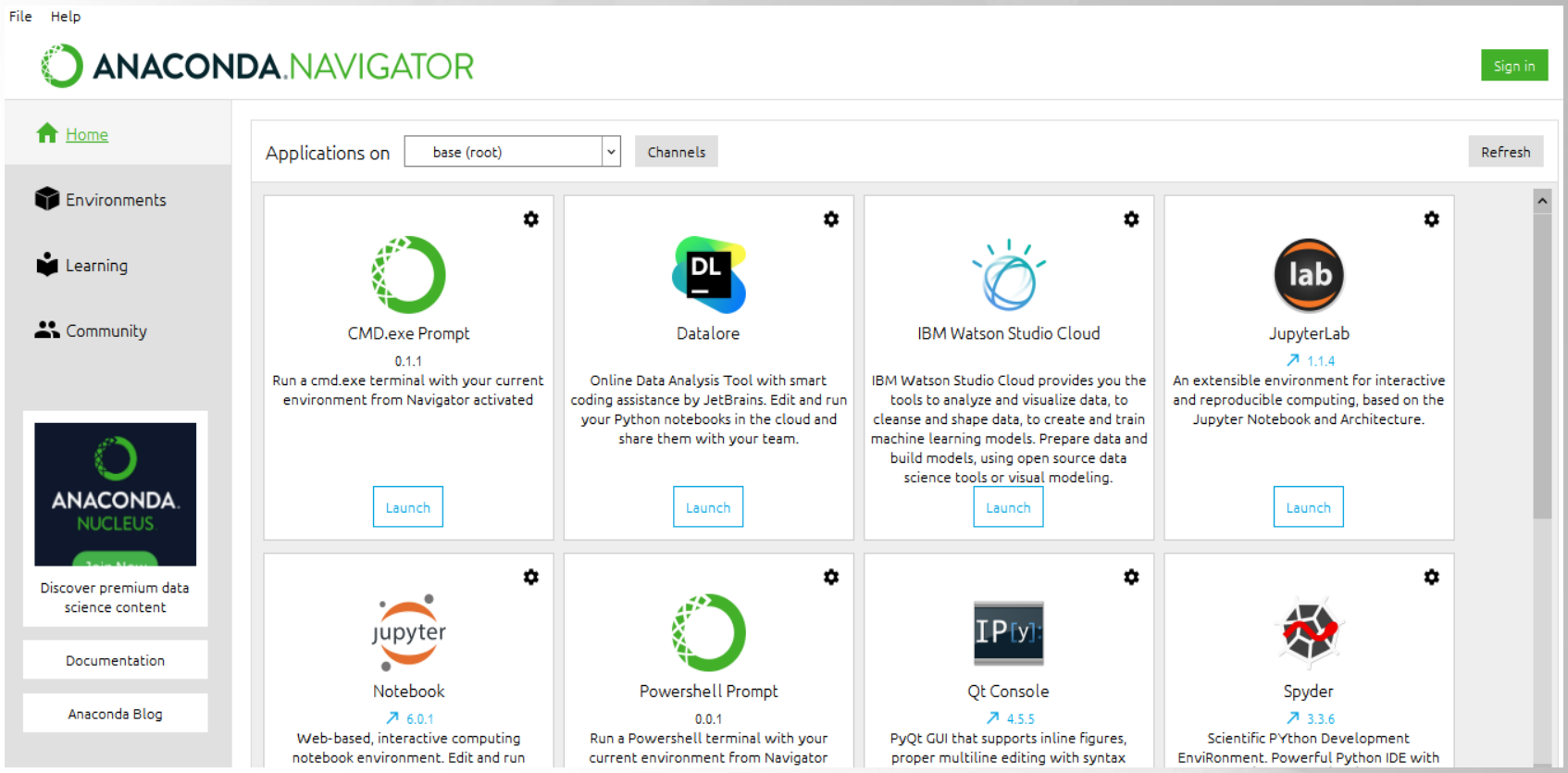
Machine Learning
CSE - 465

ML Tools

Anaconda3

- Install anaconda3 from the link :

https://docs.anaconda.com/anaconda/install/windows/



Anaconda Prompt

- Open your anaconda prompt and install the libraries given in the next slides.

scikit-learn

- Install *scikit-learn* library

  pip install sklearn

- Scikit-learn is a free software machine learning library for the Python programming language.
- It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN.
- It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

pandas

- Install *pandas* library

  pip install pandas

- DataFrame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.

pandas

- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- Provides data filtration.

matplotlib

- Install matplotlib library

  pip install matplotlib

- Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy
- It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

numpy

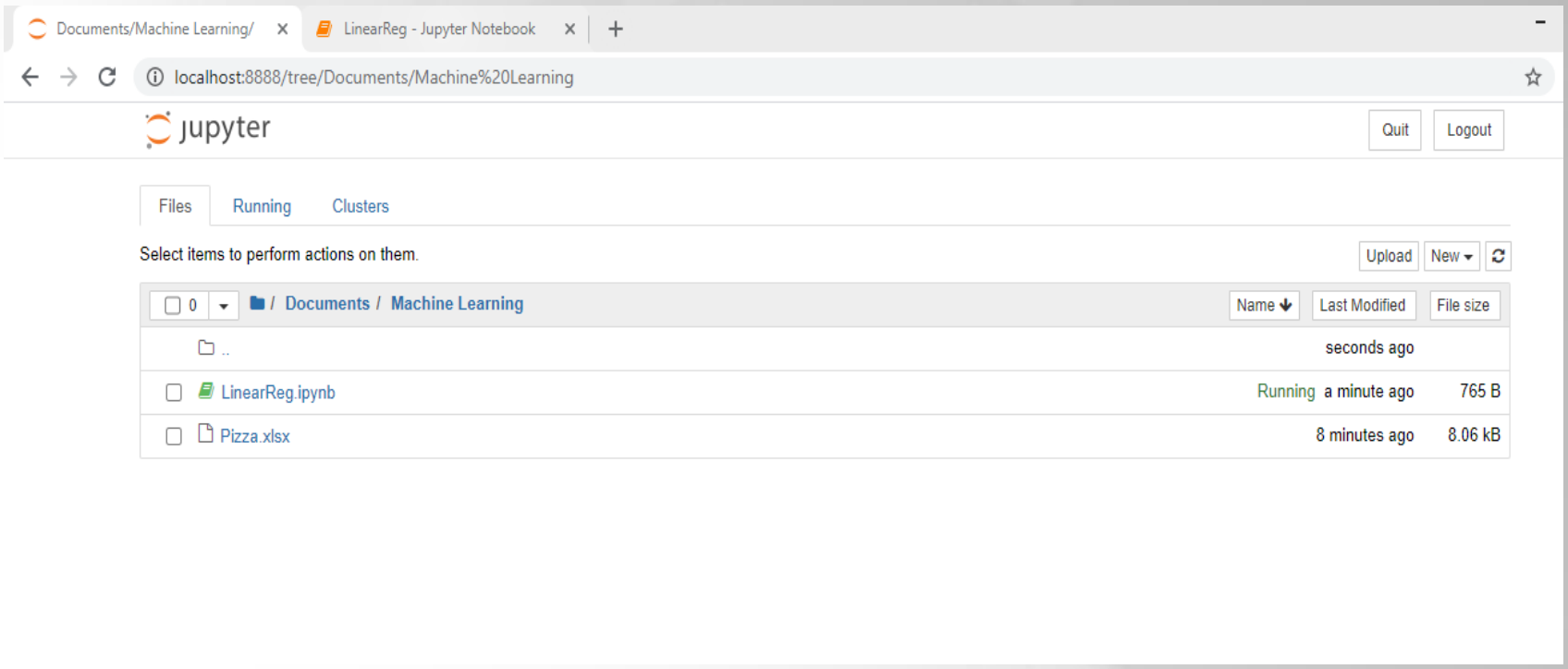
- Install numpy library

  pip install numpy

- NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Jupyter Notebook

- Now open Jupyter Notebook from Anaconda Navigator
- Go to your desired project folder



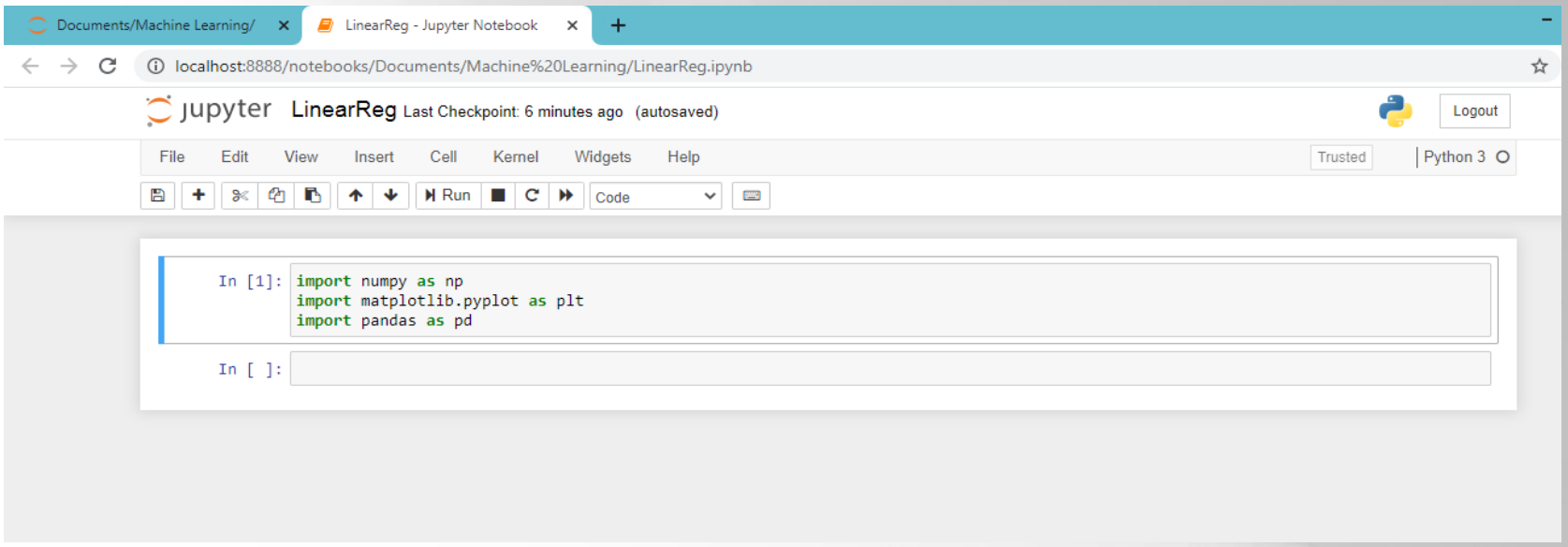
Loading Libraries

- Import necessary libraries using the following commands on Jupyter Notebook

  import numpy as np

  import matplotlib.pyplot as plt

  import pandas as pd



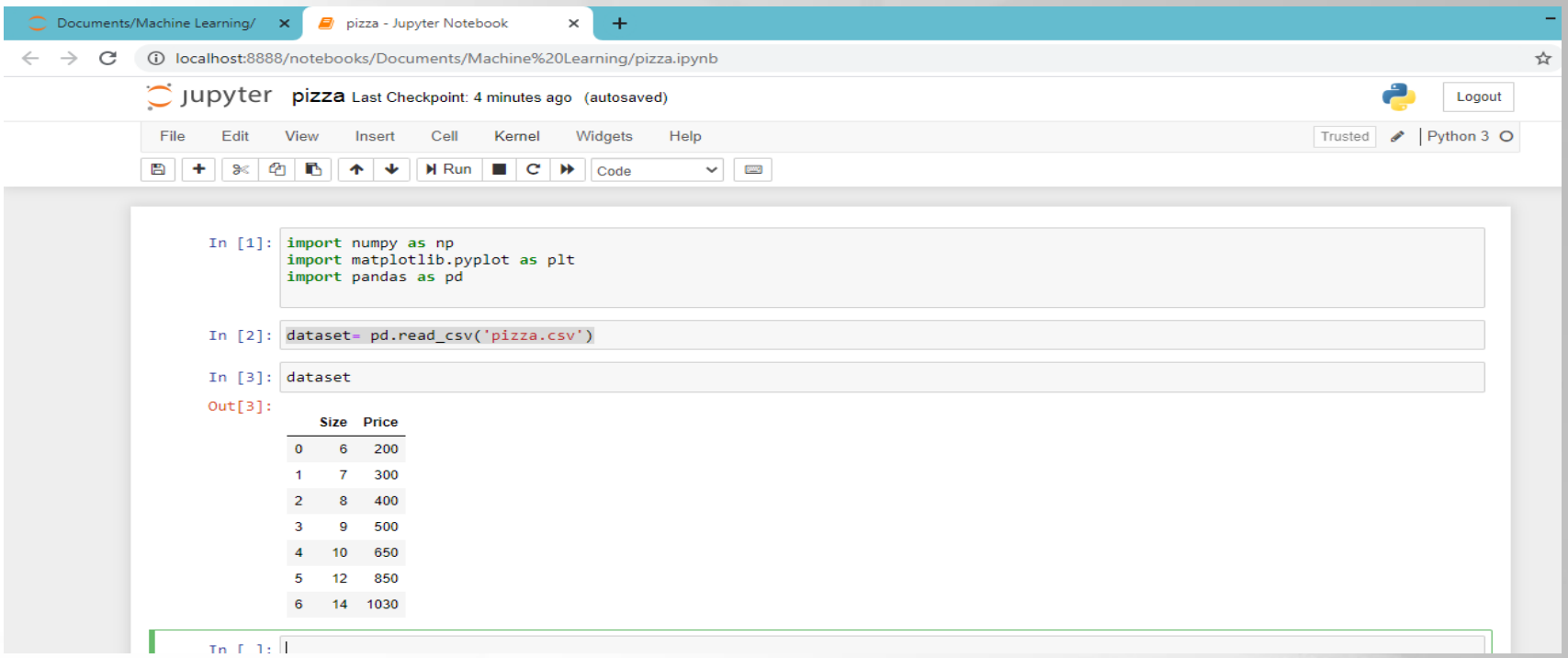
Loading Dataset

- Now load the dataset

  dataset= pd.read_csv('pizza.csv')

- Display the dataset

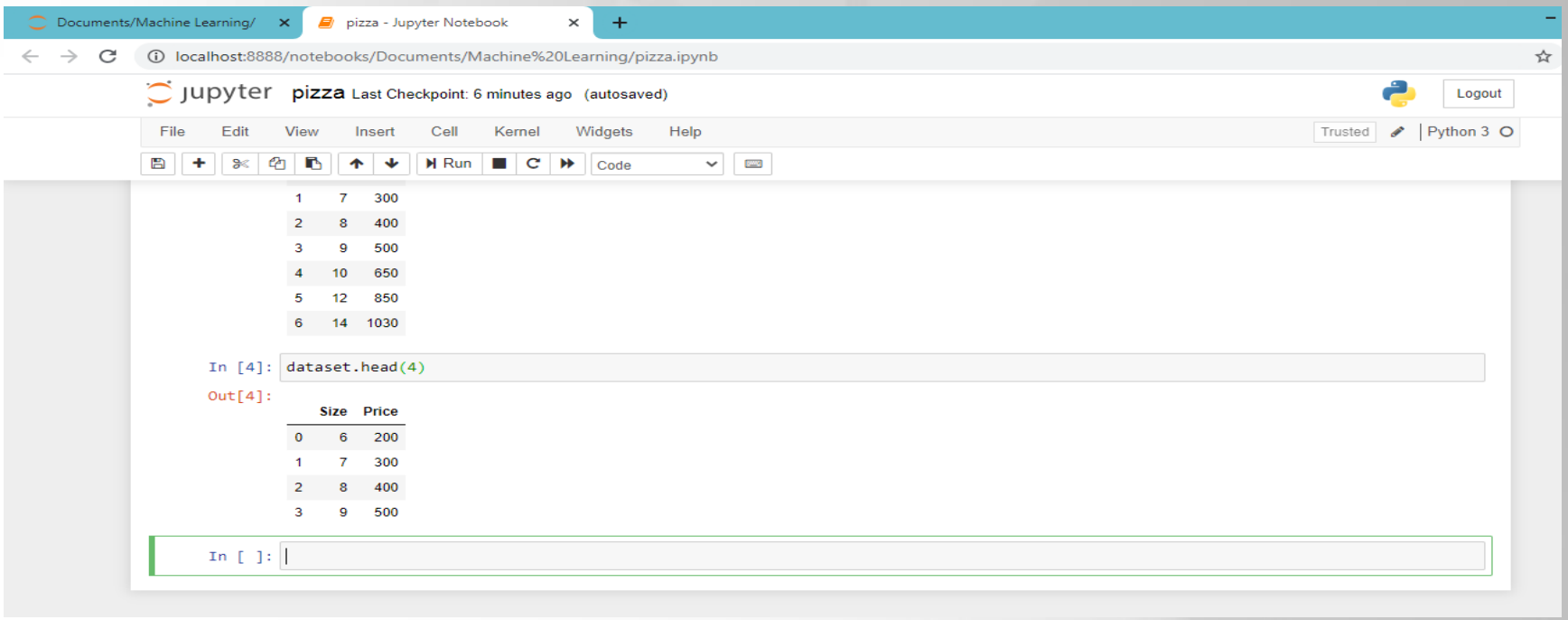
  dataset



Dataset Manipulation

- Use the following code to see the amount of data you want to display.

```
dataset.head(4)
```



Dataset Manipulation

- Check the shape of the dataset using

  dataset.shape

```
In [6]: dataset.shape
Out[6]: (7, 2)

In [ ]:
```

- Check for null entry in columns using

  dataset.isnull().any()

```
In [7]: dataset.isnull().any()
Out[7]: Size     False
        Price    False
        dtype: bool

In [ ]:
```

Dataset Manipulation

- Assigning variables for the column attributes
- Assign x to the independent variable
- Assign y to the predictor variable
- Display whichever you need just by writing the name of the variable.

```
In [16]: x = dataset[['Size']]
         y = dataset['Price']

In [18]: x
```

Out[18]:

	Size
0	6
1	7
2	8
3	9
4	10
5	12
6	14

```
In [19]: y
```

```
Out[19]: 0    200
         1    300
         2    400
         3    500
```

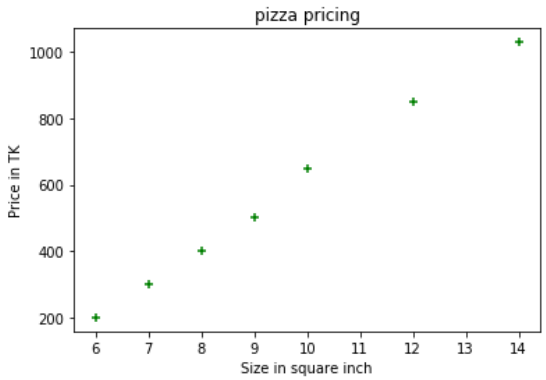
Plotting Dataset

- Use the following command to plot the dataset using matplotlib

```
plt.scatter(dataset['Size'],dataset['Price'],marker='+',color='green')
plt.xlabel('Size in square inch')
plt.ylabel('Price in TK')
plt.title('pizza pricing')
```

```
In [25]: plt.scatter(dataset['Size'],dataset['Price'],marker='+',color='green')
         plt.xlabel('Size in square inch')
         plt.ylabel('Price in TK')
         plt.title('pizza pricing')

Out[25]: Text(0.5, 1.0, 'pizza pricing')
```



Thank You