

ALGORITHM ANOMALY DETECTION

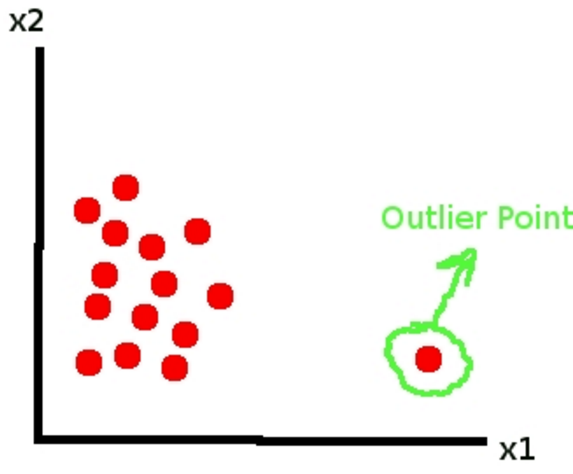
**In this session**

* Anomaly Detection
* One-Class SVM Algorithm
* PCA-Based Algorithm
* Data set
* Data attribute
* Experiment Steps

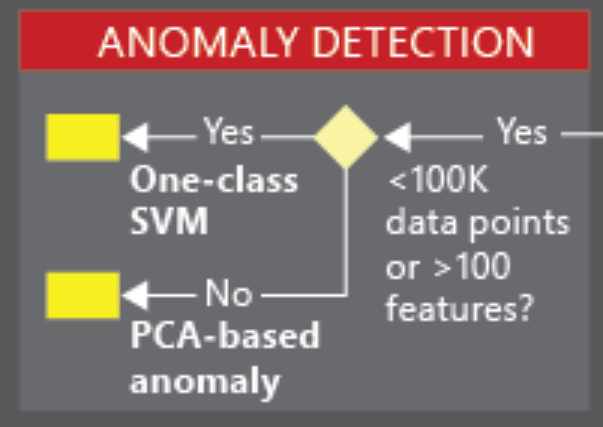
Anomaly Detection

Anomaly Detection

* Credit card fraud, transaction, medical, text etc.
* Also referred to as outliers, novelties, noise, deviations and exceptions
* The data consists of 'normal' applications and 'risky' applications
* Risky transactions = anomalous



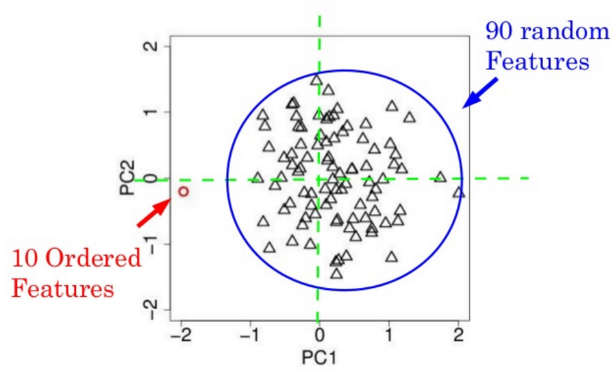
One-Class SVM



One-Class SVM

* SVM = Support Vector Model
* Supervised learning models
* Analyze data and recognize patterns
* Have a lot of "normal" data and not many cases of the anomalies
* Use with Train Anomaly Detection Model
* The train data set contain all or mostly normal cases.

PCA-Based

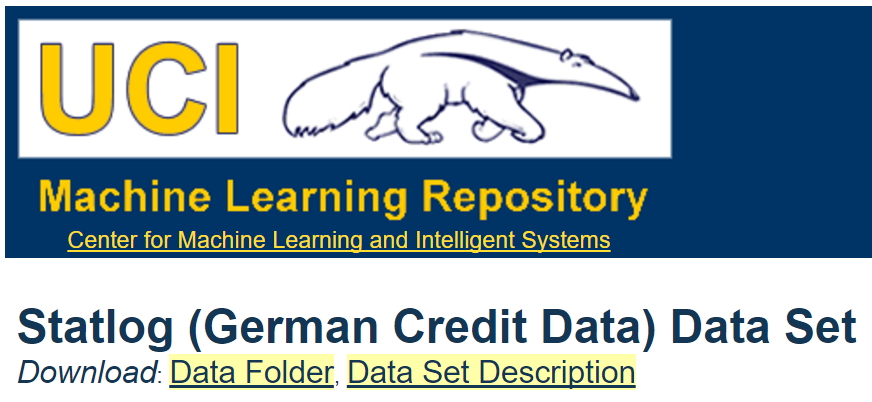
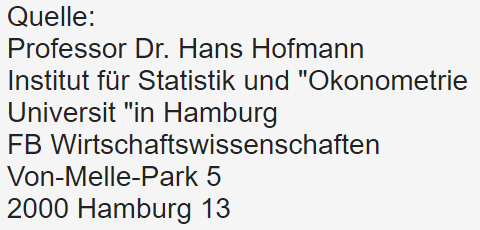


PCA-Based Anomaly Detection module

* Principal Component Analysis (PCA)
* Use when easy to obtain training data from one class
* One class = acceptable transactions
* Use when difficult to obtain sufficient samples of the targeted anomalies
* Detect fraudulent transaction
* You might not have enough examples of fraud to train the mode
* But have many examples of good transactions

Data set

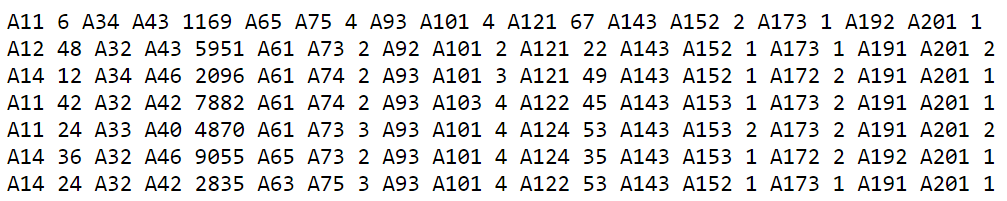
https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)



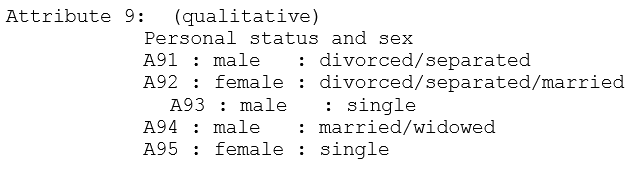
German credit dataset

* Credit card application
* 1000 instances (rows)
* Attributes = 20 (7 numerical, 13 categorical)
* Label 1 = normal, 2 = risky

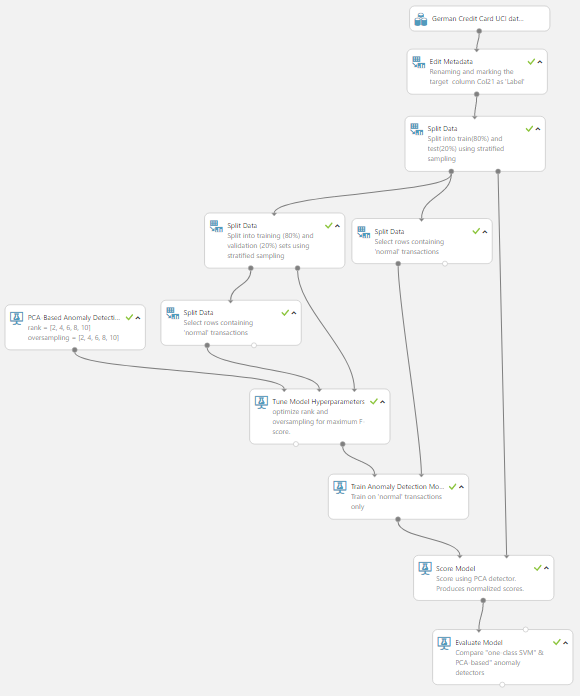
Data attribute



Attribute: Account status, month, credit history, propose, amount, saving, employ since, installment rate, sex …



Experiment Steps



**A**

Experiment steps

**B**

1. Import data set

**C**

1. Edit metadata
2. Split data for training
3. Split data for Score

**E**

**D**

1. Add PCA Base method

**G**

1. Add Tune Model Hyper parameters

**F**

1. Add Train Anomaly Detection Model
2. Add Score model

**H**

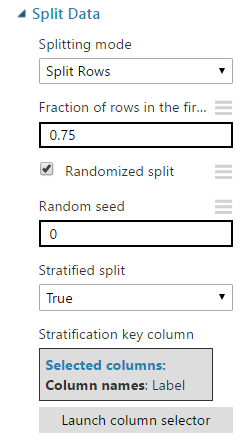
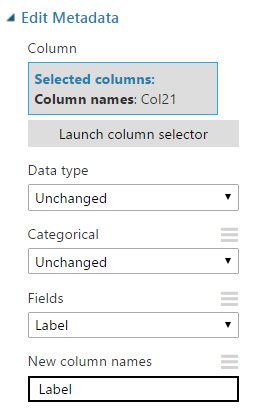
1. Add Evaluate Model

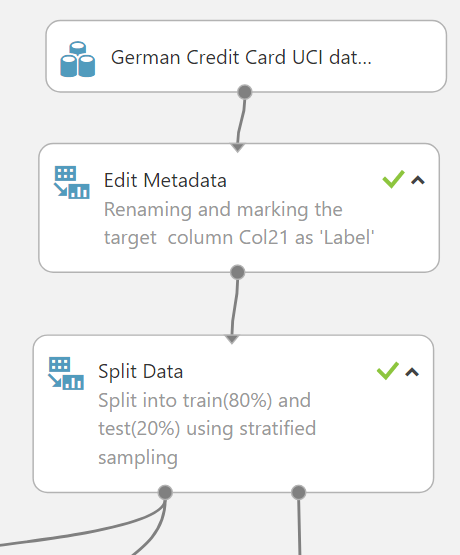
**I**

**J**

**K**

Experiment Steps

1. Import data set
2. Edit metadata
3. Split data for training



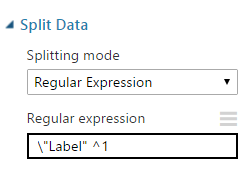
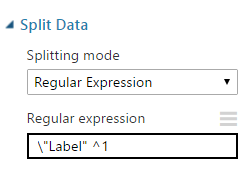
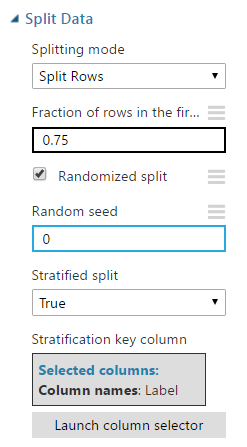
Experiment Steps

Add 4 Split data models and PCA Based

**F**

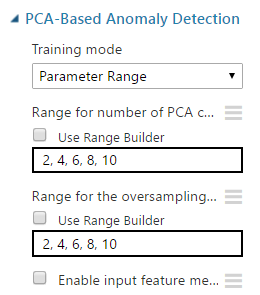
**E**

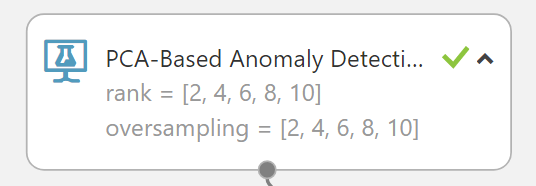
**D**

Experiment Steps

Training mode

* Single Parameter: If you know how you want to configure the model, you can provide a specific set of values as arguments. You might have learned these values by experimentation or received them as guidance.
* Parameter Range: If you are not sure of the best parameters, you can find the optimal parameters by specifying multiple values and using a parameter sweep to find the optimal configuration.

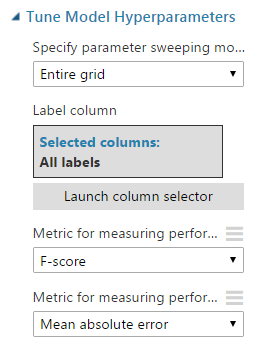


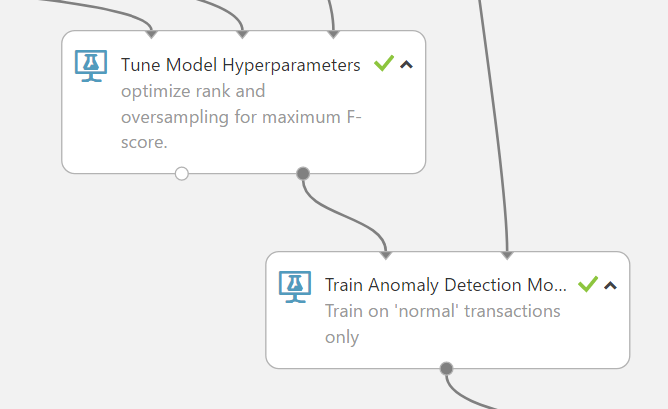
**G**

Experiment Steps

H. Add Tune Model Hyperparameters

I. Add Train Anomaly Detection Model

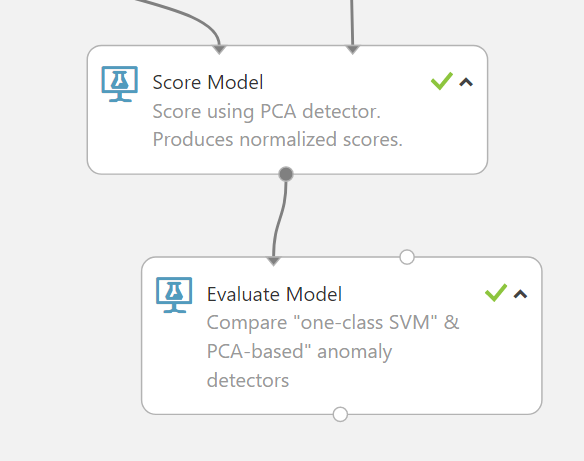
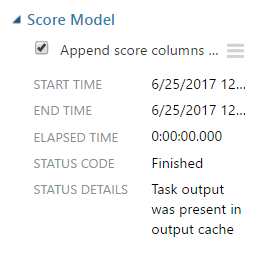




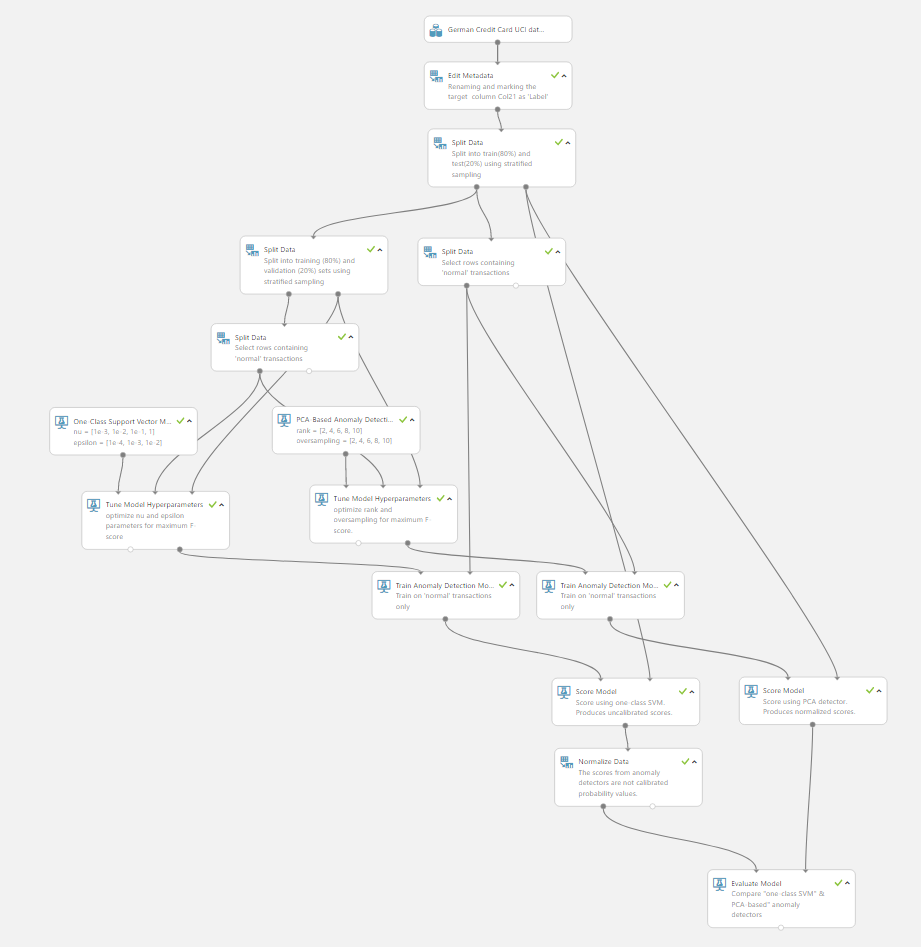
Experiment Steps

J. Add Score Model

K. Add Evaluate Model



Compare two anomaly algorithm



Compare One-Class Support Vector Machine with PCA-Based Anomaly Detection

https://gallery.cortanaintelligence.com/Experiment/Anomaly-compare

More Information

PCA-Based Anomaly Detection

<https://msdn.microsoft.com/en-us/library/azure/dn913102.aspx>

This Experiment

<https://gallery.cortanaintelligence.com/Experiment/Anomaly-Detection-9>

Anomaly compare

<https://gallery.cortanaintelligence.com/Experiment/Anomaly-compare>