
Slide 1: Introduction

Full Script (10+ lines):

“আমরা যখন high-dimensional data নিয়ে কাজ করি — যেমন image, genomic data, বা sensor readings — তখন একটা বড় চ্যালেঞ্জ হয়: কীভাবে এই data কে efficiently process, visualize, এবং interpret করা যায়।

High-dimensional space-এ data visualization প্রায় অসম্ভব, কারণ আমরা 3D-এর বাইরে দেখতে পারি না।

Dimensionality reduction techniques আমাদের এই সমস্যার সমাধান দেয় — যেখানে data কে lower-dimensional space-এ project করা হয়, যাতে তার মূল structure বজায় থাকে।

PCA বা Principal Component Analysis হলো সবচেয়ে জনপ্রিয় dimensionality reduction technique। এটি data-এর variance capture করে এবং informative directions বের করে।

কিন্তু traditional PCA-তে eigen-decomposition ব্যবহার হয় — যা computationally expensive, বিশেষ করে যখন dataset বড় হয়।

এই কারণে আমি PCA-এর একটি variant — QR-based PCA — implement করেছি।

QR decomposition numerically stable এবং faster, এবং এটি eigen-decomposition avoid করে।

এই প্রজেক্টে আমি QR-based PCA ব্যবহার করে Digits dataset-এ dimensionality reduction করেছি।

আমি evaluate করেছি visualization, reconstruction, এবং classification performance — যাতে বোঝা যায় এই approach কতটা কার্যকর।

এই presentation-এ আমি ধাপে ধাপে দেখাবো কীভাবে QR-based PCA কাজ করে, এবং কেন এটা traditional PCA-এর তুলনায় computationally advantageous।”

তার উপর বায়টিক বল (গোলাপি ভীত) প্রদোপ হচ্ছে
কিন্তু পাছের দিক বদলায়নি — সে এখানে সোজা-eigenvector
আর এই যে লম্বা বা খাটো হয় সেটা value

Slide 2: Problem Statement

Full Script (10+ lines):

“আমার প্রজেক্টের মূল সমস্যা ছিল — কীভাবে high-dimensional digit image data কে efficiently compress করা যায়, যাতে visualization এবং classification সহজ হয়।

Traditional PCA ব্যবহার করলে eigen-decomposition করতে হয়, যা time-consuming এবং numerically unstable হতে পারে large datasets-এর ক্ষেত্রে।

আমি চেয়েছি এমন একটি approach, যা faster, scalable, এবং theoretically sound হয়।

QR decomposition হলো এমন একটি technique, যা matrix কে orthonormal basis এবং upper triangular matrix-এ ভেঙে দেয় — এবং এটি eigen-decomposition avoid করে।

আমি এই QR-based PCA মডেল তৈরি করেছি, যেখানে আমি QR apply করেছি centered data-এর transpose-এর উপর, এবং তারপর SVD ব্যবহার করে principal directions বের করেছি।

এই মডেলটি আমি Digits dataset-এ apply করেছি — যাতে দেখা যায় compressed data দিয়ে visualization, reconstruction এবং classification কতটা ভালো হয়।

আমি শুধু theoretical efficiency নয়, practical performance-ও evaluate করেছি।

Reconstruction error, explained variance, এবং classification accuracy — সব কিছু বিশ্লেষণ করেছি।

এই প্রজেক্টের মাধ্যমে আমি দেখাতে চেয়েছি যে QR-based PCA শুধু faster নয়, বরং interpretability এবং accuracy-র দিক থেকেও শক্তিশালী।

এখন আমি dataset সম্পর্কে বলবো — যেটা এই মডেল evaluate করার জন্য ব্যবহার করেছি।”

PCA-এর জন্য

PCA-তে আমরা orthonormal basis চাই — QR decomposition সেট efficiently দেয় —
mainly Identity matrix

Scaling + rotation info ----> Upper triangular



Slide 3: Dataset Overview

🗣️ Full Script (10+ lines):

“আমি sklearn-এর Digits dataset ব্যবহার করেছি — যা machine learning-এর জন্য একটি benchmark dataset।

এতে 1797টি grayscale image আছে, প্রতিটি 8×8 pixel।

প্রতিটি image কে flatten করে 64-dimensional feature vector বানানো হয়েছে — অর্থাৎ প্রতিটি sample হলো একটি 64-feature vector।

Labels 0 থেকে 9 পর্যন্ত digit — এবং dataset balanced, অর্থাৎ প্রতিটি digit-এর sample প্রায় সমান।

এই dataset visualization এবং classification-এর জন্য খুবই উপযোগী, কারণ images ছোট এবং human-interpretable।

আমি এই dataset কে mean center করেছি, যাতে PCA ঠিকভাবে variance capture করতে পারে।

আমি কিছু sample image দেখিয়েছি — যাতে audience বুঝতে পারে data দেখতে কেমন।

এই dataset-এর simplicity আমাকে allow করেছে QR-based PCA-এর core idea demonstrate করতে।

যদিও dataset ছোট, কিন্তু dimensionality যথেষ্ট — 64 dimension — যাতে PCA-এর utility বোঝা যায়।

এখন আমি দেখাবো কীভাবে আমি preprocessing করেছি এবং QR decomposition apply করেছি।”



Slide 4: Preprocessing – Mean Centering

🗣️ Full Script (10+ lines):

“PCA করার আগে একটি গুরুত্বপূর্ণ ধাপ হলো mean centering।

আমরা জানি PCA মূলত data-এর variance capture করে — অর্থাৎ কোন direction-এ data সবচেয়ে বেশি ছড়ানো, সেটা খুঁজে বের করে।

কিন্তু যদি data mean-centered না হয়, তাহলে PCA ভুলভাবে biased directions choose করতে পারে।

তাই আমি প্রথমে প্রতিটি feature-এর mean বের করেছি এবং সেই mean subtract করেছি — যাতে data-এর center হয় origin-এর কাছাকাছি।

এই process কে বলা হয় mean centering বা zero-centering।

এটা mathematicalভাবে খুব simple — $X_{\text{centered}} = X - X_{\text{mean}}$ — কিন্তু PCA-এর accuracy এবং interpretability-এর জন্য অত্যন্ত গুরুত্বপূর্ণ।

Mean center করার ফলে QR decomposition এবং SVD-এর output আরও meaningful হয়।

আমি এই centered data-এর transpose নিয়ে QR decomposition apply করেছি — কারণ PCA-তে আমরা covariance matrix বা its equivalent decomposition ব্যবহার করি।

এই preprocessing step না করলে principal components biased হয়ে যেত এবং reconstruction error বেড়ে যেত।

এখন আমি দেখাবো কীভাবে আমি QR decomposition apply করেছি এবং কেন এটা traditional eigen-decomposition-এর চেয়ে efficient।”

কোন direction-এ ডেটার variance সবচেয়ে বেশি?
কোন feature combination সবচেয়ে informative?
এই প্রশ্নের উত্তর দেয় covariance matrix।



Slide 5: QR Decomposition

🔗 Full Script (10+ lines):

“QR decomposition হলো linear algebra-এর একটি fundamental technique — যা একটি matrix কে দুইটি matrix-এ ভেঙে দেয়:

- Q: orthonormal basis
- R: upper triangular matrix

আমি centered data-এর transpose matrix $H = X_{\text{centered}}.T$ এর উপর QR decomposition apply করেছি।

এর ফলে আমি পেয়েছি Q_1 এবং R_1 — যেখানে Q_1 হলো (64×64) dimension-এর orthonormal matrix, এবং R_1 হলো (64×1797) dimension-এর upper triangular matrix।

এই decomposition traditional PCA-এর eigen-decomposition avoid করে — যা computationally heavy এবং numerically unstable হতে পারে।

QR decomposition faster এবং scalable — বিশেষ করে যখন dataset বড় হয়।

আমি `np.linalg.qr(H, mode='reduced')` ব্যবহার করেছি — যাতে unnecessary zero-padding avoid হয় এবং memory-efficient result পাওয়া যায়।

এই decomposition-এর মাধ্যমে আমি data-এর column space-এর orthonormal basis পেয়েছি — যেটা PCA-এর জন্য foundation তৈরি করে।

এখন আমি এই $R1.T$ matrix-এর উপর SVD apply করবো — যাতে principal directions বের করা যায়।”

Slide 6: SVD on $R1.T$ – Extracting Principal Directions

 Full Script (10+ lines):

“QR decomposition করার পর আমি $R1.T$ matrix-এর উপর Singular Value Decomposition (SVD) apply করেছি।

SVD হলো PCA-এর core mathematical tool — যা একটি matrix কে তিনটি matrix-এ ভেঙে দেয়:

- U : left singular vectors
- Σ : singular values (diagonal matrix)
- V^T : right singular vectors

আমি এখানে শুধু singular values এবং V^T ব্যবহার করেছি — কারণ PCA-তে আমরা principal directions হিসেবে right singular vectors ব্যবহার করি।

Singular values-এর square হলো eigenvalues — যেগুলো variance represent করে।

আমি `np.linalg.svd(R1.T, full_matrices=False)` ব্যবহার করেছি — যাতে compact decomposition পাওয়া যায়।

এই decomposition থেকে আমি পেয়েছি $D1$ (singular values) এবং V^T (eigenvectors)।

আমি top h eigenvectors select করেছি — যেগুলো সবচেয়ে বেশি variance capture করে।

এই eigenvectors কে আমি $Q1$ matrix দিয়ে transform করেছি — যাতে final principal directions পাওয়া যায়:

$\Phi = Q1 @ Vh$

এই Φ হলো সেই basis যেখানে আমি data project করবো — এবং PCA visualization, reconstruction, এবং classification করবো।”

Slide 7: Principal Components Construction

 Full Script (10+ lines):

“আমরা এখন PCA-এর সবচেয়ে গুরুত্বপূর্ণ ধাপে এসেছি — principal components তৈরি করা।

আমি আগেই বলেছি, Φ matrix হলো final basis — যেটা আমি পেয়েছি $\Phi = Q1 @ Vh$ দিয়ে।

এই Φ matrix-এর প্রতিটি column হলো একটি principal direction — অর্থাৎ data-এর এমন direction যেখানে variance সবচেয়ে বেশি।

আমি top h components select করেছি — যেখানে $h = 2$ বা $h = 3$, depending on visualization needs।

এই components capture করে maximum variance — এবং lower-dimensional space-এ meaningful representation দেয়।

Principal components তৈরি করার পর আমি original data `X_centered` কে Φ basis-এ project করেছি:

```
Z = X_centered @ Phi[:, :h]
```

এই Z হলো projected data — যেটা lower-dimensional space-এ data-এর compressed version।

এই projection-এর মাধ্যমে আমি dimensionality reduce করেছি — কিন্তু essential structure retain করেছি।

এখন আমি এই Z data-কে ব্যবহার করবো visualization এবং clustering analysis-এর জন্য।”

Slide 8: Projection onto Principal Components

 Full Script (10+ lines):

“Projection হলো PCA-এর practical output — যেখানে আমরা original high-dimensional data কে lower-dimensional space-এ map করি।

আমি `Z = X_centered @ Phi[:, :h]` ব্যবহার করে projection করেছি — যেখানে $h = 2$ বা 3 । এই projected data retain করে original data-এর structure — যেমন clusters, separability, এবং class-wise distribution।

Projection-এর মাধ্যমে আমি দেখতে পারি কোন digit classes একে অপরের কাছাকাছি, আর কোনগুলো দূরে।

এই projected data visualization-friendly — কারণ আমরা এখন 2D বা 3D scatter plot তৈরি করতে পারি।

আমি এখানে digit labels ব্যবহার করেছি — যাতে প্রতিটি point-এর class বোঝা যায়।

Projection-এর মাধ্যমে আমরা not only visualize করতে পারি, বরং classification, clustering, এবং anomaly detection-এর মতো কাজও করতে পারি।

এই step-এর efficiency QR-based PCA-এর computational advantage দেখায় — কারণ আমরা faster decomposition এবং projection পেয়েছি।

এখন আমি দেখাবো 2D scatter plot — যেখানে digit-wise clusters annotate করা হয়েছে।”

Slide 9: 2D Visualization with Cluster Annotations

 Full Script (10+ lines):

“এই slide-এ আমি দেখাচ্ছি PCA projection-এর 2D visualization — যেখানে আমরা digit dataset-এর structure দেখতে পাচ্ছি।
আমি projected data z কে scatter plot-এ visualize করেছি — এবং প্রতিটি point-এর digit label color-coded করেছি।
Visualization-এর জন্য আমি `matplotlib` এবং `seaborn` ব্যবহার করেছি — যাতে plot visually appealing হয়।
আমি cluster annotations যোগ করেছি — যাতে বোঝা যায় কোন digit classes একে অপরের কাছাকাছি বা overlap করছে।
উদাহরণস্বরূপ, digit 1 এবং 7 অনেক সময় overlap করে — কারণ তাদের strokes visual similarity রাখে।
অন্যদিকে, digit 0 এবং 8 সাধারণত well-separated থাকে — কারণ তাদের shape distinctive।
আমি explained variance plot-ও তৈরি করেছি — যাতে বোঝা যায় top components কতটা variance capture করছে।
এই visualization interpretability বাড়ায় এবং model-এর performance বুঝতে সাহায্য করে।
Overall, QR-based PCA আমাকে faster computation এবং clearer visualization দিয়েছে — যা traditional PCA-এর তুলনায় অনেক efficient।”



Slide 10: Explained Variance Analysis

🔗 Full Script (10+ lines):

“PCA-এর একটি গুরুত্বপূর্ণ অংশ হলো explained variance — অর্থাৎ প্রতিটি principal component data-এর কতটা variance capture করছে।
আমি SVD থেকে পাওয়া singular values $D1$ ব্যবহার করে explained variance হিসাব করেছি:
`explained_variance = (D1 ** 2) / np.sum(D1 ** 2)`
এই formula অনুযায়ী, প্রতিটি component-এর eigenvalue proportionally দেখায় কতটা information retain হয়েছে।
আমি cumulative explained variance plot তৈরি করেছি — যাতে বোঝা যায় কতগুলো component ব্যবহার করলে total variance-এর কত শতাংশ পাওয়া যায়।
উদাহরণস্বরূপ, প্রথম 2 component যদি 60% variance capture করে, তাহলে visualization-এর জন্য এগুলো যথেষ্ট।
আমি এই plot-এ threshold line দিয়েছি — যেমন 90% বা 95% — যাতে component selection সহজ হয়।
Explained variance plot model interpretability বাড়ায় এবং dimensionality reduction-এর trade-off বোঝায়।
এই analysis QR-based PCA-এর efficiency validate করে — কারণ fewer components দিয়েই high variance capture করা সম্ভব হয়েছে।
এখন আমি reconstruction process দেখাবো — যেখানে original data কে compressed representation থেকে পুনরুদ্ধার করা হয়।”



Slide 11: Reconstruction from Principal Components

🔗 Full Script (10+ lines):

“PCA-এর আরেকটি গুরুত্বপূর্ণ ব্যবহার হলো reconstruction — অর্থাৎ compressed data থেকে original data approximate করা।

আমি projected data z কে Φ basis দিয়ে inverse transform করেছি:

```
X_reconstructed = Z @ Phi[:, :h].T + X_mean
```

এখানে z হলো lower-dimensional representation, $\Phi[:, :h]$ হলো principal directions, এবং

X_{mean} হলো original mean।

Reconstruction-এর মাধ্যমে আমি দেখতে পারি PCA কতটা information retain করেছে এবং কতটা হারিয়েছে।

আমি reconstructed images visualize করেছি — যাতে বোঝা যায় digit shapes কতটা accurate আছে।

Lower h value দিলে reconstruction blurry হয় — কারণ কম component কম variance capture করে।

Higher h value দিলে reconstruction sharp হয় — কিন্তু dimensionality বেশি হয়।

এই trade-off বুঝতে reconstruction visualization খুবই কার্যকর।

আমি original এবং reconstructed images side-by-side দেখিয়েছি — যাতে visual comparison সহজ হয়।

এখন আমি দেখাবো evaluation metrics — যেগুলো দিয়ে reconstruction quality এবং PCA performance পরিমাপ করা যায়।”



Slide 12: Evaluation Metrics

🔗 Full Script (10+ lines):

“QR-based PCA-এর performance evaluate করার জন্য আমি কয়েকটি quantitative metrics ব্যবহার করেছি।

প্রথমত, আমি reconstruction error হিসাব করেছি — যা হলো original data এবং reconstructed data-এর মধ্যে Frobenius norm:

```
error = np.linalg.norm(X_original - X_reconstructed, ord='fro')
```

এই error যত কম, reconstruction তত ভালো।

দ্বিতীয়ত, আমি classification accuracy test করেছি — projected data z ব্যবহার করে simple classifier apply করে।

PCA projection যদি class separability retain করে, তাহলে classifier accuracy ভালো হবে।

তৃতীয়ত, আমি runtime এবং memory usage compare করেছি traditional PCA-এর সঙ্গে — যাতে efficiency বোঝা যায়।

QR-based PCA faster এবং numerically stable — বিশেষ করে large dataset-এর ক্ষেত্রে।
আমি এই metrics summary table-এ দেখিয়েছি — যাতে clear comparison পাওয়া যায়।
Overall, QR-based PCA আমাকে দিয়েছে better scalability, faster computation, এবং interpretable visualization।”

✓ Slide 13: Conclusion

🔗 Full Script (10+ lines):

“এই প্রজেক্টে আমি একটি QR-based PCA framework তৈরি করেছি — যা traditional PCA-এর computational bottleneck avoid করে।

আমি Digits dataset-এ এই মডেল apply করেছি এবং তিনটি দিক থেকে এর কার্যকারিতা বিশ্লেষণ করেছি:

- Visualization: projected data-তে digit-wise clusters স্পষ্টভাবে দেখা গেছে
 - Reconstruction: compressed data থেকেও high-fidelity image পুনর্গঠন সম্ভব হয়েছে
 - Classification: reduced data দিয়েও ~96% accuracy পাওয়া গেছে
- QR decomposition এবং SVD ব্যবহার করে আমি principal components efficiently বের করেছি।
- Explained variance plot দেখিয়েছে যে মাত্র 10 component দিয়েই 90% variance retain করা যায়।
- Reconstruction error metrics যেমন RMSE, MAE, এবং R^2 score প্রমাণ করেছে যে model accurate।
- Confusion matrix analysis দেখিয়েছে কোন digit-এ classifier বেশি ভুল করছে — যেমন 5 এবং 9।
- Overall, QR-based PCA একটি scalable, interpretable, এবং efficient dimensionality reduction technique হিসেবে কাজ করেছে।
- এখন আমি এই মডেলের সীমাবদ্ধতা নিয়ে কিছু কথা বলবো।”
-

⚠ Slide 14: Limitations

🔗 Full Script (10+ lines):

“যেকোনো মডেলের মতো, QR-based PCA-এরও কিছু সীমাবদ্ধতা আছে।

প্রথমত, এটি linear technique — অর্থাৎ nonlinear relationships capture করতে পারে না।

যদি dataset-এর structure nonlinear হয় — যেমন curved manifolds — তাহলে kernel PCA বা t-SNE বেশি কার্যকর হতে পারে।

দ্বিতীয়ত, QR decomposition numerical stability দেয়, কিন্তু SVD computationally intensive হতে পারে large matrices-এর ক্ষেত্রে।

তৃতীয়ত, Digits dataset ছোট এবং clean — real-world noisy data-তে performance ভিন্ন হতে পারে।

Reconstruction quality lower h value-এ degrade করে — তাই dimensionality কমালে trade-off আসে।

Classification accuracy high হলেও, complex classifiers ব্যবহার করলে আরও improvement সম্ভব।

Visualization 2D-তে informative হলেও, higher-dimensional structure lose হতে পারে।

আমি এই সীমাবদ্ধতাগুলো acknowledge করেছি — যাতে future work-এ এগুলো address করা যায়।

এখন আমি বলবো কীভাবে এই মডেলকে আরও উন্নত করা যায় এবং কোথায় ব্যবহার করা যেতে পারে।”

Slide 15: Future Work

Full Script (10+ lines):

“এই QR-based PCA framework ভবিষ্যতে আরও অনেকভাবে extend করা যেতে পারে। প্রথমত, আমি larger dataset যেমন MNIST বা CIFAR-10-এ apply করতে চাই — যাতে scalability test করা যায়।

দ্বিতীয়ত, nonlinear PCA techniques যেমন kernel PCA integrate করে complex structure capture করা যেতে পারে।

তৃতীয়ত, reconstruction quality improve করার জন্য autoencoder-based hybrid model consider করা যেতে পারে।

Classification-এর জন্য আমি SVM, Random Forest, বা Neural Network ব্যবহার করে performance compare করতে চাই।

Visualization enhance করার জন্য interactive plotting tools যেমন Plotly বা Bokeh ব্যবহার করা যেতে পারে।

Real-time application-এর জন্য QR-based PCA embedded systems বা edge devices-এ deploy করা যেতে পারে।

আমি চাই এই framework open-source করে অন্য গবেষকরা ব্যবহার করতে পারেন এবং contribute করতে পারেন।

Overall, এই মডেল একটি foundation তৈরি করেছে — যার উপর আরও advanced dimensionality reduction techniques build করা সম্ভব।

এখন আমি আপনাদের প্রশ্নের উত্তর দিতে প্রস্তুত।”

Slide 16: Classification Results

🔗 Full Script (10+ lines):

“আমার QR-based PCA মডেল থেকে পাওয়া compressed data ব্যবহার করে আমি Logistic Regression classifier train করেছি।

আমি train/test split করেছি 70:30 ratio-তে এবং max_iter=1000 দিয়ে model convergence নিশ্চিত করেছি।

Classification performance evaluate করার জন্য আমি ব্যবহার করেছি:

- Accuracy
- Precision
- Recall
- F1 Score
- AUC Score (multi-class)

Accuracy এসেছে প্রায় 96%, যা compressed data দিয়েও strong performance প্রমাণ করে।

Precision এবং Recall দুটোই high — অর্থাৎ model both correctly predicts and misses fewer cases।

F1 Score balance করে precision এবং recall — এবং এখানে সেটা প্রায় 0.95 এর কাছাকাছি।

Multi-class AUC score ব্যবহার করেছি label_binarize() দিয়ে — যাতে class confidence measure করা যায়।

এই ফলাফল দেখায় যে QR-based PCA শুধু visualization বা reconstruction নয়, practical classification task-এও কার্যকর।

এখন আমি দেখাবো confusion matrix — যেখানে বোঝা যাবে কোন digit-এ model বেশি ভুল করেছে।”

🧩 Slide 17: Confusion Matrix Analysis

🔗 Full Script (10+ lines):

“এই slide-এ আমি দেখাচ্ছি confusion matrix — যা classification performance-এর granular view দেয়।

Matrix-এর প্রতিটি cell দেখায় কতবার একটি digit অন্য digit হিসেবে predict হয়েছে।

Diagonal values represent correct predictions — আর off-diagonal values represent misclassifications।

দেখা যাচ্ছে digit 1, 0, এবং 7 খুব ভালোভাবে classify হয়েছে — diagonal values high।

কিন্তু digit 5 এবং 9 কিছুটা overlap করেছে — অর্থাৎ model 5 কে 9 হিসেবে predict করেছে কয়েকবার।

এটা shape similarity-এর কারণে হতে পারে — যেমন 5 এবং 9-এর curvature অনেক সময় classifier confuse করে।

আমি heatmap ব্যবহার করেছি `seaborn.heatmap()` দিয়ে — যাতে visually বোঝা যায় কোন class strong, কোন class weak।
এই matrix future model refinement-এর জন্য খুব গুরুত্বপূর্ণ — কারণ এটা error pattern detect করতে সাহায্য করে।
আমি চাইলে misclassified samples visualize করতে পারি — যাতে বোঝা যায় model কোথায় ভুল করছে।
এখন আমি presentation শেষ করছি এবং আপনাদের প্রশ্নের উত্তর দিতে প্রস্তুত।”

Slide 18: Q&A and Discussion

Full Script (10+ lines):

“এই ছিল আমার QR-based PCA মডেল নিয়ে পুরো বিশ্লেষণ — যেখানে আমি dimensionality reduction, visualization, reconstruction, এবং classification performance দেখিয়েছি।
আমি traditional PCA-এর computational bottleneck avoid করেছি QR decomposition ব্যবহার করে।
আমি explained variance, reconstruction error, এবং classification metrics দিয়ে মডেল validate করেছি।
আমি আশা করি এই মডেল আপনাদের কাছে informative এবং inspiring লেগেছে।
এখন আমি আপনাদের প্রশ্নের উত্তর দিতে প্রস্তুত।
আপনি চাইলে জানতে পারেন:

- QR decomposition কেন PCA-এর জন্য উপযোগী?
 - Reconstruction error কিভাবে interpret করতে হয়?
 - Confusion matrix থেকে কীভাবে model improve করা যায়?
 - এই মডেল larger dataset-এ কেমন কাজ করবে?
- আমি চাই audience actively প্রশ্ন করুক — কারণ discussion থেকেই নতুন idea আসে।
ধন্যবাদ সবাইকে — এখন চল আলোচনা করি!”