Hamna Jamila

**AKSA- AI Internship**

**Machine Learning Fundamentals**

**Supervised Learning Models**

Supervised learning is when you have data with answers (labels) and want to predict answers for new data.

1. **Linear Regression**

Linear Regression predicts numbers (like house prices) by finding a straight line that best fits your data. It looks at your features (like house size) and learns how they connect to the number you want to predict.

* **How to plot it?**
  + plotting points on a graph (features vs. target).
  + It draws a straight line through those points to make predictions.
  + The line is described by a formula: y = a + b\*x (a is the starting point, b is the slope).
  + It adjusts a and b to make predictions as close as possible to real values.
* **When to Use**:
  + Predicting numbers, like sales, temperatures, or prices.
  + When you think features and the target have a linear line relationship.
* **Pros**:
  + Easy to understand and use and fast for small datasets.
* **Cons**:
  + Doesn’t work well if data isn’t linear (like curves) and its bad with outliers (extreme values mess up the line).

**Example**

Predicting house price (y) based on house size (x). If bigger houses cost more, Linear Regression finds the best line to predict price from size.

1. **Logistic Regression**

Logistic Regression predicts categories (like yes/no, spam/not spam) by giving probabilities. It’s like deciding if an email is spam based on certain clues.

* **How it works?**
  + Takes features (like email length) and calculates a score.
  + Uses a special function (sigmoid) to turn the score into a probability (0 to 1).
  + If probability > 0.5, predicts one category (e.g., spam); else, the other (e.g., not spam).
* **When to Use**:
  + For yes/no or two-class problems (binary classification).
  + When you want probabilities, not just predictions.
* **Pros**:
  + Gives probabilities, which are useful.
* **Cons**:
  + Struggles with complex patterns plus needs data to be somewhat linear in a transformed space.

**Example**

Predicting if a customer will buy (1) or not (0) based on their age and income.

1. **K-Nearest Neighbors (KNN)**

KNN predicts by looking at similar examples. Majority wins.

* **How It Works**:
  + Stores all your data.
  + When predicting, finds the K closest data points (neighbors) using distance (like Euclidean distance).
  + For classification: Takes the majority vote of neighbors’ labels.
  + For regression: Averages neighbors’ values.
* **When to Use**:
  + For classification (e.g., is this a cat or dog?) or regression.
  + When you don’t want to assume anything about the data.
* **Pros**:
  + Super simple: just compares distances.
  + Works for any data pattern.
* **Cons**:
  + Slow for big datasets (checks every point).
  + Needs features on the same scale (normalize data).

**Example**

Classifying a fruit as apple or orange by comparing its size and color to known fruits.

1. **Decision Trees**

Decision Trees make predictions by asking a series of yes/no questions, like a flowchart. Each question splits the data, leading to a final answer.

* **How It Works**:
  + Starts at the top (root) and asks a question about a feature (e.g., “Is age > 30?”).
  + Splits data into two groups based on the answer.
  + Keeps splitting until it reaches a final answer (leaf).
  + Chooses splits that best separate the data (e.g., reduce uncertainty).
* **When to Use**:
  + For classification or regression.
  + When you want a clear, interpretable model.
* **Pros**:
  + Easy to understand and visualize.
  + Works with numbers or categories without scaling.
* **Cons**:
  + Can overfit (learns too much noise from data).
  + Changes a lot if data changes slightly.

**Example**

Deciding if a loan applicant is risky by asking: “Is income high?” → “Is credit score good?” → Approve/Reject.

**Unsupervised Learning Models**

Unsupervised learning is when you have data without answers and want to find patterns, like grouping similar items.

1. **K-Means Clustering**

K-Means groups data into K clusters by putting similar points together. It’s like sorting candies into piles based on color.

* **How It Works**:
  + Pick K random points as cluster centers (centroids).
  + Assign each data point to the closest centroid.
  + Move centroids to the average of their points.
  + Repeat until centroids stop moving.
* **When to Use**:
  + To group data, like customers with similar buying habits.
  + When you know how many groups (K) you want.
* **Pros**:
  + Fast and easy for simple datasets.
  + Good for clear, round clusters.
* **Cons**:
  + You must choose K (can be tricky), recommended to use between 3 to 11.
  + Bad for weird-shaped or uneven clusters.
* **Example**: Grouping customers into 3 types based on spending and age.

1. **Principal Component Analysis (PCA)**

PCA simplifies data by squeezing it into fewer dimensions while keeping the important patterns.

* **How It Works**:
  + Centers data (subtracts the average).
  + Finds new directions (components) where data varies the most.
  + Projects data onto these directions to reduce dimensions.
* **When to Use**:
  + To reduce features (e.g., from 10 to 2) for visualization or faster ML.
  + When features are related (correlated).
* **Pros**:
  + Makes data smaller and easier to work with.
  + Helps remove noise.
* **Cons**:
  + Hard to interpret new dimensions.
  + Only works for linear patterns.
* **Example**: Reducing 3D data (height, weight, age) to 2D for a scatter plot.