(23A31403) AI & ML Lab

Lab Internal And External

1. Pandas Library
a) Write a python program to implement Pandas Series with labels.
b) Create a Pandas Series from a dictionary.
c) Creating a Pandas Data Frame.
d) Write a program
which makes use of the following Pandas methods
i) describe ()
ii) head ()
iii) tail ()
iv) info ()
Sub Write) .UNIT-I (Evolution of Machine Learning) from TextBook

Data Set or Data Input:

data = {
"Name" : ["A SADWIK" , "E SOWMYA" , "G SUBBAREDDY", "M SUNAYANA" , " S ASIF " , " S SHAMEERA ", "U TULASI"],
"Roll_Number" : [" 24HN1A3901" , "24HN1A3911" , " 24HN1A3921", "24HN1A3931" , "24HN1A3941" , " 24HN1A3951" , "24HN1A3961"],
"ML_subject_code" :[" 23A31401T", " 23A31401T", " 23A31401T", " 23A31401T", " 23A31401T", " 23A31401T", " 23A31401T"],
"AI_ML_Lab_code" : [" 23A31403", " 23A31403", " 23A31403", " 23A31403", " 23A31403", " 23A31403", " 23A31403"]}

2. Pandas Library: Visualization a) Write a program which use pandas inbuilt visualization to plot following graphs:
i. Bar plots
ii. Histograms
iii. Line plots
iv. Scatter plots
Sub Write). UNIT-I (Paradigms for ML) from TextBook
Data Set or Data Input:
food_data = {
" ingredients " : ["apple","bacon","banana","carrot","cheese"],
"**sweetness**": [10, 1, 10, 7, 1],
"**crunchiness**":[9, 4, 1,10, 1],
"**cat_type**":["fruit","protein","fruit","vegetable","protein"]}

3. Write a Program to Implement Breadth First Search using Python.

Sub Write). UNIT-I (Learning by Deduction and Learning by Abduction) from TextBook

Data Set or Data Input:

[('Atmakur', 'Marripadu', 43), ('Atmakur', 'Sangam', 17), ('Marripadu', 'Appiligunta', 10), ('Marripadu', 'Badvel', 51), ('Marripadu', 'Pallolu', 5), ('Sangam', 'Padalakure', 20), ('Sangam', 'Nellore', 60)]

5. Write a Program to Implement Depth First Search using Python.

Sub Write). UNIT-I (Learning by Deduction and Learning by Abduction) from TextBook

Data Set or Data Input:

[('Atmakur', 'Marripadu', 43), ('Atmakur', 'Sangam', 17), ('Marripadu', 'Appiligunta', 10), ('Marripadu', 'Badvel', 51), ('Marripadu', 'Pallolu', 5), ('Sangam', 'Padalakure', 20), ('Sangam', 'Nellore', 60)]

8. Apply the following Pre-processing techniques for a given dataset.

a. Attribute selection

b. Handling Missing Values

c. Discretization

d. Elimination of Outliers

Sub Write). UNIT-I (Feature Engineering) from TextBook

Data Set or Data Input:

```
student_data = {
   'roll_number':['901','911','921','931','941','951','961'],
   "Name" : ["A SADWIK" , "E SOWMYA" , "G SUBBAREDDY",
         "M SUNAYANA" , "S ASIF" , "S SHAMEERA", "U TULASI"],
   'Passed':[6,6,6,6,None,6,None],
   'Failed':[None,0,0,0,0,None,None],
   'Total':[6,6,6,6,6,6,6],
   'Branch':['AIML','AIML','AIML','AIML','AIML','AIML','AIML'],
   'Age':[22,20,21,20,23,19,None]
}
```

9. Apply KNN algorithm for classification and regression

Sub Write). UNIT-II (K-Nearest Neighbor Classifier, Radius Distance Nearest Neighbor Algorithm, KNN Regression) from PPT slide information

Data Set or Data Input:

logic

```
food_data = {
"ingredients": ["apple","bacon","banana","carrot","cheese"],
"sweetness": [10, 1, 10, 7, 1],
"crunchiness":[9, 4, 1, 10, 1],
"cat_type":["fruit","protein","fruit","vegetable","protein"]
}

# Features (sweetness, crunchiness)
X = np.array(list(zip(food_data["sweetness"], food_data["crunchiness"])))

# Labels
y = np.array(food_data["cat_type"])
```

Ans ?

{ "ingredients": "tomato", "sweetness": 6, "crunchiness": 4, "type": "???" }

logic

```
tomato = np.array([[6, 4]])   # sweetness=6, crunchiness=4
prediction = classifier.predict(tomato)

print("Tomato is:", prediction[0])
```

10. Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results

Sub Write). Unit-III(Decision Trees for Classification & Regression) from PPT slide information

Data Set or Data Input:

```
data = {
   'outlook': ['sunny','sunny','overcast','rainy','rainy','rainy',
               'overcast','sunny','sunny','rainy','sunny','overcast',
               'overcast','rainy'],
      'temperature': ['hot','hot','hot','mild','cold','cold',
                  'cold','mild','cold','mild','mild','mild',
                  'hot','mild'],
      'humidity': ['high','high','high','high','normal','normal',
                   'normal','high','normal','normal','normal','high',
                   'normal','high'],
      'wind': ['false','true','false','false','false','true',
               'true','false','false','false','true','true',
               'false','true'],

   'play': [0,0,1,1,1,0,1,0,1,1,1,1,1,0]   # 0 = No, 1 = Yes
}

df = pd.DataFrame(data)
# ----------------------------
# Step 2: Encode Categorical Data
# ----------------------------
le = LabelEncoder()
for column in df.columns:
   df[column] = le.fit_transform(df[column])
X = df.drop('play', axis=1)
y = df['play']
```

Ans ?

```
Outlook:
          overcast = 0,rainy    = 1,sunny    = 2
Temperature:
          cold = 0,hot  = 1,mild = 2
Humidity:
          high   = 0,normal = 1
Wind:
          false = 0,true  = 1
# Example: sunny, cool, high, true
sample = [[2, 1, 0, 1]]   # Encoded values (depends on encoding)
prediction = classifier.predict(sample)
```

11. Apply Random Forest algorithm for classification and regression

Sub Write). Unit-III(Bias–Variance Trade-off, Random Forests for Classification and Regression) from PPT slide information

Data Set or Data Input:

```
# ----------------------------
# Step 1: Create Play Tennis Dataset
# ----------------------------
data = {
'outlook': ['sunny','sunny','overcast','rainy','rainy','rainy',
'overcast','sunny','sunny','rainy','sunny','overcast',
'overcast','rainy'],
'temperature': ['hot','hot','hot','mild','cold','cold',
'cold','mild','cold','mild','mild','mild',
'hot','mild'],
'humidity': ['high','high','high','high','normal','normal',
'normal','high','normal','normal','normal','high',
'normal','high'],
'wind': ['false','true','false','false','false','true',
'true','false','false','false','true','true',
'false','true'],
'play': [0,0,1,1,1,0,1,0,1,1,1,1,1,0] # 0 = No, 1 = Yes
}
df = pd.DataFrame(data)
# ----------------------------
# Step 2: Encode Categorical Data
# ----------------------------
le = LabelEncoder()
for column in df.columns:
df[column] = le.fit_transform(df[column])
X = df.drop('play', axis=1)
y = df['play']
```

Ans ?

Outlook:

overcast = 0,rainy = 1,sunny = 2

Temperature:

cold = 0,hot = 1,mild = 2

Humidity:

high = 0,normal = 1

Wind:

false = 0,true = 1

```
# ----------------------------
# Step 6: Predict New Sample
# Example: sunny, mild, high, true
# ----------------------------
sample = [[2, 2, 0, 1]] # Encoded values
pred = model.predict(sample)
```

12. Demonstrate Naïve Bayes Classification algorithm.

Sub Write). Unit-III(Naive Bayes Classifier) from PPT slide information

Data Set or Data Input:

```
# --------------------------------
# [Study Hours, CGPA]
X = np.array([[1, 5.5],[2, 6.0],[2.5, 6.5],[1.5, 5.8],[2, 6.2],[1, 5.0], # Not Placed (0)
[4, 8.0],[5, 8.5],[3.5, 7.5],[4.5, 8.2],[5, 9.0],[3.8, 7.8] # Placed (1)
])

# 0 = Not Placed, 1 = Placed
y = np.array([0]*6 + [1]*6)
```

Ans?

```
test_point = np.array([[3, 7]])
pred = model.predict(test_point)[0]

if pred == 0:
print("Prediction: Not Placed")
else:
print("Prediction: Placed")
```

13. Apply Support Vector algorithm for classification

Sub Write). Unit-IV(Support Vector Machines, Linearly Non-Separable Case, Non-linear SVM, Kernel Trick) from PPT slide information

Data Set or Data Input:

```
from sklearn import svm, datasets

# Load digits dataset
digits = datasets.load_digits()
X = digits.data
y = digits.target

classifier = svm.SVC(gamma=0.001, C=10)
classifier.fit(X, y)
```

Ans ?

```
# Option 1: Take the first sample of digit 0 from the dataset
manual_0 = X[y == 0][0] # 64 features of the first 0
manual_0 = manual_0.reshape(1, -1)
# Predict
pred_label = classifier.predict(manual_0)[0]
```