কে-নিয়ারেস্ট নেইবার (K-Nearest Neighbors / KNN) অ্যালগরিদম

মেশিন লার্নিং নোটস — সহজ বাংলায় ব্যাখ্যা

১. KNN কী?

K-নিয়ারেস্ট নেইবার (সংক্ষেপে **KNN**) হলো মেশিন লার্নিং-এর একটি **সুপারভাইজড লার্নিং (Supervised Learning)** অ্যালগরিদম, যা **ক্লাসিফিকেশন (Classification)** এবং **রিগ্রেশন (Regression)**—দুই ধরনের সমস্যাতেই ব্যবহার করা যায়। তবে এটি ক্লাসিফিকেশনের জন্যই বেশি জনপ্রিয়।

> KNN-এর মূল ধারণা খুবই সহজ — **"তুমি কেমন মানুষ তা তোমার বন্ধুদের দেখেই বোঝা যায়"**। অর্থাৎ, কোনো নতুন ডেটা পয়েন্ট কোন ক্লাসের হবে তা নির্ধারণ করতে, তার আশেপাশে (নিকটবর্তী) থাকা "K" সংখ্যক পুরনো ডেটা পয়েন্ট দেখা হয়, এবং সেই প্রতিবেশীদের (neighbors) মধ্যে যে ক্লাসটি সবচেয়ে বেশি সংখ্যায় আছে, নতুন ডেটাকেও সেই ক্লাসে ফেলা হয়।

KNN-কে **"লেজি লার্নার" (Lazy Learner)** বলা হয়, কারণ এটি ট্রেনিং ধাপে আসলে কিছুই "শেখে" না — শুধু পুরো ডেটাসেটটি মুখস্থ (store) করে রাখে। প্রকৃত হিসাব-নিকাশ তখনই হয়, যখন একটি নতুন ডেটা পয়েন্ট প্রেডিক্ট করতে বলা হয়। একে **ইনস্ট্যান্স-বেসড লার্নিং (Instance-based Learning)**-ও বলা হয়।

২. KNN কীভাবে কাজ করে?

ধরা যাক, আমাদের কাছে ট্রেনিং ডেটাসেট আছে এবং একটি নতুন (অজানা) ডেটা পয়েন্টের ক্লাস বের করতে হবে। KNN নিচের ধাপগুলো অনুসরণ করে:

1. **K-এর মান নির্ধারণ করা:** প্রথমেই ঠিক করতে হয় কতগুলো নিকটবর্তী প্রতিবেশী (neighbor) বিবেচনা করা হবে — অর্থাৎ K-এর মান কত হবে (যেমন K=৩, K=৫)।
2. **দূরত্ব পরিমাপ করা:** নতুন ডেটা পয়েন্ট থেকে ট্রেনিং ডেটাসেটের প্রতিটি পয়েন্টের দূরত্ব (distance) হিসাব করা হয়।
3. **নিকটতম K-টি প্রতিবেশী বাছাই করা:** দূরত্বের ভিত্তিতে ছোট থেকে বড় সাজিয়ে সবচেয়ে কাছের K-সংখ্যক ডেটা পয়েন্ট বেছে নেওয়া হয়।
4. **ভোটাভুটি (Voting) বা গড় করা:**
   - **ক্লাসিফিকেশনের ক্ষেত্রে:** ঐ K-টি প্রতিবেশীর মধ্যে যে ক্লাস সবচেয়ে বেশি (majority voting) পাওয়া যায়, নতুন ডেটাকে সেই ক্লাসে ফেলা হয়।
   - **রিগ্রেশনের ক্ষেত্রে:** K-টি প্রতিবেশীর মানের গড় (average) নিয়ে সেটিই প্রেডিকশন হিসেবে দেওয়া হয়।

> **উদাহরণ:** ধরা যাক, আমরা একজন নতুন ব্যক্তির উচ্চতা ও ওজন দেখে বলতে চাই তিনি "স্বাস্থ্যবান" নাকি "অসুস্থ" ক্যাটাগরির।
>
> যদি K = ৫ হয়, তাহলে নতুন ডেটা পয়েন্টের সবচেয়ে কাছের ৫ জন পুরনো ব্যক্তিকে খুঁজে বের করা হবে। ধরা যাক তাদের মধ্যে ৩ জন "স্বাস্থ্যবান" এবং ২ জন "অসুস্থ" ক্যাটাগরির — তাহলে majority voting অনুযায়ী নতুন ব্যক্তিকেও "স্বাস্থ্যবান" হিসেবে চিহ্নিত করা হবে।

৩. দূরত্ব পরিমাপের পদ্ধতি (Distance Metrics)

KNN-এর মূল ভিত্তিই হলো দূরত্ব পরিমাপ। সবচেয়ে বেশি ব্যবহৃত পদ্ধতিগুলো হলো:

ক) ইউক্লিডিয়ান দূরত্ব (Euclidean Distance)

এটি সবচেয়ে বেশি প্রচলিত পদ্ধতি — দুটি বিন্দুর মধ্যে সরলরৈখিক (straight-line) দূরত্ব মাপা হয়।

$$\text{দূরত্ব} = \sqrt{[\,(x_2-x_1)^2 + (y_2-y_1)^2 + \dots\,]}$$

খ) ম্যানহাটন দূরত্ব (Manhattan Distance)

এখানে দুটি বিন্দুর মধ্যে দূরত্ব হিসাব করা হয় একেবারে গ্রিড আকারে (যেমন শহরের রাস্তার মতো আড়াআড়ি ও লম্বালম্বি পথ ধরে), কর্ণ বরাবর নয়।

$$\text{দূরত্ব} = |x_2-x_1| + |y_2-y_1| + \dots$$

গ) মিনকোভস্কি দূরত্ব (Minkowski Distance)

এটি ইউক্লিডিয়ান ও ম্যানহাটন দূরত্বের একটি সাধারণীকৃত (generalized) রূপ, যেখানে একটি প্যারামিটার (p) পরিবর্তন করে দুই ধরনের দূরত্বই পাওয়া যায়।

৪. সঠিক K-এর মান বাছাই করা

KNN-এ সবচেয়ে গুরুত্বপূর্ণ সিদ্ধান্তগুলোর একটি হলো K-এর মান কত হবে তা ঠিক করা। ভুল K বাছাই করলে মডেলের পারফরম্যান্স খারাপ হতে পারে।

- **K-এর মান ছোট হলে (যেমন K=১):** মডেল ডেটার প্রতিটি খুঁটিনাটি (noise) পর্যন্ত ধরে ফেলে, ফলে **ওভারফিটিং (Overfitting)** হওয়ার সম্ভাবনা বেড়ে যায় — মডেল অতিরিক্ত সংবেদনশীল (sensitive) হয়ে পড়ে।
- **K-এর মান বড় হলে:** মডেল অতিরিক্ত সাধারণীকৃত (generalized) হয়ে যায়, ফলে গুরুত্বপূর্ণ প্যাটার্নও উপেক্ষিত হতে পারে — একে **আন্ডারফিটিং (Underfitting)** বলা হয়।
- **জোড় সংখ্যা এড়িয়ে চলা:** ক্লাসিফিকেশনে সাধারণত K-এর মান বিজোড় (odd) রাখা হয়, যাতে ভোটাভুটিতে টাই (tie) না হয় (বিশেষ করে দুই-ক্লাসের সমস্যায়)।
- সাধারণত **ক্রস-ভ্যালিডেশন (Cross-validation)** পদ্ধতি ব্যবহার করে বিভিন্ন K-এর মান পরীক্ষা করে দেখা হয়, কোন মানে মডেল সবচেয়ে ভালো ফলাফল দেয়।

৫. ফিচার স্কেলিং কেন জরুরি?

KNN সম্পূর্ণভাবে দূরত্বের হিসাবের উপর নির্ভরশীল, তাই যদি ডেটাসেটের ফিচারগুলোর মানের পরিসর (scale) ভিন্ন ভিন্ন হয় (যেমন একটি ফিচারের মান ০-১ এর মধ্যে, আরেকটির মান ০-১০০০০ এর মধ্যে), তাহলে বড় মানের ফিচারটি দূরত্বের হিসাবে অতিরিক্ত প্রভাব ফেলবে। এই সমস্যা এড়াতে ডেটা ব্যবহারের আগে **নরমালাইজেশন (Normalization)** বা **স্ট্যান্ডার্ডাইজেশন (Standardization)** করে সবগুলো ফিচারকে একই স্কেলে আনা অত্যন্ত জরুরি।

৬. KNN-এর সুবিধা (Advantages)

- ✔ **সহজ ও বোঝা সহজ:**
  অ্যালগরিদমের ধারণা ও বাস্তবায়ন খুবই সরল, নতুনদের জন্য বোঝা সহজ।
- ✔ **ট্রেনিং সময় লাগে না:**
  যেহেতু এটি লেজি লার্নার, তাই ট্রেনিং ধাপে কোনো মডেল তৈরির (জটিল গণনার) প্রয়োজন হয় না — শুধু ডেটা সংরক্ষণ করা হয়।
- ✔ **নতুন ডেটা সহজে যোগ করা যায়:**
  নতুন ট্রেনিং ডেটা যোগ করতে মডেল আবার তৈরি করার দরকার হয় না।
- ✔ **নন-লিনিয়ার ডেটাতেও কার্যকর:**
  ডেটার মধ্যে জটিল সীমানা (decision boundary) থাকলেও KNN তা ভালোভাবে ধরতে পারে।
- ✔ **বহুমুখী ব্যবহার:**
  ক্লাসিফিকেশন ও রিগ্রেশন — উভয় সমস্যাতেই ব্যবহার করা যায়।

৭. KNN-এর অসুবিধা (Disadvantages)

- ✘ **প্রেডিকশনের সময় ধীরগতি:**
  প্রতিটি নতুন ডেটা পয়েন্টের জন্য পুরো ট্রেনিং ডেটাসেটের সাথে দূরত্ব হিসাব করতে হয়, যা বড় ডেটাসেটে খুবই সময়সাপেক্ষ ও ব্যয়বহুল (computationally expensive)।
- ✘ **বেশি মেমরি প্রয়োজন:**
  পুরো ট্রেনিং ডেটাসেট মেমরিতে সংরক্ষণ করে রাখতে হয়।
- ✘ **"ডাইমেনশনালিটির অভিশাপ" (Curse of Dimensionality):**
  ফিচারের সংখ্যা (dimension) বেশি হলে দূরত্বের হিসাব অর্থপূর্ণ থাকে না এবং মডেলের পারফরম্যান্স কমে যায়।
- ✘ **ফিচার স্কেলিং-এর প্রতি সংবেদনশীল:**
  স্কেলিং না করলে ফলাফল ভুল হতে পারে।
- ✘ **অপ্রাসঙ্গিক ফিচার ও noise-এর প্রতি সংবেদনশীল:**
  ডেটাসেটে অপ্রয়োজনীয় ফিচার বা outlier থাকলে দূরত্বের হিসাব বিভ্রান্ত হতে পারে।
- ✘ **ইমব্যালান্সড ডেটাতে সমস্যা:**
  কোনো একটি ক্লাসের ডেটা বেশি থাকলে, ভোটাভুটিতে সেই ক্লাসই বেশি জিতে যাওয়ার সম্ভাবনা থাকে, প্রকৃত ফলাফল যাই হোক না কেন।

৮. K-এর মান ও ওভারফিটিং/আন্ডারফিটিং সম্পর্ক

K-এর মান	প্রভাব	ঝুঁকি
খুব ছোট (K=১)	মডেল অতিরিক্ত সংবেদনশীল, noise-ও ধরে ফেলে	ওভারফিটিং
উপযুক্ত (Optimal)	ভারসাম্যপূর্ণ পারফরম্যান্স	সবচেয়ে ভালো ফলাফল
খুব বড়	মডেল অতিরিক্ত সাধারণীকৃত, প্যাটার্ন হারায়	আন্ডারফিটিং

৯. বাস্তব জীবনে ব্যবহার (Use Cases)

- **রিকমেন্ডেশন সিস্টেম:** কোনো ব্যবহারকারীর পছন্দের সাথে মিল রেখে সিনেমা, পণ্য বা গান সাজেস্ট করা (যেমন Netflix, Amazon)।
- **ইমেজ রিকগনিশন:** হাতের লেখা সংখ্যা বা ছবি শনাক্তকরণ (image classification)।
- **মেডিকেল ডায়াগনসিস:** রোগীর উপসর্গের সাথে পুরনো রোগীদের তুলনা করে রোগ শনাক্ত করা।
- **ক্রেডিট স্কোরিং:** নতুন গ্রাহকের ঋণ পরিশোধের সম্ভাবনা পুরনো গ্রাহকদের ডেটার সাথে তুলনা করে অনুমান করা।
- **প্যাটার্ন রিকগনিশন:** স্প্যাম ইমেইল শনাক্তকরণ, হাতের লেখা শনাক্তকরণ ইত্যাদি।

১০. সংক্ষিপ্ত সারাংশ (Summary)

- KNN একটি সহজ, ইনস্ট্যান্স-বেসড (instance-based) সুপারভাইজড লার্নিং অ্যালগরিদম, যা "নিকটতম প্রতিবেশী"-দের দেখে সিদ্ধান্ত নেয়।
- এটি "লেজি লার্নার" — ট্রেনিং সময়ে কিছু শেখে না, শুধু ডেটা সংরক্ষণ করে; আসল হিসাব হয় প্রেডিকশনের সময়।

- দূরত্ব পরিমাপ (সাধারণত ইউক্লিডিয়ান) এবং সঠিক K-এর মান বাছাই করাই এই অ্যালগরিদমের মূল চ্যালেঞ্জ।
- ফিচার স্কেলিং করা অত্যাবশ্যক, নয়তো ফলাফল ভুল হতে পারে।
- বড় ডেটাসেটে ধীরগতি ও বেশি মেমরির প্রয়োজন হওয়ায়, বাস্তব প্রয়োগে এটি ছোট থেকে মাঝারি আকারের ডেটাসেটে বেশি কার্যকর।