Department of Computer Engineering
University of Peradeniya
CO544: Machine Learning and Data Mining

Lab 1: Python Data Science Toolbox

29th of April 2025

---

1 Setup & Environment

1. Create and activate a Python virtual environment:

```
python3 -m venv venv
source venv/bin/activate    % macOS/Linux
venv\Scripts\activate       % Windows PowerShell
```

2. Install required libraries:

```
pip install numpy matplotlib pandas scikit-learn tabula-py requests
```

3. Launch Jupyter Notebook:

```
jupyter notebook
```

2 Objectives

By the end of this lab, you will learn to:

- Perform advanced array manipulations with **NumPy**.
- Create multi-panel plots with **Matplotlib**.
- Clean and preprocess data with **Pandas**.
- Fetch and normalize JSON data from a public API.
- Use group-by and pivot tables in Pandas.
- Load open datasets from UCI Repository.
- Build and evaluate a basic classification model on the Iris dataset.

3 Exercise 1: NumPy Advanced Operations

1. Generate an array of 20 random integers between 0 and 100:

```
import numpy as np
arr = np.random.randint(0, 101, size=20)
print("Original array:", arr)
```

2. Filter values ≥ 50 using boolean indexing:

```
mask = arr >= 50
filtered = arr[mask]
print("Values >= 50:", filtered)
```

3. Demonstrate broadcasting:

```
small = np.arange(5)
large = arr[:20].reshape(4,5)
result = large + small
print(large, small, result)
```

4. Compute dot product of two arrays of length 10:

```
a = np.arange(10)
b = np.linspace(0, 9, 10)
dp = np.dot(a, b)
print("Dot product:", dp)
```

4 Exercise 2: Matplotlib Subplots

1. Prepare data for sine and cosine functions:

```
import numpy as np
x = np.linspace(0, 2*np.pi, 200)
y1 = np.sin(x)
y2 = np.cos(x)
```

2. Create subplots:

```
import matplotlib.pyplot as plt
fig, axes = plt.subplots(1,2,sharex=True,figsize=(10,4))
axes[0].plot(x, y1)
axes[0].set( title ="Sine Wave", xlabel="x", ylabel="sin(x)")
axes[1].plot(x, y2)
axes[1].set( title ="Cosine Wave", xlabel="x", ylabel="cos(x)")
fig.suptitle("Sine and Cosine Functions")
plt.savefig('trig_functions.png')
plt.show()
```

5 Exercise 3: Pandas Cleaning & Preprocessing

1. Load Titanic dataset:

```
import pandas as pd
url = 'https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv'
df = pd.read_csv(url)
print(df.info())
```

2. Impute missing values:

```
df['Age'].fillna(df['Age'].median(), inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

3. Drop duplicates:

```
df.drop_duplicates(inplace=True)
```

4. Convert and detect outliers in Fare:

```
df['Fare_int'] = df['Fare'].round().astype(int)
Q1 = df['Fare_int'].quantile(0.25)
Q3 = df['Fare_int'].quantile(0.75)
IQR = Q3 - Q1
outliers = df[(df['Fare_int'] < Q1 - 1.5*IQR) | (df['Fare_int'] > Q3 + 1.5*IQR)]
print(outliers)
```

6 Exercise 4: Pandas Essentials

1. Create and inspect Series:

```
s1 = pd.Series ([1,2,4,5])
print(s1.shape, s1.index)
s2 = pd.Series ([1,2,4,5],  index=['a','b','c','d'])
```

2. Build DataFrame and summarize:

```
df1 = pd.DataFrame({'name':['Alice','Bob','Charlie'], 'score' :[85,92,78]})
df2 = pd.DataFrame(np.random.randn(100,3),columns=list('ABC'))
df1.head(), df1.tail(), df1.info(), df1.describe()
```

3. Indexing with `loc/iloc`, sorting, and dropping:

```
df1.loc[0, 'score'], df2.iloc[2]
df2_sorted = df2.sort_values('A',ascending=False)
df2_sorted.drop(['B'], axis=1).head()
```

4. Handle missing data:

```
df_nan = pd.DataFrame({'X':[1,None,3],'Y':[None,2,3]})
df_nan.dropna(), df_nan.fillna(0)
```

5. Excel I/O:

```
df_weather = pd.read_excel('weather.xlsx')
df_weather.tail()
df_weather.to_excel('weather_updated.xlsx')
```

7 Exercise 5: Loading Open Dataset from UCI Repository

1. Load Wine dataset:

```
wine_url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data'
cols = ['Class','Alcohol','Malic acid','Ash','Alcalinity of ash',
        'Magnesium','Total phenols','Flavanoids','Nonflavanoid phenols',
        'Proanthocyanins','Color intensity','Hue',
        'OD280/OD315 of diluted wines','Proline']
df_wine = pd.read_csv(wine_url,names=cols)
print(df_wine.head(10))
```

2. Group by class:

```
wine_means = df_wine.groupby('Class').mean()
print(wine_means)
```

8 Exercise 6: scikit-learn Iris Dataset (Extended)

1. Load and preview Iris:

```
from sklearn import datasets
iris = datasets.load_iris()
df_iris = pd.DataFrame(iris.data,columns=iris.feature_names)
df_iris['target'] = iris.target
print(df_iris.head())
```

2. Train/test split:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    df_iris[iris.feature_names], df_iris['target'],
    test_size=0.3, random_state=42)
```

3. Model training and evaluation:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
model = LogisticRegression(max_iter=200)
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
print(classification_report(y_test,y_pred))
```

Submission Guidelines

1. Compile all work in a single Jupyter Notebook.

2. Write a short report (max 4 pages PDF) addressing:
   - Insights and observations from each exercise.
   - Strategies for handling missing data and outliers.
   - Interpretation of pivot/group-by results.
   - Reflection on model performance metrics.

3. Name the PDF 'XXYYYlab01.pdf' (XX = batch, YYY = registration number).