Scientific Python (continued) and Decision Tree Learning(1)

Dr. Xiaowei Huang

https://cgi.csc.liv.ac.uk/~xiaowei/

What is SciPy?

- SciPy is a library of algorithms and mathematical tools built to work with NumPy arrays.
  - linear algebra - *scipy.linalg*
  - statistics - *scipy.stats*
  - optimization - *scipy.optimize*
  - sparse matrices - *scipy.sparse*
  - signal processing - *scipy.signal*
  - etc.

Scipy Linear Algebra

- Slightly different from numpy.linalg. Always uses BLAS/LAPACK support, so could be faster.
- Some more functions.
- Functions can be slightly different.

Scipy Optimization

- General purpose minimization: CG, BFGS, least-squares
- Constrained minimization; non-negative least-squares
- Minimize using simulated annealing
- Scalar function minimization
- Root finding
- Check gradient function Line search

Scipy Statistics

- Mean, median, mode, variance, kurtosis
- Pearson correlation coefficient
- Hypothesis tests (ttest, Wilcoxon signed-rank test, Kolmogorov-Smirnov)
- Gaussian kernel density estimation

See also SciKits (or scikit-learn).

Scipy sparse

- Sparse matrix classes: CSC, CSR, etc.
- Functions to build sparse matrices
- sparse.linalg module for sparse linear algebra
- sparse.csgraph for sparse graph routines

Scipy signal

- Convolutions
- B-splines
- Filtering
- Continuous-time linear system
- Wavelets
- Peak finding

Scipy IO

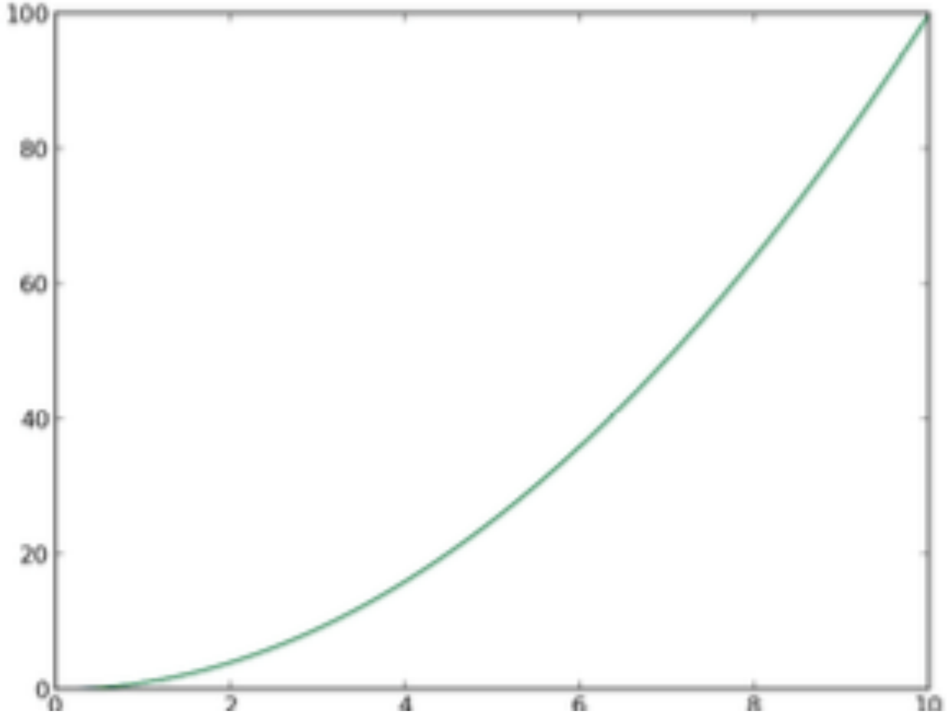
- Methods for loading and saving data
  - Matlab files
  - Matrix Market files (sparse matrices)
  - Wav files

What is Matplotlib?

- Plotting library for Python
- Works well with Numpy
- Syntax similar to Matlab

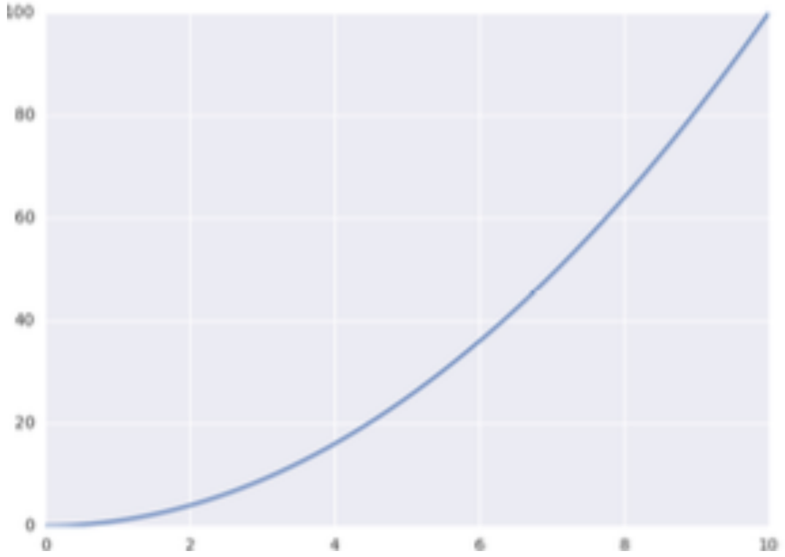
Scatter Plot

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
plt.plot(x, y)
plt.show()
```



Seaborn makes plot pretty

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
plt.plot(x, y)
plt.show()
```

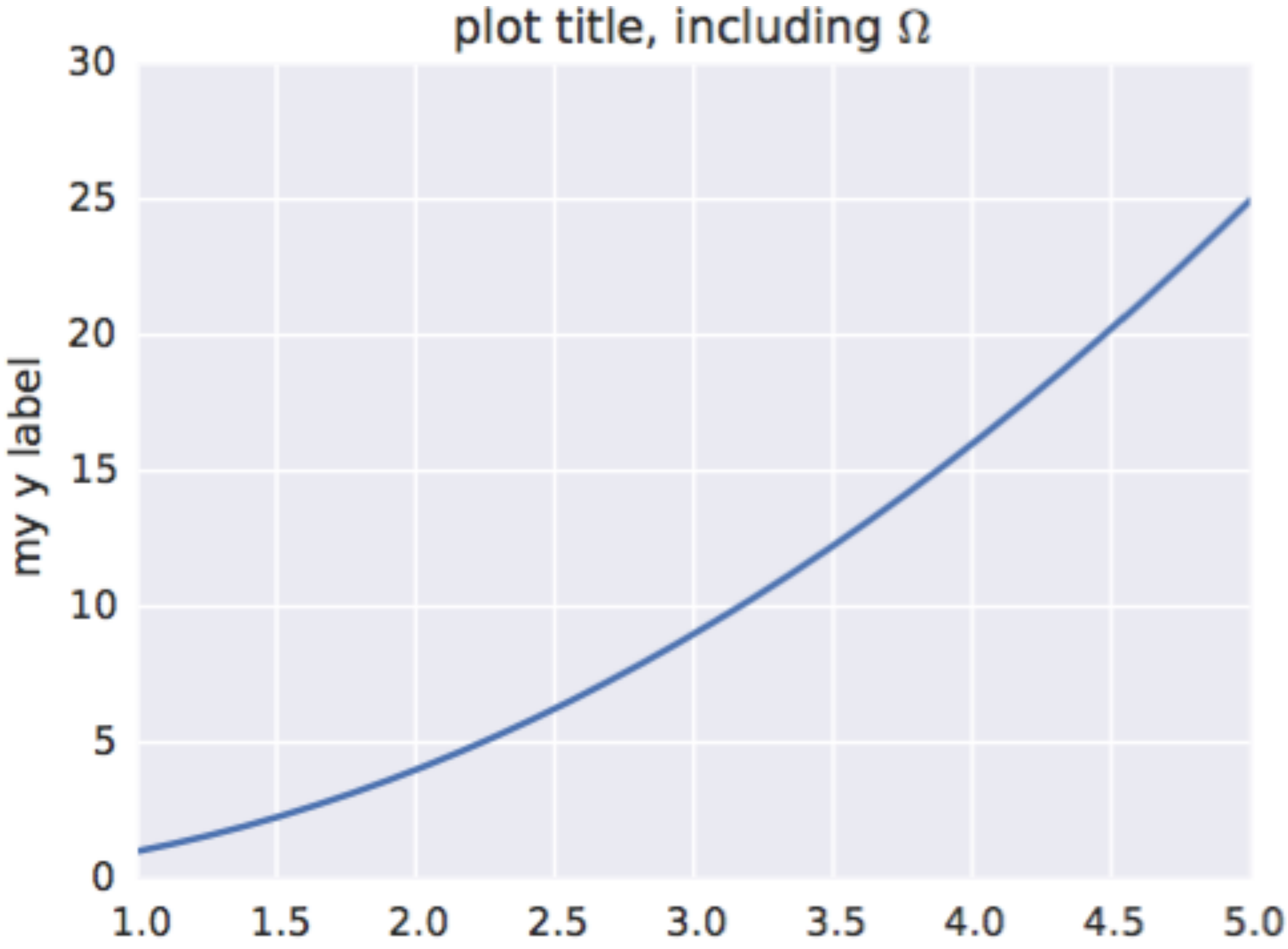


Scatter Plot

- Adding titles and labels

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

f, ax = plt.subplots(1, 1, figsize=(5,4))

x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
ax.plot(x, y)
ax.set_xlim((1, 5))
ax.set_ylim((0, 30))
ax.set_xlabel('my x label')
ax.set_ylabel('my y label')
ax.set_title('plot title, including $\Omega$')

plt.tight_layout()
plt.savefig('line_plot_plus.pdf')
```

Scatter Plot

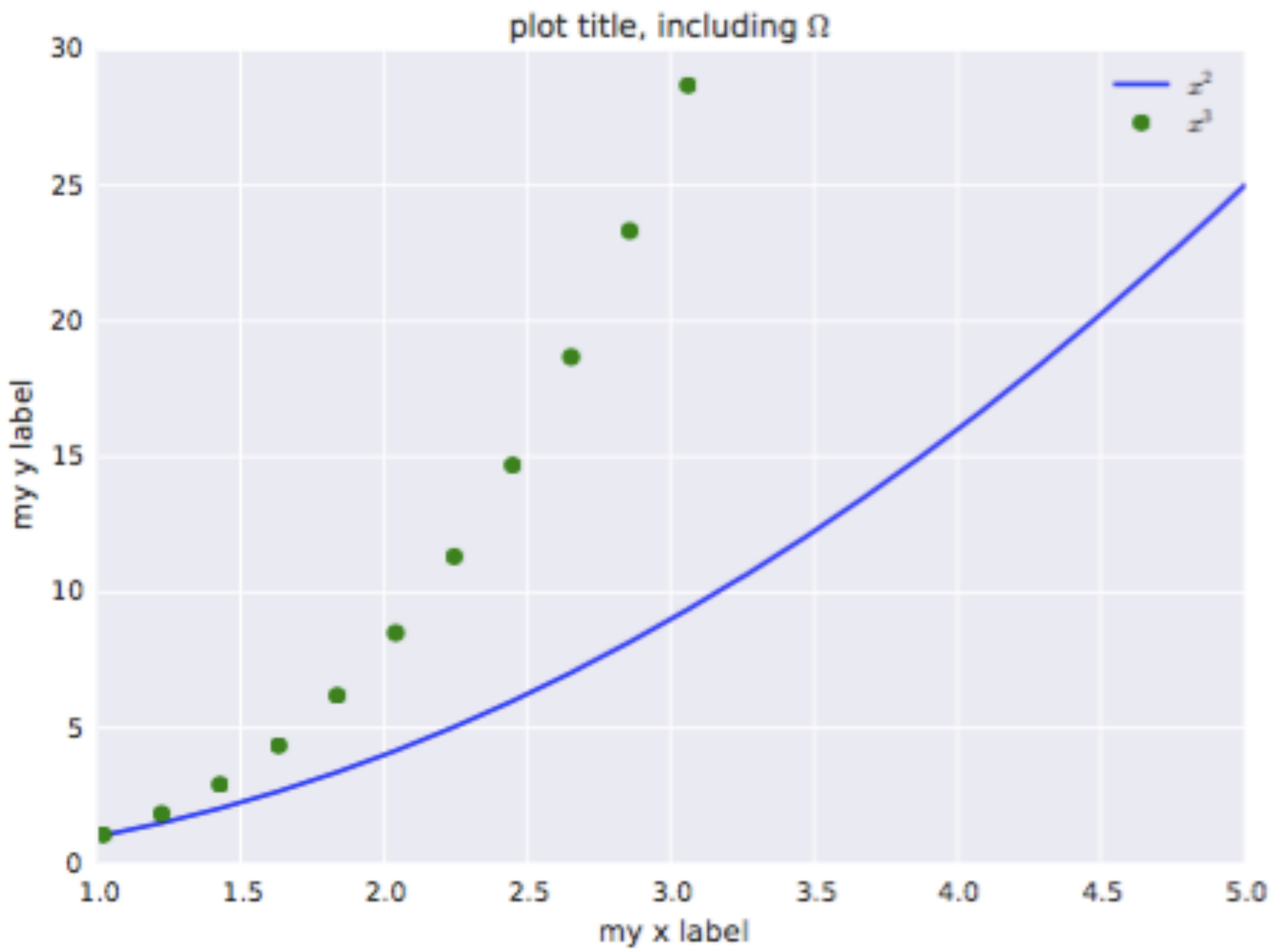


Scatter Plot

- Adding multiple lines and a legend

```
x = np.linspace(0, 10, 50)
y1 = np.power(x, 2)
y2 = np.power(x, 3)

plt.plot(x, y1, 'b-', label='$x^2$')
plt.plot(x, y2, 'go', label='$x^3$')
plt.xlim((1, 5))
plt.ylim((0, 30))
plt.xlabel('my x label')
plt.ylabel('my y label')
plt.title('plot title, including $\Omega$')
plt.legend()

plt.savefig('line_plot_plus2.pdf')
```

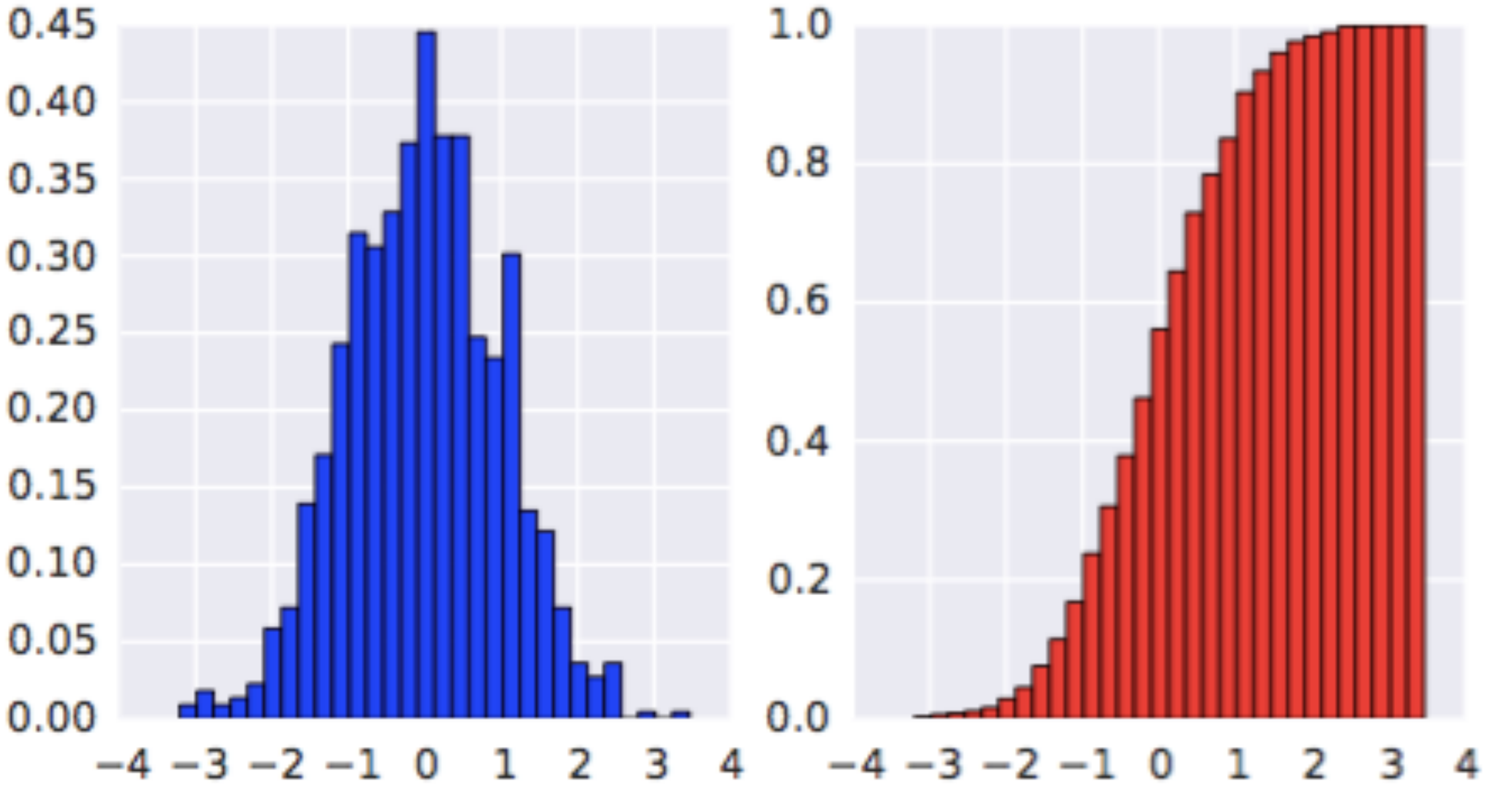
Scatter Plot



Histogram

```
data = np.random.randn(1000)

f, (ax1, ax2) = plt.subplots(1, 2, figsize=(6,3))

# histogram (pdf)
ax1.hist(data, bins=30, normed=True, color='b')

# empirical cdf
ax2.hist(data, bins=30, normed=True, color='r',
         cumulative=True)

plt.savefig('histogram.pdf')
```

Histogram


0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4 −3 −2 −1 0 1 2 3 4
1.0
0.8
0.6
0.4
0.2
0.0
−4 −3 −2 −1 0 1 2 3 4

Box Plot

```
samp1 = np.random.normal(loc=0., scale=1., size=100)
samp2 = np.random.normal(loc=1., scale=2., size=100)
samp3 = np.random.normal(loc=0.3, scale=1.2, size=100)

f, ax = plt.subplots(1, 1, figsize=(5,4))

ax.boxplot((samp1, samp2, samp3))
ax.set_xticklabels(['sample 1', 'sample 2', 'sample 3'])
plt.savefig('boxplot.pdf')
```

Box Plot

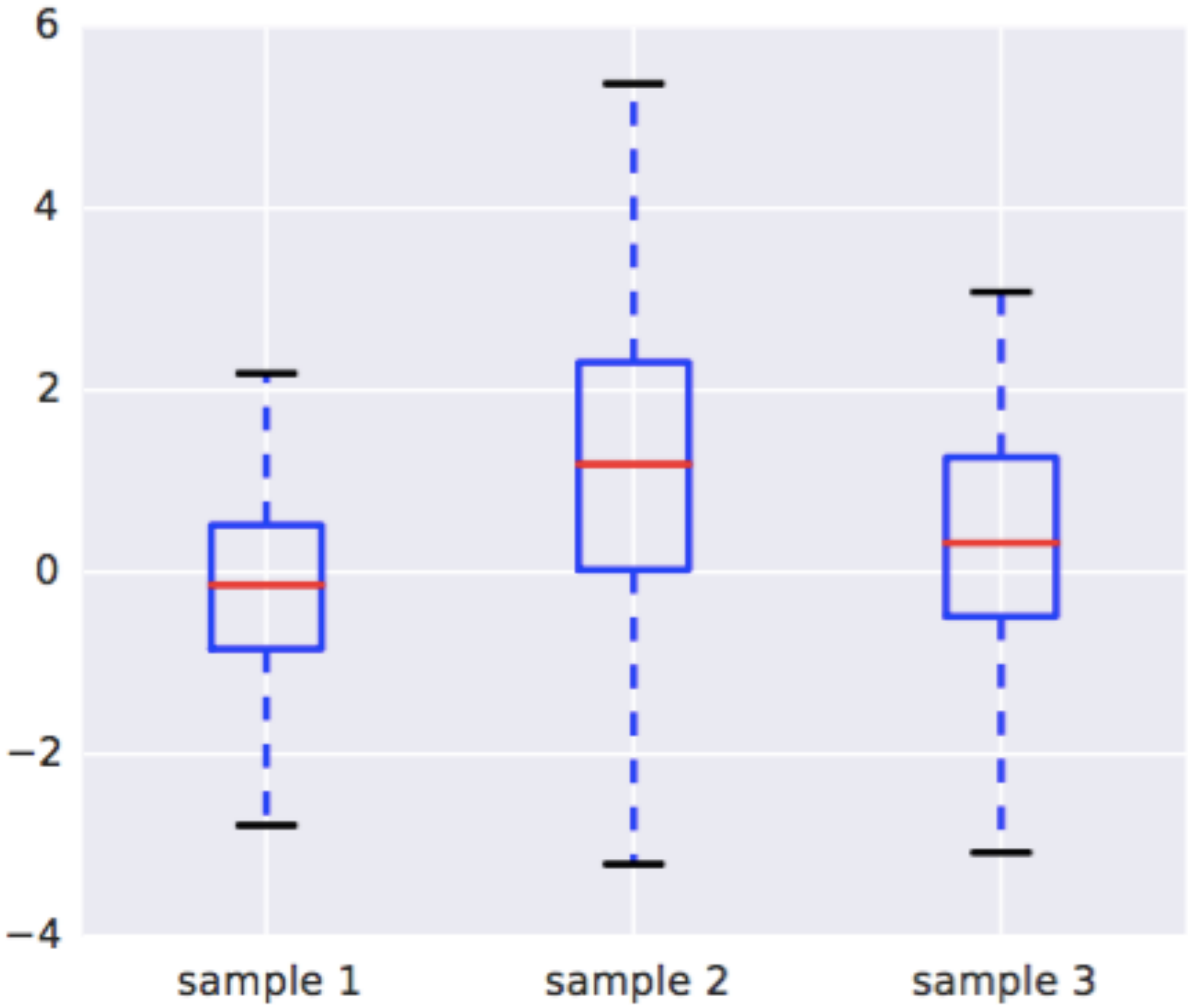
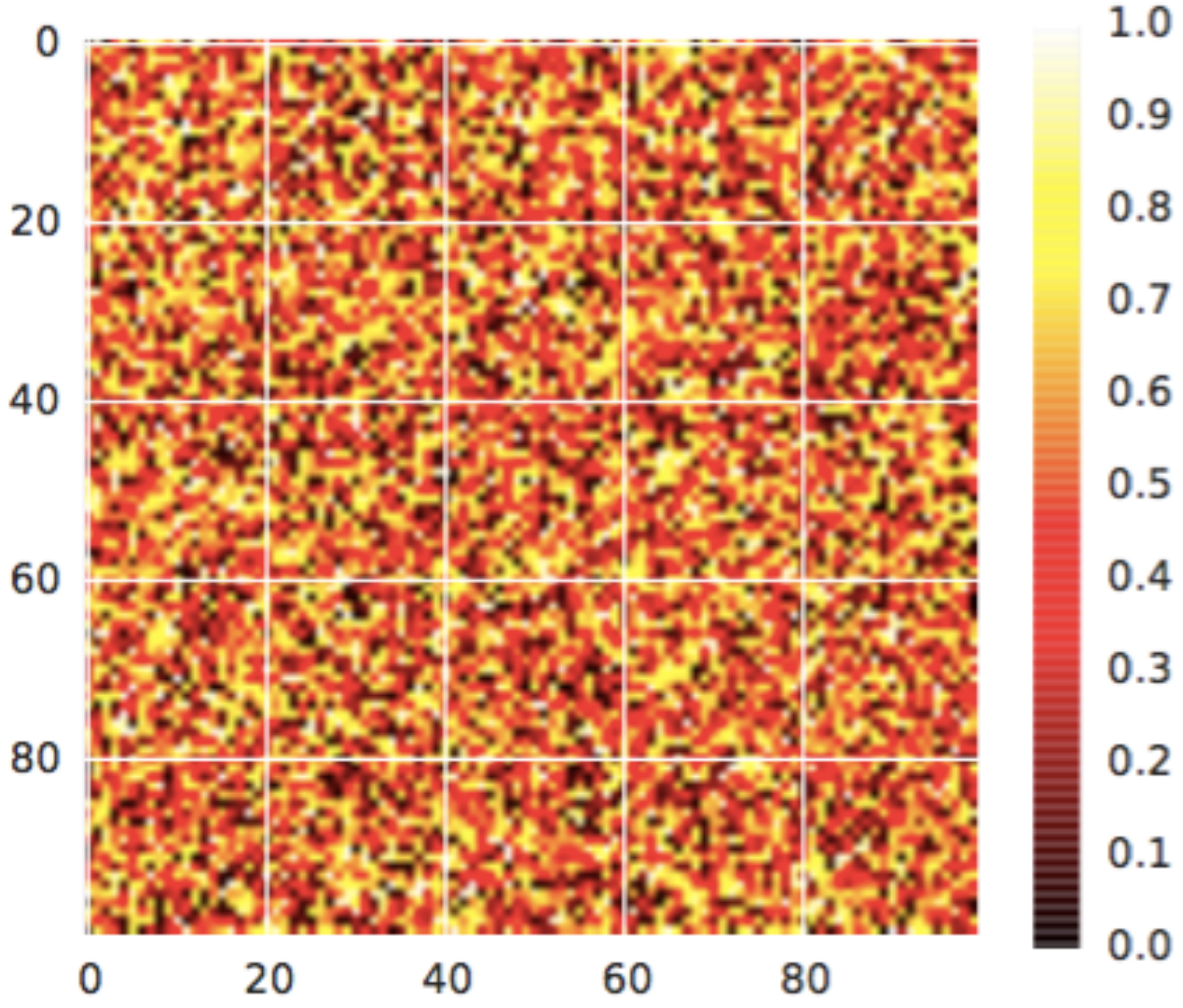


Image Plot

```
A = np.random.random((100, 100))

plt.imshow(A)
plt.hot()
plt.colorbar()

plt.savefig('imageplot.pdf')
```

Image Plot

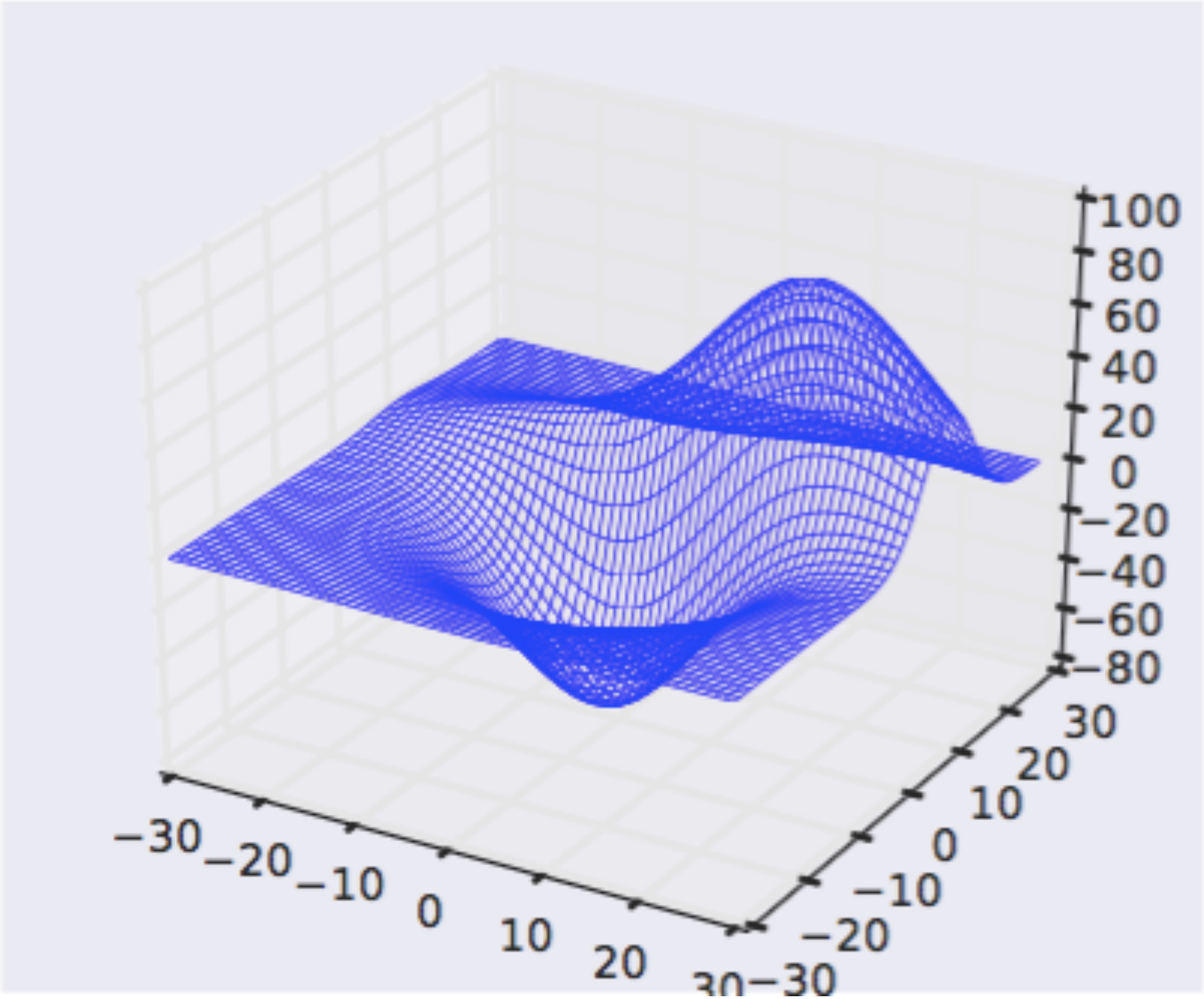


Wire Plot

- matplotlib toolkits extend funtionality for other kinds of visualization

```
from mpl_toolkits.mplot3d import axes3d

ax = plt.subplot(111, projection='3d')
X, Y, Z = axes3d.get_test_data(0.1)
ax.plot_wireframe(X, Y, Z, linewidth=0.1)

plt.savefig('wire.pdf')
```

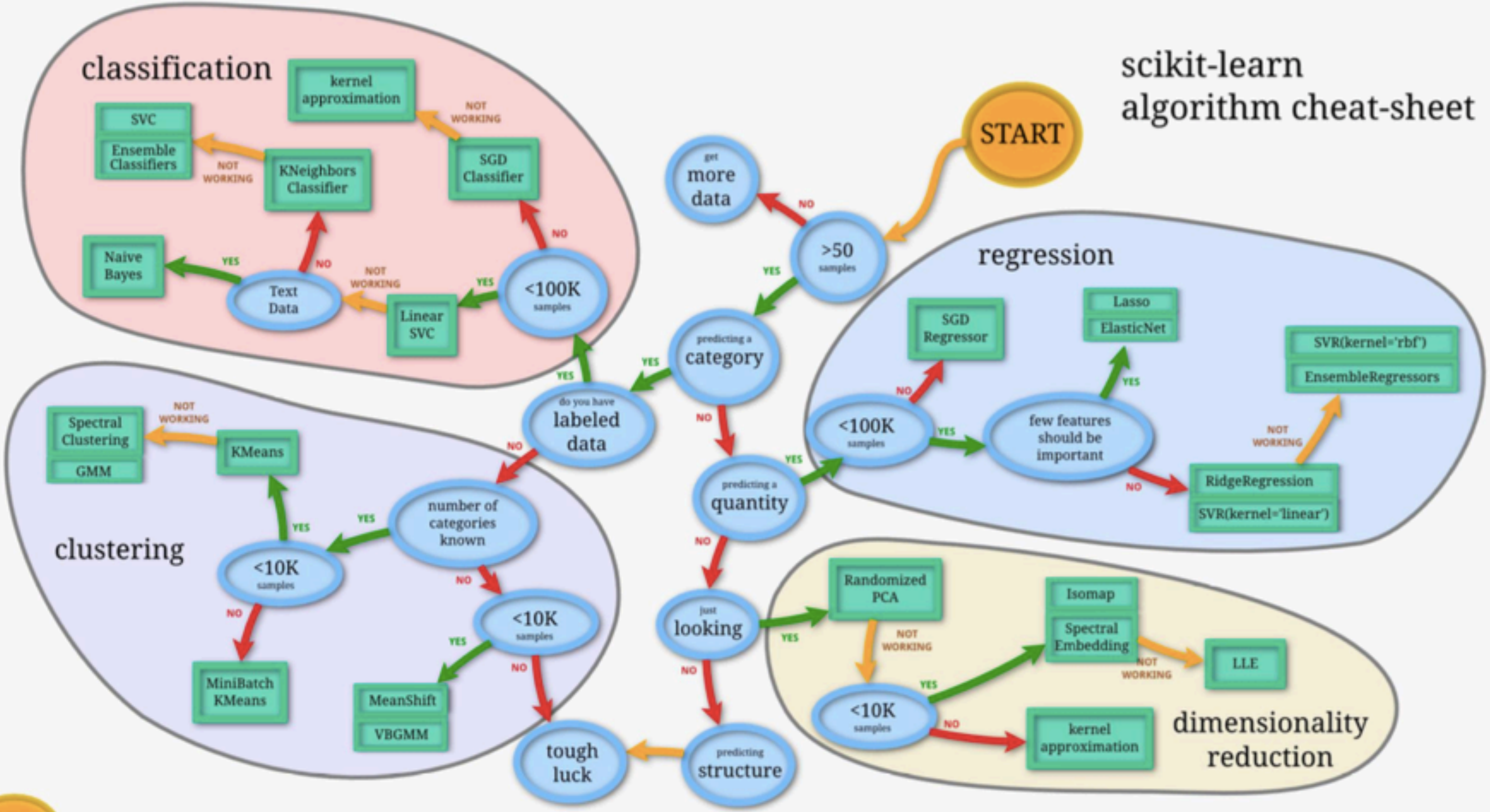
Wire Plot



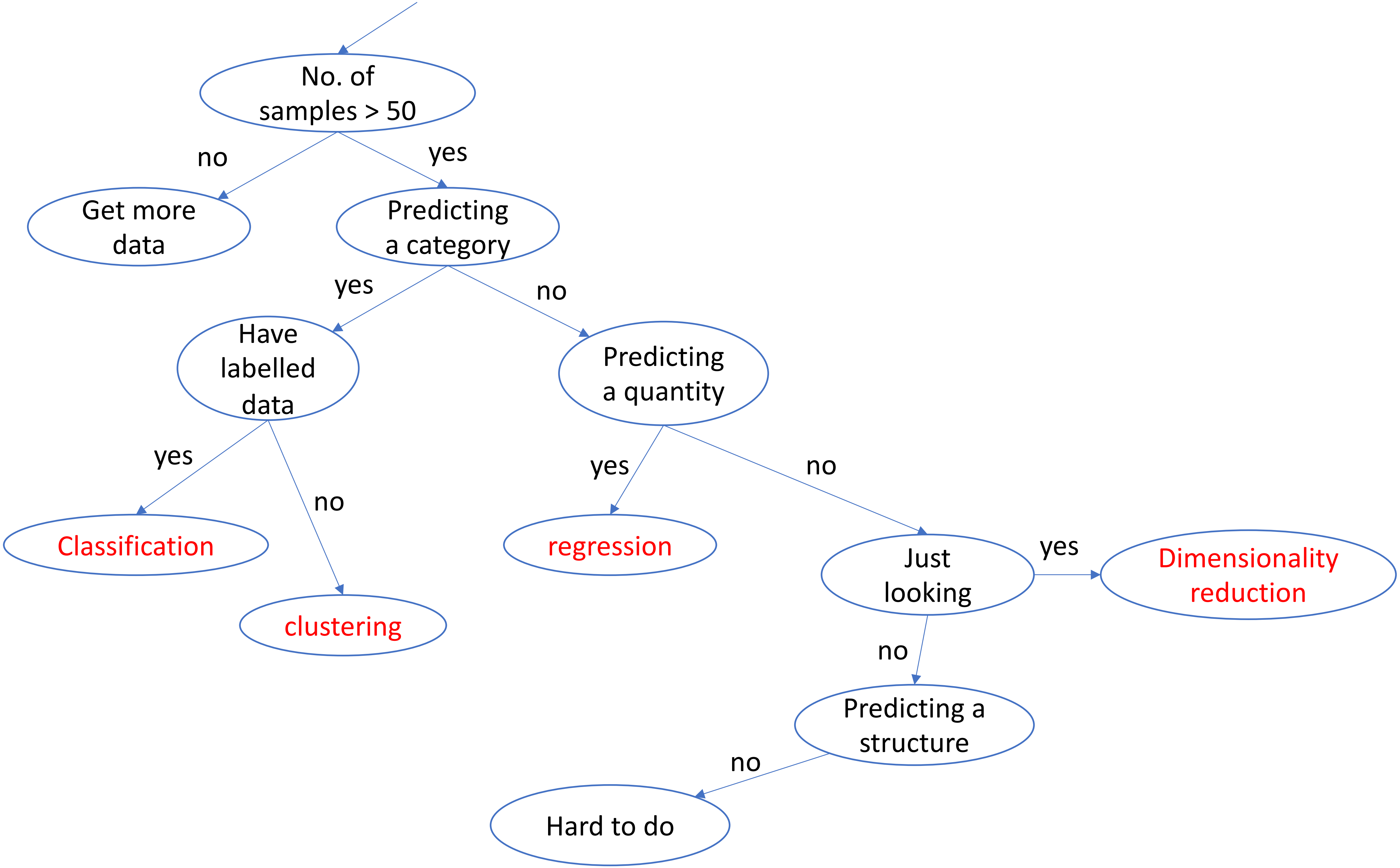
Up to now,

- Overview of Machine Learning
- Recap: Probability theory
- Recap: Linear Algebra
- Scientific Python

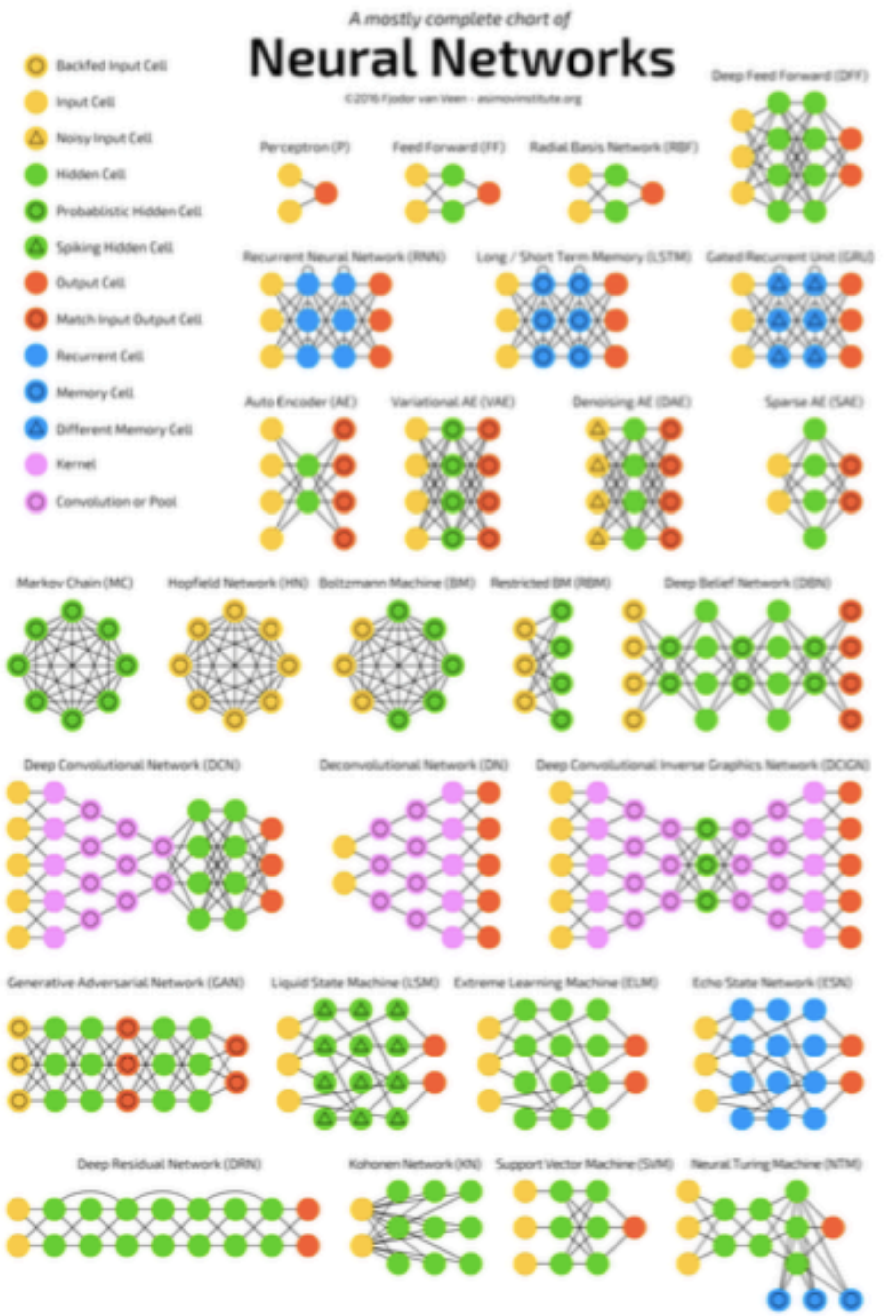
Now, are we up for real machine learning algorithm?



Note: only a subset of ML methods
Figure from scikit-learn.org


No. of samples > 50
no
yes
Get more data
Predicting a category
yes
no
Have labelled data
Predicting a quantity
yes
no
Classification
clustering
yes
no
regression
Just looking
yes
Dimensionality reduction
no
Predicting a structure
no
Hard to do

Even a subarea has its own collection



Topics

- the decision tree representation
- the standard top-down approach to learning a tree
- Occam’s razor
- entropy and information gain
- types of decision-tree splits

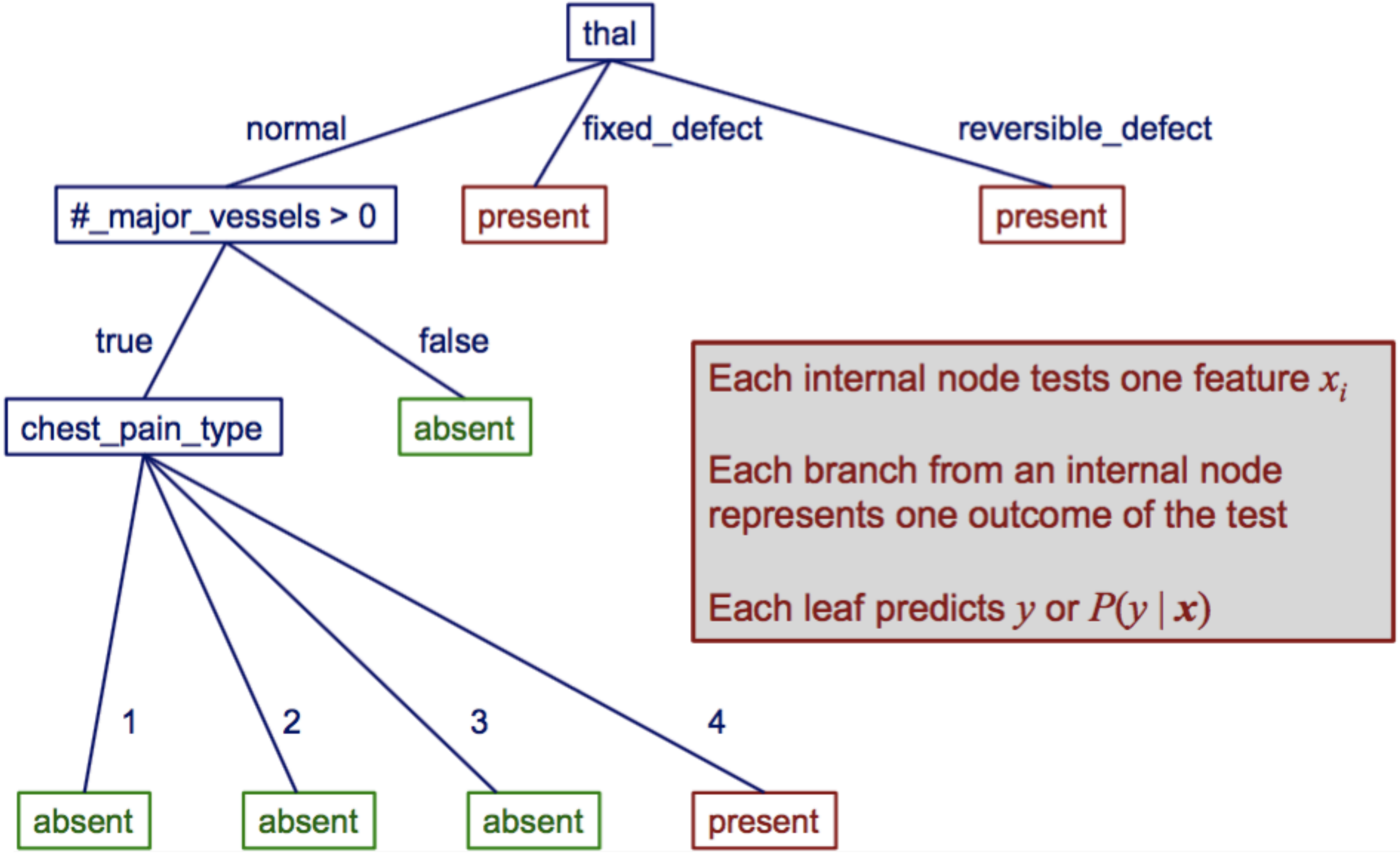
Recall: A learned decision tree

```
odor = a: e (400.0)
odor = c: p (192.0)
odor = f: p (2160.0)
odor = l: e (400.0)
odor = m: p (36.0)
odor = n
|   spore-print-color = b: e (48.0)
|   spore-print-color = h: e (48.0)
|   spore-print-color = k: e (1296.0)
|   spore-print-color = n: e (1344.0)
|   spore-print-color = o: e (48.0)
|   spore-print-color = r: p (72.0)
|   spore-print-color = u: e (0.0)
|   spore-print-color = w
|   |   gill-size = b: e (528.0)
|   |   gill-size = n
|   |   |   gill-spacing = c: p (32.0)
|   |   |   gill-spacing = d: e (0.0)
|   |   |   gill-spacing = w
|   |   |   |   population = a: e (0.0)
|   |   |   |   population = c: p (16.0)
|   |   |   |   population = n: e (0.0)
|   |   |   |   population = s: e (0.0)
|   |   |   |   population = v: e (48.0)
|   |   |   |   population = y: e (0.0)
|   spore-print-color = y: e (48.0)
odor = p: p (256.0)
odor = s: p (576.0)
odor = y: p (576.0)
```

if odor=almond, predict edible

if odor=none ∧
spore-print-color=white ∧
gill-size=narrow ∧
gill-spacing=crowded,
predict poisonous

A decision tree to predict heart disease



Decision tree exercise

- Suppose X_1 ... X_5 are Boolean features, and Y is also Boolean
- How would you represent the following with decision trees?

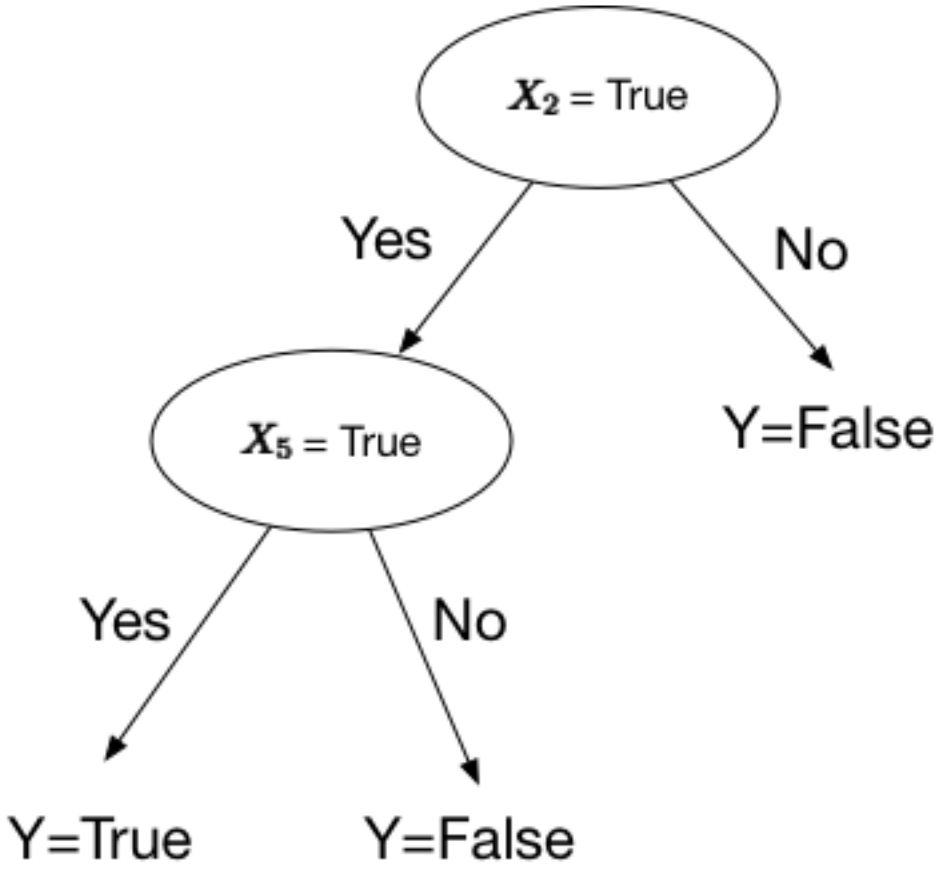
$$Y = X_2 X_5 \quad (\text{i.e.}, Y = X_2 \wedge X_5)$$

$$Y = X_2 \vee X_5$$

$$Y = X_2 X_5 \vee X_3 \neg X_1$$

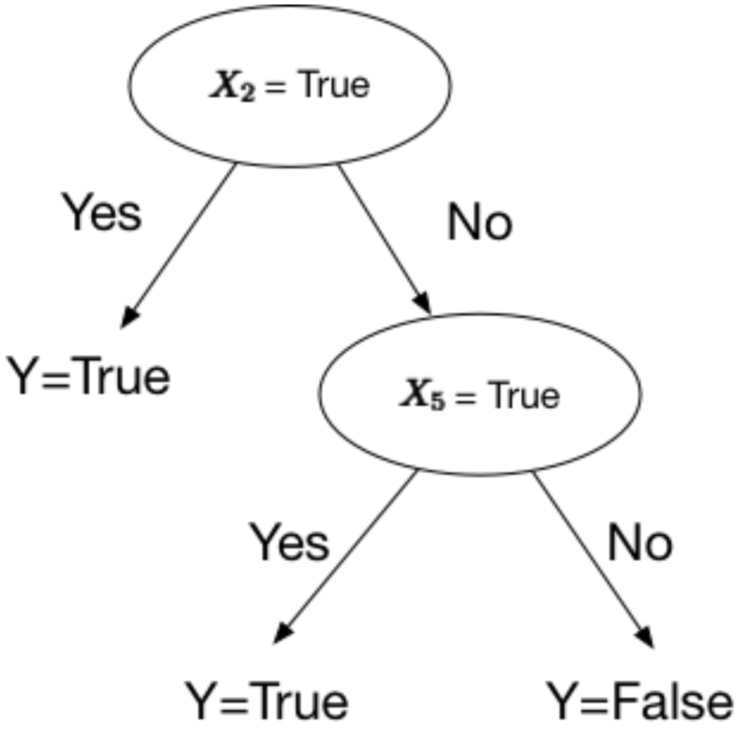
Decision tree exercise

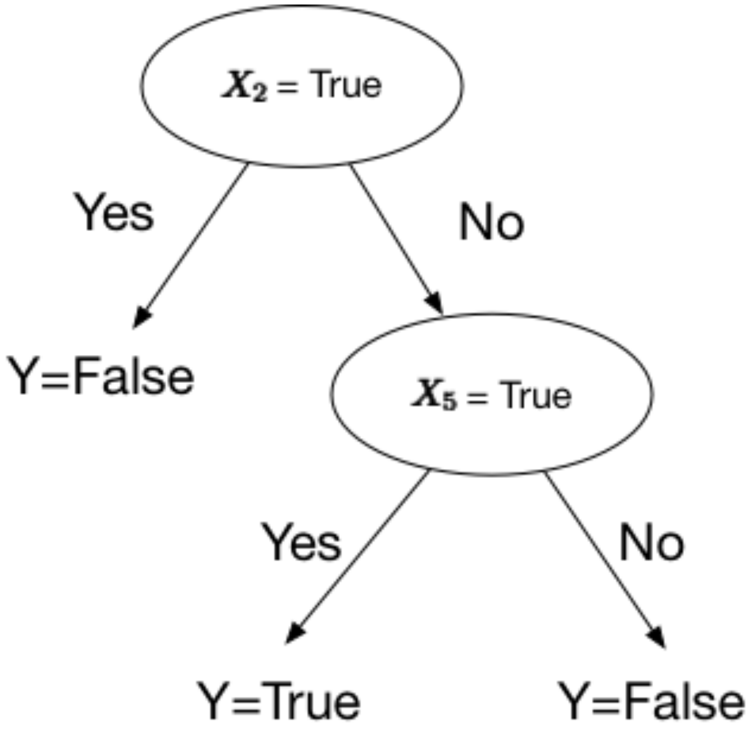
$$Y = X_2 X_5$$



Decision tree exercise

$$Y = X_2 \vee X_5$$





Wrong!

Decision tree exercise

$$Y = X_2 X_5 \vee X_3 \neg X_1$$