

লিনিয়ার রিগ্রেশন

মেশিন লার্নিংয়ের একটি সুপরিচিত অ্যালগরিদম, যা ডিপেন্ডেন্ট ভেরিয়েবল (Y) এবং ইনডিপেন্ডেন্ট ভেরিয়েবলগুলোর (X) মধ্যে সম্পর্ক নির্ধারণ করতে ব্যবহার হয়। এটি ডেটাকে একটি সরল রেখার মাধ্যমে ফিট করে, যেখানে রেখার সাধারণ সমীকরণ হলো $y=mx+cy = mx + c$, যেখানে mm হলো স্লোপ এবং cc হলো ইন্টারসেপ্ট। এটি মূলত কন্টিনিউয়াস ডেটা প্রেডিকশনের জন্য ব্যবহার করা হয়, যেমন বিক্রির পূর্বাভাস, তাপমাত্রা বা আয়ের পূর্বাভাস। লিনিয়ার রিগ্রেশন দুই ধরনের হতে পারে: সিম্পল লিনিয়ার রিগ্রেশন (একটি ইনডিপেন্ডেন্ট ভেরিয়েবল) এবং মাল্টিপল লিনিয়ার রিগ্রেশন (একাধিক ইনডিপেন্ডেন্ট ভেরিয়েবল)।

পলিনোমিয়াল রিগ্রেশন

একটি মেশিন লার্নিং অ্যালগরিদম, যা নন-লিনিয়ার সম্পর্কযুক্ত ডেটার উপর ভিত্তি করে কাজ করে। এটি লিনিয়ার রিগ্রেশনের একটি এক্সটেনশন, যেখানে ইনডিপেন্ডেন্ট ভেরিয়েবলগুলোর বিভিন্ন পাওয়ার (ডিগ্রি) যোগ করে একটি পলিনোমিয়াল সমীকরণ তৈরি করা হয়, যেমন $y=b_0+b_1x+b_2x^2+...+b_nx^n$ । এই অ্যালগরিদমটি মূলত তখন ব্যবহার করা হয়, যখন ডেটা একটি সরল রেখায় ফিট না করে এবং বক্রাকার (curve) প্যাটার্ন দেখায়। এটি জটিল ডেটা মডেলিং, যেমন গ্রাফ প্রিডিকশন বা ফিজিক্যাল প্রসেস সিমুলেশন, এর জন্য কার্যকর।

নেইভ বেইজ ক্লাসিফায়ার

বেয়েস থিওরি (Bayes' Theorem) এর উপর ভিত্তি করে কাজ করে। বেয়েস থিওরি এক ঘটনার সম্ভাবনা নির্ধারণ করে অন্য একটি ঘটনার উপর ভিত্তি করে।

থিওরির সমীকরণ:

$$P(A|B)=P(B|A) \cdot P(A)P(B)P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

এখানে,

- $P(A|B)$: B ঘটনার প্রদত্ত অবস্থায় A ঘটনার সম্ভাবনা।
- $P(B|A)$: A ঘটনার প্রদত্ত অবস্থায় B ঘটনার সম্ভাবনা।
- $P(A)$: A ঘটনার পূর্বাভাস সম্ভাবনা।
- $P(B)$: B ঘটনার পূর্বাভাস সম্ভাবনা।

নেইভ বেইজ ক্লাসিফায়ার ফিচারগুলোর মধ্যে স্বাধীনতার (independence) অনুমান করে এবং এই সূত্র ব্যবহার করে শ্রেণীবিন্যাস করে। এটি সহজ, দ্রুত, এবং টেক্সট-ভিত্তিক ক্লাসিফিকেশনে খুবই কার্যকর।

লজিস্টিক রিগ্রেশন (Binomial)

একটি মেশিন লার্নিং অ্যালগরিদম, যা মূলত বাইনারি ক্লাসিফিকেশনের জন্য ব্যবহৃত হয়। এটি ডিপেন্ডেন্ট ভেরিয়েবলকে দুটি সম্ভাব্য ক্লাস (যেমন, 0 বা 1, হ্যাঁ বা না) এ শ্রেণীবদ্ধ করে।

মূল ধারণা: লজিস্টিক রিগ্রেশন লিনিয়ার রিগ্রেশনের উপর ভিত্তি করে কাজ করে, তবে ফলাফলকে একটি লজিস্টিক (সিগময়েড) ফাংশনের মাধ্যমে ম্যাপ করে, যাতে আউটপুট সর্বদা 0 এবং 1 এর মধ্যে থাকে। সিগময়েড ফাংশনটি হল:

$$h(x)=\frac{1}{1+e^{-z}}h(x) = \frac{1}{1 + e^{-z}}$$

যেখানে,

- $z=b_0+b_1x_1+b_2x_2+...+b_nx_n$
- $h(x)$: সম্ভাবনা স্কেল (প্রেডিকশন)।

কাজের ধাপ:

- লিনিয়ার কম্বিনেশন: ইনডিপেন্ডেন্ট ভেরিয়েবল থেকে একটি স্কেল তৈরি করে।
- সিগময়েড ফাংশন: স্কেলকে সম্ভাবনায় রূপান্তরিত করে।
- ডিসিশন বাউন্ডারি: সম্ভাবনা 0.5 এর বেশি হলে এক ক্লাস, কম হলে অন্য ক্লাস হিসেবে শ্রেণীবিন্যাস করে।

লজিস্টিক রিগ্রেশন (Multinomial)

একটি মেশিন লার্নিং অ্যালগরিদম, যা মাল্টি-ক্লাস ক্লাসিফিকেশন (যখন ডিপেন্ডেন্ট ভেরিয়েবল একাধিক ক্লাসে বিভক্ত থাকে) সমাধানের জন্য ব্যবহৃত হয়। এটি বাইনারি লজিস্টিক রিগ্রেশনের এক্সটেনশন, যেখানে আউটপুট ক্লাস 0 এবং 1 এর মধ্যে সীমাবদ্ধ না থেকে একাধিক হতে পারে।

মূল ধারণা:

মাল্টিনোমিয়াল লজিস্টিক রিগ্রেশন সফটম্যাক্স ফাংশন ব্যবহার করে সম্ভাবনা নির্ধারণ করে, যাতে প্রতিটি ক্লাসের সম্ভাবনা 0 এবং 1 এর মধ্যে থাকে এবং সব সম্ভাবনার যোগফল 1 হয়।

সফটম্যাক্স ফাংশন:

$$P(y=i|x) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad P(y=i|x) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

যেখানে,

- $z_i = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$
- k : সম্ভাব্য ক্লাসের সংখ্যা।
- $P(y=i|x)P(y=i|x)P(y=i|x)$: ইনপুট x -এর ক্লাস i -এর সম্ভাবনা।

কাজের ধাপ:

- লিনিয়ার কস্টফাংশন: প্রতিটি ক্লাসের জন্য আলাদা স্কেল নির্ধারণ করা।
- সফটম্যাক্স ফাংশন: প্রতিটি ক্লাসের স্কেলকে সম্ভাবনায় রূপান্তর করা।
- সর্বোচ্চ সম্ভাবনা: সবচেয়ে বেশি সম্ভাবনার ক্লাসকে আউটপুট হিসেবে চিহ্নিত করা।

k-Nearest Neighbors (k-NN)

একটি সহজ ও বহুল ব্যবহৃত সুপারভাইজড মেশিন লার্নিং অ্যালগরিদম, যা শ্রেণীবিন্যাস (classification) এবং রিগ্রেশন (regression) কাজের জন্য ব্যবহৃত হয়। এটি কনসেপ্টually **instance-based learning** বা **lazy learning** পদ্ধতির উপর ভিত্তি করে কাজ করে।

মূল ধারণা:

- ক নিকটতম প্রতিবেশী খুঁজে বের করা:
ইনপুট ডেটাপয়েন্ট থেকে "k" সংখ্যক নিকটতম প্রতিবেশী চিহ্নিত করা হয়।
- ডিস্ট্যান্স মেট্রিক ব্যবহার:
ডেটাপয়েন্টগুলোর মধ্যে দূরত্ব পরিমাপের জন্য বিভিন্ন পদ্ধতি ব্যবহার করা হয়, যেমন:
 - ইউক্লিডিয়ান ডিস্ট্যান্স (Euclidean Distance): $d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$
 - ম্যানহাটন ডিস্ট্যান্স, কসমসিন সিমিলারিটি ইত্যাদি।
- ভোটিং (**classification**):
প্রতিবেশীদের সংখ্যাগরিষ্ঠ ভোট অনুযায়ী শ্রেণী নির্ধারণ করা হয়।
- গড় গণনা (**regression**):
নিকটতম প্রতিবেশীদের গড় মান ব্যবহার করে ফলাফল নির্ধারণ করা হয়।

কাজের ধাপ:

- ডেটাসেট থেকে নতুন ডেটাপয়েন্টের জন্য দূরত্ব পরিমাপ।
- "k" সংখ্যক নিকটতম প্রতিবেশী নির্বাচন।

3. প্রতিবেশীদের লেবেল বা মান থেকে আউটপুট প্রেডিকশন।

Classification and Regression Tree (CART) হলো একটি সুপরিচিত ডিসিশন ট্রি অ্যালগরিদম, যা ডেটাকে শ্রেণীবদ্ধ (classification) এবং পূর্বাভাস (regression) করার জন্য ব্যবহার করা হয়। এটি ডেটাকে পুনরাবৃত্তি করে বিভক্ত (split) করার মাধ্যমে কাজ করে এবং একটি গাছের (tree) আকারে ফলাফল প্রদান করে।

মূল ধারণা:

- Classification Tree:**
 - যখন টার্গেট ভেরিয়েবল শ্রেণীগত (categorical) হয়।
 - প্রতিটি গাছের পাতা (leaf) একটি শ্রেণীকে উপস্থাপন করে।
 - বিভাজনের মানদণ্ড: **Gini Impurity** বা **Information Gain**।
- Regression Tree:**
 - যখন টার্গেট ভেরিয়েবল সংখ্যাগত (numerical) হয়।
 - প্রতিটি গাছের পাতা একটি নির্দিষ্ট মান (যেমন গড়) ধারণ করে।
 - বিভাজনের মানদণ্ড: **Mean Squared Error (MSE)**।

Random Forest হলো একটি শক্তিশালী এনসেম্বল মেশিন লার্নিং অ্যালগরিদম, যা শ্রেণীবিন্যাস (classification) এবং পূর্বাভাস (regression) উভয়ের জন্য ব্যবহার করা হয়। এটি বুটস্ট্র্যাপিং এবং বাগিং (bootstrap aggregating) কৌশল ব্যবহার করে অনেকগুলো ডিসিশন ট্রি তৈরি করে এবং তাদের সম্মিলিত ফলাফলের উপর ভিত্তি করে আউটপুট প্রদান করে।

মূল ধারণা:

- এনসেম্বল মডেল:
 - অনেকগুলো ডিসিশন ট্রি তৈরি করা হয়, যা ডেটার বিভিন্ন সাবসেট এবং ফিচার ব্যবহার করে প্রশিক্ষিত হয়।
 - প্রতিটি ট্রি স্বতন্ত্রভাবে ভবিষ্যদ্বাণী করে।
- ভোটিং বা গড়:
 - Classification:** প্রতিটি ট্রি তার শ্রেণী ভোট প্রদান করে, এবং সংখ্যাগরিষ্ঠ ভোটের ভিত্তিতে আউটপুট নির্ধারণ করা হয়।
 - Regression:** প্রতিটি ট্রির গড় মান ব্যবহার করে আউটপুট নির্ধারণ করা হয়।

কাজের ধাপ:

- বুটস্ট্র্যাপ স্যাম্পলিং:
 - ডেটাসেট থেকে পুনরাবৃত্তি করে এলোমেলোভাবে ডেটার সাবসেট তৈরি করা।
- ফিচার সাবসেট নির্বাচন:
 - প্রতিটি ট্রি তৈরি করার সময় একটি এলোমেলো ফিচার সাবসেট ব্যবহার করা।
- মডেল প্রশিক্ষণ:
 - প্রতিটি ট্রি আলাদাভাবে প্রশিক্ষিত হয়।
- সম্মিলিত ফলাফল:
 - Classification-এ সংখ্যাগরিষ্ঠ ভোট।
 - Regression-এ গড় মান।

Adaptive Boosting (AdaBoost) একটি শক্তিশালী এনসেম্বল মেশিন লার্নিং অ্যালগরিদম, যা বুটস্ট্র্যাপিং বা র‍্যান্ডম স্যাম্পলিং ব্যবহার না করে, পরিবর্তে একাধিক দুর্বল মডেল (weak learners) প্রশিক্ষণ করে এবং তাদের শক্তিশালী মডেল তৈরির জন্য একত্রিত করে। এটি বাগিং এবং বুস্টিং পদ্ধতির মধ্যে একটি, যেখানে প্রতিটি পরবর্তী মডেল আগের মডেলের ভুলগুলো সংশোধন করার চেষ্টা করে।

মূল ধারণা:

- দুর্বল মডেল (**Weak Learners**):
 - AdaBoost সাধারণত একটি সাধারণ ক্লাসিফায়ার, যেমন ডিসিশন স্টাম্প (একটি ডি যার মাত্র একটানা বিভাজন) ব্যবহার করে।
 - প্রথমে, একটি দুর্বল মডেল তৈরি করা হয় এবং এর ফলাফলে ভুল সিদ্ধান্তগুলো চিহ্নিত করা হয়।
- ভুল সংশোধন:
 - পরবর্তী মডেলটি পূর্ববর্তী মডেলের ভুল সিদ্ধান্তের উপর বেশি গুরুত্ব দেয় (এগুলোর ওজন বৃদ্ধি করা হয়), যাতে এগুলোর জন্য আরো ভালো ফলাফল পাওয়া যায়।
- ওজন (**Weighting**):
 - প্রতিটি ডেটাপয়েন্টের জন্য ওজন নির্ধারণ করা হয়। প্রথমে সব ডেটাপয়েন্টের জন্য সমান ওজন থাকে, তবে প্রতিটি ভুল প্রেডিকশনের জন্য সংশ্লিষ্ট ডেটাপয়েন্টের ওজন বাড়ানো হয়।
 - শেষ পর্যন্ত, অনেকগুলো দুর্বল মডেলকে একত্রিত করে একটি শক্তিশালী (strong) মডেল তৈরি করা হয়।

কাজের ধাপ:

- প্রথমে একটি দুর্বল মডেল তৈরি করা হয় এবং তার উপর প্রশিক্ষণ দেয়া হয়।
- প্রতিটি ভুল সিদ্ধান্তের জন্য ডেটাপয়েন্টের ওজন বৃদ্ধি করা হয়।
- পরবর্তী মডেলটি এই পরিবর্তিত ডেটাসেট ব্যবহার করে প্রশিক্ষিত হয়।
- এই প্রক্রিয়া পুনরাবৃত্তি করা হয়, যতক্ষণ না নির্দিষ্ট সংখ্যক মডেল তৈরি হয়।
- শেষ পর্যন্ত, সব মডেলগুলির আউটপুট একত্রিত করে (ভোটিং বা গড়) চূড়ান্ত ফলাফল প্রদান করা হয়।

ব্যবহারের ক্ষেত্র:

- Classification:**
 - স্প্যাম ডিটেকশন।
 - চেহারা শনাক্তকরণ।
- Regression:**
 - গৃহ মূল্য পূর্বাভাস।
 - বিক্রয় পরিমাণ ভবিষ্যদ্বাণী।

Gradient Boosting

একটি শক্তিশালী এনসেম্বল লার্নিং অ্যালগরিদম, যা বুস্টিং কৌশল ব্যবহার করে একাধিক দুর্বল মডেল (যেমন, ডিসিশন ট্রি) তৈরি করে এবং তাদের সম্মিলিত ফলাফলকে শক্তিশালী মডেল হিসাবে একত্রিত করে। এটি গ্রেডিয়েন্ট ডেসেন্ট পদ্ধতি ব্যবহার করে, যেখানে প্রতিটি নতুন মডেল পূর্ববর্তী মডেলের বাকি ত্রুটি (error) সংশোধন করতে চেষ্টা করে।

মূল ধারণা:

- গ্রেডিয়েন্ট ডেসেন্ট:
 - প্রতিটি নতুন ট্রি পূর্ববর্তী ট্রির ত্রুটি (residual errors) হ্রাস করার চেষ্টা করে।
 - গ্রেডিয়েন্ট হল ফাংশনের ডেরিভেটিভ, যা প্রতিটি পয়েন্টে সবচেয়ে দ্রুত পরিবর্তনের দিকে নির্দেশ করে। গ্রেডিয়েন্ট বুস্টিং-এর মধ্যে, প্রতিটি নতুন ট্রি এই ত্রুটির গ্রেডিয়েন্ট অনুসরণ করে একটি নতুন মডেল তৈরি করে।
- প্রক্রিয়া:
 - প্রথমে, একটি বেস লাইন মডেল (যেমন, একটি ট্রি) তৈরি করা হয়।
 - পরবর্তী মডেলটি আগের মডেলের ত্রুটির উপর ভিত্তি করে প্রশিক্ষিত হয়।
 - এই প্রক্রিয়া পুনরাবৃত্তি করা হয়, যতক্ষণ না নির্দিষ্ট সংখ্যা বা কনভারজেন্স পর্যন্ত ত্রুটি কমে না যায়।
- সফট-আপডেট:
 - প্রতিটি ট্রি শুধুমাত্র একটি ছোট হারে (learning rate) আপডেট করে, যাতে মডেলটি অতিরিক্ত ফিট না হয়।

কাজের ধাপ:

1. প্রাথমিক মডেল:
প্রথমে একটি সাধারণ মডেল (যেমন, একটি ছোট ডিসিশন ট্রি) তৈরি করা হয়, যা ডেটার একটি সাধারণ সারাংশ দেয়।
2. ত্রুটি হিসাব:
প্রথম মডেলের ত্রুটি বা অবশিষ্ট মান (residual) নির্ধারণ করা হয়।
3. গ্রেডিয়েন্ট আপডেট:
পরবর্তী মডেলটি আগের মডেলের ত্রুটির দিকে নির্দেশ করে গ্রেডিয়েন্ট অনুসরণ করে। এই আপডেটটি একটি ছোট হারে (learning rate) করা হয়।
4. সম্মিলিত ফলাফল:
সব মডেলগুলির প্রেডিকশন একত্রিত করে চূড়ান্ত ফলাফল নির্ধারণ করা হয়।
 - **Classification:** সংখ্যাগরিষ্ঠ ভোট।
 - **Regression:** গড় মান।

কাজের ধাপ:

1. প্রথম মডেল:
 - প্রথমে একটি দুর্বল মডেল তৈরি করা হয় (সাধারণত ডিসিশন স্টাম্প), যা ডেটার একটি সাধারণ সারাংশ দেয়।
2. ত্রুটি হিসাব:
 - প্রথম মডেলের ত্রুটি বা অবশিষ্ট মান (residual) নির্ধারণ করা হয়।
3. গ্রেডিয়েন্ট আপডেট:
 - পরবর্তী মডেলটি পূর্ববর্তী মডেলের ত্রুটির দিকে নির্দেশ করে গ্রেডিয়েন্ট অনুসরণ করে। এই আপডেটটি একটি ছোট হারে (learning rate) করা হয়।
4. রেগুলারাইজেশন:
 - অতিরিক্ত ফিটিং রোধ করতে L1 এবং L2 রেগুলারাইজেশন ব্যবহৃত হয়।
5. সম্মিলিত ফলাফল:
 - সব মডেলগুলির প্রেডিকশন একত্রিত করে চূড়ান্ত ফলাফল নির্ধারণ করা হয়।
 - **Classification:** সংখ্যাগরিষ্ঠ ভোট।
 - **Regression:** গড় মান।

ব্যবহারের ক্ষেত্র:

- **Classification:**
 - চোরাচালান শনাক্তকরণ।
 - ক্রেডিট স্কোরিং এবং ঝুঁকি ডিটেকশন।
- **Regression:**
 - বাড়ির দাম পূর্বাভাস।
 - স্টক মার্কেট প্রেডিকশন।

Support Vector Machine (SVM)

একটি জনপ্রিয় এবং শক্তিশালী সুপারভাইজড মেশিন লার্নিং অ্যালগরিদম, যা মূলত ক্লাসিফিকেশন এবং রিগ্রেশন সমস্যা সমাধানে ব্যবহৃত হয়। SVM মূলত একটি বিষম রেখা (hyperplane) তৈরি করে, যা ডেটাসেটের দুটি শ্রেণীর মধ্যে সঠিক বিভাজন তৈরি করে।

মূল ধারণা:

1. হাইপারপ্লেন:

- SVM ডেটাসেটের মধ্যে শ্রেণী বিভাজন করার জন্য একটি হাইপারপ্লেন তৈরি করে, যা ডেটা পয়েন্টগুলিকে দুটি শ্রেণীতে আলাদা করে। 2D ক্ষেত্রে, এটি একটি রেখা হতে পারে, তবে উচ্চতর মাত্রায় এটি একটি উচ্চমাত্রিক পৃষ্ঠ (hyperplane) হয়।
- 2. মার্জিন (Margin):
 - SVM এমন একটি হাইপারপ্লেন খোঁজে, যা শ্রেণীগুলির মধ্যে সবচেয়ে বড় মার্জিন তৈরি করে। অর্থাৎ, হাইপারপ্লেনের কাছাকাছি থাকা পয়েন্টগুলির মধ্যে দূরত্ব সর্বাধিক হয়। এটি মার্জিন ম্যাক্সিমাইজেশন হিসাবে পরিচিত।
- 3. সাপোর্ট ভেক্টর (Support Vectors):
 - সাপোর্ট ভেক্টর গুলি সেই ডেটা পয়েন্ট যা হাইপারপ্লেনের সবচেয়ে কাছাকাছি থাকে এবং বিভাজনকে নির্ধারণ করতে সহায়ক। এই পয়েন্টগুলিই মূলত শ্রেণী নির্ধারণের জন্য গুরুত্বপূর্ণ।

কাজের ধাপ:

1. ডেটা পয়েন্টগুলি:
প্রথমে ডেটা পয়েন্টগুলি দুটি শ্রেণীতে বিভক্ত করা হয়। SVM তাদের মধ্যবর্তী শ্রেণীগুলিকে বিভাজন করতে একটি হাইপারপ্লেন খুঁজে বের করে।
2. হাইপারপ্লেন নির্বাচন:
SVM একটি হাইপারপ্লেন নির্বাচন করে যা শ্রেণীগুলির মধ্যে সবচেয়ে বড় মার্জিন তৈরি করে।
3. সাপোর্ট ভেক্টর নির্বাচন:
সাপোর্ট ভেক্টরগুলি খুঁজে বের করা হয়, যেগুলি হাইপারপ্লেনের কাছাকাছি থাকে এবং বিভাজন প্রক্রিয়ায় সাহায্য করে।
4. মডেল প্রশিক্ষণ:
SVM প্রশিক্ষণ ডেটা ব্যবহার করে সঠিক হাইপারপ্লেন তৈরি করে, এবং নতুন ডেটার জন্য শ্রেণী পূর্বাভাস করে।

ব্যবহার:

- **Classification:**
 - চোহারা শনাক্তকরণ।
 - স্প্যাম ডিটেকশন।
 - স্বাক্ষর শনাক্তকরণ।
- **Regression:**
 - বাড়ির দাম পূর্বাভাস।
 - স্টক প্রাইস ফোরকাস্টিং।

K-Fold Cross Validation একটি পরিমাপ পদ্ধতি যা মেশিন লার্নিং মডেলের সাধারণীকরণ (generalization) ক্ষমতা যাচাই করতে ব্যবহৃত হয়। এটি মূলত ডেটাসেটটিকে কয়েকটি ছোট ফোল্ড (subsets) বা ভাগে বিভক্ত করে এবং প্রতিটি ভাগে মডেলটি পরীক্ষা করে। এই পদ্ধতিটি মডেলের পারফরম্যান্সের একটি নির্ভরযোগ্য মূল্যায়ন প্রদান করে, যা অতিরিক্ত ফিটিং (overfitting) এবং অবিশ্বস্ত ফলাফল এড়াতে সাহায্য করে।

কাজের ধাপ:

1. ডেটা বিভাজন:
 - প্রথমে পুরো ডেটাসেটটি **K**টি ভাগে (ফোল্ডে) বিভক্ত করা হয়। উদাহরণস্বরূপ, যদি **K=5** হয়, তবে ডেটাসেটটি 5টি ভাগে ভাগ করা হবে।
2. প্রশিক্ষণ এবং পরীক্ষা:
 - একটি ফোল্ড পরীক্ষা (test) সেট হিসেবে ব্যবহৃত হয় এবং বাকি K-1টি ফোল্ড প্রশিক্ষণ (training) সেট হিসেবে ব্যবহৃত হয়।
 - মডেলটি প্রশিক্ষণ সেটে প্রশিক্ষিত হয় এবং পরবর্তীতে পরীক্ষা সেটে পরীক্ষিত হয়।
3. প্রক্রিয়া পুনরাবৃত্তি:
 - এই প্রক্রিয়াটি **K**টি ফোল্ডের প্রতিটি ক্ষেত্রে পুনরাবৃত্তি করা হয়, যাতে প্রতিটি ফোল্ড একবার পরীক্ষা হিসেবে ব্যবহৃত হয়।
4. পারফরম্যান্স গড়:
 - প্রতিটি কৃত্য বা পরীক্ষা সেটের পারফরম্যান্স পরিমাপ করা হয় এবং এর পর গড় করা হয়। এর মাধ্যমে মডেলের গড় পারফরম্যান্স পাওয়া যায়।

উদাহরণ:

ধরা যাক, $K=5$ । ডেটাসেটটি 5টি সমান ভাগে ভাগ করা হয়েছে। প্রথমে প্রথম ফোল্ডকে পরীক্ষা সেট হিসেবে রেখে বাকি 4 ফোল্ডে মডেলটি প্রশিক্ষণ করা হবে। পরবর্তী ফোল্ডকে পরীক্ষা সেট হিসেবে রেখে আবার প্রশিক্ষণ করা হবে, এবং এই প্রক্রিয়া 5টি ফোল্ডের জন্য পুনরাবৃত্তি করা হবে। শেষে, মডেলের গড় পারফরম্যান্স (যেমন, অ্যাকুরেসি) রিপোর্ট করা হবে।