Anomaly/Novelty Detection with scikit-learn

Alexandre Gramfort

Telecom ParisTech - CNRS LTCI
alexandre.gramfort@telecom-paristech.fr

GitHub : @agramfort

Twitter : @agramfort

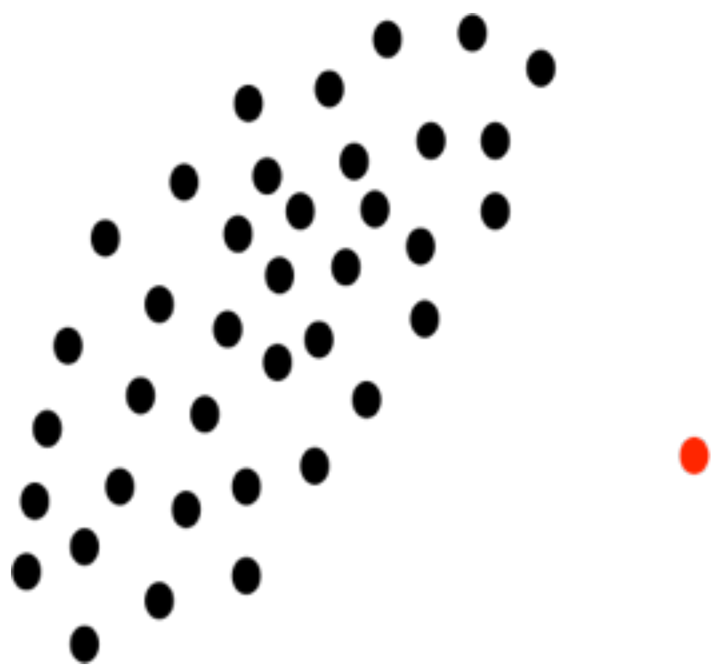
What's the problem?

Objective: Spot the red apple

What's the problem?

"*An **outlier** is an **observation** in a data set which appears to be **inconsistent** with the remainder of that set of data.*"

<u>Johnson 1992</u>



"*An **outlier** is an **observation** which **deviates** so much from the other observations as to **arouse suspicions** that it was generated by a **different mechanism**.*"

<u>Hawkins 1980</u>

Types of AD

- **Supervised AD**
  - Labels available for both normal data and anomalies
  - Similar to rare class mining / imbalanced classification
- **Semi-supervised AD (Novelty Detection)**
  - Only normal data available to train
  - The algorithm learns on normal data only
- **Unsupervised AD (Outlier Detection)**
  - no labels, training set = normal + abnormal data
  - Assumption: anomalies are very rare

Types of AD

- **Supervised AD**
  - Labels available for both normal data and anomalies
  - Similar to rare class mining / imbalanced classification
- **Semi-supervised AD (Novelty Detection)**
  - Only normal data available to train
  - The algorithm learns on normal data only
- **Unsupervised AD (Outlier Detection)**
  - no labels, training set = normal + abnormal data
  - Assumption: anomalies are very rare

ML Taxonomy

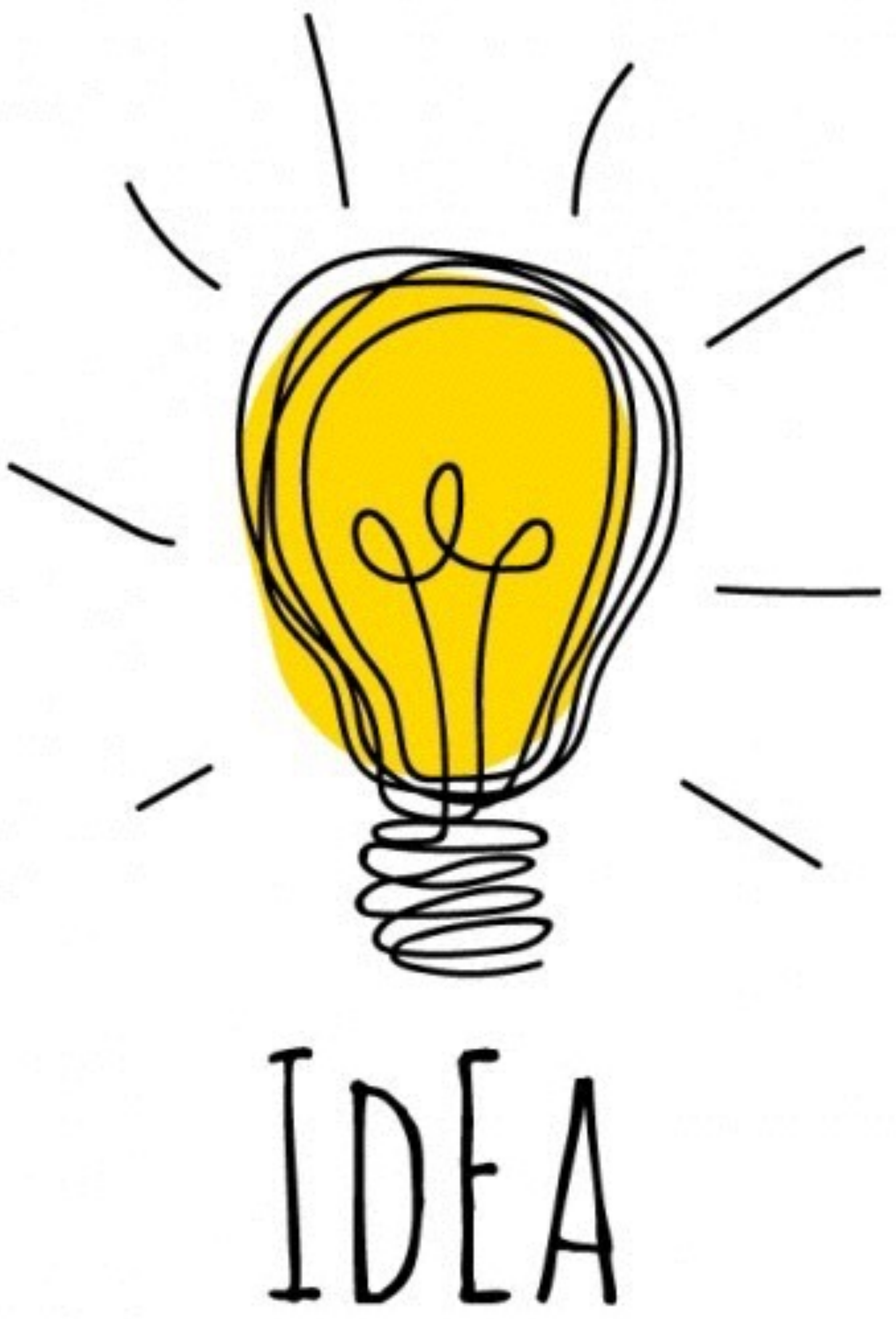
- Machine Learning
  - Unsupervised
    - Clustering
    - Dim. Red.
    - Anomaly Novelty Detection
  - Supervised “Prediction”
    - Regression
    - Classif.

Applications

- Fraud detection
- Network intrusion
- Finance
- Insurance
- Maintenance
- Medicine (unusual symptoms)
- Measurement errors (from sensors)

Any application where looking at **unusual observations is relevant**

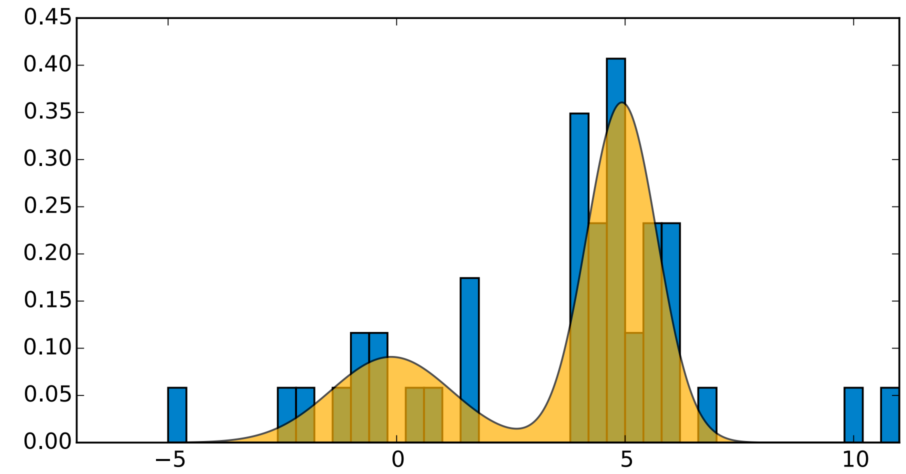

LET'S SPOCK SOME ANOMALIES



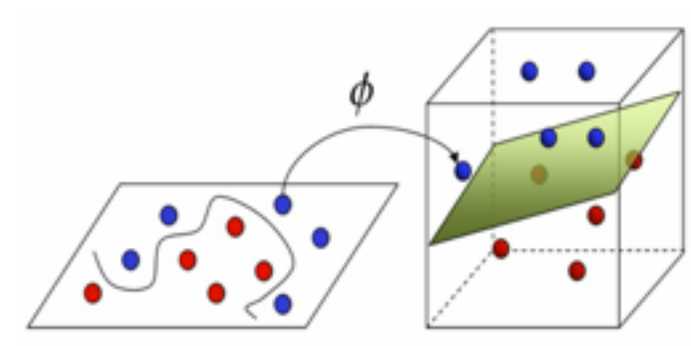
Look for samples that are in **low density** regions, **isolated**

Look for a **region** of the space that is **small in volume but contains most of the samples**

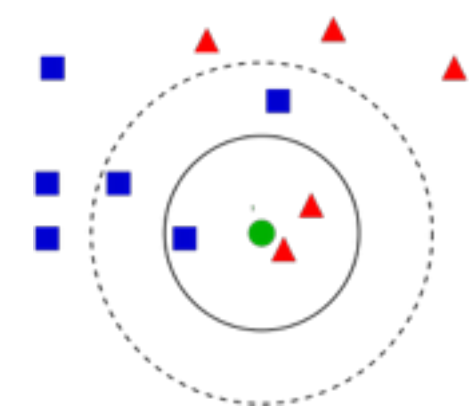
Density based approach (KDE, Gaussian Ellipse, GMM)



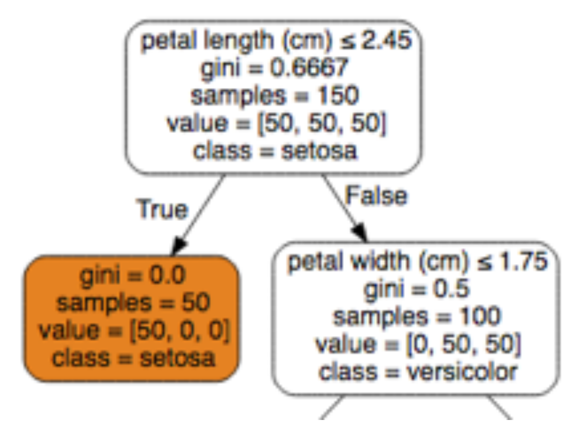
Kernel methods



Nearest neighbors

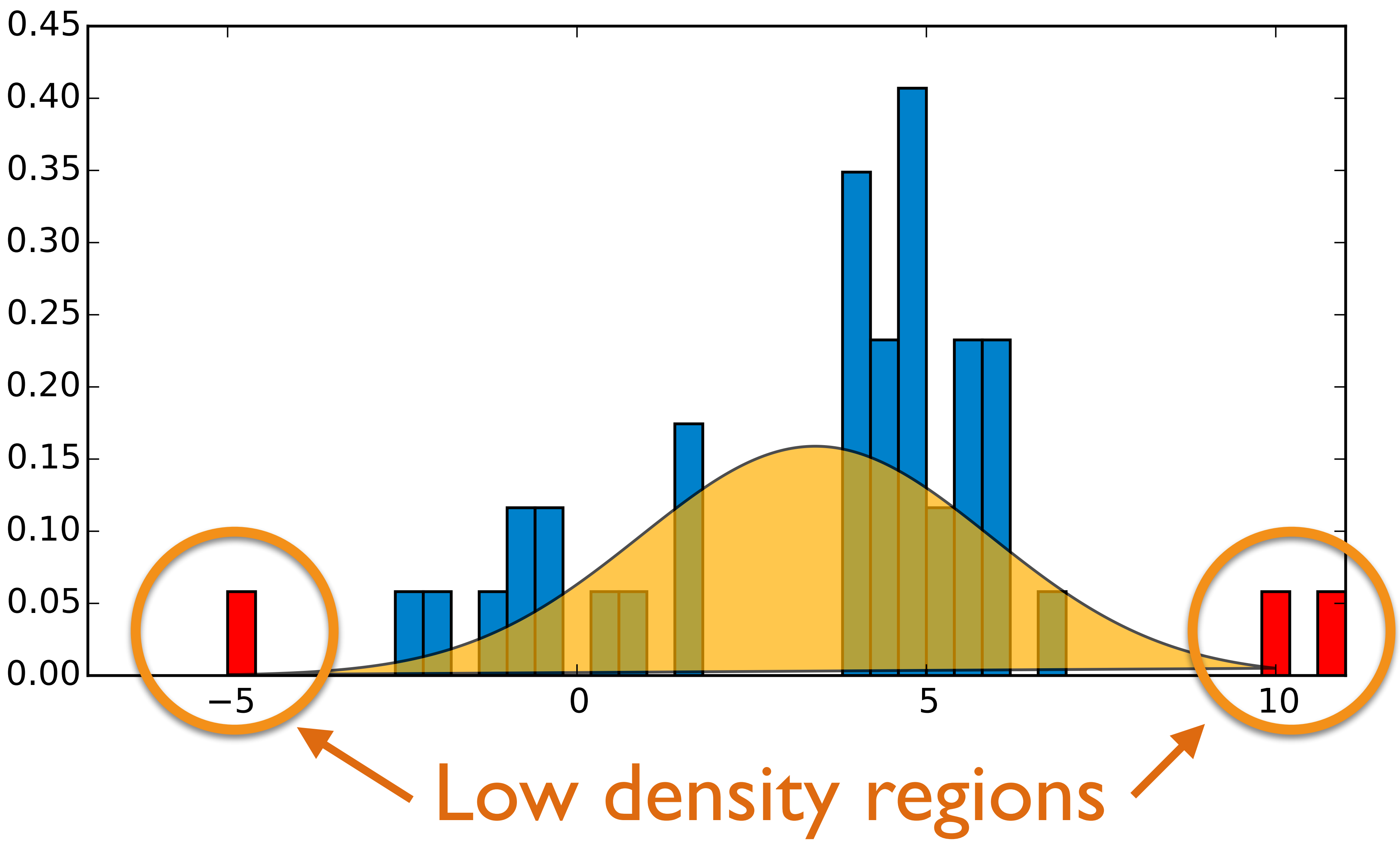


Trees / Partitioning



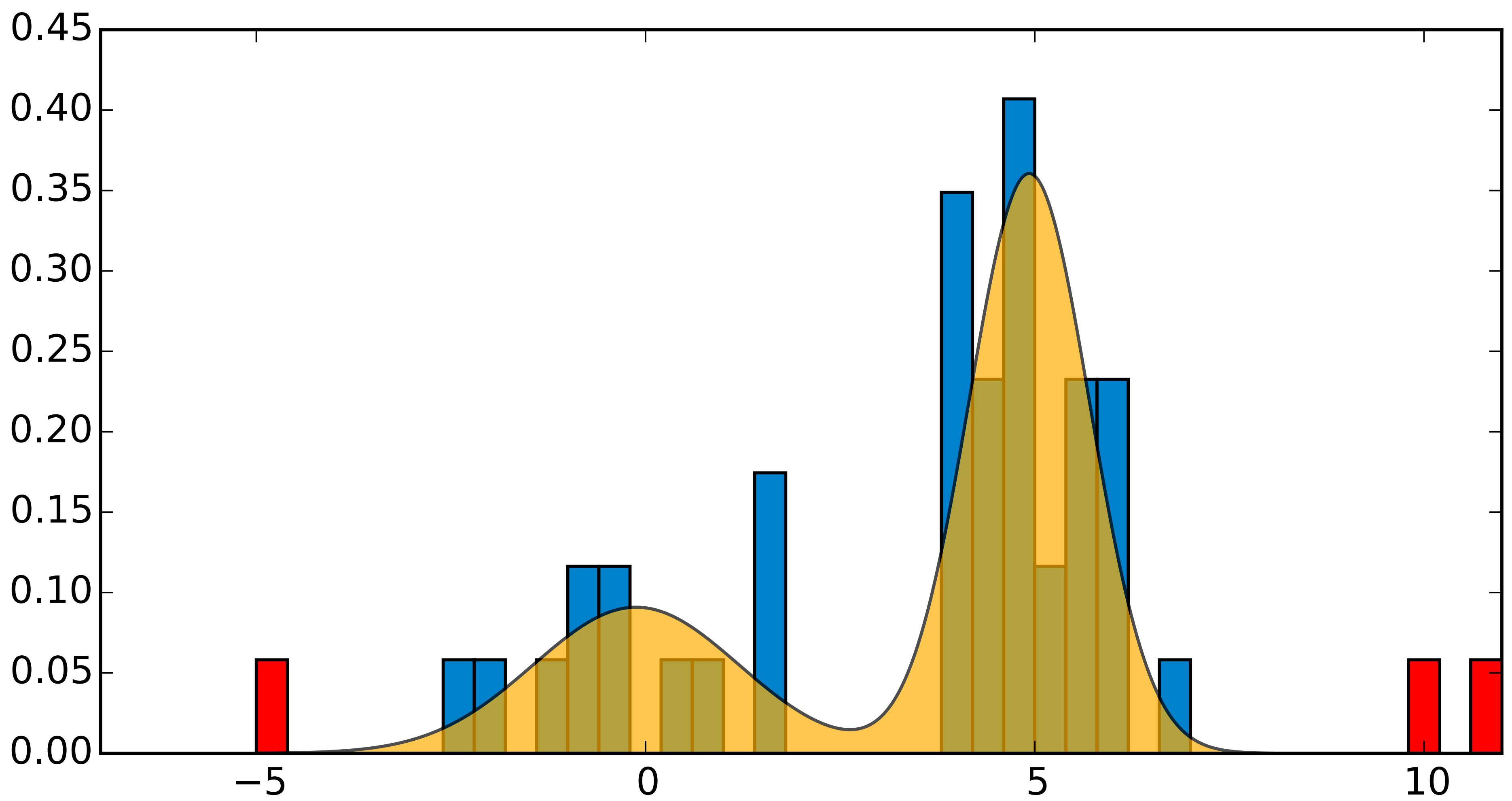
Novelty detection via density estimates

```
>>> from sklearn.mixture import GaussianMixture
>>> gmm = GaussianMixture(n_components=1).fit(X)
>>> log_dens = gmm.score_samples(X_plot)
>>> plt.fill(X_plot[:, 0], np.exp(log_dens), fc='#ffaf00', alpha=0.7)
```



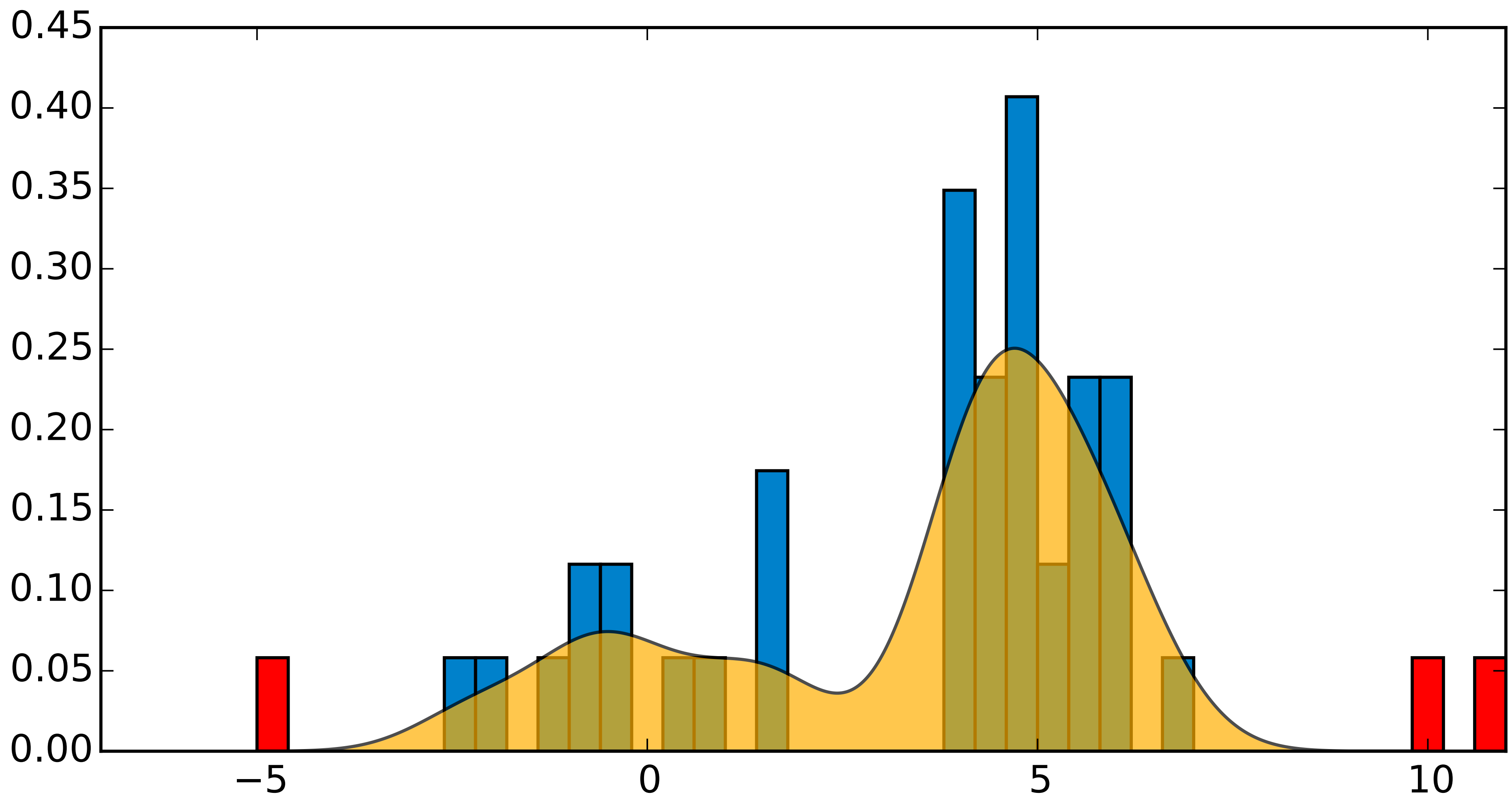
Novelty detection via density estimates

```
>>> from sklearn.mixture import GaussianMixture
>>> gmm = GaussianMixture(n_components=2).fit(X)
>>> log_dens = gmm.score_samples(X_plot)
>>> plt.fill(X_plot[:, 0], np.exp(log_dens), fc='#ffaf00', alpha=0.7)
```

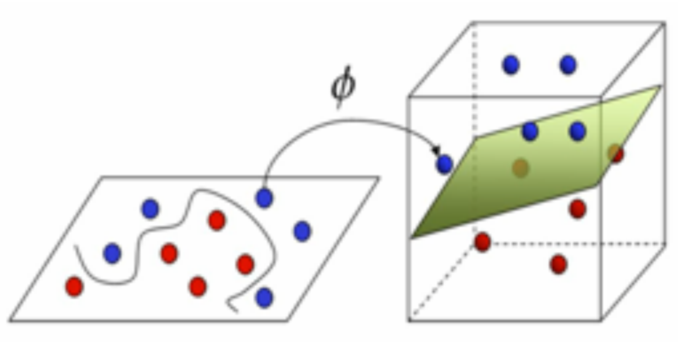


Novelty detection via density estimates

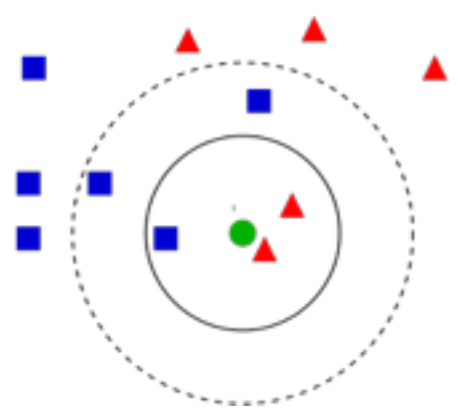
```
>>> from sklearn.neighbors import KernelDensity
>>> kde = KernelDensity(kernel='gaussian', bandwidth=0.75).fit(X)
>>> log_dens = kde.score_samples(X_plot)
>>> plt.fill(X_plot[:, 0], np.exp(log_dens), fc='#ffaf00', alpha=0.7)
```



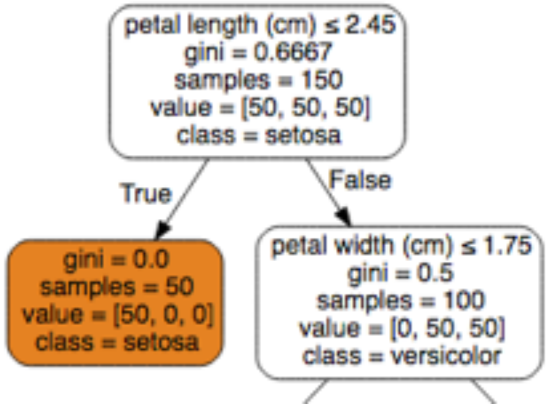
Kernel methods



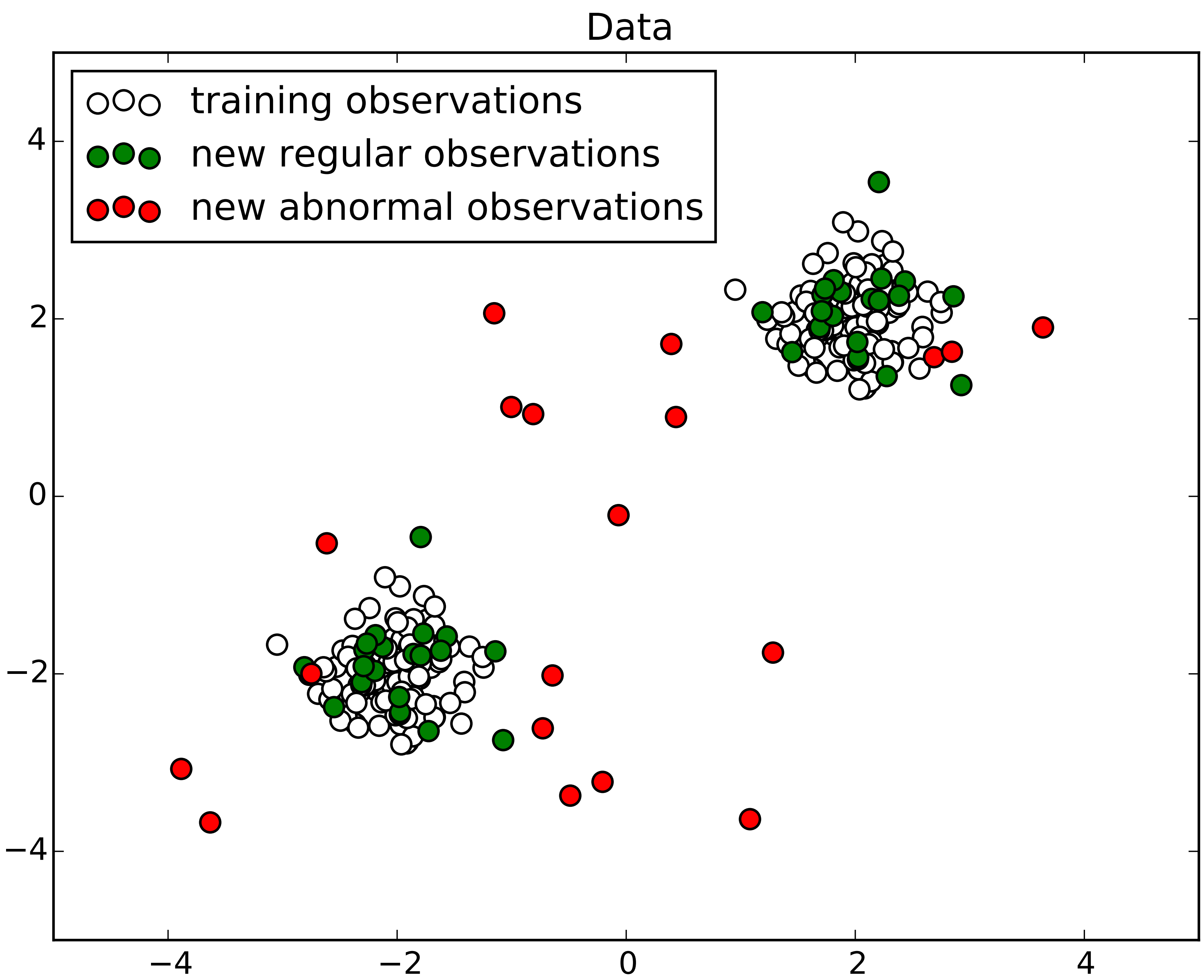
Nearest neighbors



Trees / Partitioning

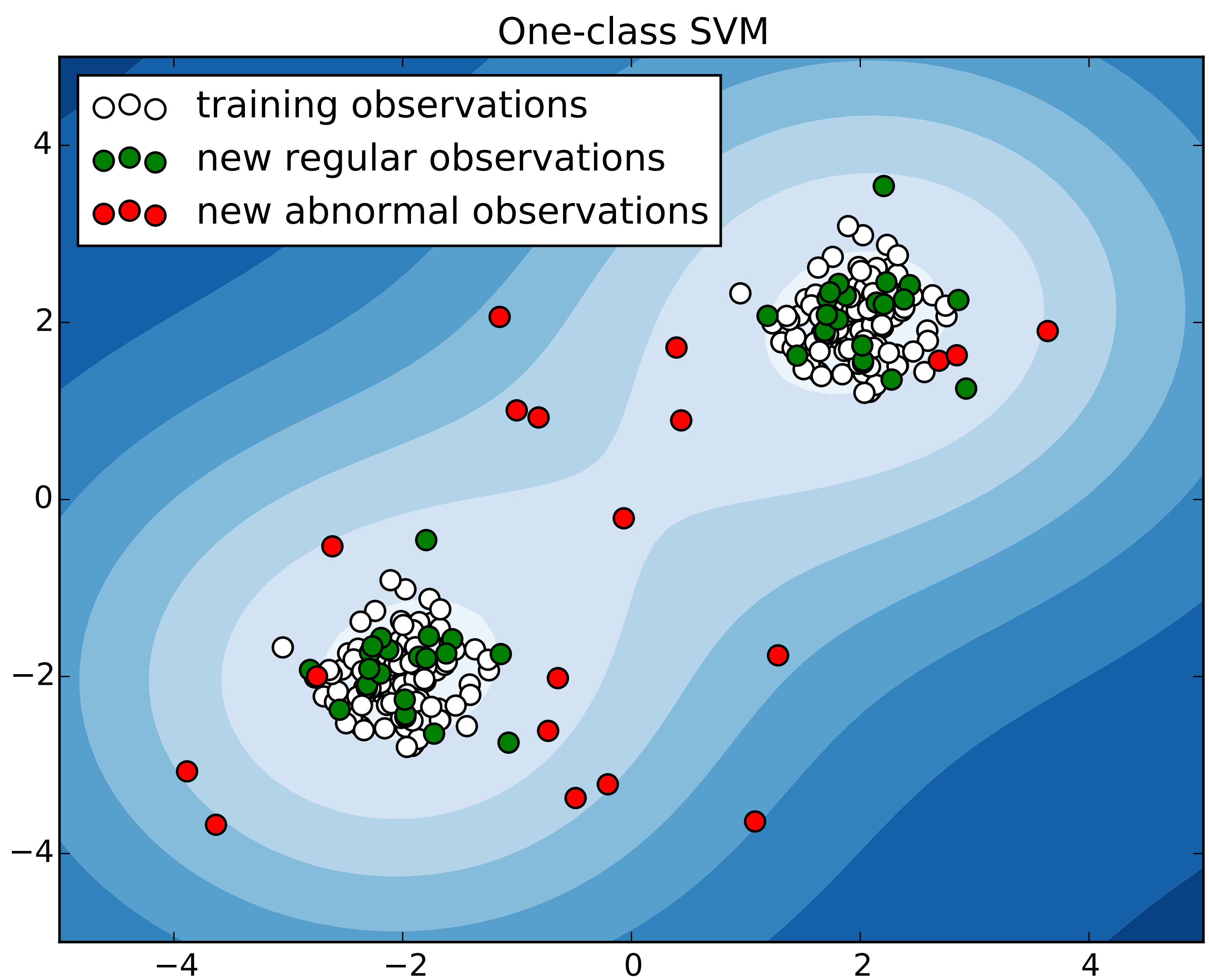


Our toy dataset



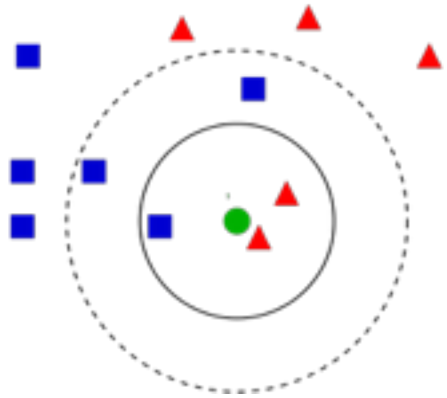
Kernel Approach

```
>>> est = OneClassSVM(nu=0.1, kernel="rbf", gamma=0.1)
```

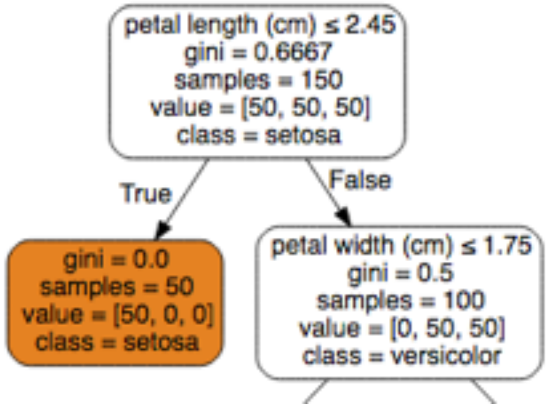


http://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html

Nearest neighbors

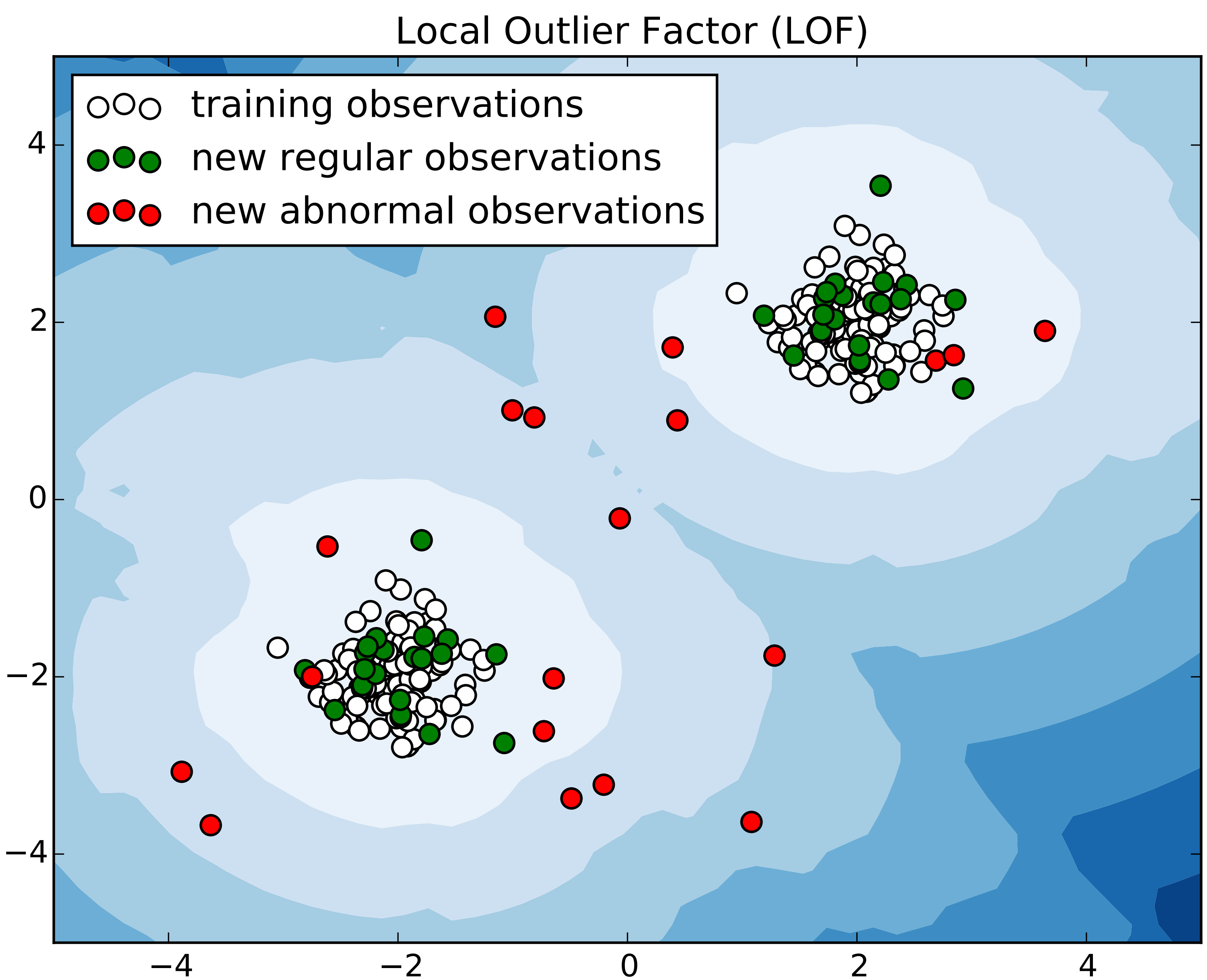


Trees / Partitioning



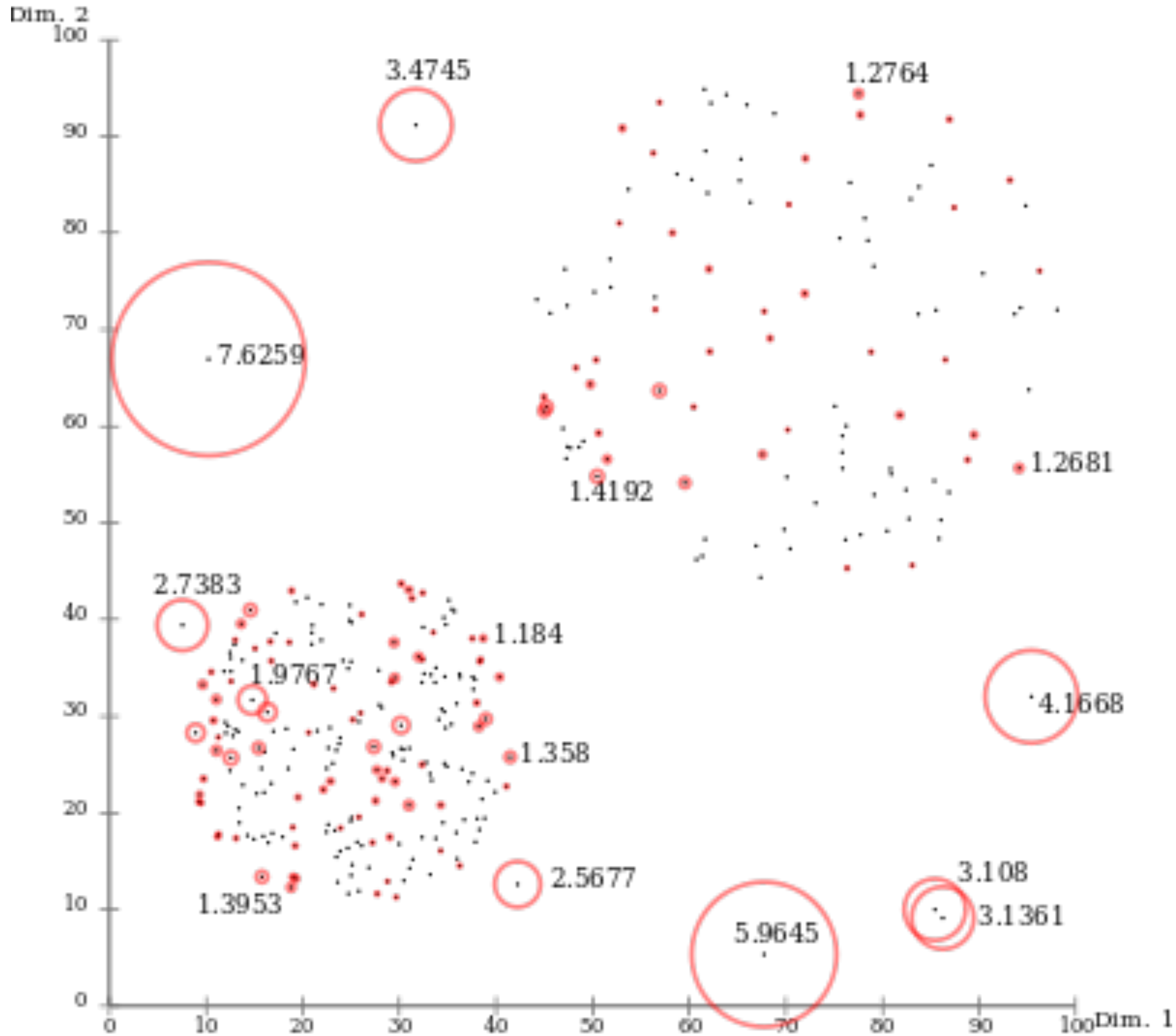
Nearest Neighbors (NN) Approach

```
>>> est = LocalOutlierFactor(n_neighbors=5)
```



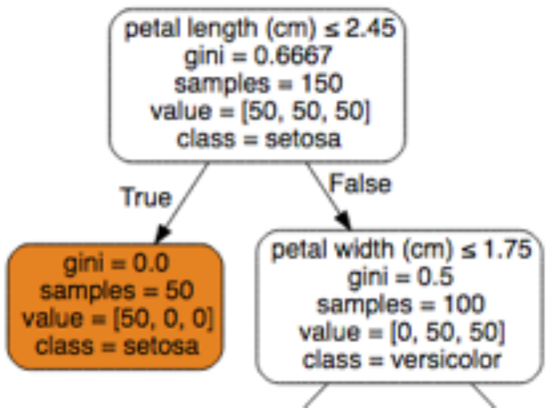
https://github.com/scikit-learn/scikit-learn/pull/5279

Local Outlier Factor (LOF)



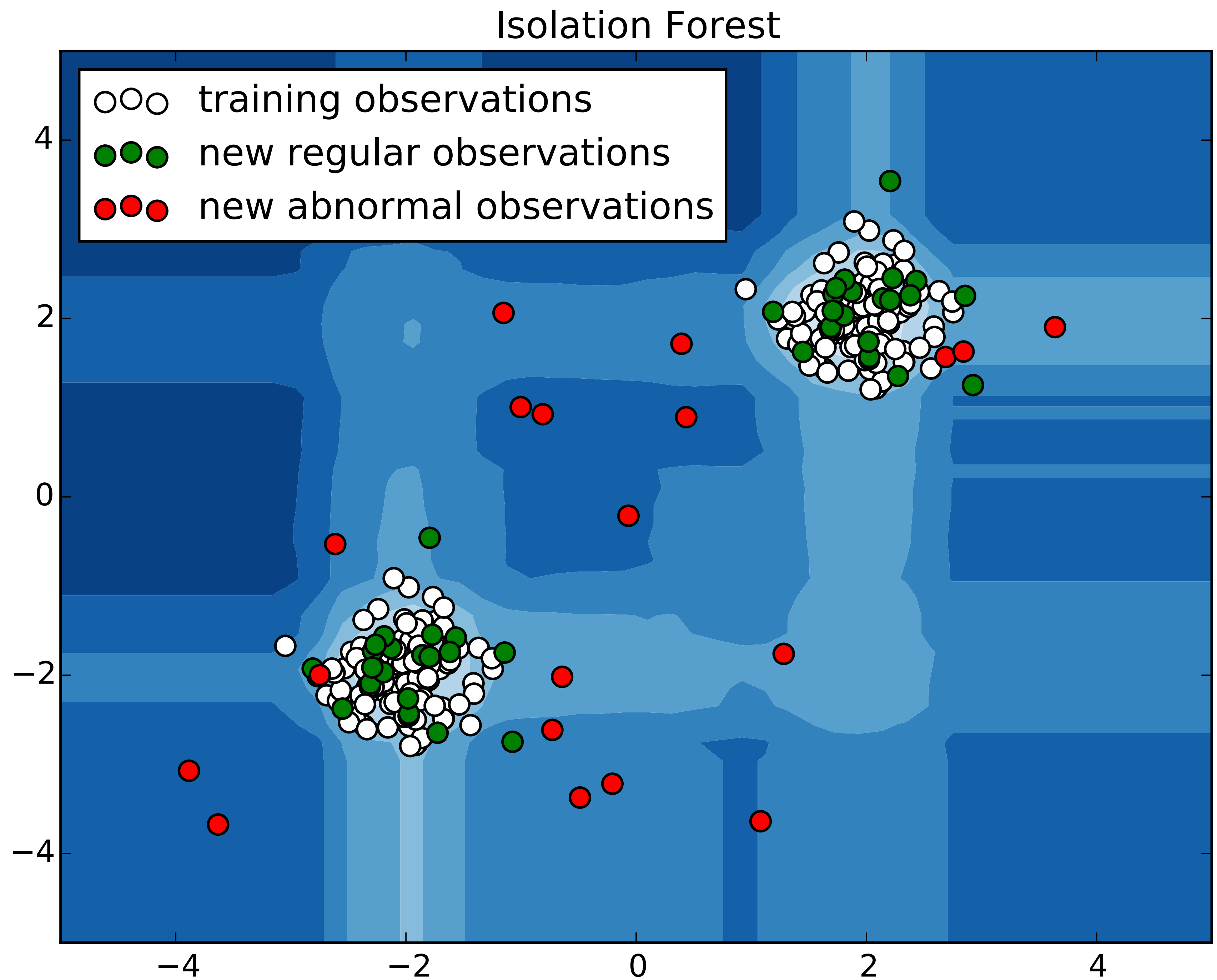
https://en.wikipedia.org/wiki/Local_outlier_factor

Trees / Partitioning



Partitioning / Tree Approach

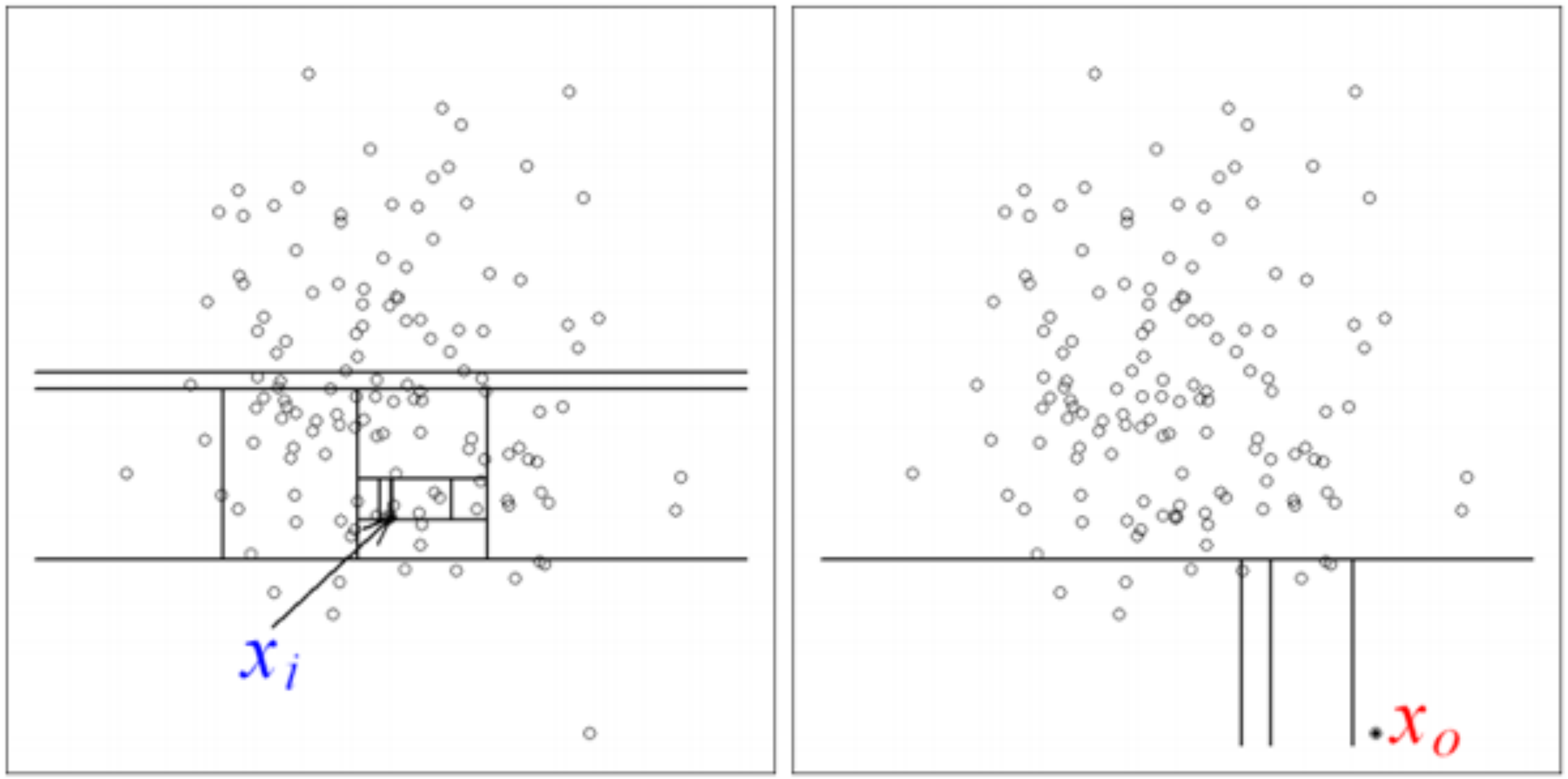
```
>>> est = IsolationForest(n_estimators=100)
```



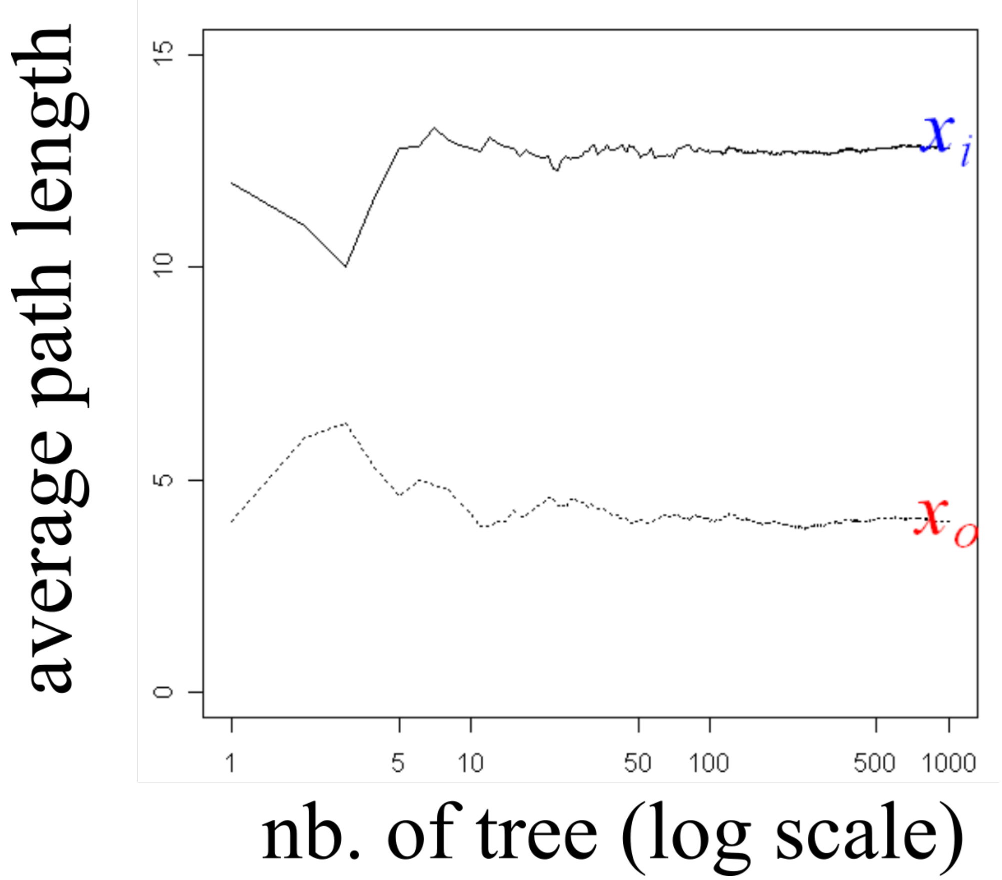
http://scikit-learn.org/dev/modules/generated/sklearn.ensemble.IsolationForest.html

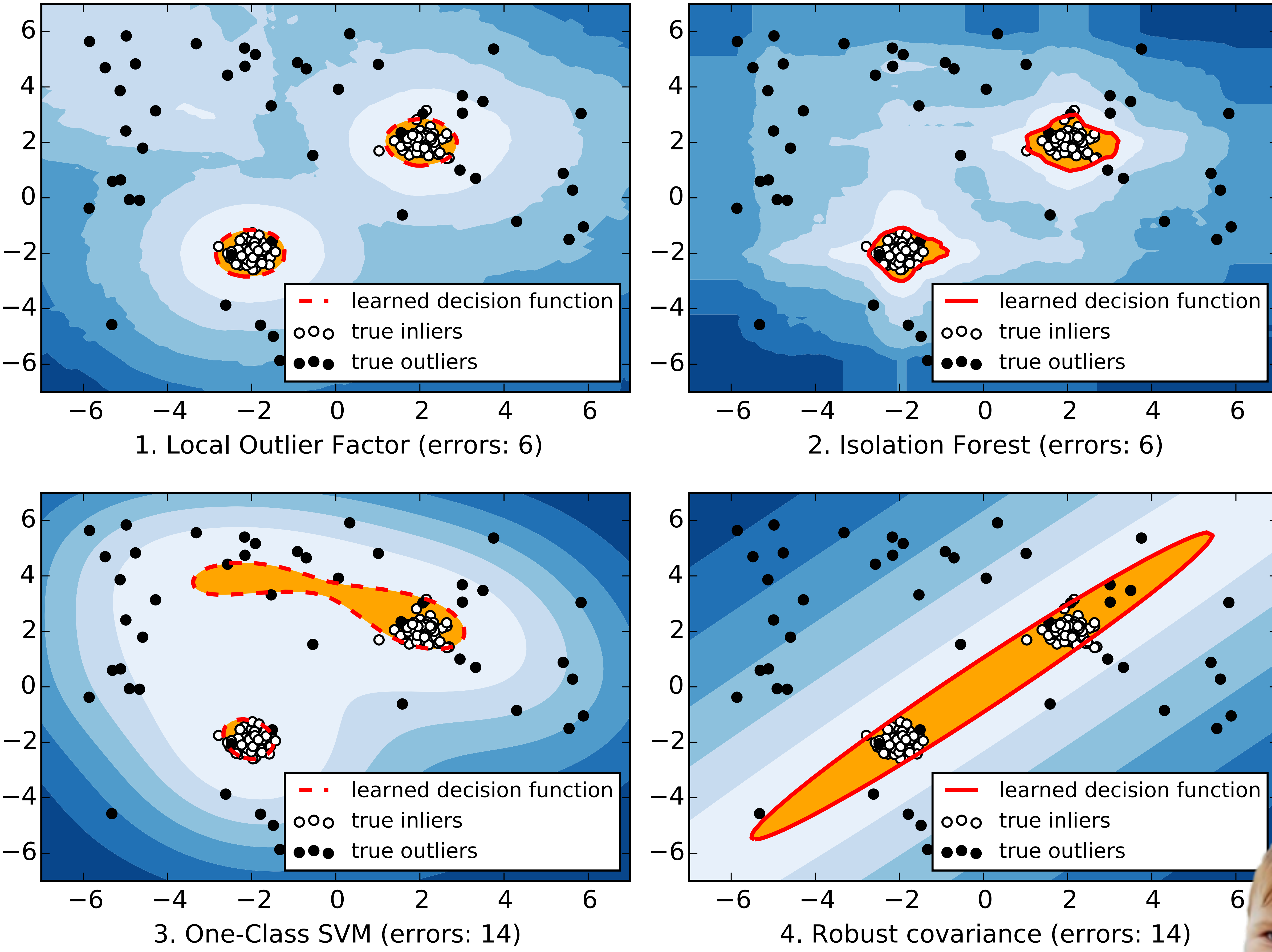
Isolation Forest

An anomaly can be isolated with a very shallow random tree



(a) Isolating x_i (b) Isolating x_o

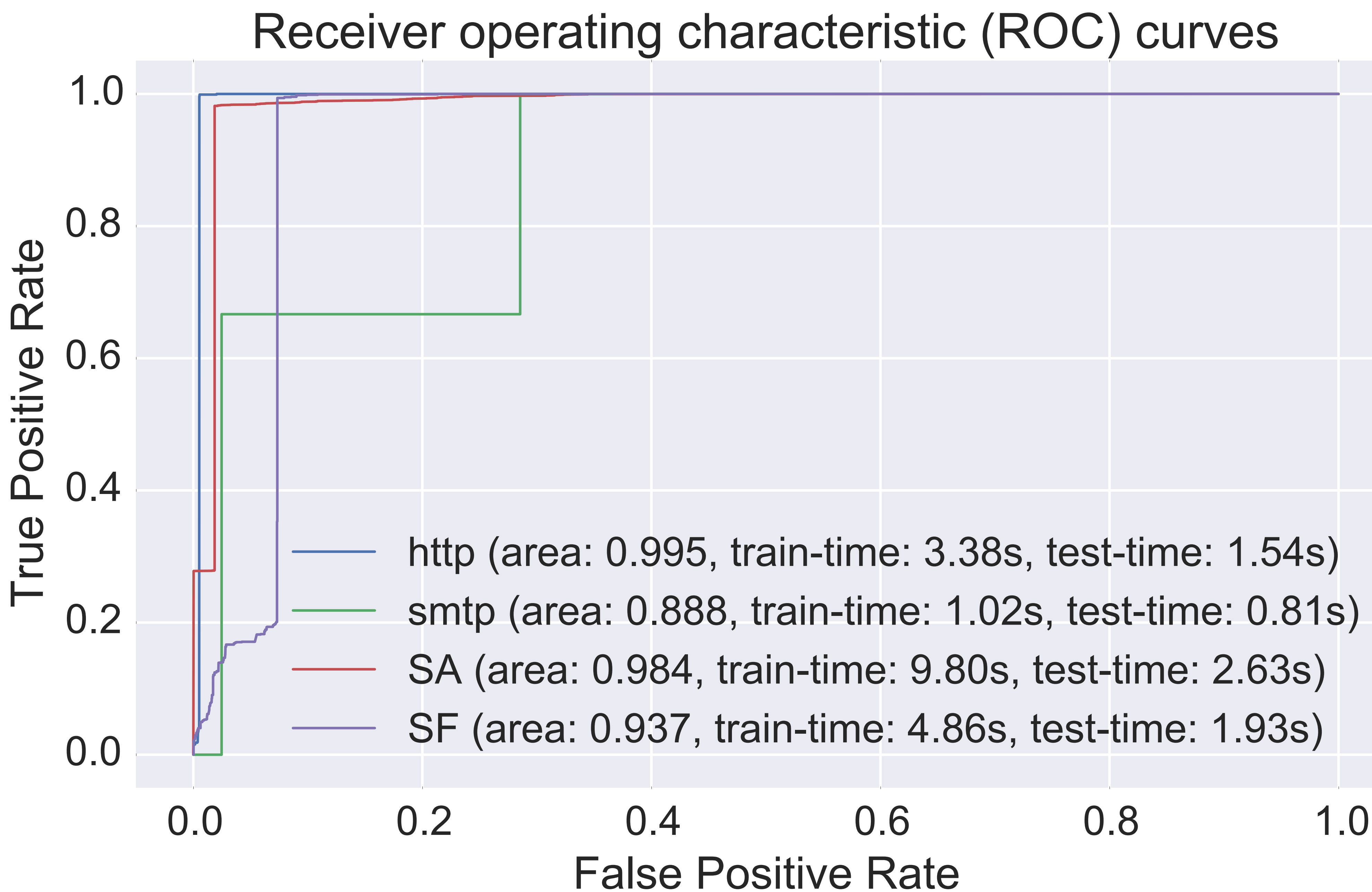




6
4
2
0
−2
−4
−6
learned decision function
true inliers
true outliers
1. Local Outlier Factor (errors: 6)
2. Isolation Forest (errors: 6)
3. One-Class SVM (errors: 14)
4. Robust covariance (errors: 14)

http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Network intrusions



https://github.com/scikit-learn/scikit-learn/blob/master/benchmarks/bench_isolation_forest.py

Some caveats

- How to set model hyperparameters?
- How to evaluate performance in the unsupervised setup?
- In any AD method there is a notion of metric/similarity between samples, e.g. Euclidian distance. Unclear how to define it (think continuous, categorical features etc.)

http://scikit-learn.org/stable/modules/outlier_detection.html

2.7. Novelty and Outlier Detection

Many applications require being able to decide whether a new observation belongs to the same distribution as existing observations (it is an inlier), or should be considered as different (it is an outlier). Often, this ability is used to clean real data sets. Two important distinction must be made:

novelty detection:
The training data is not polluted by outliers, and we are interested in detecting anomalies in new observations.

outlier detection:
The training data contains outliers, and we need to fit the central mode of the training data, ignoring the deviant observations.

The scikit-learn project provides a set of machine learning tools that can be used both for novelty or outliers detection. This strategy is implemented with objects learning in an unsupervised way from the data:

```
estimator.fit(X_train)
```

new observations can then be sorted as inliers or outliers with a predict method:

```
estimator.predict(X_test)
```

Inliers are labeled 1, while outliers are labeled -1.

2.7.1. Novelty Detection

Consider a data set of n observations from the same distribution described by p features. Consider now that we add one more observation to that data set. Is the new observation so different from the others that we can doubt it is regular? (i.e. does it come from the same distribution?) Or on the contrary, is it so similar to the other that we cannot distinguish it from the original observations? This is the question addressed by the


ANOMALIES
YOU SHALL FIND!

Contact:

Alexandre Gramfort
alexandre.gramfort@telecom-paristech.fr

GitHub : @agramfort

Twitter : @agramfort

Questions?

Thanks @ngoix & @albertthomas88 for the work

1 position to work on Scikit-Learn and Scipy stack available !

Paris-Saclay
Center for Data Science

TELECOM
ParisTech