|  |  |  |
| --- | --- | --- |
| **MACHINE LEARNING LAB** | | |
| **Course Code** | | **22CDL66** |
| **LIST OF LABORATORY PROGRAMS** | | |
| 1. | Develop a program to create histograms for all numerical features and analyze the distribution of  each feature.  Generate box plots for all numerical features and identify any outliers. Use California Housing dataset.  **Book 1: Chapter 2** | |
| 2. | Develop a program to Compute the correlation matrix to understand the relationships between pairs of features. Visualize the correlation matrix using a heatmap to know which variables have strong positive/negative correlations. Create a pair plot to visualize pairwise relationships between features. Use California Housing dataset.  **Book 1: Chapter 2** | |
| 3. | Develop a program to implement Principal Component Analysis (PCA) for reducing the  dimensionality of the Iris dataset from 4 features to 2.  **Book 1: Chapter 2** | |
| 4. | For a given set of training data examples stored in a .CSV file, implement and demonstrate the  Find-S algorithm to output a description of the set of all hypotheses consistent with the training examples.  **Book 1: Chapter 3** | |
| 5. | Develop a program to implement k-Nearest Neighbour algorithm to classify the randomly generated  100 values of *x* in the range of [0,1]. Perform the following based on dataset generated.  1. Label the first 50 points {*x1*,……,*x50*} as follows: if (xi ≤ 0.5), then *xi* ε Class1, else *xi* ε Class1  2. Classify the remaining points, *x51*,……,*x100* using KNN. Perform this for *k=1,2,3,4,5,20,30*  **Book 2: Chapter – 2** | |
| 6. | Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points.  Select appropriate data set for your experiment and draw graphs.  **Book 1: Chapter – 4** | |
| 7. | Develop a program to demonstrate the working of Linear Regression and Polynomial Regression.  Use Boston Housing Dataset for Linear Regression and Auto MPG Dataset (for vehicle fuel efficiency prediction) for Polynomial Regression.  **Book 1: Chapter – 5** | |
| 8. | Develop a program to demonstrate the working of the decision tree algorithm. Use Breast Cancer  Data set for building the decision tree and apply this knowledge to classify a new sample.  **Book 2: Chapter – 3** | |
| 9. | Develop a program to implement the Naive Bayesian classifier considering Olivetti Face Data set  for training. Compute the accuracy of the classifier, considering a few test data sets.  **Book 2: Chapter – 4** | |
| 10. | Develop a program to implement k-means clustering using Wisconsin Breast Cancer data set and  visualize the clustering result. | |

|  |  |
| --- | --- |
|  | **Book 2: Chapter – 4** |
| **Text Books:**  1. S Sridhar and M Vijayalakshmi, ―Machine Learning‖, Oxford University Press, 2021.  2. M N Murty and Ananthanarayana V S, ―Machine Learning: Theory and Practice‖, Universities Press  (India) Pvt. Limited, 2024. | |
|
| **Web links and Video Lectures (e-Resources):**  1. https:/[/www.drssridhar.com](http://www.drssridhar.com/)/?page\_id=1053  2. https:/[/www.universitiespress.com/resources](http://www.universitiespress.com/resources)?id=9789393330697  3. https://onlinecourses.nptel.ac.in/noc23\_cs18/preview | |
|
|
|
|
|
|
|
|
|

**1. Develop a program to create histograms for all numerical features and analyze the distribution of each feature. Generate box plots for all numerical features and identify any outliers. Use California Housing dataset.**

**import pandas as pd**

**import numpy as np**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**from sklearn.datasets import fetch\_california\_housing**

**# Step 1: Load the California Housing dataset**

**data = fetch\_california\_housing(as\_frame=True)**

**housing\_df = data.frame**

**# Step 2: Create histograms for numerical features**

**numerical\_features = housing\_df.select\_dtypes(include=[np.number]).columns**

**# Plot histograms**

**plt.figure(figsize=(15, 10))**

**for i, feature in enumerate(numerical\_features):**

**plt.subplot(3, 3, i + 1)**

**sns.histplot(housing\_df[feature], kde=True, bins=30, color='blue')**

**plt.title(f'Distribution of {feature}')**

**plt.tight\_layout()**

**plt.show()**

**# Step 3: Generate box plots for numerical features**

**plt.figure(figsize=(15, 10))**

**for i, feature in enumerate(numerical\_features):**

**plt.subplot(3, 3, i + 1)**

**sns.boxplot(x=housing\_df[feature], color='orange')**

**plt.title(f'Box Plot of {feature}')**

**plt.tight\_layout()**

**plt.show()**

**# Step 4: Identify outliers using the IQR method**

**print("Outliers Detection:")**

**outliers\_summary = {}**

**for feature in numerical\_features:**

**Q1 = housing\_df[feature].quantile(0.25)**

**Q3 = housing\_df[feature].quantile(0.75)**

**IQR = Q3 - Q1**

**lower\_bound = Q1 - 1.5 \* IQR**

**upper\_bound = Q3 + 1.5 \* IQR**

**outliers = housing\_df[(housing\_df[feature] < lower\_bound) | (housing\_df[feature] > upper\_bound)]**

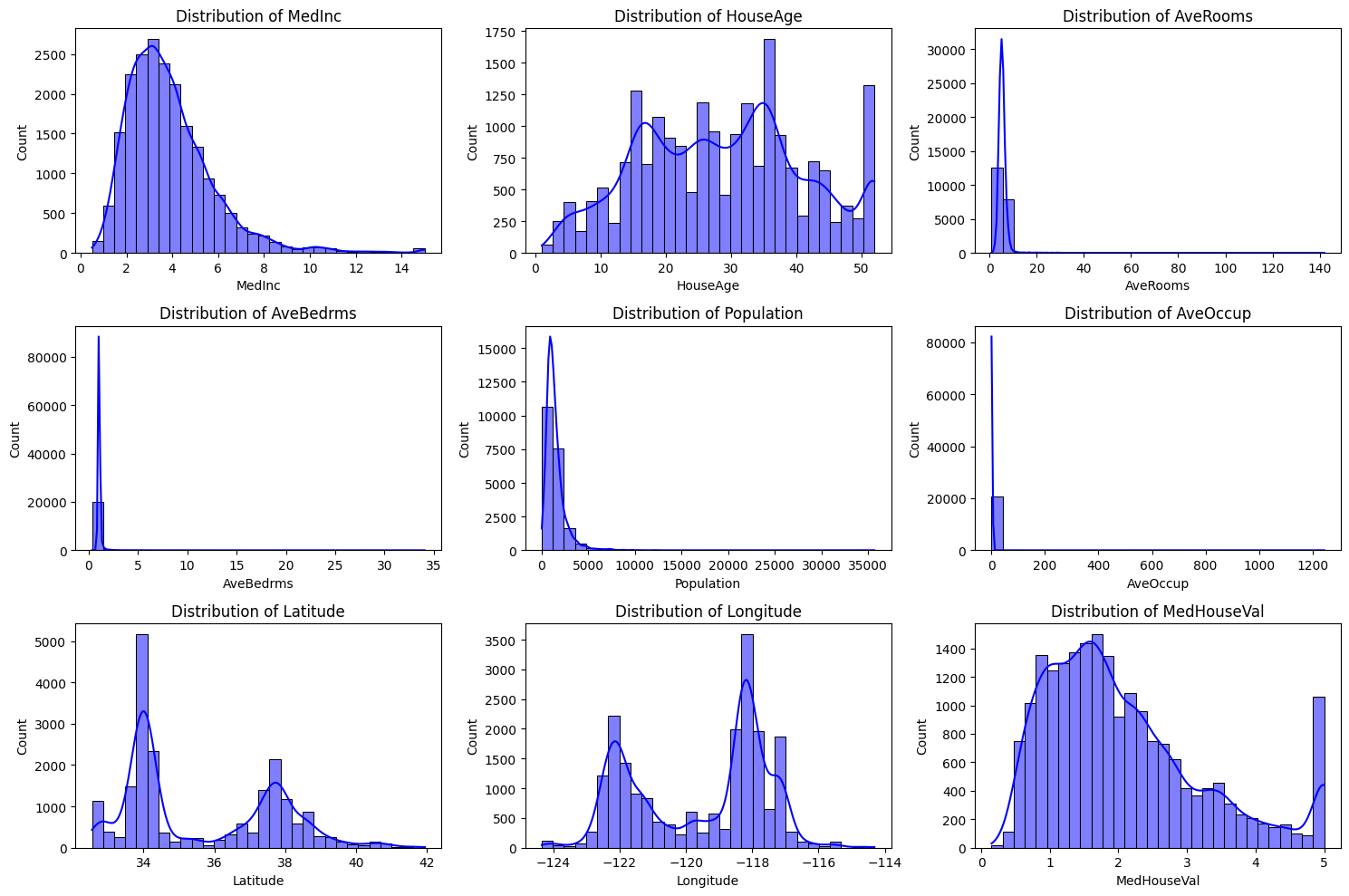
**outliers\_summary[feature] = len(outliers)**

**print(f"{feature}: {len(outliers)} outliers")**

**# Optional: Print a summary of the dataset**

**print("\nDataset Summary:")**

**print(housing\_df.describe())**

****

Outliers Detection:

MedInc: 681 outliers

HouseAge: 0 outliers

AveRooms: 511 outliers

AveBedrms: 1424 outliers

Population: 1196 outliers

AveOccup: 711 outliers

Latitude: 0 outliers

Longitude: 0 outliers

MedHouseVal: 1071 outliers

Dataset Summary:

MedInc HouseAge AveRooms AveBedrms Population \

count 20640.000000 20640.000000 20640.000000 20640.000000 20640.000000

mean 3.870671 28.639486 5.429000 1.096675 1425.476744

std 1.899822 12.585558 2.474173 0.473911 1132.462122

min 0.499900 1.000000 0.846154 0.333333 3.000000

25% 2.563400 18.000000 4.440716 1.006079 787.000000

50% 3.534800 29.000000 5.229129 1.048780 1166.000000

75% 4.743250 37.000000 6.052381 1.099526 1725.000000

max 15.000100 52.000000 141.909091 34.066667 35682.000000

AveOccup Latitude Longitude MedHouseVal

count 20640.000000 20640.000000 20640.000000 20640.000000

mean 3.070655 35.631861 -119.569704 2.068558

std 10.386050 2.135952 2.003532 1.153956

min 0.692308 32.540000 -124.350000 0.149990

25% 2.429741 33.930000 -121.800000 1.196000

50% 2.818116 34.260000 -118.490000 1.797000

75% 3.282261 37.710000 -118.010000 2.647250

max 1243.333333 41.950000 -114.310000 5.000010

**2. Develop a program to Compute the correlation matrix to understand the relationships between pairs of features. Visualize the correlation matrix using a heatmap to know which variables have strong positive/negative correlations. Create a pair plot to visualize pairwise relationships between features. Use California Housing dataset.**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.datasets import fetch\_california\_housing

# Step 1: Load the California Housing Dataset

california\_data = fetch\_california\_housing(as\_frame=True)

data = california\_data.frame

# Step 2: Compute the correlation matrix

correlation\_matrix = data.corr()

# Step 3: Visualize the correlation matrix using a heatmap

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.title('Correlation Matrix of California Housing Features')

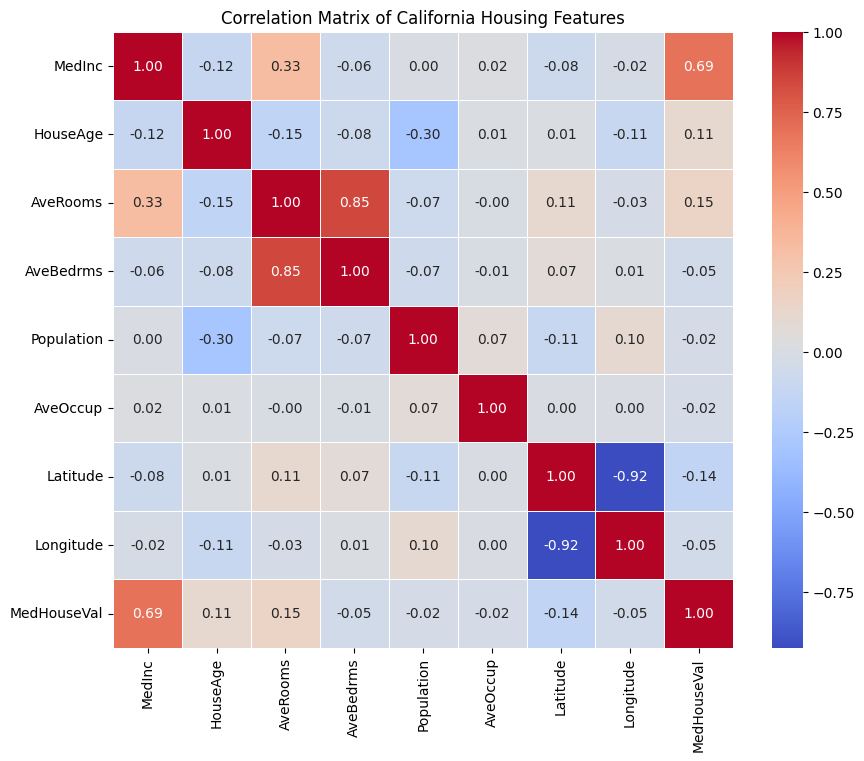
plt.show()

# Step 4: Create a pair plot to visualize pairwise relationships

sns.pairplot(data, diag\_kind='kde', plot\_kws={'alpha': 0.5})

plt.suptitle('Pair Plot of California Housing Features', y=1.02)

plt.show()

****