Scenario - Set 1

Scenario 1: Selecting Key Features for Diabetes Prediction

Concept: Feature Selection

Question:
A healthcare startup plans to build a diabetes prediction model. The dataset contains 50 features, but the team wants to use only the most relevant ones. How can feature selection be applied effectively?

Solution Approach:

1. Collect patient medical data (BMI, glucose, age, family history, etc.).
2. Preprocess the data: treat missing values & apply proper scaling.
3. Use a correlation heatmap to remove highly correlated features.
4. Apply **SelectKBest** with mutual information to extract the most informative features.
5. Optionally apply **RFE** with Logistic Regression for additional filtering.
6. Train the model using selected features.
7. Evaluate performance using **Accuracy, ROC-AUC & F1-Score**.

Scenario 2: Creating a New Django Project

Concept: Django Project Setup

Question:
Create a new Django project named **student_portal** for managing student data. What is the step-by-step process and basic logic?

Steps:

1. Open Command Prompt/Terminal.
2. Navigate to the folder where the project should be created.
3. Run:

```
django-admin startproject student_portal
```

4. A new folder *student_portal* is created with default Django structure.

Basic Logic for the Project:

- Create an app named `students` to manage student records.
- Define a **Student model** (name, age, DOB, department, marks).

- Create forms & views for:
    - Add Student
    - View Student
    - Update Student
    - Delete Student
- Display data in templates using Django ORM.
- Use Django Admin to manage data easily.

Scenario 3: Feature Reduction in a Marketing Dataset

Concept: Dimensionality Reduction

Question:
A company has 200+ behavioral features to analyze customer purchase patterns. How to reduce dimensionality without losing prediction power?

Steps:

1. Preprocess dataset (handle missing values, normalization).
2. Apply **PCA** to reduce dimensionality while retaining maximum variance.
3. Alternatively apply **Lasso Regression** to eliminate low-impact features.
4. Retain most significant components/features.
5. Build the ML model using the reduced dataset.
6. Evaluate using **RMSE** (Regression) or **Accuracy** (Classification).

Scenario 4: Personalized Book Recommendation System

Concept: Collaborative Filtering

Question:
A digital library wants to recommend books based on similar readers' preferences. How should the system be designed?

Steps:

1. Create **user-item interaction matrix** using book ratings.
2. Calculate similarity using **cosine similarity** or **Pearson correlation**.
3. Recommend books liked by similar users (but not yet read by the target user).
4. Use **KNN or Matrix Factorization** to handle sparse data.
5. Evaluate using **RMSE & Recall@K**.
6. Update recommendations as new ratings are added.

Scenario 5: Credit Risk Feature Selection

Concept: Feature Selection

Question:
A bank wants to use only key indicators to assess credit risk. How to reduce features?

Steps:

1. Collect financial data: income, credit score, employment details, etc.
2. Preprocess: handle missing data & encode categories.
3. Use **Random Forest** or similar tree-based models to calculate feature importance.
4. Remove low-value features.
5. Validate using classifiers like **Decision Tree**.
6. Evaluate using **ROC-AUC, Precision & Recall**.

Scenario 6: News Article Hybrid Recommendation

Concept: Hybrid Recommender System

Question:
A news app wants recommendations based on both article similarity and user reading activity. How to build a hybrid model?

Steps:

1. Collect user interaction data and article metadata.
2. Apply **TF-IDF** for content-based similarity.
3. Build a collaborative filtering model from reading behavior.
4. Combine both using weighted scoring.
5. Evaluate using **CTR & Time Spent**.
6. Retrain periodically as new articles/users join.

Scenario 7: Spam Email Feature Selection

Concept: Feature Selection

Question:
Thousands of text features exist in an email spam dataset. How to select the most important ones?

Steps:

1. Convert emails into **TF-IDF** or **Bag-of-Words** vectors.
2. Apply **Chi-Square Test** for feature reduction.
3. Optionally use **Lasso** for embedded selection.
4. Train a classification model on reduced features.
5. Evaluate using **Precision, Recall & F1-Score**.

Scenario 8: Course Recommendation for Online Learning

Concept: Collaborative Filtering

Question:
How can an ed-tech platform recommend courses based on similar learner enrollments?

Steps:

1. Build a **user-course interaction matrix**.
2. Apply **KNN / SVD** to identify similar learners/courses.
3. Recommend high-rated courses taken by similar learners.
4. Use popularity or content-based filtering for cold-start users.
5. Evaluate using **RMSE, Recall@K & Engagement Rate**.
6. Update periodically based on new interactions.

Scenario 9: Key Feature Identification for Car Price Prediction

Concept: Feature Selection

Question:
With 100+ features, how to simplify model training for car price prediction?

Steps:

1. Collect car data (price, mileage, brand, engine type, etc.).
2. Encode categorical & scale numeric features.
3. Remove redundant features using **correlation analysis**.

4. Rank features using **Gradient Boosting Importance**.
5. Retain top 15–20 features and retrain.
6. Evaluate using **RMSE & R^2**.

Scenario 10: Cold-Start in E-Commerce Recommendation

Concept: Recommendation System

Question:
How to recommend products to new users with no interaction history?

Steps:

1. Use **most popular / trending products** (initial fallback).
2. Use demographics (age, gender, location) to filter products.
3. If interests are collected at signup → apply content-based filtering.
4. Gradually personalize via collaborative filtering once interactions grow.
5. Evaluate based on **click-through rate & engagement**.