ML Scenario based Set-2

---

1. Predicting Loan Default

Scenario: A bank wants to predict whether a loan applicant will default based on credit score, income, and past loan history.
Problem Type: Classification
Step-by-Step Logic:

- **Collect Data** – Gather customer financial history, credit scores, income details, and loan repayment records.
- **Preprocess Data** – Handle missing values using imputation techniques; normalize numerical features like income and credit score; encode categorical variables such as loan type and employment status using one-hot or label encoding.
- **Split Dataset** – Divide the dataset into training and testing sets (e.g., 80/20 split).
- **Choose Algorithm** – Use Logistic Regression, Decision Trees, or Random Forest.
- **Train Model** – Fit the model using labeled data indicating loan default status.
- **Evaluate Performance** – Use AUC-ROC, Precision, Recall, and F1-score.
- **Make Predictions** – Predict loan default for new applicants.

---

2. Forecasting Demand for a Retail Store

Scenario: A retail store wants to predict the demand for different products to optimize inventory levels.
Problem Type: Regression
Step-by-Step Logic:

- **Collect Data** – Gather historical sales data, seasonal trends, promotions, and product attributes.
- **Preprocess Data** – Fill missing values using interpolation or mean imputation; normalize numerical features like sales volume and price; detect and remove outliers using z-score or IQR methods.
- **Split Dataset** – Divide the data into training and testing sets.
- **Choose Algorithm** – Use Linear Regression, Random Forest Regression, or XGBoost.

- **Train Model** – Fit the model using historical demand data.
- **Evaluate Performance** – Use RMSE and R^2 score.
- **Make Predictions** – Forecast demand for upcoming sales periods.

---

3. Detecting Defective Products in Manufacturing

Scenario: A factory wants to detect whether a manufactured product is defective based on sensor readings and quality control data.
Problem Type: Classification
Step-by-Step Logic:

- **Collect Data** – Gather sensor readings, production parameters, and defect labels.
- **Preprocess Data** – Impute missing sensor values; scale numerical features using MinMax or StandardScaler; encode categorical variables like machine ID or shift using one-hot encoding.
- **Split Dataset** – Divide the data into training and testing sets.
- **Choose Algorithm** – Use Decision Trees, Support Vector Machines, or Neural Networks.
- **Train Model** – Fit the model using labeled defect data.
- **Evaluate Performance** – Use accuracy, precision, recall, and F1-score.
- **Deploy Model** – Detect defective products in real time.

---

4. Classifying Medical Diagnoses

Scenario: A healthcare provider wants to classify patient symptoms into different disease categories.
Problem Type: Classification
Step-by-Step Logic:

- **Collect Data** – Gather patient records including symptoms, test results, and confirmed diagnoses.
- **Preprocess Data** – Handle missing values using domain-specific rules; normalize lab test results; encode categorical features like gender, symptom type, and medical history.
- **Split Dataset** – Perform a train-test split.
- **Choose Algorithm** – Use Random Forest, Naive Bayes, or Gradient Boosting.

- **Train Model** – Fit the model using labeled medical data.
- **Evaluate Model** – Use accuracy, confusion matrix, and F1-score.
- **Make Predictions** – Predict disease category based on patient symptoms.

---

5. Identifying Fake Online Reviews

Scenario: An e-commerce company wants to detect fake reviews posted by bots or fraudsters.
Problem Type: Classification
Step-by-Step Logic:

- **Collect Data** – Gather a dataset of verified real and fake reviews.
- **Preprocess Data** – Tokenize text, remove stopwords, apply stemming or lemmatization, and vectorize using TF-IDF or word embeddings.
- **Feature Engineering** – Extract features like review length, sentiment score, frequency of posting, and linguistic patterns.
- **Split Dataset** – Divide data into training and testing sets.
- **Choose Algorithm** – Use Naive Bayes, Logistic Regression, or Transformer-based models.
- **Train Model** – Fit the model on labeled review data.
- **Evaluate Performance** – Use accuracy, F1-score, and confusion matrix.
- **Make Predictions** – Detect fake reviews in real-time.

---

6. Predicting Stock Market Trends

Scenario: A financial firm wants to predict stock price movement based on historical price data and market indicators.
Problem Type: Regression
Step-by-Step Logic:

- **Collect Data** – Gather historical stock prices, trading volumes, technical indicators, and macroeconomic data.
- **Preprocess Data** – Handle missing values using forward fill or interpolation; normalize price changes; engineer features like moving averages, RSI, and MACD.
- **Split Dataset** – Perform a train-test split.

- **Choose Algorithm** – Use Random Forest Regression, LSTMs, or Gradient Boosting.
- **Train Model** – Fit the model on historical stock data.
- **Evaluate Performance** – Use RMSE and directional accuracy.
- **Make Predictions** – Forecast future stock price movements.

---

7. Detecting Fake Social Media Accounts

Scenario: A social media platform wants to identify and remove fake user accounts.
Problem Type: Classification
Step-by-Step Logic:

- **Collect Data** – Gather account metadata, activity logs, and engagement metrics.
- **Preprocess Data** – Handle missing values; engineer features like post frequency, follower-following ratio, and account age; scale numerical features.
- **Split Dataset** – Divide into training and testing sets.
- **Choose Algorithm** – Use Random Forest, Support Vector Machines, or XGBoost.
- **Train Model** – Fit the model using labeled real and fake account data.
- **Evaluate Performance** – Use Precision, Recall, and F1-score.
- **Make Predictions** – Identify and flag fake accounts.

---

8. Optimizing Ad Targeting for Online Marketing

Scenario: A digital marketing company wants to show the most relevant ads to users based on their browsing behavior.
Problem Type: Clustering
Step-by-Step Logic:

- **Collect Data** – Gather user clickstream data, browsing history, and demographic information.
- **Preprocess Data** – Encode categorical variables like device type and location; scale numerical features; handle missing data using imputation.
- **Choose Algorithm** – Use K-Means or Hierarchical Clustering.
- **Determine Optimal Clusters** – Use the Elbow Method or Silhouette Score.
- **Train Model** – Apply clustering algorithm to segment users.

- **Analyze Clusters** – Identify user groups (e.g., "Tech Enthusiasts," "Fashion Lovers").
- **Optimize Ads** – Deliver targeted ads based on cluster preferences.

---

9. Classifying Land Cover in Satellite Images

Scenario: A geospatial research team wants to classify different land types (forest, water, urban) using satellite images.
Problem Type: Classification
Step-by-Step Logic:

- **Collect Data** – Use satellite images labeled with land cover types.
- **Preprocess Data** – Normalize pixel values; apply image denoising techniques; extract features using PCA or CNN-based feature maps.
- **Split Dataset** – Divide into training and testing sets.
- **Choose Algorithm** – Use Decision Trees, Support Vector Machines, or CNN-based models.
- **Train Model** – Fit the model on labeled satellite images.
- **Evaluate Performance** – Use accuracy and confusion matrix.
- **Make Predictions** – Classify new satellite images into land cover types.

---

10. Predicting Customer Churn for a Subscription Service

Scenario: A streaming service wants to predict which users are likely to cancel their subscriptions.
Problem Type: Classification
Step-by-Step Logic:

- **Collect Data** – Gather user engagement metrics, subscription history, and interaction logs.
- **Preprocess Data** – Handle missing values; encode categorical variables like subscription type and device; scale numerical features such as watch time and login frequency.
- **Feature Engineering** – Create features like average session duration, days since last login, and content diversity.
- **Split Dataset** – Perform a train-test split.
- **Choose Algorithm** – Use Logistic Regression, Random Forest, or Gradient Boosting.

- **Train Model** – Fit the model using past churn data.
- **Evaluate Performance** – Use AUC-ROC, Precision, and Recall.
- **Make Predictions** – Identify customers likely to churn and apply retention strategies.
