IT3071 – Machine Learning and Optimization Methods

Lab Sheet 01

NumPy

1. What is a NumPy array and how is it different from a Python list?
A NumPy array is a powerful n-dimensional array object provided by NumPy. It is faster, memory-efficient, and supports vectorized operations unlike Python lists.

```
import numpy as np

arr = np.array([1, 2, 3])
```

2. How do you create and access elements in a 2D NumPy array?
Use a list of lists to create. Access elements with row, column indices.

```
arr2d = np.array([[1, 2, 3], [4, 5, 6]])

print(arr2d[0, 1])  # Output: 2
```

3. What are common NumPy operations?
Element-wise addition, multiplication, power, scalar ops.

```
a = np.array([1, 2, 3])

b = np.array([4, 5, 6])

print(a + b)

print(a * b)

print(a ** 2)
```

4. What are some useful NumPy functions for creating arrays?

```
np.zeros((2, 3))

np.ones(4)

np.arange(1, 10, 2)

np.linspace(0, 1, 5)

np.eye(3)
```

5. How do you perform aggregate/statistical functions?

```
arr = np.array([[1, 2], [3, 4]])
```

```
np.sum(arr)
np.mean(arr)
np.max(arr)
np.min(arr)
np.sum(arr, axis=0)
np.sum(arr, axis=1)
```

Pandas

1. How do you create a DataFrame from a dictionary?

```
import pandas as pd
data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}
df = pd.DataFrame(data)
```

2. How can you access rows and columns in a DataFrame?

```
df['Name']
df[['Name', 'Age']]
df.loc[1]
df.iloc[0]
df.loc[1, 'Age']
```

3. How do you read a CSV file?

```
df = pd.read_csv('data.csv')
df.head()
df.info()
df.to_csv('output.csv', index=False)
```

4. Common DataFrame operations:

```
df[df['Age'] > 25]
df.sort_values(by='Age')
```

```
df['NewCol'] = [1, 2]

df.drop('Age', axis=1)

df.rename(columns={'Name': 'FullName'})
```

5. How to do aggregation/statistics?

```
df.describe()

df['Age'].mean()

df['City'].value_counts()

df.groupby('City').mean()
```

Matplotlib

1. Plot a simple line chart:

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]

y = [2, 4, 1, 8, 7]

plt.plot(x, y)

plt.title("Simple Line Chart")

plt.xlabel("X Axis")

plt.ylabel("Y Axis")

plt.grid(True)

plt.show()
```

2. Scatter plot with markers and color:

```
x = [5, 7, 8, 7, 2, 17, 2, 9]

y = [99, 86, 87, 88, 100, 86, 103, 87]

plt.scatter(x, y, color='red', marker='o')

plt.title("Scatter Plot Example")

plt.xlabel("X")

plt.ylabel("Y")
plt.show()
```

3. Plotting from a DataFrame:

```
data = {'Month': ['Jan', 'Feb', 'Mar', 'Apr'], 'Sales': [220, 270, 300, 250]}
df = pd.DataFrame(data)
plt.plot(df['Month'], df['Sales'], marker='o')
plt.title("Monthly Sales")
plt.xlabel("Month")
plt.ylabel("Sales")
plt.show()
```

4. Add titles, labels, legends, styles:

```
plt.style.use('seaborn')
plt.plot(x, y, color='green', linestyle='--', marker='o', label='Square')
plt.title("Square Numbers", fontsize=14, color='blue')
plt.xlabel("Input Value", fontsize=12)
plt.ylabel("Output Value", fontsize=12)
plt.grid(True, linestyle=':', color='gray')
plt.legend()
plt.xticks([0, 1, 2, 3, 4])
plt.yticks([0, 5, 10, 15])
plt.show()
```

5. Multiple lines on one chart:

```
x = [1, 2, 3, 4]
y1 = [1, 4, 9, 16]
y2 = [1, 2, 3, 4]
plt.plot(x, y1, label='Square')
plt.plot(x, y2, label='Linear')

plt.title("Multiple Lines")

plt.xlabel("X")

plt.ylabel("Y")

plt.legend()

plt.grid(True)

plt.show()
```

TensorFlow

1. What is TensorFlow?
An open-source library by Google for building ML/DL models.

2. What is a tensor?
A multi-dimensional array: scalar (0D), vector (1D), matrix (2D), etc.

```
import tensorflow as tf

tensor = tf.constant([[1, 2], [3, 4]])
```

3. How do you create tensors in TensorFlow?

```
tf.constant([1, 2, 3])

tf.zeros((2, 2))

tf.ones((3,))

tf.random.uniform((2, 3))
```

4. How do you check the shape and data type of a tensor?

```
t = tf.constant([[1.0, 2.0], [3.0, 4.0]])

t.shape

t.dtype
```

5. What is the difference between tf.constant and tf.Variable?

```
c = tf.constant(5)
```

```
v = tf.Variable(5)
v.assign(10)  # Updates variable value
```

6. How do you perform basic operations on tensors?

```
a = tf.constant([1, 2])
b = tf.constant([3, 4])
tf.add(a, b)
tf.multiply(a, b)
tf.reduce_sum(b)
```

7. What is eager execution?
TensorFlow 2.x runs operations immediately without building a graph first.

```
x = tf.constant([1, 2])
y = tf.constant([3, 4])
print(x + y)
```

8. How do you convert a NumPy array to a tensor?

```
import numpy as np
arr = np.array([1, 2, 3])
tensor = tf.convert_to_tensor(arr)
```

9. What does tf.reshape() do?

```
x = tf.constant([[1, 2], [3, 4]])
reshaped = tf.reshape(x, [4])
```

10. What is the purpose of @tf.function?

```
@tf.function
def add(x, y):
    return x + y
```

It converts a Python function into a high-performance graph for optimization.

Scikit-learn

1. What is Scikit-learn?

A Python library for machine learning — used for classification, regression, clustering, etc.

2. How do you load a sample dataset?

```
from sklearn.datasets import load_iris
data = load_iris()
data.data  # Features
data.target  # Labels
```

3. How do you split the dataset into training and testing sets?

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    data.data, data.target, test_size=0.2, random_state=42)
```

4. How do you train a simple model using Scikit-learn?

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
```

5. How do you make predictions and evaluate the model?

```
y_pred = model.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```