Machine Learnig

Practice Problems

Python Numpy: Practice Problems

1. Create a NumPy array containing numbers 0–9.
2. Generate a 3x3 matrix with values ranging from 2 to 10.
3. Create an array of 10 zeros and set the fifth value to 1.
4. Reverse a NumPy array without using loops.
5. Create a 3x3 identity matrix.
6. Generate an array of 9 random integers between 1 and 100, and reshape it to 3x3.
7. Find the maximum, minimum, and mean of a given array.
8. Replace all odd numbers in an array with -1.
9. Create a 5x5 matrix with values from 1 to 25, and extract the middle 3x3 submatrix.
10. Normalize a random 1D array (subtract mean, divide by std).
11. Stack two 2x2 matrices vertically and horizontally.
12. Create an array with random values and sort it in descending order.
13. Given two arrays, compute their element-wise product and dot product.
14. Use Boolean indexing to select all even numbers from an array.
15. Create a 10x10 matrix and fill its border with 1s and the inside with 0s.
16. Compute the Euclidean distance between two random 3D points using NumPy only.
17. Generate a random 5x5 matrix and compute its determinant and inverse.
18. Replace all NaN values in an array with the column mean.
19. Given a matrix, find its eigenvalues and eigenvectors.
20. Simulate rolling two dice 1000 times and compute the frequency of each sum (2–12).
21. Save a NumPy array to disk and reload it from a .npy file.

Python Pandas: Practice Problems

1. Create a DataFrame of 5 students with columns: `Name`, `Age`, `Marks`. Find the student with the highest marks.
2. Load a CSV file of your choice. Display its first 10 rows and summary statistics.
3. Add a new column `Pass/Fail` where students with marks ≥ 40 pass, otherwise fail.
4. Group the data by a categorical column and compute the average of numeric columns.
5. Merge two DataFrames: one containing `Employee IDs and Names`, another containing `Employee IDs and Salaries`.
6. Sort the DataFrame by multiple columns: first by `Department`, then by `Salary` (descending).
7. Replace all missing values in a DataFrame with the column mean.
8. Create a new column that shows the rank of employees by salary within their department.
9. Use `apply()` to increase every salary by 15%.
10. Convert a date column into day, month, and year components.

Python Matplotlib: Practice Problems

1. Plot $\sin(x)$ and $\cos(x)$ on the same graph for $x \in [0, 2\pi]$ with legends, gridlines, and different colors.
2. Create a bar chart of 5 subjects (Math, English, Science, History, Art) with random student scores. Label axes properly.
3. Generate a scatter plot of random (x, y) points colored by their values using a colormap.
4. Plot a histogram of 1000 random normal values with 40 bins. Change the color and add a grid.
5. Create 4 subplots showing: (1) $\sin(x)$, (2) $\cos(x)$, (3) $\tan(x)$, (4) $\sin(x) + \cos(x)$.
6. Plot a pie chart showing the percentage of time you spend on 4 daily activities (Sleep, Study, Exercise, Leisure).
7. Create a box plot comparing 3 datasets of random values with different variances.
8. Plot a 3D surface for $z = \sin(\sqrt{x^2 + y^2})$ using `Axes3D`.
9. Save any plot you create as a PNG image at 300 DPI.
10. Use `subplots()` to create a figure with two plots: one bar chart and one scatter plot side by side.

Python Scikit-learn: Practice Problems

Instructions

- Each assignment requires you to load data from an Excel file using `pandas.read_excel`.
- Perform train-validation split or cross-validation using `scikit-learn`.
- Visualize relevant results using `matplotlib` or `seaborn`.
- Implement the specified algorithm and interpret your findings.
- The possible source of datasets will be UCI repository or Kaggle.

1. Linear Regression
   (a) Load `data.xlsx` with features $X_1, X_2, \ldots$ and target y.
   (b) Display summary statistics.
   (c) Split into 80% training and 20% validation.
   (d) Fit `LinearRegression`.
   (e) Plot predictions vs. ground truth.
   (f) Report Mean Squared Error on the validation set.
2. Logistic Regression
   (a) Load `classification_data.xlsx`, $y \in \{0, 1\}$.
   (b) Check for missing values and preprocess.
   (c) Standardize features.
   (d) Perform stratified 5-fold cross-validation.
   (e) Fit `LogisticRegression`, report accuracy, precision, recall.
   (f) Plot cross-validation performance.
3. K-Nearest Neighbors Classification
   (a) Load `knn_data.xlsx`, separate X and y.
   (b) Split into 70% train, 30% test.
   (c) Train `KNeighborsClassifier`, try $k = 1, 3, 5, 7$.
   (d) Plot confusion matrix and accuracy for each k.
   (e) Interpret the best k.

4. Naive Bayes Classification
   (a) Load `nb_data.xlsx`, encode categorical features.
   (b) Perform 10-fold cross-validation with `GaussianNB` or `MultinomialNB`.
   (c) Plot ROC curve (binary) or summary metrics (multiclass).

5. Decision Trees for Classification
   (a) Load `dt_data.xlsx` and preprocess.
   (b) Train `DecisionTreeClassifier`.
   (c) Visualize the tree using `plot_tree`.
   (d) Plot and interpret feature importances.

6. Random Forests for Feature Importance
   (a) Load `rf_data.xlsx`.
   (b) Train `RandomForestClassifier` ($n = 100$).
   (c) Evaluate with out-of-bag score or cross-validation.
   (d) Rank features by importance and plot the top 10.

7. Feature Selection: Wrapper Method
   (a) Load `feature_data.xlsx`.
   (b) Apply forward selection.
   (c) Apply backward elimination.
   (d) Compare selected features and validation performance based on logistic classifier.

8. Feature Selection: Filter Method
   (a) Load `feature_data.xlsx`.
   (b) Apply F-Score based feature selection.
   (c) Apply correlation based feature selection.
   (d) Compare selected features and validation performance based on logistic classifier.

9. Clustering: K-Means, GMM, Hierarchical
   (a) Load `cluster_data.xlsx`, visualize raw data (scatter plot/pairplot).
   (b) Standardize features.
   (c) Apply `KMeans`, select k via elbow method.
   (d) Apply `GaussianMixture`, compare with KMeans.
   (e) Apply `AgglomerativeClustering`, plot dendrogram.
   (f) Visualize and compare cluster assignments (silhouette score).