Practical-2

AIM: Introduction to reproducible Machine Learning Operations.

The aim of the practical is to get the hands-on experience of reproducing the machine learning operations at each stage.

1. Ensure that the NumPy, scikit learn, and matplotlib libraries are available in your system. Create the requirements.txt file and make a note of the versions of these libraries.

```
import numpy as np
print(np.__version__)
```

1.22.4

```
import pandas as pd
print(pd.__version__)
```
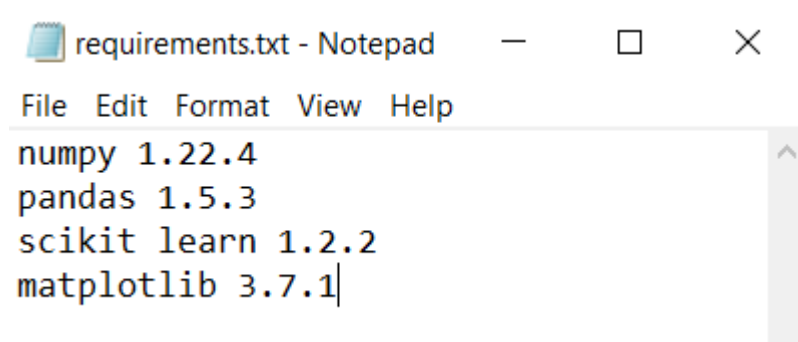
1.5.3

```
import sklearn
print(sklearn.__version__)
```

1.2.2

```
import matplotlib
print(matplotlib.__version__)
```

3.7.1

requirements.txt - Notepad   —   □   ✕

File Edit Format View Help

```
numpy 1.22.4
pandas 1.5.3
scikit learn 1.2.2
matplotlib 3.7.1
```

2. Write a python code to import the Sample.txt data. Further, apply the following processes on the imported data.

a) Import the data and scale it using StandardScaler:

```
 import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import pickle

data = np.loadtxt('Sample.txt')

scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
pickle.dump(scaler, open('scaler_object.pkl', 'wb'))
```

b) Splitting the normalized data:

```
train_data, test_data = train_test_split(scaled_data, test_size=0.2, random_state=42)
```

c) Storing the snapshot of the data as a numpy file:

```
np.save('train_data.npy', train_data)
np.save('test_data.npy', test_data)
```

| test_data.npy | 28-Jul-23 11:56 AM | NPY File | 1 KB |
| train_data.npy | 28-Jul-23 11:56 AM | NPY File | 1 KB |

Step 3 :

Apply the linear regression algorithm on the dataset and assess the prediction on the test dataset.

a) Store the trained model into the local file system to ensure the reproducibility of the prediction. Import the model and the test dataset into other python file. Check whether the same prediction is obtained in the latter case.

```
X_train = train_data[:, :-1]
y_train = train_data[:, -1]

model = LinearRegression()
model.fit(X_train, y_train)

X_test = test_data[:, :-1]
y_test = test_data[:, -1]

predictions = model.predict(X_test)

with open('linear_regression_model.pkl', 'rb') as model_file:
```

```
  loaded_model = pickle.load(model_file)

loaded_predictions = loaded_model.predict(X_test)

if np.array_equal(predictions, loaded_predictions):
  print("Predictions match!")
else:
  print("Predictions differ!")
```

Output :

```
Predictions match!
```