SLIIT UNI
THE KNOWLEDGE UNIVERSITY

IT2011 - Artificial Intelligence and Machine Learning

Department of Information Technology, Faculty of Computing

Year 2 semester 1 (2025)

Workshop 02

Supervised Learning Workshop 2

Part 1 – Classification Tree

1. How do you load the diabetes.CSV dataset and display its first five rows? do you separate the dataset into feature variables (X) and the target variable (y)?
2. How do you split the dataset into training and testing sets (80–20 split)?
3. How do you create and train a DecisionTreeClassifier on the training dataset?
4. How do you visualize the trained decision tree?
5. How do you use the trained model to predict on the test set?
6. How do you generate a confusion matrix and classification report to evaluate the model?
7. How do you calculate the model's accuracy score on both training and testing
8. How do you avoid overfitting by adjusting max_depth or min_samples_split?

Part 2 – K Nearest Neighbors (KNN)

1. How do you standardize the dataset before applying KNN? Why is this step important?
2. How do you create and train a KNeighborsClassifier with k=5?
3. How do you predict outcomes on the test dataset using KNN?
4. How do you generate the confusion matrix and classification report for KNN?
5. . How do you calculate the model's accuracy on training and testing sets?
6. How does accuracy vary when you try different values of k? (e.g., k=3,5,7,9)
7. How do you visualize the effect of different k values on accuracy using a plot?

Part 3 – Support Vector Machines (SVM)

1. How do you scale/standardize the data for SVM?
2. How do you create and train a SVC (Support Vector Classifier) with linear kernel?
3. How do you predict outcomes on the test dataset using the trained model?
4. How do you generate the confusion matrix and classification report for SVM?
5. How do you compare accuracy between linear, polynomial, and RBF kernels?

Part 4 – Cross-Validation & Hyperparameter Optimization

1. What is cross-validation and why is it useful compared to a single train-test split?
2. How do you perform k-fold cross-validation (k=5) on Decision Tree, KNN, and SVM models?
3. How do you implement GridSearchCV to tune hyperparameters? Examples:
   - Decision Tree: max_depth, min_samples_split
   - KNN: n_neighbors, weights
   - SVM: kernel, C, gamma
4. How do you retrieve the best hyperparameters and best score from GridSearchCV?
5. How do you evaluate the best model on the test dataset?
6. How do you compare performance of the tuned models (Decision Tree, KNN, SVM)?

Discuss how can these algorithms be used in Regression tasks when a numerical response variable is available.