**🔹 1. Data Collection**

* Understand the problem you're solving.
* Identify and collect relevant datasets (CSV, database, API, etc.).
* Merge multiple sources if needed.

**🔹 2. Data Understanding / Exploration**

* **Inspect the data:**
  + Use .head(), .info(), .describe(), and .shape()
* **Check data types and ranges**
* **Understand target variable distribution**
* **Class imbalance?**
* **Check for duplicates**
* **Check for outliers** using boxplots, z-scores, or IQR.

**🔹 3. Data Cleaning / Wrangling**

* **Handle missing values**
  + Drop missing rows/columns
  + Impute using mean/median/mode or domain knowledge
* **Fix data types**
  + Convert dates, categories, numerics properly
* **Standardize/normalize string formats**
  + E.g., "Male", "male", "M" → "Male"
* **Remove/encode outliers (if needed)**
* **Handle duplicates**

**🔹 4. Exploratory Data Analysis (EDA)**

* **Univariate Analysis**
  + Histograms, KDE plots, value counts
* **Bivariate/Multivariate Analysis**
  + Scatter plots, pair plots
  + Correlation matrix (for numerical)
  + Groupby aggregation (for categorical)
* **Visualize the target vs. features**
  + Boxplots for classification
  + Scatter plots for regression
* **Check class distribution (classification)**

**🔹 5. Feature Engineering**

* **Create new features** from domain knowledge or interactions
  + Date → day, month, year, weekday, weekend
  + Ratio features (e.g., price per unit)
* **Aggregate features** (e.g., groupby stats)
* **Transform skewed features**
  + Use log, sqrt, or Box-Cox transforms
* **Convert categorical to numerical**
  + One-hot encoding
  + Label encoding
  + Target/mean encoding (with caution)

**🔹 6. Feature Scaling**

* **StandardScaler (Z-score)**: for most ML models
* **MinMaxScaler**: when features must be in [0,1]
* **RobustScaler**: if outliers are present
* **No scaling needed for**: tree-based models (e.g., Random Forest, XGBoost)

**🔹 7. Feature Selection**

* **Remove irrelevant/constant features**
  + Using variance threshold
* **Correlation analysis**
  + Drop highly correlated features (multicollinearity)
* **Univariate feature selection**
  + ANOVA, Chi-square test
* **Model-based feature importance**
  + RandomForest, XGBoost, etc.
* **Recursive Feature Elimination (RFE)**
* **PCA / Dimensionality reduction**
  + If interpretability is less important

**🔹 8. Model Selection**

* **Split data**
  + Train/Validation/Test or use Cross-validation (KFold/StratifiedKFold)
* **Try multiple models**
  + Regression: Linear, Ridge, Lasso, ElasticNet, Tree-based
  + Classification: Logistic, RandomForest, XGBoost, SVM, KNN, etc.
* **Baseline model**: Simple model to benchmark against

**🔹 9. Model Evaluation**

* **Classification Metrics**: Accuracy, Precision, Recall, F1, ROC-AUC, Confusion Matrix
* **Regression Metrics**: RMSE, MAE, R², Adjusted R²
* **Use Cross-validation scores**

**🔹 10. Hyperparameter Tuning**

* **Grid Search / Random Search**
* **Bayesian Optimization (Optuna, Hyperopt)**
* Use cross-validation during tuning

**🔹 11. Model Interpretation**

* **Feature importance plots**
* **SHAP / LIME for explainability**
* Check for **bias** and **fairness**

**🔹 12. Deployment (Optional)**

* Export model with pickle, joblib, or ONNX
* Create a REST API (Flask/FastAPI)
* Deploy to cloud (AWS, GCP, Azure) or Streamlit web app

**🔹 13. Monitoring & Maintenance**

* Track model performance over time
* Retrain periodically if concept drift happens
* Log predictions and monitor live data