**Phase 1 : Bagging and Boosting**

**Concepts: Ensemble learning, Bagging (Random Forest), Boosting (Adaboost, XGBoost)**

**DATASET:** [**Breast Cancer Dataset**](https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset) **from Kaggle**

**Mini Task:**

* **Train:**
  + **RandomForestClassifier**
  + **AdaBoostClassifier**
  + **XGBoostClassifier**
* **Compare accuracy**

**Deliverable: Jupyter Notebook comparing all three with accuracy, basic error analysis.**

**Phase 2: Support Vector Machines (SVM)**

**Concepts: Margin, kernels, linear/non-linear classification, Dimensionality Reduction using PCA**

**DATASET:** [**Credit card dataset: SVM Classification**](https://www.kaggle.com/code/pierra/credit-card-dataset-svm-classification) **from Kaggle**

**Mini Task:**

* **Train SVM with:**
  + **Linear kernel**
  + **RBF kernel**
  + **Polynomial kernel**
* **Visualize (just a 2D PCA plot to keep it minimal)**

**Deliverable: Notebook with accuracy and explanation of kernel effect**

**Phase 3: Unsupervised Learning**

**DATASET :** [**Iris Species**](https://www.kaggle.com/datasets/uciml/iris) **from kaggle**

**Concepts:**

* **Clustering: k-Means & Hierarchical**
* **Dimensionality reduction recap (PCA/t-SNE)  
  Mini Task:**
* **Apply:**
  + **k-Means (elbow method)**
  + **Hierarchical Clustering (dendrogram using scipy)**
* **Bonus: Use PCA or t-SNE before clustering**

**Deliverable: Notebook showing clusters and a few real-world applications.**

**Phase 4: SVD + PCA**

**Concepts: TF-IDF with SVD**

**DATASET:** [**20 Newsgroups**](https://www.kaggle.com/datasets/crawford/20-newsgroups) **from kaggle**

**Mini Task:**

* **Apply TF-IDF → SVD → Visualize top 2 components**
* **Optional: After performing TF-IDF on the given dataset, try to apply clustering with 20 clusters to see if the data forms a group roughly identical to the original groups.**

**Deliverables:**

1. **Code Output:**
   * **Display shape of:**
     + **Raw TF-IDF matrix**
     + **After applying SVD or PCA**
   * **Include a 2D scatter plot of the first two components**
   * **For clustering: Show silhouette score and comparison plot of predicted vs actual clusters (optional but insightful)**
2. **Explanation:**
   * **A brief explanation (3–4 lines):**
     + **Why dimensionality reduction is used (mention sparsity of TF-IDF)**
     + **What SVD/PCA achieves**
     + **How clustering helps understand the structure of data without labels**

**Phase 5 : Model Validation & Selection**

**Concepts:**

* **Cross-validation**
* **Bias-variance trade-off**
* **Learning curves**

**Mini Task:**

* **Choose a model (SVM or Random Forest)**
* **Apply:**
  + **KFold Cross-validation**
  + **Plot Learning Curve**
* **Discuss overfitting vs underfitting**

**Main Challenge: Tweet Sentiment Analysis**

**Goal:**

**Classify tweets into Positive, Neutral, or Negative categories using Machine Learning techniques learned during Cycle-2.**

**Workflow & Task Breakdown**

**Task 1: Data Cleaning**

**Objective: Prepare raw tweet text for analysis  
Steps:**

* **Load the dataset (CSV format; contains tweets and sentiment labels).**
* **Remove unnecessary parts of each tweet:**
  + **URLs (e.g., https://...)**
  + **Mentions (e.g., @username)**
  + **Hashtags (keep the word but remove the #)**
  + **Special characters and emojis (basic cleaning)**
  + **Convert all text to lowercase for consistency.**

**Task 2: Label Mapping**

**Objective: Make sentiment labels more understandable  
Steps:**

* **Convert numeric sentiment labels into:**
  + **Negative (originally 0)**
  + **Neutral (originally 2)**
  + **Positive (originally 4)**
* **Optionally, map them to string labels or cleaner numbers like -1, 0, 1.**

**Task 3: Feature Extraction**

**Objective: Convert tweet text into numerical format for ML models  
Steps:**

* **Use a TF-IDF vectorizer to capture the importance of words in the tweets.**
* **Limit vocabulary size (e.g., top 5,000 words) to reduce noise and overfitting.**
* **Each tweet will now become a numerical vector (based on word importance).**

**Task 4: Model Building**

**Objective: Train a classifier to learn sentiment from features  
Steps:**

* **Choose one algorithm: Logistic Regression / SVM / Naive Bayes / XGBoost.**
* **Split your dataset into training and testing sets (e.g., 80% training, 20% testing).**
* **Train the model using the features from the previous step and sentiment labels.**

**Task 5: Model Evaluation**

**Objective: Check how well your model performs  
Steps:**

* **Use the model to predict sentiment on the test set.**
* **Generate a Confusion Matrix to visualize where the model is going wrong.**
* **Review a Classification Report to understand:**
  + **Precision (how accurate positive predictions are)**
  + **Recall (how many actual positives are captured)**
  + **F1-score (balance between precision and recall)**

**Deliverables**

* **Accuracy score of your model**
* **Confusion matrix visualization (can be a table or plot)**
* **A short write-up:**
  + **Which model you chose and why**
  + **How TF-IDF helped capture important features**
  + **Challenges faced and how you solved them**