📘 Machine Learning Notes (Beginner to Intermediate)

---

1. Types of Machine Learning

🔹 Supervised Learning

- Data has **input features (X)** and **labels (y)**.
- Goal: Learn mapping X→yX \to y.
- Examples:
    - House price prediction (Regression)
    - Spam detection (Classification)

🔹 Unsupervised Learning

- Data has only **features (X)**, no labels.
- Goal: Find hidden patterns/structure.
- Examples:
    - Customer segmentation (Clustering)
    - Dimensionality reduction (PCA)

🔹 Reinforcement Learning

- Agent interacts with environment.
- Learns via **rewards and penalties**.
- Examples:
    - Self-driving cars
    - Game-playing AI

---

2. Supervised ML Algorithms with Code

1️⃣ Linear Regression (Continuous Prediction)

- **Goal**: Predict continuous numbers.
- **Equation**: $y=m1x1+m2x2+...+by = m\_1x\_1 + m\_2x\_2 + ... + b$

```
from sklearn.linear_model import LinearRegression
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Dataset: size (sqft) vs price
X = np.array([[1000], [1500], [2000], [2500], [3000]])
y = np.array([50, 70, 90, 110, 150])  # prices in lakhs

# Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train
model = LinearRegression()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
print("Predictions:", y_pred)
print("MSE:", mean_squared_error(y_test, y_pred))
```

2️⃣ Logistic Regression (Binary Classification)

- **Goal**: Classify into two classes (Yes/No).
- **Equation**:
  $P(y=1∣x)=11+e−(mx+b)P(y=1|x) = \frac{1}{1+e^{-(mx+b)}}$

```
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_breast_cancer
from sklearn.metrics import accuracy_score

model = LogisticRegression(max_iter=10000)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

3️⃣ Decision Tree 🌳

- **Goal**: Learn rules by splitting data.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_iris

# Dataset
iris = load_iris()
X, y = iris.data, iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

4️⃣ Random Forest 🌲🌲🌲

- **Goal**: Combine many decision trees → better accuracy.

```
from sklearn.ensemble import RandomForestClassifier

# Train
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

5️⃣ k-Nearest Neighbors (KNN) 👯

- **Goal**: Classify based on nearest “k” neighbors.

```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

6️⃣ Support Vector Machine (SVM) ✈️

- **Goal**: Find the best separating boundary (hyperplane).

```
from sklearn.svm import SVC

# Train
model = SVC(kernel='linear')
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

7️⃣ Naive Bayes 🎲

- **Goal**: Text and probability-based classification.

```
from sklearn.naive_bayes import GaussianNB

# Train
model = GaussianNB()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

---

3. Quick Comparison Table

Algorithm	Type	Best For	Example
Linear Regression	Regression	Predict continuous values	House prices 🏠
Logistic Regression	Classification	Binary classification	Spam detection 📧
Decision Tree	Both	Rule-based models	Loan approval 🌳
Random Forest	Both	High accuracy, less overfitting	Price prediction
KNN	Classification	Similarity-based classification	Fruit classification 🍎
SVM	Classification	High-dimensional data	Email filtering ✈️
Naive Bayes	Classification	Text-based classification	Spam, sentiment 🎲