

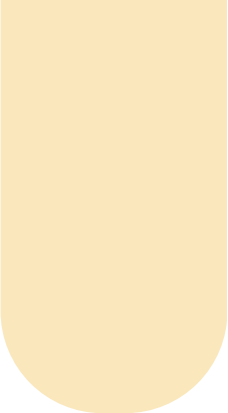




# INTRO TO ML

By Rishabh Dubey





# Quiz Time

# QUICK REVISION QUIZ (MCQS & TRUE/FALSE)

Question 1: What does `df.describe()` do in Pandas?

- (A) Shows first 5 rows
- (B) Shows statistical summary
- (C) Deletes missing values
- (D) Merges two DataFrames

# QUICK REVISION QUIZ (MCQS & TRUE/FALSE)

Question 2: Which of the following is not a NumPy function?

- (A) `np.mean()`
- (B) `np.array()`
- (C) `np.plot()`
- (D) `np.linspace()`

# QUICK REVISION QUIZ (MCQS & TRUE/FALSE)

Question 3: What is the primary purpose of Matplotlib?

- (A) Data visualization
- (B) Machine Learning
- (C) Data cleaning
- (D) API development

# QUICK REVISION QUIZ (MCQS & TRUE/FALSE)

Question 4: Which method is used in Pandas to remove NaN values?

- (A) dropna()
- (B) fillna()
- (C) replace()
- (D) clear()



# **Introduction to Machine Learning**

# WHAT IS MACHINE LEARNING?

ML allows computers to learn from data instead of being explicitly programmed.

Examples: Spam email detection

- Netflix recommendations
- Self-driving cars



# HOW ML DIFFERS FROM TRADITIONAL PROGRAMMING

## Traditional Programming



## Machine Learning



# TYPES OF ML

## **Supervised Learning**

- Uses labeled data
  - Examples:Spam detection
  - House Price Prediction

## **Unsupervised Learning**

- Uses unlabeled data
  - Examples:Customer Segmentation
  - Anomaly Detection

# TYPES OF ML

## Reinforcement Learning

Learning via rewards and penalties

Examples:

- Game AI
- Robotics
- Self-driving cars

💡 Example: Surveys, Elections, Market Research

# TYPES OF DATASETS

## What are Datasets?

- Collection of structured or unstructured data used for training ML models.

**A label is additional information (a classification, category, or meaningful value) mapped to raw data.**

## Labeled vs Unlabeled Data

Type	Definition	Example
<b>Labeled Data</b>	Contains input-output pairs	(X-ray, Disease Yes/No)
<b>Unlabeled Data</b>	No predefined labels	(Customer purchase history)



# **Machine Learning Lifecycle**



# ML LIFECYCLE OVERVIEW

1. Data Collection
2. Data Preprocessing
3. Feature Engineering
4. Model Selection
5. Model Training
6. Model Evaluation
7. Deployment

# ML LIFECYCLE OVERVIEW

## Data Collection

- Why is it needed? Models need quality data.
- What happens if skipped? Garbage in, garbage out.

## Data Preprocessing

- Removing Missing Values
- Handling Outliers
- Standardization/Normalization

# ML LIFECYCLE OVERVIEW

## Feature Engineering

- **Feature Selection:** Picking the most useful variables.
- **Feature Extraction:** Creating new features from existing ones.

## Model Selection

- Choosing the right algorithm (Linear Regression, Decision Trees, Neural Networks).



# ML LIFECYCLE OVERVIEW

## Model Training & Evaluation

- Splitting data into Train-Test Sets.
- Metrics: Accuracy, Precision, Recall, F1-score.

## Deployment & Monitoring

- Continuous Model Monitoring: If ignored, model degrades over time



# **Introduction to Regression**



# WHAT IS REGRESSION?

- Predicts continuous values.
- Examples: House prices, Sales prediction.

## Types of Regression

Type	Use Case
<b>Simple Linear Regression</b>	One input, one output
<b>Multiple Regression</b>	Multiple inputs
<b>Polynomial Regression</b>	Non-linear relationships

# ERROR METRICS FOR REGRESSION

**MSE** (Mean Squared Error)

**RMSE** (Root Mean Squared Error)

**MAE** (Mean Absolute Error)



# **Introduction to Scikit- Learn**

# INTRODUCTION TO SCIKIT-LEARN

What is Scikit-Learn?

- Popular Python ML Library

## Key Functions in Scikit-Learn

Function	Purpose
<b>train_test_split()</b>	Splits data into train-test
<b>LinearRegression()</b>	Creates a regression model
<b>fit()</b>	Trains the model
<b>predict()</b>	Makes predictions

# TASK - IMPLEMENT SIMPLE LINEAR REGRESSION

```
import numpy as np
from sklearn.linear_model import LinearRegression
X = np.array([1, 2, 3, 4, 5]).reshape(-1,1)
y = np.array([2, 4, 5, 4, 5])

model = LinearRegression()
model.fit(X, y)
print("Predictions:", model.predict(X))
```



**THANK YOU**