in Machine Learning

Code smells: Understanding the domain of A1

Data Science Project - Process Overview



What do you understand by bias?

A Little Bit of Philosophy: Aristotle

- Greek philosopher
  - Why should we live?
    - The pursuit of happiness
    - Eudemonism: literally translating to the state or condition of ‘good spirit’
  - Polymath

- One of the founding fathers of democracy
  - Relying on someone else's authority was completely against the spirit of Aristotle's research

- Theory of “Dualism”

Dualism

- Aristotle wanted to reduce and structure the complexity of the world. Used Pythagoras's *Table of Opposites*
  - finite, infinite; odd, even; one, many; right, left

- Applied the dualism to people, animas, and society
  - items he ordained to have more worth became 1s,
  - and those item of lesser importance 0s

↓ He wrote about women: "*The relation of male to female is by nature a relation of superior to inferior and ruler to ruled.*"
  - 1 = true = rational = right = male
  - 0 = false = emotional = left = female

Source: https://finance.yahoo.com/news/aristotle-binary-philosophies-created-today-133048283.html?soc_src=social-sh&soc_trk=ma

Binary System

- Gottfried Wilhelm Leibniz (1689) → 0s and 1s
  - ↑ Large numbers can be reduced and calculated efficiently
  - ↑ Conversion of the alphabet and punctuation into ASCII
  - ↓ Aristotle's biased basis

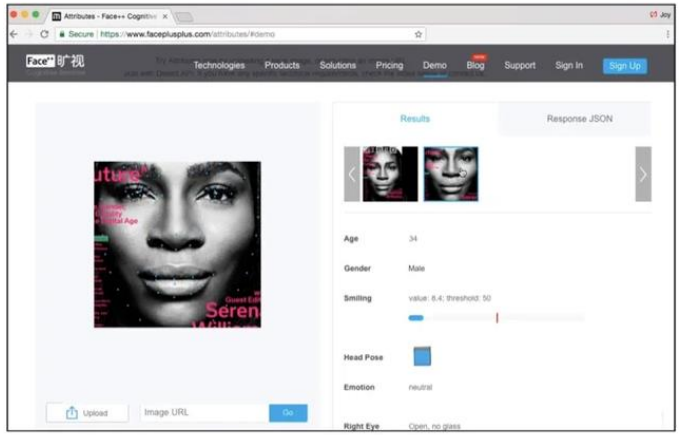
- Aristotle's binary classifications are now manifest throughout today's data systems
  - Tinder swiping right = 1, swipe left = 0
  - NLP frameworks: positive = 1, negative = 0
  - Clicking "like" on Facebook = 1, not clicking like = 0

Source: https://finance.yahoo.com/news/aristotle-binary-philosophies-created-today-133048283.html?soc_src=social-sh&soc_trk=ma

Joy Buolamwini: Gender and Racial Bias

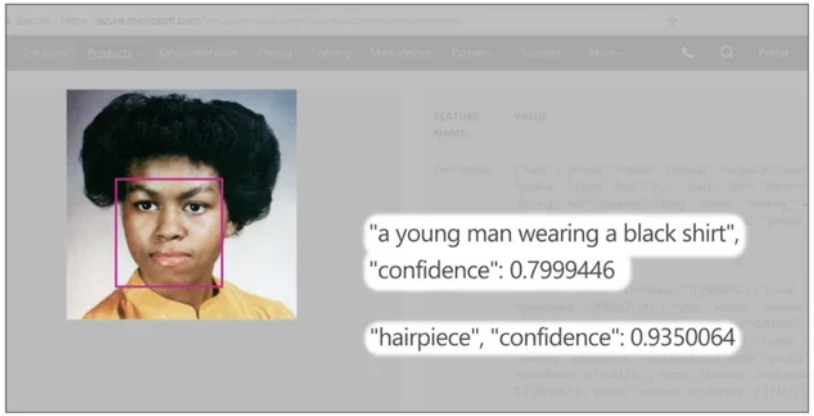
- Research uncovered large gender and racial bias in AI systems
  - Task: guess the gender of a face
  - IBM, Microsoft, and Amazon
- All companies performed substantially better on male faces than female faces
  - Error rates of no more than 1% for lighter-skinned men
  - For darker-skinned women, the errors soared to 35%

Serena Williams



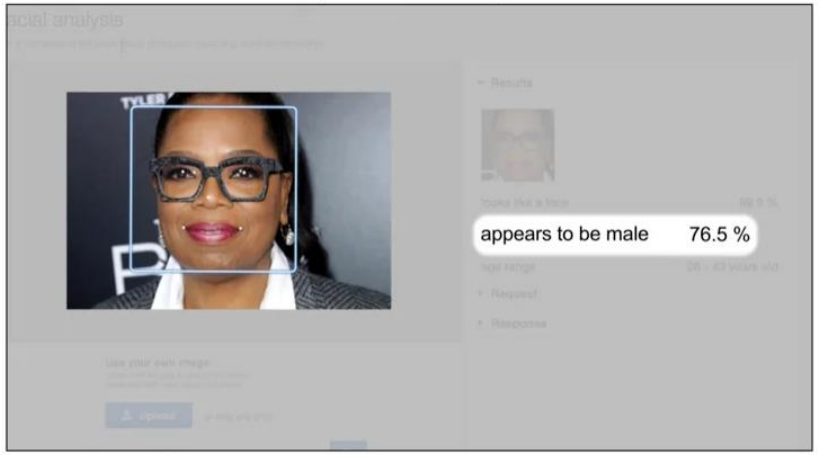
Face++ 旷视

Michelle Obama



Microsoft

Oprah Winfrey



amazon

Source: https://time.com/5520558/artificial-intelligence-racial-gender-bias/

Bias in Machine Learning

- Bias is the model's inability to predict accurately, leading to some difference or error between the model's expected and actual values
    - Bias refers to systematic errors that lead the model to make inaccurate predictions or poor generalizations
    - Bias can originate from various sources, including the data, algorithms, or assumptions made during the training process

Bias

- Bias measures the error introduced by approximating a real-world problem (which may be complex) with a simplified model

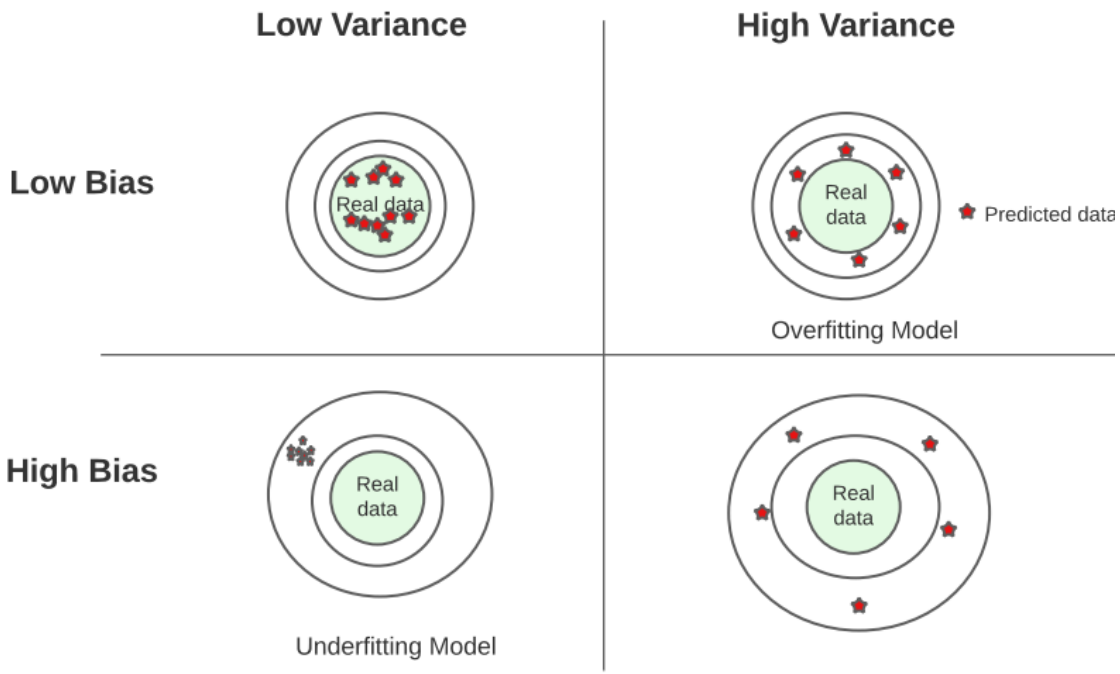
- E.g.: use the NBA players (6’6”) to find the male height average in US (5’9”)

Bias and Variance Trade-off

Low Variance

It indicates that the model's predictions do not change much when trained on different subsets of the data

- It is stable and produces consistent results across different datasets



High Variance

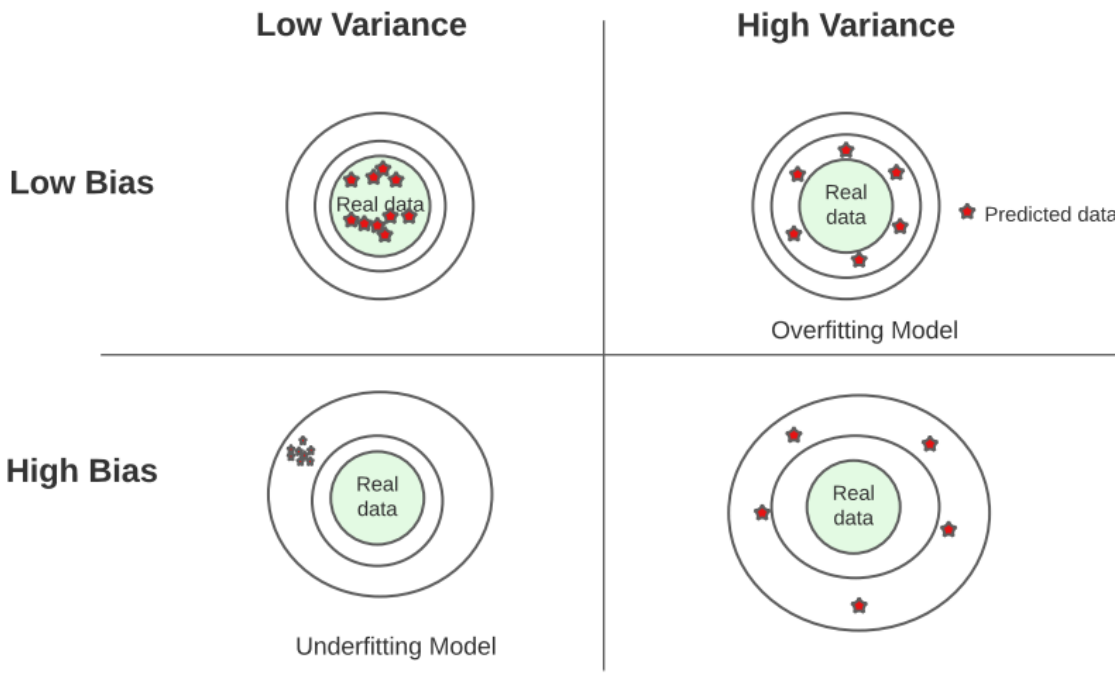
Model is overly complex and fits the noise in the training data rather than just the underlying patterns

- It tends to perform well on the training data but poorly on the test data or new data.

Bias and Variance Trade-off

Low Bias

A low-bias model makes fewer assumptions about the data and can fit the training data closely, capturing all the patterns, including the noise



High Bias

Model makes strong assumptions about the data, leading to a significant difference between the predicted values and the actual values

- A high-bias model will perform poorly on both the training and test datasets because it oversimplifies the underlying relationships.

Different Views of Bias in ML

Algorithm Bias

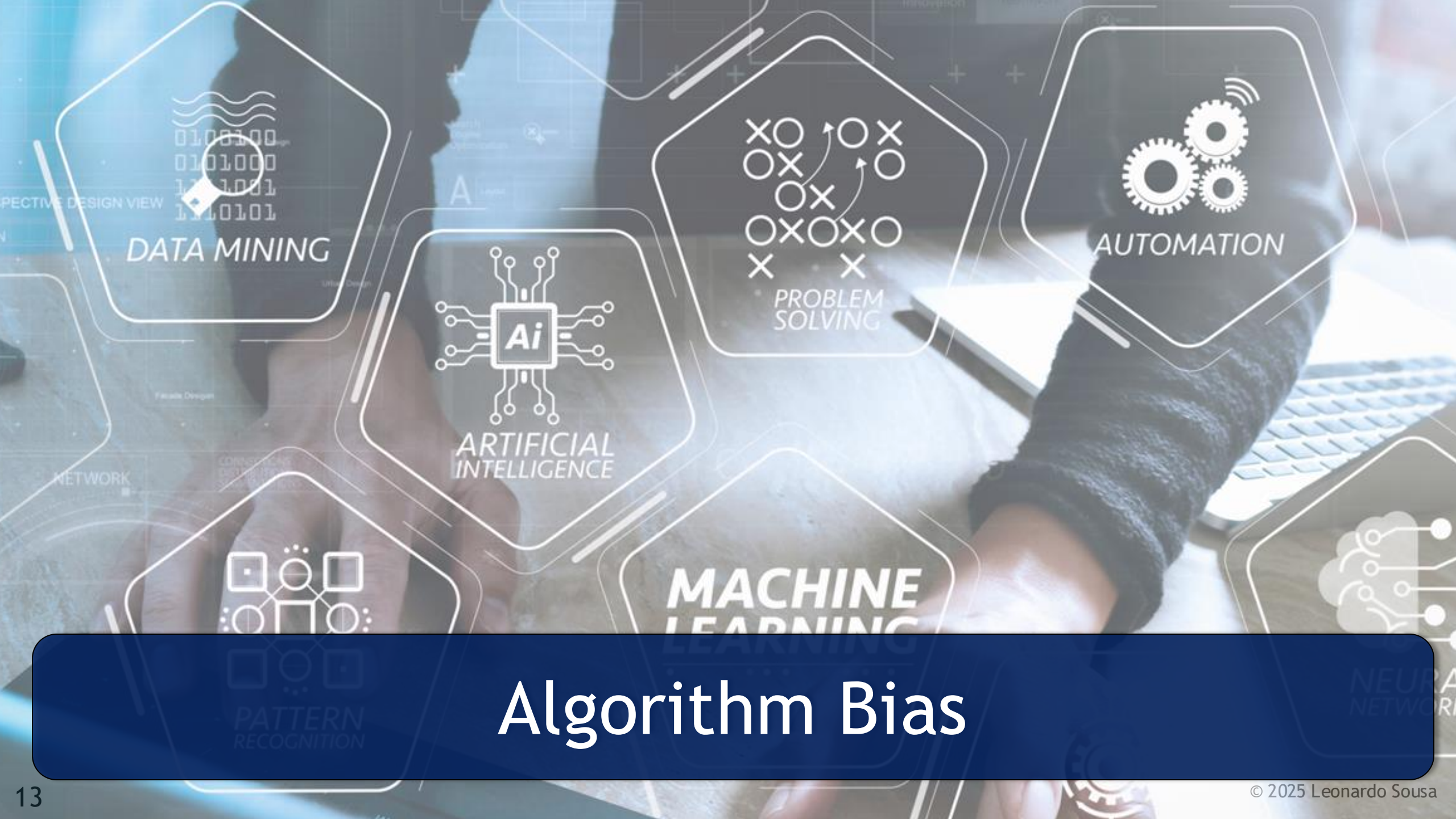
This occurs when there's a problem within the algorithm that performs the calculations or other processing that powers the ML computations

Evaluation Bias

Some evaluation metrics might favor certain models over others, especially if they don’t fully capture the relevant context

Data Bias

Occurs when the training data is not representative of the real-world distribution, leading to a model that performs poorly on unseen data



Algorithm Bias

Algorithm Bias

- It refers to biases that arise from the **assumptions** made by the learning algorithms, which affect how they process data and make predictions
  - This bias is typically the result of the algorithm's structure, or the decisions made during the training process that led to systematic errors or incorrect predictions
  - It is often related to the algorithm's ability to generalize from training data to unseen data.

Algorithm Bias

- It is the tendency to consistently learn the wrong thing by not considering all the information in the data (underfitting)

COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	PASTA
HEAT	STOVE
TOOL	SPATULA
PLACE	KITCHEN

COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	FRUIT
HEAT	—
TOOL	KNIFE
PLACE	KITCHEN

COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	MEAT
HEAT	GRILL
TOOL	TONGS
PLACE	OUTSIDE

COOKING

ROLE	VALUE
AGENT	WOMAN
FOOD	VEGETABLES
HEAT	STOVE
TOOL	TONGS
PLACE	KITCHEN

COOKING

ROLE	VALUE
AGENT	MAN
FOOD	—
HEAT	STOVE
TOOL	SPATULA
PLACE	KITCHEN

In this example of gender bias, adapted from a report published by researchers from the University of Virginia and the University of Washington, a visual semantic role labeling system has learned to identify a person cooking as female, even when the image is male.

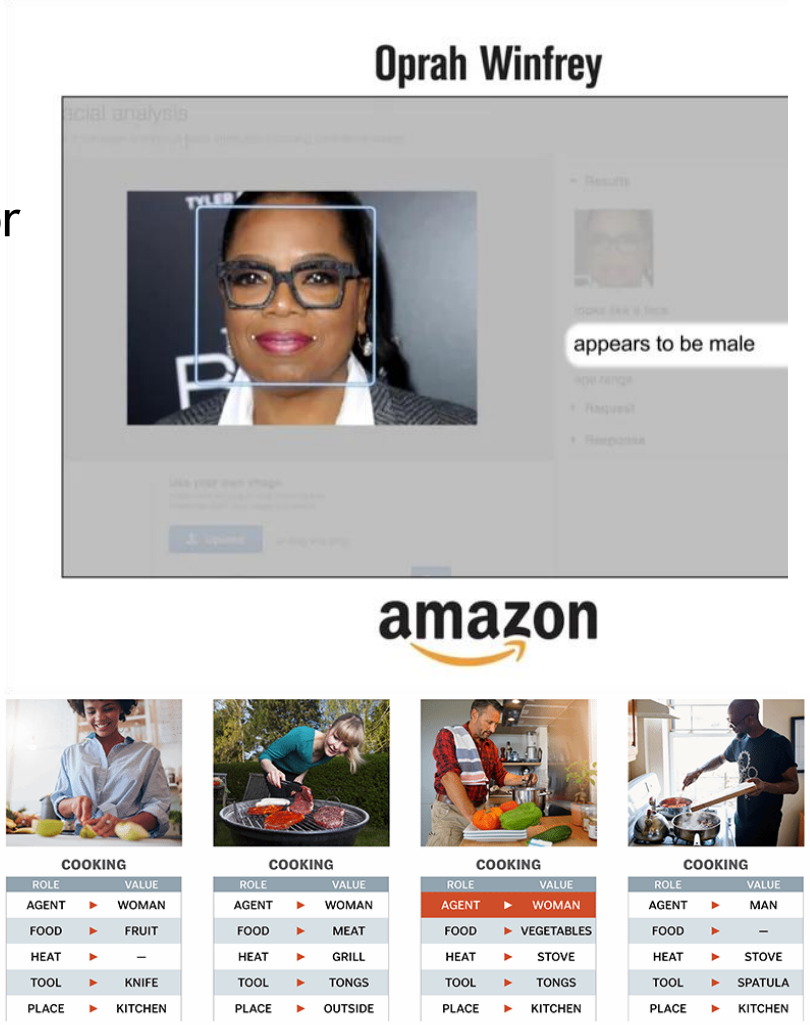
Source: https://searchenterpriseai.techtarget.com/definition/machine-learning-bias-algorithm-bias-or-AI-bias

Example of Algorithm Bias

Example	Bias	Outcome
Facial Recognition	Race Bias	Higher error rates for people of color.
Loan Approval	Socioeconomic Bias	Lower approval rates for applicants with lower incomes.
Criminal Justice	Racial Bias	Higher rates of false positives for individuals of certain racial backgrounds.

Algorithm Bias Due to Data Representation

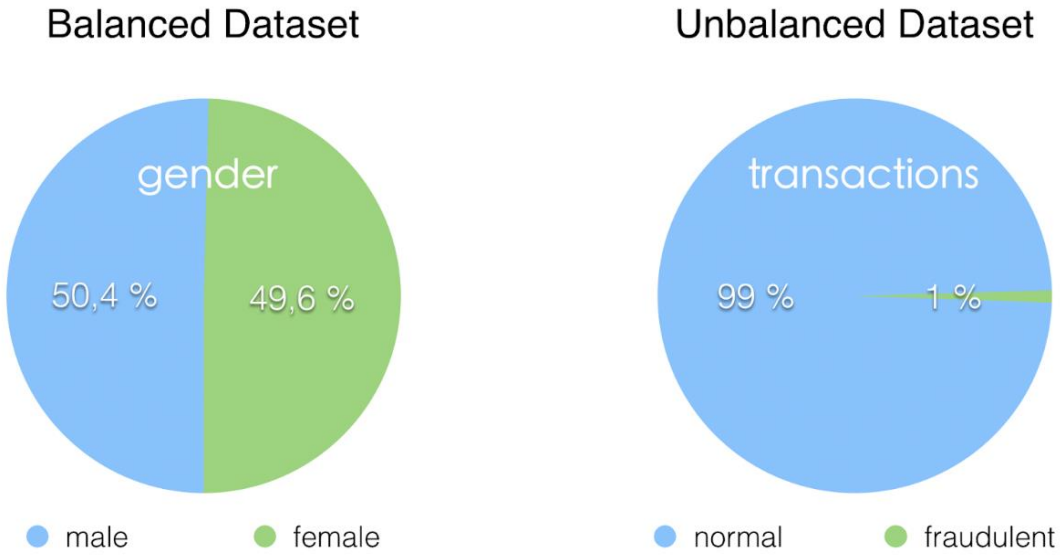
- The way data is represented can influence how algorithms learn patterns.
- **Feature Selection Bias**: If certain important features are missing or underrepresented, the model may learn incorrect associations
- **Imbalanced Training Data**: If certain groups/classes are overrepresented, the model may favor them
- Example: A facial recognition system trained on mostly light-skinned faces may perform poorly on darker-skinned individuals.



In this example of gender bias, adapted from a report published by researchers from the University of Virginia and the University of Washington, a visual semantic role labeling system has learned to identify a person cooking as female, even when the image is male.

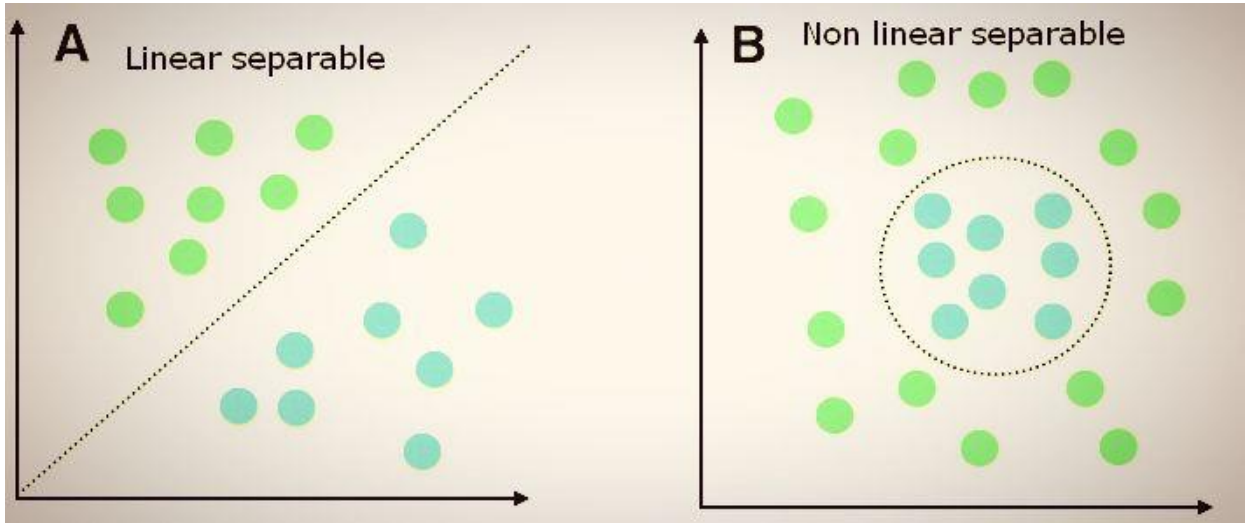
Algorithm Bias from Learning Objectives

- The choice of loss function and optimization criteria can introduce bias.
- **Optimizing for Accuracy in Imbalanced Datasets**: A classifier might predict the majority class most of the time, appearing "accurate" but failing for minority classes.
- **Mean Squared Error (MSE) in Regression**: MSE gives more weight to larger errors, which may bias predictions toward the majority population



Algorithm Bias due to Model Architecture

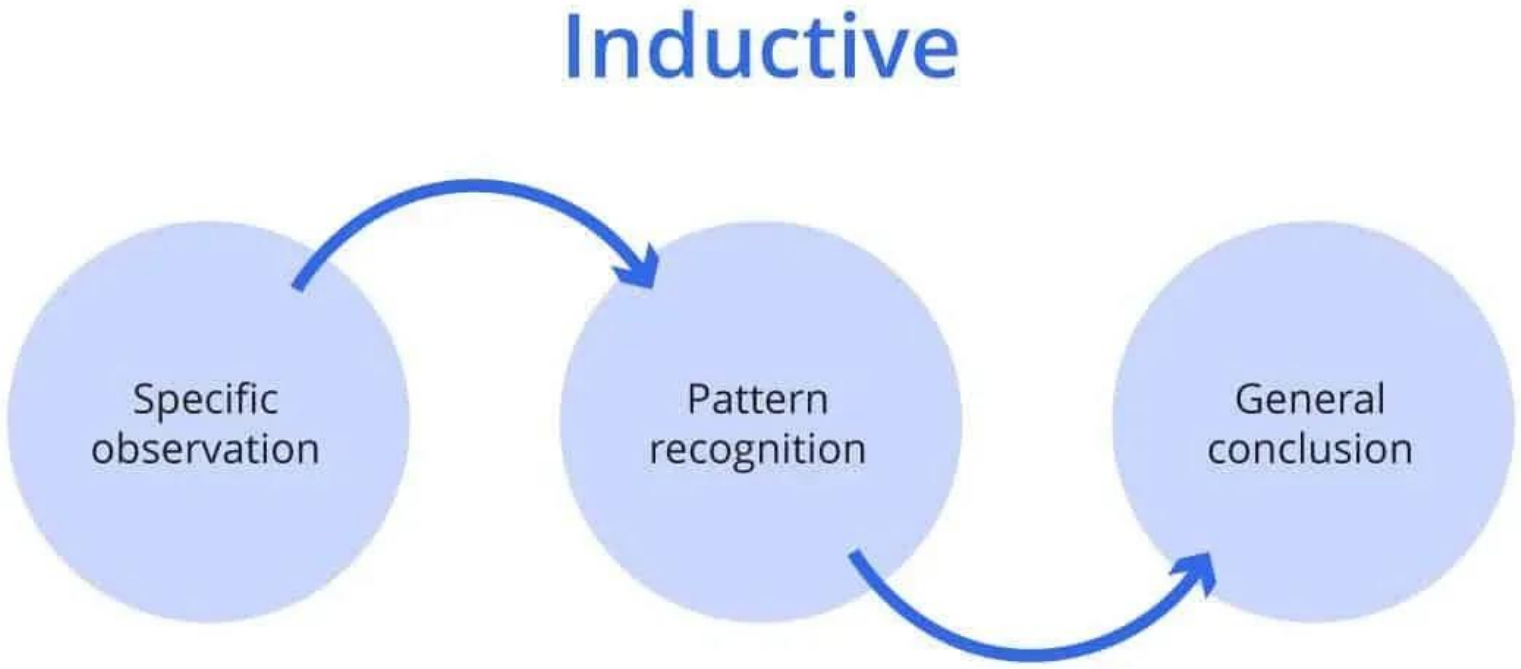
- Certain model architectures inherently introduce bias
- **Linear Models**: These assume linear relationships in the data, which may not hold for complex tasks



- **Deep Learning Models:** They may be overfitting to the most frequent patterns and fail to generalize to rare but important cases

Inductive Learning → Inductive Bias

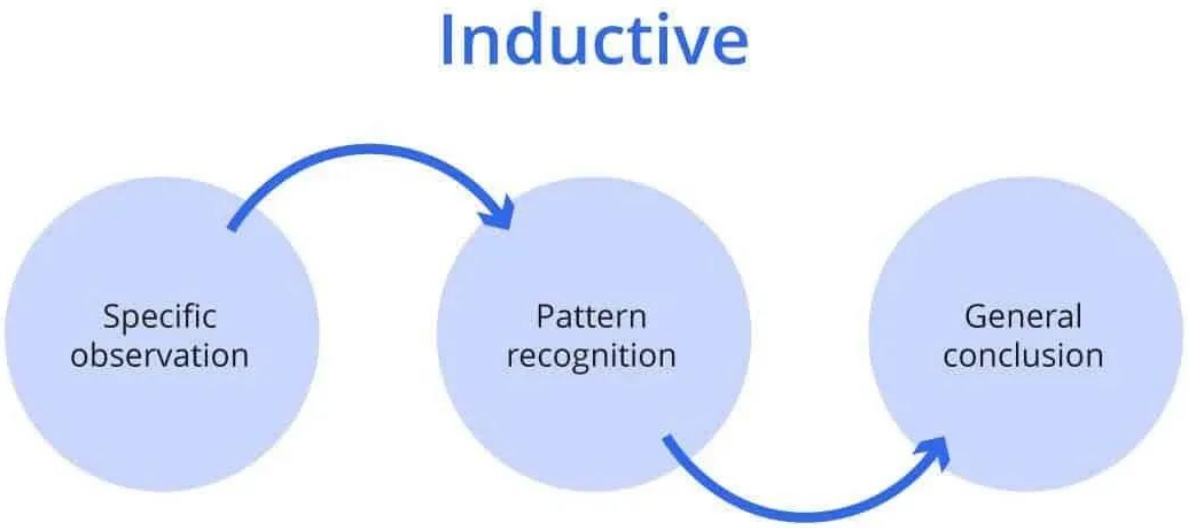
- Inductive learning involves generalizing from specific instances or examples to make broader generalizations or predictions
  - It also known as inductive reasoning or inductive inference



Inductive Bias

- Every ML algorithm operates with an inductive bias, which refers to the set of assumptions the algorithm uses to predict outputs given new inputs

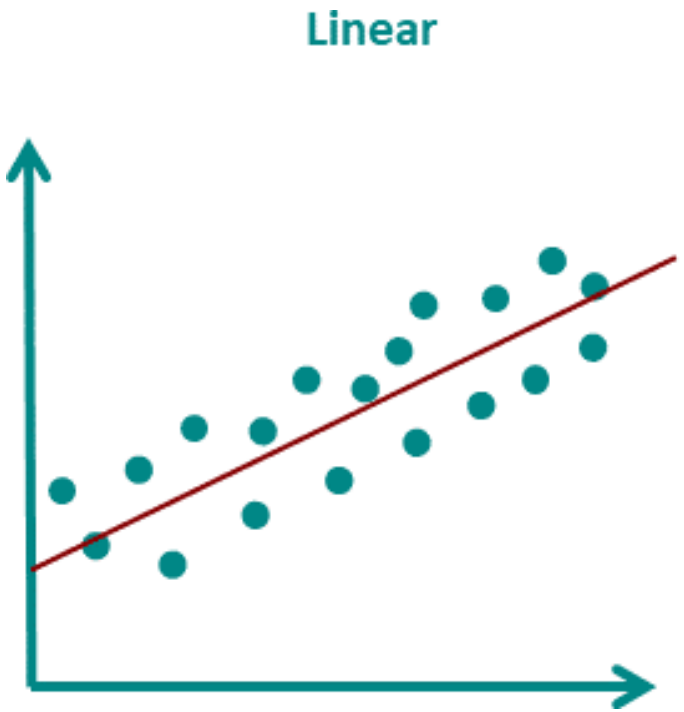
- Since data alone is not always sufficient for learning (especially with limited or noisy data), inductive biases help guide the learning process by making certain assumptions about the underlying patterns in the data

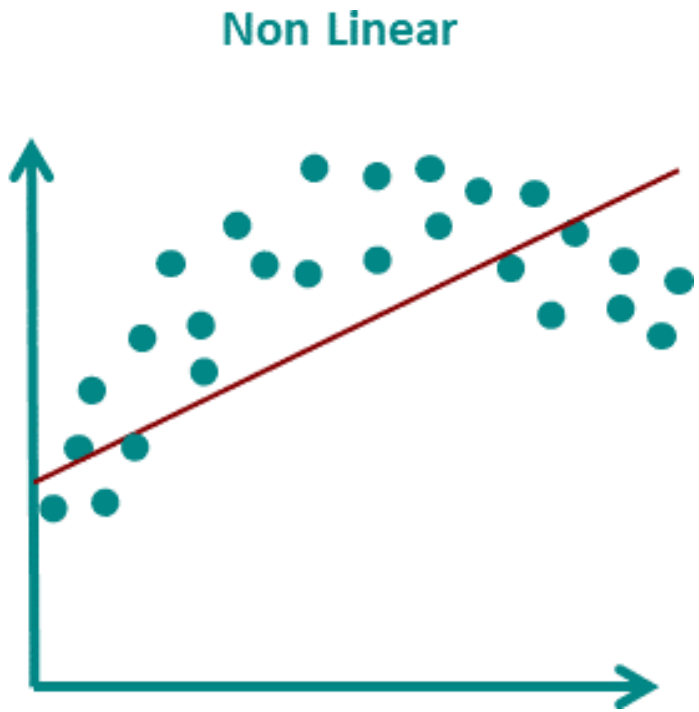


Inductive Bias: Linear Regression

- A **linear regression** model assumes a linear relationship between the input features and the target variable

- If the actual relationship is non-linear, this inductive bias leads to poor performance because the model cannot capture the true complexity of the data

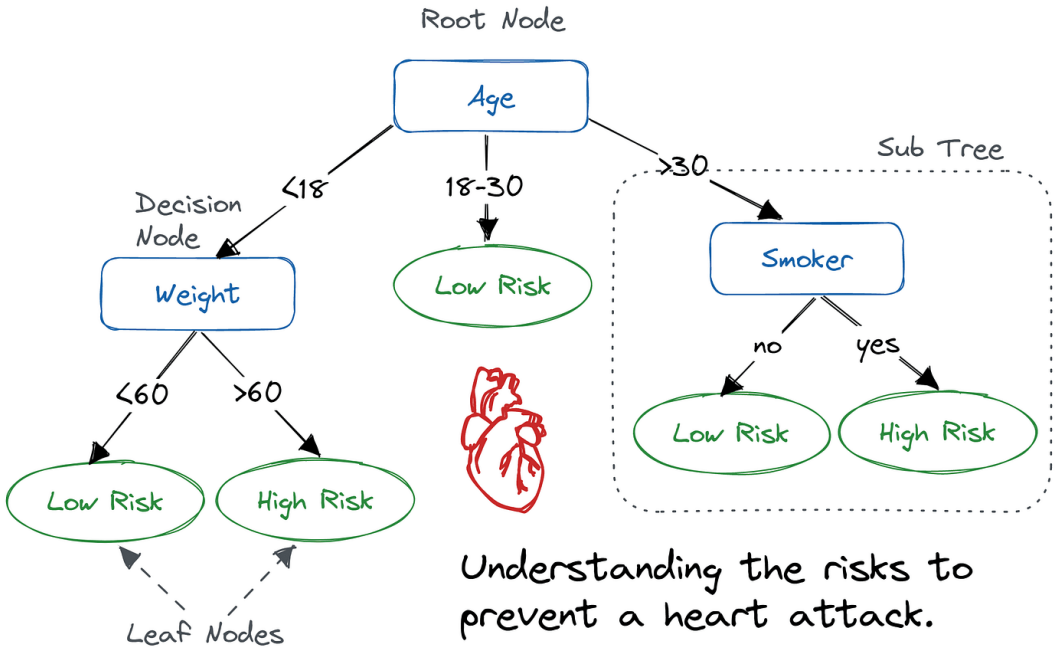


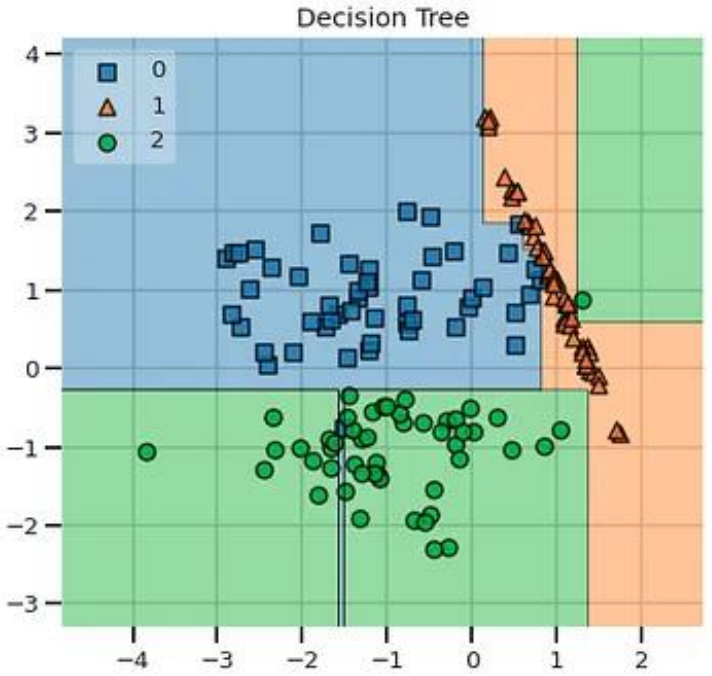


Inductive Bias: Decision Tree

- A **decision tree** algorithm favors partitioning the data based on feature values
  - which may lead it to capture fine-grained patterns (high variance) and overfit small datasets

- One of its main inductive biases is the assumption that an objective can be achieved by asking a series of binary questions
  - As a result, the decision boundary of the tree classifier becomes orthogonal





Recognizing Algorithm Bias

1

Analyze and Understand Data and Algorithm

- Examine the algorithm to understand its behavior
- Examine the training data to find underrepresentation or overrepresentation of certain groups

2

Assess Model Performance

Test the model's performance across different groups to identify disparities in accuracy or fairness

3

Investigate Predictions

Examine the predictions made by the model to understand how they relate to the training data and potential biases

Mitigating Algorithm Bias

Fairness Metrics

Using fairness metrics to evaluate the model's performance across different groups, ensuring equitable outcomes

Bias Mitigation Techniques

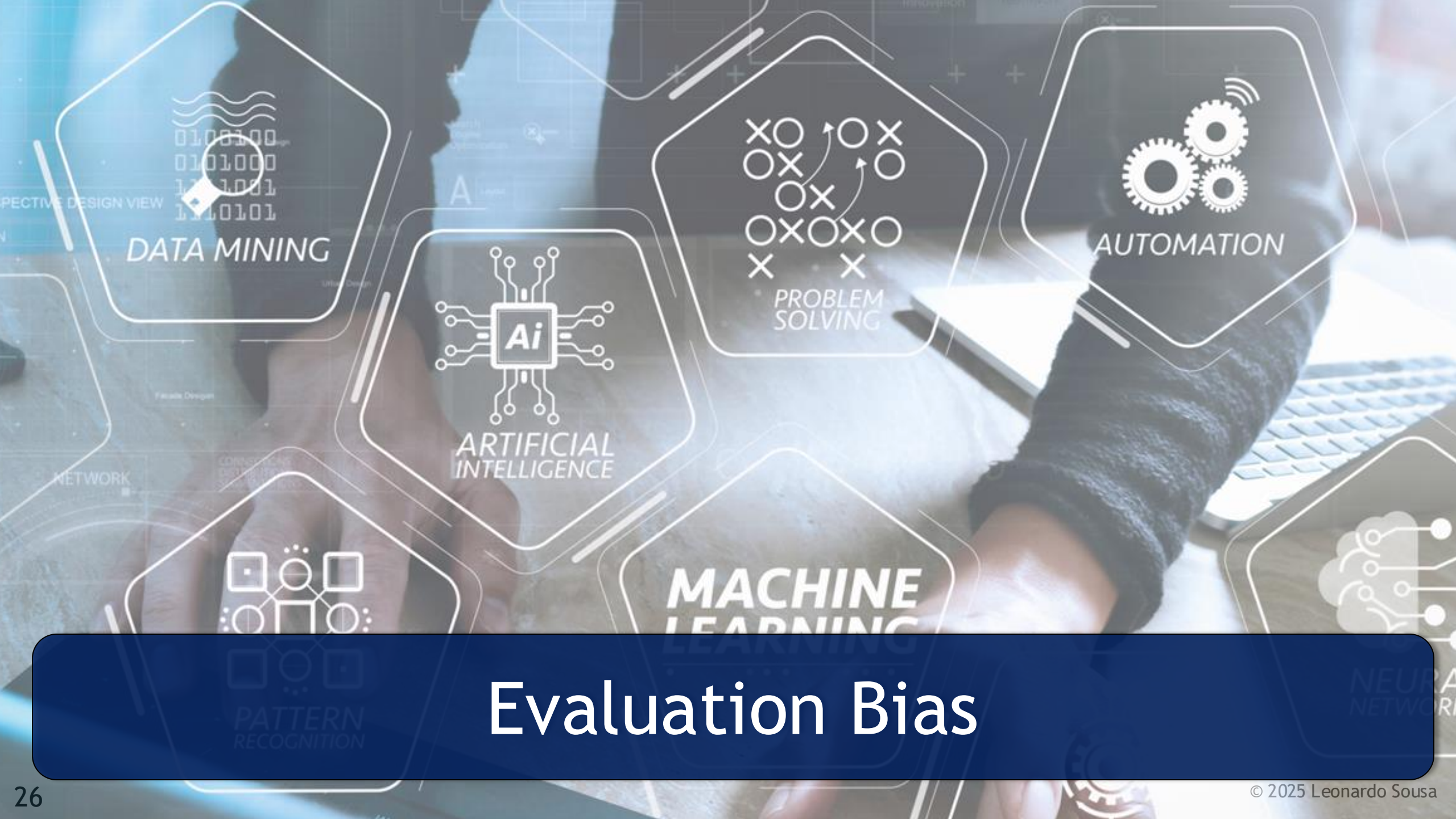
Implementing techniques such as adversarial training or calibration to reduce bias in the model's predictions.

Transparency and Explainability

Promoting transparency and explainability in the model's decision-making process, allowing for scrutiny and identification of potential biases.

Ethical Guidelines

Developing and adhering to ethical guidelines for the development and deployment of machine learning systems, including considerations for bias mitigation.

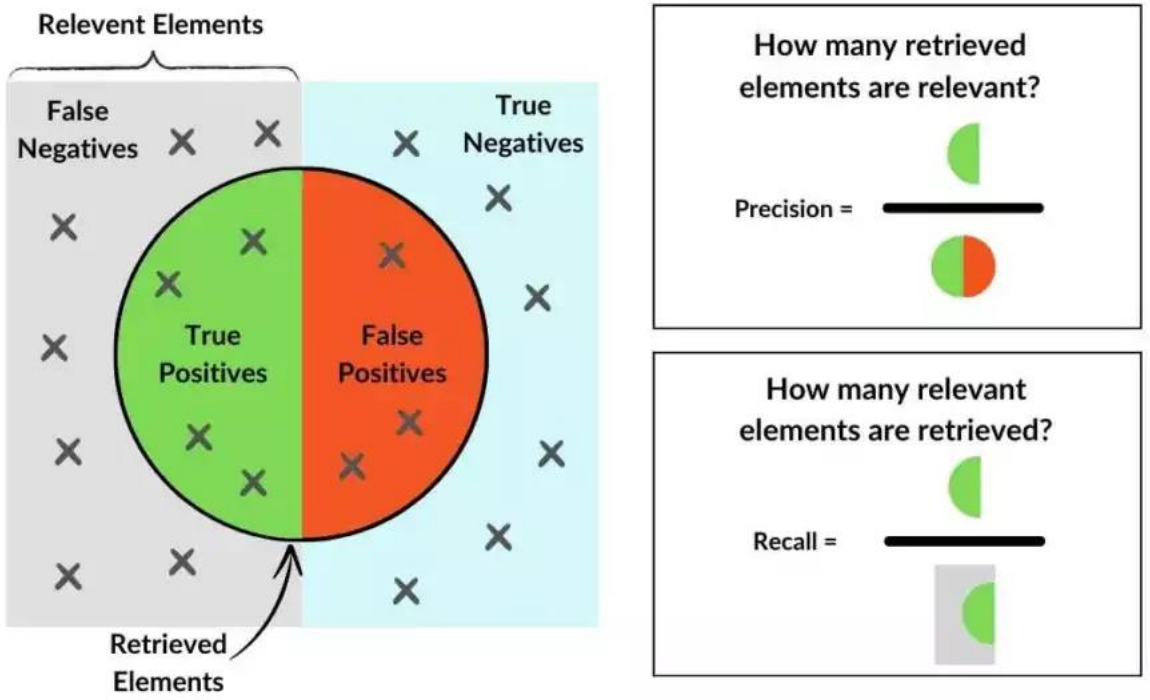


Evaluation Bias

Evaluation Bias

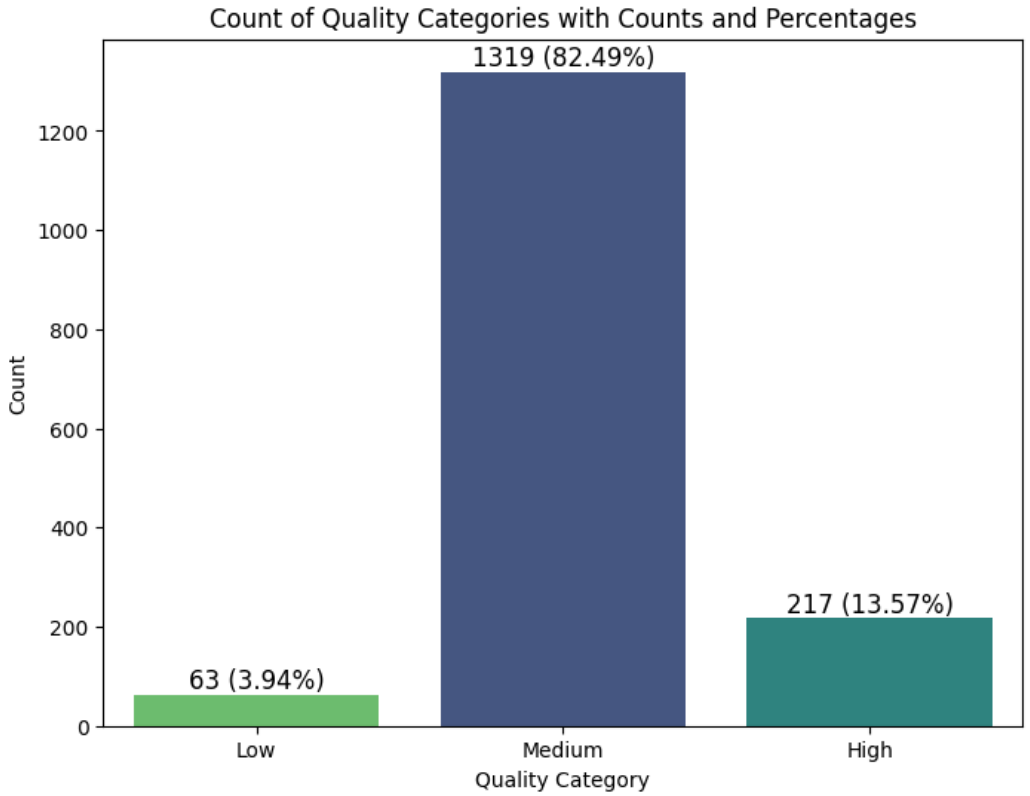
- It occurs when certain evaluation metrics favor specific models over others in a way that does not fully reflect their real-world performance

- This can happen due to limitations in how the metric captures the underlying task or because the metric is misaligned with the actual goals of the application



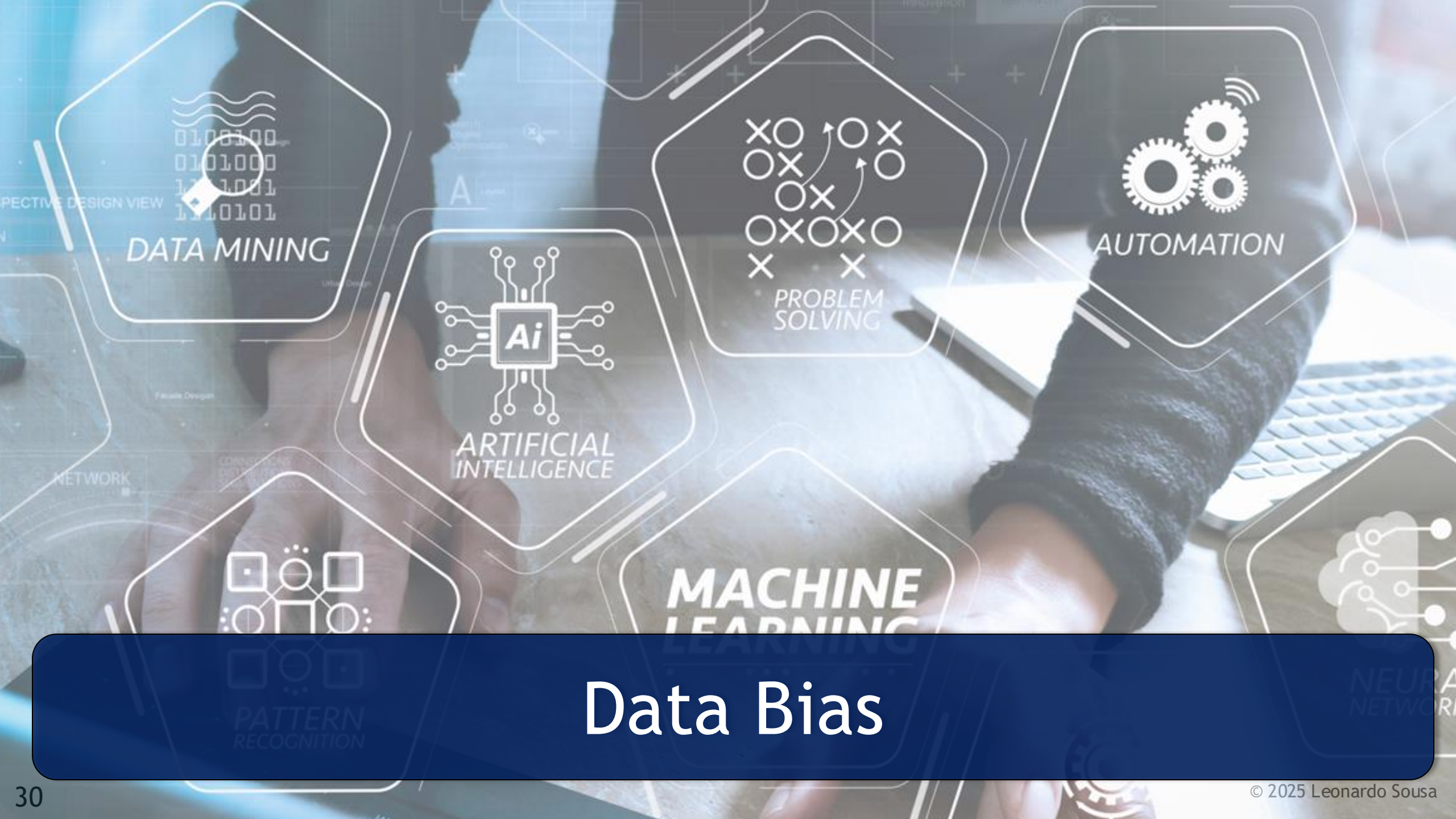
Ignoring Class Imbalance

- Some metrics may prioritize certain types of errors over others
- Example: **Accuracy** may favor models that predict the majority class well but fail on minority classes in imbalanced datasets



Choice of Metrics

- If a dataset has imbalanced classes, metrics like **accuracy** may not reflect performance correctly
  - Example: In a fraud detection problem (where fraud is only 1% of the data), a model predicting "no fraud" for all cases could achieve 99% accuracy but be useless
- A single metric may not capture all relevant aspects of a model's performance
  - Example: Using only **Precision** might favor models that avoid false positives but suffer from high false negatives
- Many metrics depend on thresholds (e.g., 0.5 for classification probabilities)
  - Example: Changing the threshold in an **ROC** curve can make a model appear better or worse.



Data Bias

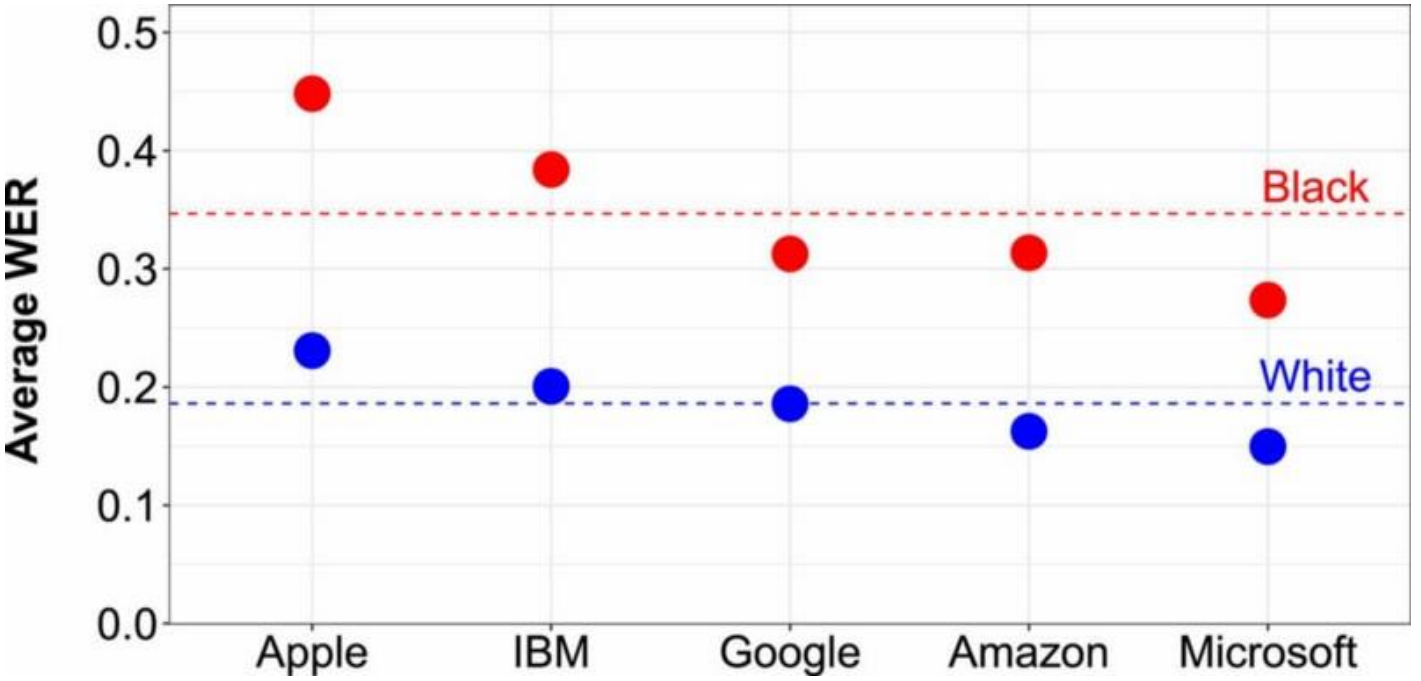
Data Bias

- When there's a problem with the data used to train the machine learning model
  - Training data does not accurately reflect the makeup of the real-world usage of the model

Example: training a speech-to-text system from audio clips together with their corresponding transcriptions

Sample: audiobooks

Bias: well, educated, middle aged, white men



Word Error Rate [WER] for speech recognition system

Understanding Data Bias

1 Representation Bias

Occurs when the training data doesn't accurately reflect the real world

This is common when certain groups are underrepresented or overrepresented in the data

2 Measurement Bias

Arises when data is collected or measured in a way that systematically favors certain groups or outcomes

3 Historical Bias

Embodied in datasets that reflect past societal biases, such as racial discrimination or gender inequality

4 Sampling Bias

Arises when the sample data used for training is not representative of the population it's intended to model

Examples of Data Bias

Facial Recognition

Facial recognition systems trained on primarily white faces may struggle to identify people of color accurately.

Loan Approvals

Loan approval algorithms trained on historical data that reflects past discriminatory practices may perpetuate those biases

Hiring Processes

Hiring algorithms trained on historical data that reflects biases against certain groups may perpetuate those biases in the hiring process

Causes of Data Bias

Human Biases

The people collecting, labeling, or selecting data may unknowingly introduce their own biases into the process

Limited Data Availability

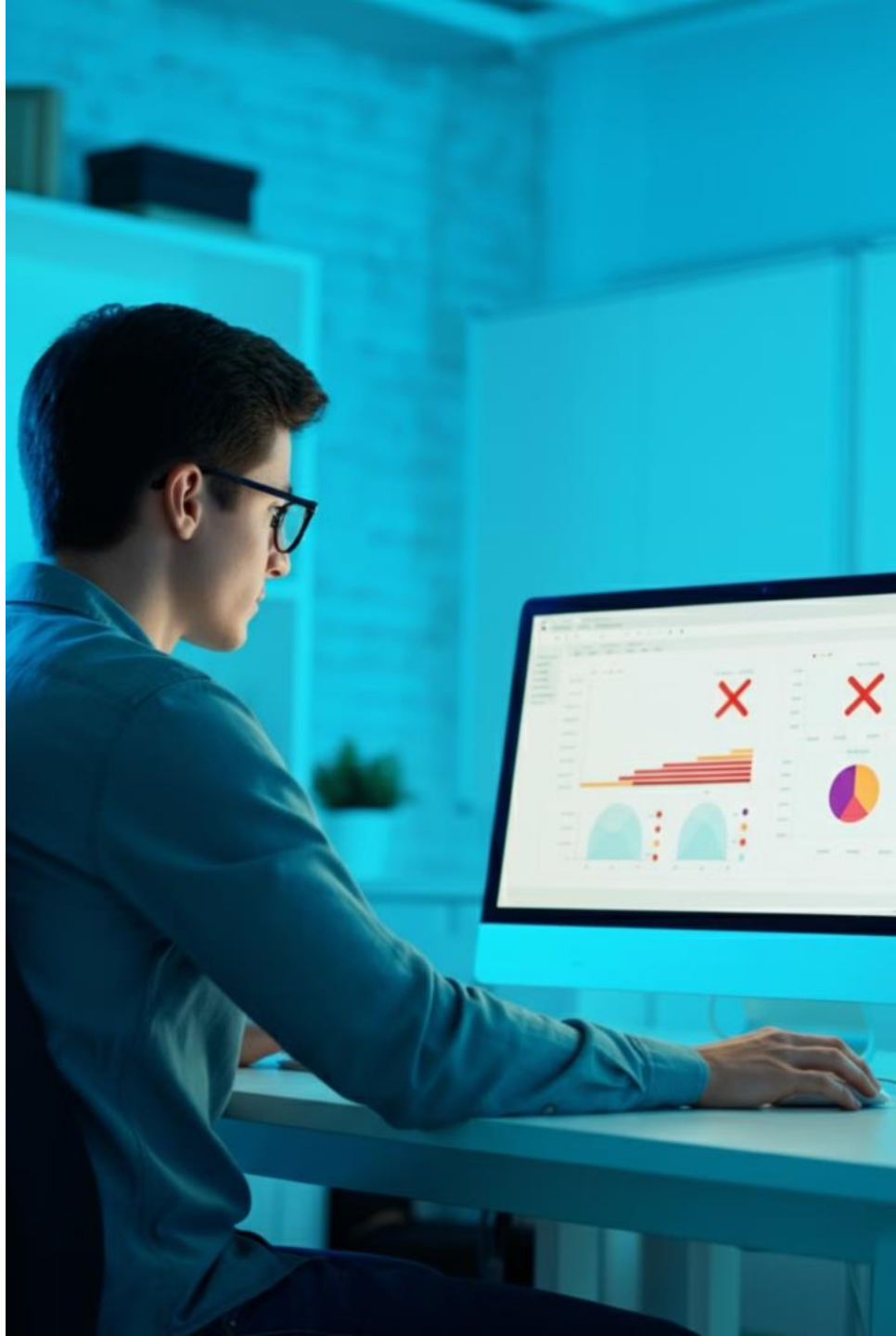
Lack of data from diverse groups can lead to models that perform poorly on those groups.

Data Collection Methods

The way data is collected can inherently favor certain groups or outcomes.

Historical Injustices

Historical biases embedded in data can be difficult to identify and rectify



Mitigatin Data Bias

1

Data Augmentation

Expanding the data set to include more diverse and representative examples can help mitigate biases

2

Data Balancing

Addressing imbalances in the data set by oversampling underrepresented groups or undersampling overrepresented groups

3

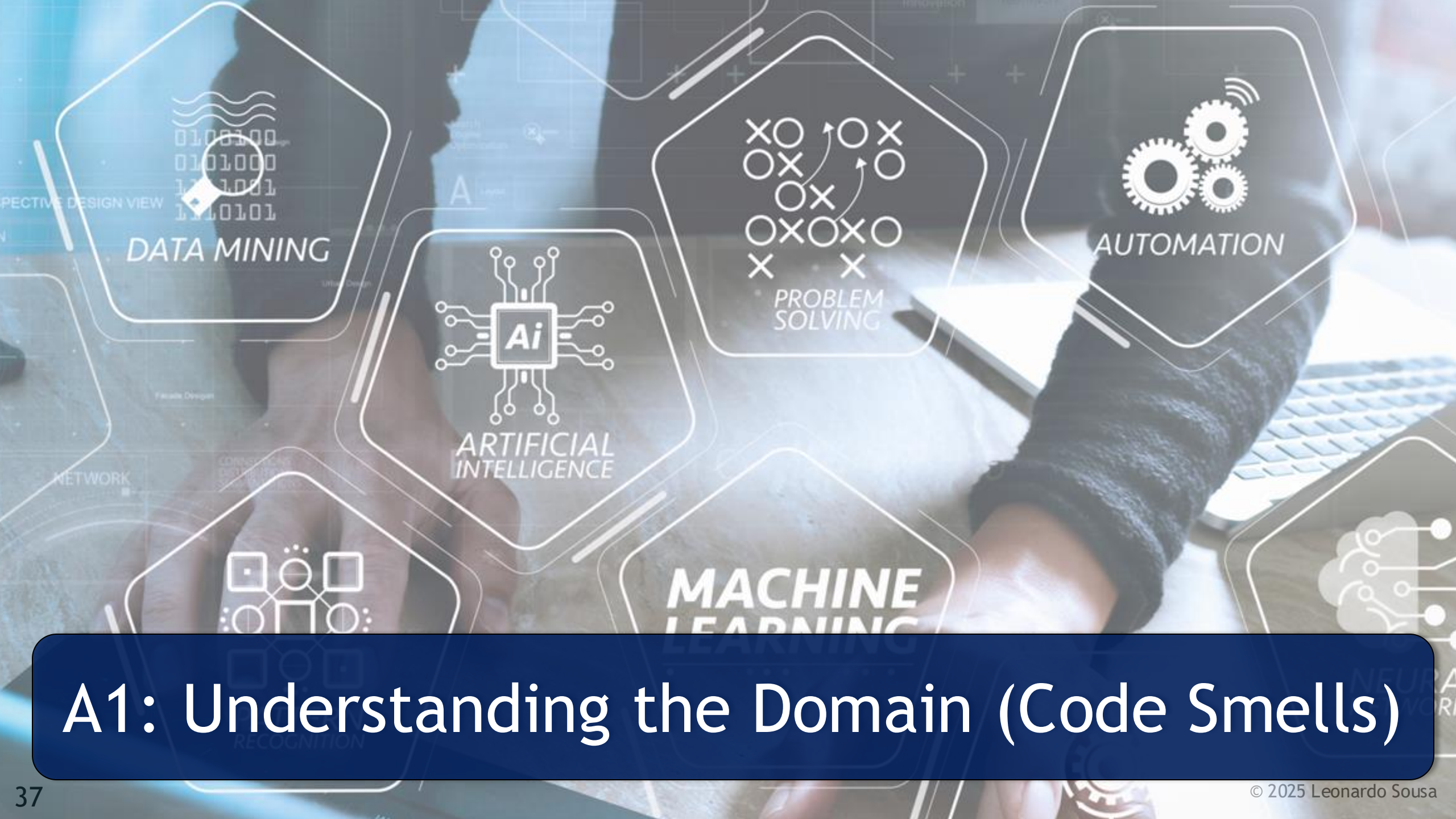
Bias Detection

Employing techniques to identify and quantify bias in data, enabling targeted efforts to mitigate its effects

Conclusion and Key Takeaways

Bias in machine learning is a complex issue that requires careful attention and proactive measures to address

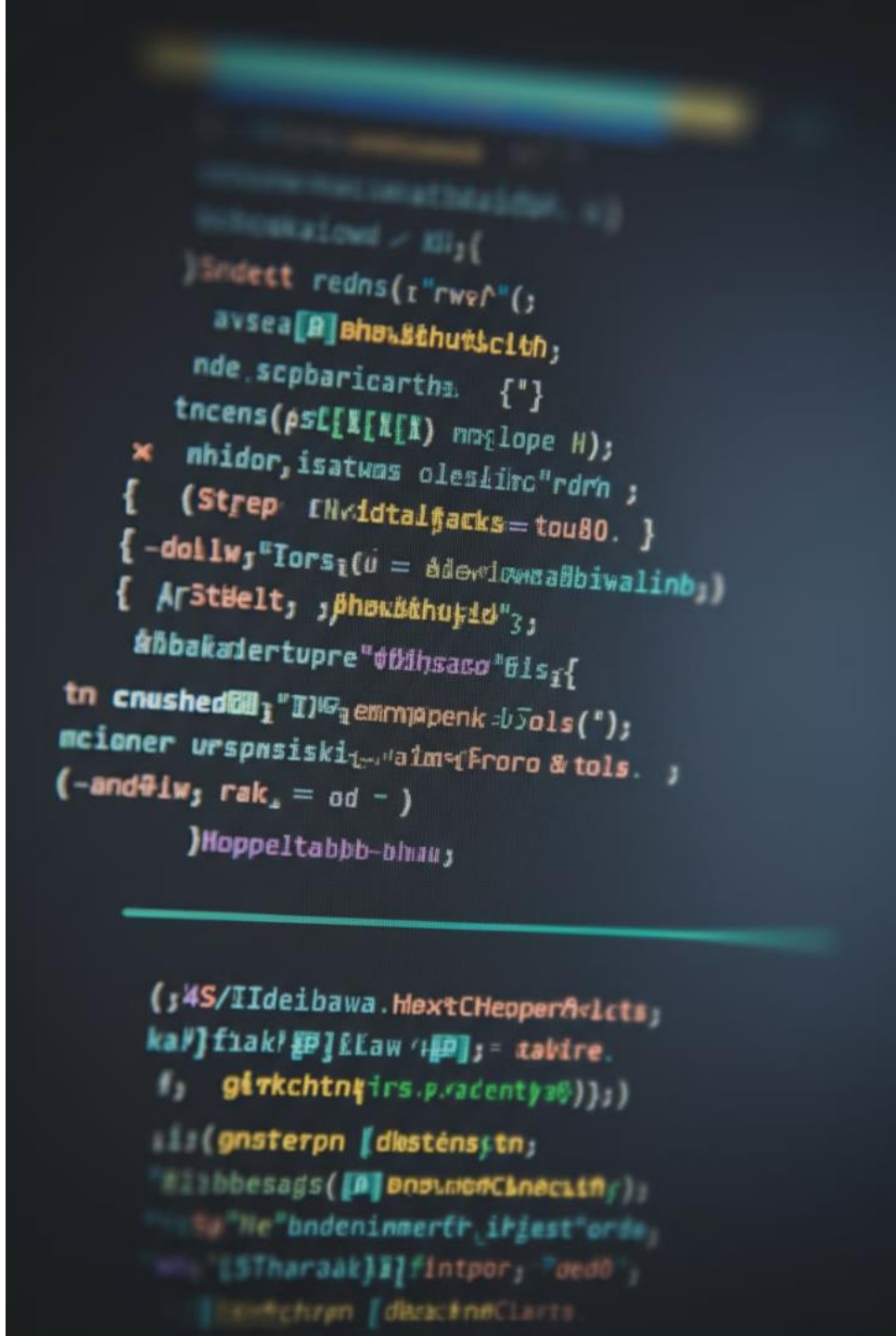
By understanding the causes, consequences, and mitigation strategies, we can strive to build fairer and more ethical AI systems that benefit everyone



A1: Understanding the Domain (Code Smells)

Introduction to Code Smells

- Code smells are indicators of potential problems in your code, often reflecting poor design or implementation choices
- They are not necessarily bugs, but they can increase complexity and hinder maintainability.



What is a God Class

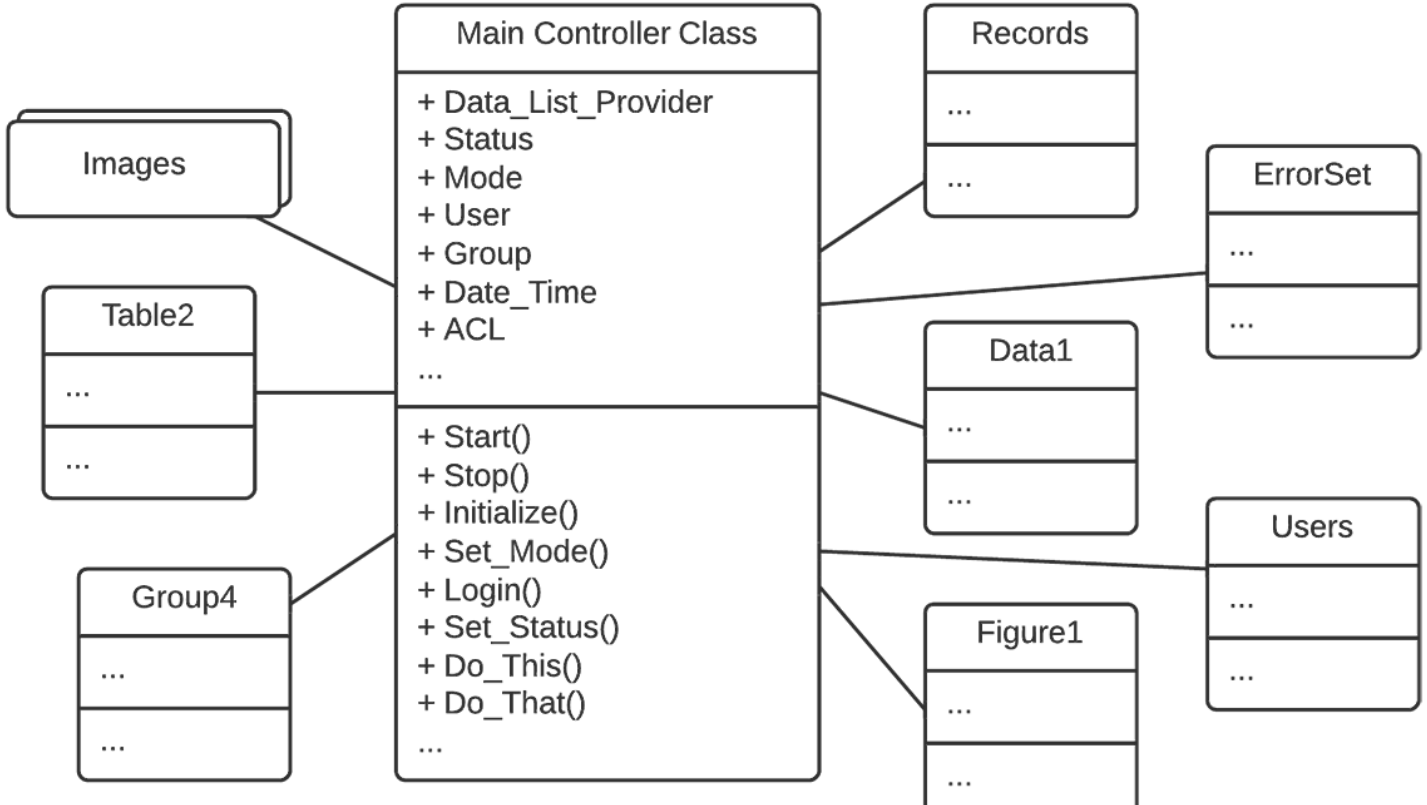
Overloaded Responsibility

A God Class takes on too many responsibilities

- It handles tasks that belong in other classes
- This violates the Single Responsibility Principle

Symptoms of a God Class

- Large number of methods
- Many instance variables
- Tight coupling with other classes



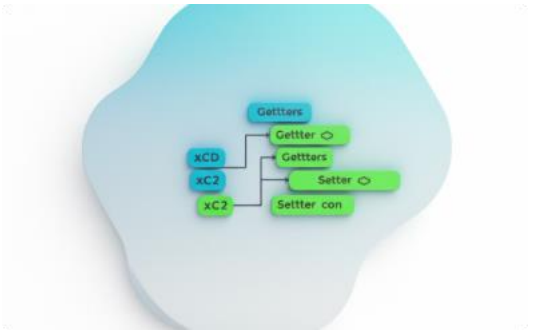
What is a Data Class?

Data Container

A data class primarily stores data. It may contain simple data structures like strings or numbers.

Limited Behavior

Data classes have limited behavior, typically just getters and setters for accessing data.



Encapsulation

They encapsulate data, providing controlled access through methods rather than directly exposing fields.

Potential for Complexity

As data classes grow, they can become complex and difficult to maintain.

What is a Long Method?

Code Readability

Long methods are difficult to read and understand, especially for large codebases

This can lead to confusion, errors, and increased maintenance time.

Code Maintainability

Long methods are harder to maintain and update, leading to potential bugs and regressions

Code should be modular and easy to change.

Code Complexity

Long methods can be complex and difficult to follow, making it challenging to debug and identify issues. Keeping methods concise improves clarity.

What is a Feature Envy

1 **Focus on Other Classes**

Methods in one class are more concerned with data or logic from another class.

2 **Violation of Encapsulation**

Breaks down the encapsulation of classes, leading to tight coupling.

3 **Excessive Access**

Excessive accessing or manipulating data from other classes.

4 **Code Complexity**

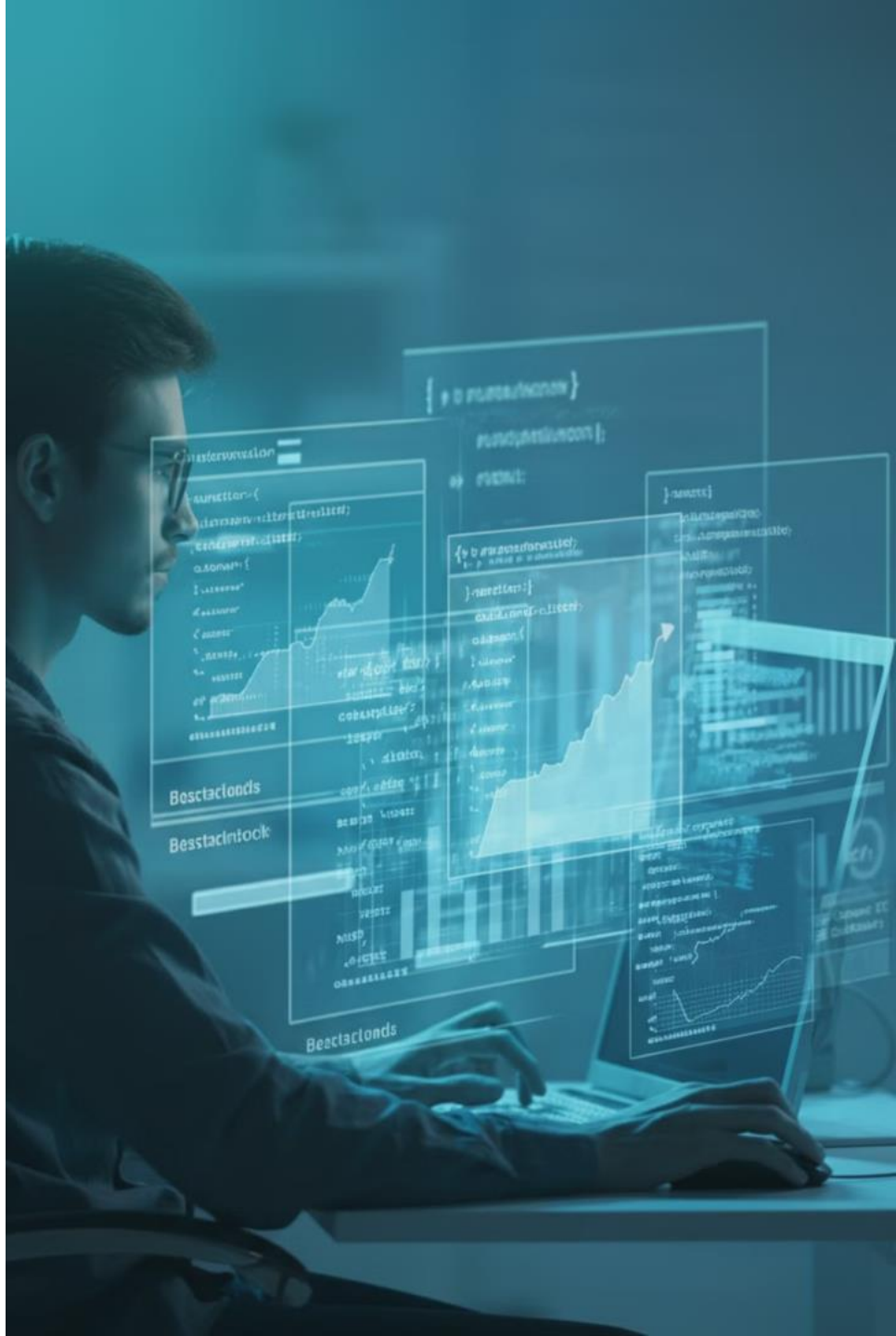
Increases code complexity and makes it difficult to maintain and understand.

```
public class Phone {
    private final String unformattedNumber;

    public Phone(String unformattedNumber) {
        this.unformattedNumber = unformattedNumber;
    }
    public String getAreaCode() {
        return unformattedNumber.substring(0,3);
    }
    public String getPrefix() {
        return unformattedNumber.substring(3,6);
    }
    public String getNumber() {
        return unformattedNumber.substring(6,10);
    }
}
public class Customer…
    private Phone mobilePhone;
    public String getMobilePhoneNumber() {
        return "(" +
            mobilePhone.getAreaCode() + ") " +
            mobilePhone.getPrefix() + "-" +
            mobilePhone.getNumber();
    }
```

Identifying Code Smells using Metrics

- Code metrics provide valuable insights into code quality and potential code smells.
  - Metrics like cyclomatic complexity, number of methods, and lines of code can indicate a potential God class
  - High method counts or methods too long can suggest a Long Method smell
- By analyzing these metrics, developers can identify areas of code requiring attention

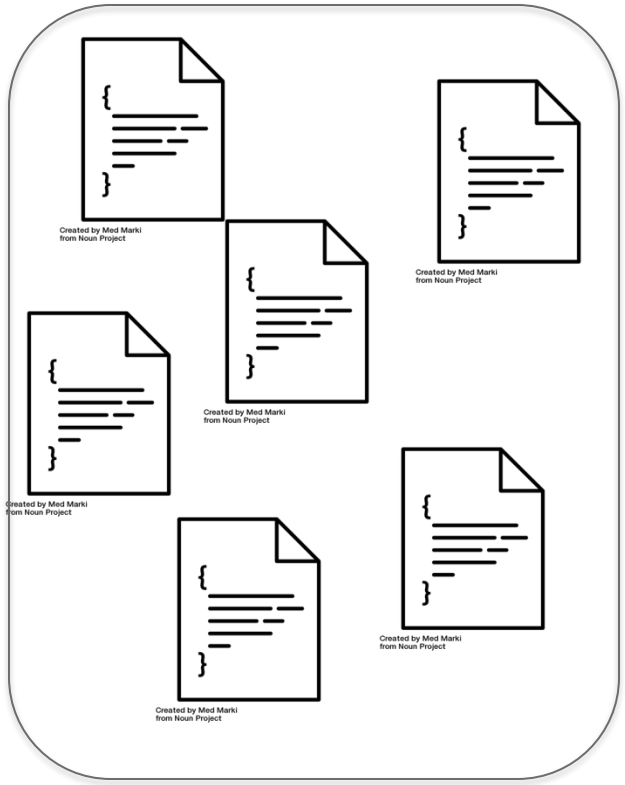


Metric-based Strategy

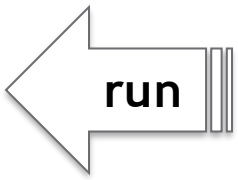
- Based on a set of detection rules that compare **metric** values with predefined **thresholds** according to logical **operators**
  - God Class:
    - CLOC > 500 && TCC < TCCAvg
  - Feature Envy:
    - METHODCALLS > INTERNAL CLASS CALLS

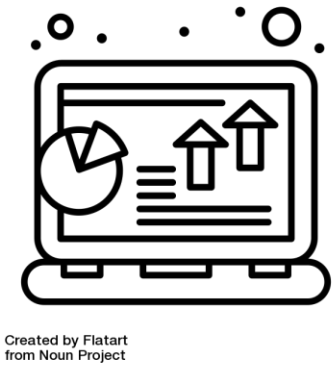
Code Smell Detection

- Step 1: collect metrics for the software system

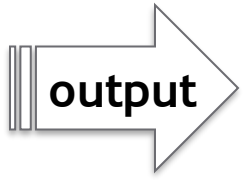


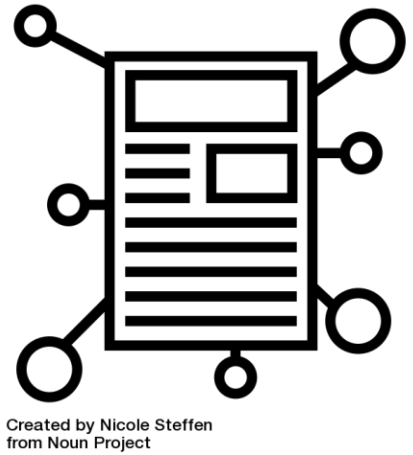
Source code





Static analysis tool
(e.g.: Understand[1])



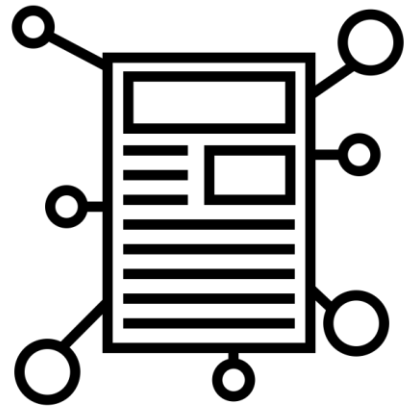


Set of Code Quality Metrics
(e.g.: LOC, TCC, FanIn, FanOut…)

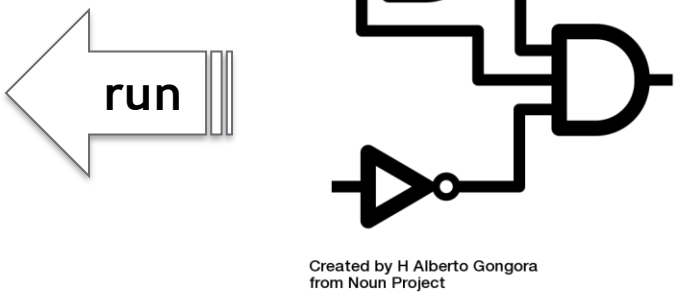
1. https://emenda.com/scitools-understand/

Code Smell Detection (cont'd)

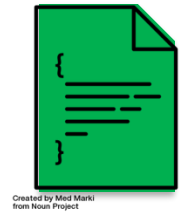
- Step 2: run the detection strategy (metric-based strategy)



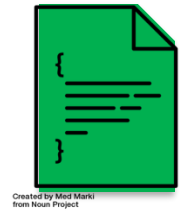
Set of Metrics



Metric-based strategy
(e.g.: God Class = CLOC > 500 && TCC < TCCAvg)

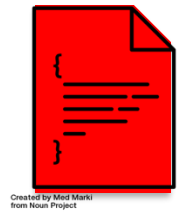


Non-smelly element



Non-smelly element

.
.
.



Smelly element

https://emenda.com/scitools-understand/

Limitations of Traditional Detection Strategies

- Code smells detected by existing approaches can be subjectively perceived and interpreted by developers
  - The agreement between the detectors is low
  - Different tools can lead to different results

- Detectors require the specification of thresholds to distinguish smelly and non-smelly instances
  - **Thresholds strongly influence the detectors' performance**

- Several other limitations

Machine Learning Detection Approach

- There are several ways that we can use ML to detect smells

- Simplest one: train a model with the code metrics
  - Collect the metric values for every class/method (LOC, TCC, FanIn…)
    - They are the features (independent variables)

  - Conduct a manual validation
    - Every instance in dataset: classify it as smelly or not
    - They are the labels/classes (smelly or non-smelly): dependent variables

Instance	**Features (Independent variables)**			**Label (dependent variable)**
	LOC	TCC	(other metrics)	Feature Envy
0	500	0.6	…	1
1	356	1.0	…	0

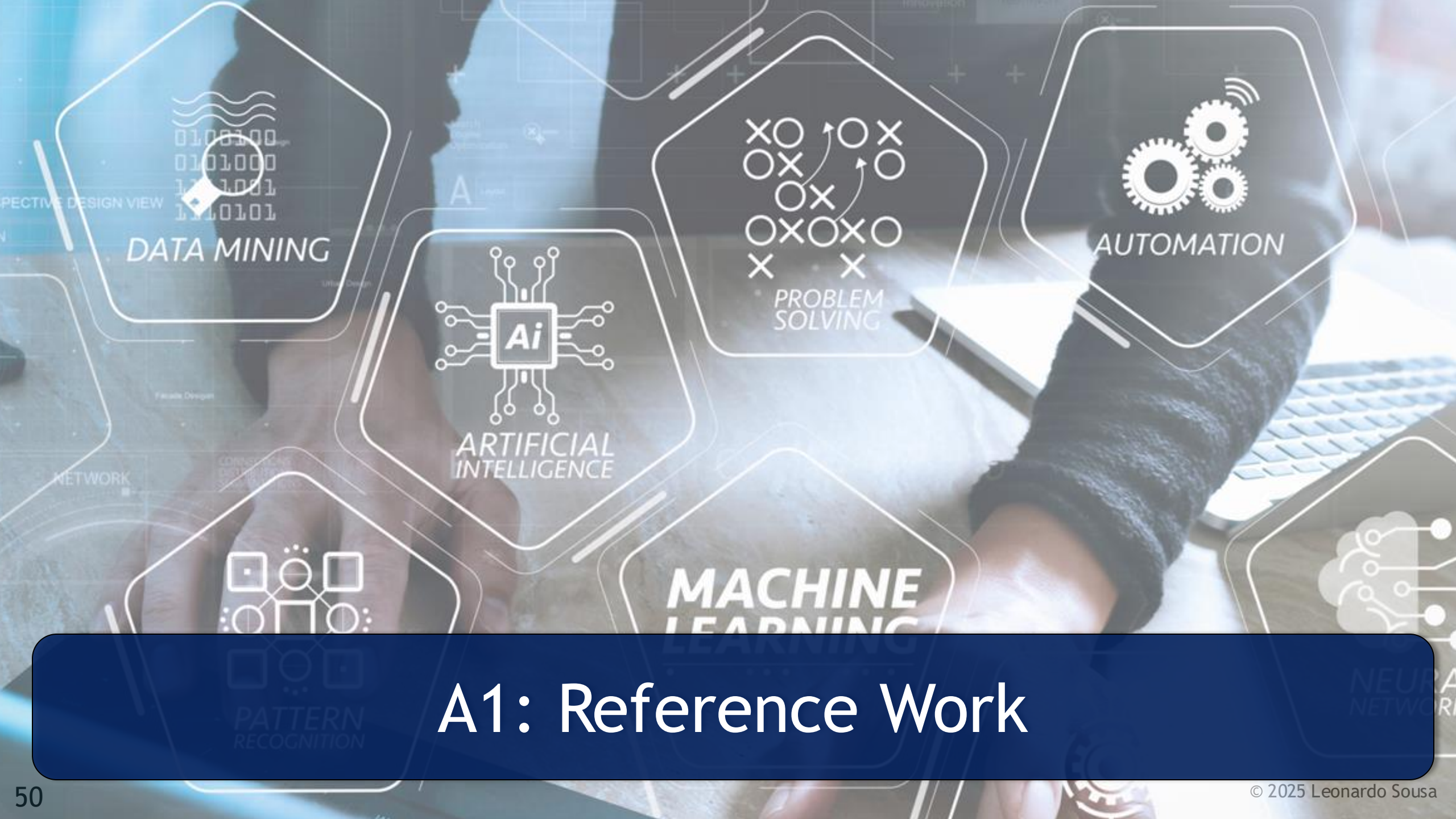
Evaluating the Code Smell Detection Model

We need to evaluate the model's performance on unseen code

This helps us understand how well it generalizes to new codebases and how reliable it is.

Metric	Description
Accuracy	The percentage of correctly classified code smells.
Precision	The percentage of correctly identified code smells out of all predictions.
Recall	The percentage of correctly identified code smells out of all actual code smells.
F1 Score	The harmonic mean of precision and recall.

By evaluating the model, we can gain insights into its limitations and areas for improvement



A1: Reference Work

ML techniques for Code Smell Detection

- Francesca Arcelli *et al.'s* study[1]
- Experiment 16 different machine-learning algorithms
- Dataset: 4 code smells
  - Data Class, Large Class
  - Feature Envy and Long Method
  - 74 software systems
  - Code metrics

Empir Software Eng (2016) 21:1143–1191
DOI 10.1007/s10664-015-9378-4

Comparing and experimenting machine learning techniques for code smell detection

Francesca Arcelli Fontana · Mika V. Mäntylä · Marco Zanoni · Alessandro Marino

Published online: 6 June 2015

Abstract Several code smell detection tools have been developed providing different results, because smells can be subjectively interpreted, and hence detected, in different ways. In this paper, we perform the largest experiment of applying machine learning algorithms to code smells to the best of our knowledge. We experiment 16 different machine-learning algorithms on four code smells (Data Class, Large Class, Feature Envy, Long Method) and 74 software systems, with 1986 manually validated code smell samples. We found that all algorithms achieved high performances in the cross-validation data set, yet the highest performances were obtained by J48 and Random Forest, while the worst performance were achieved by support vector machines. However, the lower prevalence of code smells, i.e., imbalanced data, in the entire data set caused varying performances that need to be addressed in the future studies. We conclude that the application of machine learning to the detection of these code smells can provide high accuracy (>96 %), and only a hundred training examples are needed to reach at least 95 % accuracy.

Keywords Code smells detection · Machine learning techniques · Benchmark for code smell detection

 1. Comparing and experimenting machine learning techniques for code smell detection. 2016. F. Arcelli Fontana, M. V. Mäntylä, M. Zanoni, and A. Marino

Selected Metrics

Table 4 Selected metrics

Size	Complexity	Cohesion	Coupling	Encapsulation	Inheritance
LOC	CYCLO	LCOM5	FANOUT	LAA	DIT
LOCNAMM*	WMC	TCC	ATFD	NOAM	NOI
NOM	WMCNAMM*		FDP	NOPA	NOC
NOPK	AMWNAMM*		RFC		NMO
NOCS	AMW		CBO		NIM
NOMNAMM*	MAXNESTING		CFNAMM*		NOII
NOA	WOC		CINT		
	CLNAMM		CDISP		
	NOP		MaMCL§		
	NOAV		MeMCL§		
	ATLD*		NMCS§		
	NOLV		CC		
			CM		

Instance	Features (Independent variables)			Label (dependent variable)
	LOC	TCC	(other metrics)	is_feature_envy
0	500	0.6	…	1
1	356	1.0	…	0

Building the Dataset

The authors used well-known smell detectors to collect the smells

- Detectors cannot usually achieve 100% recall
- False positives

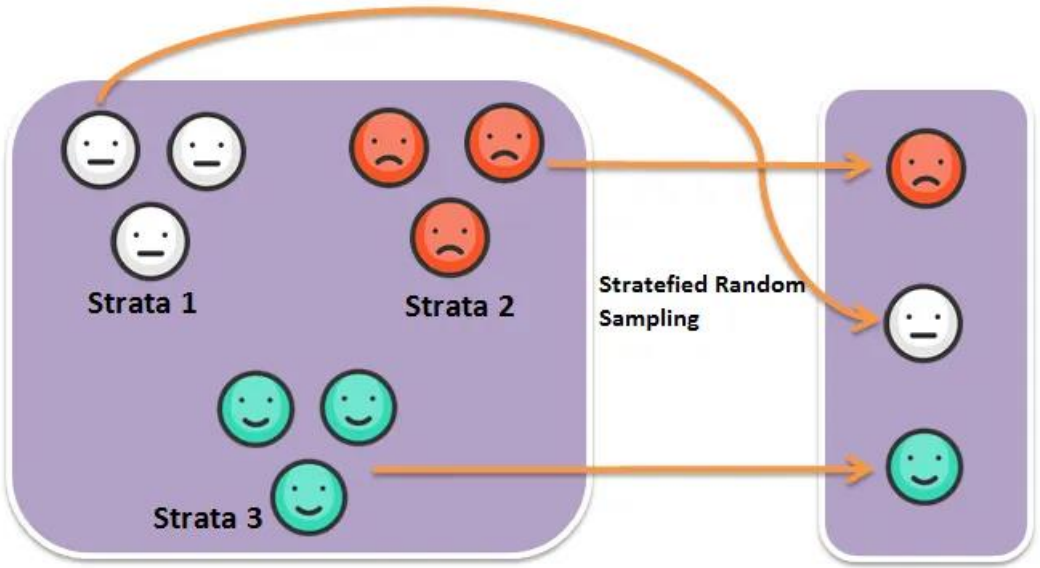
- Manual validation
  - 1,986 instances (826 smelly elements and 1,160 non-smelly ones)

- Stratified random sampling

- Training set was normalized in size
  - $\frac{1}{3}$ *positive instances* and $\frac{2}{3}$ *negative instances*

- **4 datasets: 140 positive instances and 280 negative instances**

Stratified random sampling



Training Phase

- 32 variants of different ML algorithms
  - 6 basic techniques
  - J48 → 3 types of pruning
  - 8 techniques were combined with AdaBoost
- Hyperparameter tuning
  - Grid-search algorithm
  - Cross validation

Arcelli et al's Results

- Most classifiers exceed 95% of both **Accuracy**
  - J48 and RANDOM FOREST obtaining the best performance

- Results
  - You can use any ML algorithm
  - Code smell detection can be solved almost perfectly through ML

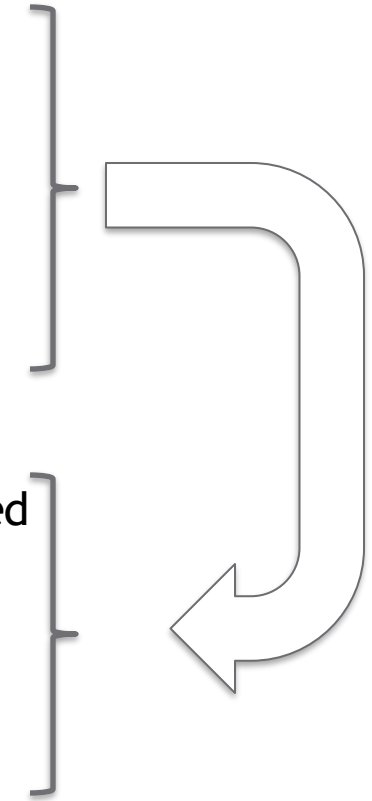
↓ Biased results

A1: Bias Identification

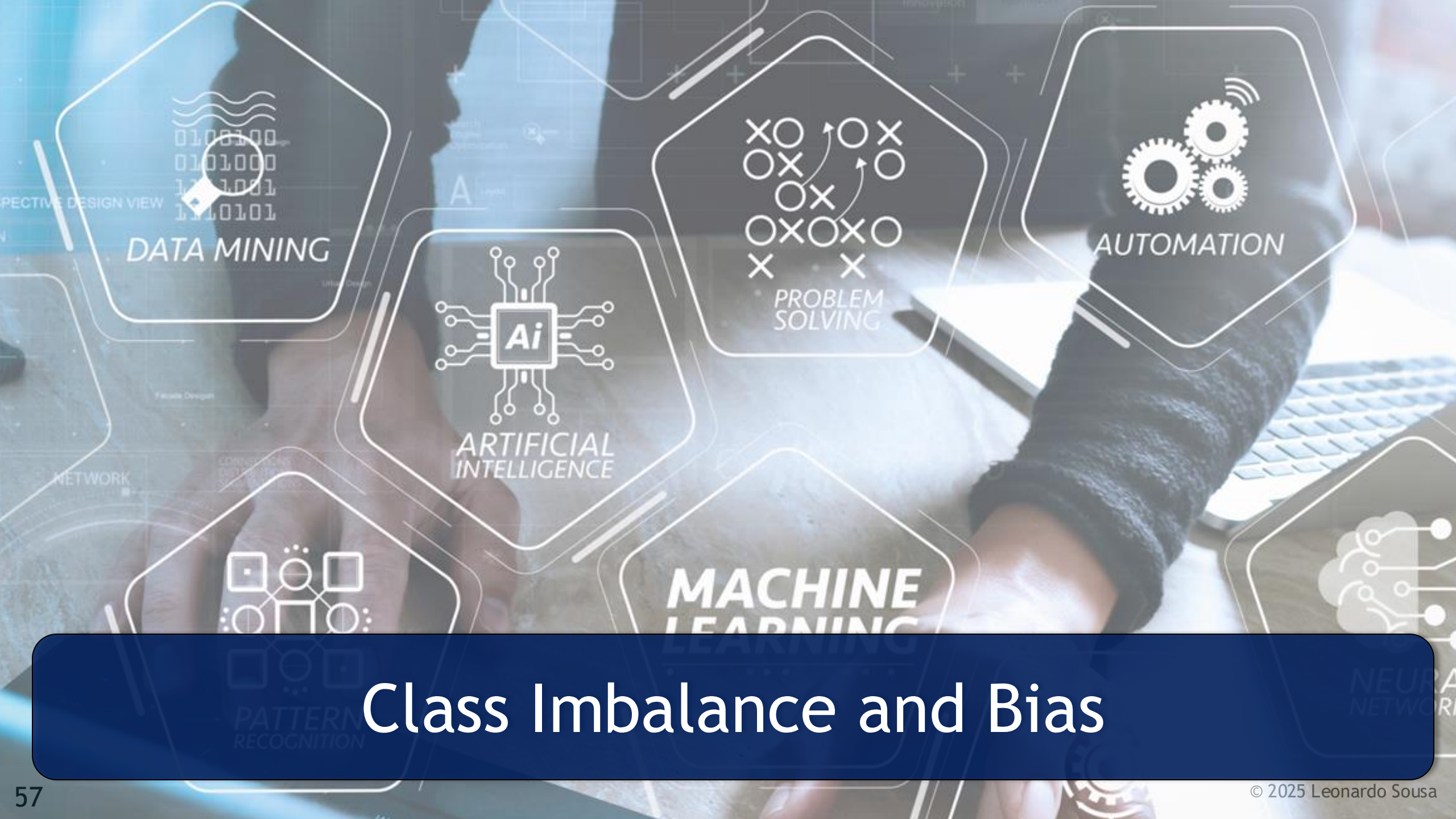
- **Task 1: Understanding the domain**
  - Read the reference paper[1] to better understand the domain and what the authors did in their work

- **Task 2: Experimenting with Machine Learning**
  - Create a notebook (Google Colab)
  - Experiment with Machine Learning models
    - 1 to 4 datasets
    - Train multiple models (1 to N models)
    - Use any ML technique

- **Task 3: Explaining the Bias**
  - Write a PDF report (<3pages) explaining why/how the authors results are biased
  - Report should have 3 sections
    - Explain the bias from different perspectives
      - DATASET
      - TRAINING PROCESS



1. Comparing and experimenting machine learning techniques for code smell detection. 2016. F. Arcelli Fontana, M. V. Mäntylä, M. Zanoni, and A. Marino

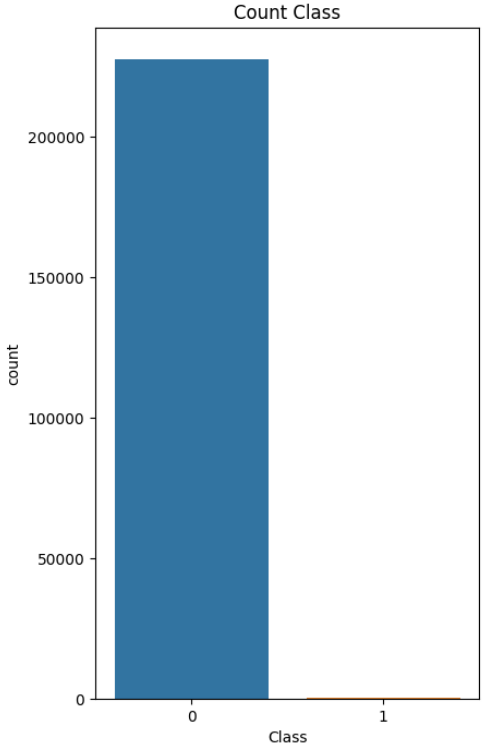


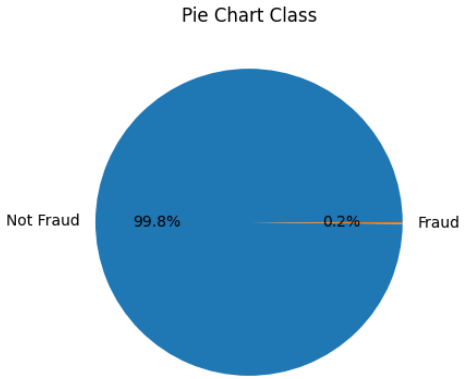
Class Imbalance and Bias

Class Imbalance and Bias

- While class imbalance can lead to unfair or misleading model performance, **it is not inherently a form of bias.**

- Instead, it is a **data characteristic** that can **cause biased outcomes** in machine learning models if not handled properly

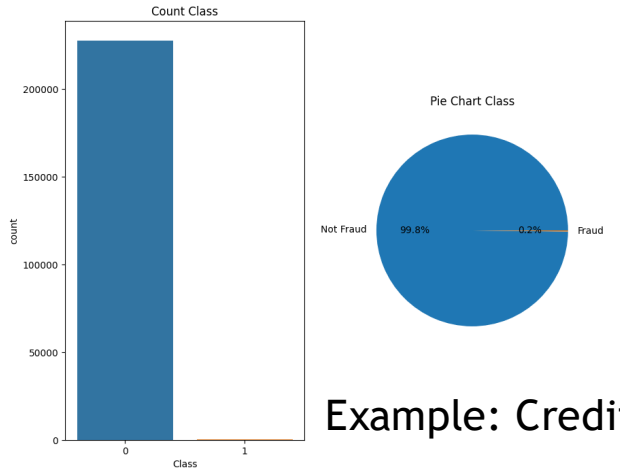




Example: Credit Card Fraud Detection Data

Class Imbalance is a Data Distribution Issue, Not a Systematic Bias

- Class imbalance means that one class appears far more frequently than others in the dataset
- However, imbalance itself is not biased unless it reflects a systematic unfairness (e.g., underrepresentation of certain groups due to historical discrimination)
- Example: In fraud detection, only ~1% of transactions may be fraudulent.
  - This is an inherent characteristic of the real world, not necessarily a bias.
  - A model trained on this data may struggle to detect fraud, but this is due to data imbalance, not bias.



Example: Credit Card Fraud Detection Data

Bias Implies Systematic Favoritism or Discrimination

- **Bias refers to an unfair, systematic error** that favors certain groups or outcomes
- Class imbalance **can contribute to bias** but is not always biased itself
- **Example:** A medical AI trained mostly on data from one demographic group (e.g., mostly white patients) may perform worse for underrepresented groups (e.g., Black patients)
  - The issue is not just imbalance; it is **demographic bias** because it leads to **systematic disparities in model performance** across groups

A Balanced Dataset Can Still Be Biased

- Even if classes are perfectly balanced, a dataset can still contain **hidden biases** in feature selection, labeling, or sampling

- **Example:** A hiring model trained on a **balanced** dataset of men and women **may still be biased** if historical hiring decisions favored men
  - **The bias comes from historical discrimination, not class imbalance**

Class Imbalance Can Be Handled Without Addressing Bias

- Techniques like **oversampling (SMOTE), undersampling, class weighting, or cost-sensitive learning** can handle class imbalance
- However, these techniques **do not correct biases** present in the dataset
- **Example:** If a facial recognition dataset has 50% light-skinned and 50% dark-skinned faces, but the images of dark-skinned faces are lower quality, the model **will still be biased**, even though the dataset is balanced

Summary: When Class Imbalance Becomes Bias

Situation	Is It Bias?	Why?
Fraud detection dataset has 99% "No Fraud" and 1% "Fraud"	❌ No	This is an inherent characteristic of fraud, not bias.
Medical AI trained mostly on data from young adults performs poorly for elderly patients	✅ Yes	The underrepresentation of elderly patients leads to **systematic errors**, making it **bias**.
A hiring AI trained on a dataset with 50% men and 50% women, but the past decisions favored men	✅ Yes	Even though the dataset is balanced, the **labeling process carries historical bias.**
A facial recognition dataset has 50% light-skinned and 50% dark-skinned images, but	✅ Yes	The data collection process introduces bias, even though the class distribution

Assignment 1

- An imbalanced dataset is not inherently biased unless the imbalance misrepresents the real-world distribution of code smells.
- Most of the classes do not have code smells!
- If code smells are naturally rare in real-world software, then an imbalanced dataset (e.g., 90% clean code, 10% smelly code) is **a faithful representation** rather than a biased dataset
- **Therefore: the class imbalance characteristic of the the datasets used in the first assignment cannot be used to explain the bias in Section 1 or 2**

You will be deducted 1 point if you say that the dataset is biased due to class imbalance