

## অধ্যায় ১০ : প্রিসিপাল কম্পোনেন্ট অ্যানালাইসিস (Principal Component Analysis - PCA)

ধরুন, কোন ডেটাসেটের জন্য, ফিচারের সংখ্যা অনেক বেশি ( $> 10,000$ ) এবং সেই তুলনায় ট্রেইনিং ডেটা পর্যাপ্ত নেই। এ ধরনের ক্ষেত্রে হিসাবনিকাশ অনেক বেশি জটিল হয়ে যায় এবং Costing-ও অনেক বেড়ে যায় (সময়, মেমোরি বেশি লাগে ইত্যাদি)।

এসব ক্ষেত্রে, এত এত ফিচারের মধ্যে খুব ভালোমতো যদি ডেটা অ্যানালাইসিস করা যায়, তাহলে দেখা যাবে যাত্র কিছুসংখ্যক ফিচার আমাদের দরকার, বাকিগুলো না হলেও হবে। তখন, আমরা অপ্রযোজনীয় ফিচারগুলো বাদ দিয়ে দিই, যাকে বলে ডাইমেশন রিডাকশন (Dimension Reduction)।

এক ধরনের অ্যালগরিদম আছে, যেগুলো ডেটার এই ডাইমেশন রিডাকশনের কাজ করে দেয়, এগুলোকে বলে ডাইমেনশনালিটি রিডাকশন অ্যালগরিদম (Dimensionality Reduction Algorithm)। এই ধরনের কোনো একটি অ্যালগরিদম ব্যবহার করে আমরা ডেটার ফিচারের সংখ্যা কমিয়ে নিয়ে আসি।

এই ধরনেরই একটি অ্যালগরিদম হলো প্রিসিপাল কম্পোনেন্ট অ্যানালাইসিস (Principal Component Analysis) বা পিসিএ (PCA)। নাম থেকেই বোঝা যাচ্ছে, কোথাও অনেকগুলো কম্পোনেন্ট আছে, আমাদের সেখান থেকে যে কম্পোনেন্ট/কম্পোনেন্টগুলো সবচেয়ে বেশি গুরুত্বপূর্ণ সেগুলো রেখে বাকিগুলো বাদ দিয়ে দিতে হবে। ব্যাপারটি আসলেই অনেকটা এরকম। মেশিন লার্নিংয়ের ক্ষেত্রে, এই কম্পোনেন্ট বলতে বোঝায় ফিচার ডেটা।

এই বিষয় নিয়ে পড়ার আগে পরিসংখ্যান এবং লিনিয়ার অ্যালজেব্রার কিছু ছোটো ছোটো বিষয় আমরা চট করে দেখে নেব। অনেকেই হয়তো এগুলো জানেন, অনেকেই হয়তো জানেন না। লেখাটি মূলত যাঁরা জানেন না তাঁদের জন্যই, আর যাঁরা জানেন তাঁরাও আরেকবার ঝালাই করে নিলে ক্ষতি কী? আমরা শুরু করব পরিসংখ্যান এর কিছু জনপ্রিয় ধারণা – মিন (Mean), স্ট্যান্ডার্ড ডেভিয়েশন (Standard Deviation) ও ভ্যারিয়েন্স (Variance) দিয়ে, হাতে-কলমে দেখব – কীভাবে এগুলো বের করতে হয়। এরপর আমরা লিনিয়ার অ্যালজেব্রার দুটি বিষয় দেখব – কীভাবে একটি ম্যাট্রিক্সের জন্য আইগেনভ্যালু (Eigenvalue) ও আইগেনভেক্টর (Eigenvector) বের করতে হয়। সবশেষে আমরা, এগুলো ব্যবহার করে কীভাবে পিসিএ (PCA) কাজ করে সেটি দেখব।

## পরিচ্ছন্দ ১০.১ : মিন (Mean), স্ট্যান্ডার্ড ডেভিয়েশন (Standard Deviation) এবং ভ্যারিয়েন্স (Variance)

এগুলো আমরা সবাই-ই ছোটোবেলায় কমবেশি পড়ে এসেছি। তাও, আরেকবার বালাই করে নিই। মিন (Mean) হচ্ছে গাণিতিক গড়।

$$\checkmark \text{ গড় বের করার সূত্র}, \bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

এই ভ্যাবহ চেহারার সমীকরণের মূল বক্তব্য হলো, যতগুলো ডেটা থাকবে, সবগুলোকে যোগ করে, ডেটার সংখ্যা দিয়ে ভাগ দিলেই গড় পাওয়া যাবে।

কোনো ডেটাসেটের গড় থেকে সেই ডেটাসেটের মানগুলোর ব্যাপারে একটু ধারণা পাওয়া যায়। তবে এর পাশাপাশি, যদি আমরা সেই ডেটাসেটের স্ট্যান্ডার্ড ডেভিয়েশন বের করতে পারি, তাহলে ডেটাসেটের মানগুলোর ব্যাপারে আরো পরিষ্কার ধারণা পাওয়া যায়।

$$\checkmark \text{ স্ট্যান্ডার্ড ডেভিয়েশন বের করার সূত্র}, SD = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{x})^2}{(n-1)}}$$

সহজ করে বলতে গেলে স্ট্যান্ডার্ড ডেভিয়েশন যা বোঝায় তা হলো, আমাদের ডেটাসেটের প্রতিটি ডেটা সেই ডেটাসেটের গড়মান থেকে কত দূরে আছে (কম বা বেশি), সেই দূরত্বের মানগুলোর বর্গের গড় মানের বর্গমূল বা সংক্ষেপে আরএমএস (RMS) গড়।

এখানে, একটি প্রশ্ন প্রায়ই সবার মাথায় আসে যে, আমাদের মোট ডেটার সংখ্যা তো  $n$ , তাহলে কেন আমরা  $n$  দিয়ে ভাগ না করে  $(n-1)$  দিয়ে ভাগ করছি? এর কারণটি ভেঞ্জে বলার চেষ্টা করছি।

ধরা যাক, আপনি পৃথিবীর সব মানুষের বয়সের গড় বের করতে চান। তাহলে আপনাকে কী করতে হবে? পৃথিবীর সব মানুষের বয়স এক এক করে যোগ করে তাকে মোট মানুষের সংখ্যা দিয়ে ভাগ করতে হবে, তাই না? কিন্তু, বাস্তবে এই কাজটি করা অসম্ভব; কোনোভাবেই প্রতিটি মানুষের কাছে গিয়ে, তার বয়স জেনে নিয়ে সব যোগ করে গড় বের করা সম্ভব নয়।

তাই এ ক্ষেত্রে যাঁরা এই ধরনের ডেটা নিয়ে কাজ করতে যান, তাঁরা যেটি করেন, তা হলো, এই সাড়ে সাত বিলিয়ন মানুষের ডেটাসেট না বানিয়ে আরো ছোটো সাইজের ডেটাসেট (ধরুন, 1000 জন মানুষের ডেটাসেট) তৈরি করেন। তাঁরা এই 1000 জন মানুষকে এমনভাবে নির্বাচন করেন যাতে, এতে পৃথিবীর ভিন্ন ভিন্ন প্রায় সব দেশের, সব বয়সের, ধর্মের, বর্ণের, গোত্রের মানুষ থাকে। কিংবা তাঁরা যে ডোমেইন নিয়ে কাজ করতে চান, সে ডোমেইনের সব ধরনের ভ্যারিয়েশন যেন এই ডেটাসেটে বিদ্যমান থাকে, সেটি নিশ্চিত করেন।



## অধ্যায় ১০ : প্রিসিপাল কম্পোনেন্ট অ্যানালাইসিস (Principal Component Analysis - PCA)

যেমন, তাঁরা যদি বিভিন্ন দেশের মানুষের মধ্যে পর্যালোচনা করতে চান, তাঁরা নিশ্চিত করেন বা করতে চান যে তাঁদের ডেটাসেটে যেন প্রতিটি দেশ থেকে একজন মানুষের তথ্য থাকে। এইভাবে ধরে নেওয়া হয় যে, ওই একজন মানুষের সম্পর্কে তথ্য পেলে সেটি ওই গোটা দেশের মানুষের সম্পর্কেই তথ্য পেয়ে যাওয়া হবে। এই ধরনের ডেটাসেটকে বলা হয় স্যাম্পল ডেটা সেট এবং সেটি আমাদের প্রধান সেট, যেটি পৃথিবীর সব মানুষকে নিয়ে যে সেট তৈরি হবে, তার একটি সাবসেট।

এখন ধরা যাক, আপনি যদি আপনার প্রধান ডেটাসেট নিয়ে কাজ করতেন, তাহলে তার স্ট্যান্ডার্ড ডেভিয়েশন হতো  $S$  এবং স্যাম্পল ডেটাসেট নিয়ে কাজ করার কারণে যদি আপনি  $n$  দিয়ে ভাগ করেন, তাহলে আপনার স্ট্যান্ডার্ড ডেভিয়েশন হবে  $S_n$  এবং যদি আপনি  $n - 1$  দিয়ে ভাগ করেন, তাহলে সেটি হবে  $S_{n-1}$ ।

বাস্তবে দেখা যায়, প্রধান ডেটাসেটের বদলে স্যাম্পল ডেটাসেট ব্যবহার করা হলে,  $S_n$ -এর চেয়ে  $S_{n-1}$ -এর মান প্রধান ডেটাসেটের  $S$ -এর বেশি কাছাকাছি হয়। আর আমরা যেহেতু বেশিরভাগ সময়ই প্রধান ডেটাসেটের বদলে স্যাম্পল ডেটাসেট ব্যবহার করি, তাই আমরা  $n$ -এর পরিবর্তে  $n - 1$  দিয়ে ভাগ করি।

যদি, আমরা স্যাম্পল ডেটাসেট ব্যবহার না করে প্রধান ডেটাসেটই ব্যবহার করতাম, তাহলে অবশ্যই আমরা  $n$  দিয়ে ভাগ করতাম।

সবশেষের যে সূত্রটি, সেটি হলো ভ্যারিয়েন্স (Variance)-এর সূত্র। ভ্যারিয়েন্স ও স্ট্যান্ডার্ড ডেভিয়েশন প্রায় একই জিনিস। স্ট্যান্ডার্ড ডেভিয়েশনের মানকে বর্গ করলেই আমরা ভ্যারিয়েন্স পেয়ে যাই।

$$\text{সূত্রাঃ, } Var(X) = SD^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)}$$

## পরিচ্ছেদ ১০.২ : কোভ্যারিয়েন্স (Covariance) ও কোভ্যারিয়েন্স ম্যাট্রিক্স (Covariance Matrix)

কোভ্যারিয়েন্স (Covariance) ও ভ্যারিয়েন্সের মধ্যে অনেকখানিই মিল আছে, পার্থক্য শুধু, ভ্যারিয়েন্স কাজ করে একমাত্রিক বা 1-D ডেটা দিয়ে, আর কোভ্যারিয়েন্স কাজ করে দ্বিমাত্রিক বা 2-D ডেটা দিয়ে। উদাহরণ দিয়ে বোঝানোর চেষ্টা করছি।

আমাদের সেই পিংজার উদাহরণটিতেই ফিরে যাই :

পিংজার সাইজ (ইঞ্জিতে)	পিংজার দাম (টাকায়)
6"	350/-
8"	775/-
12"	1150/-
14"	1395/-
18"	1675/-

## টেবিল 10.2.1

এখানে দেখুন, পিংজার সাইজ হচ্ছে একটি ডাইমেনশন এবং পিংজার দাম হচ্ছে আরেকটি ডাইমেনশন (যদিও পিংজার দামকে আসলে ঠিক ডেটার ডাইমেনশন হিসেবে সরাসরি ধরা যাবে না, তবু এখানে ধরে নিচ্ছি বোঝানোর সুবিধার্থে)।

এখন আমরা যদি শুধু পিংজার সাইজ কিংবা পিংজার দাম নিয়ে কাজ করতাম, তাহলে আমরা ব্যবহার করতাম ভ্যারিয়েন্স। কিন্তু যদি আমরা বের করতে যাই যে, পিংজার সাইজ বাড়ার সঙ্গে সঙ্গে কি পিংজার দাম বাড়ছে, নাকি কমছে, ইত্যাদি, অর্থাৎ দুটি ভিন্ন ভিন্ন ডাইমেনশনের ডেটার মধ্যেকার সম্পর্কটি আসলে কী, সেটি যদি বুঝতে চাই, তাহলে আমাদের ব্যবহার করতে হবে কোভ্যারিয়েন্স।

কোভ্যারিয়েন্সকে লেখা হয় এভাবে –  $Cov(X, Y)$ ; এখানে  $X$  হচ্ছে প্রথম ডাইমেনশনের ডেটাসেট,  $Y$  হচ্ছে দ্বিতীয় ডাইমেনশনের ডেটাসেট। এটি বের করার সূত্রও অনেকটাই ভ্যারিয়েন্সের সূত্রের মতো –

$$\checkmark Cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

এখন,  $Cov(X, Y)$ -এর মান যা-ই আসুক, সেটি আমাদের মাথাব্যথা নয়। আমাদের শুধু দেখতে হবে যে, মানটি ধনাত্মক, শূন্য নাকি ঋণাত্মক।

- যদি মান ধনাত্মক হয়, তাহলে আমরা বলতে পারব যে ডাইমেনশন দুটি ধনাত্মক সম্পর্কযুক্ত বা পজিটিভলি কোরিলেটেড (Positively Correlated) অর্থাৎ, একটির মান বাড়লে অন্যটির মান বাড়বে, একটির মান কমলে অন্যটি কমবে।
- আবার, মান যদি ঋণাত্মক হয়, তাহলে আমরা বলতে পারিব যে এরা ঋণাত্মক সম্পর্কযুক্ত বা নেগেটিভলি কোরিলেটেড (Negatively Correlated) অর্থাৎ একটির মান বাড়লে অন্যটির মান কমবে।

## অধ্যায় ১০ : প্রিমিপাল কম্পোনেন্ট অ্যানালাইসিস (Principal Component Analysis - PCA)

- সবশেষে, যদি মান শূন্য আসে, তাহলে বুঝে নিতে হবে যে এই দুটি ডাইমেনশনের মধ্যে আসলে কোনো ধরনের সম্পর্ক নেই, তারা প্রস্পর স্বাধীন বা ইনডিপেনডেন্ট (Independent)।

আরেকটি বিষয় মনে রাখতে হবে, যদি আমরা  $X$  ডাইমেনশনের সঙ্গে এর নিজেরই কোভ্যারিয়েন্স হিসাব করি, অর্থাৎ যদি  $Cov(X, X)$  নিই, তাহলে আমরা এই  $X$  ডাইমেনশন বরাবর ভ্যারিয়েন্স পেয়ে যাব।

এখন আসি কোভ্যারিয়েন্স ম্যাট্রিক্স (Covariance Matrix)-এ। আমরা ইতিমধ্যেই জা যে, কোভ্যারিয়েন্স ম্যাট্রিক্স শুধু দুটি ডাইমেনশনের মধ্যেকার সম্পর্ক নিয়ে কাজ করে। এখন, ধরা যাক, আমাদের ডাইমেনশন আছে তিনটি কিংবা তারও বেশি। তখন কী করা? তখন সবগুলো ডাইমেনশনের মধ্যে থেকে দুটি দুটি করে নিয়ে আমাদেরকে তাদের কোভ্যারিয়েন্স বের করতে হবে এবং সেগুলো নিয়ে কাজ করতে হবে। যখন, এমন অবস্থা থাকবে, তখন আসলে কোভ্যারিয়েন্স ম্যাট্রিক্স তৈরি করে নিতে হয় এবং তাতে প্রতি জোড়া ডাইমেনশনের মধ্যেকার কোভ্যারিয়েন্সের মান রাখতে হয়।

যেমন, ধরা যাক, আমাদের ডাইমেনশন এখন তিনটি -  $X$ ,  $Y$  এবং  $Z$ । তাহলে আমাদের কোভ্যারিয়েন্স ম্যাট্রিক্স হবে এরকম -

$$C = \begin{pmatrix} Cov(X, X) & Cov(X, Y) & Cov(X, Z) \\ Cov(Y, X) & Cov(Y, Y) & Cov(Y, Z) \\ Cov(Z, X) & Cov(Z, Y) & Cov(Z, Z) \end{pmatrix}$$

এখানে উল্লেখ্য যে  $Cov(X, Y)$  ও  $Cov(Y, X)$  আসলে একই মান।

## পরিচ্ছেদ ১০.৩ : আইগেনভ্যালু (Eigenvalue) ও আইগেনভেক্টর (Eigenvectors)

পরিশেষে, পিসিএ পড়ার আগে আমাদের শেষ আরেকটি বিষয় একটু জানতে হবে এবং এটিই হচ্ছে সবচেয়ে গুরুত্বপূর্ণ বিষয়। এটি ভালোভাবে বুঝতে হবে। এটি হলো আইগেনভেক্টর (Eigenvector) ও আইগেনভ্যালু (Eigenvalues)। আগে খুব সহজ করে ধারণাটি দিই, এরপরে কীভাবে এ দুটি হিসাব করতে হয় তা বর্ণনা করছি।

নিচের ম্যাট্রিক্স গুণনটি দেখি -

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

এখানে আমরা যেটি করলাম, তা হলো একটি  $2 \times 2$  ম্যাট্রিক্সকে একটি কলাম ভেট্টর দিয়ে গুণ করলাম, করে আমরা  $\begin{pmatrix} 12 \\ 8 \end{pmatrix}$  পেলাম। এতটুকু পর্যন্ত খুব সহজ, সাধারণ ম্যাট্রিক্স গুণনের সূত্র। পরবর্তী সময়ে দেখুন, এই ফলাফলটিকে আমরা আবার আমাদের প্রথম যেই কলাম ভেট্টর ছিল  $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$  মেট্রি একটি পূর্ণ সংখ্যার গুণিতক আকারে লিখতে পারলাম, অর্থাৎ এই পুরোনো কলাম ভেট্টরটিকে একটি ইন্টিজার দিয়ে গুণ করেই আমরা আমাদের ফলাফল  $\begin{pmatrix} 12 \\ 8 \end{pmatrix}$  পেয়ে যাচ্ছি। এরকম ক্ষেত্রে,  $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ -কে বলা হবে এই  $\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$  ম্যাট্রিক্সের একটি আইগেনভেট্টর এবং  $4$ -কে বলা হবে  $\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$  ম্যাট্রিক্সের একটি আইগেনভ্যালু।

### আইগেনভেট্টরের কিছু গুরুত্বপূর্ণ ধর্ম –

- আইগেনভেট্টর সব সময়  $n \times n$  অর্থাৎ বর্গ বা স্কয়ার ম্যাট্রিক্সের থাকে।
- সব স্কয়ার ম্যাট্রিক্সের আইগেনভেট্টর থাকে না।
- কোনো  $n \times n$  ম্যাট্রিক্সের যদি আইগেনভেট্টর থাকে, তাহলে আইগেনভেট্টরের সংখ্যা হবে  $n$ -সংখ্যক, অর্থাৎ যদি  $3 \times 3$  ম্যাট্রিক্স হয়, তাহলে তার ৩টি আইগেনভেট্টর থাকতে পারে।
- প্রতিটি আইগেনভেট্টর একে অপরের ওপরে লম্ব (Perpendicular)।
- প্রতিটি আইগেনভেট্টরের সঙ্গে একটিমাত্র আইগেনভ্যালু জড়িত থাকবে।

এই গেল মোটামুটি আইগেনভেট্টর ও আইগেনভ্যালু সম্পর্কে একটি প্রাথমিক ধারণা। এগুলো লিনিয়ার অ্যালজেব্রা কোর্সে আরো বিষদভাবে জানতে পারা যাবে, যেটি আপাতত আমাদের এই বইয়ে সংযুক্ত করা হচ্ছে না।

এখন, কীভাবে আইগেনভ্যালু ও আইগেনভেট্টর বের করতে হয় সেটি একটু হাতে-কলমে দেখা যাক। একটি কথা মনে রাখতে হবে যে,  $2 \times 2$  কিংবা  $3 \times 3$  ম্যাট্রিক্সের আইগেনভেট্টর ও আইগেনভ্যালু হয়তো হাতে-কলমে কিংবা কোড করে বের করা সম্ভব, কিন্তু এর চেয়ে উচ্চতর ডাইমেনশনের ম্যাট্রিক্সের আইগেনভ্যালু ও আইগেনভেট্টর বের করতে হলে অবশ্যই উচিত হবে কোনো সফটওয়্যার লাইব্রেরি কিংবা প্যাকেজ ব্যবহার করা। কেননা এগুলো হাতে-কলমে বের করা একেবারেই অপ্রয়োজনীয় এবং কষ্টসাধ্যও বটে। আমি শুধু বোঝার সুবিধার্থে এখানে একটি  $2 \times 2$  ম্যাট্রিক্সের আইগেনভেট্টর ও আইগেনভ্যালু বের করে দেখাচ্ছি।

ধরা যাক, আমরা নিচের ম্যাট্রিক্সের আইগেনভেট্টর ও আইগেনভ্যালু বের করব –

$$A = \begin{pmatrix} 7 & 3 \\ 3 & -1 \end{pmatrix}$$

অধ্যায় ১০ : প্রিসিপাল কম্পোনেন্ট অ্যানালাইসিস (Principal Component Analysis - PCA)

- প্রথমে ধরে নিতে হবে যে আইগেনভ্যালু হচ্ছে  $\lambda$  এবং একটি  $2 \times 2$  আইডেন্টিটি ম্যাট্রিক্স (Identity Matrix) দিয়ে গুণ করতে হবে, যেহেতু আমাদের A ম্যাট্রিক্সটি  $2 \times 2$  সাইজের। তাহলে আমরা পাই -

$$\lambda I = \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

- এরপরে আমাদের বের করতে হবে,

$$A - \lambda I = \begin{pmatrix} 7 & 3 \\ 3 & -1 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} = \begin{pmatrix} 7 - \lambda & 3 \\ 3 & -1 - \lambda \end{pmatrix}$$

- তৃতীয় ধাপে এসে, আমাদের বের করতে হবে  $\det(A - \lambda I)$ , অর্থাৎ  $\begin{pmatrix} 7 - \lambda & 3 \\ 3 & -1 - \lambda \end{pmatrix}$  এর নির্ণয়ক। এটি আমরা সবাইই করেছি ছোটোবেলায়। নির্ণয়কটি দাঁড়ায়,

$$(7 - \lambda)(-1 - \lambda) - 3 \cdot 3 = \lambda^2 - 6\lambda - 16$$

- আমাদের এখন  $\lambda^2 - 6\lambda - 16 = 0$  সমীকরণটি সমাধান করতে হবে। সমাধান করে পাই,  
 $\lambda = 8, -2$
- এই ৮ এবং -2 ই হচ্ছে আমাদের  $2 \times 2$  ম্যাট্রিক্সের আইগেনভ্যালু। এখন শুধু আইগেনভ্যেটর  
 বের করা বাকি।

- $\lambda = 8$  ধরে পাই,  $\begin{pmatrix} 7 - \lambda & 3 \\ 3 & -1 - \lambda \end{pmatrix} = \begin{pmatrix} 7 - 8 & 3 \\ 3 & -1 - 8 \end{pmatrix} = \begin{pmatrix} -1 & 3 \\ 3 & -9 \end{pmatrix}$

- এখন ধরি,  $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$  হচ্ছে আমাদের আইগেনভ্যেটর, যার জন্য আইগেনভ্যালু হচ্ছে 8।  
 সুতরাং এখন আমরা  $\begin{pmatrix} -1 & 3 \\ 3 & -9 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$  এই সমীকরণটি সমাধান করলেই X এর  
 মান পেয়ে যাব।

- $\begin{pmatrix} -1 & 3 \\ 3 & -9 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$  এখান থেকে আমরা পাই দুটি সমীকরণ -

$$-x_1 + 3x_2 = 0$$

$$3x_1 - 9x_2 = 0$$

- এখন এই দুটি সমীকরণ সমাধান করে আমরা খুব সহজেই পাই,  $x_1 = 3$  এবং  $x_2 = 1$ ।
- সুতরাং প্রথম আইগেনভ্যালু 8-এর জন্য আইগেনভ্যেটর হচ্ছে  $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ ।
- একই পদ্ধতিতে আমরা আইগেনভ্যালু -2-এর জন্য সমাধান করে পাই, আইগেনভ্যেটর হচ্ছে  $\begin{pmatrix} 1 \\ -3 \end{pmatrix}$ ।

তাহলে আমরা পেয়ে গেলাম আমাদের  $\begin{pmatrix} 7 & 3 \\ 3 & -1 \end{pmatrix}$  ম্যাট্রিক্সের দুটি আইগেনভ্যেটর এবং  
 আইগেনভ্যালু। এরপর আমরা সরাসরি চলে যাব আমাদের পিসিএ-তে।

## পরিচ্ছেদ 10.8 : কীভাবে প্রিলিপাল কম্পোনেন্ট অ্যানালাইসিস করতে হয়

এবার আমরা দেখব কীভাবে ধাপে ধাপে পিসিএ প্রয়োগ করে, ডেটাসেটের ডাইমেনশন কমিয়ে আনা যায়।

ধরা যাক, আমাদের নতুন ডেটাসেট এরকম :

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1.0	1.1
1.5	1.6
1.1	0.9

টেবিল 10.4.1

ডেটাসেটের দুটি ডাইমেনশন হচ্ছে X ও Y। আমরা এখন নিচের ধাপগুলো অনুসরণ করব :

- প্রথমেই আমরা যেটি করব, সেটি হচ্ছে ডেটা অ্যাডজাস্টমেন্ট। এই ধাপে, আমরা মূলত প্রতিটি ডাইমেনশনের সব ডেটা থেকে ওই ডাইমেনশনের গড়মান বিয়োগ করে দেব। অর্থাৎ, X-এর প্রতিটি মান থেকে এদের গড়মান 1.81 বিয়োগ করে দেব, একইভাবে Y-এর প্রতিটি মান থেকে এদের গড়মান 1.91 বিয়োগ করে দেব। এর ফলে, X ও Y ডাইমেনশনের ডেটার গড়মান 0 হয়ে যাবে।

এখন তাহলে, নতুন ডেটাসেট দাঁড়াবে এরকম :

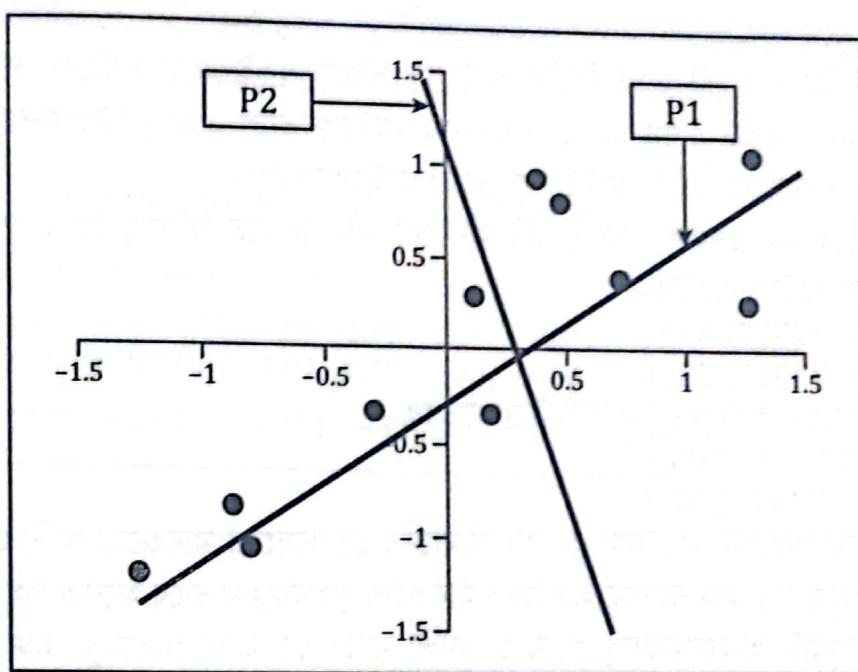
X	Y
0.69	0.49
-1.31	-1.21
0.39	0.99

অধ্যায় ১০ : প্রিসিপাল কম্পোনেন্ট অ্যানালাইসিস (Principal Component Analysis - PCA)

0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01

টেবিল 10.4.2

মানগুলোকে গ্রাফ 10.4.1-এ দেখানো হলো :



গ্রাফ 10.4.1

- এর পরের ধাপে আমরা এই ডেটাসেটের জন্য কোভ্যারিয়েল ম্যাট্রিক্স তৈরি করব। কোভ্যারিয়েল ম্যাট্রিক্স কীভাবে তৈরি করতে হয়, সে সম্পর্কে আমরা আগেই বলেছি। আমাদের এই ডেটাসেটের ক্ষেত্রে, কোভ্যারিয়েল ম্যাট্রিক্সটি হবে –

$$Cov = \begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{pmatrix}$$

এই কোভ্যারিয়েল ম্যাট্রিক্স থেকে আমরা দেখতে পাচ্ছি যে,  $Cov(X, Y)$ -এর মান ধনাত্মক। সূতরাং বলা যায়  $X$ -এর মান বাড়ার সঙ্গে সঙ্গে  $Y$ -এর মান বাড়ছে।

- এর পরের ধাপে আমাদের এই কোভ্যারিয়েজ ম্যাট্রিক্সের আইগেনভেক্টর ও আইগেনভ্যালু বের করতে হবে। আমরা ইতিমধ্যেই দেখে ফেলেছি কীভাবে আইগেনভেক্টর ও আইগেনভ্যালু বের করতে হয়। খুব ভালো হয় যদি আপনারা হাতে-কলমে বা কোড করে এগুলো বের না করে সরাসরি লিনিয়ার অ্যালজেব্রার কোনো লাইব্রেরি বা মডিউল ব্যবহার করেন। তাহলে মানগুলো একেবারে নিখুঁত আসবে। তবে চাইলে হাত পাকানোর জন্য পরীক্ষামূলকভাবে হাতে-কলমে করে দেখতে পারেন।
- আমি এখানে সরাসরই লিখে দিচ্ছি মানগুলো –

$$\text{আইগেনভ্যালু, } \begin{pmatrix} 0.490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{আইগেনভেক্টর, } \begin{pmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix}$$

এখানে বলে রাখা ভালো, আইগেনভেক্টর ম্যাট্রিক্সের প্রথম কলামটি হচ্ছে প্রথম আইগেনভেক্টর এবং দ্বিতীয় কলামটি হচ্ছে দ্বিতীয় আইগেনভেক্টর। একইভাবে, আইগেনভ্যালু ম্যাট্রিক্সের প্রথম মানটি হচ্ছে প্রথম আইগেনভেক্টরের আইগেনভ্যালু এবং আইগেনভ্যালু ম্যাট্রিক্সের দ্বিতীয় মানটি হচ্ছে দ্বিতীয় আইগেনভেক্টরের আইগেনভ্যালু।

- গ্রাফ 10.4.1-এ খেয়াল করলে, দুটি বাঁকানো অক্ষ দেখতে পাবেন, P1 ও P2। P1 হচ্ছে আমাদের আইগেনভেক্টর –

$$\begin{pmatrix} -0.677873399 \\ -0.735178656 \end{pmatrix}$$

এবং P2 হচ্ছে আমাদের অপর আইগেনভেক্টর –

$$\begin{pmatrix} -0.735178656 \\ 0.677873399 \end{pmatrix}$$

আরো লক্ষ করবেন, P1 অক্ষ বরাবর আমাদের ডেটাসেটের মানগুলো বেশি ছড়ানো, কিন্তু সেই তুলনায় P2 অক্ষ বরাবর আমাদের মানগুলো তুলনামূলকভাবে অনেক কম ছড়ানো।

- এই ব্যাপারটি আইগেনভ্যালু থেকেও বোৰা যায়। যে আইগেনভ্যালুর মান বড়ো, সেই আইগেনভ্যালুর সঙ্গে সম্পৃক্ত আইগেনভেক্টরের অক্ষ বরাবর ডেটা বেশি ছড়ানো থাকবে।
- আইগেনভ্যালু দুটির মধ্যে, যেটির মান বড়ো (এখানে দ্বিতীয়টি), সেই আইগেনভ্যালুটি নিয়েই আমরা কাজ করব। যদি আরেকটু জেনারালাইজ করে বলি, তাহলে আমরা যতগুলো আইগেনভ্যালু পাব, সেগুলোকে বড়ো থেকে ছোটো এই ক্রমে সাজাতে হবে। এরপর যদি ইচ্ছা হয়, তাহলে আমরা এখান থেকে যেই আইগেনভ্যালুগুলো কম গুরুত্বপূর্ণ অর্থাৎ যাদের মান ছোটো, চাইলে সেগুলো বাদ দিয়ে বড়ো মানবিশিষ্ট আইগেনভ্যালুগুলো শুধু রেখে দিতে পারি। এতে যদিও আমাদের কিছু তথ্য হারিয়ে যাবে বা ইনফরমেশন লস (Information Loss) হবে, কিন্তু আমাদের ডেটার ডাইমেনশন কমে আসবে এবং শুধু সেই গুরুত্বপূর্ণ ডাইমেনশনের ডেটাই থাকবে যেগুলো ওই ডেটাসেটকে উপস্থাপন করার জন্য যথেষ্ট।

## অধ্যায় ১০ : প্রিসিপাল কম্পোনেন্ট আনালাইসিস (Principal Component Analysis - PCA)

আমাদের এই ক্ষেত্রে আমরা ধরে নিচ্ছি যে, আমরা দ্বিতীয় আইগেনভ্যালুটি রেখে দিয়ে প্রথমটি রাখ দিয়ে দেব।

এখন আমাদের একটি ফিচার ডেটার তৈরি করতে হবে, যেটি আসলে আমরা যে-যে আইগেনভ্যালু নিয়ে কাজ করব বলে সিদ্ধান্ত নিয়েছি, সেসব আইগেনভ্যালুর সঙ্গে সেসব আইগেনভেক্টরের জড়িত, সেগুলো নিয়ে তৈরি করা একটি ডেটার। আমাদের এই ক্ষেত্রে আমরা শুধু দ্বিতীয় আইগেনভেক্টরটি নিয়েই আমাদের ফিচার ডেটার তৈরি করব। তাহলে আমাদের ফিচার ডেটারটি হবে –

$$(-0.677873399) \\ (-0.735178656)$$

এর পরে আমাদের শেষ ধাপ। সেটি হচ্ছে, আমাদের এখন নতুন ডেটা পেতে হবে যে ডেটার ডাইমেনশন আমাদের অরিজিনাল ডেটাসেটের চেয়ে কম হবে। আমাদের কিন্তু মূল লক্ষ্যই ছিল এটি যে, আমরা আমাদের ডেটাসেটের ডেটার ডাইমেনশন কমিয়ে নিয়ে আসব, যাতে আমাদের কম্পিউটেশনাল কমপ্লেক্সিটি (Computational Complexity) কমে আসে।

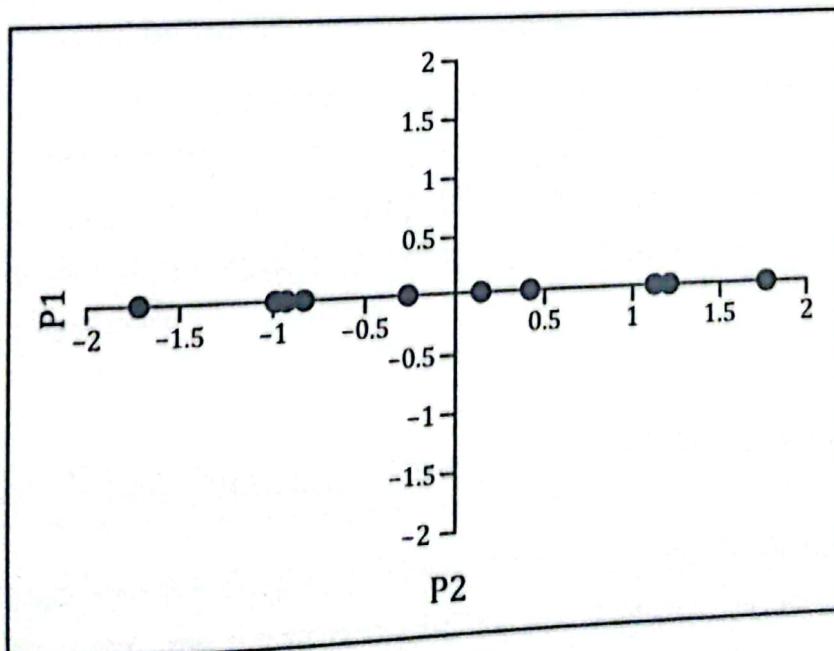
নতুন ডেটা পাওয়ার সূত্র হচ্ছে –

$$\text{Final Data} = (\text{Feature Vector})^T \times (\text{Data Adjust})^T$$

$$= (-0.677873399 \quad -0.735178656) \times$$

$$(0.69 \quad -1.31 \quad 0.39 \quad 0.09 \quad 1.29 \quad 0.49 \quad 0.19 \quad -0.81 \quad -0.31 \quad -0.71) \\ (0.49 \quad -1.21 \quad 0.99 \quad 0.29 \quad 1.09 \quad 0.79 \quad -3.1 \quad -0.81 \quad -0.31 \quad -1.01)$$

$$= (-0.827 \quad 1.777 \quad -0.992 \quad -0.274 \quad -1.675 \quad -0.912 \quad 0.099 \quad 1.144 \quad 0.438 \quad 1.223)$$



গ্রাফ 10.4.2

এখানে যেটি করা হলো যে, আমাদের টেটাসেটকে তবু P1 অক্ষের ওপরে প্রোটোট (Project) করা হলো এবং এর P2 অক্ষ বরাবর ওই ডাইমেনশনে যাবতীয় যা টেটা ছিল, সব মুছে ফেল হলো।

আমরা যদি এখন আমাদের P1 অক্ষকে একটি রোটেট (Rotate) করে X-অক্ষের ওপর প্রতিষ্ঠাপন করি এবং সেইসঙ্গে P2 অক্ষকে Y-অক্ষের ওপর প্রতিষ্ঠাপন করি, তাহলে আমরা আমাদের সর্বশেষ টেটা পাই এরকম (ঋক 10.4.2)।

এই ভেটাই আমাদের নতুন টেটাসেট যার ডাইমেনশন একটিই। এই টেটাসেট দিয়েই প্রথম সময়ে আমাদের যাবতীয় মেশিন লার্নিংয়ের কাজ করতে হবে।

এখানে আরেকটি বিষয় উল্লেখ্য যে, পিসিএ আমরা টেটার ডাইমেনশন কমানোর কাজে ব্যবহার করা ছাড়াও ফিচার এক্সট্রাকশন (Feature Extraction)-এর কাজেও ব্যবহার করতে পারি। তার বিস্তারিত আলোচনায় গেলাম না, কিন্তু জেনে রাখাটা ভালো।