Major Assignment – 2 (Chapter 3 – 4)

Part A – Theory Questions

Q1.

(a) Describe the concept of the holdout method for model evaluation.
(b) Apply the holdout method to evaluate a classification model on a given dataset and explain each step.

Q2.

(a) Explain why cross-validation provides a better estimate of model performance than the holdout method.
(b) Apply k-fold cross-validation (k = 5) to a dataset of 500 samples and describe how the data is split and used.

Q3.

(a) What is meant by bias–variance trade-off?
(b) Analyze how bias and variance together influence model generalization.

Q4.

(a) Evaluate how the confusion matrix helps in assessing classification performance.
(b) Using the confusion matrix, compute precision, recall, and F1-score for a binary classifier.
True Positives = 50, False Positives = 10, False Negatives = 5, True Negatives = 35

Q5.

(a) A model achieves 95% training accuracy but 75% test accuracy. Identify the issue and explain how to mitigate it.
(b) Assess how bias–variance trade-off influences model selection and performance.

Q6.

(a) A model performs well on training data but poorly on test data. Identify the problem and explain it, then suggest a solution.
(b) Apply R-squared to assess the performance of a regression model and explain its interpretation.

Q7.

(a) Explain why feature engineering is considered a crucial step before model training.

Q8.

(a) What are the main steps involved in the feature selection process?
(b) How would you apply feature selection techniques to reduce a dataset from 100 features to only the 10 most relevant ones?

Q9.

(a) Explain how high-dimensional data affects model computation and accuracy.

(b) Apply the filter method for feature selection using correlation-based ranking and explain each step.

Q10.

(a) Explain why feature engineering is considered a crucial step before model training.
(b) A model trained with 50 features performs worse than one trained with 15. Identify the likely cause and propose how feature engineering can resolve it.

Part B – Lab Questions

Q1. An automobile company wants to predict a car's mpg value from its physical attributes.
Tasks:

(a) Load the dataset auto_mpg.csv and remove missing values.
(b) Identify predictor and target variables.
(c) Perform data splitting (80% train, 20% test).
(d) Fit a Linear Regression model and predict test outcomes.
(e) Evaluate the model using Mean Squared Error and R^2 score.
(f) Discuss: If the R^2 score = 0.85, what does it imply about model performance?

Q2. Exploring random sampling methods to estimate model uncertainty.
Tasks:

(a) From the btissue.csv data, extract only the feature columns (excluding labels).
(b) Using the resample() method, create a bootstrap sample of 100 observations.
(c) Show the first 10 rows of the sample and identify if any rows are repeated.

Q3. Instead of relying on a single train–test split, you want to check how consistent your model is.
Tasks:

(a) Using the btissue.csv dataset, implement 5-fold cross-validation.
(b) For each fold, print the train/test indices and record how many samples are used for training vs testing.
(c) Visualize or summarize how different folds cover the entire dataset without overlap.

Q4. Testing two validation techniques to measure model generalization.
Tasks:

(a) Use the btissue.csv dataset and a Decision Tree Classifier.
(b) Evaluate model performance using:
    i) Holdout (80/20 split)
    ii) 5-Fold Cross-Validation
(c) Compare the accuracy results from both methods.

Q5. Feature Creation from Structured Data

(a) Using a dataset containing columns like Age, Income, and Spending Score, construct new derived features such as Age Group, Income-to-Spending Ratio, and Normalized Spending.
(b) Plot and analyze how the new features correlate with the target variable.

Q6. Load the Iris dataset and select a subset of features manually using the .iloc function. Train a simple Decision Tree Classifier using only the selected subset of features and compare its performance with the model trained using all features.
Tasks:

(a) Load the Iris dataset from sklearn.datasets.
(b) Create a DataFrame and display the first few rows.
(c) Train a Decision Tree Classifier using all features and record the accuracy.
(d) Select a subset of columns (for example, the first two features: sepal length and sepal width) using .iloc.
(e) Train another model using only the selected features and evaluate its accuracy.
(f) Compare and discuss the results of both models.

Q7. Load the Iris dataset and apply Principal Component Analysis (PCA) to reduce its four numerical features (sepal length, sepal width, petal length, petal width) into two principal components. Visualize the transformed data in a 2D scatter plot to observe how the classes (Setosa, Versicolor, Virginica) are separated in the reduced feature space. Additionally, display the explained variance ratio for each component.
Tasks:

(a) Load the Iris dataset using sklearn.datasets.
(b) Perform PCA to reduce the dataset to two components.
(c) Create a new DataFrame containing the two principal components and target labels.
(d) Plot the two components using a scatter plot with different colors for each class.

Q8. Create a dataset containing employee information, including Department, Job Role, and Marital Status. Convert all categorical columns into numeric form so that the dataset can be used effectively for training machine learning models. Use appropriate encoding techniques such as Label Encoding and One-Hot Encoding.
Tasks:

(a) Create a DataFrame with the following columns and sample data:
    Department (e.g., HR, IT, Finance)
    Job_Role (e.g., Manager, Analyst, Clerk)
    Marital_Status (e.g., Single, Married, Divorced)
(b) Display the original dataset.
(c) Encode categorical columns using:
    Label Encoding for ordered or binary categories.
    One-Hot Encoding for nominal categories.