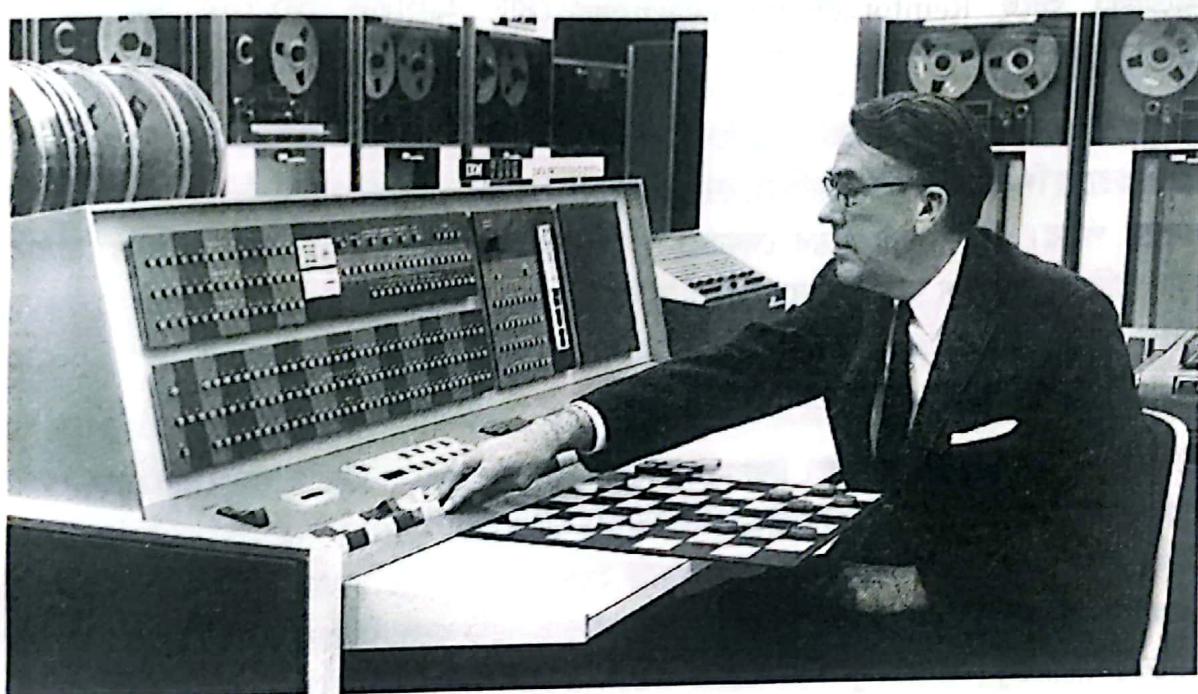


অধ্যায় ২: বিভিন্ন ধরনের লার্নিং ও অন্যান্য

মেশিন লার্নিং – এই টার্মটি সবার আগে ব্যবহার করেন আর্থার স্যামুয়েল (Arthur Samuel, 1901-1990) নামের একজন আমেরিকান বিজ্ঞানী। তাঁকে আর্টিফিশিয়াল ইন্টেলিজেন্স ও কম্পিউটার গেমিংয়ের একজন পথিকৃৎ বলা চলে।



ছবি 2.1 : আর্থার স্যামুয়েল (Arthur Samuel, 1901-1990)

তাঁর তৈরি করা স্যামুয়েল চেকার্স প্লেয়িং প্রোগ্রামটিই বিশ্বের প্রথম প্রোগ্রাম, যা নিজে নিজে খেলা শিখতে পেরেছিল। স্যামুয়েল পড়াশোনা করেছিলেন ম্যাসাচুসেটস ইনসিটিউট অব টেকনোলজি (Massachusetts Institute of Technology – MIT) থেকে। পরবর্তীকালে তিনি বিশ্বখ্যাত বেল ল্যাবরেটরিতে কাজ করেছেন। এর মধ্যে বেশ কিছুদিন তিনি আইবিএম (IBM)-এ কাজ করেছেন। শেষ বয়সে এসে তিনি স্ট্যানফোর্ড বিশ্ববিদ্যালয় (Stanford University)-এ প্রফেসর হিসেবে যোগদান করেন। তিনি আইইইই (IEEE) কম্পিউটার সোসাইটি থেকে কম্পিউটার পাইওনিয়ার অ্যাওয়ার্ড (Computer Pioneer Award)-সহ আরো অসংখ্য উল্লেখযোগ্য পুরস্কারে ভূষিত হয়েছেন। তাঁর মৃত্যু হয় পারকিনসন্স রোগের কারণে।

পরিচ্ছেদ ২.১ : সুপারভাইজড ও আনসুপারভাইজড লার্নিং (Supervised and Unsupervised Learning)

মেশিন লার্নিং অ্যালগরিদমগুলোকে সাধারণত দুটি ভাগে ভাগ করা যায় :

- Supervised Learning
- Unsupervised Learning

আরেকটি আছে Reinforcement Learning, সেটি আপাতত আমাদের এই বইয়ের একত্তিয়ারের বাইরে থাকবে। ওটা নিয়ে পরবর্তী সময়ে অন্য কোনো সময় লেখা হবে।

আমরা এখন একটু সুপারভাইজড লার্নিং নিয়ে কথা বলি। সুপারভাইজ মানে হচ্ছে কাউকে কোনো কিছু করতে শিখিয়ে দেওয়া, দেখিয়ে দেওয়া – যাতে সে পরবর্তী সময়ে নিজে নিজে কাজটি করতে পারে। মেশিন লার্নিংয়ের ক্ষেত্রেও, সুপারভাইজড লার্নিং অ্যালগরিদম হচ্ছে সেই সব অ্যালগরিদম, যেগুলো ব্যবহার করে আমরা কম্পিউটারকে মানুষের মত চিন্তা করা শেখানোর জন্য ট্রেনিং দেওয়ার জন্য আমাদের যেটি করতে হবে সেটি হচ্ছে কম্পিউটারকে **পর্যাপ্তসংখ্যক উদাহরণ দিতে হবে।**

একটি খুব সহজ উদাহরণ দিই। ধরা যাক, আপনি জীবনে কখনো জিরাফ দেখেননি। আমি যদি আপনাকে জিরাফ চেনাতে চাই, তাহলে কী করব? কী করাটা সবচেয়ে সহজ হবে? আমি, কিংবা আরো দশজন সাধারণ মানুষ যেটি করবে সেটি হলো আপনাকে চিড়িয়াখানায় নিয়ে গিয়ে জিরাফ দেখাবে, তাই না? জিরাফ দেখিয়ে বলবে, ‘দেখুন ভাই, এটি হলো জিরাফ।’ এর বিশাল লম্বা গলা থাকে। গায়ে হলুদ-বাদামি ফুটকি। এদের পাঞ্জলোও বেশ লম্বা। পেছনে লেজ আছে। মাথার ওপরে কান ছাড়াও দুটি ছোটো শিঙের মতো জিনিস আছে।’ তখন আপনি কী করবেন? আপনি ভালো করে জিরাফটি দেখবেন, ওর চেহারা আর ওর যে বৈশিষ্ট্যগুলো আছে, সেইগুলো মনে রাখার চেষ্টা করবেন।

এরপর আপনাকে কেউ যদি একটি প্রাণীর ছবি দেখিয়ে শনাক্ত করতে বলে যে, সেটি জিরাফ কি না, তখন কিন্তু আপনি অবশ্যই সেটি করতে পারবেন। কীভাবে? আপনি মনে মনে মিলিয়ে দেখবেন যে, এই নতুন প্রাণীর শরীরে জিরাফের বৈশিষ্ট্যগুলো আছে কি না। যদি থেকে থাকে, তখন আপনি বলবেন এটি জিরাফ, আর যদি না থাকে, তাহলে বলবেন এটি জিরাফ নয়।

আরেকটি উদাহরণ দিই। মনে করুন, আপনি পিংজা খেতে পছন্দ করেন। পিংজা হাটের পিংজা আপনার খুবই পছন্দের খাদ্য। এখন, যেহেতু আপনি নিয়মিতই সেখানে পিংজা খেতে যান, তাই পিংজা হাটের কিছু সাইজের পিংজার দাম আপনি আগে থেকেই জানেন (টেবিল 2.1-এ দেখানো হলো)।

অধ্যায় ২ : বিভিন্ন ধরনের লার্নিং ও অন্যান্য

পিংজার সাইজ (ইঞ্চিতে)	পিংজার দাম (টাকায়)
6"	399/-
9"	699/-
12"	945/-
15"	1215/-

টেবিল 2.1

এখন আপনার কোনো বন্ধু যদি আপনাকে জিজ্ঞাসা করে – ‘দোস্ত, নতুন একটি পিংজা এসেছে শুনলাম, 14 ইঞ্চি পিংজা। পুরো চিকেন আর চিজে মাখামাখি। পিংজার দাম কত হতে পারে আন্দাজ কর তো?’ তাহলে আপনি কিন্তু খুব সহজেই আন্দাজ করে বলতে পারবেন যে 1150/- টাকার মতো হবে দাম। এটি কীভাবে বলবেন?

আপনি নিজের মনের অজান্তেই চিন্তা করবেন যে 15 ইঞ্চি পিংজার দাম আপনি জানেন 1215/- টাকা, আর 12 ইঞ্চি পিংজার দামও আপনি জানেন 945/- টাকা। তার মানে 14 ইঞ্চি পিংজার দাম এ দুটির মাঝামাঝি কিছু একটা হবে। যেহেতু 14 ইঞ্চি পিংজা আর 15 ইঞ্চি পিংজার সাইজ বেশ কাছাকাছি, সুতরাং এদের দামও কাছাকাছি হবে। তাই 1150/- টাকার কাছাকাছি কিছু একটা হবে দাম। মজার ব্যাপার হচ্ছে, এই চিন্তাটি আপনি করবেন আপনার মনের অজান্তেই এবং এত দ্রুত করবেন যে আপনি টেরই পাবেন না কীভাবে আপনি দাম আন্দাজ করলেন। একইভাবে আপনাকে যদি 16 ইঞ্চি পিংজার দাম আন্দাজ করতে বলা হয় আপনি সেটিও করতে পারবেন (এই উদাহরণটি নিয়ে আমরা সামনে আরো বিস্তারিত আলোচনা করব)।

ওপরের যে দুটি উদাহরণ দেখানো হলো, দুটিই সুপারভাইজড লার্নিংয়ের উদাহরণ। প্রথম উদাহরণে প্রথমে জিরাফ দেখিয়ে আর জিরাফের বৈশিষ্ট্য বলে আপনার মন্ত্রিককে ট্রেনিং দেওয়া হলো জিরাফ চেনার জন্য। এরপর আপনাকে যত প্রাণীর ছবিই দেওয়া হোক না কেন, আপনি আলাদা করতে পারবেন যে কোনটি জিরাফ আর কোনটি জিরাফ নয়। এ ধরনের সমস্যাগুলোকে আমরা বলি **ক্লাসিফিকেশন প্রবলেম (Classification Problem)**।

আবার দ্বিতীয় উদাহরণে আপনাকে প্রথমে বিভিন্ন পিংজার সাইজ ও তাদের দাম বলে দেওয়া হলো, আর তারপরে সেই তথ্যের ওপরে ভিত্তি করে আপনাকে আন্দাজ করতে বলা হলো অন্য আরেক সাইজের পিংজার দাম যেটি আপনার অজানা। এ ধরনের সমস্যাগুলোকে আমরা বলি **রিগ্রেশন প্রবলেম (Regression Problem)**।

আমরা রিগ্রেশন ও ক্লাসিফিকেশন, দুই ধরনের প্রবলেম নিয়েই সামনে বিস্তারিত আলোচনা করব এবং এইসব সমস্যা সমাধান করার জন্য বিভিন্ন অ্যালগরিদম দেখব।

এবার আগি আনসুপারভাইজড লার্নিংয়ে আমরা যেরকম ডেটার পাশাপাশি সঠিক উত্তরটিও দিয়ে দিতাম, আনসুপারভাইজড লার্নিংয়ে সেরকম ধরনের কোনো ট্রেনিং দিয়ে নেওয়া হবে না ফল্পিডটারকে। তাকে ট্রেনিং-এর জন্য শুধু ডেটা দিয়ে দেওয়া হবে, সঠিক উত্তর নয়। সে নিজের মতো করে ঢেটা করবে ওই ডেটার মধ্যে প্যাটার্ন বা বিভিন্ন ডেটার মধ্যে সামঞ্জস্য খুঁজে বের করতে। সেই সামঞ্জস্যের ওপরে ভিত্তি করে সে সবগুলো ডেটা বিভিন্ন গ্রুপে সাজাবে। এই ধরনের সমস্যাগুলোকে আমরা বলি **ক্লাস্টারিং প্রবলেম** (Clustering Problem), যা কিনা আনসুপারভাইজড লার্নিংয়ের সবচেয়ে জনপ্রিয় উদাহরণ। কিন্তু এর বাইরেও আরো বেশ কিছু ধরনের সমস্যা আছে আনসুপারভাইজড লার্নিংয়ের ক্ষেত্রে। আমাদের এই বইতে আমরা অবশ্য ক্লাস্টারিং প্রবলম নিয়েই বিষয় আলোচনা করব।

পরিচ্ছেদ ২.২ : ফিচার (Feature)

এখন আমাদের আরো বি মেশিন লার্নিংয়ের ধারণার ব্যাপারে একটু জেনে নিতে হবে। এগুলো একেবারে গোড়ার দিকে কিছু ধারণা এবং এগুলো মেশিন লার্নিং শুরু করার জন্য জানা খুবই দরকার। এই ধারণাগুলে যদি পরিষ্কার না থাকে, তাহলে পরবর্তী সময়ে বাকি বিষয়গুলো বুঝতে অসুবিধা হবে।

টেবিল 2.2 ভালো করে লক্ষ করি। এই টেবিলে কয়েকটি ভিন্ন ভিন্ন প্রাণীর ছবি থেকে ওই প্রাণী সম্পর্কে কিছু ভিন্ন ভিন্ন ডেটা দেওয়া আছে। এই টেবিল ব্যবহার করে আমরা আমাদের পরবর্তী কনসেপ্টগুলো শিখব। ডেটাটি কাল্পনিক, বুঝতেই পারছেন। শুধু আপনাদের বোঝানোর উদ্দেশ্যে ব্যবহার করা হচ্ছে। এর সঙ্গে কেউ বাস্তবের সামঞ্জস্য খুঁজতে যাবেন না যেন।

ছবি নম্বর	লেজের দৈর্ঘ্য (সে.মি.)	গলার দৈর্ঘ্য (সে.মি.)	শিঙের মতো বস্তু কি আছে?	প্রাণীটি কি জিরাফ?
1	5	8	হ্যাঁ	হ্যাঁ
2	2	3	না	না
3	1	2	না	না
4	0	2	না	না
5	5.5	7.5	হ্যাঁ	হ্যাঁ

টেবিল 2.2

একটি বিষয় চিন্তা করে দেখুন, আমরা যখন কাউকে নতুন কোনো কিছু চেনাই, তখন তাকে সেই বস্তুটির সবচেয়ে অনন্য বা ইউনিক (unique) বৈশিষ্ট্যগুলোর কথা বলি, যা দিয়ে সে সহজেই তাকে আলাদা করতে পারবে আর দশটি অন্য কিংবা একই রকম জিনিস থেকে।

যেমন ধরা যাক, আপনার ক্লাসে দুজন আছে আরিফ নামে। একজন চশমা পরে, আরেকজন পরে না। এখন, একদিন ক্লাসে গিয়ে আপনি আরিফকে খুঁজছেন কোনো একটি দরকারে। তাকে ক্লাসে না পেয়ে আপনি আপনার কোনো এক বন্ধুকে জিগ্যেস করলেন, ‘দোষ্ট, আরিফ কোথায় রে?’ তখন সে স্বাভাবিকভাবেই আপনাকে জিগ্যেস করবে, ‘কোন আরিফ? ক্লাসে তো দুজন আরিফ আছে।’ তখন আপনি বলবেন হয় চশমা-পরা আরিফ, কিংবা চশমা-ছাড়া আরিফ।

এই যে আপনি দুজন আরিফকে তাদের একটি বৈশিষ্ট্য দিয়ে আলাদা করলেন, এই বৈশিষ্ট্যটিই হলো ফিচার (Feature)। তার মানে ফিচার হচ্ছে সেই সমস্ত অন্য বৈশিষ্ট্য, যা কোনো কিছুকে আর দশটি সামঞ্জস্যপূর্ণ জিনিস থেকেও সহজে আলাদা করে ফেলতে পারে।

এখন, আমরা যদি কম্পিউটারকে কোনো কিছু চেনাতে চাই, তাহলে কী করব? ধরুন, আমরা যদি কম্পিউটারকে একটি জিরাফ চেনাতে চাই, তাহলে, প্রথমে আমরা কী করব? কম্পিউটারকে ট্রেনিং দেব। কীভাবে ট্রেনিং দেব? তাকে বিভিন্ন প্রাণীর ছবি ইনপুট হিসেবে দেব এবং সেই সঙ্গে সেই সব প্রাণীর লেজের দৈর্ঘ্য, গলার দৈর্ঘ্য, মাথার ওপরে শিং আছে কি নেই – এসব ডেটাও ইনপুট হিসেবে দেব (টেবিল 2.2-এর মতো)। অথবা তাদের এমনভাবে নির্দেশ দেব, যাতে তারা সেই ফিচারগুলো নিজেরাই ছবি থেকে বিভিন্ন অ্যালগরিদম ব্যবহার করে বের করে নিতে পারে।

এখন ধরা যাক, যদি সুপারভাইজড লার্নিং হয়, তাহলে তাদেরকে এইসব ফিচার ডেটার সঙ্গে সঙ্গে এটিও বলে দেওয়া হবে কোনটি জিরাফ, আর কোনটি জিরাফ নয়। তখন, আমাদের ডেটাকে বলা হবে লেবেলড ডেটা (Labelled Data)। লেবেলড ডেটাতে আউটপুটের প্রকৃত মান বলে দেওয়া থাকে। টেবিল 2.2 থেকে যদি বলতে যাই, শেষের কলামটি হচ্ছে প্রাণীর লেবেল (অর্থাৎ সে জিরাফ, নাকি জিরাফ নয়), আর তাঁর আগের যাবতীয় সব কলাম হচ্ছে ফিচার ডেটা।

কিন্তু, যদি আনসুপারভাইজড লার্নিং হয়, তাহলে আর কম্পিউটারকে বলে দেওয়া হবে না যে, কোনটি জিরাফ আর কোনটি নয়। কম্পিউটার নিজে নিজেই ফিচার ডেটার মধ্যে একটি সামঞ্জস্য বের করার চেষ্টা করবে। তখন আমাদের ডেটাকে বলা হবে আনলেবেলড ডেটা (Unlabelled Data) অর্থাৎ ডেটাতে আউটপুটের প্রকৃত মান বলে দেওয়া থাকবে না। টেবিল 2.2 থেকে যদি বলতে যাই, টেবিলের শেষের কলামটি দেওয়া না থাকলেই এটি হয়ে যাবে আনলেবেলড ডেটা।

এইসব ডেটা (লেবেলড/আনলেবেলড) কম্পিউটার তার ডেটাবেজে সংরক্ষণ করে রাখবে। এরপরে কম্পিউটারকে আমরা নতুন কতগুলো ভিন্ন ভিন্ন প্রাণীর ছবি দেব, যে ছবিগুলো আগে সে

দেখেনি। সে ছবিগুলোর মধ্যে কোনগুলো জিরাফ আর কোনগুলো জিরাফ নয়, কম্পিউটার সেটি চিহ্নিত করবে।

এটি করার জন্য সে প্রথমে যা করবে, সেটি হলো নতুন ছবির প্রাণীটির গলা, লেজ, মাথা – এগুলো শনাক্ত করবে। তারপরে গলার দৈর্ঘ্য (নিউমেরিক ডেটা), লেজের দৈর্ঘ্য (নিউমেরিক ডেটা), মাথায় শিঙের মতো বস্তু আছে কি নেই (হ্যাঁ/না, বাইনারি ডেটা হতে পারে) ইত্যাদি বের করে নেবে।

এরপরে নিজের ডেটাবেজে থাকা তথ্যের সঙ্গে এই নতুন পাওয়া তথ্য মিলিয়ে দেখবে যে দুটো মোটামুটি কাছাকাছি হয় কি না। যদি হয়, তাহলে ছবিটাকে জিরাফ বলে শনাক্ত করবে, নচেৎ নয়।

আগেই বলেছি, আমাদের কাছে থাকা সমস্ত ডেটার মধ্যে যেসব ডেটা কম্পিউটারকে সাহায্য করছে কোনটি জিরাফ আর কোনটি জিরাফ নয় সেটি শনাক্ত করতে (লেজের দৈর্ঘ্য, গলার দৈর্ঘ্য, শিঙের উপস্থিতি ইত্যাদি), সেগুলোকেই আমরা বলব ফিচার। আর ইনপুট থেকে ফিচার ডেটা হিসাব করে বের করে নেওয়াকে বলে ফিচার এক্সট্রাকশন (Feature Extraction)। তার মানে, টেবিল 2.2 থেকে যদি বলি, লেজের দৈর্ঘ্য, গলার দৈর্ঘ্য এবং শিঙের মতো বস্তুর উপস্থিতি– এই তিনটি তথ্য ফিচার ডেটা হিসেবে কাজ করবে।

ফিচার ডেটা নিয়ে হিসাব করে, অ্যানালাইসিস করে কম্পিউটার যে ফলাফল দেবে সেটাই হবে আমাদের কঙ্গিষ্ঠ আউটপুট। এই আউটপুটকে আবার **রেসপন্স ভ্যারিয়েবল (Response variable)**-ও বলে।

কম্পিউটারকে ভিন্ন ভিন্ন কতগুলো প্রাণীর ছবি দিয়ে সেগুলোর মধ্যে কোনগুলো জিরাফ আর কোনগুলো জিরাফ নয়, সেগুলো শনাক্ত করতে দিলে কম্পিউটার বিভিন্ন অ্যানালাইসিস করে প্রতিটি ছবির জন্য ছবিটি জিরাফের হলে ক্রিনে ‘হ্যাঁ’ লেখা প্রিন্ট করবে, আর জিরাফ না হলে ‘না’ লেখা প্রিন্ট করবে। এটাই হলো **রেসপন্স ভ্যারিয়েবল বা আউটপুট**। যেমন টেবিল 2.2-এ, একেবারের ডান কলামের মানগুলো হলো রেসপন্স ভ্যারিয়েবলের মান।

তাহলে **রেসপন্স ভ্যারিয়েবল আর লেবেল**-এর মধ্যে পার্থক্য কী? **রেসপন্স ভ্যারিয়েবল** হচ্ছে আমার অ্যালগরিদম যে মান আউটপুট হিসেবে দিচ্ছে সেটি। আর **লেবেল** হচ্ছে সেই মান দ্বারা আসলে কী বোঝানো হচ্ছে সেটি।

ধরা যাক, কম্পিউটার যদি জিরাফের ছবি পায়, তাহলে সে 1 আউটপুট দেবে, আর না পেলে 0 আউটপুট দেবে। এই 1 ও 0 হচ্ছে **রেসপন্স ভ্যারিয়েবল** যেটি অ্যালগরিদম আমাকে সরাসরি দিচ্ছে। আর 1 মানে হচ্ছে ‘জিরাফ’, এখানে এই জিরাফটি হলো লেবেল।

একইভাবে 0 মানে হচ্ছে ‘জিরাফ নয়’, এখানে ‘জিরাফ নয়’ এটি হচ্ছে লেবেল।



পরিচ্ছেদ ২.৩ : ট্রেনিং, টেস্ট ও ভ্যালিডেশন ডেটা (Training, Test and Validation Data)

যে ডেটা দিয়ে আমরা কম্পিউটারকে ট্রেনিং দেব, সে ডেটাকে বলে 'ট্রেনিং ডেটা (Training Data)'। আর যে ডেটা দিয়ে আমরা কম্পিউটারের ট্রেনিং শেষ হওয়ার পর তার পারফরম্যান্স অ্যানালাইসিস করব যে তার কাজে সে কতটুকু ভালো করছে, সেই ডেটাকে আমরা বলব 'টেস্ট ডেটা (Test Data)'। সাধারণত পুরো ডেটাসেটকে দুই ভাগে ভাগ করে একটি ভাগকে ট্রেনিং ডেটা, অন্যটিকে টেস্ট ডেটা এভাবে ব্যবহার করা হয় (Validation Data-এর ব্যাপারে পরে বর্ণনা করছি)। যেমন, আমার কাছে 100 ডেটা থাকলে 60টি ডেটা দিয়ে আমি ট্রেনিং দিতে পারি এবং বাকি 40টি ডেটা দিয়ে আমি টেস্ট করতে পারি। এই অনুপাতটি আমার কাজের প্রকারভেদের ওপরে নির্ভর করবে। কোনটি ট্রেনিং সেটে যাবে, আর কোনটি টেস্ট সেটে, এগুলো দৈবভাবে বা র্যানডমলি নির্বাচন করা হয়।

এই ট্রেনিং আর টেস্ট ডেটার ব্যাপারটি আরেকটু খোলাসা করে বলি। ধরা যাক, সুপারভাইজড লার্নিং নিয়ে আমরা কাজ করছি। তার মানে আমাদের ডেটা অবশ্যই হবে লেবেলড ডেটা, অর্থাৎ, রেসপন্স ভ্যারিয়েবলের আসল মান কিংবা সোজা কথায় আউটপুট কী হবে তা আমাদের আগে থেকেই জানা। এখন ধরা যাক, আমাদের কাছে 100টি ছবির লেবেলড ডেটা আছে। এর মধ্যে 50টি আমাদের ট্রেনিং ডেটা, আর বাকি 50টি টেস্ট ডেটা।

এখন ট্রেনিং ডেটা দিয়ে কম্পিউটার মূলত যা করবে সেটি হলো, সে মনে রাখার চেষ্টা করবে ফিচার ডেটার কোনটির মান ঠিক কী রকম হলে সেটি জিরাফ হচ্ছে, আর কী রকম হলে সেটি জিরাফ বাদে অন্য প্রাণী হচ্ছে। এরপর তাকে যখন টেস্ট ডেটা দেওয়া হবে, তখন সে সেই টেস্ট ডেটার ওপরে ফিচার ডেটার কোনটির মান ঠিক কী রকম হলে সেটি জিরাফ হচ্ছে, আর কী রকম হলে সেটি জিরাফ বাদে অন্য প্রাণী হচ্ছে' – এই জ্ঞানটুকু, যা সে ট্রেনিং ডেটা অ্যানালাইসিস করে পেয়েছে, প্রয়োগ করে দেখবে। অ্যানালাইসিস প্রয়োগ করার পরে সে যে রেসপন্স ভ্যারিয়েবল তৈরি করবে সেটি যদি টেস্ট ডেটার অরিজিনাল রেসপন্স ভ্যারিয়েবলের মানের সঙ্গে মিলে যায়, তাহলেই বুঝব আমাদের মেশিন শিখতে পেরেছে।

দুর্বোধ্য মনে হচ্ছে, তাই না?

তাহলে সময় নষ্ট না করে উদাহরণে যাই। টেবিল 2.2 দিয়েই বর্ণনা করি। টেবিলে দেখা যাচ্ছে, 5টি ছবির ডেটা দেওয়া আছে। আগেই বলেছি, আমরা সুপারভাইজড লার্নিং ব্যবহার করছি। তাই, আমাদের ডেটা হবে লেবেলড ডেটা, মানে রেসপন্স ভ্যারিয়েবলের আসল ভ্যালু দেওয়া থাকবে (ডান দিকের কলাম)।

এখন ধরা যাক, এই ৫টি ডেটার মধ্যে প্রথম ৩টি আমাদের ট্রেনিং, পরের ২টি টেস্ট ডেটা। ধরি, প্রথম ৩টি ট্রেনিং ডেটা থেকে কম্পিউটার হিসাবকিতাব করে বের করল যে লেজের দৈর্ঘ্য ৫ সেমি বা এর কাছাকাছি হলে, গলার দৈর্ঘ্য ৪ সেমি বা এর কাছাকাছি হলে এবং শিঙের মতো বস্তু উপস্থিত থাকলে সেই প্রাণী জিরাফ, আর নাহলে সেটি অন্য প্রাণী। জিরাফ হতে হলে এই ৩টি রিকোয়ারমেন্ট/কন্ডিশনই সত্য হতে হবে।

এখন যদি কম্পিউটারকে টেস্ট ডেটা দেওয়া হয় (৪ ও ৫ নম্বর ছবির ডেটা) তাহলে কম্পিউটার কী আউটপুট দেবে, বলতে পারবে? ৪ নম্বর ছবির জন্য কম্পিউটার অবশ্যই 'না' আউটপুট দেবে, কারণ শিঙের মতো বস্তু নেই, আর গলা-লেজের দৈর্ঘ্যও আমাদের রিকোয়ারমেন্টের কাছাকাছি নেই। কিন্তু ৫ নম্বর ছবির জন্য কম্পিউটার অবশ্যই 'হ্যাঁ' আউটপুট দেবে, কেননা এটি আমাদের জিরাফ হওয়ার ৩টি রিকোয়ারমেন্টই পূরণ করে।

এখন তাহলে ৪ ও ৫-এর জন্য কম্পিউটার আউটপুট দেবে যথাক্রমে 'না' ও 'হ্যাঁ' – যা কিনা আমাদের কাছে থাকা অরিজিনাল আউটপুট ভ্যালুর সঙ্গে মিলে যাচ্ছে (চাটে মিলিয়ে দেখুন)।

সুতরাং আমাদের কম্পিউটারের শেখার একিউরেসি আমরা 100% বলতে পারি এই ক্ষেত্রে (বাস্তবে এটি হয় না বললেই চলে কারণ আমরা বাস্তবে আরো অনেক ডেটা নিয়ে কাজ করি, ফলাফল ভুল হওয়ার আশঙ্কাও অনেক বেশি থাকে সে ক্ষেত্রে)।

একটি কথা খুব ভালোমতো বোঝার চেষ্টা করতে হবে, টেস্ট ডেটা নিয়ে কম্পিউটার যখন কাজ করে, তখন সে কিন্তু যেই রেসপন্স ভ্যারিয়েবলের ভ্যালু দেওয়া আছে, সেইটাই আউটপুট হিসেবে দিয়ে দেয় না। সে ট্রেনিং ডেটা থেকে পাওয়া রিকোয়ারমেন্ট অনুযায়ী অ্যানালাইসিস করে প্রতিটি টেস্ট ডেটার জন্য একটি করে আউটপুট দেয়।

এরপর সেটি আমাদের কাছে থাকা টেস্ট ডেটার অরিজিনাল আউটপুটের সঙ্গে মিলিয়ে দেখা হয় যে কয়টি মিলল আর কয়টি ভুল করল কম্পিউটার। এরপর সেই হিসাবে তখন একিউরেসি মাপা হয়, যেভাবে আমরা একটু আগে করে দেখালাম।

*output
yes/no* ↗
যা-ই হোক, এতক্ষণ আমরা দেখলাম ট্রেনিং ডেটা ও টেস্ট ডেটার ধারণাটুকু। এখন থেকে এতটুকু বোঝা যাচ্ছে যে, মডেল একেবারে নিজেকে ট্রেনিং দিয়ে পুরোপুরি তৈরি করে না ফেলা পর্যন্ত টেস্ট ডেটার দেখা পায় না, তাই না?

আমরা যদি আমাদের নিজেদের জীবনের সঙ্গে মেলাতে যাই, ধরুন আপনি কোনো পরীক্ষা (ধরুন GRE পরীক্ষা) দেবেন। পরীক্ষা দেওয়ার আগে আপনি অবশ্যই দু-তিন মাস খুব ভালোমতো পড়াশোনা করবেন, নিজেকে প্রস্তুত করবেন, তাই না? কী কী ধরনের প্রশ্ন আসতে পারে, সব ধরনের প্রশ্ন যাচাইবাছাই করে, সমস্ত টপিক ঠিকমতো শেষ করে নিজেকে প্রস্তুত করে তবেই তো পরীক্ষা দিতে যাবেন।

নিজেকে এই যে প্রস্তুত করার ব্যাপারটি হলো অনেকটা ট্রেনিং ডেটা দিয়ে নিজেকে ট্রেনিং দেওয়ার মতো।

আর যখন সমস্ত প্রস্তুতি শেষে পরীক্ষা দিতে যাবেন এবং পরীক্ষার রেজাল্ট আপনাকে বলে দেবে আপনার ক্ষেত্রে কত হয়েছে, সেটাই কিন্তু বলে দেবে আপনার প্রস্তুতি আসলেই কতটুকু ভালো ছিল। পরীক্ষার প্রশ্নই হচ্ছে আপনার টেস্ট ডেটা। আপনি কি পরীক্ষার প্রশ্ন কখনো আগেভাগে দেখে ফেলতে পারেন? কখনোই না, তাই না? ঠিক সেরকম, টেস্ট ডেটাও কখনো মডেলকে দেখতে দেওয়া হয় না। আগে সে সম্পূর্ণ প্রস্তুত হয়, এর পরে টেস্ট ডেটা হাতে পায়।

সবশেষে বলি, ভ্যালিডেশন ডেটার কথা। ভ্যালিডেশন ডেটা আপনার ট্রেনিং ডেটারই একটি অংশ, যেটি আপনার মডেল কর্তৃক ভালো হয়েছে, আরো কর্তৃক ভালো করা দরকার, মডেলের বিভিন্ন প্যারামিটারগুলো টিউন করার কাজ ইত্যাদি করতে সাহায্য করে।

এটি অনেকটা মূল পরীক্ষার পূর্বে, পরীক্ষার প্রস্তুতি হিসেবে মডেল টেস্ট দেওয়ার মতো। আপনি পড়াশোনা করে নিজেকে প্রস্তুত করলেন মূল পরীক্ষার জন্য (ট্রেনিং ডেটা), এরপর মূল পরীক্ষার আগে নিজের দুর্বলতা ঝালাই করে নেওয়ার জন্য মূল পরীক্ষার প্রশ্নের আদলেই কয়েকটি মডেল টেস্ট দিলেন (ভ্যালিডেশন ডেটা) এবং সেখান থেকে নিজের দুর্বলতা সব বের করে নিজেকে একেবারে প্রস্তুত করে তবেই ফাইনাল পরীক্ষা দিতে গেলেন (টেস্ট ডেটা)।

অর্থাৎ বলা চলে, ভ্যালিডেশন ডেটা হচ্ছে টেস্ট ডেটার মতোই এক ধরনের ডেটা, যা মডেল আগেভাগে ট্রেনিংয়ের সময় দেখতে পারে না। ট্রেনিং শেষ হওয়ার পরে নিজেকে কিছুটা যাচাই করার জন্য এটি ব্যবহার করে নিজেকে ঠিকমতো ‘টিউন’ করে নিতে পারে, যাতে সে টেস্ট ডেটার ওপরে ভালো ফলাফল করে। মূলত মডেলকে ভালো পারফরম করতে ঠিকমতো আপডেট করাই হচ্ছে এই ভ্যালিডেশন ডেটার উদ্দেশ্য।

আমরা তাহলে যেটি করতে পারি, আমাদের গোটা ডেটাসেটকে এখন তিন ভাগ করতে পারি –

- ট্রেনিং ডেটা,
- ভ্যালিডেশন ডেটা ও
- টেস্ট ডেটা

ধরুন আপনার কাছে যদি 100টি ডেটা থাকে, আপনি 60টি দিয়ে ট্রেনিং দিলেন, 20টি দিয়ে ভ্যালিডেশন করলেন এবং বাকি 20টি একেবারে সব প্রস্তুতি শেষে টেস্ট করার জন্য রেখে দিলেন।

মোটামুটি এই হচ্ছে ট্রেনিং, ভ্যালিডেশন ও টেস্ট ডেটার ধারণা। আশা করি, সবাই বুঝতে পেরেছেন।

পরিচেদ ২.৪ : ক্রস ভ্যালিডেশন (Cross Validation)

ক্রস ভ্যালিডেশন (Cross Validation) মেশিন লার্নিংয়ের অত্যন্ত গুরুত্বপূর্ণ একটি ধারণা। এর সঙ্গে কেউ কিন্তু আবার ভ্যালিডেশন ডেটাকে গুলিয়ে ফেলবেন না। দুটি একেবারে ভিন্ন ধারণা। এটির বিস্তারিত উদাহরণে যাওয়ার আগে দুটো ছোটো গল্প বলি।

ধরা যাক, ইশতিয়াক স্কুলে পড়ে। সামনে তার অঙ্ক পরীক্ষা। অঙ্ক বইতে মোট অধ্যায় আছে 10টি। পরীক্ষার আগে স্যার বলে দিলেন যে প্রথম 5 অধ্যায়ের ওপরে পরীক্ষা হবে, তাই ইশতিয়াক বইয়ের প্রথম 5 অধ্যায় পড়ে পরীক্ষা দিতে গেল। এখন ইশতিয়াক যদি পরীক্ষায় দিতে গিয়ে দেখে, পুরো বই থেকেই প্রশ্ন হয়েছে, শুধু প্রথম 5 অধ্যায় থেকে নয়; তাহলে ইশতিয়াক পরীক্ষায় খারাপ করবে। তাই না? এখন, তার এই পরীক্ষায় খারাপ করার ওপরে ভিত্তি করে কেউ যদি বলে যে ইশতিয়াক খারাপ ছাত্র, সে পড়াশোনা করে না একদমই, তাহলে কি ঠিক হবে?

আবার ধরা যাক, ইশতিয়াকের স্যার বললেন, পরীক্ষা হবে পুরো বইয়ের ওপরে, অর্থাৎ 10টি অধ্যায়ই সিলেবাসে থাকবে। ইশতিয়াক এবারে পুরো বইটা পড়ে পরীক্ষার হলে গিয়ে দেখল, শুধু প্রথম 5 অধ্যায় থেকেই সব প্রশ্ন এসেছে, পরের 5 অধ্যায় থেকে কিছুই আসেনি। এই পরীক্ষায় ইশতিয়াক খুব ভালো নম্বর পেল। কিন্তু, এই পরীক্ষায় কি ইশতিয়াকের সঠিক মূল্যায়ন হলো? আমরা কি এই শুধু ফলাফলের ওপর ভিত্তি করেই তাকে ভালো ছাত্র বলতে পারব?

ওপরের দুটো ক্ষেত্রের কোনোটিতেই আমরা সঠিকভাবে ইশতিয়াককে মূল্যায়ন করিনি। সঠিকভাবে মূল্যায়ন তখনই হতো, যদি যতগুলো অধ্যায় পরীক্ষায় দেওয়া হয়েছে সবগুলো থেকেই ইশতিয়াককে প্রশ্ন করা হতো এবং যতটুকু সিলেবাস তার মধ্যে থেকেই প্রশ্ন করা হতো।

এখন আসি ক্রস ভ্যালিডেশনের ক্ষেত্রে। আমরা ইতিমধ্যেই দেখেছি যে, আমরা যে ডেটাসেট নিই তার কিছু অংশ আমরা ট্রেনিং ডেটা হিসেবে ব্যবহার করি, আর বাকি অংশ টেস্ট ডেটা হিসেবে। এখন, ধরা যাক, আমার কাছে 100টি ডেটা পয়েন্ট আছে। এই ডেটা সেটকে, 5টি সমান ভাগে ভাগ করব। তাহলে আমার প্রতি ভাগে ডেটা পয়েন্ট থাকবে 20টি করে, ঠিক? টেবিল 2.4.1-এর দিকে তাকালে বোৰা যাবে ভালোমতো। আমাদের ডেটাসেটের এই পাঁচটি সাবসেটকে আমরা যথাক্রমে D1, D2, D3, D4 ও D5 নাম দিয়ে চিহ্নিত করে দিই।

D1	D2	D3	D4	D5
20	20	20	20	20

টেবিল : 2.4.1

এখন আমরা যদি এতক্ষণ যেভাবে ট্রেনিং ও টেস্ট ডেটার বিষয়টি বুঝে এলাম, সেইভাবে ডেটাসেটটিকে ভাগ করি, তাহলে ধরি, D1, D2, D3, D4 গেল ট্রেনিং সেট-এ এবং D5 গেল টেস্ট

সেট-এ। এটি আমরা র্যানডমলি সিদ্ধান্ত নিলাম। আমরা শুধু D1, D2 ও D3-কেও ট্রেনিং সেট-এ দিয়ে D4 ও D5-কে টেস্ট সেটে দিতে পারতাম। এটি কীভাবে ভাগ হবে তার স্বাধীনতা পুরোপুরি প্রোগ্রামারের কাছে থাকবে।

এখন, আমরা যদি D1, D2, D3, D4 দিয়ে ট্রেইন করে D5 দিয়ে টেস্ট করে যে মডেল ইভ্যালুয়েশন পাব (মডেল ইভ্যালুয়েশন মানে হচ্ছে আমাদের মেশিন লার্নিং অ্যালগরিদম কর্তৃক ভালো পারফরম করল, বা কর্তৃক ভুলভাষ্টি করল সেটি বের করা), সেটাকে কি আমাদের পুরোপুরি সঠিক বলে ধরে নেওয়াটা ঠিক হবে? এটি অনেকটা ইশতিয়াকের প্রথম উদাহরণটির মতো হয়ে যাচ্ছে না, যেখানে এমন জিনিস পরীক্ষায় চলে এসেছে, যেটি সিলেবাসে ছিল না, যার ফলে ইশতিয়াক সেটি পড়েইনি? সাধারণত হ্যাঁ, আমরা টেস্ট ডেটা হিসেবে এমন ডেটা ব্যবহার করি, যেটি আমাদের মেশিন লার্নিং অ্যালগরিদম আগে কখনো দেখেনি। কিন্তু এই ক্ষেত্রে শুধু একটি অচেনা ডেটা সেট দিয়ে ইভ্যালুয়েশন করেই, আমাদের মডেল ভালো না খারাপ, সেই সিদ্ধান্তে উপনীত হওয়াটা ঠিক হবে না। আমাদের আরো একটু বেশি পরীক্ষা-নিরীক্ষা করতে হবে।

আবার, আমরা যদি, D1, D2, D3, D4, D5 সবগুলো ডেটা দিয়ে প্রথমেই একবারে ট্রেইন করে ফেলি এবং তারপর শুধু D1, D2, D3, D4 কিংবা D5 এদের যে-কোনো একটি দিয়ে টেস্ট করি, তাহলে সেটাও ভুল হবে। কেননা, আমাদের টেস্ট করতে হয় এমন ডেটা দিয়ে, যেটি আমাদের মেশিন লার্নিং অ্যালগরিদম আগে কখনো দেখেনি। কিন্তু এই ক্ষেত্রে মেশিন সমস্ত ডেটাই দেখে ফেলেছে।

আমাদের ট্রেনিং ও টেস্ট ডেটায় গোটা ডেটাসেট ভাগ করার সময় লক্ষ্য এটি থাকে যে আমরা ট্রেনিং ডেটা যত বেশি পারা যায় নেব, যাতে করে মেশিন ভালোভাবে শিখতে পারে; এবং টেস্ট ডেটাও যত বেশি পারা যায় নেব, যাতে আমরা ভালোভাবে টেস্ট করতে পারি। আমরা যদি মাত্র 2-3টি ডেটা নিয়ে টেস্ট করে ভালো ফলাফল পেয়েই বলে দিই যে আমাদের মেশিন শিখে গেছে, তাহলে কি ব্যাপারটা ঠিক হবে? মোটেও নয়। আমাদের যত ভিন্ন ভিন্ন উপায়ে, যত ভালোভাবে পারা যায় টেস্ট করতে হবে, যাতে মেশিনের শেখায় কোনো ধরনের ফাঁকফোকর না থেকে যায়।

আর এখান থেকেই ক্রস ভ্যালিডেশনের উৎপত্তি। ক্রস ভ্যালিডেশনে যেটি হয় যে, আমরা প্রথমেই ডেটাসেটকে K-সংখ্যক সমান ভাগে ভাগ করে ফেলি। ভাগটা হয় র্যানডমলি। এটাকে K-fold Cross Validationও বলা হয়। এরপর এই K-সংখ্যক ভাগ থেকে প্রতিবার (K-1)-সংখ্যক ভাগ দিয়ে মেশিন লার্নিং অ্যালগরিদমকে ট্রেইন করি এবং বাকি ভাগটি দিয়ে টেস্ট করি। তাই, যেটি হয় যে, আমাদের K-সংখ্যক ভাগের ভেতরে প্রতিটি ভাগই একবার-না-একবার টেস্ট হিসেবে ব্যবহৃত হবে। একটি লুপ চালিয়ে সেটাকে K-সংখ্যকবার ঘুরিয়ে এই কাজটি করা হয় এবং প্রতিটি ভিন্ন ভিন্ন টেস্ট সেটের জন্য ভিন্ন ভিন্ন পারফরম্যান্স ভ্যালু বের করা হয়। এরপর সবশেষে, সমস্ত

পারফরম্যান্স ভ্যালুর গড়মান নেওয়া হয়, যেটি কিনা এই মডেল সত্যিকারেই কতটুকু ভালো বা খারাপ, সেটি প্রকাশ করবে।

নিচের চার্টটি (চার্ট 2.4.1) দেখুন :

Iteration (1 to K)	Training Set	Test Set	Performance Score
1	D2, D3, D4, D5	D1	S1
2	D1, D3, D4, D5	D2	S2
3	D1, D2, D4, D5	D3	S3
4	D1, D2, D3, D5	D4	S4
5	D1, D2, D3, D4	D5	S5

চার্ট 2.4.1

এখন তাহলে ফাইনাল মডেল ইভ্যালুয়েশন স্কোর হচ্ছে $= \frac{1}{k} \sum_{i=1}^k S_i$

অর্থাৎ, আমাদের এই উদাহরণের ক্ষেত্রে $= \frac{S1+S2+S3+S4+S5}{5}$

ক্রস ভ্যালিডেশন একটি চমৎকার জিনিস। এখানে একটু খেয়াল করলে দেখবেন, আমরা কৌশলে কিন্তু পুরো ডেটাসেটকেই ট্রেনিং ও টেস্ট উভয় কাজেই ব্যবহার করছি এবং এমনভাবে সেটি করছি, যাতে ট্রেনিং ও টেস্টের আসল ধারণাটুকুও বজায় থাকে। অর্থাৎ ব্যাপারটি কিন্তু আমাদের সেই ইশতিয়াকের উদাহরণটির মতো হয়ে গেল, যেখানে পুরো বই-ই সিলেবাসে ছিল এবং প্রশ্ন পুরো বই থেকেই হয়েছে। সুতরাং আমরা বলতে পারি যে, ক্রস ভ্যালিডেশন করে আমরা একটি মডেল ভালো না খারাপ – এ ব্যাপারে যে সিদ্ধান্তে পৌঁছাব সেটি মোটামুটি যুক্তিযুক্ত ও গ্রহণযোগ্য হবে। তো, মোটামুটি এই হচ্ছে ক্রস ভ্যালিডেশনের ধারণাটুকু। আশা করি, সবাই বুঝতে পেরেছেন।

এরকম ছোটোখাটো আরো কিছু বিষয় আছে। সব এখানে একেবারে লিখলাম না। সামনে কাজ করার সঙ্গে সঙ্গে যখন যেটি দরকার হবে তখন সেটি লিখে দেব। আর আগেই বলেছি, এই বইতে মূলত কঠিন কঠিন থিওরি আর ম্যাথ আমরা একটু কম কম করে দেখিয়ে (যেতুকু না হলেই নয় ততটুকু শিখব) মেশিন লার্নিং আসলে কীভাবে ব্যবহার করা যায় সেটি উদাহরণ আকারে দেখব। আর চেষ্টা করব যতটা সহজ করে পারা যায়, বোঝার।

তো শুরু করা যাক?