Predictive modeling ~= machine learning

- Make predictions of outcome on new data
- Extract the structure of historical data
- Statistical tools to summarize the training data into a executable predictive model
- Alternative to hard-coded rules written by experts

type (category)	# rooms (int)	surface (float m2)	public trans (boolean)
Apartment	3	50	TRUE
House	5	254	FALSE
Duplex	4	68	TRUE
Apartment	2	32	TRUE

type (category)	# rooms (int)	surface (float m2)	public trans (boolean)	sold (float k€)
Apartment	3	50	TRUE	450
House	5	254	FALSE	430
Duplex	4	68	TRUE	712
Apartment	2	32	TRUE	234

features

target

samples (train)

type (category)	# rooms (int)	surface (float m2)	public trans (boolean)	sold (float k€)
Apartment	3	50	TRUE	450
House	5	254	FALSE	430
Duplex	4	68	TRUE	712
Apartment	2	32	TRUE	234

features

target

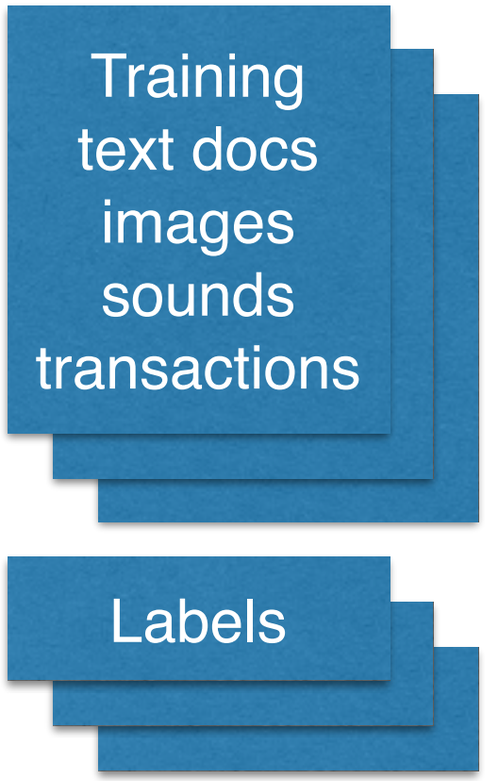
samples (train)

type (category)	# rooms (int)	surface (float m2)	public trans (boolean)	sold (float k€)
Apartment	3	50	TRUE	450
House	5	254	FALSE	430
Duplex	4	68	TRUE	712
Apartment	2	32	TRUE	234

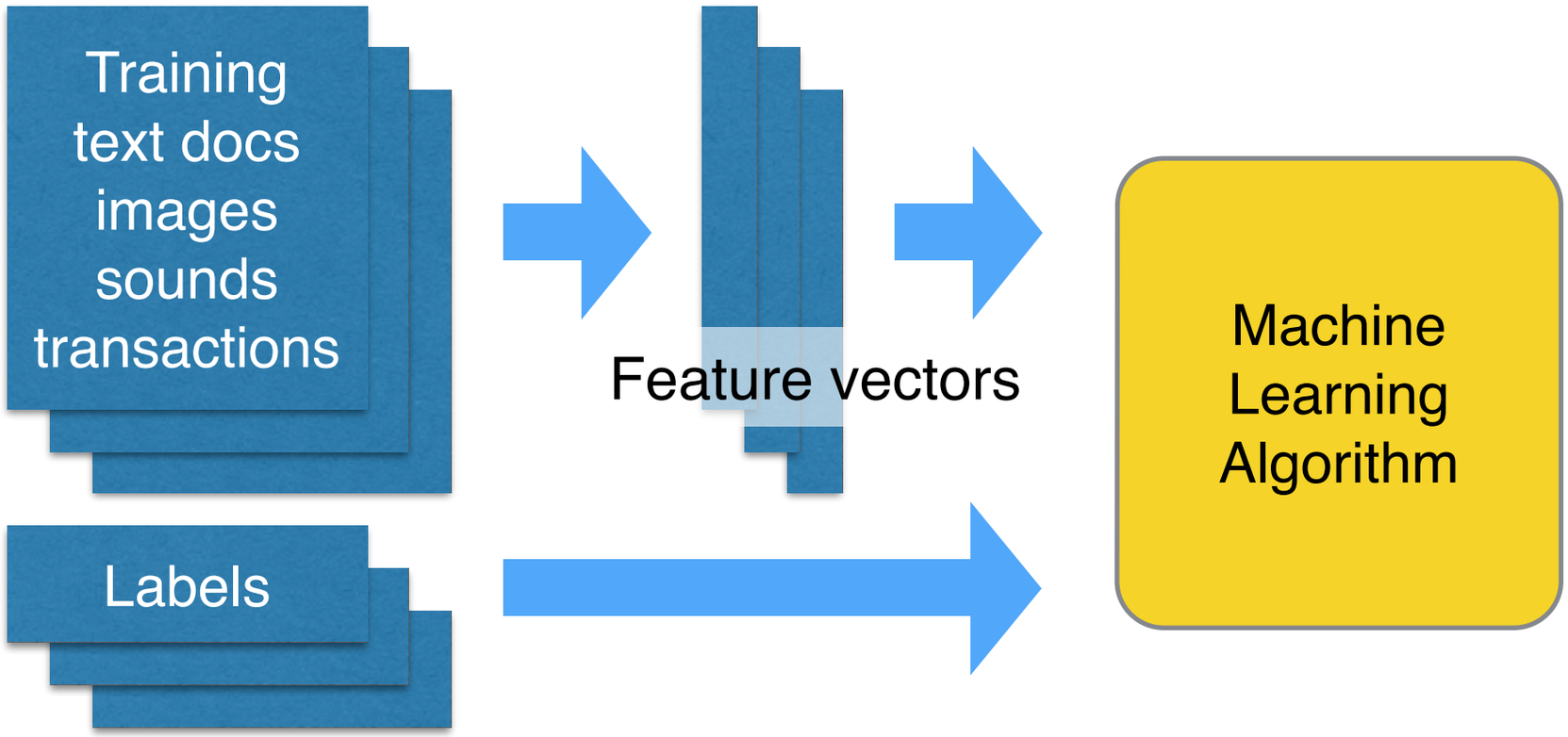
samples (test)

Apartment	2	33	TRUE	?
House	4	210	TRUE	?

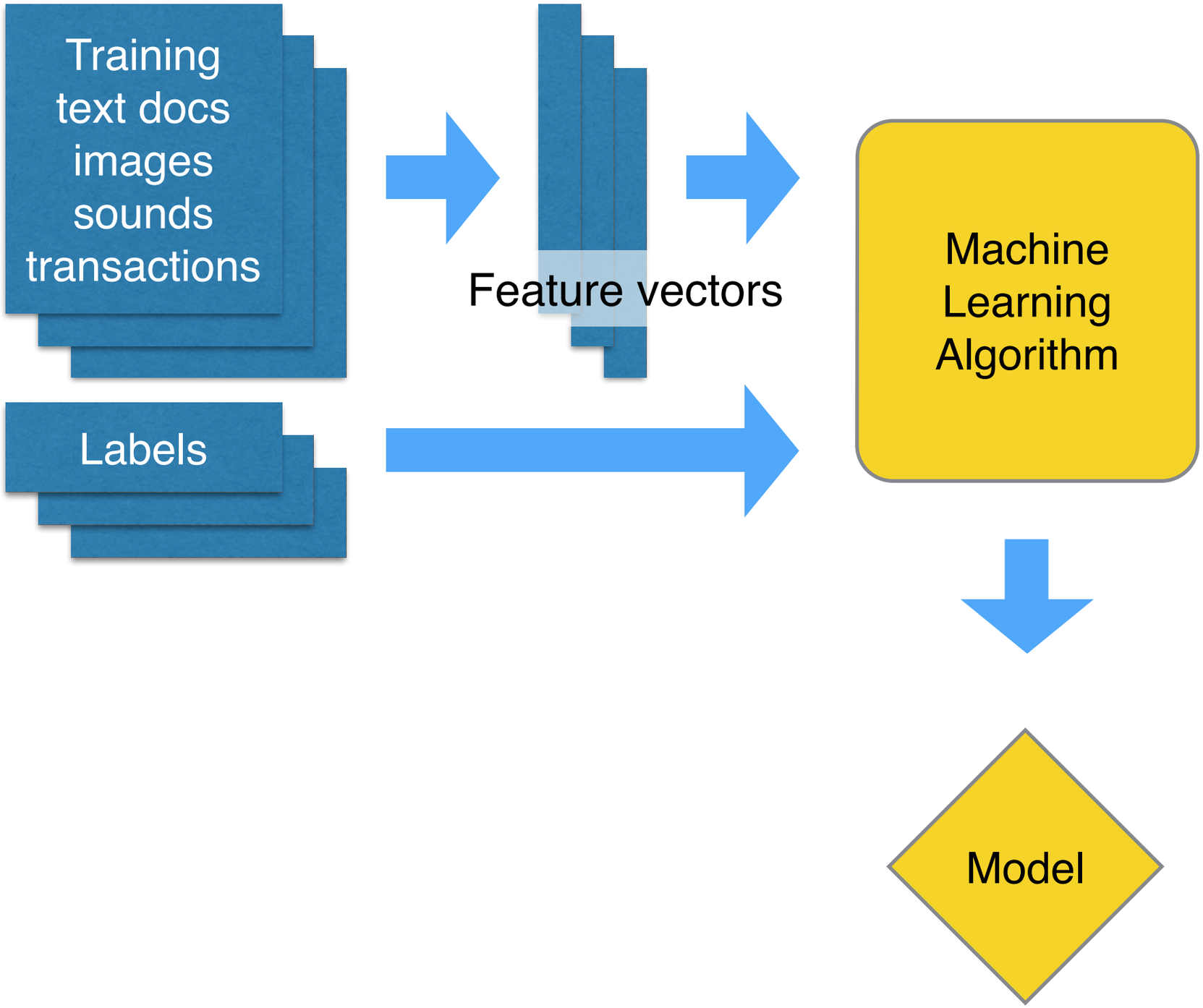
Predictive Modeling Data Flow



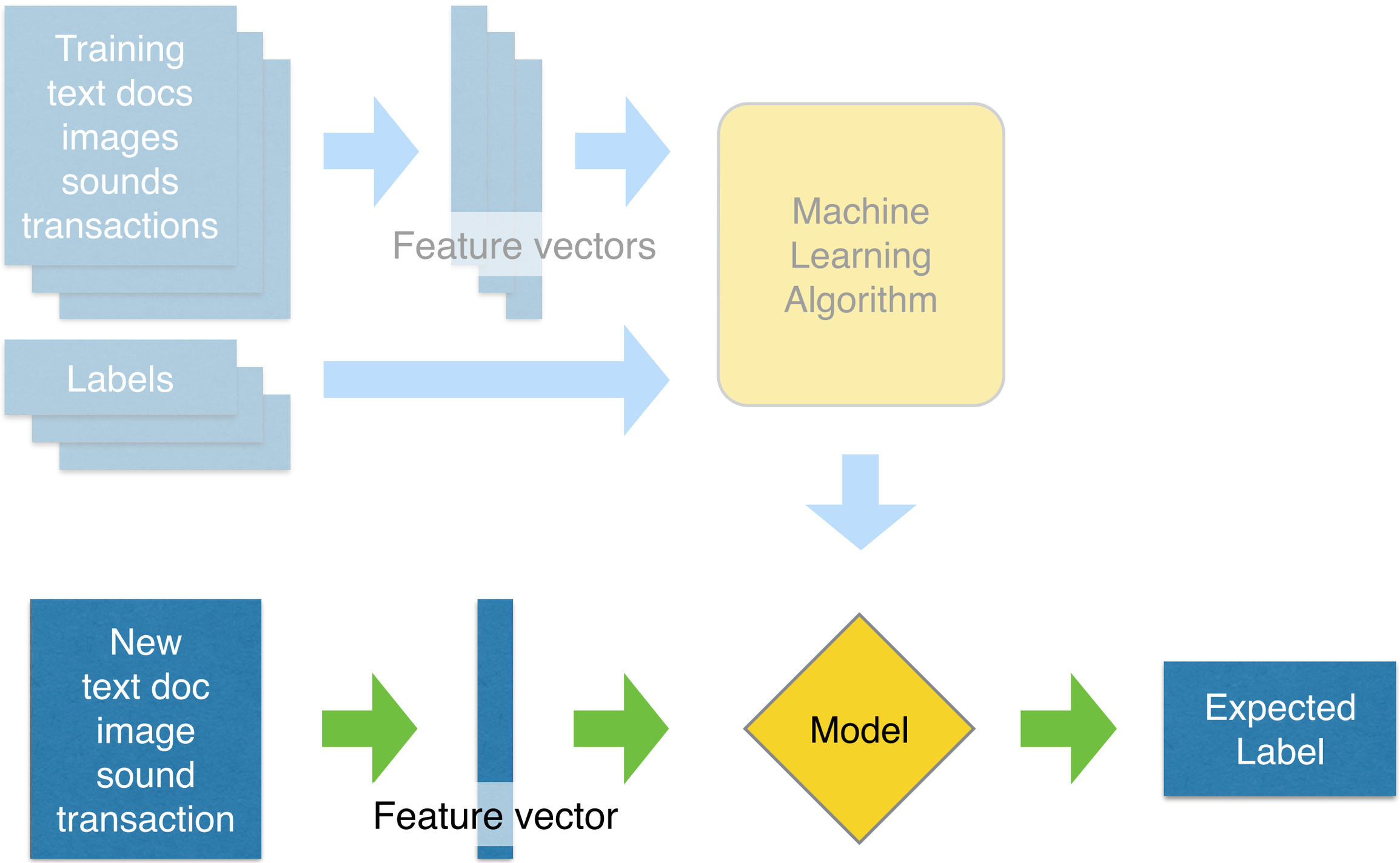
Predictive Modeling Data Flow



Predictive Modeling Data Flow



Predictive Modeling Data Flow



Predictive Modeling Data Flow

Applications in Business

- Forecast sales, customer churn, traffic, prices
- Predict CTR and optimal bid price for online ads
- Build computer vision systems for robots in the industry and agriculture
- Detect network anomalies, fraud and spams
- Recommend products, movies, music

- Library of Machine Learning algorithms
- Focus on established methods (e.g. ESL-II)
- Open Source (BSD)
- Simple **`fit`** / **`predict`** / **`transform`** API
- Python / NumPy / SciPy / Cython
- Model Assessment, Selection & Ensembles

Support Vector Machine

```
from sklearn.svm import SVC

model = SVC(kernel="rbf", C=1.0, gamma=1e-4)
model.fit(X_train, y_train)


y_predicted = model.predict(X_test)

from sklearn.metrics import f1_score
f1_score(y_test, y_predicted)
```

Random Forests

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=200)
model.fit(X_train, y_train)


y_predicted = model.predict(X_test)

from sklearn.metrics import f1_score
f1_score(y_test, y_predicted)
```

scikit-learn: machine learning in Python — scikit-learn 0.14 documentation

scikit-learn: machine learning i...

scikit-learn.org/stable/

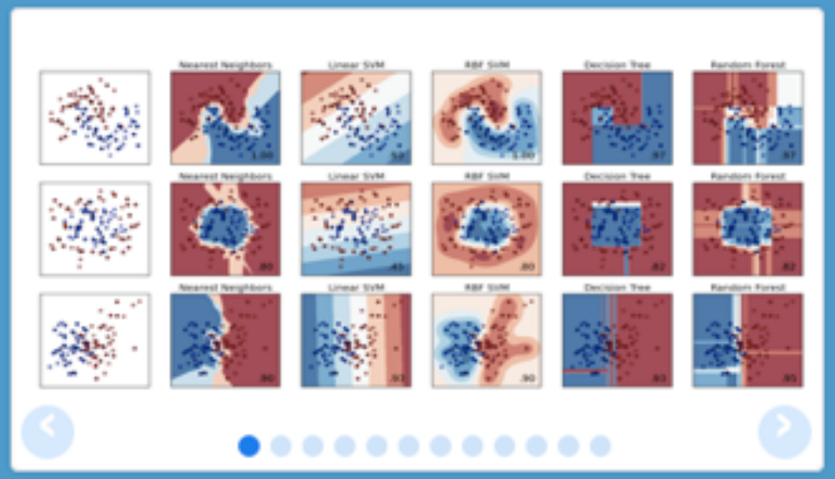
Google

scikit learn

Home | Installation | Documentation | Examples

Google™ Custom Search | Search

Fork me on GitHub



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*. — *Examples*

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*. — *Examples*