### 🎯 Objective:

Develop a strong conceptual and practical understanding of core machine learning workflows — from analyzing and preparing data to training, tuning, and evaluating machine learning models. This phase is essential before progressing into Deep Learning or Generative AI.

## 🔍 Curriculum Breakdown

### 1. Data Analysis & Visualization

#### 📌 1.1 Exploratory Data Analysis (EDA)

* Understand the structure, distribution, and quality of data
* Identify missing values, outliers, and anomalies
* Use .describe(), .info(), .isnull() from Pandas for quick inspection
* Grouping and aggregating data to uncover patterns
* Feature correlation matrices to evaluate relationships

#### 📌 1.2 Data Cleaning & Preprocessing

* Handling missing data:
  + Imputation: mean, median, mode
  + Dropping nulls based on context
* Outlier detection:
  + Z-score, IQR method
* Encoding categorical variables:
  + Label Encoding, One-Hot Encoding
* Scaling and Normalization:
  + MinMaxScaler, StandardScaler

#### 📌 1.3 Visualization Tools

* Histograms: distribution of numerical features
* Box plots: outlier and spread detection
* Scatter plots: bivariate analysis
* Heatmaps: correlation across all features
* Pair plots: combined views of data interactions

🧪 *Lab:* Load a dataset (e.g., Titanic, Iris, or a custom CSV), perform full EDA, visualize patterns, and handle missing/dirty data.

### 2. Statistics & Mathematics for ML

#### 📌 2.1 Probability & Distributions

* Basic probability rules
* Normal, binomial, and Poisson distributions
* Sampling techniques and population assumptions

#### 📌 2.2 Descriptive vs. Inferential Statistics

* Mean, median, mode, standard deviation
* Central limit theorem
* Hypothesis testing:
  + Null vs. alternative hypotheses
  + p-values, confidence intervals

#### 📌 2.3 Linear Algebra & Calculus Essentials

* Vectors and matrices: operations, dot products
* Matrix transformations: basis for PCA
* Gradients and derivatives: introduction to cost optimization

🧪 *Notebook Exercise:* Visualize a cost function and demonstrate gradient descent manually using Python/NumPy.

### 3. Machine Learning Foundations

#### 📌 3.1 Types of Machine Learning

* Supervised Learning: labeled data → classification/regression
* Unsupervised Learning: pattern discovery → clustering, dimensionality reduction
* Reinforcement Learning: reward-based learning (brief introduction)

#### 📌 3.2 Core Concepts

* Train-test split and the purpose of validation
* Features vs. target variables
* Overfitting and underfitting:
  + Model complexity vs. performance
  + Use of learning curves
* Bias-variance tradeoff:
  + How simplicity and flexibility affect generalization

🧪 *Mini-Activity:* Use train\_test\_split and build a simple model to demonstrate overfitting visually.

### 4. Key Algorithms

#### 📌 4.1 Regression

* Linear Regression: line fitting using least squares
* Polynomial Regression: curve fitting with degree > 1
* Regularization:
  + L1 (Lasso): feature selection
  + L2 (Ridge): prevents overfitting

#### 📌 4.2 Classification

* Logistic Regression: sigmoid and class probabilities
* Decision Trees: splits, entropy, Gini index
* Random Forests: ensemble learning and bootstrapping
* SVM: linear and non-linear classification with margin maximization

#### 📌 4.3 Clustering (Unsupervised)

* K-Means: centroid-based grouping
* DBSCAN: density-based clustering for irregular shapes
* Hierarchical Clustering: dendrograms and agglomerative strategies

🧪 *Lab:* Train models using Scikit-learn and visually compare regression/classification results on structured datasets.

### 5. Model Evaluation Techniques

#### 📌 5.1 Performance Metrics

* Classification:
  + Accuracy, Precision, Recall, F1-score
  + Confusion Matrix
  + ROC-AUC curve for probability-based classifiers
* Regression:
  + MAE, MSE, RMSE, R² Score

#### 📌 5.2 Model Validation & Tuning

* Cross-validation:
  + k-fold, stratified k-fold
  + Why it's essential for small/imbalanced datasets
* Hyperparameter tuning:
  + GridSearchCV, RandomizedSearchCV
  + Practical tuning examples (e.g., depth of trees, number of neighbors)

🧪 *Project:* Evaluate multiple classifiers on the same dataset and tune hyperparameters to maximize F1-score.

### 6. Tools & Libraries

* Scikit-learn: end-to-end ML — preprocessing, modeling, tuning, and evaluation
* Pandas & NumPy: data transformation, statistical analysis, matrix operations
* Matplotlib & Seaborn: visualizing both raw data and model results

## 🧪 Capstone Mini-Project

Project Title: *Predict Customer Churn for a Telecom Company*

* Clean and explore a real customer dataset
* Perform EDA and visualize churn trends
* Build and evaluate 3 classifiers
* Tune models for F1 and AUC
* Document insights and push to GitHub