Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Exercise No. 2			
Topic:	**Topic 2: Supervised Learning Techniques**	**Week No.**	4
Course Code:	**CSST102**	**Term:**	1st Semester
Course Title:	**Basic Machine Learning**	**Academic Year:**	2024-2025
Student Name		**Section**	
Due date		**Points**	

Exercises for K-Nearest Neighbors (KNN) and Logistic Regression on Breast Cancer Diagnosis Dataset

Exercise 1: Data Exploration and Preprocessing

1. **Load and Explore the Data:**
   - Load the **Breast Cancer Diagnosis Dataset** into a pandas DataFrame.
   - Display the first 10 rows and check for missing values.
   - Explore the distribution of features using descriptive statistics (mean, std, min, max, etc.).

   Task:
   - Summarize the dataset: How many instances and features are there? Are there any missing values?
   - Which features have the highest variance and might be the most important for classification?

2. **Preprocessing:**
   - Drop irrelevant columns (e.g., id and unnamed columns).
   - Convert the target variable diagnosis (M = Malignant, B = Benign) into numerical format.
   - Normalize or standardize the features to ensure they're on the same scale (optional).

   Task:
   - After preprocessing, split the dataset into 80% training and 20% testing data using train_test_split.

Exercise 2: Implementing the K-Nearest Neighbors (KNN) Model

1. **Implement a KNN Classifier:**
   - Use the KNeighborsClassifier from scikit-learn.
   - Train the KNN classifier using the training data (use n_neighbors=5 by default).
   - Predict the tumor diagnosis on the test data.

   Task:
   - Calculate the **accuracy** of the KNN model.
   - Present the **confusion matrix** for the predictions.

2. **Experiment with Different Values of n_neighbors:**
   - Vary the number of neighbors (e.g., 3, 5, 7, 9) and observe the model's performance.

   Task:
   - Plot a graph showing how accuracy changes with different values of n_neighbors.
   - What is the optimal value of n_neighbors based on the accuracy?

Exercise 3: Implementing Logistic Regression

1. **Implement a Logistic Regression Classifier:**
   - Use the LogisticRegression from scikit-learn.
   - Train the model using the training data and predict the test data labels.

   Task:
   - Calculate the **accuracy** of the Logistic Regression model.
   - Present the **confusion matrix** and **classification report** (precision, recall, F1-score).

2. **Comparison of KNN and Logistic Regression:**
   - Compare the performance (accuracy, precision, recall) of both models on the same dataset.

   Task:
   - Which model performs better in terms of accuracy and F1-score?
   - Discuss which model you think is more appropriate for this classification problem and why.

Exercise 4: Hyperparameter Tuning and Cross-Validation

1. **Grid Search for Hyperparameter Tuning:**
   - Use GridSearchCV to tune the hyperparameters of the KNN model.
   - Tune parameters such as n_neighbors, weights, and p (for distance metric).

   Task:
   - Perform cross-validation using GridSearchCV to find the best hyperparameters for KNN.
   - Report the best combination of parameters and the corresponding accuracy.

2. **Cross-Validation for Logistic Regression:**
   - Perform **k-fold cross-validation** on the Logistic Regression model (use k=5).

   Task:
   - Report the cross-validated accuracy for the Logistic Regression model.

Exercise 5: Decision Boundary Visualization

1. **Visualizing the Decision Boundary:**
   - Reduce the dimensionality of the dataset to 2D using Principal Component Analysis (PCA).
   - Visualize the decision boundary of the KNN and Logistic Regression models.

   Task:
   - Plot the decision boundary for both models using the top two principal components.
   - Discuss how each model separates the malignant and benign tumors in the 2D space.

Inability to follow this instruction will be deducted 5 points each for filename format and late submission per day. Also, cheating and plagiarism will be penalized.

Rubric for Exercises on KNN and Logistic Regression

Criteria	Excellent (90-100%)	Good (75-89%)	Satisfactory (60-74%)	Needs Improvement (0-59%)
Exercise 1: Data Exploration and Preprocessing (20%)	Comprehensive data exploration with insights into feature distributions and correlations; data preprocessing thoroughly handled, including appropriate scaling and feature selection.	Adequate exploration of data; preprocessing steps correctly implemented with minor issues; clear feature selection and scaling.	Basic data exploration and preprocessing; some issues with handling missing data, scaling, or feature selection.	Minimal exploration; poor handling of missing values, scaling, or data preprocessing. Little or no meaningful analysis of data.
Exercise 2: KNN Implementation (20%)	KNN model is accurately implemented with proper hyperparameters; model performance is clearly evaluated using multiple metrics (accuracy, confusion matrix, etc.).	KNN model implemented with minor issues; performance metrics evaluated correctly but could include more insights.	KNN model implemented with noticeable errors or lack of proper evaluation; missing or incorrect performance metrics.	Incorrect or incomplete KNN model implementation; evaluation metrics missing or poorly interpreted.
Exercise 3: Logistic Regression Implementation (20%)	Logistic Regression implemented accurately with well-documented code; metrics like accuracy, confusion matrix, and classification report correctly calculated and interpreted.	Logistic Regression implemented with minor issues; metrics calculated and interpreted with some depth but could be more detailed.	Basic implementation of Logistic Regression with some errors or incomplete evaluation; limited interpretation of results.	Poor or incorrect Logistic Regression implementation; missing evaluation metrics or incorrect interpretations of model results.
Exercise 4: Hyperparameter Tuning and Cross-Validation (20%)	Thorough tuning of hyperparameters using GridSearchCV; cross-validation performed correctly; best parameters reported with detailed explanation of results.	Hyperparameter tuning and cross-validation performed with minor issues; the best parameters identified but analysis lacks depth.	Basic hyperparameter tuning and cross-validation; issues with implementation or lack of clarity in parameter selection.	Poor or no hyperparameter tuning; cross-validation not implemented or incorrectly done; no clear parameter selection or results reported.
Exercise 5: Decision Boundary Visualization (10%)	Clear and insightful visualization of decision boundaries using PCA; visualization effectively shows how models separate classes; thoughtful discussion of model boundaries.	Decision boundary visualizations provided but lacking clarity or depth; discussion of results is limited.	Basic visualizations provided but unclear or poorly executed; limited discussion on decision boundaries and class separation.	Minimal or missing visualizations; poor or no discussion of decision boundaries; little effort to explain class separation.
Report Quality and Visualizations (10%)	Report is well-organized, professional, and clearly explains all steps; visualizations (e.g., confusion matrices, accuracy graphs) are integrated and support analysis effectively.	Report is organized with minor issues; visualizations are present but could better support the analysis; some sections lack clarity or depth.	Basic report with limited depth and clarity; visualizations are present but may not fully support the analysis or are unclear.	Report is unclear, disorganized, or incomplete; visualizations are missing, irrelevant, or do not support the analysis; minimal documentation.