# MACHINE LEARNING

24AD2204

STUDENT ID: ACADEMIC YEAR: 2025−26

STUDENT NAME: EVEN SEMESTER

**K L E F**

**KONERU LAKSHMAIAH EDUCATION FOUNDATION**

**(Deemed to be university estd, u/s, 3 of the UGC Act, 1956)**

**(NAAC Accredited “A” Grade University)**

## Vision

To be a globally renowned university.

## Mission:

To impart quality higher education and to undertake research and extension with emphasis on application and innovation that cater to the emerging societal needs through all-round development of the students of all sections enabling them to be globally competitive and socially responsible citizens with intrinsic values.

# Table of Contents

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#1: INSTALL ANACONDA AND DEMONSTRATE BASIC PYTHON PROGRAMS

**Aim/Objective:**

Install ANACONDA and demonstrate basic syntax, data types, control flow statements, functions, data structures, and usage of libraries and modules in Python.

**Description:**

Students will install Anaconda and implement basic Python programs covering syntax, data types, control structures, functions, data structures, and use standard libraries.

**Pre-Requisites:**

- Basics Basic Computer Skills
- Logical Thinking
- Basic Mathematics
- Familiarity with Programming Environments

**Pre-Lab:**

1. What are the basic data types in Python? Give examples.

2. Why is indentation important in Python syntax?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. How are if, elif, and else used for decision making?

4. What are the differences between for and while loops?

5. How do you accept user input and display output in Python?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Display “Hello World” and your name.
- Perform arithmetic operations between two user inputs.
- Determine if a number is even or odd using conditional statements.
- Find the factorial of a number using loops.
- Demonstrate indexing and slicing in lists.
- Create a dictionary to store student names and marks, and display the highest scorer.
- Import and use math and random libraries to perform basic operations.

**Procedure/Program:**

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. What are the rules for variable naming in Python?
2. Explain mutable and immutable data types with examples.
3. What is the use of the range() function?
4. Explain indexing and slicing in Python lists.
5. How do you define and access dictionary elements?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Post-Lab:**

1. Write a program to calculate the sum of digits of a given number.
2. Write a program to print a multiplication table up to 10 for a user-entered number.
3. Create a list of integers, remove duplicates using a set, and display both lists.
4. Write a program to accept student details (roll number, name) as a tuple and print them formatted.
5. Create a dictionary of products and their prices, and display products costing more than Rs.100.

**Procedure/Program:**

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#2: IMPLEMENT K-FOLD CROSS-VALIDATION AND BOOTSTRAP SAMPLING

**Aim/Objective:**

Implement k-Fold Cross-Validation to reliably estimate the performance of a k-NN classifier. Use Bootstrap sampling to assess the stability of model parameters in a simple linear regression.

**Description:**

Students will implement k-Fold Cross-Validation to evaluate k-NN classifier performance and use Bootstrap sampling to analyze parameter stability in linear regression.

**Pre-Requisites:**

- Understanding of cross-validation
- Basic knowledge of k-NN and linear regression
- Familiarity with scikit-learn

**Pre-Lab:**

1. What is the purpose of k-Fold Cross-Validation?

2. How does Bootstrap sampling work?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What is the bias-variance tradeoff in model evaluation?

4. Write Python code to split data into k folds.

5. How do you calculate confidence intervals using Bootstrap?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load a dataset (e.g., Iris).
- Implement k-Fold Cross-Validation for k-NN classifier.
- Calculate and print average accuracy across folds.
- Implement Bootstrap sampling for linear regression parameters.
- Plot confidence intervals for regression coefficients.

**Procedure/Program**:

**Sample VIVA-VOCE Questions (In-Lab):**

1. Difference between k-Fold and Leave-One-Out Cross-Validation.
2. Why use Bootstrap for parameter stability?
3. How does k affect cross-validation results?
4. What are confidence intervals in Bootstrap?
5. When would you prefer Bootstrap over cross-validation?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Post-Lab:**

1. Compare k-Fold results with train-test split.
2. Implement stratified k-Fold for imbalanced data.
3. Calculate Bootstrap standard error for regression parameters.
4. Visualize Bootstrap distribution of parameters.
5. Write a function to automate k-Fold for any classifier.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#3: PLOT LEARNING CURVES FOR POLYNOMIAL REGRESSION

**Aim/Objective:**

Diagnose a model's performance by plotting learning curves for polynomial regression models of varying complexity, visually demonstrating the tradeoff between underfitting and overfitting.

**Description:**

Students will generate learning curves for polynomial regression to identify underfitting and overfitting.

**Pre-Requisites:**

• Understanding of polynomial regression

• Knowledge of learning curves

• Matplotlib for visualization

**Pre-Lab:**

1. What are learning curves?

2. How do learning curves indicate underfitting/overfitting?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What is polynomial regression?

4. Write code to generate polynomial features.

5. How to plot learning curves in sklearn?

**In-Lab:**

Write Python programs to:

- Generate synthetic data.
- Fit polynomial regression models (degrees 1, 3, 10).
- Plot learning curves (training vs validation error).
- Identify underfitting and overfitting visually.
- Find the optimal polynomial degree.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

**Sample VIVA-VOCE Questions (In-Lab):**

1. What does high bias indicate in learning curves?
2. What does high variance indicate?
3. How to address underfitting?
4. How to address overfitting?
5. Role of training set size in learning curves.

**Post-Lab:**

- Apply learning curves to real dataset.
- Use cross-validated learning curves.
- Plot learning curves for different sample sizes.
- Compare with validation curves.
- Automate optimal degree selection.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured ______ out of 50** |
| | **Signature of the Evaluator with Date** |

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#4: PREDICT HOUSE PRICES AND CLASSIFY IRIS WITH REGRESSION MODELS

**Aim/Objective:**

Predict house prices using Linear Regression and classify the Iris dataset using Logistic Regression. Evaluate both models using appropriate metrics (e.g., $R^2$, MSE, Accuracy, F1-Score) and analyse feature weights.

**Description:**

Students will implement linear regression for predicting continuous target values and logistic regression for classification, perform evaluation using various metrics, and analyze model coefficients.

**Pre-Requisites:**

- Understanding of regression and classification concepts
- Familiarity with sklearn, pandas, numpy libraries
- Basic Python programming skills

**Pre-Lab:**

1. What is the difference between regression and classification?

2. Write a Python snippet to import Linear Regression and Logistic Regression from sklearn.

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. How is $R^2$ score interpreted in regression?

4. What do precision and recall measure in classification?

5. How are feature weights interpreted in linear models?

**In-Lab:**

Write Python programs to:

- Load Boston housing dataset and perform linear regression.
- Calculate and interpret $R^2$, MSE, MAE, and RMSE.
- Analyze feature coefficients and their significance.
- Load Iris dataset and perform logistic regression for multiclass classification.
- Calculate accuracy, precision, recall, F1-score, and confusion matrix.
- Analyze feature weights and their contribution to each class.
- Visualize decision boundaries for logistic regression.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. What does a negative $R^2$ score indicate?
2. 2. Why is MSE sensitive to outliers?
3. 3. How is multiclass classification handled in logistic regression?
4. 4. What is the difference between micro and macro averaging of F1-score?
5. 5. How do you interpret a coefficient in logistic regression?

**Post-Lab:**

- Implement regularization (L1 and L2) for both linear and logistic regression.
- Compare performance metrics before and after regularization.
- Plot learning curves for both models.
- Perform feature selection based on coefficient magnitudes.
- Implement polynomial features for linear regression and compare results.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#5: PERFORM FEATURE ENGINEERING ON TITANIC DATASET

**Aim/Objective:**

Perform comprehensive feature engineering on the Titanic dataset by creating new features (e.g., title, family size), handling missing data, and applying encoding/scaling. Use filter methods (e.g., correlation) and wrapper methods (e.g., Recursive Feature Elimination) to select the most impactful features.

**Description:**

Students will apply advanced feature engineering techniques including feature creation, transformation, selection methods, and evaluate their impact on model performance.

**Pre-Requisites:**

• Understanding of feature engineering concepts

• Familiarity with pandas, numpy, and sklearn libraries

• Knowledge of data preprocessing techniques

**Pre-Lab:**

1. What is feature engineering and why is it important?

2. How would you extract titles from names in the Titanic dataset?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What are filter, wrapper, and embedded feature selection methods?

4. How does Recursive Feature Elimination (RFE) work?

5. What is the difference between one-hot encoding and label encoding?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load Titanic dataset and perform exploratory data analysis.
- Create new features: Title (Mr, Mrs, Miss, etc.), Family Size, Is Alone, Deck from Cabin.
- Handle missing values using advanced imputation techniques.
- Apply appropriate encoding (one-hot, label, target) for categorical variables.
- Scale numerical features using standardization and normalization.
- Apply filter methods: correlation with target, mutual information.
- Apply wrapper method: Recursive Feature Elimination with cross-validation.
- Compare model performance before and after feature selection.

**Procedure/Program:**

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why create Family Size feature from SibSp and Parch?
2. How does target encoding differ from one-hot encoding?
3. What is the advantage of RFE over simple correlation filtering?
4. How do you handle high-cardinality categorical features?
5. What is feature interaction and how can it be created?

**Post-Lab:**

- Implement feature selection using L1 regularization (LASSO).
- Create interaction features and test their significance.
- Apply PCA for dimensionality reduction and compare with feature selection.
- Implement automated feature engineering using feature tools library.
- Build a pipeline that includes all feature engineering steps.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#6: MONTE CARLO SIMULATION FOR PI & BAYESIAN COIN BIAS ESTIMATION

**Aim/Objective:**

Use a Monte Carlo simulation to approximate the value of Pi ($\pi$) by randomly sampling points within a square. Implement Bayesian concept learning for a simple problem like estimating the bias of a coin from a sequence of flip outcomes.

**Description:**

Students will implement probabilistic methods including Monte Carlo simulation and Bayesian inference to solve computational problems and learn from data.

**Pre-Requisites:**

• Understanding of probability and statistics

• Familiarity with numpy and matplotlib

• Knowledge of Bayesian inference concepts

• Basic Python programming skills

**Pre-Lab:**

1. Explain the Monte Carlo method for estimating Pi.

2. What is Bayesian inference and how does it differ from frequentist approach?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What is a prior distribution in Bayesian statistics?

4. How does the likelihood function update the prior?

5. What is Markov Chain Monte Carlo (MCMC)?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Implement Monte Carlo simulation to estimate Pi:
- Generate random points in a unit square
- Count points inside quarter circle
- Calculate Pi estimate and error
- Plot convergence with increasing sample size
- Implement Bayesian coin flip experiment:
- Define prior distribution (Beta distribution)
- Simulate coin flip outcomes
- Update posterior distribution after each flip
- Plot prior, likelihood, and posterior distributions
- Calculate credible intervals
- Compare Bayesian estimate with maximum likelihood estimate

**Procedure/Program:**

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why does Monte Carlo error decrease with more samples?
2. What is the Central Limit Theorem's role in Monte Carlo?
3. How does choice of prior affect Bayesian inference?
4. What is conjugate prior and why is it useful?
5. How do you determine when the Monte Carlo estimate has converged?

**Post-Lab:**

- Extend Monte Carlo to estimate integral of arbitrary function.
- Implement Metropolis-Hastings MCMC sampling.
- Apply Bayesian inference to A/B testing problem.
- Compare computational efficiency of different Monte Carlo techniques.
- Implement sequential Bayesian updating for streaming data.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#7: CLASSIFY IRIS USING K-NEAREST NEIGHBORS AND DECISION TREE

**Aim/Objective:**

Classify the Iris dataset using a k-Nearest Neighbors (k-NN) classifier, tuning the hyperparameter 'k' via cross-validation. Build a Decision Tree classifier for the same task, visualise the tree, and compare the performance and interpretability of both models.

**Description:**

Students will implement and compare two fundamentally different classification algorithms, understand hyperparameter tuning, and analyze model interpretability.

**Pre-Requisites:**

• Understanding of k-NN and decision tree algorithms

• Familiarity with sklearn library

• Knowledge of cross-validation and hyperparameter tuning

• Basic Python programming skills

**Pre-Lab:**

1. How does the k-NN algorithm make predictions?

2. What is the effect of different distance metrics in k-NN?

3. How does a decision tree split nodes?

4. What are impurity measures (Gini, entropy) in decision trees?

5. How does cross-validation help in hyperparameter tuning?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load Iris dataset and perform train-test split.
- Implement k-NN classifier:
- Try different k values (1, 3, 5, 7, 9, 11)
- Use cross-validation to find optimal k
- Evaluate with different distance metrics
- Visualize decision boundaries for different k
- Implement Decision Tree classifier:
- Train with different max_depth values
- Visualize tree structure using graphviz

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why is k-NN called a lazy learner?
2. What is the curse of dimensionality in k-NN?
3. How does decision tree handle categorical vs numerical features?
4. What is information gain and how is it calculated?
5. When would you prefer decision tree over k-NN?

**Post-Lab:**

- Implement weighted k-NN and compare results.
- Apply decision tree to regression problem (CART).
- Implement post-pruning using cost-complexity pruning.
- Compare k-NN performance with and without feature scaling.
- Create ensemble of both models using voting classifier.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#8: COMPARE DECISION TREE, RANDOM FOREST, AND ADABOOST

**Aim/Objective:**

Compare the accuracy of a single Decision Tree against a Random Forest and an AdaBoost classifier on a structured dataset. Tune their key parameters, measure and visualise feature importance, and discuss the difference between bagging and boosting.

**Description:**

Students will implement and compare ensemble methods, understand their working principles, tune hyperparameters, and analyze feature importance.

**Pre-Requisites:**

- Understanding of decision trees
- - Knowledge of ensemble methods concepts
- - Familiarity with sklearn ensemble module
- - Basic Python programming skills

**Pre-Lab:**

1. What is the difference between bagging and boosting?

2. How does Random Forest reduce overfitting compared to single decision tree?

3. Explain the AdaBoost algorithm.

4. What is out-of-bag error in Random Forest?

5. How are feature importance’s calculated in ensemble methods?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load a structured dataset (e.g., Wine, Breast Cancer).
- Implement single Decision Tree as baseline.
- Implement Random Forest: Tune n_estimators, max_depth, max_features, calculate out-of-bag score, analyze feature importance
- Implement AdaBoost: Tune n_estimators, learning_rate, use decision stump as base estimator, analyze sample weights evolution
- Compare all three models: Accuracy, precision, recall, F1-score, training time vs prediction time, feature importance comparison, overfitting analysis using learning curves
- Visualize: Feature importance bar plots, decision boundaries for 2D projections, learning curves comparison

**Procedure/Program:**

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why does Random Forest use bootstrap sampling?
2. What is the effect of increasing n_estimators?
3. How does AdaBoost handle misclassified samples?
4. What is the tradeoff between bias and variance in ensembles?
5. When would you prefer boosting over bagging?

**Post-Lab:**

- Implement Gradient Boosting and compare with AdaBoost.
- Tune hyperparameters using GridSearchCV and RandomizedSearchCV.
- Implement feature selection based on ensemble feature importance.
- Compare computational requirements of different ensembles.
- Implement custom voting classifier combining all three methods.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#9: CLASSIFY DATA WITH SVM USING DIFFERENT KERNELS

## Aim/Objective:

Classify non-linearly separable synthetic data points using Support Vector Machines (SVM) with linear, polynomial, and RBF kernels. Visualise the resulting decision boundaries, margins, and support vectors to understand the effect of the kernel trick.

## Description:

Students will implement SVM with different kernels, understand the kernel trick, visualize decision boundaries, and analyze the role of support vectors.

## Pre-Requisites:

• Understanding of SVM concepts

• Knowledge of kernel methods

• Familiarity with sklearn SVM module

• Basic Python programming and matplotlib skills

## Pre-Lab:

1. What is the maximum margin principle in SVM?

2. Explain the kernel trick in SVM.

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What are support vectors and why are they important?

4. How does the C parameter affect SVM?

5. Compare linear, polynomial, and RBF kernels.

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Generate non-linearly separable synthetic datasets (moons, circles, XOR).
- Implement SVM with linear kernel: Visualize decision boundary and margin, Identify support vectors, Analyze effect of C parameter
- Implement SVM with polynomial kernel: Experiment with different degrees (2, 3, 5), Visualize decision boundaries, Count support vectors
- Implement SVM with RBF kernel: Experiment with gamma parameter, Visualize decision boundaries, Analyze kernel space transformation
- Compare all kernels: Accuracy on test data, Number of support vectors, Training time, Decision boundary complexity
- Create comprehensive visualizations: Side-by-side comparison of all kernels, Support vectors highlighted, Margin boundaries shown, Confidence scores contour plots

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why RBF kernel does often performs well on non-linear data?
2. How does gamma parameter affect RBF kernel?
3. What happens when C is too large or too small?
4. How does kernel choice affect computational complexity?
5. Why are support vectors important for prediction?

**Post-Lab:**

- Implement custom kernel function.
- Apply SVM to real dataset and compare with other classifiers.
- Implement multi-class SVM using one-vs-one and one-vs-rest.
- Tune hyperparameters using grid search with cross-validation.
- Compare SVM performance with and without feature scaling.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#10: PERFORM CUSTOMER SEGMENTATION USING CLUSTERING

**Aim/Objective:**

Perform customer segmentation by applying K-Means and Hierarchical Clustering to a marketing dataset. Determine the optimal number of clusters using the elbow method and silhouette score, and visualize the clusters using PCA.

**Description:**

Students will implement unsupervised learning techniques for customer segmentation, evaluate cluster quality, and visualize results using dimensionality reduction.

**Pre-Requisites:**

- Understanding of clustering concepts
- Knowledge of K-Means and hierarchical clustering
- Familiarity with sklearn clustering module
- Basic Python programming and visualization skills

**Pre-Lab:**

1. What is the difference between supervised and unsupervised learning?

2. How does K-Means clustering work?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What are dendrograms in hierarchical clustering?

4. Explain the elbow method for determining optimal clusters.

5. What does silhouette score measure?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load marketing/customer dataset (e.g., Mall Customers, RFM data).
- Perform data preprocessing: scaling, handling categorical variables.
- Apply K-Means clustering: Determine optimal k using elbow method, Calculate silhouette scores for different k, Visualize clusters in 2D using PCA, Analyze cluster centers and characteristics
- Apply Hierarchical clustering: Create dendrogram using different linkage methods, Cut dendrogram to obtain clusters, Compare with K-Means results
- Evaluate clustering: Compare silhouette scores, Analyze cluster separation, Interpret business meaning of each cluster
- Create comprehensive visualizations: Elbow plot with inertia values, Silhouette plots for each k, PCA scatter plots colored by clusters, Parallel coordinates plot for cluster profiles, Heatmap of cluster centers

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| --- | --- | --- | --- |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why is feature scaling important for K-Means?
2. What are the limitations of K-Means clustering?
3. How does hierarchical clustering handle outliers?
4. What is the difference between single and complete linkage?
5. How would you validate clustering results without ground truth?

**Post-Lab:**

- Implement DBSCAN clustering and compare results.
- Apply clustering to image segmentation problem.
- Implement Gaussian Mixture Models (GMM) for soft clustering.
- Create customer personas based on cluster analysis.
- Implement clustering evaluation using external indices if labels available.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#11: IMPLEMENT APRIORI FOR ASSOCIATION RULE MINING

**Aim/Objective:**

Implement the Apriori algorithm on a market basket transactions dataset (e.g., store sales data) to discover interesting association rules. Filter rules based on support, confidence, and lift, and interpret the business implications.

**Description:**

Students will implement association rule mining using Apriori algorithm, discover patterns in transactional data, and interpret business insights for market basket analysis.

**Pre-Requisites:**

- Understanding of association rule mining
- Knowledge of support, confidence, lift metrics
- Familiarity with pandas and mlxtend libraries
- Basic Python programming skills

**Pre-Lab:**

1. What is market basket analysis?

2. Define support, confidence, and lift with formulas.

3. What is the Apriori principle?

4. How do you generate candidate itemsets?

5. What are strong association rules?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load market basket transactions dataset (e.g., Groceries, Online Retail).
- Preprocess data: convert to one-hot encoded format.
- Implement Apriori algorithm (or use mlxtend library): Generate frequent itemsets with minimum support threshold, Generate association rules from frequent itemsets, Filter rules based on confidence and lift thresholds
- Analyze discovered rules: Sort rules by support, confidence, lift, Identify most interesting rules, Analyze rule characteristics (antecedent, consequent)
- Create visualizations: Support vs confidence scatter plot, Top rules visualization, Itemset network graph, Heatmap of item co-occurrences
- Interpret business implications: Product placement recommendations, Cross-selling opportunities, Inventory management insights, Customer behavior patterns

**Procedure/Program:**

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why is lift considered better than confidence?
2. What is the downward closure property in Apriori?
3. How does minimum support affect rule discovery?
4. What are redundant rules and how to eliminate them?
5. How would you handle large itemsets efficiently?

**Post-Lab:**

- Implement FP-Growth algorithm and compare with Apriori.
- Create association rules for sequential patterns.
- Apply rule mining to text data (document co-occurrence).
- Implement multi-level association rule mining.
- Create recommendation system based on association rules.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

# EXPERIMENT#12: DIMENSIONALITY REDUCTION ON FASHION-MNIST

**Aim/Objective:**

Reduce the high-dimensionality of the Fashion-MNIST image dataset to 2D using PCA and t-SNE. Plot the reduced data, colouring points by their true label, and interpret the visualisations in terms of retained variance (PCA) and cluster preservation (t-SNE).

**Description:**

Students will apply dimensionality reduction techniques to high-dimensional image data, visualize results, and understand the differences between linear and non-linear methods.

**Pre-Requisites:**

- Understanding of dimensionality reduction concepts
- Knowledge of PCA and t-SNE algorithms
- Familiarity with sklearn decomposition and manifold modules
- Basic Python programming and matplotlib skills

**Pre-Lab:**

1. What is the curse of dimensionality?

2. Explain how PCA works mathematically.

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What is the difference between PCA and t-SNE?

4. How is explained variance calculated in PCA?

5. What are perplexity and learning rate in t-SNE?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load Fashion-MNIST dataset (70,000 28x28 grayscale images).
- Preprocess data: flatten images, normalize pixel values.
- Apply PCA: Fit PCA to training data, Plot explained variance ratio vs number of components, Determine components needed for 95% variance retention, Reduce to 2D and 3D for visualization, Create biplot showing component loadings
- Apply t-SNE: Experiment with different perplexity values (5, 30, 50), Adjust learning rate for optimal convergence, Reduce to 2D for visualization, Compare different initialization methods
- Create comprehensive visualizations: 2D scatter plots colored by class labels, Side-by-side comparison of PCA vs t-SNE, 3D plots for PCA, Class separation analysis, Outlier detection in reduced space
- Quantitative analysis: Calculate silhouette scores in reduced space, Measure class separation metrics, Compare reconstruction error for PCA, Analyze computational time for both methods

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. Why does t-SNE preserve local structure better than PCA?
2. How do you interpret the explained variance plot?
3. What happens when perplexity is too high or too low?
4. Why is t-SNE non-deterministic?
5. How would you choose between PCA and t-SNE for a given problem?

**Post-Lab:**

- Apply Implement UMAP and compare with PCA and t-SNE.
- Apply autoencoders for dimensionality reduction.
- Use reduced features for classification and compare accuracy.
- Implement kernel PCA for non-linear dimensionality reduction.
- Create interactive visualization using plotly.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

# EXPERIMENT#13: REDUCE ANOMALY DETECTION IN CREDIT CARD DATA

**Aim/Objective:**

Detect anomalies in a credit card transaction dataset using the Isolation Forest algorithm. Evaluate the model's performance using a confusion matrix, ROC curve, and AUC, and report on the false positive rate.

**Description:**

Students will implement anomaly detection for fraud detection, evaluate model performance on imbalanced data, and analyze tradeoffs between detection rate and false alarms.

**Pre-Requisites:**

- Understanding of anomaly detection concepts
- Knowledge of imbalanced classification metrics
- Familiarity with sklearn ensemble module
- Basic Python programming and evaluation skills

**Pre-Lab:**

1. What are the challenges of anomaly detection?

2. How does Isolation Forest work differently from normal Random Forest?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What is the contamination parameter?

4. Explain precision-recall tradeoff in anomaly detection.

5. What metrics are appropriate for imbalanced datasets?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Load credit card fraud dataset (highly imbalanced).
- Perform exploratory data analysis: Check class distribution, Analyze feature distributions for normal vs fraud, Handle class imbalance if needed
- Implement Isolation Forest: Tune contamination parameter, Adjust n_estimators and max_samples, Calculate anomaly scores, Set threshold for classification
- Evaluate model performance: Generate confusion matrix, Calculate precision, recall, F1-score, Plot ROC curve and calculate AUC, Plot precision-recall curve, Calculate false positive rate and false negative rate
- Analyze results: Feature importance for anomaly detection, Characterize detected anomalies, Compare with random guessing baseline, Analyze cost of false positives vs false negatives
- Create comprehensive visualizations: Distribution of anomaly scores, ROC and PR curves, Confusion matrix heatmap, Feature importance plot, Parallel coordinates plot of detected frauds

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. What Why is Isolation Forest suitable for anomaly detection?
2. How does the algorithm isolate anomalies?
3. What is the effect of increasing n_estimators?
4. Why is AUC more informative than accuracy for imbalanced data?
5. How would you set the threshold in practice?

**Post-Lab:**

- Implement One-Class SVM and compare with Isolation Forest.
- Apply ensemble of multiple anomaly detection methods.
- Implement adaptive thresholding based on business costs.
- Create feature engineering specific for fraud detection.
- Implement real-time anomaly detection pipeline.

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| | |
|---|---|
| **Evaluator Remark(if Any):** | **Marks Secured______ out of 50** |
| | **Signature of the Evaluator with Date** |

# EXPERIMENT#14: GAUSSIAN PROCESS REGRESSION FOR NOISY SINE PREDICTION

**Aim/Objective:**

Apply Gaussian Process Regression to predict a noisy sine wave, plotting the mean prediction and uncertainty intervals. Compare its performance and properties against previously implemented models like Linear Regression and Random Forest.

**Description:**

Students will implement Gaussian Process Regression, understand probabilistic predictions, quantify uncertainty, and compare with traditional regression methods.

**Pre-Requisites:**

• Understanding of Gaussian processes

• Knowledge of kernel methods

• Familiarity with sklearn gaussian_process module

• Basic Python programming and matplotlib skills

**Pre-Lab:**

1. What are Gaussian Processes?

2. How do GPs provide uncertainty estimates?

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

3. What is the role of kernel in GP regression?

4. How does GP differ from traditional regression?

5. What are the computational challenges of GPs?

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**In-Lab:**

Write Python programs to:

- Generate synthetic data: Sine wave with added Gaussian noise, Training and test sets with different distributions
- Implement Gaussian Process Regression: Experiment with different kernels (RBF, Matern, Rational Quadratic), Tune kernel hyperparameters, Fit GP model to training data, Make predictions with uncertainty estimates
- Implement comparison models: Linear Regression (polynomial features), Random Forest Regression, Support Vector Regression
- Evaluate all models: Calculate MSE, MAE, $R^2$ on test set, Compare prediction intervals, Analyze computational time, Assess extrapolation capabilities
- Create comprehensive visualizations: GP predictions with confidence intervals, Side-by-side comparison of all models, Kernel matrix visualization, Learning curves for all models, Residual analysis
- Special analysis for GP: Effect of different kernels on predictions, Uncertainty calibration assessment, Hyperparameter sensitivity analysis, Scaling to larger datasets

**Procedure/Program:**

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

- **Data and Results:**
  *(Record outputs, screenshots, and observations here)*

- **Analysis and Inferences:**
  *(Write your learning outcomes and inferences here)*

| | | | |
|---|---|---|---|
| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

**Sample VIVA-VOCE Questions (In-Lab):**

1. How does GP handle uncertainty quantification?
2. What is the difference between epistemic and aleatoric uncertainty?
3. How does kernel choice affect GP predictions?
4. Why is GP computationally expensive for large datasets?
5. When would you prefer GP over other regression methods?

**Post-Lab:**

- Implement sparse Gaussian Processes for scaling.
- Apply GP to real-world regression problem.
- Implement Bayesian optimization using GP.
- Compare GP with neural network uncertainty methods.
- Create interactive visualization of GP predictions.

**Procedure/Program:**

| Student ID | <TOBEFILLEDBYSTUDENT> | Course Title | Machine Learning |
|---|---|---|---|
| Student Name | <TOBEFILLEDBYSTUDENT> | Course Code | 22AD2204 |

## Data and Results:

## Analysis and Inferences:

| Evaluator Remark(if Any): | Marks Secured ______ out of 50 |
|---|---|
| | Signature of the Evaluator with Date |