Machine Learning with scikit-learn

Andreas Mueller (NYU Center for Data Science, co-release manager scikit-learn)
What is scikit-learn?
Classification
Regression
Clustering
Semi-Supervised Learning
Feature Selection
Feature Extraction
Manifold Learning
Dimensionality Reduction
Kernel Approximation
Hyperparameter Optimization
Evaluation Metrics
Out-of-core learning

......
What is machine learning?
Hi Andy,

I just received an email from the first tutorial speaker, presenting right before you, saying he's ill and won't be able to make it.

I know you have already committed yourself to two presentations, but is there anyway you could increase your tutorial time slot, maybe just offer time to try out what you've taught? Otherwise I have to do some kind of modern dance interpretation of Python in data :-)

-Leah

Hi Andreas,

I am very interested in your Machine Learning background. I work for X Recruiting who have been engaged by Z, a worldwide leading supplier of Y. We are expanding the core engineering team and we are looking for really passionate engineers who want to create their own story and help millions of people.

Can we find a time for a call to chat for a few minutes about this?

Thanks
Hi Andy,

I just received an email from the first tutorial speaker, presenting right before you, saying he's ill and won't be able to make it.

I know you have already committed yourself to two presentations, but maybe you could increase your tutorial time slot, maybe just offer time to try out what you've taught? Otherwise I have to do some kind of modern dance interpretation of Python in data :-)

-Leah

---

Hi Andreas,

I am very interested in your Machine Learning background. I work for X Recruiting who have been engaged by Z, a worldwide leading supplier of Y. We are expanding the core engineering team and we are looking for really passionate engineers who want to create their own story and help millions of people.

Can we find a time for a call to chat for a few minutes about this?

Thanks
Doing Machine Learning With Scikit-Learn
Representing Data

\[ X = \begin{pmatrix}
  1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\
  6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\
  2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\
  1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\
  0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\
  5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\
  3.7 & 7.8 & 2.6 & 3.2 & 6.3
\end{pmatrix} \]
Representing Data

\[
X = \begin{pmatrix}
1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\
6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\
2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\
1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\
0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\
5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\
3.7 & 7.8 & 2.6 & 3.2 & 6.3
\end{pmatrix}
\]
Representing Data

\[ X = \begin{pmatrix}
1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\
6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\
2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\
1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\
0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\
5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\
3.7 & 7.8 & 2.6 & 3.2 & 6.3
\end{pmatrix} \]

one sample | one feature
Representing Data

\[ X = \begin{pmatrix} 1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\ 6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\ 2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\ 1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\ 0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\ 5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\ 3.7 & 7.8 & 2.6 & 3.2 & 6.3 \end{pmatrix} \]

\[ y = \begin{pmatrix} 1.6 \\ 2.7 \\ 4.4 \\ 0.5 \\ 0.2 \\ 5.6 \\ 6.7 \end{pmatrix} \]

one sample

one feature

outputs / labels
Training and Testing Data

\[ X = \begin{pmatrix}
1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\
6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\
2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\
1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\
0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\
5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\
3.7 & 7.8 & 2.6 & 3.2 & 6.3 \\
\end{pmatrix} \quad y = \begin{pmatrix}
1.6 \\
2.7 \\
4.4 \\
0.5 \\
0.2 \\
5.6 \\
6.7 \\
\end{pmatrix} \]
Training and Testing Data

\[
X = \begin{pmatrix}
1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\
6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\
2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\
1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\
0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\
\end{pmatrix}
\]

\[
y = \begin{pmatrix}
1.6 \\
2.7 \\
4.4 \\
0.5 \\
0.2 \\
\end{pmatrix}
\]

\[
X = \begin{pmatrix}
5.1 & 9.7 & 3.5 & 7.9 & 5.1 \\
3.7 & 7.8 & 2.6 & 3.2 & 6.3 \\
\end{pmatrix}
\]

training set

test set
Training and Testing Data

\[ X = \begin{pmatrix}
1.1 & 2.2 & 3.4 & 5.6 & 1.0 \\
6.7 & 0.5 & 0.4 & 2.6 & 1.6 \\
2.4 & 9.3 & 7.3 & 6.4 & 2.8 \\
1.5 & 0.0 & 4.3 & 8.3 & 3.4 \\
0.5 & 3.5 & 8.1 & 3.6 & 4.6 \\
\end{pmatrix} \]

\[ y = \begin{pmatrix}
1.6 \\
2.7 \\
4.4 \\
0.5 \\
0.2 \\
\end{pmatrix} \]

from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)
Supervised Machine Learning

Training Data

Training Labels

Model
Supervised Machine Learning

- Training Data
- Training Labels
- Test Data
- Model
- Prediction
Supervised Machine Learning

Training Data

Training Labels

Test Data

Test Labels

Model

Prediction

Evaluation
Supervised Machine Learning

Training Data → Training Labels → Model → Prediction → Evaluation → Test Labels

Training

Generalization
clf = RandomForestClassifier()

clf.fit(X_train, y_train)
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

clf.score(X_test, y_test)
Unsupervised Machine Learning

Training Data → Model
Unsupervised Machine Learning

- Training Data
- Test Data
- Model
- New View
Unsupervised Transformations

```python
pca = PCA()
pca.fit(X_train)
X_new = pca.transform(X_test)
```
# Basic API

```python
estimator.fit(X, [y])
estimator.predict
estimator.transform
```

<table>
<thead>
<tr>
<th>Classification</th>
<th>Preprocessing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>Dimensionality reduction</td>
</tr>
<tr>
<td>Clustering</td>
<td>Feature selection</td>
</tr>
<tr>
<td></td>
<td>Feature extraction</td>
</tr>
</tbody>
</table>
Sample application: Sentiment Analysis
Review:
One of the worst movies I've ever rented. Sorry it had one of my favorite actors on it (Travolta) in a nonsense role. In fact, anything made sense in this movie.

Who can say there was true love between Eddy and Maureen? Don't you remember the beginning of the movie?

Is she so lovely? Ask her daughters. I don't think so.

Label: negative

Training data: 12500 positive, 12500 negative
Bag Of Word Representations

CountVectorizer / TfidfVectorizer
Bag Of Word Representations

CountVectorizer / TfidfVectorizer

“This is how you get ants.”
Bag Of Word Representations

CountVectorizer / TfidfVectorizer

“This is how you get ants.”

tokenizer

['this', 'is', 'how', 'you', 'get', 'ants']
Bag Of Word Representations

CountVectorizer / TfidfVectorizer

“This is how you get ants.”

tokenizer

['this', 'is', 'how', 'you', 'get', 'ants']

Build a vocabulary over all documents

['aardvark', 'amsterdam', 'ants', ... 'you', 'your', 'zyxst']
Bag Of Word Representations

CountVectorizer / TfidfVectorizer

“This is how you get ants.”

tokenizer

['this', 'is', 'how', 'you', 'get', 'ants']

Build a vocabulary over all documents

['aardvak', 'amsterdam', 'ants', ... 'you', 'your', 'zyxst']

Sparse matrix encoding

aardvak  ants  get  you  zyxst
[0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ...., 0]
Implementation and Results

text_pipe = make_pipeline(CountVectorizer(), LinearSVC())
clf.fit(X_train, y_train)
clf.score(X_test, y_test)

> 0.85
text_pipe = make_pipeline(CountVectorizer(), LinearSVC())
clf.fit(X_train, y_train)
clf.score(X_test, y_test)

> 0.85
Model Complexity
Overfitting and Underfitting

Model complexity

Accuracy

Training
Overfitting and Underfitting

Accuracy vs. Model complexity

- Training curve
- Generalization curve
Overfitting and Underfitting

Model complexity

Accuracy

Training

Sweet spot

Generalization

Underfitting

Overfitting
Model Complexity Examples
Linear SVM

\[ \hat{y} = \text{sign}(w_0 + \sum_i w_i x_i) \]
Linear SVM

\[ \hat{y} = \text{sign}(w_0 + \sum_i w_i x_i) \]
(RBF) Kernel SVM

\[ \hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(x^{(j)}, x)) \]
(RBF) Kernel SVM

\[ \hat{y} = \text{sign}(\alpha_0 + \sum_{j} \alpha_j y_j k(x^{(j)}, x)) \]

\[ k(x, x') = \exp(-\gamma \|x - x'\|^2) \]
(RBF) Kernel SVM

\[ \hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(x^{(j)}, x)) \]

\[ k(x, x') = \exp(-\gamma ||x - x'||^2) \]
(RBF) Kernel SVM

\[
\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(x^{(j)}, x))
\]

\[
k(x, x') = \exp(-\gamma \|x - x'\|^2)
\]
Decision Trees
Decision Trees

max_depth = 1

X[0] <= 0.9963

value = [ 23, 6.]

value = [ 2, 19.]
Decision Trees

max_depth = 2

X[0] <= 0.9963

X[1] <= 1.5284

value = [23, 4]

X[1] <= -1.2058

value = [0, 2]

value = [1, 0]

value = [1, 19]
Decision Trees

max_depth = 3

- X[0] <= 0.9963
  - X[1] <= 1.5284
  - X[1] <= 0.1055
    - value = [11, 0]
  - X[1] <= -1.2058
    - value = [1, 0]
  - X[1] <= 0.2131
    - value = [0, 16]
    - value = [12, 4]
    - value = [13, 1]
Decision Trees
Decision Trees

max_depth = 7

X[0] <= 0.9963
X[1] <= 1.5284
X[1] <= -1.2058

X[0] <= 0.1055
value = [0, 2]

X[1] <= 0.3755

X[1] <= 0.8215
X[0] <= 0.6284
value = [0, 3]
value = [1, 0]

value = [5, 0]

X[0] <= -0.1072
value = [0, 2]

value = [5, 0]

value = [0, 2]

X[1] <= 1.1913
value = [0, 2]

value = [1, 0]
Random Forests
Random Forests
Random Forests
Model Evaluation and Model Selection
| All Data
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Training data</td>
</tr>
</tbody>
</table>
All Data

Training data

Test data

Fold 1  Fold 2  Fold 3  Fold 4  Fold 5
All Data

Training data  Test data

Fold 1  Fold 2  Fold 3  Fold 4  Fold 5

Split 1

Fold 1  Fold 2  Fold 3  Fold 4  Fold 5
All Data

Training data

Test data

Fold 1  Fold 2  Fold 3  Fold 4  Fold 5

Split 1

Fold 1  Fold 2  Fold 3  Fold 4  Fold 5

Split 2

Fold 1  Fold 2  Fold 3  Fold 4  Fold 5
Cross-Validation

```python
from sklearn.cross_validation import cross_val_score

scores = cross_val_score(SVC(), X, y, cv=5)
print(scores)

>> [ 0.92  1.  1.  1.  1.  1.  ]
```
SVC(C=0.001, gamma=0.001)
SVC(C=0.001, gamma=0.001)  SVC(C=0.01, gamma=0.001)  SVC(C=0.1, gamma=0.001)  SVC(C=1, gamma=0.001)  SVC(C=10, gamma=0.001)
<table>
<thead>
<tr>
<th>SVC(C=0.001, gamma=0.001)</th>
<th>SVC(C=0.01, gamma=0.001)</th>
<th>SVC(C=0.1, gamma=0.001)</th>
<th>SVC(C=1, gamma=0.001)</th>
<th>SVC(C=10, gamma=0.001)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC(C=0.001, gamma=0.01)</td>
<td>SVC(C=0.01, gamma=0.01)</td>
<td>SVC(C=0.1, gamma=0.01)</td>
<td>SVC(C=1, gamma=0.01)</td>
<td>SVC(C=10, gamma=0.01)</td>
</tr>
</tbody>
</table>
SVC(C=0.001, gamma=0.001)  SVC(C=0.01, gamma=0.001)  SVC(C=0.1, gamma=0.001)  SVC(C=1, gamma=0.001)  SVC(C=10, gamma=0.001)
SVC(C=0.001, gamma=0.01)  SVC(C=0.01, gamma=0.01)  SVC(C=0.1, gamma=0.01)  SVC(C=1, gamma=0.01)  SVC(C=10, gamma=0.01)
SVC(C=0.001, gamma=0.1)  SVC(C=0.01, gamma=0.1)  SVC(C=0.1, gamma=0.1)  SVC(C=1, gamma=0.1)  SVC(C=10, gamma=0.1)
<table>
<thead>
<tr>
<th>SVC(C=0.001, gamma=0.001)</th>
<th>SVC(C=0.01, gamma=0.001)</th>
<th>SVC(C=0.1, gamma=0.001)</th>
<th>SVC(C=1, gamma=0.001)</th>
<th>SVC(C=10, gamma=0.001)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC(C=0.001, gamma=0.01)</td>
<td>SVC(C=0.01, gamma=0.01)</td>
<td>SVC(C=0.1, gamma=0.01)</td>
<td>SVC(C=1, gamma=0.01)</td>
<td>SVC(C=10, gamma=0.01)</td>
</tr>
<tr>
<td>SVC(C=0.001, gamma=0.1)</td>
<td>SVC(C=0.01, gamma=0.1)</td>
<td>SVC(C=0.1, gamma=0.1)</td>
<td>SVC(C=1, gamma=0.1)</td>
<td>SVC(C=10, gamma=0.1)</td>
</tr>
<tr>
<td>SVC(C=0.001, gamma=1)</td>
<td>SVC(C=0.01, gamma=1)</td>
<td>SVC(C=0.1, gamma=1)</td>
<td>SVC(C=1, gamma=1)</td>
<td>SVC(C=10, gamma=1)</td>
</tr>
<tr>
<td>SVC(C=0.001, gamma=10)</td>
<td>SVC(C=0.01, gamma=10)</td>
<td>SVC(C=0.1, gamma=10)</td>
<td>SVC(C=1, gamma=10)</td>
<td>SVC(C=10, gamma=10)</td>
</tr>
<tr>
<td>Split</td>
<td>Fold 1</td>
<td>Fold 2</td>
<td>Fold 3</td>
<td>Fold 4</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>Fold 1</td>
<td>Fold 2</td>
<td>Fold 3</td>
<td>Fold 4</td>
</tr>
<tr>
<td>2</td>
<td>Fold 1</td>
<td>Fold 2</td>
<td>Fold 3</td>
<td>Fold 4</td>
</tr>
<tr>
<td>3</td>
<td>Fold 1</td>
<td>Fold 2</td>
<td>Fold 3</td>
<td>Fold 4</td>
</tr>
<tr>
<td>4</td>
<td>Fold 1</td>
<td>Fold 2</td>
<td>Fold 3</td>
<td>Fold 4</td>
</tr>
<tr>
<td>5</td>
<td>Fold 1</td>
<td>Fold 2</td>
<td>Fold 3</td>
<td>Fold 4</td>
</tr>
</tbody>
</table>
All Data

Training data

Test data

Finding Parameters

Final evaluation

Test data
Cross-Validated Grid Search

```python
from sklearn.grid_search import GridSearchCV
from sklearn.cross_validation import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y)

param_grid = {'C': 10. ** np.arange(-3, 3),
               'gamma': 10. ** np.arange(-3, 3)}
grid = GridSearchCV(SVC(), param_grid=param_grid)
grid.fit(X_train, y_train)
grid.score(X_test, y_test)
```
Pipelines
Training Labels

Training Data

Model
Training Labels \rightarrow Model \rightarrow Training Data
Training Labels

Training Data

Feature Extraction

Scaling

Feature Selection

Model
Training Labels → Feature Extraction → Feature Selection → Model
Training Data → Scaling → Feature Selection → Model

Cross Validation and Parameter selection
Training Data

Training Labels

Feature Extraction

Scaling

Feature Selection

Model

Cross Validation and Parameter selection
Pipelines

```python
from sklearn.pipeline import make_pipeline

pipe = make_pipeline(StandardScaler(), SVC())
pipe.fit(X_train, y_train)
pipe.predict(X_test)
```
from sklearn.pipeline import make_pipeline

pipe = make_pipeline(StandardScaler(), SVC())
pipe.fit(X_train, y_train)
pipe.predict(X_test)
Combining Pipelines and Grid Search

Proper cross-validation

```
param_grid = {'svc__C': 10. ** np.arange(-3, 3),
              'svc__gamma': 10. ** np.arange(-3, 3)}

scaler_pipe = make_pipeline(StandardScaler(), SVC())
grid = GridSearchCV(scaler_pipe, param_grid=param_grid, cv=5)
grid.fit(X_train, y_train)
```
Combining Pipelines and Grid Search II

Searching over parameters of the preprocessing step

```python
param_grid = {'selectkbest__k': [1, 2, 3, 4],
              'svc__C': 10. ** np.arange(-3, 3),
              'svc__gamma': 10. ** np.arange(-3, 3)}

scaler_pipe = make_pipeline(SelectKBest(), SVC())
grid = GridSearchCV(scaler_pipe, param_grid=param_grid, cv=5)
grid.fit(X_train, y_train)
```
Do cross-validation over all steps jointly. Keep a separate test set until the very end.
Scoring Functions
GridSearchCV

cross_val_score

Default:
Accuracy (classification)
R2 (regression)
Scoring with imbalanced data

cross_val_score(SVC(), X_train, y_train)

>>> array([ 0.9,  0.9,  0.9])
Scoring with imbalanced data

cross_val_score(SVC(), X_train, y_train)

>>> array([ 0.9,  0.9,  0.9])

cross_val_score(DummyClassifier("most_frequent"), X_train, y_train)

>>> array([ 0.9,  0.9,  0.9])
Scoring with imbalanced data

cross_val_score(SVC(), X_train, y_train)
>>> array([ 0.9,  0.9,  0.9])

cross_val_score(DummyClassifier("most_frequent"), X_train, y_train)
>>> array([ 0.9,  0.9,  0.9])

cross_val_score(SVC(), X_train, y_train, scoring="roc_auc")
>>> array([ 1.0,  1.0,  1.0])
Scoring with imbalanced data

cross_val_score(SVC(), X_train, y_train)
>>> array([ 0.9,  0.9,  0.9])

cross_val_score(DummyClassifier("most_frequent"), X_train, y_train)
>>> array([ 0.9,  0.9,  0.9])

cross_val_score(SVC(), X_train, y_train, scoring="roc_auc")
>>> array([ 1.0,  1.0,  1.0])
Video Series
Advanced Machine Learning with scikit-learn

50% Off Coupon Code: AUTHHD
Video Series
Advanced Machine Learning with scikit-learn

50% Off Coupon Code: AUTHD
CDS is hiring Research Engineers

Work on your favorite data science open source project full time!
Thank you for your attention.

@t3kcit
@amueller
importamueller@gmail.com
http://amueller.github.io