Introduction to scikit-learn

Predictive modeling in Python

Olivier Grisel

Slides: ogrisel.github.io/decks/2017_intro_sklearn

Agenda

Machine Learning refresher

Scikit-learn

Where do predictive models fit?

Predictive Modeling 101

Make predictions of outcome of repeated events

Extract the structure of historical records

Statistical tools to summarize the training data into an executable 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

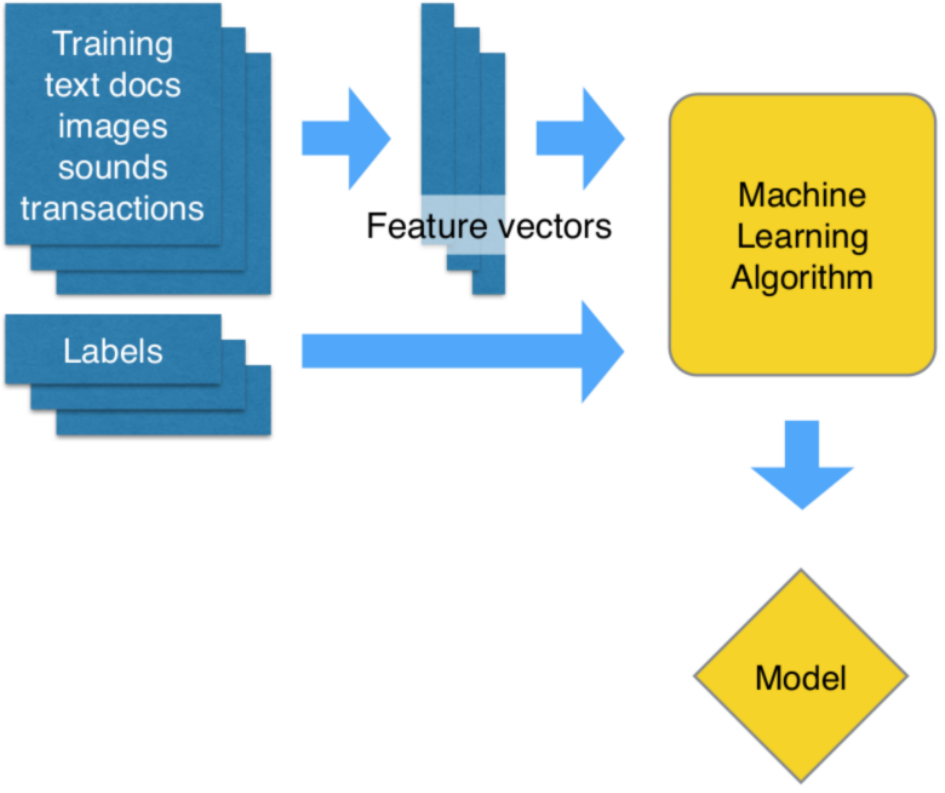
type (category)	# rooms (int)	surface (float m2)	public trans (boolean)	sold (float k€)
Apartment	3	50	TRUE	450
House	5	254	FALSE	430
Duplex	4	68	TRUE	712
Apartment	2	32	TRUE	234

features target

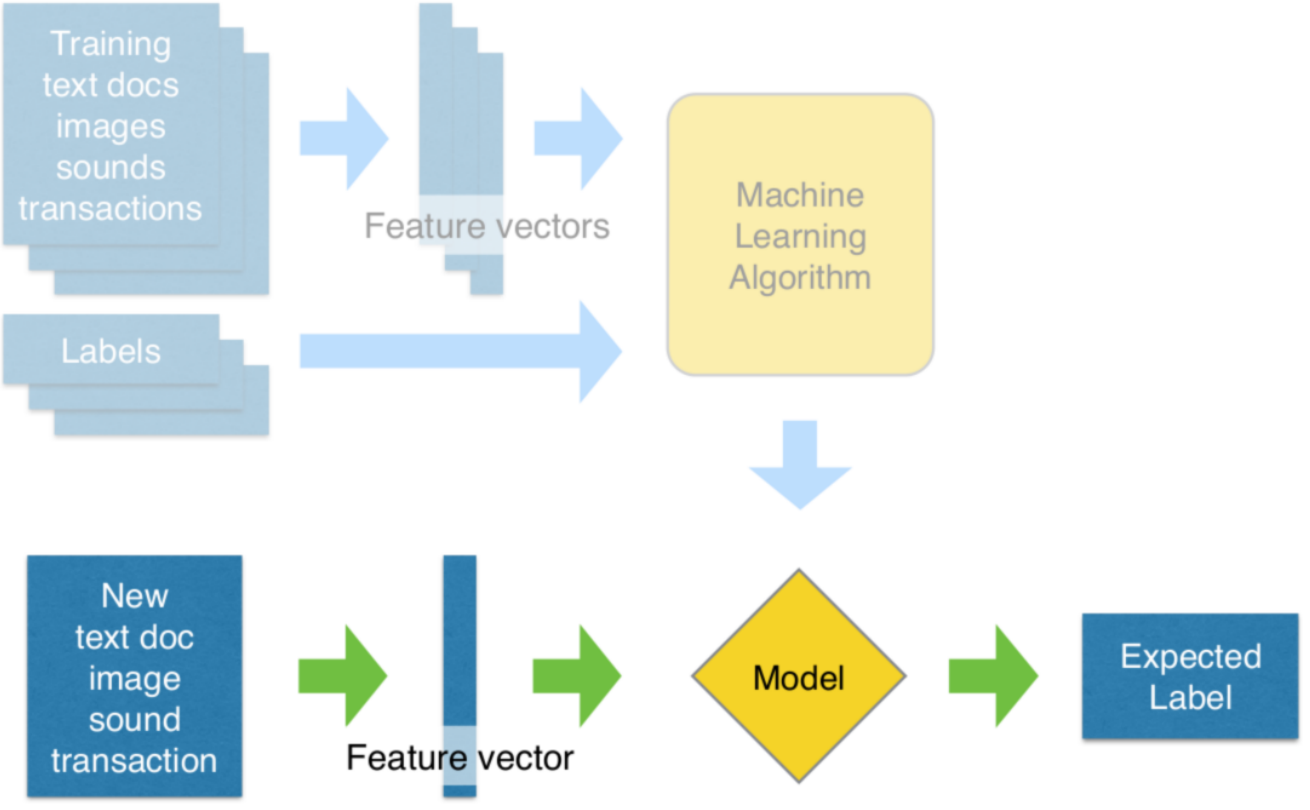
samples (train)

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

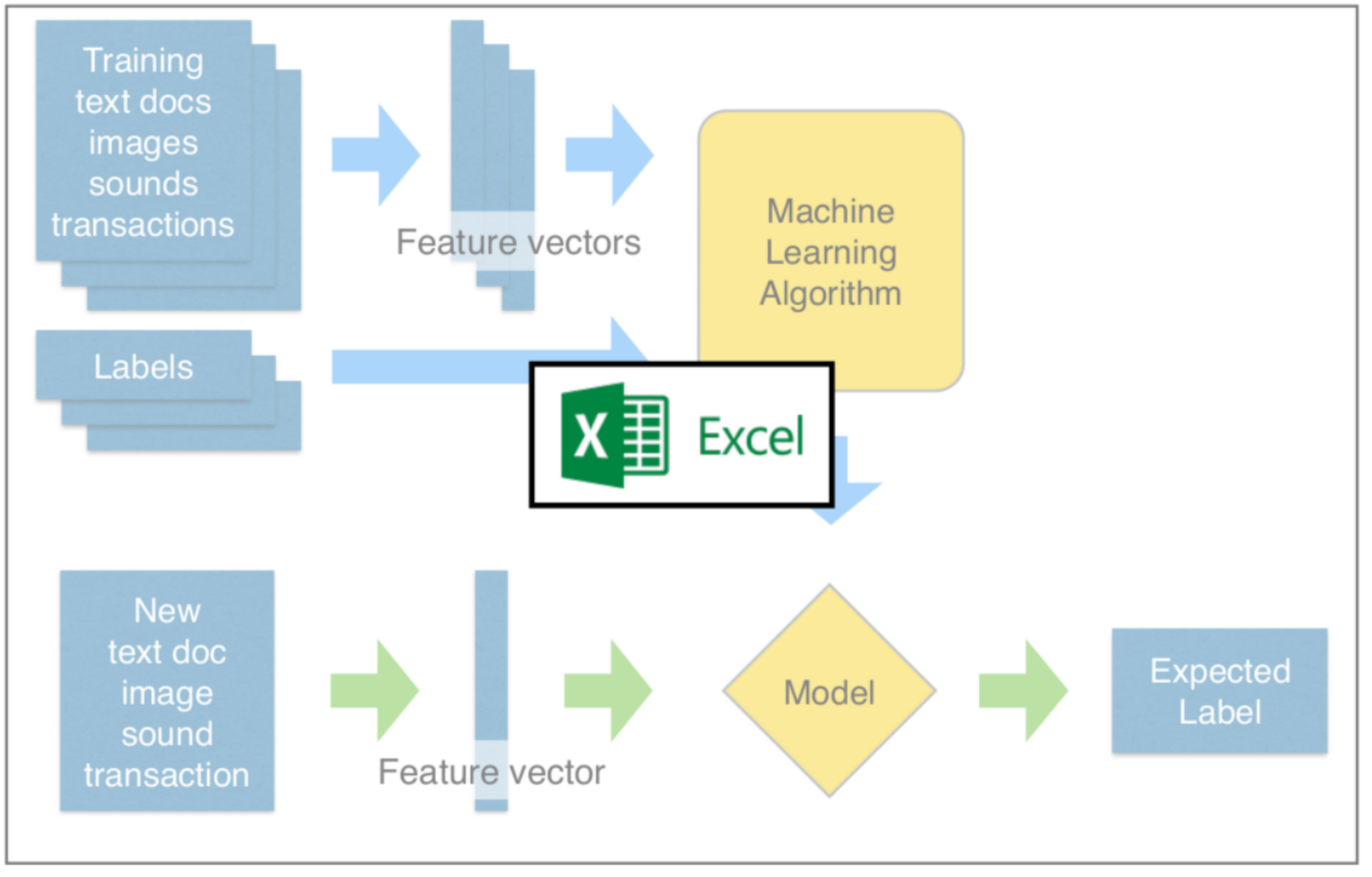
	features				target
	type (category)	**# rooms (int)**	**surface (float m2)**	**public trans (boolean)**	**sold (float k€)**
samples (train)	Apartment	3	50	TRUE	450
	House	5	254	FALSE	430
	Duplex	4	68	TRUE	712
	Apartment	2	32	TRUE	234
samples (test)	Apartment	2	33	TRUE	?
	House	4	210	TRUE	?



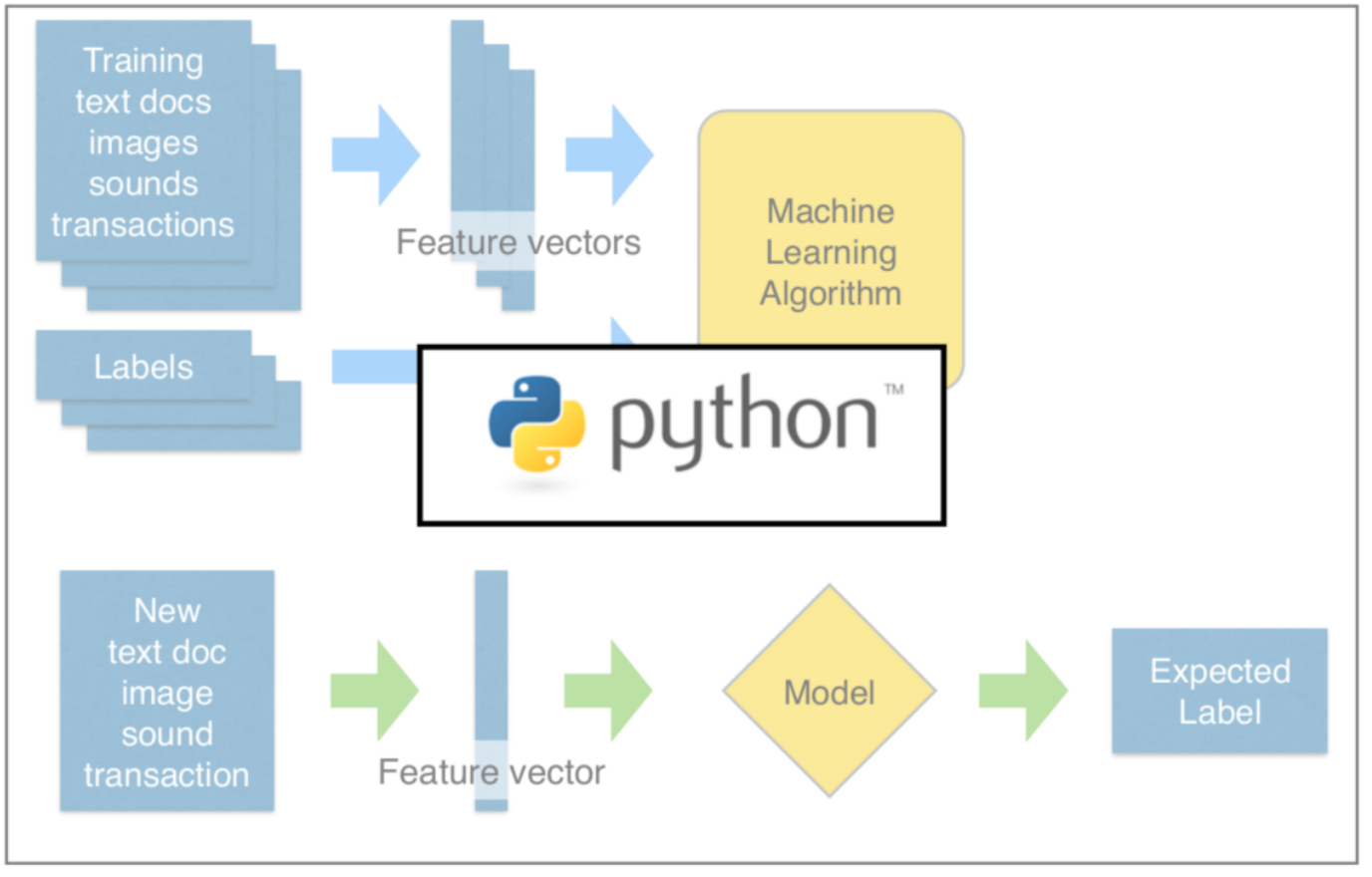
Predictive Modeling Data Flow



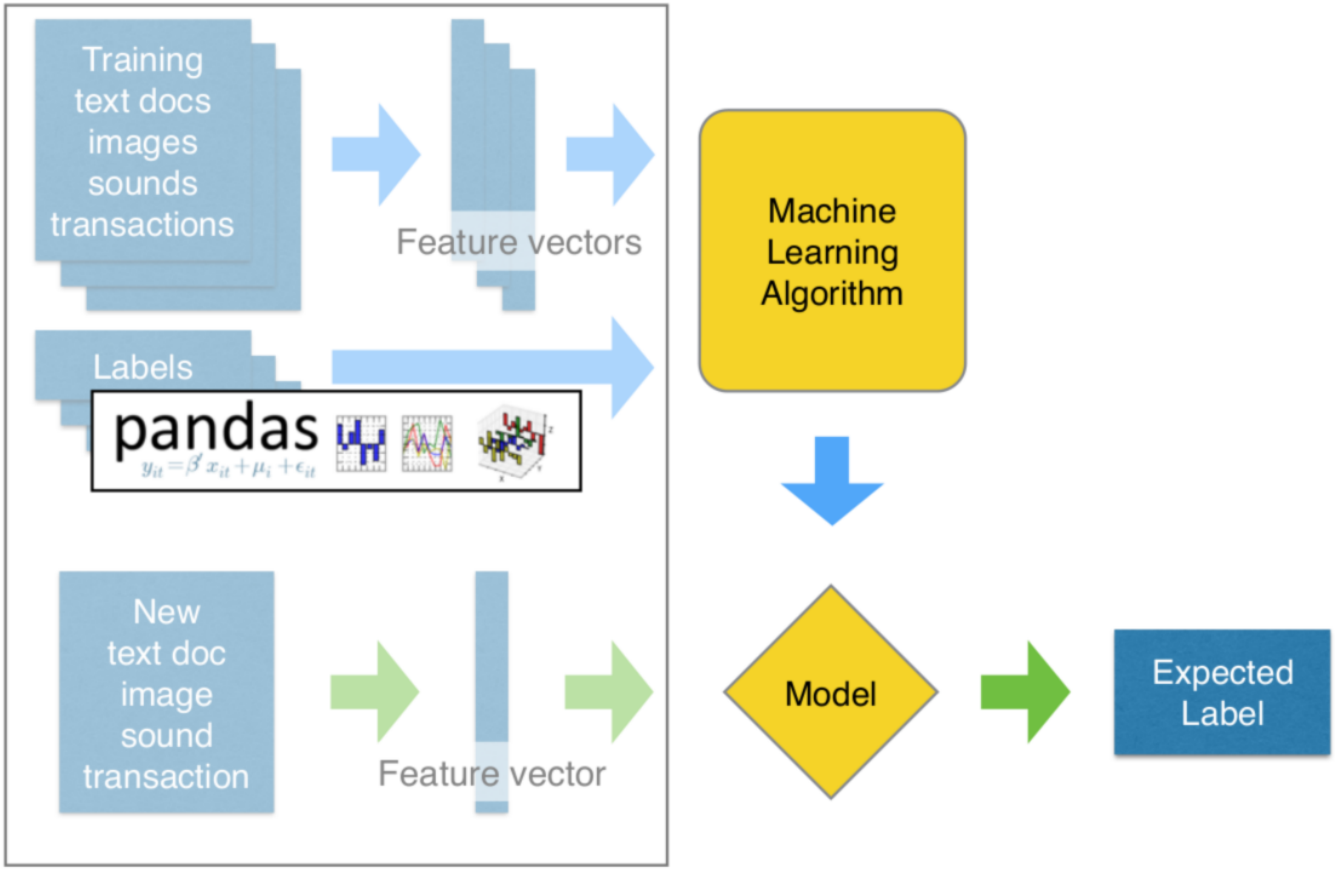
Predictive Modeling Data Flow


Training
text docs
images
sounds
transactions
Feature vectors
Machine
Learning
Algorithm
Labels
Excel
New
text doc
image
sound
transaction
Feature vector
Model
Expected
Label

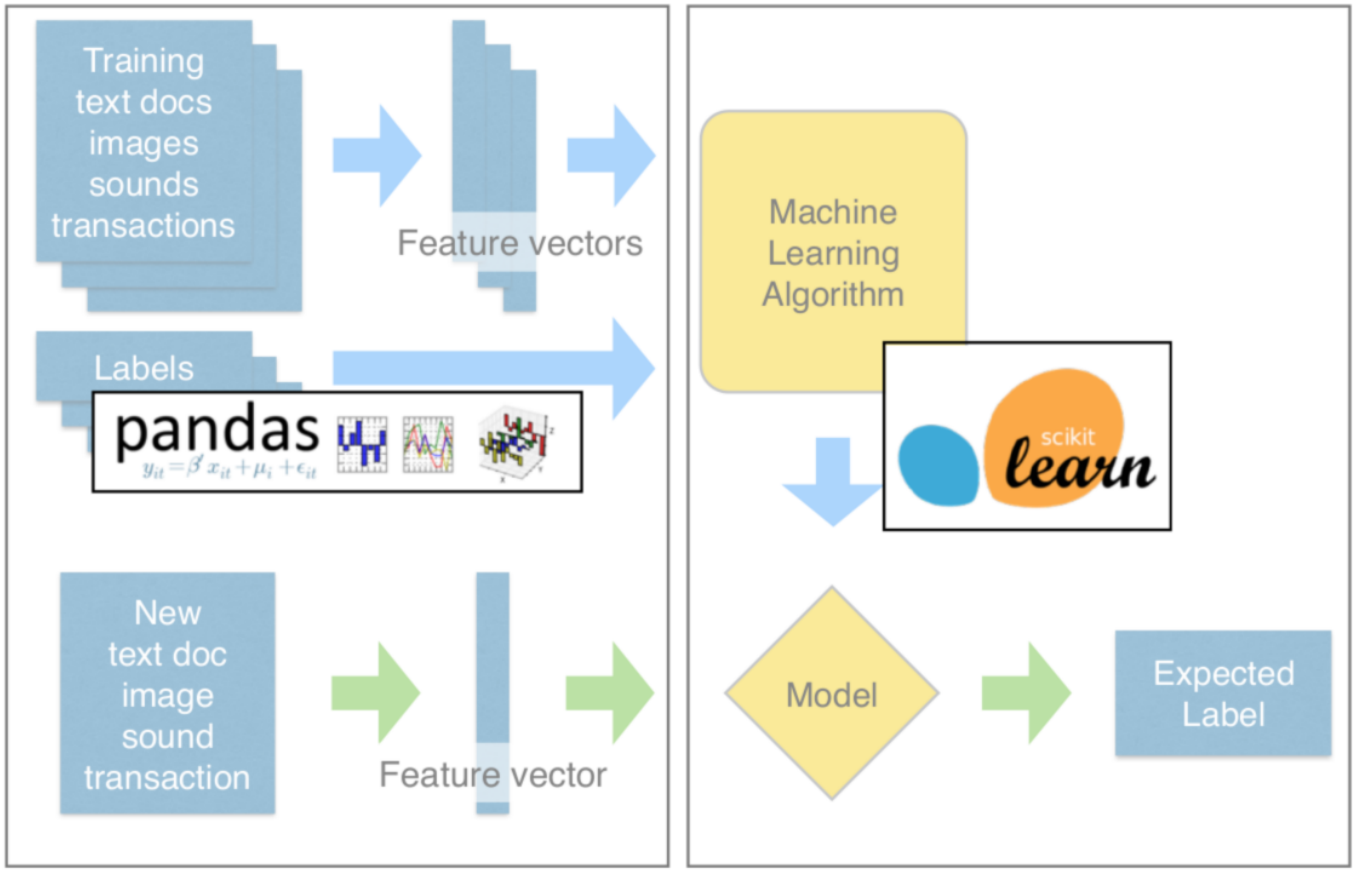
Small data



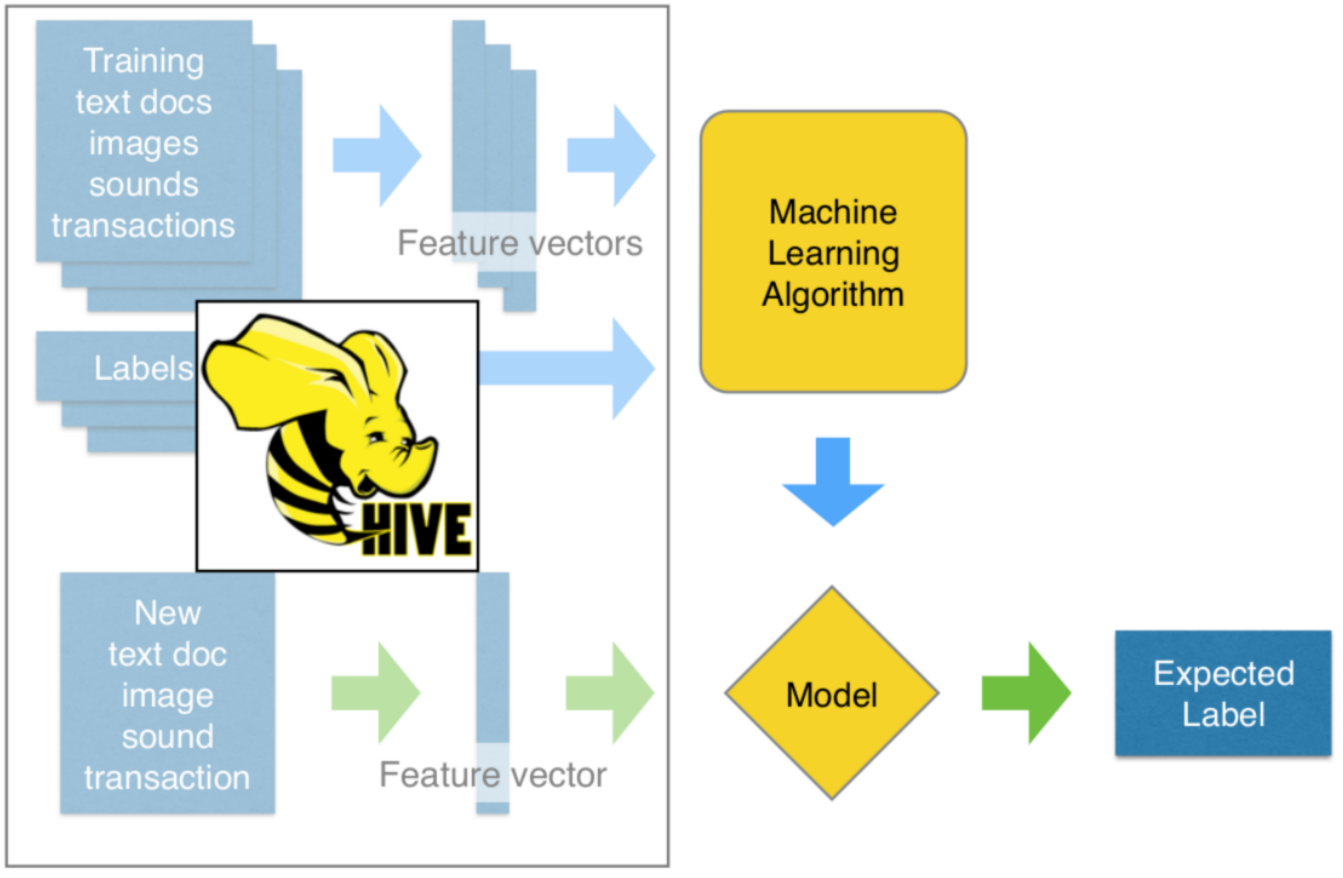
Small / Medium data



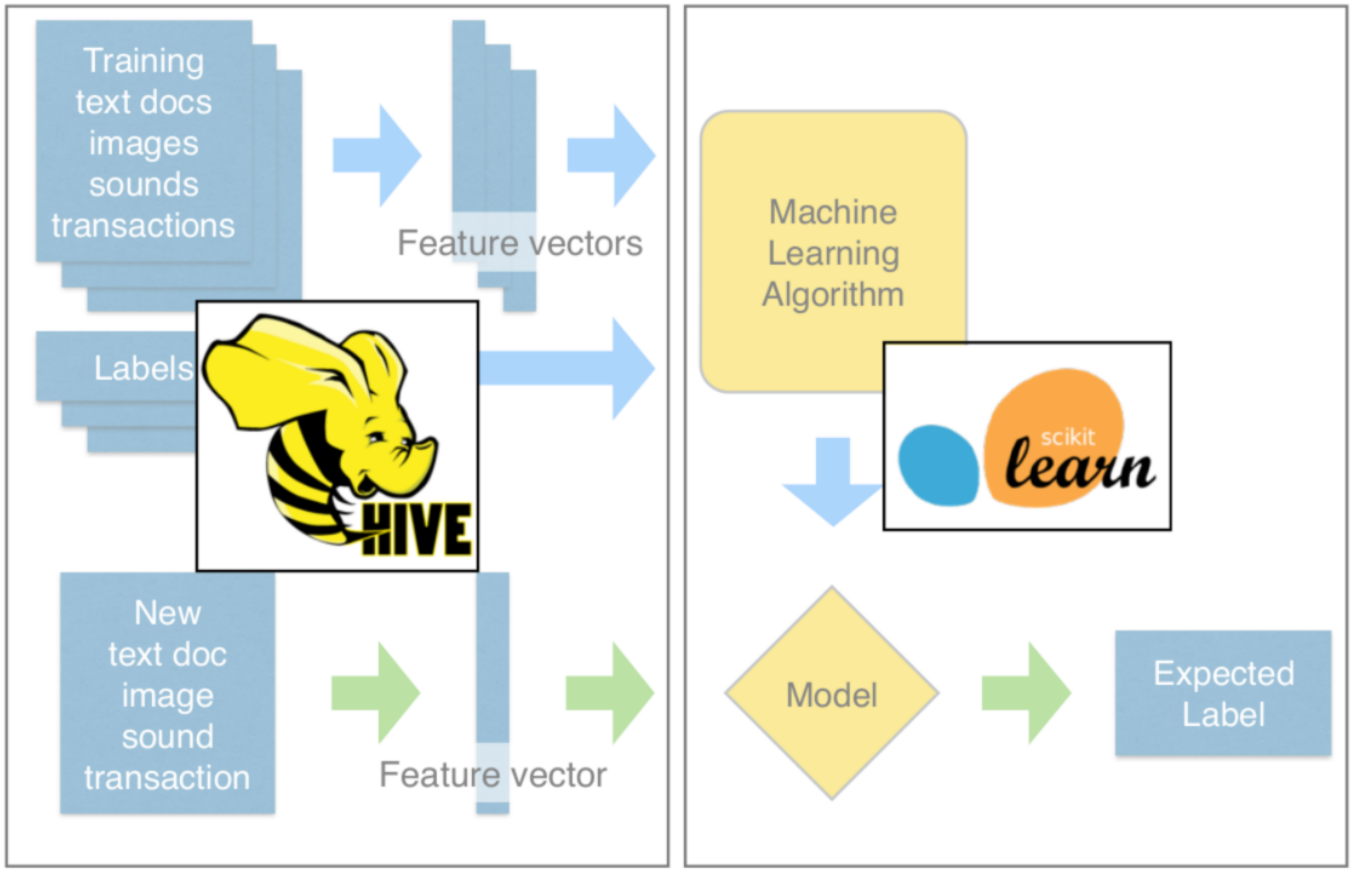
Small / Medium data with python™



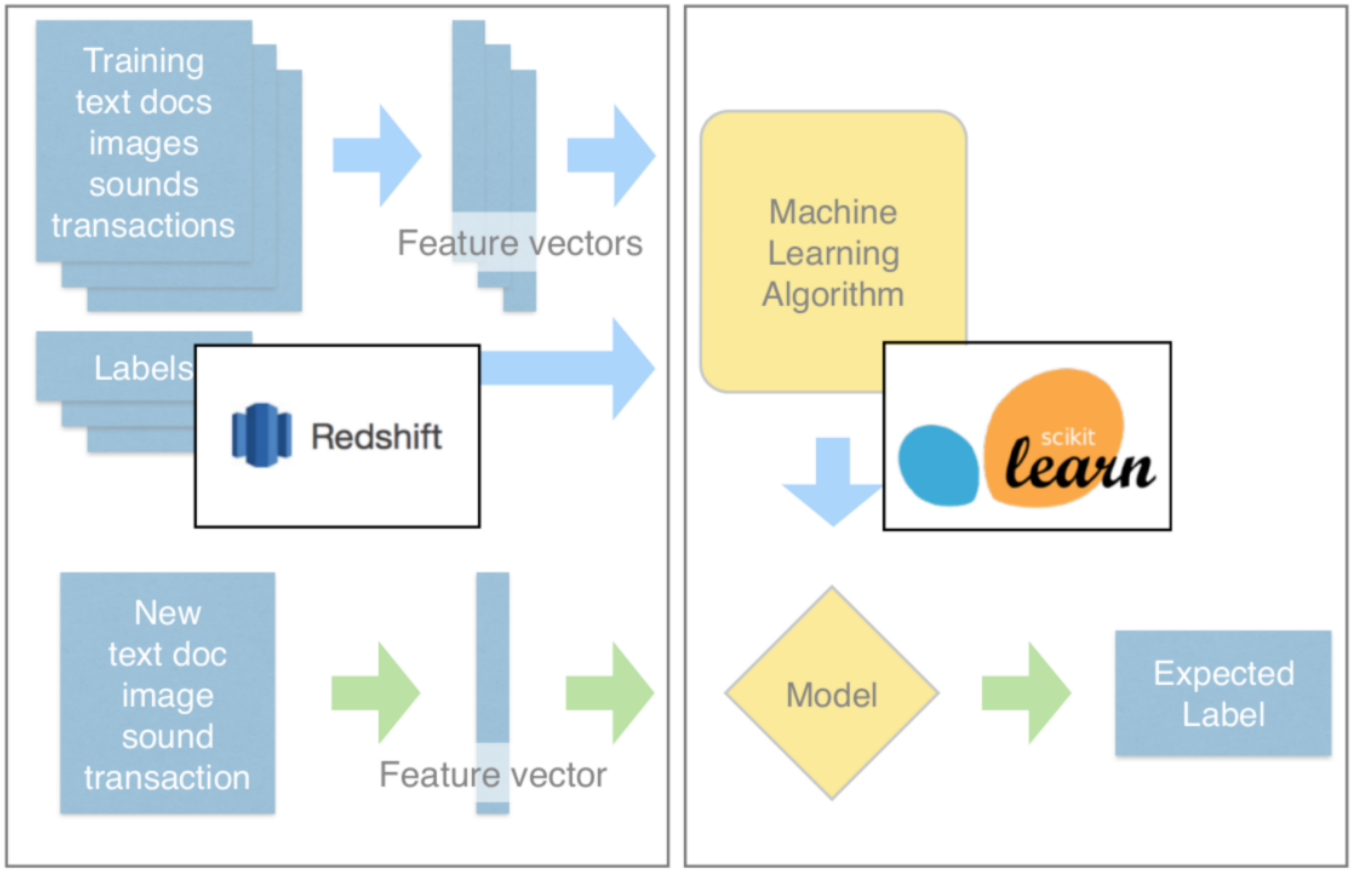
Small / Medium data with python™



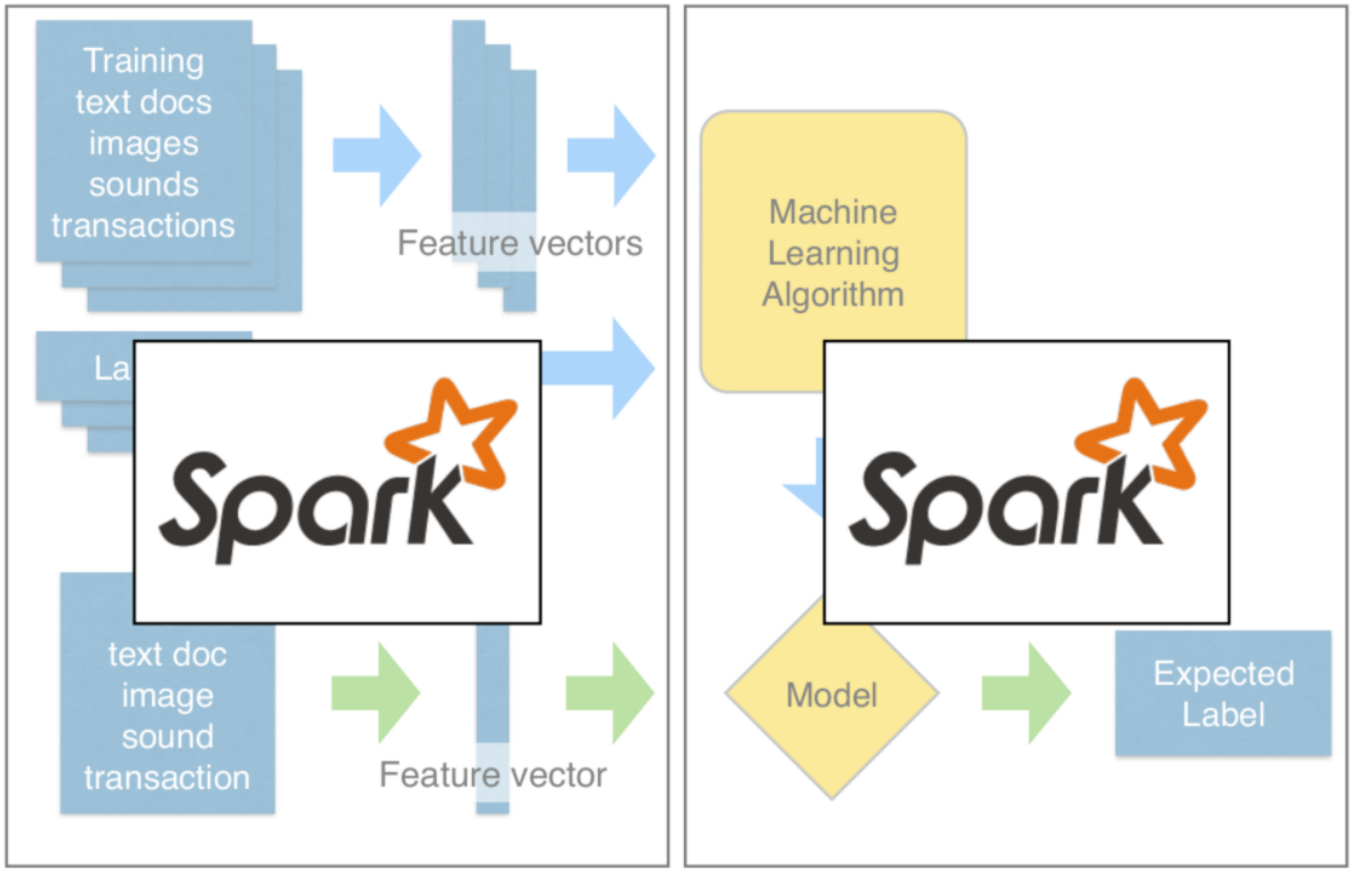
Big data with hadoop



Big data with hadoop python™



Big data with Redshift python™



Big data with Spark

Predictive modeling in the wild

Virality and readers engagement

Fraud detection
Pricing

Personalized radios

Inventory forecasting & trends detection

Predictive maintenance

Personality matching

Scikit-learn

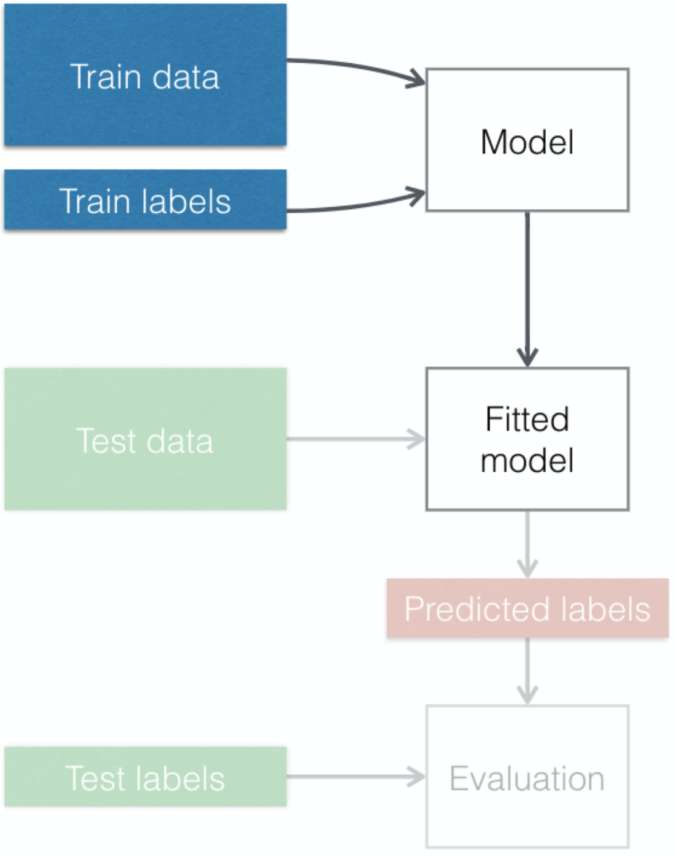
Library of Machine Learning algorithms

Open Source project

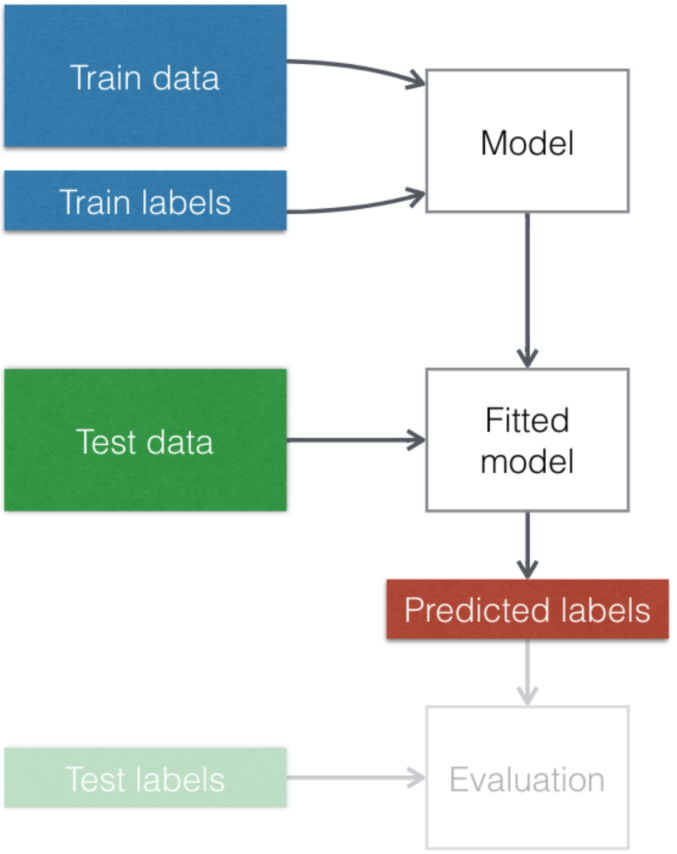
Python / NumPy / SciPy / Cython

Simple **fit** / **predict** / **transform** API

Model Assessment, Selection, Ensembles

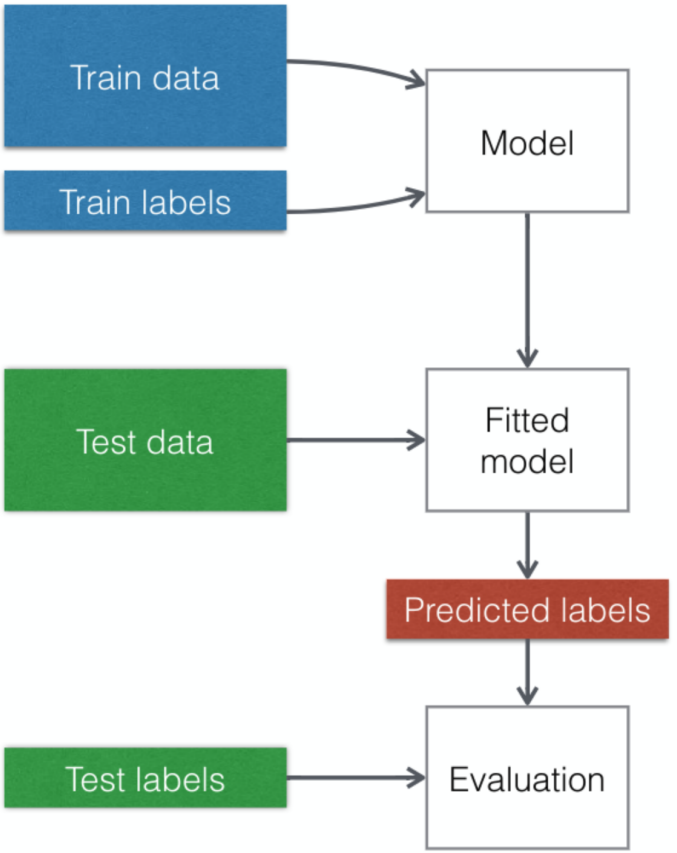



```
model = LogisticRegression(C=1)
model.fit(X_train, y_train)
```




```
model = LogisticRegression(C=1)
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```




```
model = LogisticRegression(C=1)
model.fit(X_train, y_train)
```

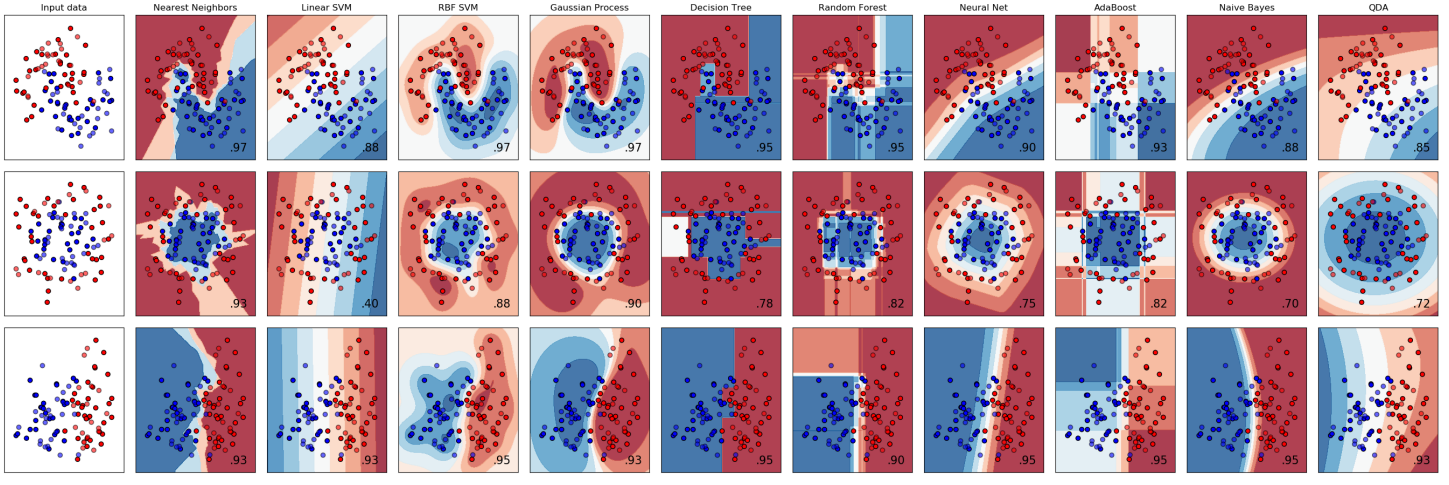
```
y_pred = model.predict(X_test)
```

```
accuracy_score(y_test, y_pred)
```

Support Vector Machine

Linear Classifier

Random Forest


Input data
Nearest Neighbors
Linear SVM
RBF SVM
Gaussian Process
Decision Tree
Random Forest
Neural Net
AdaBoost
Naive Bayes
QDA
.97
.88
.97
.97
.95
.95
.90
.93
.88
.85
.93
.40
.88
.90
.78
.82
.75
.82
.70
.72
.93
.93
.95
.93
.95
.90
.95
.95
.95
.93

scikit-learn: machine lear

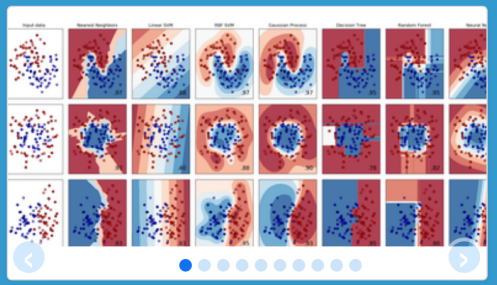
scikit-learn.org/stable/index.html

Search

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 category an object belongs to.

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

Regression

Predicting a continuous-valued attribute associated with an object.

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, feature selection, non-negative matrix factorization.
— Examples

Model selection

Comparing, validating and choosing parameters and models.

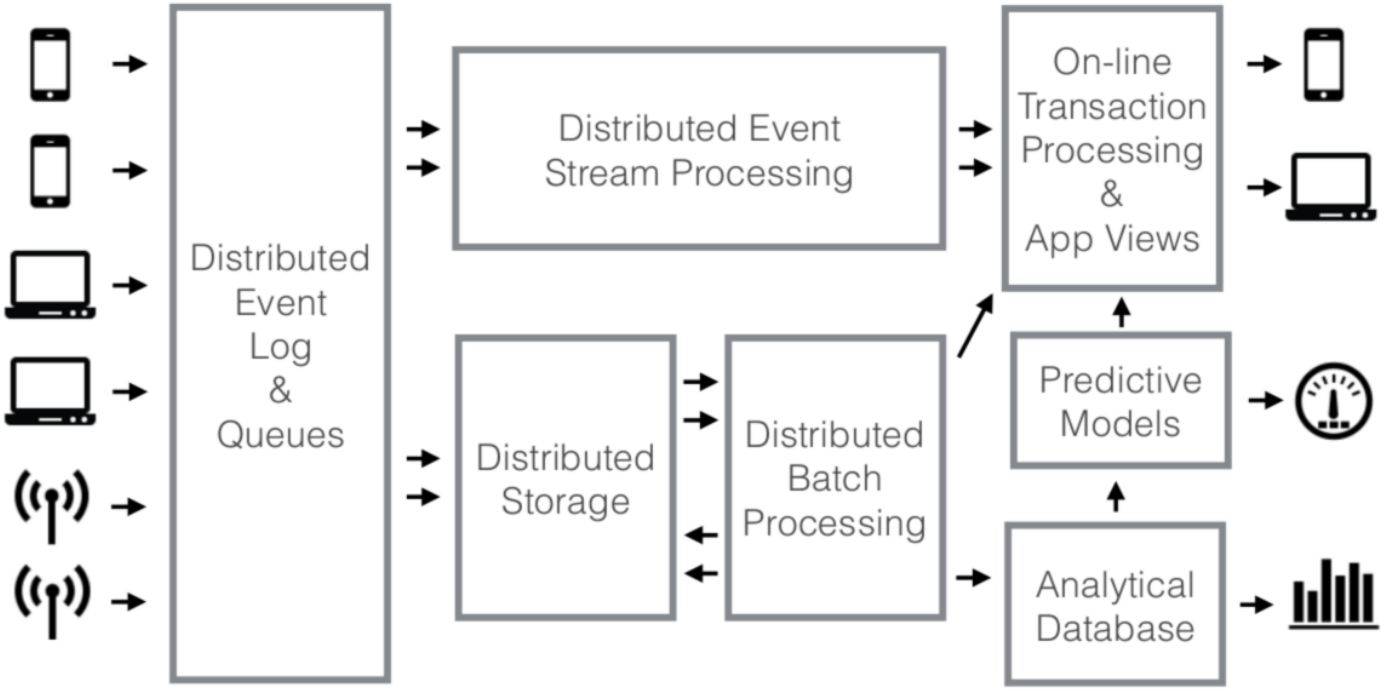
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics.
— Examples

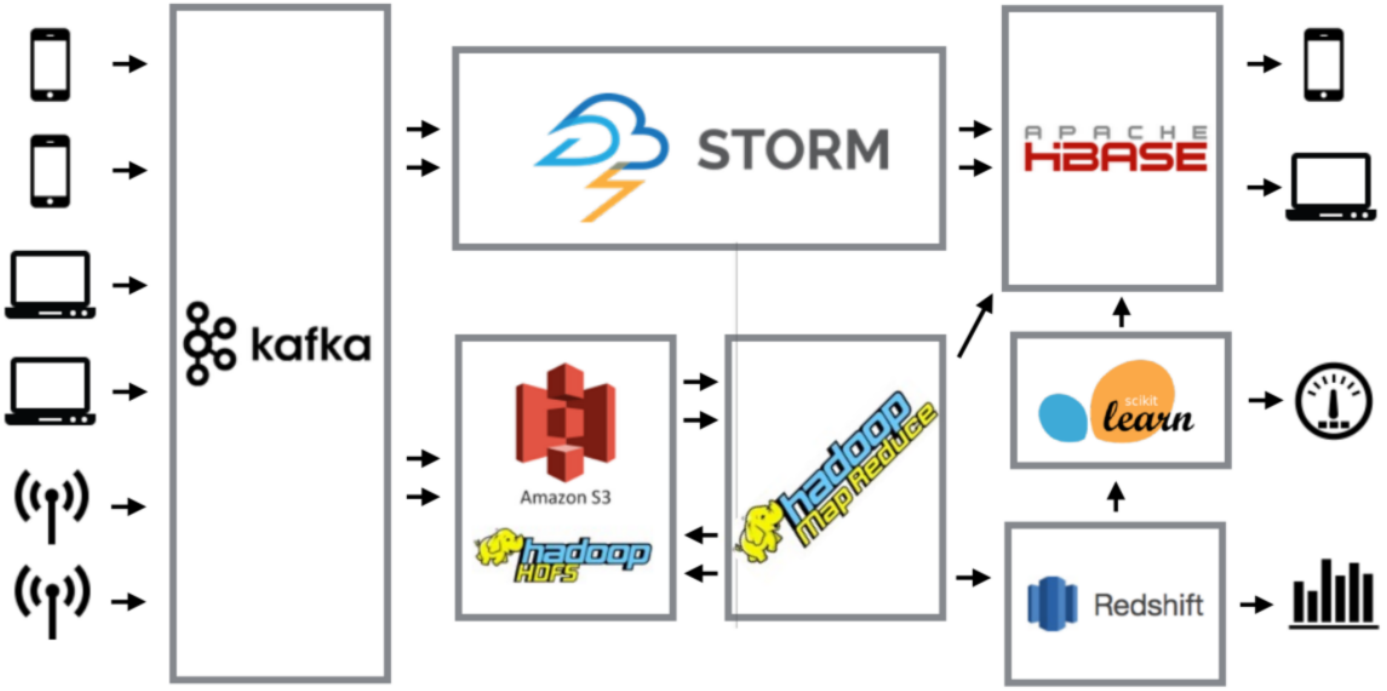
Preprocessing

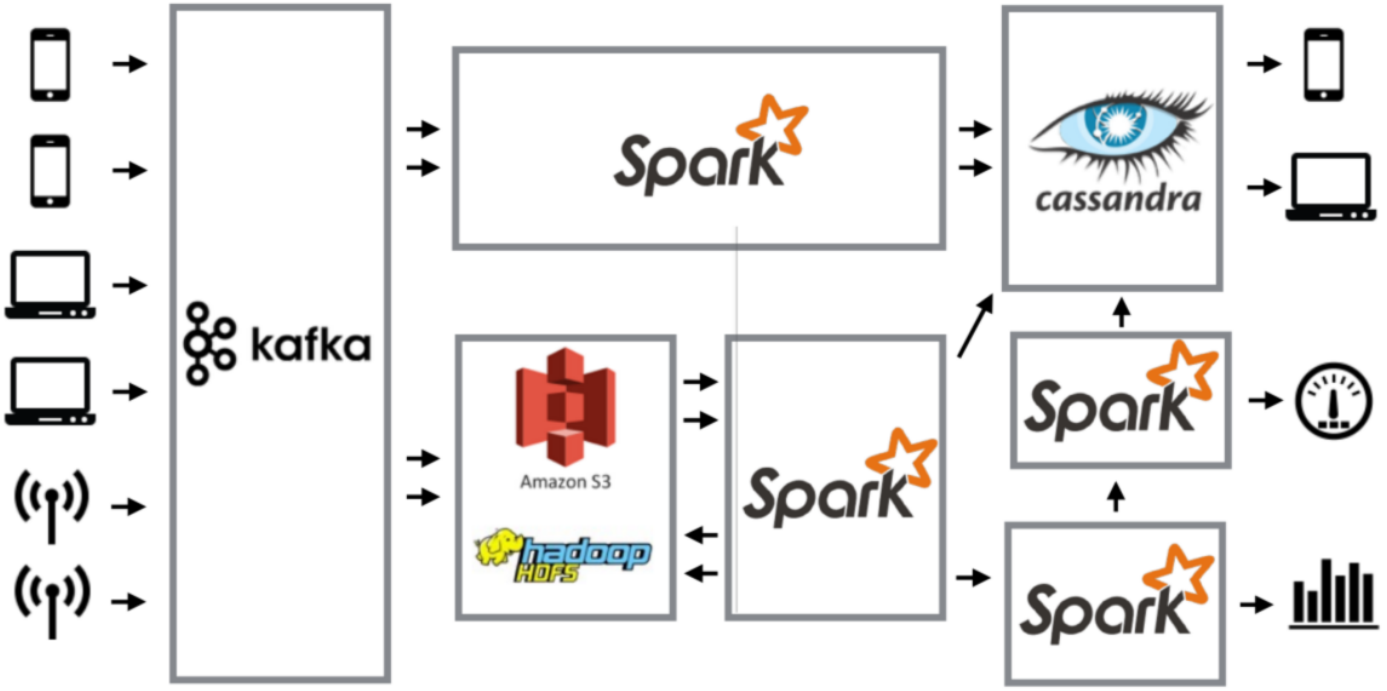
Feature extraction and normalization.

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


Where do predictive models fit?


Distributed
Event
Log
&
Queues
Distributed Event
Stream Processing
Distributed
Storage
Distributed
Batch
Processing
On-line
Transaction
Processing
&
App Views
Predictive
Models
Analytical
Database


kafka
STORM
APACHE
HBASE
Amazon S3
hadoop
HDFS
hadoop
Map Reduce
scikit
learn
Redshift


kafka
Spark
Amazon S3
hadoop
HDFS
Spark
cassandra
Spark
Spark


Scaling predictive modeling

The need for scaling out in ML

I/O intensive feature engineering and model scoring

Loading, filtering, joining, aggregating: SQL-land

Click log ⇒ Session features ⇒ User activity features

CPU intensive model fitting

Hyper-parameter search and cross-validation

Gradient Boosting, Random Forests, Large Neural Networks

Limitations of PySpark

Python driver -> Scala / JVM -> Python worker

Latency induced by the networked architecture

Complex traceback / errors for non-scala developers

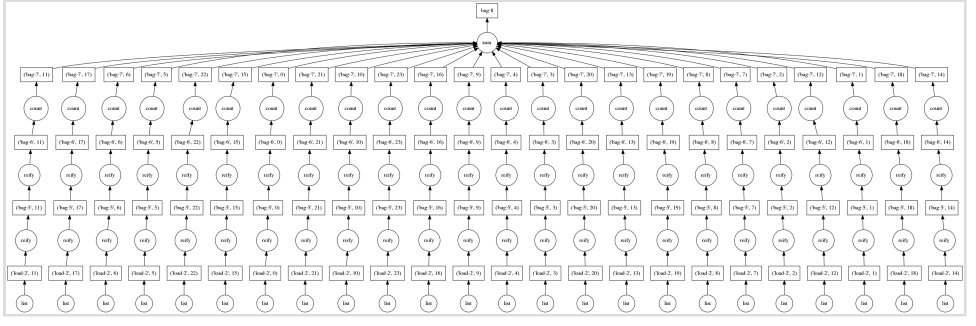
No pure Python local mode for PySpark

Impossible to use profiler or ipdb for inner calls

dask

- Collection API similar to NumPy array and Pandas DataFrame objects
- Custom workloads via tasks scheduling API
- Pure python and low overhead
- **Scales up**: runs on clusters with 1000s of cores
- **Scales down**: runs on a laptop in a single process
- Experimental integration with sklearn: dask-ml

dask bags & compute graphs

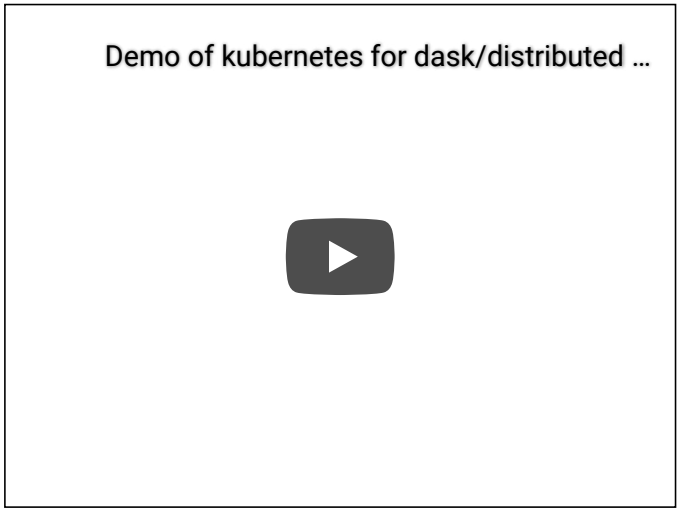


dask arrays & compute graphs

Demo

https://github.com/ogrisel/docker-distributed

Have a look a the notebooks in the folder.



https://youtu.be/6mKNSEQ0FIQ - Local video

Dask + Distributed limitations

Younger project (under very active development)

⊂ or PySpark

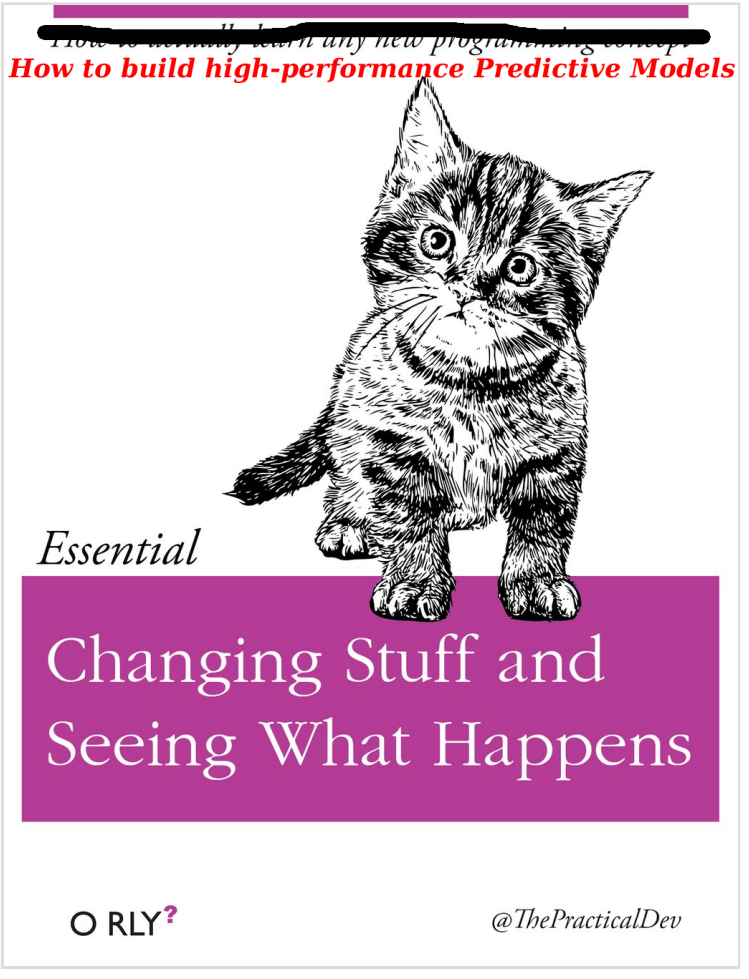
Simple design: e.g. no predicate push-down in dataframe

dask-scheduler is a single point of failure

Not meant for multi-tenancy (yet?)

Scikit-learn integration: experimental

Conclusion


How to build high-performance Predictive Models
Essential
Changing Stuff and Seeing What Happens
O RLY?
@ThePracticalDev

Secrets of the success of Python (& R) in Data Science

Iterative exploration with built-in plotting tools

Low latency of single host in-memory computing

Easy to install, easy to teach: no-sysadmin required

Rich ecosystem of libraries

Conclusion

Scikit-learn is a versatile ML toolkit

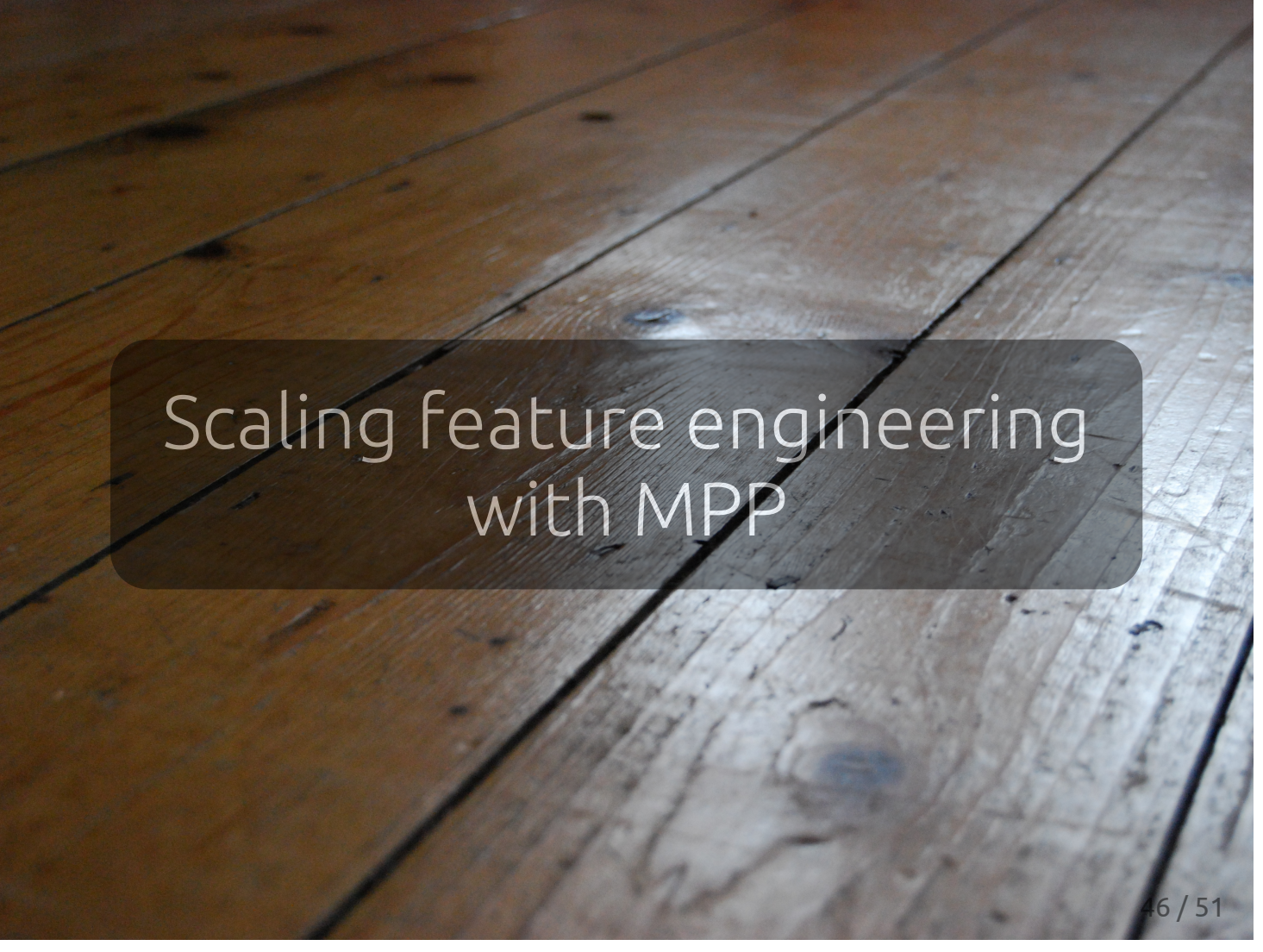
with NumPy and pandas for feature engineering

with Jupyter and matplotlib for interactive data exploration

PyData moving towards big data / compute for (e.g. dask and xarray)

Thank you for your attention!

- https://scikit-learn.org
- Slides: ogrisel.github.io/decks/2017_intro_sklearn
- @ogrisel on twitter


Scaling feature engineering
with MPP

Massively Parallel Processing

Problem with the use of SQL in MPP

Great for ad-hoc queries but no plotting

SQL strings in Python code is sad :(

Not easy to write test / CI

Standard ORM not a solution

MPP in Python

& provide 'dataframe' like Python API

Under the hood it can generate SQL for MPP engines

also targets non-SQL backends (pandas, MongoDB, PySpark...)

Background image credits

- https://www.flickr.com/photos/jemimus/8533890844/
- https://www.flickr.com/photos/antcaz/2249694239/
- https://www.flickr.com/photos/benjamine-s/14004414605
- https://www.flickr.com/photos/a-herzog/9026372290