

INTRODUCTION TO MACHINE LEARNING

Systems Sciences Foundations

Author: Eng. Carlos Andrés Sierra, M.Sc.
cavirguezs@udistrital.edu.co

Lecturer
Department of Computer Engineering
School of Engineering
Universidad Distrital Francisco José de Caldas

2025-I



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Outline

- 1 Fundamentals of Machine Learning
- 2 Python Tools for Machine Learning
- 3 Supervised Machine Learning
- 4 Machine Learning Models Evaluation



Outline

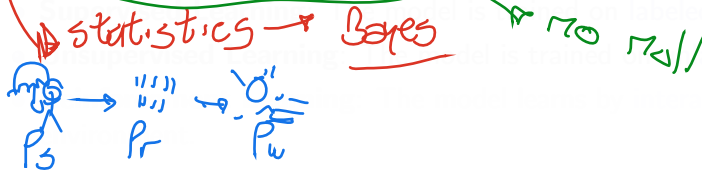
- 1 Fundamentals of Machine Learning
- 2 Python Tools for Machine Learning
- 3 Supervised Machine Learning
- 4 Machine Learning Models Evaluation



Key Concepts in Machine Learning

Machine Learning

- **Machine learning** is a method of ~~data~~ analysis that automates analytical model building.
- It is a **branch** of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.



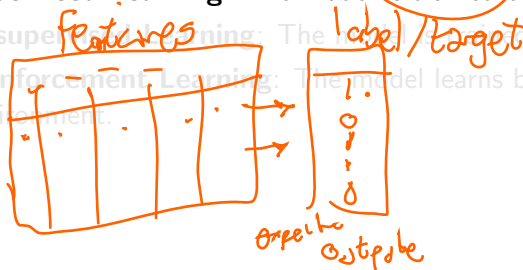
Key Concepts in Machine Learning

Machine Learning

- **Machine learning** is a method of data analysis that **automates** analytical model building.
- It is a **branch** of **artificial intelligence** based on the idea that systems can **learn from data**, **identify patterns** and **make decisions** with minimal human intervention.

- ① • **Supervised Learning:** The model is trained on labeled data.

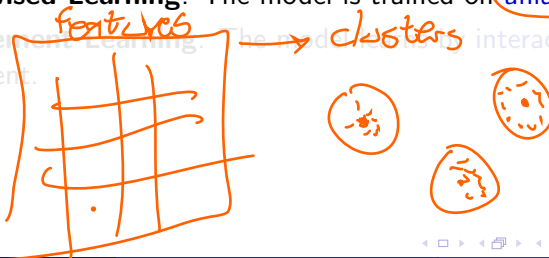
- **Unsupervised Learning:** The model is trained on unlabeled data.
- **Reinforcement Learning:** The model learns by interacting with an environment.



Key Concepts in Machine Learning

Machine Learning

- **Machine learning** is a method of data analysis that **automates** analytical model building.
- It is a **branch** of **artificial intelligence** based on the idea that systems can **learn from data**, **identify patterns** and **make decisions** with minimal human intervention.
- **Supervised Learning**: The model is trained on **labeled data**.
- **Unsupervised Learning**: The model is trained on **unlabeled data**.
- **Reinforcement Learning**: The model learns by interacting with an environment.



Key Concepts in Machine Learning

Machine Learning

- **Machine learning** is a method of data analysis that **automates** analytical model building.
- It is a **branch** of **artificial intelligence** based on the idea that systems can **learn from data**, **identify patterns** and **make decisions** with minimal human intervention.

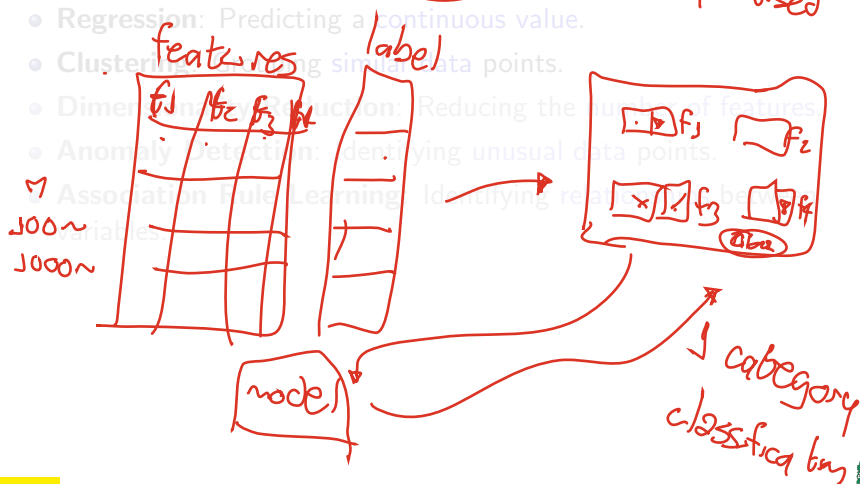
- **Supervised Learning**: The model is trained on **labeled data**.
- **Unsupervised Learning**: The model is trained on **unlabeled data**.
- **Reinforcement Learning**: The model learns by **interacting** with an environment.

1 step
mult. - step → cybernetics



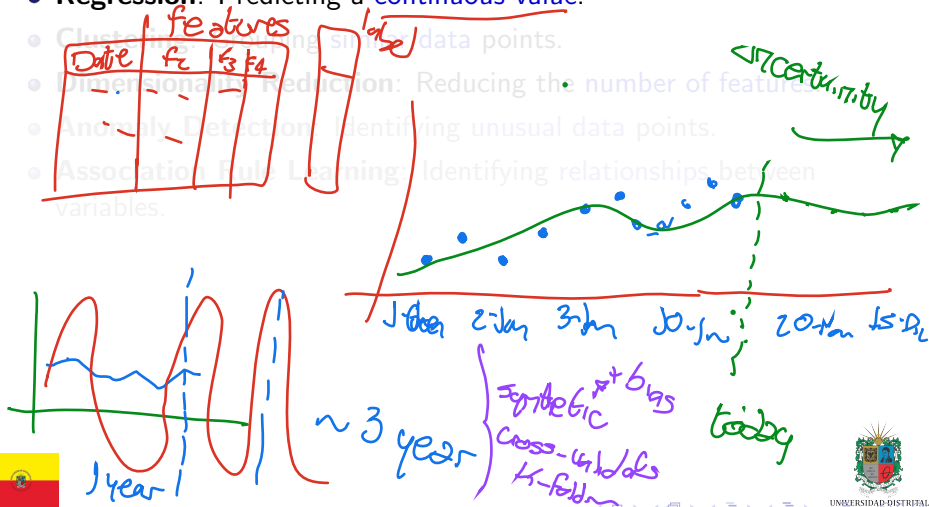
Typical Machine Learning Problems

- **Classification:** Predicting a label. *targets → supervised*



Typical Machine Learning Problems

- **Classification:** Predicting a **label**.
- **Regression:** Predicting a **continuous value**.



- 3 supervised
unsupervised



- f_1, f_2, f_3

number of features.

resources

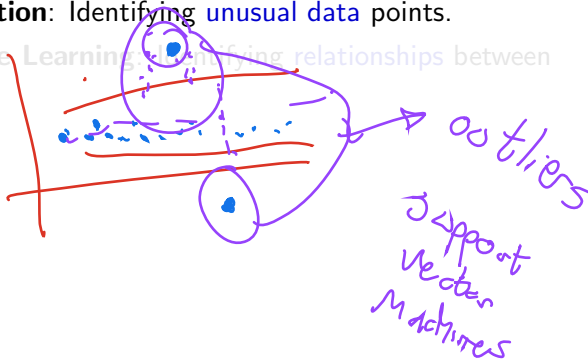
Correlation matrix

Ex: $\begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$



Typical Machine Learning Problems

- **Classification:** Predicting a **label**.
- **Regression:** Predicting a **continuous value**.
- **Clustering:** Grouping **similar data** points.
- **Dimensionality Reduction:** Reducing the **number of features**.
- **Anomaly Detection:** Identifying **unusual data** points.
- **Association Rule Learning:** Identifying relationships between variables.



Typical Machine Learning Problems

- **Classification:** Predicting a **label**.
- **Regression:** Predicting a **continuous value**.
- **Clustering:** Grouping **similar data** points.
- **Dimensionality Reduction:** Reducing the **number of features**.
- **Anomaly Detection:** Identifying **unusual data** points.
- **Association Rule Learning:** Identifying **relationships** between variables.

Language
 ↳ words
 sequence

correlation

f_1
 f_2 } $\rightarrow f_3$



The Machine Learning Workflow

- **Data Collection:** Gathering the data. ①

- **Data Preprocessing:** Cleaning and preparing the data.

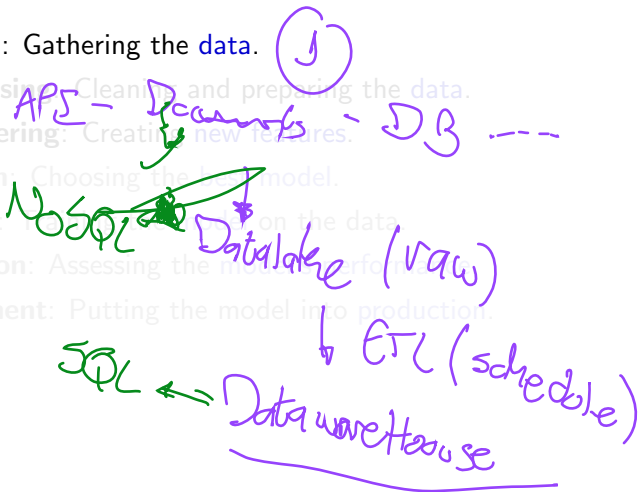
- **Feature Engineering:** Creating new features.

- **Model Selection:** Choosing the best model.

- **Model Training:** Training the model on the data.

- **Model Evaluation:** Assessing the model's performance.

- **Model Deployment:** Putting the model into production.

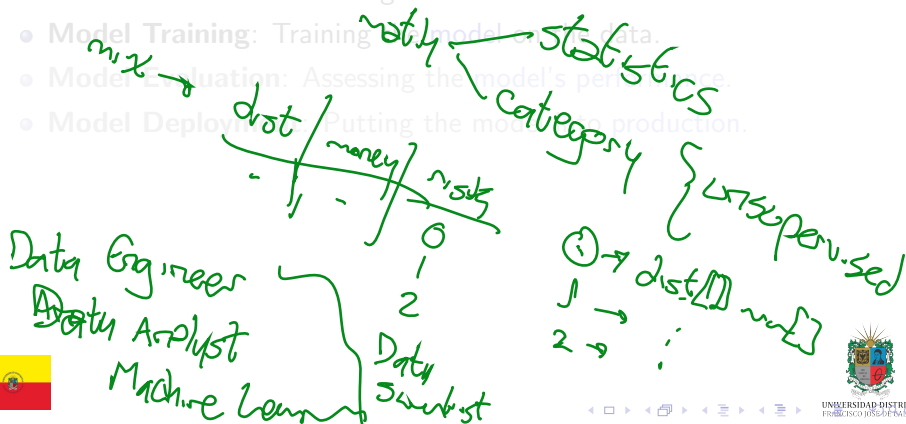


- Model Deployment: Putting the model into

A handwritten diagram illustrating the Feature Selection process. At the top, the text "Comparing the data." is written, with "data." circled in green. An arrow points from this circle down to a bracketed list of "features". From the "features" bracket, an arrow points down to another bracketed list containing "new features" and "transform". To the right of this list, the word "quality" is written. An arrow points from the "transform" bracket down to the word "selection", which is underlined. To the right of "selection", the phrase "Format?" is written. At the top right, the word "High" is written and underlined. A curved arrow points from "High" down to "quality".



- Model Deployment: Putting the model into production



The Machine Learning Workflow

- **Data Collection:** Gathering the data.
- **Data Preprocessing:** Cleaning and preparing the data.
- **Feature Engineering:** Creating new features.
- **Model Selection:** Choosing the best model.

any ~ 50
108
202

- **Model Training:** Training the model on the data.
- **Model Evaluation:** Assessing the model's performance.
- **Model Deployment:** Putting the model into production.

Depends on data → Trees

cluster → unsupervised

AutoML



experience



The Machine Learning Workflow

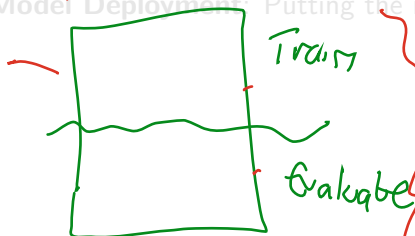
- **Data Collection:** Gathering the data.
- **Data Preprocessing:** Cleaning and preparing the data.
- **Feature Engineering:** Creating new features.
- **Model Selection:** Choosing the best model.
- **Model Training:** Training the model on the data.

one-hot
encoding



3x3
2x3
1x3
1x3
1x3

1m
2m



3x3=9
3x4=12

12x3=36
4x3=12

Stewart

10 6 rows → groups
↓
synthetic
numeric

6 - train
2 - evaluate

1.5% / 1.5%



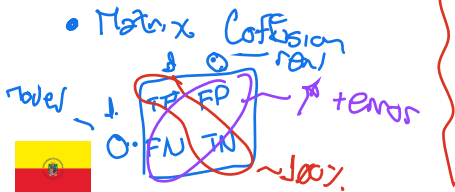
The Machine Learning Workflow

- **Data Collection:** Gathering the **data**.
- **Data Preprocessing:** Cleaning and preparing the **data**.
- **Feature Engineering:** Creating **new features**.
- **Model Selection:** Choosing the **best model**.
- **Model Training:** Training the **model** on the data.
- **Model Evaluation:** Assessing the **model's performance**

- **Model Deployment:** Putting the model into production.

Metrics \Rightarrow Error

Classification



Regression



The Machine Learning Workflow

- No SW* → *SW* → *Model* → *Deployment*
- **Data Collection:** Gathering the **data**.
 - **Data Preprocessing:** Cleaning and preparing the **data**.
 - **Feature Engineering:** Creating **new features**.
 - **Model Selection:** Choosing the **best model**.
 - **Model Training:** Training the **model** on the data.
 - **Model Evaluation:** Assessing the **model's** performance.
 - **Model Deployment:** Putting the model into **production**.



a = ""
b = ""
c = ""

Notebooks

Docker

Cloud (serverless)

classes
 ↓
 oop
 ↓
 API Rest

direct Pickle

Parameters
 ↓
 Features



Examining the Data

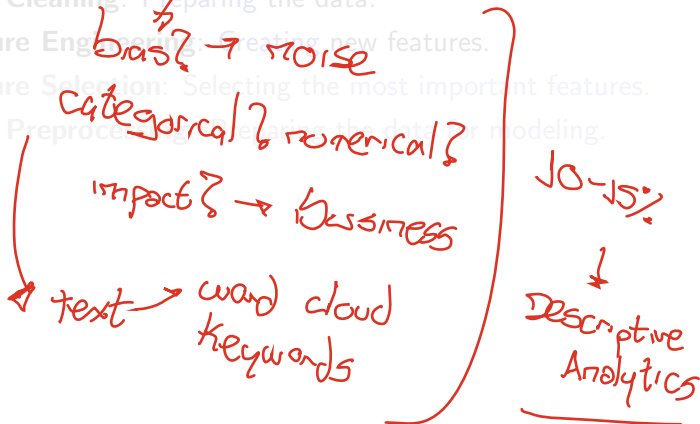
- **Data Exploration:** Understanding the data.

- Data Cleaning: Preparing the data.

- Feature Engineering: Creating new features.

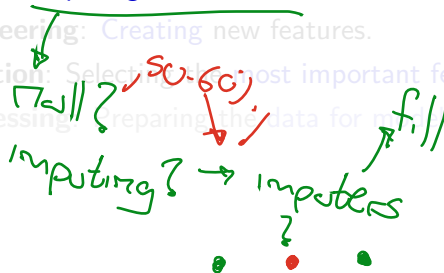
- Feature Selection: Selecting the most important features.

- Data Preprocessing: Preparing the data for modeling.



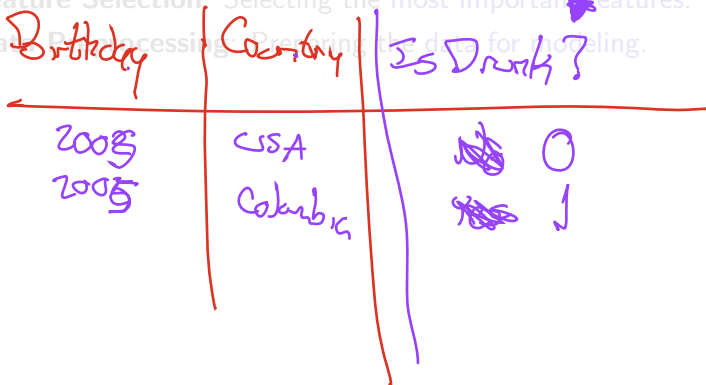
Examining the Data

- **Data Exploration:** Understanding the data.
- **Data Cleaning:** Preparing the data.
- **Feature Engineering:** Creating new features.
- **Feature Selection:** Selecting the most important features.
- **Data Preprocessing:** Preparing the data for modeling.



Examining the Data

- **Data Exploration:** Understanding the data.
- **Data Cleaning:** Preparing the data.
- **Feature Engineering:** Creating new features.
- **Feature Selection:** Selecting the most important features.
- **Data Preprocessing:** Preparing the data for modeling.



A hand-drawn table with three columns and two rows of data. The columns are labeled 'Birthday', 'Country', and 'Is Drunk?'. The first row contains the values '2003', 'USA', and '0'. The second row contains the values '2005', 'Colombia', and '1'. The 'Is Drunk?' column has some scribbled-out text before the final values. A purple arrow points from the 'Creating' text in the Feature Engineering bullet point to the 'Is Drunk?' column header.

Birthday	Country	Is Drunk?
2003	USA	Is Drunk? 0
2005	Colombia	Is Drunk? 1



Examining the Data

- **Data Exploration:** Understanding the data.
- **Data Cleaning:** Preparing the data.
- **Feature Engineering:** Creating new features.
- **Feature Selection:** Selecting the most important features.
- **Data Preprocessing:** Preparing the data for modeling.

features

A	B	C	D	E

Exp. 1. ABC

Exp. 2. ABCDE

Exp. 3. ACE

Exp. 4. BD



Examining the Data

- **Data Exploration:** Understanding the data.
- **Data Cleaning:** Preparing the data.
- **Feature Engineering:** Creating new features.
- **Feature Selection:** Selecting the most important features.
- **Data Preprocessing:** Preparing the data for modeling.

Categorical
Text } → Numeric → Vector

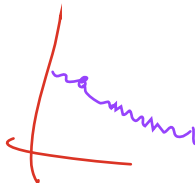


Algorithmic Bias

- **Algorithmic bias** is a **systematic error** in a model that results in **unfair outcomes**.
- It can be caused by **biased training data**, biased algorithms, or biased **decision-making**.

\$ Families →

Loj	car	codes	inc.
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-



Outline

- 1 Fundamentals of Machine Learning
- 2 Python Tools for Machine Learning**
- 3 Supervised Machine Learning
- 4 Machine Learning Models Evaluation



Python Tools for Machine Learning

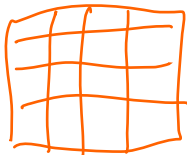
Python Tools

- NumPy: A library for numerical computing. → ~~Trans~~ ~~Ally~~ ~~dent~~
- Pandas: A library for data manipulation and analysis.
- Matplotlib: A library for data visualization.
- Scikit-learn: A library for machine learning.

Vectorization → no loops

linear algebra

Tabular Data



Dataframe (df)

Analysis

SQL (ORM)



C
↓
arrays
matrix (GPU)
operations



Jupyter Notebooks

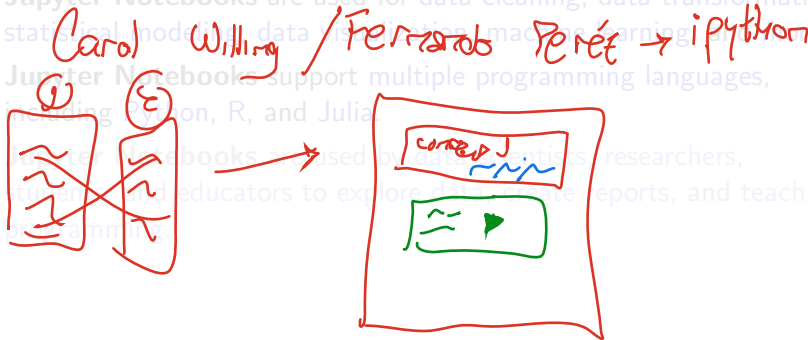
CC LAB

- **Jupyter Notebooks** are a web-based interactive computing environment that allows you to create and share documents that contain live code, equations, visualizations, and narrative text.

- Jupyter Notebooks are used for data cleaning, data transformation, statistical modeling, data visualization, machine learning, etc.

- Jupyter Notebooks support multiple programming languages, including Python, R, and Julia.

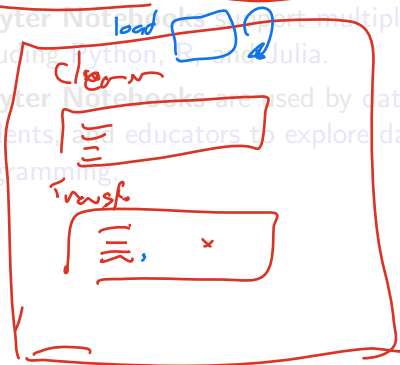
- Jupyter Notebooks are used by data scientists, researchers, students, and educators to explore data, create reports, and teach programming.



Jupyter Notebooks

- **Jupyter Notebooks** are a **web-based interactive computing environment** that allows you to create and share documents that contain **live code**, **equations**, **visualizations**, and **narrative text**.
- **Jupyter Notebooks** are used for data cleaning, data transformation, statistical modeling, data visualization, machine learning, and more.

- Jupyter Notebooks support multiple programming languages, including Python, R, and Julia.
- Jupyter Notebooks are used by data scientists, researchers, students, and educators to explore data, create reports, and teach programming.



Jupyter Notebooks

- **Jupyter Notebooks** are a **web-based interactive computing environment** that allows you to create and share documents that contain **live code**, **equations**, **visualizations**, and **narrative text**.
- **Jupyter Notebooks** are used for **data cleaning**, **data transformation**, **statistical modeling**, **data visualization**, **machine learning**, and **more**.
- **Jupyter Notebooks** support **multiple programming languages**, including **Python**, **R**, and **Julia**.
- **Jupyter Notebooks** are used by **data scientists**, **researchers**, **students**, and **educators** to **explore data**, **create reports**, and **teach programming**.

Jupyter

Regular — Data Science



Jupyter Notebooks

- **Jupyter Notebooks** are a **web-based interactive computing environment** that allows you to create and share documents that contain **live code**, **equations**, **visualizations**, and **narrative text**.
- **Jupyter Notebooks** are used for **data cleaning**, **data transformation**, **statistical modeling**, **data visualization**, **machine learning**, and **more**.
- **Jupyter Notebooks** support **multiple programming languages**, including **Python**, **R**, and **Julia**.
- **Jupyter Notebooks** are used by **data scientists**, **researchers**, **students**, and **educators** to **explore data**, **create reports**, and **teach programming**.

Machine Learning Introduction



Lambda Functions

$$f \ x: x+2$$

Definition

A **lambda function** is a **small** **anonymous** function. A **lambda function** can take any number of **arguments**, but can only have one expression.

lambda arguments: expression

$x = \text{lambda } a: a+2$

$\text{print}(x(10)) \rightarrow 20$

↳ lambda (arguments) {
 return expression;



$df['adult'] = \text{lambda}$
 $df['age']: \text{if } x \geq 18 \text{ else } 0$



Numerical Python Library — Numpy

- **Numpy** is the core library for scientific computing in Python. It is the fundamental package for scientific computing with Python.
- **Numpy** is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. *vector*
- **Numpy** was created by Travis Oliphant in 2005, and it is an open-source project. Coming soon, Numpy version 2.0 will be released. *2.1*



Lineal Algebra with Numpy

- **Numpy** provides a comprehensive set of linear algebra functions.
- **Numpy** provides the functionality to create and manipulate matrices.
- **Numpy** provides the functionality to solve linear systems of equations.
- **Numpy** provides the functionality to calculate the determinant of a matrix.
- **Numpy** provides the functionality to calculate the inverse of a matrix.

$$\begin{bmatrix} - & - & - \\ - & - & - \\ - & - & - \end{bmatrix}$$

Eigen
Solve Equations

$$\begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$



Lineal Algebra with Numpy

- **Numpy** provides a **comprehensive set** of **linear algebra** functions.
- **Numpy** provides the **functionality** to **create** and **manipulate matrices**.
- **Numpy** provides the functionality to solve linear systems of equations.
- **Numpy** provides the functionality to calculate the determinant of a matrix.
- **Numpy** provides the functionality to calculate the inverse of a matrix.



Lineal Algebra with Numpy

- **Numpy** provides a **comprehensive set** of **linear algebra** functions.
- **Numpy** provides the **functionality** to **create** and **manipulate matrices**.
- **Numpy** provides the **functionality** to **solve linear systems** of equations.
- **Numpy** provides the **functionality** to **calculate the determinant** of a matrix.
- **Numpy** provides the **functionality** to **calculate the inverse** of a matrix.



Lineal Algebra with Numpy

- **Numpy** provides a **comprehensive set** of **linear algebra** functions.
- **Numpy** provides the **functionality** to **create** and **manipulate matrices**.
- **Numpy** provides the **functionality** to **solve linear systems** of equations.
- **Numpy** provides the **functionality** to **calculate** the **determinant** of a matrix.
- **Numpy** provides the **functionality** to **calculate** the **inverse** of a matrix.



Lineal Algebra with Numpy

- **Numpy** provides a **comprehensive set** of **linear algebra** functions.
- **Numpy** provides the **functionality** to **create** and **manipulate matrices**.
- **Numpy** provides the **functionality** to **solve linear systems** of equations.
- **Numpy** provides the **functionality** to **calculate** the **determinant** of a matrix.
- **Numpy** provides the **functionality** to **calculate** the **inverse** of a matrix.



Vectorization with Numpy

- **Vectorization** is the **process** of **converting** an **algorithm** from **operating** on a single value at a time to **operating** on a set of values at one time.
- **Vectorization** is the **process** of **replacing** explicit **loops** with **array expressions** or **matrix operations**.

- The advantages of vectorization are speed and clarity. The disadvantages are memory and complexity.
- Numpy provides the functionality to vectorize operations on arrays.



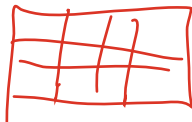
Vectorization with Numpy

- **Vectorization** is the process of converting an algorithm from operating on a single value at a time to operating on a set of values at one time.
- **Vectorization** is the process of replacing explicit loops with array expressions or matrix operations.
- The advantages of vectorization are speed and clarity. The disadvantages are memory and complexity.
- **Numpy** provides the functionality to vectorize operations on arrays.



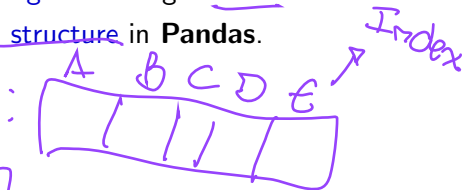
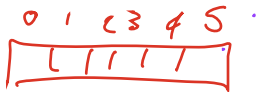
Introduction to Pandas

- **Pandas** is a fast, powerful, flexible, and easy-to-use open-source data manipulation and data analysis library built on top of the Python programming language.
- **Pandas** is a high-level data manipulation tool developed by Wes McKinney in 2008. → *Polars (Rust)*
- **Pandas** is a fast and efficient data manipulation tool that is built on top of NumPy. → *numerical*
- **Pandas** is one of the most popular and widely-used data manipulation libraries in the world.



The “Series” Data Structure

- A **Series** is a **one-dimensional array-like object** that contains a **sequence of values** and an **associated array of data labels**, called the **index**.
- The **index** of a **Series** is an **array of labels** that **correspond** to the **values** in the **Series**. The **index** of a **Series** is an **optional parameter** that **defaults** to a **sequence of integers** starting at **zero**.
- The **Series** object is a **core data structure** in **Pandas**.

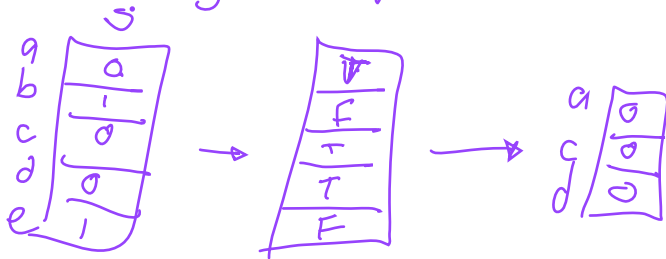


Querying a Series

- You could query a **Series** using indexing (boolean or fancy).

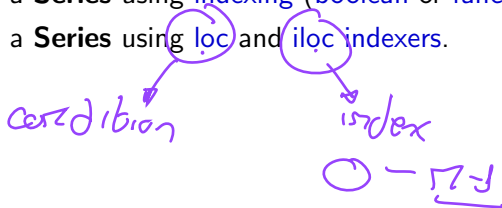
- You could query a Series using loc and ioc indexers.

$$s = s[cond] \rightarrow x=0$$



Querying a Series

- You could **query** a **Series** using **indexing** (boolean or fancy).
- You could **query** a **Series** using **loc** and **iloc** indexers.



 condition

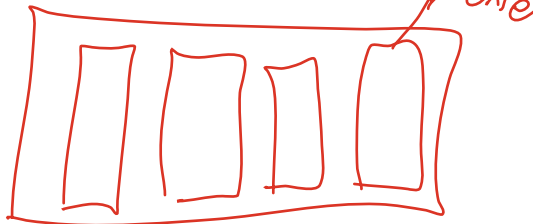
 index

 0 - 17



The "DataFrame" Data Structure

- A **DataFrame** is a two-dimensional labeled data structure with columns of potentially different types. *→ matrix*
- A **DataFrame** is a tabular data structure that is similar to a spreadsheet or a SQL table. *→ operations*
- A **DataFrame** is a core data structure in **Pandas**. It is a two-dimensional size-mutable data structure with labeled axes (rows and columns).
- A **DataFrame** is a container for Series objects. *→ serie*



DataFrame Indexing and Loading

- You could **index** a **DataFrame** using **column names**.
- You could **load** a **DataFrame** from a **CSV** file.
- You could **load** a **DataFrame** from a **JSON** file.
- You could **load** a **DataFrame** from a **SQL** database.



DataFrame Indexing and Loading

- You could **index** a **DataFrame** using **column names**.
- You could **load** a **DataFrame** from a **CSV file**.
- You could **load** a **DataFrame** from a **JSON file**.
- You could **load** a **DataFrame** from a **SQL database**.



DataFrame Indexing and Loading

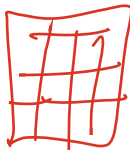
- You could **index** a **DataFrame** using **column names**.
- You could **load** a **DataFrame** from a **CSV file**.
- You could **load** a **DataFrame** from a **JSON file**.
- You could **load** a **DataFrame** from a **SQL database**.

different
SQL



DataFrame Indexing and Loading

- You could **index** a **DataFrame** using **column names**.
- You could **load** a **DataFrame** from a **CSV** file.
- You could **load** a **DataFrame** from a **JSON** file.
- You could **load** a **DataFrame** from a **SQL database**.



DateTime Handling in Pandas

- You could convert a string to a datetime object using the to_datetime() method.
- You could convert a datetime object to a string using the `strftime()` method.
Y-M-D H:M:S
- You could convert a datetime object to a timestamp using the `timestamp()` method.



DateTime Handling in Pandas

- You could **convert** a **string** to a **datetime** object using the **to_datetime()** method.
- You could **convert** a **datetime** object to a **string** using the **strftime()** method.
- You could **convert** a **datetime** object to a **timestamp** using the **timestamp()** method.



DateTime Handling in Pandas

- You could **convert** a **string** to a **datetime** object using the `to_datetime()` method.
- You could **convert** a **datetime** object to a **string** using the `strftime()` method.
- You could **convert** a **datetime** object to a **timestamp** using the `timestamp()` method.

Handwritten notes illustrating datetime formats and conversions:

Y-M-D H:M:S → Timestamp Y-M-D H:M:S.000 TZ

2025-06-18 10:00:00.000000 (7) → 0 70/10

The diagram shows a handwritten example of a datetime string "2025-06-18 10:00:00.000000" with a circled "7" and an arrow pointing to "0 70/10". Above it, the general format "Y-M-D H:M:S" is shown converting to a "Timestamp" format "Y-M-D H:M:S.000 TZ".



Querying a DataFrame

- You could **query** a **DataFrame** using **indexing** (**boolean** or **fancy**).
- You could **query** a **DataFrame** using **loc** and **iloc** indexers.
- You could **query** a **DataFrame** using **query** method.



Querying a DataFrame

- You could **query** a **DataFrame** using **indexing** (**boolean** or **fancy**).
- You could **query** a **DataFrame** using **loc** and **iloc** indexers.
- You could **query** a **DataFrame** using **query** method.



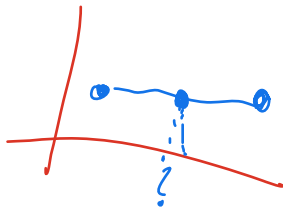
Querying a DataFrame

- You could **query** a **DataFrame** using **indexing** (**boolean** or **fancy**).
- You could **query** a **DataFrame** using **loc** and **iloc** indexers.
- You could **query** a **DataFrame** using **query method**.



Missing Values in a DataFrame

- You could **detect** missing values in a **DataFrame**. The **isnull()** method returns a **Boolean DataFrame** indicating the **presence** of missing values.
 - You could **fill** missing values in a **DataFrame**. The **fillna()** method returns a **DataFrame** with missing values filled.
 - You could **drop** missing values in a **DataFrame**. The **dropna()** method returns a **DataFrame** with missing values dropped.
- Handwritten notes:*
 Clear (next to fillna)
 Imputer → Fill (next to dropna)



Missing Values in a DataFrame

- You could **detect** missing values in a **DataFrame**. The **isnull()** method returns a **Boolean DataFrame** indicating the **presence** of missing values.
- You could **fill** missing values in a **DataFrame**. The **fillna()** method returns a **DataFrame** with missing values filled.
- You could **drop** missing values in a **DataFrame**. The **dropna()** method returns a **DataFrame** with missing values dropped.

fill → NaN



Missing Values in a DataFrame

- You could **detect** missing values in a **DataFrame**. The `isnull()` method returns a **Boolean DataFrame** indicating the **presence** of missing values.
- You could **fill** missing values in a **DataFrame**. The `fillna()` method returns a **DataFrame** with missing values filled.
- You could **drop** missing values in a **DataFrame**. The `dropna()` method returns a **DataFrame** with missing values dropped.

Null → NaN → row at least 1
 NaN → gone



Merging DataFrames

- You could **merge** two **DataFrames** using the **merge()** method.
- You could **concatenate** two **DataFrames** using the **concat()** method.
- You could **join** two **DataFrames** using the **join()** method.

Join
concatenations



Merging DataFrames

- You could **merge** two **DataFrames** using the **merge()** method.
- You could **concatenate** two **DataFrames** using the **concat()** method.
- You could ~~join~~ two **DataFrames** using the **join()** method.



Merging DataFrames

- You could **merge** two **DataFrames** using the **merge()** method.
- You could **concatenate** two **DataFrames** using the **concat()** method.
- You could **join** two **DataFrames** using the **join()** method.



GroupBy in Pandas

- You could **group** a **DataFrame** using the **groupby()** method.
- You could **aggregate** a **DataFrame** using the **agg()** method.
- You could **transform** a **DataFrame** using the **transform()** method.
- You could **filter** a **DataFrame** using the **filter()** method.



GroupBy in Pandas

- You could **group** a **DataFrame** using the **groupby()** method.
- You could **aggregate** a **DataFrame** using the **agg()** method.
- You could **transform** a **DataFrame** using the **transform()** method.
- You could **filter** a **DataFrame** using the **filter()** method.



GroupBy in Pandas

- You could **group** a **DataFrame** using the `groupby()` method.
- You could **aggregate** a **DataFrame** using the `agg()` method.
- You could **transform** a **DataFrame** using the `transform()` method.
- You could **filter** a **DataFrame** using the `filter()` method.



GroupBy in Pandas

- You could **group** a **DataFrame** using the `groupby()` method.
- You could **aggregate** a **DataFrame** using the `agg()` method.
- You could **transform** a **DataFrame** using the `transform()` method.
- You could **filter** a **DataFrame** using the `filter()` method.



Outline

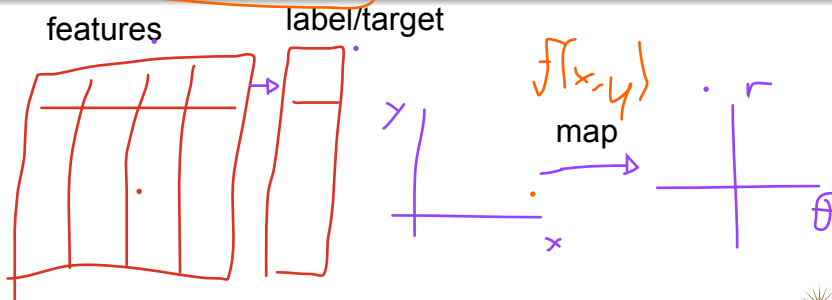
- 1 Fundamentals of Machine Learning
- 2 Python Tools for Machine Learning
- 3 Supervised Machine Learning
- 4 Machine Learning Models Evaluation



Introduction to Supervised Machine Learning

Definition

- **Supervised learning** is a type of **machine learning** where the model is trained on **labeled data**.
- It involves training a model to **map** **input data** to **output data** based on example **input-output pairs**.



Overfitting and Underfitting

Overfitting

Overfitting occurs when a model learns the training data too well and performs poorly on new data.

Underfitting

Underfitting occurs when a model is too simple to capture the underlying structure of the data.



Overfitting and Underfitting

Overfitting

Overfitting occurs when a model learns the training data too well and performs poorly on new data.

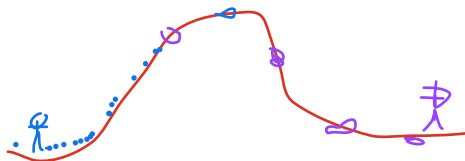
Underfitting

Underfitting occurs when a model is too simple to capture the underlying structure of the data.

Iterations

Hill Climbing

Learning Rate



Supervised Learning Datasets

- **Training Dataset:** The data used to **train the model**.

- **Validation Dataset:** The data used to **tune the model hyperparameters**.

- **Test Dataset:** The data used to **evaluate the model performance**.
Learning a pattern to map
features -> label

80-70
9



Supervised Learning Datasets

- **Training Dataset:** The data used to **train the model**.
- **Validation Dataset:** The data used to **tune** the model **hyperparameters**.
- **Test Dataset:** The data used to **evaluate** the model **performance**.

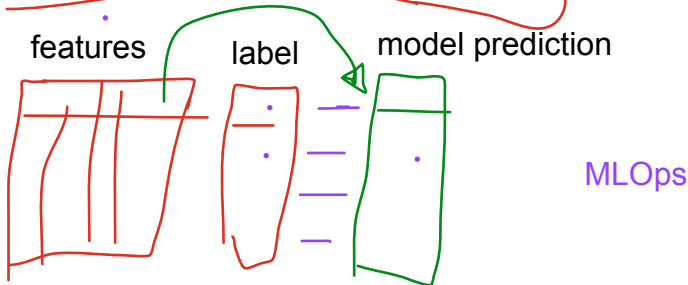
model setup

- * iterations
- * learning rate
- * optimization metric (error)



Supervised Learning Datasets

- **Training Dataset:** The data used to **train the model**.
- **Validation Dataset:** The data used to **tune** the model **hyperparameters**.
- **Test Dataset:** The data used to **evaluate** the model **performance**.



Cross-Validation

- **Cross-validation** is a technique for assessing the performance of a model.
- It involves splitting the data into multiple subsets, training the model on some subsets, and evaluating it on others.
- Common cross-validation **techniques** include k-fold cross-validation and leave-one-out cross-validation. 13 4
- Cross-validation helps to reduce overfitting and provides a more accurate estimate of the model's performance.



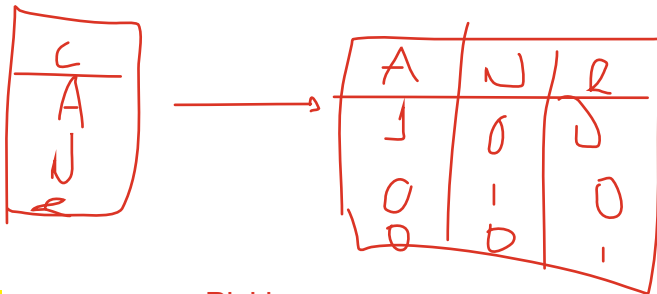
Training 1: 1234 Testing: 5
 • Training 2: 1235 Testing: 4



One-Hot Encoding

One-Hot Encoding

- One-hot encoding is a technique for converting categorical variables into numerical variables.
- It creates a binary vector for each category, with a 1 for the *category* and 0s for all other categories.

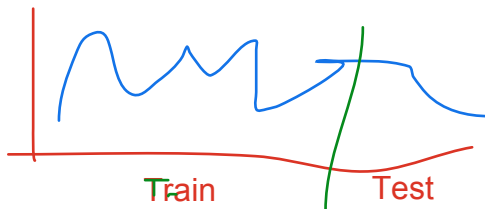


Pickle •



Data Leakage

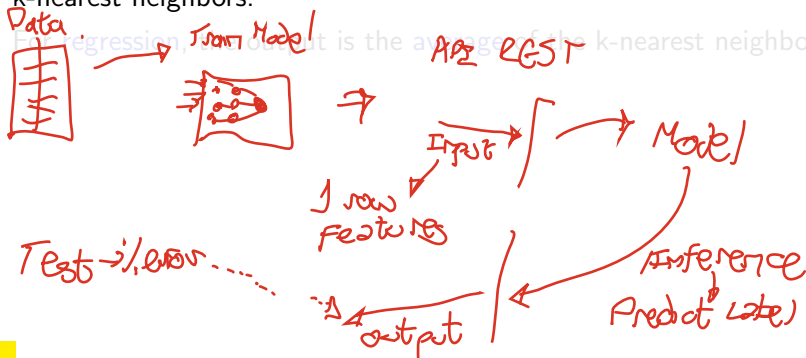
- **Data leakage** occurs when information from the test set is inadvertently used to train the model.
- It can lead to overfitting and inflated performance metrics.
- Common sources of **data leakage** include target leakage, train-test contamination, and information leakage.
- To prevent **data leakage**, it is important to carefully separate the training and test data and avoid using information from the test set during training.



K-Nearest Neighbors: Classification and Regression

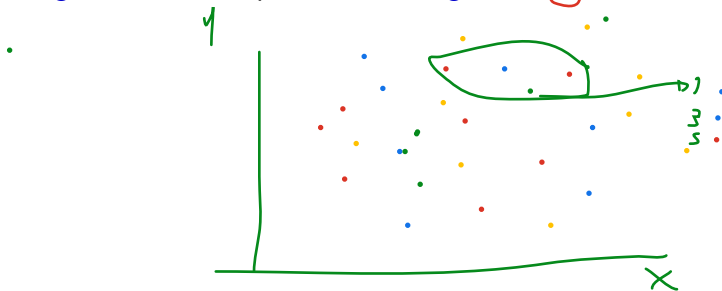
- **K-Nearest Neighbors (KNN)** is a simple algorithm that stores all available cases and classifies new cases based on a **similarity measure**.
- It can be used for both **classification** and **regression** tasks.
- For **classification**, the output is the **class label** of the majority of the k-nearest neighbors.

- For **regression**, the output is the **average** of the k-nearest neighbors.



K-Nearest Neighbors: Classification and Regression

- **K-Nearest Neighbors (KNN)** is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure.
- It can be used for both **classification** and **regression** tasks.
- For **classification**, the output is the **class label** of the majority of the k-nearest neighbors.
- For **regression**, the output is the **average** of the **k**-nearest neighbors.



Linear Regression with Least Squares

Linear Regression

- **Linear regression** is a type of regression analysis used for predicting the value of a continuous dependent variable.
- It works by finding the line that best fits the data.

Least Squares

Least squares is a method for finding the best-fitting line by minimizing the sum of the squared differences between the predicted and actual values.



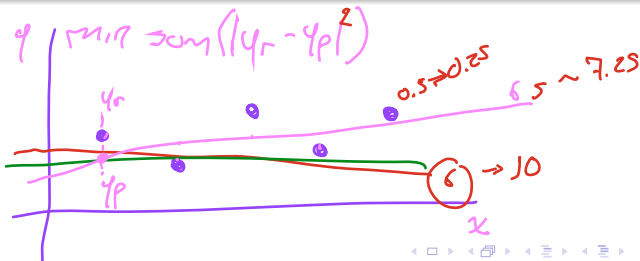
Linear Regression with Least Squares

Linear Regression

- **Linear regression** is a type of **regression analysis** used for predicting the value of a **continuous dependent variable**.
- It works by finding the **line that best fits the data**.

Least Squares

Least squares is a method for finding the **best-fitting line** by **minimizing the sum** of the squared differences between the **predicted** and **actual** values.

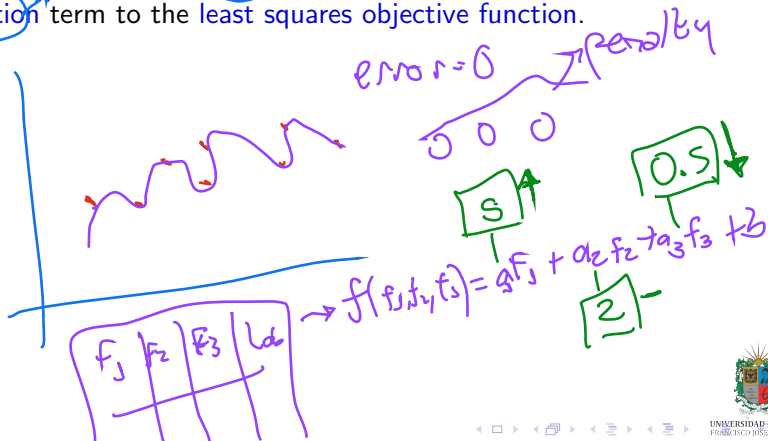


Ridge & Lasso

L1 → overfitting

L2 → feature importance

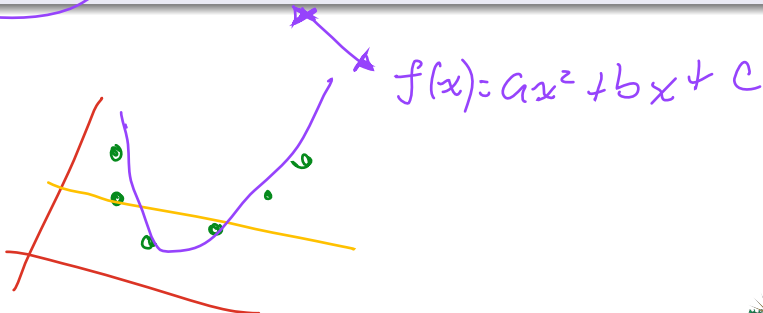
Ridge regression & **Lasso regression** are a type of linear regression that includes a penalty term to prevent overfitting. It works by adding a regularization term to the least squares objective function.



Polynomial Regression

Polynomial Regression

- **Polynomial regression** is a type of regression analysis that models the relationship between the independent and dependent variables as an n th-degree polynomial.
- It can capture non-linear relationships between the variables.



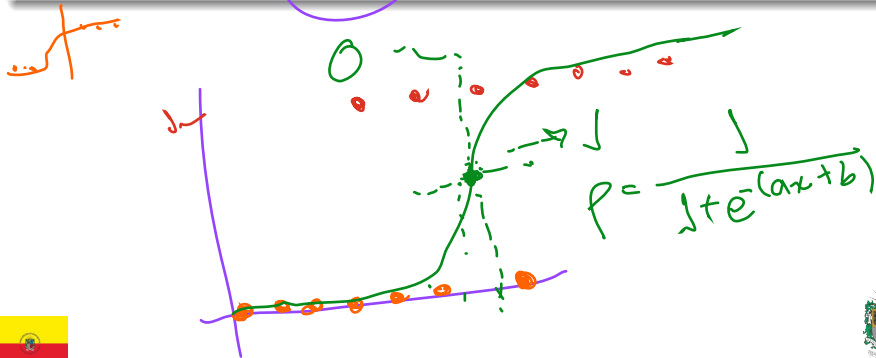
$$f(x) = ax^2 + bx + c$$



Logistic Regression

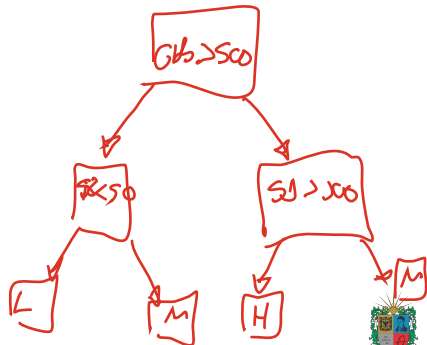
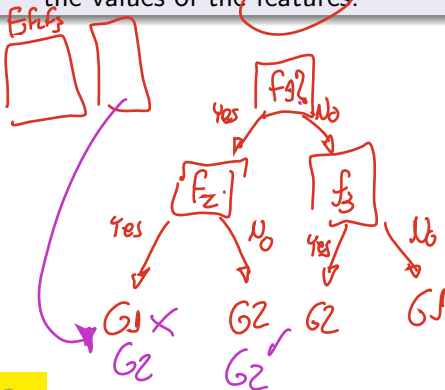
Logistic Regression

- **Logistic regression** is a type of regression analysis used for predicting the outcome of a categorical dependent variable.
- It is used for binary classification tasks, where the output is a probability between 0 and 1.



Decision Trees

- **Decision trees** are a type of **machine learning model** that can be used for both **classification** and **regression** tasks.
- They work by recursively **partitioning** the data into **subsets** based on the values of the features.



Naive Bayes Classifier

- The **naive Bayes classifier** is a simple probabilistic **classifier** based on Bayes' theorem.
- It assumes that the features are **conditionally independent** given the class label.

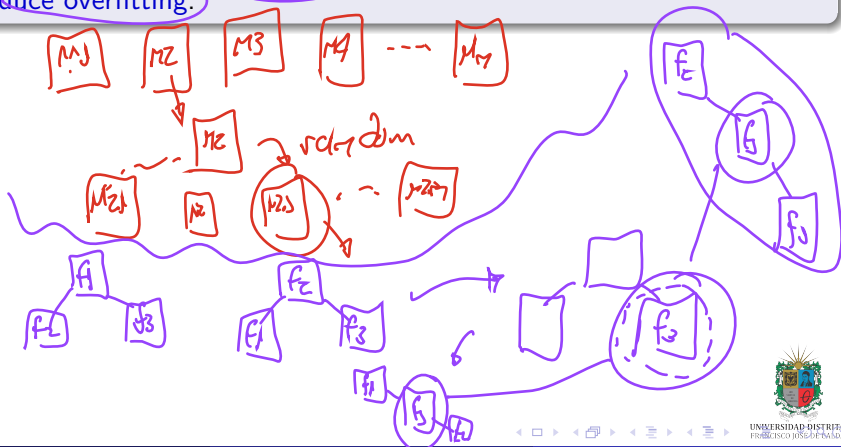
$P_{\text{rain}} \rightarrow P_{\text{wind}} \rightarrow P_{\text{sun}} \rightarrow P_{\text{dependent}}$

Baseline / performance



Random Forest

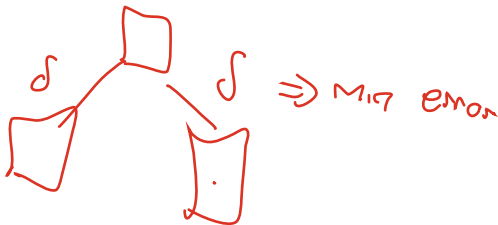
- **Random forest** is an **ensemble learning** method that combines **multiple decision trees** to create a strong predictive model.
- It works by building **multiple trees** and averaging their predictions to reduce overfitting.



Gradient Boosted Decision Trees

- **Gradient boosted decision trees** are an ensemble learning method that combines multiple decision trees and gradient descent optimization to create a strong predictive model.
- They work by building trees sequentially, with each tree correcting the errors of the previous trees.

Bagging



Neural Networks

- **Neural networks** are a type of **machine learning model** inspired by the **human brain**.
- They consist of **layers** of interconnected nodes that process **input data** and produce **output data**.



Outline

- 1 Fundamentals of Machine Learning
- 2 Python Tools for Machine Learning
- 3 Supervised Machine Learning
- 4 Machine Learning Models Evaluation**



Model Evaluation & Selection

- **Model Evaluation:** Assessing the **performance** of a model.
- **Model Selection:** Choosing the **best model** for the task.



Confusion Matrices

Definition

- A **confusion matrix** is a **table** that summarizes the **performance** of a **classification model**.
- It shows the number of **true positives**, **true negatives**, **false positives**, and **false negatives**.



Basic Evaluation Metrics

- **Accuracy:** The proportion of correct predictions.
- **Precision:** The proportion of true positives among all positive predictions.
- **Recall:** The proportion of true positives among all actual positives.
- **F1 Score:** The harmonic mean of precision and recall.



Basic Evaluation Metrics

- **Accuracy:** The proportion of correct predictions.
- **Precision:** The proportion of true positives among all positive predictions.
- **Recall:** The proportion of true positives among all actual positives.
- **F1 Score:** The harmonic mean of precision and recall.



Basic Evaluation Metrics

- **Accuracy:** The proportion of correct predictions.
- **Precision:** The proportion of true positives among all positive predictions.
- **Recall:** The proportion of true positives among all actual positives.
- **F1 Score:** The harmonic mean of precision and recall.



Basic Evaluation Metrics

- **Accuracy:** The proportion of correct predictions.
- **Precision:** The proportion of true positives among all positive predictions.
- **Recall:** The proportion of true positives among all actual positives.
- **F1 Score:** The harmonic mean of precision and recall.



Classifier Metrics

- **ROC Curve:** A plot of the true positive rate against the false positive rate.
- Precision-Recall Curve: A plot of precision against recall.
- AUC-ROC: The area under the ROC curve.
- AUC-PR: The area under the precision-recall curve.



Classifier Metrics

- **ROC Curve:** A plot of the true positive rate against the false positive rate.
- **Precision-Recall Curve:** A plot of precision against recall.
- **AUC-ROC:** The area under the ROC curve.
- **AUC-PR:** The area under the precision-recall curve.



Classifier Metrics

- **ROC Curve:** A plot of the true positive rate against the false positive rate.
- **Precision-Recall Curve:** A plot of precision against recall.
- **AUC-ROC:** The area under the ROC curve.
- **AUC-PR:** The area under the precision-recall curve.



Classifier Metrics

- **ROC Curve:** A plot of the true positive rate against the false positive rate.
- **Precision-Recall Curve:** A plot of precision against recall.
- **AUC-ROC:** The area under the ROC curve.
- **AUC-PR:** The area under the precision-recall curve.



Regression Metrics

- **Mean Squared Error:** The **average** of the **squared differences** between the predicted and actual values.
- **Mean Absolute Error:** The **average** of the **absolute differences** between the predicted and actual values.
- **R-Squared:** The **proportion** of the **variance** in the dependent variable that is predictable from the independent variables.
- **Adjusted R-Squared:** A modified version of R-squared that adjusts for the **number of predictors** in the model.
- **Root Mean Squared Error:** The **square root** of the **mean squared error**.



Regression Metrics

- **Mean Squared Error:** The **average** of the **squared differences** between the predicted and actual values.
- **Mean Absolute Error:** The **average** of the **absolute differences** between the predicted and actual values.
- **R-Squared:** The **proportion** of the **variance** in the dependent variable that is predictable from the independent variables.
- **Adjusted R-Squared:** A modified version of R-squared that adjusts for the **number of predictors** in the model.
- **Root Mean Squared Error:** The **square root** of the **mean squared error**.



Regression Metrics

- **Mean Squared Error:** The **average** of the **squared differences** between the predicted and actual values.
- **Mean Absolute Error:** The **average** of the **absolute differences** between the predicted and actual values.
- **R-Squared:** The **proportion** of the **variance** in the dependent variable that is predictable from the independent variables.
- **Adjusted R-Squared:** A modified version of R-squared that adjusts for the **number of predictors** in the model.
- **Root Mean Squared Error:** The **square root** of the mean squared error.



Regression Metrics

- **Mean Squared Error:** The **average** of the **squared differences** between the predicted and actual values.
- **Mean Absolute Error:** The **average** of the **absolute differences** between the predicted and actual values.
- **R-Squared:** The **proportion** of the **variance** in the dependent variable that is predictable from the independent variables.
- **Adjusted R-Squared:** A modified version of R-squared that adjusts for the **number of predictors** in the model.
- **Root Mean Squared Error:** The **square root** of the mean squared error.



Regression Metrics

- **Mean Squared Error:** The **average** of the **squared differences** between the predicted and actual values.
- **Mean Absolute Error:** The **average** of the **absolute differences** between the predicted and actual values.
- **R-Squared:** The **proportion** of the **variance** in the dependent variable that is predictable from the independent variables.
- **Adjusted R-Squared:** A modified version of R-squared that adjusts for the **number of predictors** in the model.
- **Root Mean Squared Error:** The **square root** of the mean squared error.



Outline

- 1 Fundamentals of Machine Learning
- 2 Python Tools for Machine Learning
- 3 Supervised Machine Learning
- 4 Machine Learning Models Evaluation



Thanks!

Questions?



Repo: <https://github.com/EngAndres/ud-public/tree/main/courses/systems-sciences-foundations>

