# Machine Learning

## scikit-learn Session Lab

Jacob Montiel, Rodrigo Mello, Albert Bifet

September 26, 2018

# scikit-learn

- scikit-learn is the leading machine learning software in Python
- scikit-learn is a project started in Paris, Inria and Telecom ParisTech
- scilkit-learn is easy to use and extend

# scikit-learn Session Lab

- Install scikit-learn
  - Anaconda: `https://www.continuum.io/`
  - `http://scikit-learn.org/stable/install.html`
- Follow the scikit-learn Start Tutorial
  - `http://scikit-learn.org/stable/tutorial/basic/tutorial.html`

# scikit-learn Session Lab

- Start the python Shell or jupyter
- Import classes

```
>>> import numpy as np
>>> from sklearn import datasets
```

- Load and parse the data file.

```
>>> iris = datasets.load_iris()
>>> iris_X = iris.data
>>> iris_y = iris.target
>>> np.unique(iris_y)
array([0, 1, 2])
```

- Split the data into training and test sets

```
>>> # Split iris data in train and test data
>>> # A random permutation, to split the data randomly
>>> np.random.seed(0)
>>> indices = np.random.permutation(len(iris_X))
>>> iris_X_train = iris_X[indices[:-10]]
>>> iris_y_train = iris_y[indices[:-10]]
>>> iris_X_test  = iris_X[indices[-10:]]
>>> iris_y_test  = iris_y[indices[-10:]]
```

# scikit-learn Session Lab

- Train a *k*-nearest-neighbor model.

```
>>> # Create and fit a k−nearest−neighbor classifier
>>> from sklearn.neighbors import KNeighborsClassifier
>>> knn = KNeighborsClassifier()
>>> knn.fit(iris_X_train, iris_y_train)
KNeighborsClassifier(algorithm='auto', leaf_size=30,
                     metric='minkowski', metric_params=None,
                     n_jobs=1, n_neighbors=5, p=2,
                     weights='uniform')
```

- Evaluate model on test instances and compute test error

```
>>> from sklearn.metrics import accuracy_score
>>> knn.predict(iris_X_test)
array([1, 2, 1, 0, 0, 0, 2, 1, 2, 0])
>>> iris_y_test
array([1, 1, 1, 0, 0, 0, 2, 1, 2, 0])
>>> accuracy_score(iris_y_test, knn.predict(iris_X_test))
```

# scikit-learn Session Lab Assignment

1/ Write a jupyter notebook with the following tasks:

1. Write error of the classifier
2. What is the optimal parameter *k* of the *k*-nearest-neighbor classifier for this dataset?

# scikit-learn Session Lab Assignment

2/ Write a jupyter notebook with the following tasks:

With the iris dataset:

1. Use two other classifiers
2. Use cross-validation to evaluate the classifiers
3. Compare evaluation results of the three classifiers

# scikit-learn Session Lab

3/ In this part of the lab, we are going to create a classifier to use in scikit-learn

- Classifiers in scikit-learn has two main methods:
  - Build a model: `fit(self, X, Y)`
  - Make a prediction: `predict(self, X)`
- Classifiers are built using this template.

```python
class NewClassifier:

    def __init__(self):
        # TODO

    def fit(self, X, Y):
        # TODO
        return self

    def predict(self, X):
        # TODO
        return Y
```

# scikit-learn Session Lab

3/ Write a jupyter notebook with the following tasks:

1. Write a majority class classifier: a classifier that predicts the class label that is more frequent in the dataset
2. Use the majority class classifier to evaluate one dataset, and justify why the evaluation results using the new classifier are correct
3. OPTIONAL: create another classifier with higher performance than the majority class classifier