Applications of Machine Learning for Networking

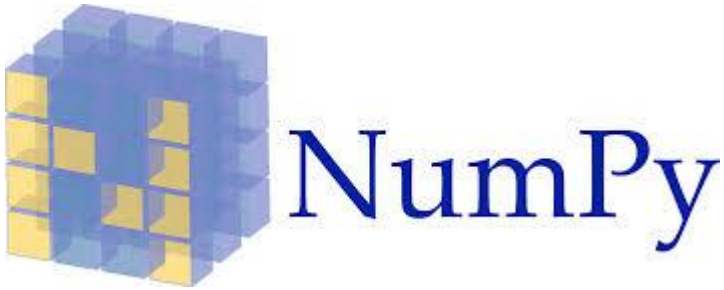
Lab 1 (Classification)

Outlines

- Useful Packages
  - Numpy
  - SciPy
  - Matplotlib
  - Pandas
- The Steps in ML-based Solutions
- Lab Requirement

Numpy

- Base N-dimensional array operation
- [Numpy Document]



NumPy

In [2]:
```
import numpy as np

x = np.array([[1, 2, 3], [4, 5, 6]])
print("x:\n{}".format(x))
```

```
x:
[[1 2 3]
 [4 5 6]]
```

SciPy

- Scientific Computing
- SciPy Document

SciPy

In [3]:
```
from scipy import sparse

# Create a 2D NumPy array with a diagonal of ones, and zeros everywhere else
eye = np.eye(4)
print("NumPy array:\n", eye)
```

```
NumPy array:
 [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

In [4]:
```
# Convert the NumPy array to a SciPy sparse matrix in CSR format
# Only the nonzero entries are stored
sparse_matrix = sparse.csr_matrix(eye)
print("\nSciPy sparse CSR matrix:\n", sparse_matrix)
```

```
SciPy sparse CSR matrix:
   (0, 0)       1.0
  (1, 1)        1.0
  (2, 2)        1.0
  (3, 3)        1.0
```

In [5]:
```
data = np.ones(4)
row_indices = np.arange(4)
col_indices = np.arange(4)
eye_coo = sparse.coo_matrix((data, (row_indices, col_indices)))
print("COO representation:\n", eye_coo)
```

```
COO representation:
   (0, 0)       1.0
  (1, 1)        1.0
  (2, 2)        1.0
  (3, 3)        1.0
```

Matplotlib

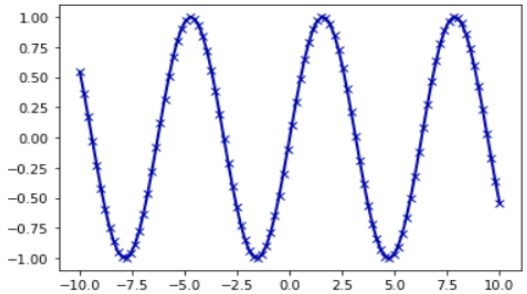
- Comprehensive 2-D plotting
- Matplotlib Document

matplotlib

In [6]:
```
%matplotlib inline
import matplotlib.pyplot as plt

# Generate a sequence of numbers from -10 to 10 with 100 steps in between
x = np.linspace(-10, 10, 100)
# Create a second array using sine
y = np.sin(x)
# The plot function makes a line chart of one array against another
plt.plot(x, y, marker="x")
```

Out[6]:
```
[<matplotlib.lines.Line2D at 0x7f93fb04d2d0>]
```



Pandas

- Data structures & analysis
- [Pandas Document]

pandas

In [7]:
```
import pandas as pd

# create a simple dataset of people
data = {'Name': ["John", "Anna", "Peter", "Linda"],
        'Location' : ["New York", "Paris", "Berlin", "London"],
        'Age' : [24, 13, 53, 33]
       }

data_pandas = pd.DataFrame(data)
# IPython.display allows "pretty printing" of dataframes
# in the Jupyter notebook
display(data_pandas)
```

	Name	Location	Age
0	John	New York	24
1	Anna	Paris	13
2	Peter	Berlin	53
3	Linda	London	33

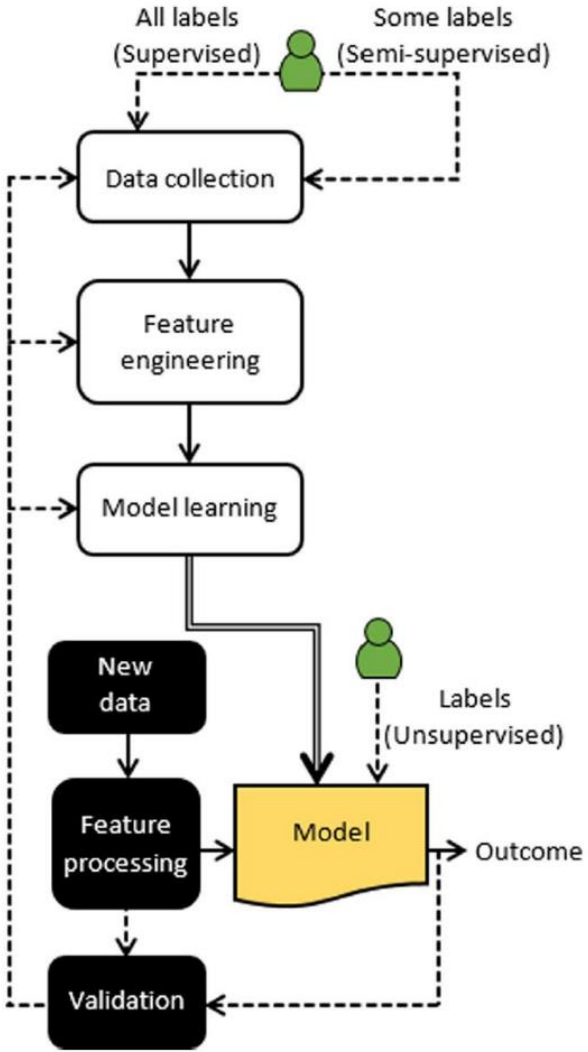
In [8]:
```
# Select all rows that have an age column greater than 30
display(data_pandas[data_pandas.Age > 30])
```

	Name	Location	Age
2	Peter	Berlin	53
3	Linda	London	33

The steps in ML-based solutions

1. Data collection & preprocessing

- We will prepare dataset for you in lab1
- You can try …
  - Clean data
  - Shuffle data
  - Re-sample data
  - Balance data
  - Split data into the training set and validation set



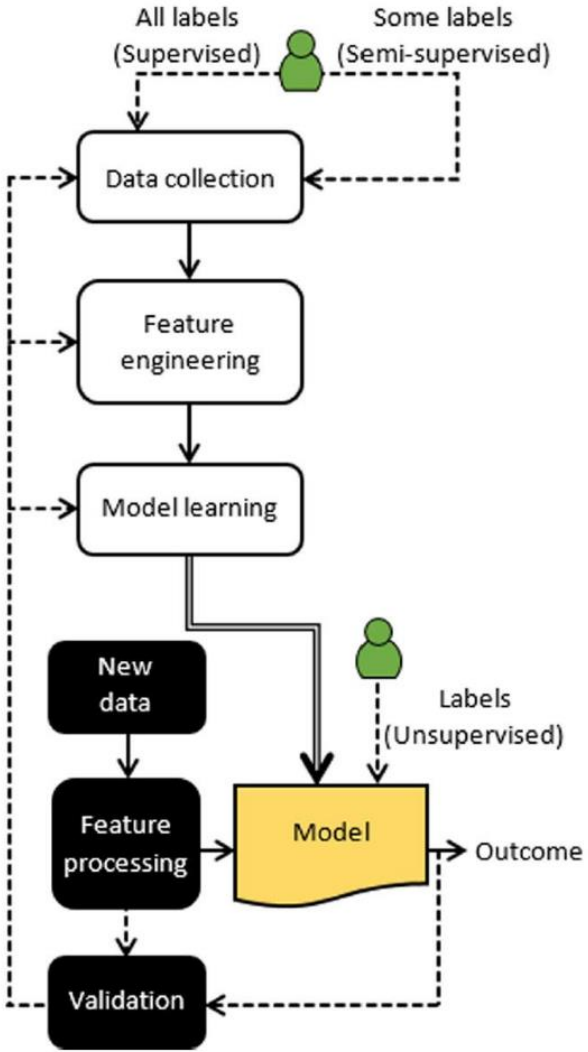
The steps in ML-based solutions

2. Feature engineering

- You can try …
  - Data type conversion
  - Data transformation

    Normalization, Standardization
  - Regularization
  - Visualization

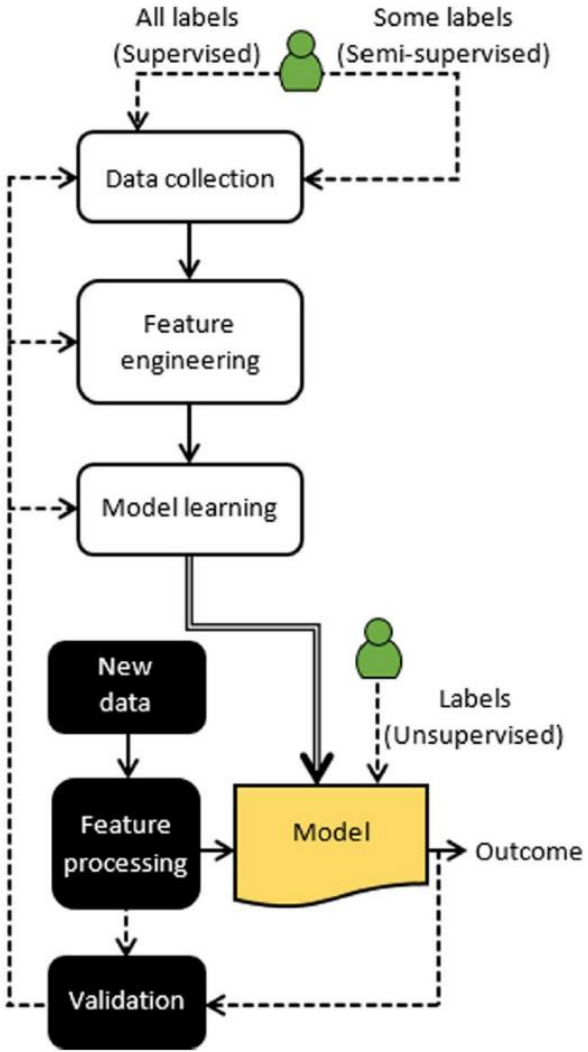
    (For more easily understand your dataset)
  - Feature extraction
  - Feature selection



The steps in ML-based solutions

3. Model learning

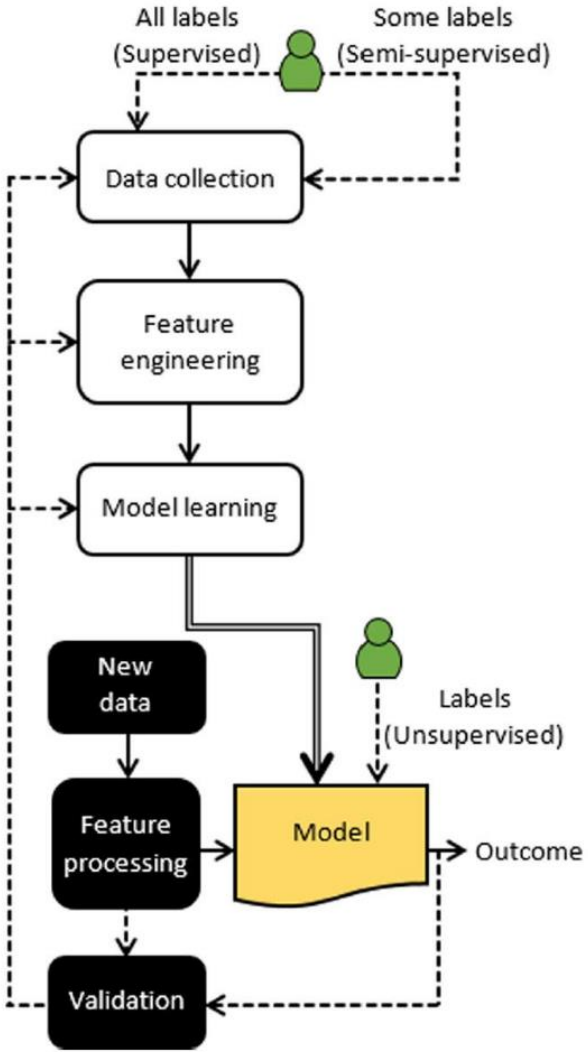
- You can try …
  - Choose model
  - Train model
  - Evaluate model
  - Tune parameters



The steps in ML-based solutions

4. Prediction

- Time to show how good your model is



Lab 1 - Task

- Build a model that is capable of distinguishing between legitimate and illegitimate connections in a computer network
- Classify 5 types of connections including (normal, DOS, R2L, U2R, Probing)
- You can follow the recommended steps in ML-based solution.
- More Detail

Lab 1 - Dataset

- Dataset
  - kddcup.names: A list of features.
  - kddcup.data.zip: The full data set.
  - kddcup.data_10_percent.zip: A subset containing only 10% of the original data.
  - kddcup.testdata.unlabeled.zip.
  - corrected.zip: Test data with corrected labels.
  - training_attack_types: A list of intrusion types.
- You can try the full dataset or the 10% dataset depending on the computing power of the your computer
- Dataset download

Lab 1 - Dataset

- Field description
- Attack type mapping
  - Classify to 5 types of connections

0	normal
1	probe
2	denial of service (DOS)
3	user-to-root (U2R)
4	remote-to-local (R2L)

Lab 1 - Requirement

1. Report (.pdf)
    - Explain what you did in this lab (Data preprocessing, Feature engineering, Model learning, etc.)
    - Show, explain, and discuss your results
2. Source code (.py or .ipynb)

Note: Please zip all your files into a **.zip** extension file and name it with your **student ID** (e.g., 0123456.zip). You can discuss with your classmates, but **Plagiarism is forbidden.**

Lab 1 – Report Requirement

Do K-fold cross validation (K > 2)	10%
Describe how you performed data processing	10%
Visualize data	10%
Do feature transformation	10%
Do feature selection and explain why you select the features	10%
Describe what feature engineering skills you used	10%
Try at least 3 different ML algorithms	10%
Tuning a model with at least 100 iterations	10%
Recall, precision, F1-Score	5%
Confusion matrix	5%
Discussions and conclusion	10%

Lab 1 – Requirement Hints

1. Do K-fold cross validation: **from sklearn.model_selection import KFold**
2. Describe how you performed data processing
3. Visualize data: **Ref**
4. Do feature transformation: **Ref**
5. Do feature selection and explain why you select the features: **Ref**
6. Describe what feature engineering skills you used
7. Try at least 3 different ML algorithms: **Decision trees, Random forest, SVM, KNN, Naïve Bayes, Ensemble, MLP...**
8. Tuning a model with at least 100 iterations (different combinations of parameters)
9. Recall, percision F1-Score: **Ref**
10. Confusion matrix: **Ref**
11. Discussions and conclusion

Lab 1 - Supplement

- https://github.com/Yorko/mlcourse.ai
- https://developers.google.com/machine-learning/crash-course/ml-intro
- https://www.kaggle.com/learn/overview

Enjoy the lab