MACHINE LEARNING LAB

Pandas and MatplotLib

MUNADI SIAL

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

Pandas

- Pandas (Panel Data) is a library used for loading, handling and cleaning datasets before feeding to a machine learning algorithm
- Two data types: Series (column), Dataframe (table)
- A simple column can be created using the Series type

```
import pandas as pd
B = [10, 2, 7]

serB = pd.Series(B)
print(serB)
print(serB[0])

serC = pd.Series(B, index = ["X", "Y", "Z"])
print(serC)
print(serC["Z"])
```

Pandas

- Dictionaries can also be used to define Pandas series

```
import pandas as pd
D = {"Burger": 420, "Sandwich": 320, "Milkshake": 200}

serD = pd.Series(D)
print(serD)
```

Pandas Dataframe

- Datasets can be stored in the Data Frame type in pandas

```
import pandas as pd
datasetA = {
  'Company': ["Suzuki", "Toyota", "Ford", "Ford", "Suzuki"],
  'Number': [9553, 4314, 2192, 8317, 2555],
  'Color': ["White", "Black", "Black", "Red", "White"]}

dataframeA = pd.DataFrame(datasetA)
print(dataframeA)

print(dataframeA.loc[0]) # print row 0
print(dataframeA.loc[[0, 1, 4]]) # print rows 0, 1, 4
print(df.loc[0].at["x2"]) # print item in row 0, column x2
```

Pandas Dataframe

- Serial numbers in a dataframe can be changed to descriptive names

```
df2 = pd.DataFrame(datasetA, index = ["day1","day2","day3"])
print(df2)
print(df2.loc["day2"])
```

CSV Dataframe

- CSV tables can be loaded from disk into a dataframe

```
df3 = pd.read_csv(‘D:\\ML\\Lab4\\mydataset.csv')
```

- The loaded dataset can be displayed:

```
print(df3) # display first and last 5 rows
print(df3.head()) # display first 5 rows
print(df3.tail()) # display last 5 rows
print(df3.to_string()) # print entire table
print(df3.info()) # print table info
x1 = df3[‘column_name’].values.tolist() # get column in list
```

- To get the number of training examples in the dataset:

```
m = len(df3[‘column_name’].values.tolist())
```

Column Average

- At times, it is useful to calculate the average (mean, mode or median) of a specific column in the dataset

- To calculate mean of a column in a dataset

```
x = df[“x1"].mean()
```

- To calculate mean of a column from a range of rows

```
x = df[“x1"][10:75].mean()
```

- To calculate mode of a column in a dataset

```
x = df[“x1"].mode()[0]
```

- To calculate median of a column in a dataset

```
x = df[“x1"].median()
```

Dataset Cleaning

- Datasets need to be cleaned of rows with empty cells, duplicated rows, data in wrong format

- To remove rows with empty cells in the dataset

```
df.dropna(inplace = True) # remove rows with empty cells
```

- To replace empty cells with average value in the column

```
x = df[“x1"].mean()
df[“x1"].fillna(x, inplace = True)
```

- To remove duplicated rows from the dataset

```
df.drop_duplicates(inplace = True) # remove duplicate rows
```

Plotting

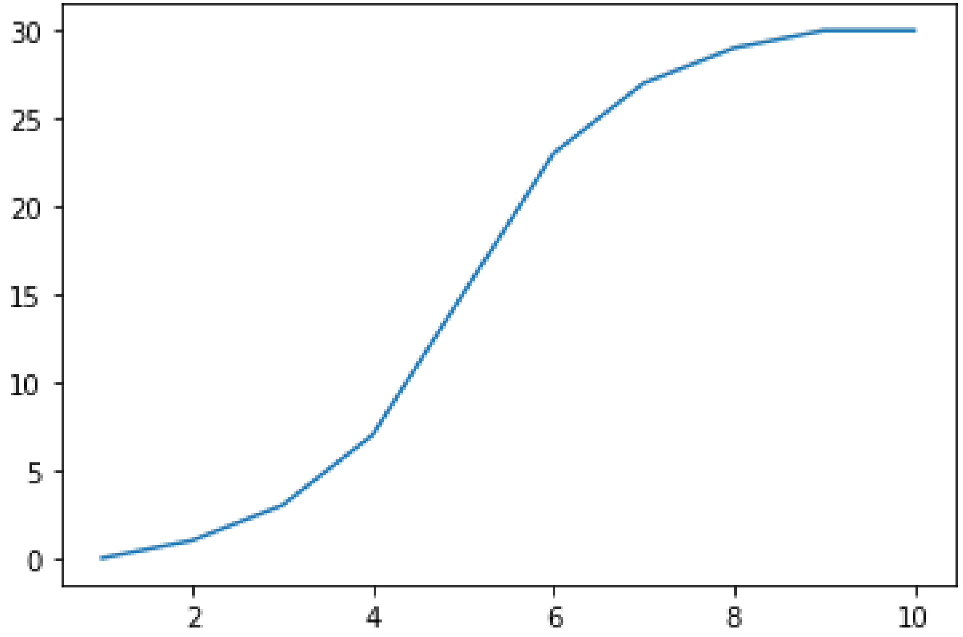
- For plotting, we can use the PyPlot submodule from the MatplotLib module

```
import matplotlib.pyplot as plt

time = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
speed = np.array([0, 1, 3, 7, 15, 23, 27, 29 ,30, 30])

plt.plot(time, speed)

plt.show()
```



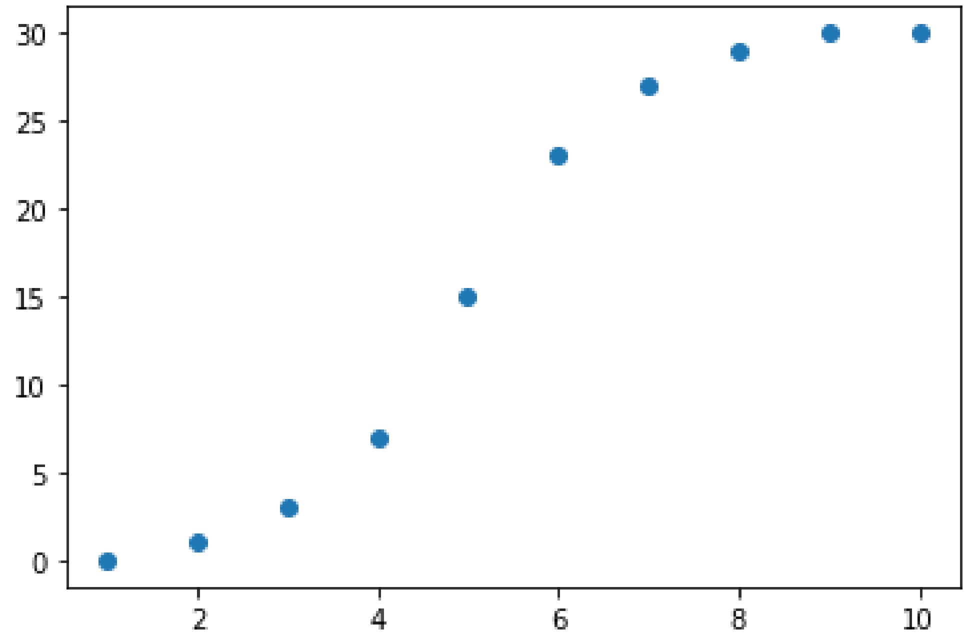
Plotting

```
import matplotlib.pyplot as plt

time = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
speed = np.array([0, 1, 3, 7, 15, 23, 27, 29 ,30, 30])

plt.scatter(time, speed)

plt.show()
```



Plotting

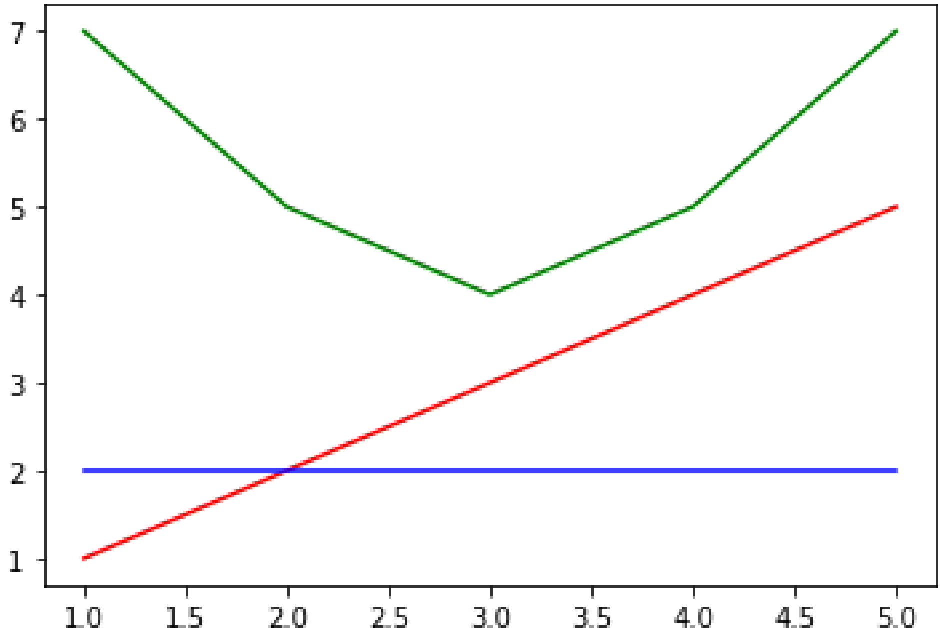
- Multiple plots can be made in the same graph

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

y1 = [1, 2, 3, 4, 5]
y2 = [2, 2, 2, 2, 2]
y3 = [7, 5, 4, 5, 7]

plt.plot(x, y1, 'r')
plt.plot(x, y2, 'b')
plt.plot(x, y3, 'g')

plt.show()
```



Plotting

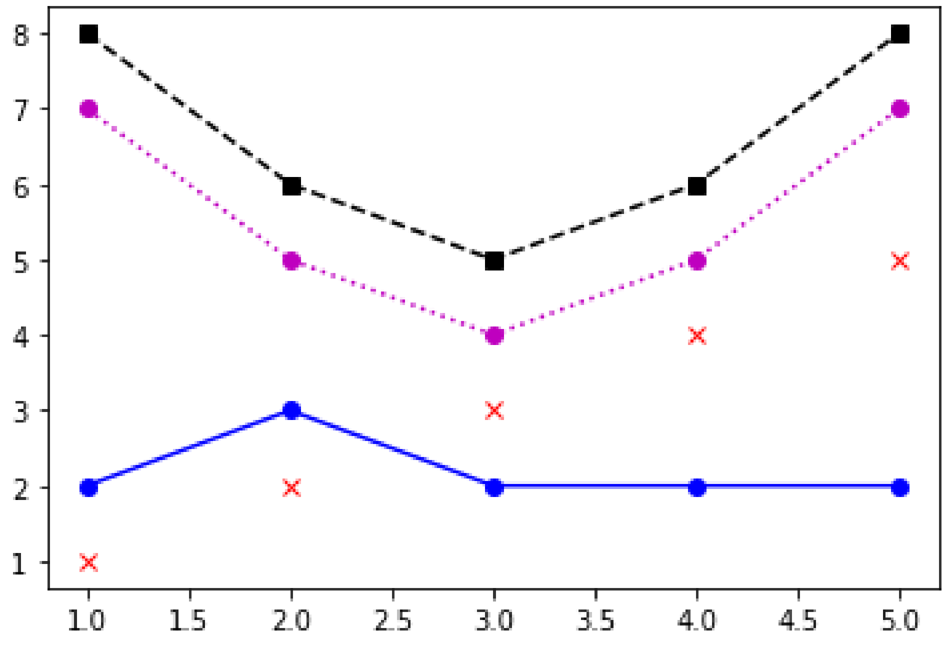
- Plots can be marked using a ‘marker line color’ string argument

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

y1 = [1, 2, 3, 4, 5]
y2 = [2, 3, 2, 2, 2]
y3 = [7, 5, 4, 5, 7]
y4 = [8, 6, 5, 6, 8]

plt.plot(x, y1, 'xr')
plt.plot(x, y2, 'o-b')
plt.plot(x, y3, 'o:m')
plt.plot(x, y4, 's--k')

plt.show()
```



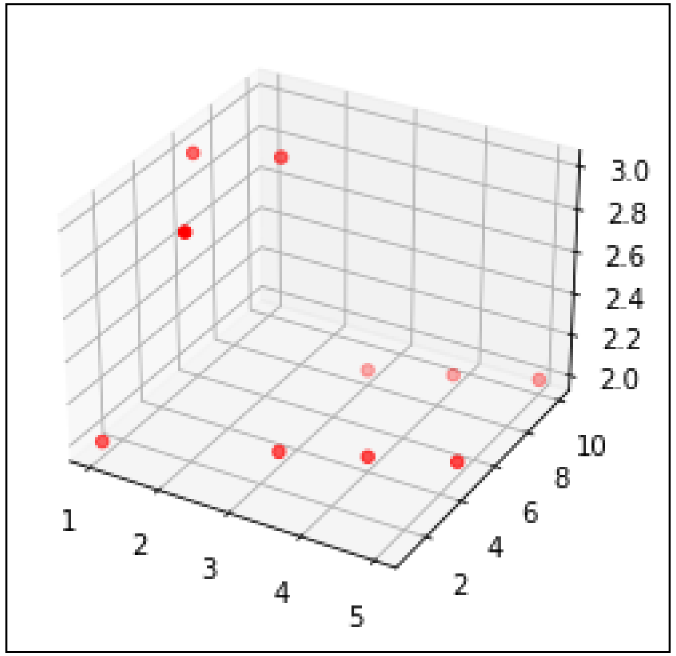
Plotting

- 3-D plots can also be implemented

```
x = [1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
y1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y2 = [2, 3, 2, 2, 2, 3, 3, 2, 2, 2]


ax = plt.axes(projection ="3d")
ax.scatter3D(x, y1, y2, color = "r")

plt.show()
```



Lab Tasks

- Download the materials from LMS
- Perform the Lab Tasks given in the manual
- Convert the completed manual into .pdf and submit on LMS