

অধ্যায় ৯ : ডিসিশন ট্ৰি (Decision Tree)

ডিসিশন ট্ৰি - নাম থেকেই আন্দাজ কৰা যাচ্ছে অনেকটা যে এটি এক ধৰনের ট্ৰি, যেটি কিনা আমাদেৱ কোনো পৰিস্থিতিতে একটি যুক্তিযুক্ত সিদ্ধান্তে পৌছতে সাহায্য কৰে। ট্ৰি তৈরিৱ ব্যাপারগুলো অনেকে হয়তো আগে থেকে জানেন, যৰা গ্ৰাফ থিওৱ নিয়ে কিছুটা পড়াশোনা কৰেছেন। এ ছাড়া সাধাৰণ অ্যালগৱিন্দম কিংবা ডেটা স্ট্ৰাকচাৰ কোৰ্সেও ট্ৰি সম্পর্কে পড়েছেন অনেকেই আশা কৰি।

আমোৱা যে ডিসিশন ট্ৰি তৈরিৱ অ্যালগৱিন্দম নিয়ে পড়া (ID3, সামনে এটি নিতে বিস্তাৰিত আলোচনা আছে) তাৰ উজ্জ্বালক রস রুইনল্যান (Ross Quinlan, 1943)।

এটি আমাৰ সবচেয়ে প্ৰিয় মেশিন লাৰ্নিং অ্যালগৱিন্দম, আৱ তাই এই অধ্যায়টি বেশ বড়োসড়ো হয়ে যাবে। সবাইকে একটু হাত-পা ছড়িয়ে নিয়ে তাৰপৰে অধ্যায়টি পড়তে বসাৱ আমন্ত্ৰণ জানাচ্ছি।



ছবি ৯.১ : Ross Quinlan (1943)

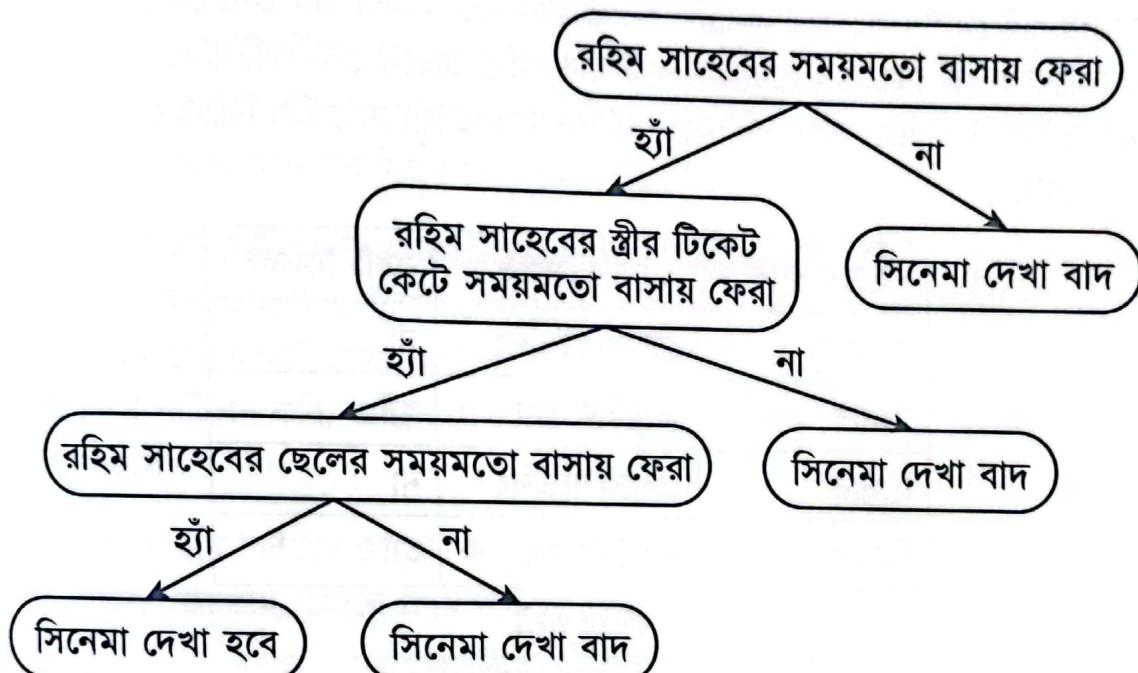
পৱিত্ৰেণ্ড ৯.১ : ডিসিশন ট্ৰি কী

ডিসিশন ট্ৰি তৈরিৱ পদ্ধতিতে যাওয়াৱ আগে, আগে একটু বুৱে নিই ডিসিশন ট্ৰি আসলে কীৱৰক। ধৰা যাক, রহিম সাহেব ঠিক কৰেছেন তিনি বাসায় ফিৱে আজকে তাৰ পৱিত্ৰেণ্ডৰ সবাইকে নিয়ে সিনেমা দেখতে যাবেন। বাসাৱ পাশেই বলাকা সিনেমা হল। যদিও রাস্তাঘাটেৱ যে অবস্থা, তিনি সময়মতো বাসায় ফিৱতে পাৱেন কি না সেটি এক দুষ্পিত্তা। বাসাৱ কাছাকাছি এমে তাৰ মনে পড়ল যে তাৰ অনলাইনে টিকিট কাটাৰ কথা ছিল, কিন্তু তিনি কাটতে ভুলে গেছেন। এখন সিনেমা হলে গিয়েই দেখতে হবে টিকিট আছে কি না। তিনি তাৰ স্ত্ৰীকে ফোন দিলেন টিকিট কাটতে যাওয়াৱ জন্য, কিন্তু জানতে পাৱলেন তাৰ স্ত্ৰী মাৰ্কেটে গেছেন, কেৱাৱ পথে কেটে আনতে পাৱলেন তিনি টিকিট। তাৰ হেলেকে ফোন দিলেন, হেলে গেছে কোচিংয়ে। তাৰ মানে দাঁড়াচ্ছে, যদি তিনি

অধ্যায় ৯ : ডিসিশন ট্ৰি (Decision Tree)

সময়মতো বাসায় পৌঁছাতে পারেন, তাঁৰ স্ত্ৰী যদি মার্কেট থেকে টিকিট কিনে সময়মতো বাসায় ফিরতে পারেন, ছেলে যদি সময়মতো কোচিং থেকে বাসায় ফিরতে পারে, তাহলেই কেবল সবাই রেডি হয়ে সুন্দরমতো হলে যেতে পারবেন সিনেমা দেখার জন্য, নাহলে পারবেন না।

এখন এই পরিস্থিতির জন্য আমরা যদি একটি ডিসিশন ট্ৰি বানাই, তাহলে সেটি দেখতে এরকম হবে (ছবি 9.1.1) :



ছবি 9.1.1

এই হচ্ছে মোটামুটি একটি ডিসিশন ট্ৰি-র চেহারা। এই ট্ৰি-ৰ রুট (root) হিসেবে আছে রহিম সাহেবের বাসায় ফেরার ব্যাপারটি।

এখন কথা হচ্ছে, এটাকে রুট হিসেবে না নিয়ে তো আমরা রহিম সাহেবের ছেলের সময়মতো বাসায় ফেরার ব্যাপারটিকেও রুট হিসেবে নিতে পারতাম। কিংবা রহিম সাহেবের স্ত্ৰীর টিকিট কাটার ব্যাপারটিকেও আমরা রুট হিসেবে নিতে পারতাম। তাহলে? কোনটিকে নেব রুট হিসেবে এবং কেন? আৱ ট্ৰি-এৰ যে নোডগুলো ইন্টাৱিমিডিয়েট নোড (যেগুলো লিফ নোড নয়, অৰ্থাৎ যাদের আৱ কোনো চাইল্ড নোড নেই) তাৰে ক্ৰমই বা কীভাৱে ঠিক কৰব যে, কোনটিৰ পৱে কোনটি আসবে?

এটি ঠিক কৰাৰ জন্যই ডিসিশন ট্ৰি তৈরিতে ইটাৱেটিভ ডাইকটোমাইজিং (Iterative Dichotomiser 3 -ID3) নামে একটি অ্যালগৱিদম ব্যবহাৰ কৰা হয়। ID3 অ্যালগৱিদম শিখতে হলে আগে দুটি বিষয় সম্বন্ধে ধাৰণা রাখতে হবে – এন্ট্ৰপি ও গেইন। সেগুলো নিয়ে আমরা এখন জানব।

পরিচ্ছেদ ৯.২ : এন্ট্রপি (Entropy)

এন্ট্রপির সোজাসাপ্তা বাংলা হচ্ছে বিশুক্ষলা। আমরা অনেকেই হয়তো উচ্চমাধ্যমিক পর্যায়ে
পদাৰ্থবিজ্ঞানে তাপগতিবিজ্ঞান (Thermodynamics) পড়াৰ সময় এই এন্ট্রপি বিষয়
পড়েছিলাম।

আমাদেৱ এই মেশিন লার্নিংয়ে এন্ট্রপি বলতে বোঝাবে, আমরা আমাদেৱ কোনো টেটোসেটকে
যদি কোনো একটি ফিচারেৰ সাপেক্ষে পার্টিশনিং কৱি, তাহলে সেই পার্টিশনিং কৱি ভালোভাবে
আমাদেৱ টার্গেট ভ্যারিয়েবলেৰ কলামকে পার্টিশন কৱতে পাৰবে সেটি। নিচেৱ টেবিলটি (টেবিল
9.2.1) দেখি।

বইয়েৰ ধৰণ	বইয়েৰ লেখক	বইটি কিনব?
ফিকশন	জে কে রাওলিং	না
থ্রিলার	সত্যজিৎ রায়	হ্যাঁ
থ্রিলার	জে কে রাওলিং	না
ফিকশন	সত্যজিৎ রায়	হ্যাঁ

টেবিল 9.2.1

প্ৰথমে বলে নেই, পার্টিশনিং ব্যাপারটি আসলে এৱকম – কোনো একটি ফিচার কলামেৰ যতজি
মান থাকবে, প্ৰতিটিৰ জন্য আমৱা টার্গেট ভ্যারিয়েবলেৰ ভিন্ন ভিন্ন মান নেব।

যেমন, ‘বইয়েৰ লেখক’ – এই কলামে দেখুন, দুটি ভিন্ন লেখকেৰ নাম আছে, ‘জে কে রাওলিং’
ও ‘সত্যজিৎ রায়’। আমৱা যদি এখন, এই ‘বইয়েৰ লেখক’ কলাম দিয়ে পুৱো টেটোসেটকে
পার্টিশন কৱি, তাহলে দুটো পার্টিশন পাৰ। প্ৰথমটি জে কে রাওলিংয়েৰ জন্য, যে ক্ষেত্ৰে টার্গেট
কলাম ‘বই কিনব?’ এৱ সবগুলো মানই ‘না’।

বইয়েৰ ধৰণ	বইয়েৰ লেখক	বইটি কিনব?
ফিকশন	জে কে রাওলিং	না
থ্রিলার	জে কে রাওলিং	না

টেবিল 9.2.2

আৱ দ্বিতীয়টি সত্যজিৎ রায়েৰ জন্য, যে ক্ষেত্ৰে টার্গেট কলাম ‘বই কিনব?’ এৱ সবগুলো মানই
‘হ্যাঁ’।

অধ্যায় ৯ : ডিসিশন ট্রি (Decision Tree)

বইয়ের ধরন	বইয়ের লেখক	বইটি কিনব?
ফিকশন	সত্যজিৎ রায়	হ্যাঁ
থ্রিলার	সত্যজিৎ রায়	হ্যাঁ

টেবিল 9.2.3

এখন, আমরা যদি একটু খেয়াল করি, তাহলে দেখব, আমরা ‘বইয়ের লেখক’ দিয়ে পার্টিশনিং করার ফলে যে দুটো পার্টিশন তৈরি হলো, তারা কিন্তু আমাদের টার্গেট কলামটিকে মিনিমাম বা শূন্য এন্ট্রিপিতে পার্টিশন করতে পেরেছে। অর্থাৎ, প্রথম পার্টিশনের জন্য সব টার্গেট ভ্যালুর মানই না, এখানে কোনো ‘হ্যাঁ-না’র মিশ্রণ নেই, অর্থাৎ একটি ‘হ্যাঁ’, আরেকটি ‘না’ – এরকম নয়।

এইভাবে, দ্বিতীয় পার্টিশনের ক্ষেত্রেও সবগুলো টার্গেট ভ্যালু ‘হ্যাঁ’, কোনো ‘হ্যাঁ-না’র মিশ্রণ নেই। যদি মিশ্রণ থাকত, তাহলে এন্ট্রিপি অনেক বেশি হতো, অর্থাৎ বিশুঙ্খলা বেশি হতো। আমরা সেটি চাই না। আমাদের লক্ষ্যই হলো এন্ট্রিপির মান কমিয়ে নিয়ে আসা। যদি আমরা ‘বইয়ের লেখক’ কলাম দিয়ে পার্টিশন করি, তাহলে দেখতেই পাচ্ছি দুটো পার্টিশনের জন্যই আমাদের এন্ট্রিপির মান শূন্য হবে।

এখন, আমরা যদি ‘বইয়ের লেখক’ কলাম দিয়ে পার্টিশন না করে ‘বইয়ের ধরন’ নামের কলাম দিয়ে পার্টিশন করতাম, তাহলে প্রথম পার্টিশন হতো ‘ফিকশন’ আউটকামের জন্য –

বইয়ের ধরন	বইয়ের লেখক	বইটি কিনব?
ফিকশন	জে কে রাওলিং	না
ফিকশন	সত্যজিৎ রায়	হ্যাঁ

টেবিল 9.2.4

প্রবর্তী পার্টিশন হতো ‘থ্রিলার’ আউটকামের জন্য –

বইয়ের ধরন	বইয়ের লেখক	বইটি কিনব?
থ্রিলার	সত্যজিৎ রায়	হ্যাঁ
থ্রিলার	জে কে রাওলিং	না

টেবিল 9.2.5

এখন এই ‘বইয়ের ধরন’ দিয়ে করা পার্টিশন দুটি যদি দেখি, তাহলে দেখব যে এখানে এন্ট্রিপি রয়েছে, কারণ পার্টিশনের কারণে টার্গেট কলামে হ্যাঁ-না মিশ্রণ চলে এসেছে।

অর্থাৎ, আমাদের ডিসিশন প্রি-তে যখনই আমরা কোনো ডিসিশন নোড বেছে নেব (কেট নোড কিংবা অন্য যে-কোনো অন্তর্বর্তী নোড) তখন সেটি বেছে নেওয়ার সময় আমাদের এই এন্ট্রপির আশ্রয় নিতে হবে এবং দেখতে হবে কোনটি ব্যবহার করলে আমাদের সবচেয়ে কম এন্ট্রপি হবে।

এন্ট্রপি গাণিতিকভাবে বের করার একটি সূত্র আছে। সূত্রটি হচ্ছে -

যে-কোনো সেট S-এর জন্য,

$$\checkmark \quad Entropy(S) = \sum_{i=1}^n -P_i \log_2 P_i$$

এবং এই এন্ট্রপি হিসাব করতে হবে যে-কোনো পার্টিশনের জন্য মোট কয়টি ভিন্ন টার্গেট ভ্যারিয়েবল আছে সেখান থেকে।

যেমন, আমাদের ডেটাসেটের জন্য, টার্গেট ভ্যারিয়েবলের মোট চারটি মানের ভেতরে 'না' আছে দুটি, 'হ্যাঁ' আছে দুটি। সুতরাং

$$\begin{aligned} Entropy(S) &= \sum_{i=1}^n -P_i \log_2 P_i \\ &= -\left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4}\right) \\ &= 1 \end{aligned}$$

এখানে P_i হচ্ছে যে-কোনো একটি টার্গেট ভ্যারিয়েবলের মান ঘটার সম্ভাব্যতা। যেমন, আমাদের ডেটাসেটে 'না' আছে দুটি, মোট ডেটা চারটি। সুতরাং, টার্গেট ভ্যারিয়েবল 'না' হওয়ার সম্ভাব্যতা, $\frac{2}{4}$ ।

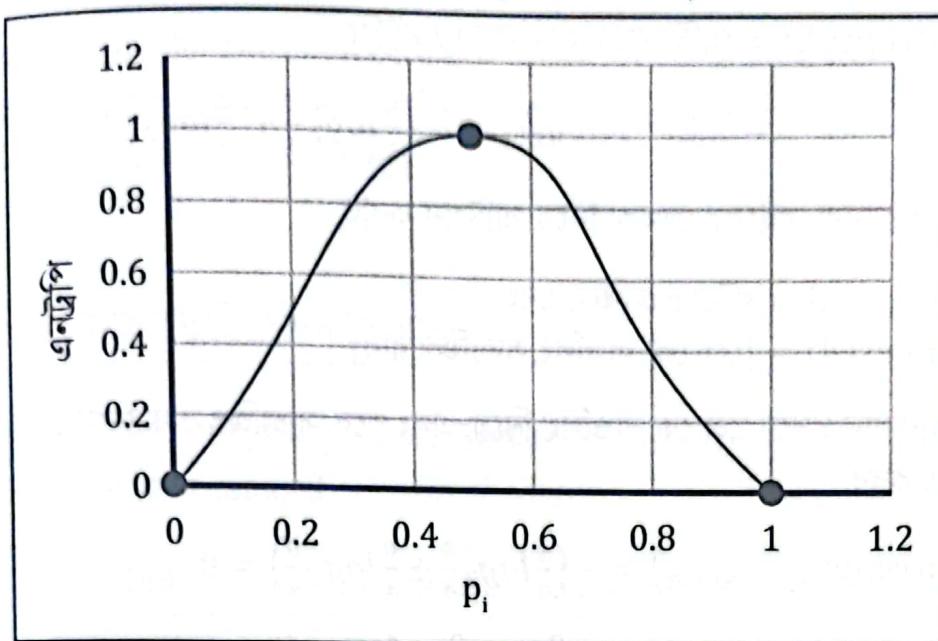
গ্রাফ 9.2.1 লক্ষ করুন। এই গ্রাফটি হচ্ছে, এন্ট্রপি গ্রাফ, যেটি থেকে এন্ট্রপি সম্বন্ধে একটি ধারণা পাওয়া যায়। গ্রাফের X-অক্ষ বরাবর নেওয়া হয়েছে P_i ও Y-অক্ষ বরাবর নেওয়া হয়েছে এন্ট্রপির মান।

গ্রাফ থেকে দেখা যাচ্ছে যে, যদি টার্গেট ভ্যারিয়েবলের আউটকামগুলো কোনো একটি পার্টিশনের জন্য সব এক ধরনের হয় (সবগুলো 'হ্যাঁ' কিংবা সবগুলো 'না') তাহলে সেই পার্টিশনের এন্ট্রপি হয় শূন্য।

আর যদি, সমানসংখ্যক থাকে প্রতিটি আউটকাম (যেমন, ধরা যাক 10টি আউটকামের মধ্যে 5টি হ্যাঁ, 5টি না – এরকম) সে ক্ষেত্রে সম্ভাব্যতা হয় 0.5 এবং এই মানের জন্যই এন্ট্রপি ভ্যালু দেখুন সবচেয়ে বেশি।

আমাদের লক্ষ্যই হচ্ছে এমনভাবে পার্টিশন করা, যাতে এন্ট্রপির মান সর্বনিম্নে অর্থাৎ শূন্যে নামিয়ে আনতে পারি।

অধ্যায় ৯: ডিসিশন ট্ৰি (Decision Tree)



গ্রাফ 9.2.1

পরিচেদ ৯.৩ : গেইন (Gain)

গেইন মানে হচ্ছে কোনো কিছু পাওয়া, অর্জন করা, তাই না? মেশিন লার্নিংয়ের ক্ষেত্রেও গেইন মানে ঠিক তা-ই। আমরা কোনো গাণিতিক হিসাবনিকাশ করে কিছু একটা পাব, সেটিই হবে আমাদের গেইন। এখন কথা হচ্ছে কী পাব?

আমরা এতক্ষণ এন্ট্রোপি পড়ার সময় বলেছি যে, আমাদের লক্ষ্য হচ্ছে পার্টিশনের পরের এন্ট্রোপি কমানো। অর্থাৎ, যদি আমাদের পার্টিশনের আগের মোট এন্ট্রোপি হয় E_A এবং পরের মোট এন্ট্রোপি হয় E_B তাহলে, আমরা চাইছি E_A ও E_B -এর মধ্যেকার পার্থক্য যতটা স্তুব বাড়াতে, অর্থাৎ $E_B \ll E_A$ করতে।

এই E_A ও E_B -এর মধ্যেকার মানের পার্থক্যই হচ্ছে গেইন। আমাদের লক্ষ্য হচ্ছে E_B -এর মান যত পারা যায় মিনিমাইজ করার মাধ্যমে গেইনের মান যত পারা যায় ম্যাক্সিমাইজ করা।

তাহলে, গেইনের সূত্র হচ্ছে :

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v)$$

আমাদের বইয়ের ডেটাসেটের জন্য, আমরা যদি এখন, 'বইয়ের লেখক' দিয়ে পার্টিশন করি, তাহলে,

$Entropy(S)$ = পার্টিশন করার আগের এন্ট্রপি

$$= \sum_{i=1}^n -P_i \log_2 P_i = -\left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4}\right) = 1$$

এখন, যেহেতু আমরা 'বইয়ের লেখক' দিয়ে পার্টিশন করছি,

তাই, A = বইয়ের লেখক, এবং

$Values(A) = \{\text{জে কে রাওলিং, সত্যজিৎ রায়}\}$

সুতরাং, v -এর মান প্রথমে 'জে কে রাওলিং' নিয়ে, এবং পরে 'সত্যজিৎ রায়' নিয়ে সেই পার্টিশন এন্ট্রপি বের করব -

$$Entropy(S_{\text{জে কে রাওলিং}}) = -\left(\frac{4}{4} \log_2 \frac{4}{4} + \frac{0}{4} \log_2 \frac{0}{4}\right) = 0$$

$$Entropy(S_{\text{সত্যজিৎ রায়}}) = -\left(\frac{0}{4} \log_2 \frac{0}{4} + \frac{4}{4} \log_2 \frac{4}{4}\right) = 0$$

সবশেষে, 'জে কে রাওলিং' নিয়ে পার্টিশনের ফ্রেটে $\frac{|S_v=\text{জে কে রাওলিং}|}{|S|} = \frac{2}{4}$, যেহেতু জে কে রাওলিং নিয়ে মোট চারটি উদাহরণের মধ্যে 2টি উদাহরণ আছে এবং একইভাবে সত্যজিৎ রায় নিয়ে পার্টিশনের ফ্রেটে, $\frac{|S_v=\text{সত্যজিৎ রায়}|}{|S|} = \frac{2}{4}$ ।

সুতরাং, সবশেষে,

$$\begin{aligned} Gain(S, A) &= Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v) \\ &= 1 - \left(\frac{2}{4} \times 0 + \frac{2}{4} \times 0\right) \\ &= 1 \end{aligned}$$

একইভাবে, 'বইয়ের ধরন' দিয়ে পার্টিশন করে আমরা গেইন পাই,

$$\begin{aligned} Gain(S, A) &= Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v) \\ &= 1 - \left[\frac{2}{4} \times \left\{-\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right)\right\} + \frac{2}{4} \times \left\{-\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right)\right\}\right] \\ &= 1 - \left(\frac{2}{4} \times 2\right) \\ &= 0 \end{aligned}$$

অধ্যায় ৯ : ডিসিশন ট্ৰি (Decision Tree)

আমাদের লক্ষ্য ছিল যে পার্টিশনের গেইন বেশি হবে, আমরা সেই পার্টিশন ব্যবহার করব। আমরা দেখতে পাচ্ছি যে, 'বইয়ের লেখক' দিয়ে পার্টিশন করলে আমরা গেইন সর্বোচ্চ পাচ্ছি, সুতরাং আমরা পার্টিশন করব 'বইয়ের লেখক' দিয়ে।

পরিচেদ ৯.৪ : কীভাবে ডিসিশন ট্ৰি বানাব

এখন আমরা গোড়া থেকে সম্পূর্ণ নতুন একটি উদাহরণ করে দেখব কীভাবে ডিসিশন ট্ৰি তৈরি কৰতে হয়। আমাদের নতুন ডেটাসেট :

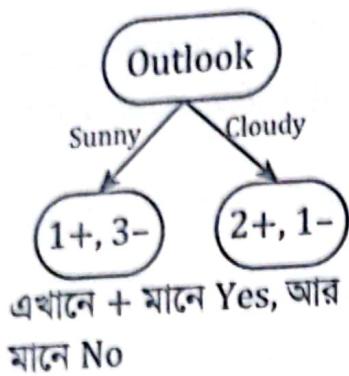
Day	Outlook	Temperature	Routine	Wear Coat?
D_1	Sunny	Cold	Indoor	No
D_2	Sunny	Warm	Outdoor	No
D_3	Cloudy	Warm	Indoor	No
D_4	Sunny	Warm	Indoor	No
D_5	Cloudy	Cold	Indoor	Yes
D_6	Cloudy	Cold	Outdoor	Yes
D_7	Sunny	Cold	Outdoor	Yes

মনে করে দেখুন, এই ডেটাসেটটিই আমরা নাইভ বেইজ ক্লাসিফায়ার তৈরি করার সময় ব্যবহার কৰেছিলাম। সেটিই আবারও ব্যবহার করে আমরা ডিসিশন ট্ৰি-ও তৈরি কৰব।

Root Selection:

আমাদের সবার আগে ডিসিশন ট্ৰি-এর Root Selection কৰতে হবে। একটু আগে যে পদ্ধতি দেখলাম, সেই পদ্ধতিতে। আমাদের এখানে ফিচার আছে তিনটি – Outlook, Temperature ও Routine। আর আমাদের ডেটা আছে মোট ৭টি। এখন আমরা তিনটি ফিচারের প্রতিটির জন্য আলাদা আলাদা কৰে, এদের দিয়ে পুরো ডেটাসেট পার্টিশন কৰলে কত গেইন হবে সেটি বের কৰব। এরপৰ, যেটি সর্বোচ্চ গেইন দেবে তাকে রুট নোড হিসেবে নির্বাচন কৰব।

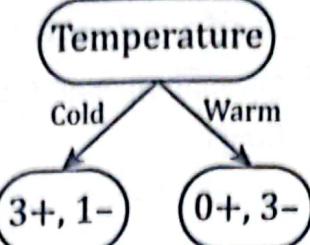
$$\text{এখানে, } \text{পার্টিশন কৰার আগের এন্ট্রপি } E(S) = -\left(\frac{3}{7} \log_2 \frac{3}{7} + \frac{4}{7} \log_2 \frac{4}{7}\right) = 0.985$$



$$E(\text{Outlook_Sunny}) = -\left(\frac{1}{4} \log_2 \frac{1}{4} + \frac{3}{4} \log_2 \frac{3}{4}\right) = 0.811$$

$$E(\text{Outlook_Cloudy}) = -\left(\frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3}\right) = 0.918$$

$$\begin{aligned} Gain(S, \text{Outlook}) &= \\ &= E(S) - \frac{4}{7} \times 0.811 \\ &\quad - \frac{3}{7} \times 0.918 \\ &= 0.985 - \frac{4}{7} \times 0.811 \\ &\quad - \frac{3}{7} \times 0.918 \\ &= 0.128 \end{aligned}$$

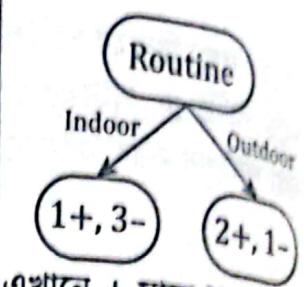


এখনে + মানে Yes, আর - মানে No

$$E(\text{Temperature_Cold}) = -\left(\frac{1}{4} \log_2 \frac{1}{4} + \frac{3}{4} \log_2 \frac{3}{4}\right) = 0.811$$

$$E(\text{Temperature_Warm}) = -\left(\frac{0}{3} \log_2 \frac{0}{3} + \frac{3}{3} \log_2 \frac{3}{3}\right) = 0$$

$$\begin{aligned} Gain(S, \text{Temperature}) &= \\ &= E(S) - \frac{4}{7} \times 0.811 \\ &\quad - \frac{3}{7} \times 0 \\ &= 0.985 - \frac{4}{7} \times 0.811 \\ &\quad - \frac{3}{7} \times 0 \\ &= 0.521 \end{aligned}$$

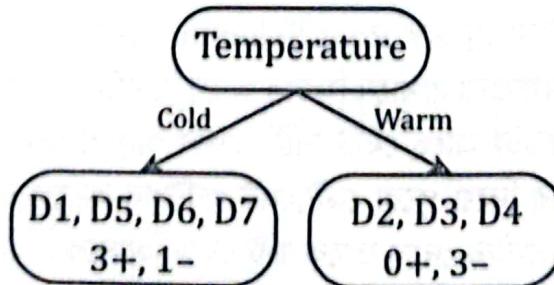


$$E(\text{Routine_Indoor}) = -\left(\frac{1}{4} \log_2 \frac{1}{4} + \frac{3}{4} \log_2 \frac{3}{4}\right) = 0.811$$

$$E(\text{Routine_Outdoor}) = -\left(\frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3}\right) = 0.918$$

$$\begin{aligned} Gain(S, \text{Routine}) &= \\ &= E(S) - \frac{4}{7} \times 0.811 \\ &\quad - \frac{3}{7} \times 0.918 \\ &= 0.985 - \frac{4}{7} \times 0.811 \\ &\quad - \frac{3}{7} \times 0.918 \\ &= 0.128 \end{aligned}$$

এখন, ওপরের তিনটি গেইন-এর মানের মধ্যে, Temperature-এর মাধ্যমে করা পার্টিশনের গেইন সবচেয়ে বেশি। সুতরাং, আমরা Temperature-কেই আমাদের রুট নোড হিসেবে নির্বাচন করব। তাহলে, আমরা এখন একটি পার্শিয়াল ডিসিশন ট্ৰি (Partial Decision Tree) পেয়ে গেলাম, যেটি দেখতে এরকম (ছবি 9.4.1) :



ছবি 9.4.1

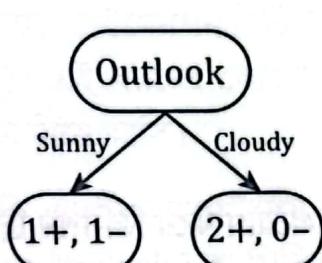
অধ্যায় ৯ : ডিসিশন ট্ৰি (Decision Tree)

এখন, আমাদের ডেটাসেটটি দুটি সাবসেটে ভাগ হয়ে গেল। এর মধ্যে একটি সাবসেটের এন্ট্রপি শূন্য (ডান দিকেরটি), সুতরাং ওটি নিয়ে আমাদের আর কোনো মাথাব্যথা থাকবে না। ওই নোডে পৌছানো মানে Wear Coat-এর মান সব সময়েই No হবে।

এখন বাকি থাকে, বাঁ দিকের সাবসেট – D1, D5, D6 এবং D7। এটিকে আমাদের আবার পার্টিশন করতে হবে এবং ততক্ষণ পর্যন্ত এই পার্টিশন করা চালিয়ে যেতে হবে যতক্ষণ পর্যন্ত না আমরা এন্ট্রপি শূন্য পাচ্ছি। আমাদের এই নতুন সাবসেটের নাম দিই S1। এই S1-এর এন্ট্রপি হবে –

$$E(S1) = -\left(\frac{3}{4} \log_2 \frac{3}{4} + \frac{1}{4} \log_2 \frac{1}{4}\right) = 0.811$$

এখন আমরা এই নতুন সাবসেটকে আমাদের মূল ডেটাসেট বিবেচনা করে এর ওপরে আগের মতো পার্টিশন চালিয়ে দেখব কোনটিতে গেইন বেশি পাওয়া যায়। আমরা যেহেতু Temperature ফিচারটি ইতিমধ্যেই ব্যবহার করে ফেলেছি, তাই আমাদের এখন শুধু Outlook ও Routine – এই দুটি ফিচার দিয়ে পার্টিশন করে দেখলেই চলবে।

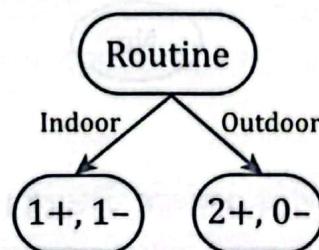


এখানে + মানে Yes, আর – মানে No

$$E(\text{Outlook_Sunny}) \\ = -\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right) = 1$$

$$E(\text{Outlook_Cloudy}) \\ = -\left(\frac{2}{2} \log_2 \frac{2}{2} + \frac{0}{2} \log_2 \frac{0}{2}\right) = 0$$

$$\begin{aligned} Gain(S, \text{Outlook}) \\ &= E(S1) - \frac{2}{4} \times 1 - \frac{2}{4} \times 0 \\ &= 0.811 - \frac{2}{4} \times 1 - \frac{2}{4} \times 0 \\ &= 0.311 \end{aligned}$$



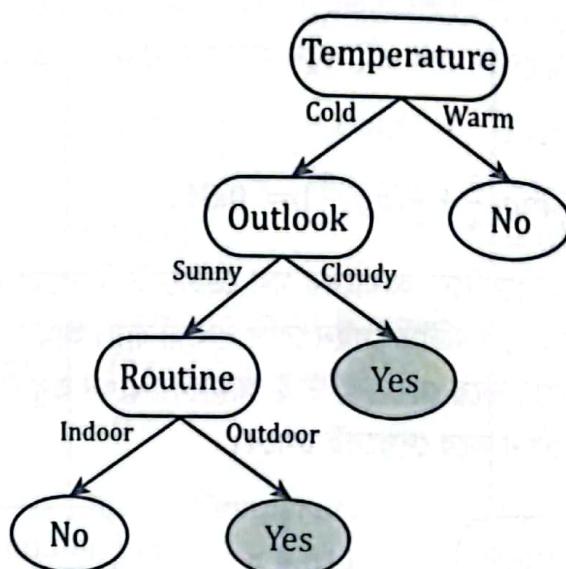
এখানে + মানে Yes, আর – মানে No

$$E(\text{Routine_Indoor}) \\ = -\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right) = 1$$

$$E(\text{Routine_Outdoor}) \\ = -\left(\frac{2}{2} \log_2 \frac{2}{2} + \frac{0}{2} \log_2 \frac{0}{2}\right) = 0$$

$$\begin{aligned} Gain(S, \text{Routine}) \\ &= E(S1) - \frac{2}{4} \times 1 - \frac{2}{4} \times 0 \\ &= 0.811 - \frac{2}{4} \times 1 - \frac{2}{4} \times 0 \\ &= 0.311 \end{aligned}$$

এখন দেখা যাচ্ছে, আমাদের Outlook ও Routine দুটির ক্ষেত্রেই গেইনের পরিমাণ একই।
 সুতরাং যে-কোনোটিকেই আমরা পরবর্তী ডিসিশন নোড হিসেবে ধরে নিতে পারি। আমরা যদি
 Outlook-কে আমাদের পরবর্তী ডিসিশন নোড ধরে নিই, তাহলে বাকি থাকে শুধু Routine
 সেটি দিয়েও আমরা একইভাবেই পার্টিশন করে দেখব। সব ঠিকঠাকমতো করতে পারলে
 আমাদের সর্বশেষ ডিসিশন ট্রি দাঁড়াবে এরকম (ছবি 9.4.2) :



ছবি 9.4.2

সুতরাং আমরা শিখে ফেললাম কীভাবে ID3 অ্যালগরিদম প্রয়োগ করে ডিসিশন ট্রি তৈরি করতে
 হয়। আশা করি সকলেই বুবাতে পেরেছেন।