Machine Learning

Exemples avec Scikit-Learn

K-Means

```
from sklearn.cluster import KMeans
import numpy as np

X = np.array([[1, 2], [1, 4], [1, 0], [4, 2], [4, 4], [4, 0]])
kmeans = KMeans(n_clusters=2, random_state=0).fit(X)
lab = kmeans.labels_
pred = kmeans.predict([[0, 0], [4, 4]])
print(lab)
print(pred)
```

Pour les K-means regarder :

http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Le plus proche voisin

```
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn import neighbors, datasets

n_neighbors = 1

# import some data to play with
iris = datasets.load_iris()
# we only take the first two features.
X = iris.data[:, :2]
y = iris.target

# Create color maps
cmap_bold = ListedColormap(['#FF0000', '#00FF00', '#0000FF'])
clf = neighbors.KNeighborsClassifier(n_neighbors, weights='distance')
clf.fit(X, y)

# Plot the training points
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap_bold, edgecolor='k',
s=20)
plt.show()
```

Pour le modèle du plus proche voisin :

http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Arbres de Décision

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion = "gini", max_depth=4, min_samples_split=3)
model.fit(X,y)
predictions = model.predict(Xtest)
```

Pour les arbres de décision :

http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Perceptron Multi Couches

```
from sklearn.neural_network import MLPClassifier
from sklearn import datasets

# import some data to play with
iris = datasets.load_iris()
X = iris.data
y = iris.target

clf = MLPClassifier(solver='sgd', activation='logistic',
…                    max_iter=1000, hidden_layer_sizes=5,
…                    verbose='true', learning_rate_init=0.1)
clf.fit(X, y)
print(clf.score(X, y, sample_weight=None))
```

Source pour le Perceptron multi-couches :
http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Support Vector Machines

```
from sklearn.svm import SVC
model = SVC(C=1, kernel = 'linear')
model.fit(Xtrain, Ytrain)
predictions = model.predict(Xtest)
```

Random Forests

```
from sklearn.ensemble import RandomForestClassifier

model=RandomForestClassifier(n_estimators=1000,criterion="gin,
max_features="sqrt")

model.fit(X,y)
predictions = model.predict(Xtest)
print(model.feature_importances_)
```