MIT-ADT UNIVERSITY
PUNE, INDIA
A leap towards World Class Education
(Established by MIT Art, Design and Technology University Act, 2015
(Maharashtra Act No. XXXIX of 2015)

MIT Art Design and Technology University
MIT School of Computing, Pune

Department of Computer Science and Engineering

Lab Manual

Subject – Machine Learning Essentials Lab

Class - T.Y. (SEM-II), AIEC

Prof. Vijaya Patil

Prepared By: - Prof. Vijaya Patil
Year: - Third Year-AIEC
Required H/W and S/W: - 64 bit processor based machine and Visual Studio Code, Python

A.Y. 2025 – 2026 (SEM-II)

Lab Manual Index		
Sl. No.	Title	Page No.
1	Data Acquisition and Exploration • Scope: Use the "Titanic: Machine Learning from Disaster" dataset. Import CSV, clean missing ages, encode categorical columns, summarize with Pandas, and visualize survival rate by gender/class. • Datasets: Titanic (Kaggle), UCI Adult dataset • Types: Data preprocessing, EDA	4-6
2	Data Visualization Techniques • Scope: Use the Iris dataset. Create scatter plots, histograms, heatmaps, violin plots, and bar charts to visualize class distribution, feature correlation, and species separation. • Datasets: Iris, Penguins • Types: Visualization, Matplotlib, Seaborn	7-9
3	End-to-End ML Pipeline • Scope: "California Housing" dataset. Complete pipeline: Impute missing values, feature scaling, train/test split, train linear regression, evaluate, and save predictions. • Datasets: California Housing (scikit-learn) • Types: Regression, Pipeline	10-12
4	Regression: Linear & Polynomial • Scope: Use the "Auto MPG" dataset to predict fuel efficiency (mpg) with linear and polynomial regression. Compare underfitting, overfitting, plot learning/validation curves. • Datasets: Auto MPG, Boston Housing Types: Regression	13-15
5	Classification: Binary & Multiclass • Scope: Apply logistic regression and SVM to the MNIST dataset for digit classification. Train, predict, evaluate with confusion matrix and F1-score. • Datasets: MNIST (digits), Breast Cancer Wisconsin Types: Classification	16-18
6	Model Performance Evaluation • Scope: For Adult Census dataset, train decision trees for income prediction. Evaluate accuracy, precision, recall, ROC, and AUC. Use k-fold cross-validation for robust estimates. • Datasets: Adult Census, Titanic Types: Evaluation	19-21
7	Regularized Regression Models • Scope: Apply Ridge, Lasso, ElasticNet to the Diabetes dataset. Compare model coefficients and interpret regularization effects on feature selection. • Datasets: Diabetes Types: Regression, Regularization	22-23

8	Support Vector Machines • Scope: Use the "Iris" and "Wine" datasets. Classify using linear and RBF kernels; visualize support vectors/margins; compare SVM, logistic regression, and k-NN on same task. • Datasets: Iris, Wine Types: Classification	24-26
9	Decision Trees & Random Forests • Scope: Train classification trees and random forests with the Titanic dataset. Visualize tree graph, analyze depth/pruning, compare random forest feature importances. • Datasets: Titanic, Heart Disease Types: Classification	27-29
10	Gradient Descent Optimization • Scope: Train a regression model on California Housing using batch, stochastic, and mini-batch GD. Plot cost function convergence, discuss trade-offs (speed, variance). • Datasets: California Housing Types: Optimization	30-32
11	Dimensionality Reduction: PCA • Scope: Apply PCA to the Fashion-MNIST images. Visualize variance explained, reconstruct images, plot with principal components, and use PCs as input to classifiers. • Datasets: Fashion-MNIST, MNIST Types: Dimensionality Reduction	33-35
12	Manifold Learning Techniques • Scope: Use t-SNE and LLE on facial recognition dataset (e.g., Olivetti faces), or on MNIST. Project data to 2D/3D and visualize cluster separation. • Datasets: Olivetti Faces, MNIST Types: Unsupervised, Visualization	36-38
13	Clustering Algorithms • Scope: Apply K-Means and DBSCAN to Mall Customer Segmentation data. Visualize clusters, assess purity when true labels are available, and analyze cluster profiles. • Datasets: Mall Customer Segmentation, Iris • Types: Clustering	39-41
14	Anomaly Detection • Scope: Use Credit Card Fraud dataset. Train Gaussian Mixture Model and Isolation Forest, identify outliers, compare detection rates/false positives. • Datasets: Credit Card Fraud Detection (Kaggle), KDD Cup 99 Types: Anomaly Detection	42-44
15	Viva Questions	45-46

Prepared By – Prof. Vijaya Patil **Head - Dept. of CSE,**

ASSIGNMENT 1: Data Acquisition and Exploration

Problem Statement

Understand and apply fundamental data acquisition and exploratory data analysis (EDA) techniques using Pandas, NumPy, and visualization tools such as Matplotlib and Seaborn.

Scenario

A data analyst has received the **Titanic passenger dataset**.
The goal is to clean the dataset, understand the passenger demographics, and analyze survival trends.
Insights from this analysis may help researchers understand which factors influenced survival during the disaster.

Aim

To load, clean, transform, and visualize the Titanic dataset using Pandas, Matplotlib, and Seaborn while performing meaningful exploratory data analysis.

Objectives

1. Load CSV data from local storage or URL.
2. Inspect dataset structure, missing values, and data types.
3. Clean missing values (especially Age and Embarked).
4. Encode categorical variables where needed.
5. Generate descriptive statistics.
6. Visualize survival distribution across different groups.

Tasks

Question 1: Load and Inspect Dataset

1. Load the file **titanic_data.csv** using Pandas.
2. Display:
    - Number of rows and columns (.shape)
    - Column names (.columns)
    - Data types (.dtypes)
    - First 10 rows (.head())

Question 2: Identify Missing Values

1. Display total missing values per column.
2. Check percentage of missing values for each column.

Question 3: Clean the Dataset

1. Fill missing **Age** values using the **median** age.
2. Fill missing **Embarked** values with the most frequent value.
3. Drop any rows that still contain large amounts of missing data (only if necessary).

Question 4: Encode Categorical Features

Convert the following columns into numeric form using Label Encoding / Mapping:

- Sex → (male = 0, female = 1)
- Embarked → numeric categories

Question 5: Compute Summary Statistics

1. Display:
    - Mean, Median, Standard deviation
    - Minimum and Maximum values
2. Compute survival counts:
    - How many survived?
    - How many did not survive?

Question 6: Grouped Survival Analysis

Analyze survival patterns:

1. Survival by gender
2. Survival by passenger class (Pclass)
3. Survival by embarkation port

Display results using grouped tables.

Question 7: Visualization – Survival Count

Create a **count plot** showing total survivors vs non-survivors.

Question 8: Visualization – Survival by Gender

Create a bar chart showing:

- Male survivors vs non-survivors
- Female survivors vs non-survivors

Question 9: Visualization – Survival vs Passenger Class

Create a bar plot showing survival rates across different classes:

- 1st class
- 2nd class
- 3rd class

Question 10: Histogram of Age

Plot:

- Age distribution of passengers
- Separate distributions for survived vs not survived (optional bonus)

Conclusion

By completing this assignment, students will:

Learn how to load and inspect real-world datasets
Handle missing values appropriately
Convert categorical features into numeric form
Perform summary statistical analysis
Visualize survival trends using Pandas, Matplotlib, and Seaborn
Gain foundational skills required for machine learning preprocessing

---

Student Evaluation Rubric (Total: 10 Marks)

Criteria	Marks
Loading and inspecting dataset	1
Handling missing values correctly	2
Encoding categorical features	2
Computing descriptive statistics	2
Meaningful visualizations	2
Code clarity and formatting	1

ASSIGNMENT 2: **Data Visualization Techniques**

Problem Statement

Understanding raw data only by looking at numbers is difficult. Data visualization helps convert numeric data into meaningful visual patterns. Using visualization tools such as **Matplotlib** and **Seaborn**, students will learn how to analyze and communicate insights effectively.

Scenario
A data analyst has been given two biological datasets — **Iris** and **Penguins**.
The analyst must study species characteristics, identify visual patterns, detect correlations, and present observations using meaningful graphs.
Proper visualization will help biologists classify species, understand measurement variations, and detect trends.

Aim
To visualize and analyze biological datasets using Matplotlib and Seaborn, and demonstrate the role of visual analytics in data-driven decision making.

Objectives
Students will be able to:

1. Import datasets and understand their structure.
2. Visualize distributions, relationships, and class separation.
3. Create and interpret different plot types.
4. Compare patterns between different features.
5. Draw meaningful conclusions from graphs.

Datasets
Iris Dataset
Penguins Dataset
Format: CSV (from Kaggle / Seaborn library)

Software / Tools

- Python
- Jupyter Notebook / PyCharm
- Matplotlib
- Seaborn
- Pandas

TASKS (Answer All — 10 Questions)
Question 1 — Load and Inspect
Load the dataset and print:

- first 5 rows
- column names
- number of rows and columns

Question 2 — Class Distribution
Create a **bar chart** showing the count of each species.

Question 3 — Feature Distribution

Create **histograms** for at least two numerical features.

Question 4 — Scatter Plot
Plot a scatter graph showing:
sepal_length vs sepal_width
Color points by species.

Question 5 — Pair Plot
Create a **pairplot** to visualize all feature relationships.

Question 6 — Heatmap
Create a **correlation heatmap** and identify which features are strongly related.

Question 7 — Box Plot
Draw boxplots for numerical features to observe spread and outliers.

Question 8 — Violin Plot
Create violin plots comparing feature distribution across species.

Question 9 — Compare Datasets
Visualize and compare one feature from **Iris vs Penguins** datasets (for example: body mass vs flipper length).
Explain differences.

Question 10 — Summary Interpretation
Write short answers for:

- What did you observe?
- Which plot explained data best?
- How visualization helped?

Conclusion
After completing this experiment, students will:
Understand importance of data visualization
Learn to select appropriate plot types
Interpret visual insights instead of raw tables
Identify correlations, distributions, and species separation
Gain confidence in using Matplotlib and Seaborn
Visualization transforms data into meaningful stories.

Student Evaluation Rubrics (Total: 10 Marks)

Criteria	Marks
Loading and inspecting dataset	1
Correct graphs created	3
Proper labeling & formatting	2
Interpretation of results	3
Code readability / comments	1

ASSIGNMENT 3: End-to-End ML Pipeline (California Housing)

Problem Statement
Students often learn individual ML steps separately but struggle to integrate them into a complete workflow. This experiment focuses on building a full regression pipeline — from raw data to predictions.

Scenario
A data scientist working for a real-estate company needs to predict **house prices** using demographic and geographical factors from the **California Housing dataset**.
The model must:
handle missing values
scale features
split into train/test sets
train a regression model
evaluate performance
generate predictions
The goal is to automate the pipeline so it can be reused in future projects.

Aim
To develop a complete machine learning pipeline for regression using the California Housing dataset.

Objectives
Students will be able to:

1. Load the dataset and understand feature meanings.
2. Handle missing values using imputation.
3. Perform feature scaling.
4. Split dataset into train and test sets.
5. Train a Linear Regression model.
6. Evaluate model performance.
7. Save model predictions for future analysis.

Dataset
California Housing Dataset (from scikit-learn)
Loaded using:
from sklearn.datasets import fetch_california_housing

Tools Required
Python
Pandas
NumPy
Matplotlib / Seaborn
Scikit-learn

TASKS (Answer All — 10 Questions)
Question 1 — Load Dataset
Load the California Housing dataset and print:
• Feature names
• Target name
• Number of samples and features

Question 2 — Convert to DataFrame
Convert data into a Pandas DataFrame and display first 10 rows.

Question 3 — Check Missing Values
Check for missing values and report findings.

Question 4 — Handle Missing Values
Impute missing values using:
SimpleImputer(strategy="median")

Question 5 — Train-Test Split
Split dataset into:
80% training
20% testing

Question 6 — Feature Scaling
Apply StandardScaler and explain why scaling is needed.

Question 7 — Train Model
Train **Linear Regression** on the processed dataset.

Question 8 — Model Evaluation
Evaluate model using:
• Mean Squared Error (MSE)
• Root Mean Squared Error (RMSE)
• R^2 Score
Interpret the results in 3–4 lines.

Question 9 — Save Predictions
Generate predictions on the test set and save them into:
predictions.csv

Question 10 — Reflection
Answer briefly:
• Which pipeline step was most useful?
• Where could errors occur?
• How could accuracy be improved?

Conclusion
After completing this assignment, students will be able to:
Build a full ML pipeline
Understand importance of preprocessing
Train and evaluate regression models
Save and interpret predictions
This assignment prepares students for real-world ML workflows.

Student Evaluation Rubric (Total: 10 Marks)

Criteria	Marks
Dataset loading & preprocessing	2
Missing value handling	2
Pipeline & scaling	2
Model training & evaluation	3
Code readability & organization	1

ASSIGNMENT 4: Linear & Polynomial Regression

Problem Statement

Understanding how different regression models behave is important when solving prediction problems. Simple linear regression may underfit real-world data, while higher-degree polynomial models may overfit.
This experiment focuses on predicting **fuel efficiency (MPG)** and comparing model performance.

Scenario
An automobile research company wants to predict **miles-per-gallon (MPG)** based on engine characteristics such as horsepower, weight, displacement, and cylinders.
The analyst must:
build linear regression and polynomial regression models
study underfitting and overfitting
observe learning and validation curves
explain performance differences

Aim
To develop and compare **linear and polynomial regression models** for predicting automobile fuel efficiency using the Auto MPG dataset.

Objectives
Students will be able to:

1. Load and preprocess the Auto MPG dataset.
2. Handle missing values and encode categorical features.
3. Train a **Linear Regression model**.
4. Train **Polynomial Regression models** (degree 2, 3…).
5. Observe underfitting vs overfitting.
6. Plot learning/validation curves.
7. Compare results using evaluation metrics.

Datasets
Primary:
Auto MPG Dataset (UCI Repository / Seaborn)
Optional for practice:
Boston Housing (scikit-learn)

Tools Required
Pandas
NumPy
Matplotlib / Seaborn
Scikit-learn

TASKS (Answer All — 10 Questions)
Question 1 — Load Dataset
Load Auto MPG dataset and display:
• first 10 rows
• dataset shape
• column names

Question 2 — Data Cleaning

Handle missing values in mpg and horsepower.
Mention the method used (drop / median / mean).

Question 3 — Feature Selection
Select independent variables such as:
• cylinders
• horsepower
• weight
• displacement
Explain why these features matter.

Question 4 — Train-Test Split
Split data into:
80% training
20% testing

Question 5 — Build Linear Regression Model
Train model and print:
• coefficients
• intercept

Question 6 — Evaluate Linear Regression
Compute:
• MAE
• MSE
• RMSE
• R^2 score
Interpret results in 3–4 lines.

Question 7 — Polynomial Regression
Train polynomial models (degree 2 and 3).
Compare training accuracy with test accuracy.
Explain signs of underfitting/overfitting.

Question 8 — Learning Curve
Plot learning curve showing training vs validation error as dataset size increases.
Describe behavior.

Question 9 — Validation Curve
Plot validation curve (degree vs R^2).
Identify degree with best performance.

Question 10 — Reflection
Answer briefly:
• Which model performed better and why?
• What degree caused overfitting?
• How can model generalization be improved?

Conclusion
After completing this assignment students will:

understand linear vs polynomial regression
analyze underfitting and overfitting
evaluate and compare models
interpret results using metrics and curves
This prepares students for more advanced predictive modeling tasks.

Student Evaluation Rubric (Total 10 Marks)

Criteria	Marks
Dataset loading & preprocessing	2
Model training (linear + polynomial)	3
Learning/validation curve analysis	3
Code organization & explanation	2

ASSIGNMENT 5: Classification — Binary & Multiclass

Problem Statement

Classification helps machines decide “which group an object belongs to.”
In real applications, models classify:

emails (spam / not spam)
medical reports (disease / no disease)
handwritten digits (0–9)
This assignment focuses on building and evaluating **binary and multiclass classification models**.

Scenario
A data scientist works on two problems:
Digit Recognition (MNIST) — multiclass problem (0–9 digits)
Breast Cancer Diagnosis — binary classification (malignant / benign)
The task is to:
build Logistic Regression & SVM models
train & evaluate performance
interpret confusion matrix & F1-score
compare results

Aim
To implement binary and multiclass classification using Logistic Regression and Support Vector Machine (SVM).

Objectives
Students will be able to:

1. Load and preprocess classification datasets.
2. Differentiate binary vs multiclass classification.
3. Train Logistic Regression models.
4. Train Support Vector Machine (SVM) models.
5. Use confusion matrix and F1-score for evaluation.
6. Compare performance of both models.

Datasets
Primary:
MNIST Handwritten Digits (scikit-learn)
Optional:
Breast Cancer Wisconsin Dataset (binary classification)

Tools Required
Python
Pandas / NumPy
Scikit-learn
Matplotlib / Seaborn

TASKS (Answer All — 10 Questions)
Question 1 — Load Datasets
Load:
• MNIST digits dataset
• Breast Cancer dataset
Display:
• shapes
• feature names
• sample outputs

Question 2 — Data Splitting
Split each dataset into:
70% training
30% testing
Explain why splitting is required.

Question 3 — Logistic Regression (Binary Case)
Train Logistic Regression on the **Breast Cancer dataset**.
Print:
• accuracy
• confusion matrix

Question 4 — F1-Score (Binary Case)
Calculate and interpret:
precision
recall
F1-score

Question 5 — Logistic Regression (Multiclass)
Train Logistic Regression on **MNIST**.
Observe performance and discuss challenges.

Question 6 — SVM Model (Binary)
Train SVM classifier on Breast Cancer dataset.
Compare with Logistic Regression.

Question 7 — SVM (Multiclass MNIST)
Train SVM classifier for MNIST.
Explain:
• accuracy difference
• time taken
• advantages / disadvantages

Question 8 — Confusion Matrix Visualization
Plot confusion matrix for at least one model.
Explain what misclassifications mean.

Question 9 — Model Comparison
Compare models based on:
• accuracy
• F1-score
• confusion matrix
Discuss which model performs better and why.

Question 10 — Reflection
Write short answers:
• When should we use Logistic Regression?
• When is SVM better?
• Why is F1-score more useful than accuracy sometimes?

Conclusion

After completing this assignment students will:
understand binary vs multiclass classification
build Logistic Regression and SVM models
evaluate using confusion matrix & F1-score
compare and analyze real model behavior
This gives strong foundations for AI and ML applications.

Student Evaluation Rubric (Total: 10 Marks)

Criteria	Marks
Dataset loading & preprocessing	2
Correct implementation of models	3
Confusion matrix + F1-score interpretation	3
Code clarity & explanation	2

ASSIGNMENT 6: Model Performance Evaluation

Problem Statement
Building a machine learning model is not enough — we must ensure it performs well and generalizes to unseen data.
Many students rely only on **accuracy**, which can be misleading. Proper evaluation requires metrics such as **precision, recall, ROC, and AUC**, along with **cross-validation**.
This assignment focuses on evaluating classification models rigorously.

Scenario
A data analyst is asked to predict whether a person's income is:
<= 50K or > 50K
based on demographic and employment details from the **Adult Census dataset**.
The analyst must:
build a Decision Tree classifier
evaluate the model with multiple performance metrics
validate using k-fold cross-validation
compare evaluation with another dataset (Titanic – optional)

Aim
To evaluate a classification model using multiple performance metrics and cross-validation techniques.

Objectives
Students will be able to:

1. Load and preprocess the Adult Census dataset.
2. Train a Decision Tree classifier.
3. Calculate accuracy, precision, recall, and F1-score.
4. Plot ROC curve and compute AUC.
5. Apply k-fold cross-validation.
6. Interpret results and identify overfitting/underfitting.

Datasets
Primary:
Adult Census Income Dataset (UCI / Kaggle)
Optional practice:
Titanic Dataset

Tools Required
Python
Pandas / NumPy
Scikit-learn
Matplotlib / Seaborn

TASKS (Answer All — 10 Questions)
Question 1 — Load Dataset
Load Adult Census dataset and display:
• first 10 rows
• dataset shape
• list of features

Question 2 — Data Cleaning
Handle missing values and encode categorical columns.
Explain technique used (label encoding / one-hot encoding).

Question 3 — Define Target Variable
Define income column as:
0 = <= 50K
1 = > 50K
Explain why binary encoding is required.

Question 4 — Train-Test Split
Split dataset into:
70% training
30% testing

Question 5 — Train Decision Tree
Train Decision Tree classifier with default settings.
Print:
• depth
• number of leaves

Question 6 — Accuracy
Compute model accuracy.
Explain in 2–3 lines why accuracy alone is not enough.

Question 7 — Precision & Recall
Compute:
• precision
• recall
• F1-score
Explain in 3–4 lines which metric is more important for income prediction and why.

Question 8 — ROC & AUC
Plot ROC curve and calculate AUC.
Answer:
• What does AUC represent?
• Is the model good or weak?

Question 9 — K-Fold Cross-Validation
Apply 5-fold or 10-fold cross-validation.
Report:
• mean accuracy
• standard deviation
Explain why cross-validation improves reliability.

Question 10 — Reflection
Write short answers:

• Did the decision tree overfit?
• Which metric gave the best understanding?
• How could the model be improved?

Conclusion

After completing this assignment, students will:
understand importance of multiple evaluation metrics
analyze precision, recall, F1, ROC, and AUC
apply cross-validation for reliable results
avoid relying only on accuracy
This develops strong evaluation and analysis skills.

Student Evaluation Rubric (Total: 10 Marks)

Criteria	Marks
Data loading & preprocessing	2
Decision Tree training	2
Evaluation metrics (accuracy/precision/recall/F1)	3
ROC–AUC + cross-validation interpretation	2
Code clarity & explanation	1

ASSIGNMENT 7: Regularized Regression Models

Problem Statement

Traditional Linear Regression may overfit when:
dataset is small
features are correlated
noise is high
Regularization techniques such as **Ridge, Lasso, and ElasticNet** help control overfitting and improve model generalization.

Scenario

A healthcare researcher wants to predict **disease progression** in diabetic patients using the **Diabetes dataset** containing medical measurements such as BMI, blood pressure, age, and glucose levels.
The goal is to:
train multiple regularized regression models
compare coefficients
observe effect of regularization
decide which model works best

Aim

To apply Ridge, Lasso, and ElasticNet regression and interpret how regularization impacts coefficients and feature selection.

Objectives

Students will be able to:

1. Load and understand the Diabetes dataset.
2. Train baseline Linear Regression model.
3. Apply Ridge, Lasso, and ElasticNet models.
4. Compare performance using regression metrics.
5. Analyze model coefficients.
6. Interpret the effect of regularization on feature selection.

Dataset

Diabetes Dataset (scikit-learn)

Tools Required

Python
Pandas / NumPy
Matplotlib / Seaborn
Scikit-learn

TASKS (Answer All — 10 Questions)

Question 1 — Load Dataset

Load Diabetes dataset and display:
• features
• target name
• first 10 records

Question 2 — Train-Test Split

Split into:

80% training
20% testing

Question 3 — Baseline Linear Regression
Train Linear Regression and calculate:
• MAE
• MSE
• RMSE
• R^2
Write short interpretation.

Question 4 — Ridge Regression
Train Ridge Regression with different alpha values:
alpha = 0.1, 1, 10
Observe changes in performance.

Question 5 — Lasso Regression
Train Lasso model.
Identify which coefficients become **zero** and explain what that means.

Question 6 — ElasticNet
Train ElasticNet model.
Compare results with Ridge and Lasso.

Question 7 — Coefficient Comparison Table
Create a table comparing coefficients from:
• Linear Regression
• Ridge
• Lasso
• ElasticNet
Explain trends.

Question 8 — Effect of Regularization
Answer in 3–4 lines:
• Which model overfits least?
• Which model removes unnecessary features?

Question 9 — Visualization
Plot coefficient values for each model.
Discuss differences.

Question 10 — Reflection
Short reflection:
• When should we use Lasso?
• When should we use Ridge?
• Why use ElasticNet instead of only one?

Conclusion
After completing this assignment, students will:

understand why regularization is needed
compare Ridge, Lasso, and ElasticNet
analyze coefficient shrinkage
identify feature selection impact
This assignment builds strong regression analysis skills.

Student Evaluation Rubric (Total: 10 Marks)

Criteria	Marks
Dataset loading & preprocessing	2
Model implementation (3 models)	3
Coefficient interpretation & comparison	3
Code clarity & explanation	2

ASSIGNMENT 8: Support Vector Machines (SVM)

Problem Statement

Support Vector Machines are powerful classifiers that separate classes by finding the "best possible boundary" (maximum margin). Choosing the correct kernel (Linear / RBF) significantly affects performance.

Students often struggle to understand:

what support vectors are
how margins work
when to use Linear vs RBF kernels

This assignment explains SVM through visualization and comparison.

Scenario

A data analyst is classifying:

Iris species

Wine quality classes

The analyst must:

train SVM with linear & RBF kernels
visualize margin + support vectors
compare performance with Logistic Regression and k-NN
decide best classifier and justify

Aim

To implement Support Vector Machine classifiers, visualize decision boundaries, and compare with other classification models.

Objectives

Students will be able to:

1. Load and preprocess Iris and Wine datasets.
2. Train SVM with Linear and RBF kernels.
3. Visualize decision boundaries and support vectors.
4. Evaluate performance using accuracy, confusion matrix, and F1-score.
5. Compare SVM with Logistic Regression and k-NN.
6. Interpret strengths and limitations of SVM.

Datasets

Iris Dataset (scikit-learn)
Wine Dataset (scikit-learn)

Tools Required

Python
Pandas / NumPy
Matplotlib / Seaborn
Scikit-learn

TASKS (Answer All — 10 Questions)

Question 1 — Load Datasets

Load Iris and Wine datasets and print:

• number of records
• number of classes
• feature names

Question 2 — Train-Test Split
Split each dataset into:
80% training
20% testing
Explain why splitting is important.

Question 3 — SVM (Linear Kernel)
Train SVM with:
kernel = "linear"
Print:
• accuracy
• number of support vectors
Explain meaning of support vectors.

Question 4 — SVM (RBF Kernel)
Train SVM with:
kernel = "rbf"
Compare performance with linear kernel.

Question 5 — Visualize Decision Boundary (Iris)
Plot decision regions for two selected features.
Identify:
• class overlap
• margin width

Question 6 — Confusion Matrix & F1-score
For at least one dataset:
• confusion matrix
• precision
• recall
• F1-score
Explain misclassifications.

Question 7 — Logistic Regression Comparison
Train Logistic Regression on same data and compare accuracy.
Explain when Logistic works better.

Question 8 — k-NN Comparison
Train k-NN classifier and compare results.
Discuss sensitivity to neighbors (k).

Question 9 — Model Comparison Table
Create a table:

Model	Dataset	Accuracy	F1-Score
Logistic Regression			
SVM (Linear)			
SVM (RBF)			
k-NN			

Interpret which model is best and why.

Question 10 — Reflection
Answer briefly:
• When should we choose SVM?
• When NOT to use SVM?
• Why kernels matter?

Conclusion
After completing this assignment, students will:
understand SVM intuition and kernels
identify support vectors and margins
compare SVM with Logistic Regression & k-NN
evaluate models using meaningful metrics
This develops strong conceptual clarity for classification models.

Student Evaluation Rubric (Total: 10 Marks)

Criteria	Marks
Dataset loading & understanding	2
SVM implementation (linear + RBF)	3
Visualization + interpretation	3
Comparison & discussion	2

ASSIGNMENT 9: Decision Trees & Random Forests

Problem Statement

Decision Trees are easy to understand but can overfit.
Random Forests improve performance by combining many trees and reducing variance.
Students should understand:
how trees split data
tree depth and pruning
feature importance
difference between single tree and forest

Scenario
A data scientist works with:
Titanic dataset — predict survival
Heart Disease dataset — predict disease presence
The analyst must:
build Decision Tree & Random Forest models
visualize decision tree structure
study pruning and depth effects
compare accuracy and feature importance

Aim
To implement Decision Tree and Random Forest classifiers, visualize decision rules, and analyze feature importance.

Objectives
Students will be able to:

1. Load Titanic and Heart Disease datasets.
2. Perform preprocessing and encoding.
3. Train Decision Tree models.
4. Apply pruning / depth control.
5. Train Random Forest models.
6. Compare performance using accuracy, F1-score, and confusion matrix.
7. Interpret feature importance.
8. Explain why Random Forest often performs better.

Datasets
Titanic Dataset
Heart Disease Dataset

Tools Required
Python
Pandas / NumPy
Matplotlib / Seaborn
Scikit-learn

TASKS (Answer All — 10 Questions)
Question 1 — Load Dataset
Load Titanic or Heart disease dataset.

Display:
• first 10 rows
• number of rows & columns
• target variable

Question 2 — Preprocessing
Handle:
missing values
categorical encoding
Explain methods used.

Question 3 — Split Data
Split dataset into:
80% training
20% testing

Question 4 — Train Decision Tree
Train Decision Tree classifier.
Print:
• tree depth
• number of nodes

Question 5 — Visualize Tree
Visualize tree graph.
Describe:
• main decision nodes
• interpretation of splits

Question 6 — Pruning / Depth Control
Limit tree depth (example: max_depth=3)
Compare performance before and after pruning.
Explain effect.

Question 7 — Random Forest Model
Train Random Forest classifier.
Compare with Decision Tree.
Discuss improvement or decline in accuracy.

Question 8 — Feature Importance
Plot feature importance.
Answer:
• which features matter most?
• why?

Question 9 — Evaluation Metrics
Compute and interpret:
accuracy
precision
recall

F1-score
confusion matrix
Explain misclassifications.

Question 10 — Reflection
Short discussion:
• Why does Random Forest reduce overfitting?
• When would we still choose a Decision Tree?

Conclusion
After completing this assignment, students will:
understand decision trees clearly
control overfitting via pruning
interpret Random Forest behavior
analyze feature importance and evaluation metrics
This strengthens real-world classification understanding.

Student Evaluation Rubric (Total: 10 Marks)

Criteria	Marks
Data loading & preprocessing	2
Decision Tree + pruning	3
Random Forest + feature importance	3
Interpretation & clarity	2

ASSIGNMENT 10: Gradient Descent Optimization (California Housing)

Problem Statement

Machine learning models learn by minimizing a **cost (loss) function**.
However, different Gradient Descent strategies behave differently:

- Batch Gradient Descent — **stable but slow**
- Stochastic Gradient Descent — **fast but noisy**
- Mini-Batch Gradient Descent — **balanced approach**

Students must understand:
how parameters update
how loss decreases
why convergence curves differ

Scenario
A data scientist predicts **house prices** using California Housing dataset.
They want to choose the **best optimizer** for accuracy and training time.
They will:
train regression model
apply three gradient descent strategies
plot cost function vs iterations
analyze stability, variance, and speed

Aim
To compare Batch, Stochastic, and Mini-Batch Gradient Descent and analyze convergence performance.

Learning Objectives
After completing this experiment, students will be able to:

1. Load and preprocess California Housing dataset.
2. Define cost (MSE) and hypothesis function.
3. Implement:
    - Batch Gradient Descent (BGD)
    - Stochastic Gradient Descent (SGD)
    - Mini-Batch Gradient Descent (MBGD)
4. Plot and compare convergence graphs.
5. Evaluate convergence speed vs noise trade-off.
6. Interpret optimization choice in real projects.

Dataset
California Housing dataset (scikit-learn)

Software / Tools Required
Python
NumPy / Pandas
Matplotlib
Scikit-learn

EXPERIMENT TASKS (Answer ALL – 10 Questions)
Question 1 — Load Dataset

Load California Housing dataset.
Print:
• number of samples
• number of features
• sample records

Question 2 — Split and Normalize Data
Split into:
80% training
20% testing
Normalize features (important for GD).

Question 3 — Define Hypothesis & Cost Function
Implement:

- Hypothesis: $h(x) = XW$
- Cost: Mean Squared Error

Explain why normalization helps.

Question 4 — Batch Gradient Descent
Implement full-batch training:

- Update weights after **entire dataset**
- Track cost per iteration

Plot cost vs iterations.

Question 5 — Stochastic Gradient Descent
Update weights **per sample**.
Observe:
fast learning
noisy convergence
Plot cost curve.

Question 6 — Mini-Batch Gradient Descent
Choose batch sizes like:
batch_size = 16 or 32
Discuss why mini-batch is widely used.

Question 7 — Compare Convergence Graphs
Plot 3 curves on same graph:

- BGD
- SGD
- Mini-Batch

Discuss:
smoothness
speed
oscillations

Question 8 — Evaluate Final Model
Compare RMSE on test data for each optimizer.
Which performs best? Why?

Question 9 — Hyperparameter Discussion
Discuss:

- learning rate
- iterations
- batch size

How do wrong choices affect training?

Question 10 — Reflection
In 4–5 lines:
• When should we use **SGD**?
• When is **Batch GD** preferred?
• Why is **Mini-Batch** usually default?

Conclusion
After this experiment, students will:
understand gradient descent mathematically and practically
interpret convergence patterns
choose optimizers intelligently
explain speed vs variance trade-off
This builds strong foundations for deep learning and advanced ML.

Student Evaluation Rubric (10 Marks)

Criteria	Marks
Dataset preparation & normalization	2
Correct implementation of GD variants	4
Convergence plots & comparison	3
Clarity of explanation	1

ASSIGNMENT 11: Dimensionality Reduction using PCA

Problem Statement

High-dimensional datasets (like images) are:
expensive to store
slow to train
difficult to visualize
Principal Component Analysis (PCA) reduces dimensionality by projecting data onto **new axes (principal components)** that capture maximum variance.
Students should understand:
how PCA compresses data
how many components to keep
effect on accuracy
when PCA is useful

Scenario
A data scientist works with **Fashion-MNIST** clothing images.
Goal:
compress images
reduce noise
still keep classification accuracy reasonable
They will also compare results with **original MNIST digits**.

Aim
To apply PCA on image datasets and analyze variance, reconstruction quality, and classifier performance.

Learning Objectives
After completing the experiment, students will be able to:

1. Load Fashion-MNIST / MNIST datasets.
2. Normalize and reshape image data.
3. Apply PCA and interpret variance explained.
4. Reconstruct compressed images.
5. Visualize transformed data using principal components.
6. Train classifiers **before and after PCA**.
7. Compare performance and compression trade-offs.

Datasets
Fashion-MNIST
MNIST (Digits)
(Both available through keras.datasets or scikit-learn.)

Software / Tools
Python
NumPy / Pandas
Matplotlib / Seaborn
Scikit-learn
Keras (for datasets)

EXPERIMENT TASKS (Answer ALL — 10 Questions)

Question 1 — Load Dataset
Load Fashion-MNIST (and optionally MNIST).
Display:
• shape
• number of classes
• few sample images

Question 2 — Normalize Data
Scale pixel values into **0–1 range**.
Explain why normalization is important.

Question 3 — Reshape for PCA
Flatten images:
28 × 28 → 784 features
Explain why PCA needs vectors.

Question 4 — Apply PCA
Apply PCA and calculate:
principal components
cumulative variance explained
Plot variance vs components.
Answer:
how many components cover 90–95% variance?

Question 5 — Dimensionality Reduction
Transform original images into PCA space.
Compare shapes:
• before PCA
• after PCA
Discuss memory savings.

Question 6 — Image Reconstruction
Reconstruct images from reduced components.
Compare:
original
reconstructed
Explain loss of detail.

Question 7 — Train Classifier (Without PCA)
Train Logistic Regression / SVM on **original images**.
Record accuracy.

Question 8 — Train Classifier (With PCA Features)
Train same classifier using PCA-reduced features.
Compare accuracy vs training time.

Question 9 — Scatter Plot of Principal Components
Plot **first two principal components**.
Check if classes form clusters.

Question 10 — Reflection
Explain:
• When should PCA be used?
• When should we NOT use PCA?
• Is some information always lost?

Conclusion
After completing this assignment, students will:
understand dimensionality reduction
visualize principal components
balance compression vs accuracy
apply PCA confidently in ML workflows

Student Evaluation Rubric (10 Marks)

Criteria	Marks
Dataset prep & normalization	2
Correct PCA implementation & variance plots	3
Reconstruction + classification comparison	3
Interpretation & clarity	2

ASSIGNMENT 12: Manifold Learning — t-SNE & LLE

Problem Statement

High-dimensional datasets (like images) often lie on **curved manifolds** instead of simple flat planes.
Linear methods (like PCA) cannot always reveal true structure.
Manifold learning techniques such as:
t-SNE (t-Distributed Stochastic Neighbor Embedding)
LLE (Locally Linear Embedding)
help visualize complex datasets in **2D or 3D** while preserving neighborhood relationships.
Students must understand:

- why manifolds exist
- how data clusters form
- where these methods work well (and fail)

Scenario
A researcher is analyzing:
Olivetti Faces dataset
MNIST handwritten digits
Goal:
visualize natural clusters
discover patterns
compare algorithms
No labels are used during transformation — this is **unsupervised learning**.

Aim
To apply t-SNE and LLE to reduce high-dimensional data into 2D/3D and visualize cluster separation.

Learning Objectives
After completing this experiment, students will be able to:

1. Load MNIST / Olivetti Faces datasets.
2. Preprocess / flatten image data.
3. Apply **t-SNE**.
4. Apply **LLE**.
5. Visualize embeddings in 2D/3D.
6. Compare quality of cluster separation.
7. Discuss limitations and computational cost.

Datasets
Olivetti Faces (scikit-learn)
MNIST (Keras / scikit-learn)

Software / Tools Needed
Python
NumPy / Pandas
Matplotlib / Seaborn
Scikit-learn
(optional) Keras datasets

EXPERIMENT TASKS (Answer ALL — 10 Questions)

Question 1 — Load Dataset

Load **Olivetti Faces or MNIST**.

Print:

• number of samples
• image resolution
• number of classes

Display 6–10 sample images.

Question 2 — Flatten Images

Convert each image into a **single vector**.

Explain:

why manifold learning requires vectors.

Question 3 — Standardize Features

Normalize / scale image data.

Explain why scaling matters.

Question 4 — Apply t-SNE (2D)

Run t-SNE and create scatter plot.

Discuss:

Are similar samples grouped together?

Question 5 — Apply t-SNE (3D)

Plot 3D embedding.

Observe whether clusters become clearer.

Question 6 — Apply LLE

Run LLE on same dataset and visualize.

Comment:

more separated or more overlapped?

Question 7 — Compare t-SNE vs LLE

Fill a comparison table:

Aspect	t-SNE	LLE
Preserves	Local neighborhoods	Local geometry
Speed	Slow	Faster
Best for	Visualization	Structure discovery

Explain with examples.

Question 8 — Parameter Sensitivity

Change key parameters:

t-SNE → *perplexity*

LLE → *n_neighbors*

Explain impact on visualization.

Question 9 — Cluster Interpretation

If labels are available:

color points by class
check separation quality
Explain mis-clustered points.

Question 10 — Reflection
Answer in 4–5 lines:
• Why can’t we use these models for prediction directly?
• Why are they mainly used for visualization?

Conclusion
After this experiment, students will:
understand manifold learning concepts
visualize high-dimensional data
recognize when t-SNE or LLE is useful
interpret clusters meaningfully
This builds intuition for advanced ML topics.

Student Evaluation Rubric (10 Marks)

Criteria	Marks
Dataset preparation & preprocessing	2
Correct implementation of t-SNE & LLE	4
Visualization & comparison	3
Interpretation & clarity	1

ASSIGNMENT 13: Clustering — K-Means & DBSCAN

Problem Statement

Unlike classification, clustering does NOT use labels.
It groups data based on similarity.
Two widely-used algorithms:
K-Means — partitions data into *k* clusters
DBSCAN — finds dense regions and marks outliers as noise
Students need to understand:

- when clusters are spherical vs irregular
- effect of parameters
- identifying noise points
- interpreting cluster meaning

Scenario
A marketing analyst wants to segment mall customers based on:
age
income
spending score
Another dataset (Iris) helps verify clusters where ground truth labels exist.

Aim
To apply K-Means and DBSCAN, visualize cluster formation, and interpret cluster profiles.

Learning Objectives
After completing this experiment, students will:

1. Load Mall Customers & Iris datasets.
2. Preprocess and scale features.
3. Apply **K-Means** clustering.
4. Determine suitable *k* using the elbow method.
5. Apply **DBSCAN** clustering.
6. Visualize clusters and noise.
7. Compare clustering performance.
8. Interpret business meaning or cluster purity.

Datasets
Mall Customer Segmentation (CSV)
Iris Dataset (scikit-learn)

Tools Required
Python
NumPy / Pandas
Matplotlib / Seaborn
Scikit-learn

EXPERIMENT TASKS (Answer ALL — 10 Questions)
Question 1 — Load Dataset
Load Mall dataset.
Print:

• first 10 rows
• description of variables

Question 2 — Feature Selection & Scaling
Choose meaningful variables (example):
Annual Income, Spending Score
Scale features using StandardScaler.

Question 3 — Apply K-Means
Run K-Means with default clusters (k=3 or 4).
Plot scatter with cluster colors.

Question 4 — Determine Optimal K (Elbow Method)
Plot WCSS vs k.
Explain why elbow occurs.

Question 5 — Interpret Mall Clusters
Describe clusters:
• high income-high spend
• low income-low spend
• moderate shoppers

Question 6 — Apply DBSCAN
Use DBSCAN parameters:
eps
min_samples
Discuss noisy / outlier points.

Question 7 — Visualize DBSCAN Clusters
Plot clusters and noise separately.
Observe differences vs K-Means.

Question 8 — Apply Clustering on Iris Dataset
Cluster using K-Means and DBSCAN.
Compare with true labels using cluster purity or confusion matrix.

Question 9 — Comparison Table
Fill the table:

Property	K-Means	DBSCAN
Requires number of clusters	Yes	No
Handles noise	Weak	Strong
Cluster shapes	Mostly spherical	Arbitrary
Works well when	Balanced, separated	Varying density

Question 10 — Reflection
Explain in 4–5 lines:
• Which algorithm worked better and why?
• When would you prefer DBSCAN over K-Means?

Conclusion
Students understand practical clustering approaches and real-world business interpretation.

Student Evaluation Rubric (10 Marks)

Criteria	Marks
Implementation of algorithms	4
Visualizations & elbow analysis	3
Interpretation / cluster meaning	3

ASSIGNMENT 14: Anomaly Detection — Fraud Detection using GMM & Isolation Forest

Problem Statement

In real systems, most data is normal — only a **very small portion** is fraudulent or malicious.
Examples:
credit-card fraud
cyber-attacks
medical abnormalities
system failure alerts
Traditional models fail because they assume **balanced data**.
Anomaly detection techniques detect rare and unusual behavior without relying heavily on labeled data.
In this assignment, students explore:

- Gaussian Mixture Model (probabilistic anomaly detection)
- Isolation Forest (tree-based anomaly detection)

Scenario
A bank wants to detect **suspicious credit card transactions**.
Dataset is highly imbalanced:

- majority transactions = legitimate
- very few = fraud

Goal:
detect frauds
reduce false positives
compare models

Aim
To apply Gaussian Mixture Model and Isolation Forest and analyze anomaly detection performance.

Learning Objectives
After completing this experiment, students will:

1. Understand the nature of imbalanced datasets.
2. Load and preprocess Credit Card Fraud dataset.
3. Apply **Gaussian Mixture Model (GMM)** for anomaly detection.
4. Apply **Isolation Forest** to detect suspicious records.
5. Evaluate detection rate vs false positives.
6. Visualize anomaly separation.
7. Interpret advantages and limitations of both methods.

Datasets
Credit Card Fraud Detection Dataset (Kaggle)
KDD Cup 99 (optional extension — cyber intrusion detection)

Tools Required
Python
Pandas / NumPy

Matplotlib / Seaborn
Scikit-learn

EXPERIMENT TASKS (Answer ALL — 10 Questions)

Question 1 — Load Dataset

Load Kaggle credit card dataset.
Print:
• number of rows
• number of fraud vs non-fraud transactions
Explain imbalance problem.

Question 2 — Data Preparation

Perform:
feature selection
scaling (StandardScaler)
Explain why scaling is important.

Question 3 — Visualize Class Distribution

Plot bar chart showing fraud vs normal.
Discuss why accuracy alone is misleading.

Question 4 — Gaussian Mixture Model

Train GMM with two components:
fraud vs normal
Classify transactions by probability threshold.

Question 5 — Evaluate GMM

Calculate:
• precision
• recall
• F1-score
Explain false positives vs false negatives.

Question 6 — Isolation Forest

Train Isolation Forest.
Identify top suspicious records.

Question 7 — Compare Models

Prepare comparison:

Metric	GMM	Isolation Forest
Recall (fraud detected)		
False Positives		
Speed		

Discuss which is safer and why.

Question 8 — Visualization

Create scatter plot (PCA 2D) marking anomalies.
Observe:
✓ Are anomalies separated or mixed?

Question 9 — Threshold Experiment
Change detection threshold.
Explain:
more strict → fewer frauds detected
more relaxed → more false alarms

Question 10 — Reflection
Write short answers:
• Why are anomaly datasets always imbalanced?
• Which model performed better and why?
• Where could these methods be dangerous?

Conclusion
After this assignment, students will:
understand anomaly detection approaches
know how fraud detection is performed
evaluate trade-off between sensitivity and false alarms
apply GMM and Isolation Forest confidently

Student Evaluation Rubric (10 Marks)

Criteria	Marks
Dataset understanding & preprocessing	2
Implementation of both models	4
Evaluation & comparison	3
Interpretation	1

VIVA QUESTIONS

1. Which library is used for numerical computations in Python? → NumPy
2. What function in Pandas is used to read CSV files? → read_csv
3. Which Matplotlib function is used to plot a line graph? → plot()
4. What function in Pandas returns the shape of a DataFrame? → shape
5. Which library is commonly used for statistical data visualization? → Seaborn
6. What type of probability distribution follows a bell curve? → Normal
7. Which test is used to compare two means? → t-test
8. What is the measure of dispersion in a dataset? → Variance
9. Which statistical metric is used to measure data spread? → Standard Deviation
10. What does the R-squared value indicate in regression? → Goodness of Fit
11. What is the rank of a square matrix with all zero rows? → Zero
12. What is the determinant of an identity matrix? → One
13. Which operation is used to transform a vector? → Matrix Multiplication
14. What do you call a matrix with all elements zero except the diagonal? → Diagonal Matrix
15. What is the inverse of a non-invertible matrix? → Not Defined
16. What is the derivative of a constant function? → Zero
17. Which function is used to optimize loss in ML? → Gradient Descent
18. What is the partial derivative of x^2 with respect to x? → 2x
19. What is the gradient of a function? → Vector of Derivatives
20. What is the term for a point where a function's derivative is zero? → Critical Point
21. Which package manager is used in Anaconda? → Conda
22. What is the default IDE in Anaconda? → Jupyter Notebook
23. Which command lists installed packages in Anaconda? → conda list
24. What is the primary use of VS Code in ML? → Code Editing
25. Which library is used for deep learning in Python? → TensorFlow
26. Which Pandas function shows the first five rows? → head()
27. Which metric measures data central tendency? → Mean
28. Which chart is best for visualizing categorical data? → Bar Chart
29. What type of plot is used for correlation analysis? → Heatmap
30. Which function finds missing values in a dataset? → isnull()
31. What is the target variable in regression? → Continuous
32. What is the formula for linear regression? → $y = mx + c$
33. Which regression model minimizes squared error? → OLS (Ordinary Least Squares)
34. What type of regression prevents overfitting? → Ridge Regression
35. What is the loss function used in linear regression? → Mean Squared Error (MSE)
36. Which ML algorithm is used for spam detection? → Naïve Bayes
37. What is the decision boundary in SVM? → Hyperplane
38. What is the full form of KNN? → K-Nearest Neighbors
39. Which metric is used for classification accuracy? → F1-Score
40. What does the ROC curve represent? → Model Performance
41. What is the number of clusters in K-Means called? → K
42. What type of clustering does DBSCAN perform? → Density-Based
43. Which algorithm uses hierarchical clustering? → Agglomerative Clustering
44. What is the primary metric used in clustering evaluation? → Silhouette Score
45. What does PCA stand for? → Principal Component Analysis
46. Which technique reduces data dimensions? → PCA
47. What does t-SNE stand for? → t-Distributed Stochastic Neighbor Embedding
48. What is the main purpose of dimensionality reduction? → Reduce Complexity
49. Which technique is used before PCA for better results? → Feature Scaling

50. What is the key hyperparameter in PCA? → Number of Components
51. What is the most commonly used deployment framework? → Flask
52. Which cloud platform offers ML model deployment? → Google Cloud (GCP)
53. What is a trained model saved as in TensorFlow? → .h5 or .pb
54. What is the purpose of API in ML deployment? → Model Serving
55. Which library is used to serialize ML models? → Pickle