Machine Learning in Python

Rohith Mohan

GradQuant

Spring 2018

Interviewer: What's your biggest strength?

Me: I'm an expert in machine learning.

Interviewer: What's 9 + 10?

Me: Its 3.

Interviewer: Not even close. It's 19.

Me: It's 16.

Interviewer: Wrong. Its still 19.

Me: It's 18.

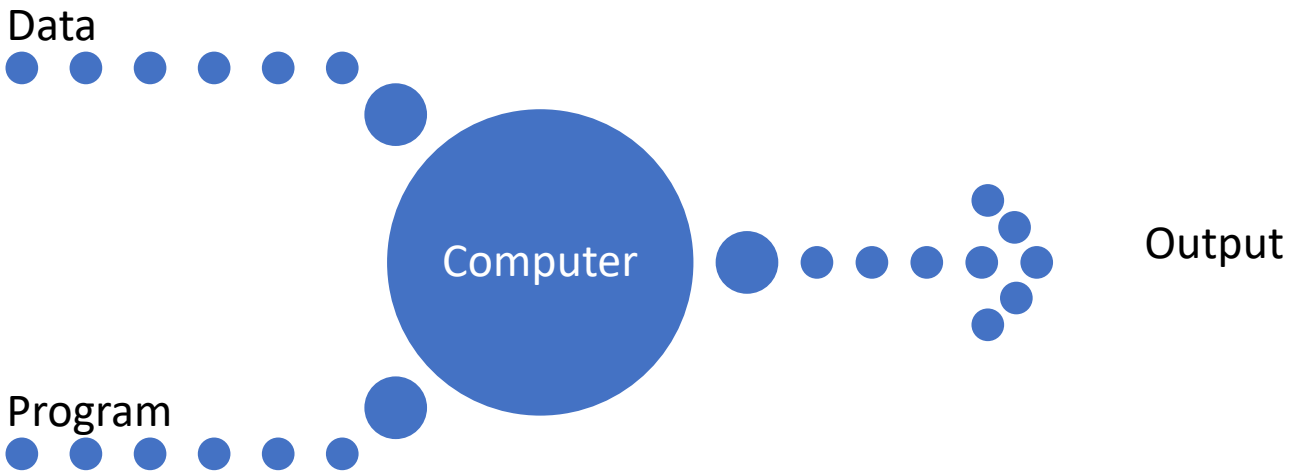
Interviewer: No, it's 19.

Me: it's 19.

Interviewer: You're hired

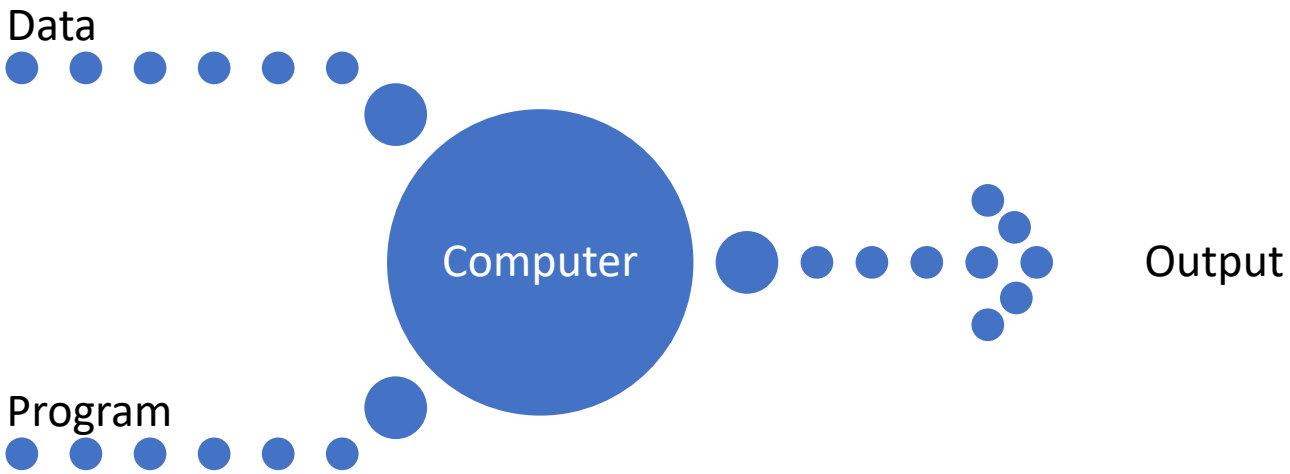
https://twitter.com/myusuf3/status/995425049170489344

Traditional Programming



- Getting computers to program themselves
- Coding is the bottleneck, let data dictate programming

Machine Learning



http://www.hlt.utdallas.edu/~vgogate/ml/2013f/lectures.html

Formal Definitions

- Arthur Samuel (1959)
  - "Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed."
  - Created a program for computer to play itself in checkers (10000s games) and learn at IBM

- Tom Mitchell (1998)
  - "Well-posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

Machine Learning

- Developed out of initial work in Artificial Intelligence (AI)
- Increased availability of large datasets and advances in computing architecture boosted usage in recent times

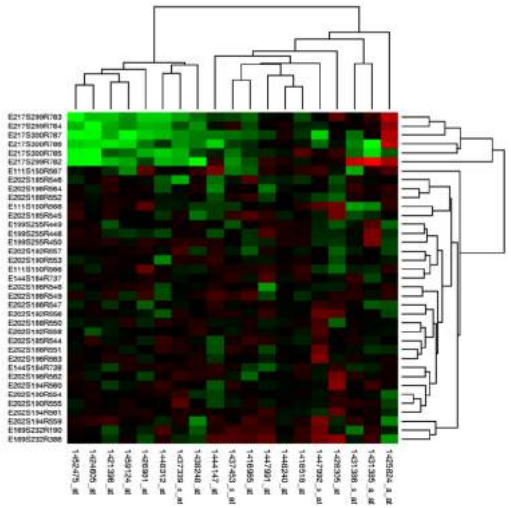
Overview [edit]

Decade	Summary
<1950s	Statistical methods are discovered and refined.
1950s	Pioneering machine learning research is conducted using simple algorithms.
1960s	Bayesian methods are introduced for probabilistic inference in machine learning.[1]
1970s	'AI Winter' caused by pessimism about machine learning effectiveness.
1980s	Rediscovery of backpropagation causes a resurgence in machine learning research.
1990s	Work on machine learning shifts from a knowledge-driven approach to a data-driven approach. Scientists begin creating programs for computers to analyze large amounts of data and draw conclusions – or "learn" – from the results.[2] Support vector machines (SVMs) and recurrent neural networks (RNNs) become popular.
2000s	Kernel methods grow in popularity,[3] and competitive machine learning becomes more widespread.[4]
2010s	Deep learning becomes feasible, which leads to machine learning becoming integral to many widely used software services and applications.

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

Usage

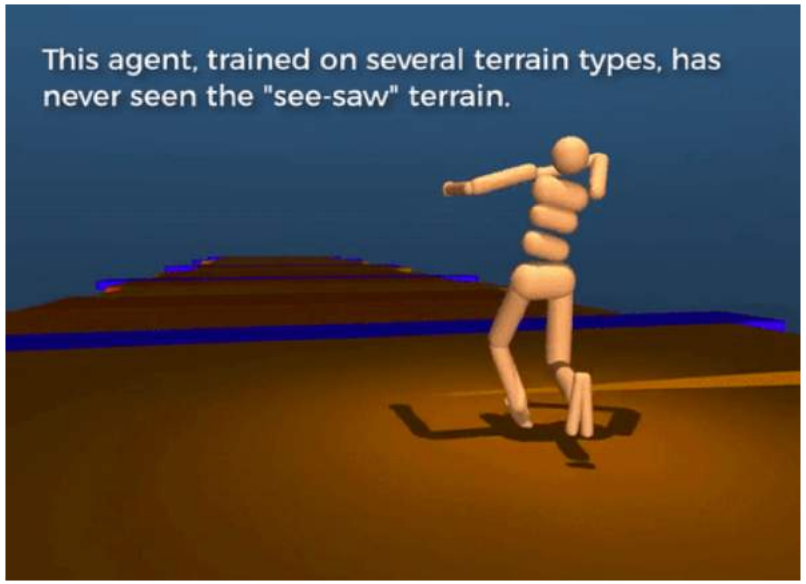
Natural Language Processing
+ Computer Vision



Mining and clustering gene expression data to identify individuals

Recommendation algorithms

Reproducing human behavior (True AI)



https://www.irishnews.com/magazine/science/2018/01/01/news/12-of-the-biggest-scientific-breakthroughs-of-2017-that-might-just-change-the-world-1222695/

http://www.idownloadblog.com/2016/05/12/google-translate-offline-mode/

https://www.flickr.com/photos/theadamclarke/2589233355

https://de.wikipedia.org/wiki/Genexpressionsanalyse

Common steps in ML workflow

- Collect data (various sources, UCI data repository, news orgs, Kaggle)
- Prepare data (exploratory analysis, feature selection, regularization)
- Selecting and training model (train and test datasets, what model?)
- Evaluating model (accuracy, precision, ROC curves, F1 score)
- Optimizing performance (change model, # of features, scaling)

scikit-learn

http://scikit-learn.org/stable/index.html

Preprocessing

- Clean data and deal with missing values, etc.
- Feature scaling - rescaling features to be more sensible
- Standardization - getting various features into similar range (e.g. -1 to 1)
  - Square footage of a house (100s of ft) vs # of rooms (1-5)

```
>>> from sklearn import preprocessing
>>> import numpy as np
>>> X_train = np.array([[ 1., -1.,  2.],
...                     [ 2.,  0.,  0.],
...                     [ 0.,  1., -1.]])
>>> X_scaled = preprocessing.scale(X_train)

>>> X_scaled
array([[ 0.  ..., -1.22...,  1.33...],
       [ 1.22...,  0.  ..., -0.26...],
       [-1.22...,  1.22..., -1.06...]])
```

```
>>> X_scaled.mean(axis=0)
array([ 0.,  0.,  0.])

>>> X_scaled.std(axis=0)
array([ 1.,  1.,  1.])
```

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

Preprocessing

- Clean data and deal with missing values, etc.
- Feature scaling - rescaling features to be more sensible
- Standardization - getting various features into similar range (e.g. -1 to 1)
  - Square footage of a house (100s of ft) vs # of rooms (1-5)
- Normalization – scaling to some standard (e.g. subtract mean & divide by SD)
- Many others (regularization,imputation, generating polynomial features, etc.)

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

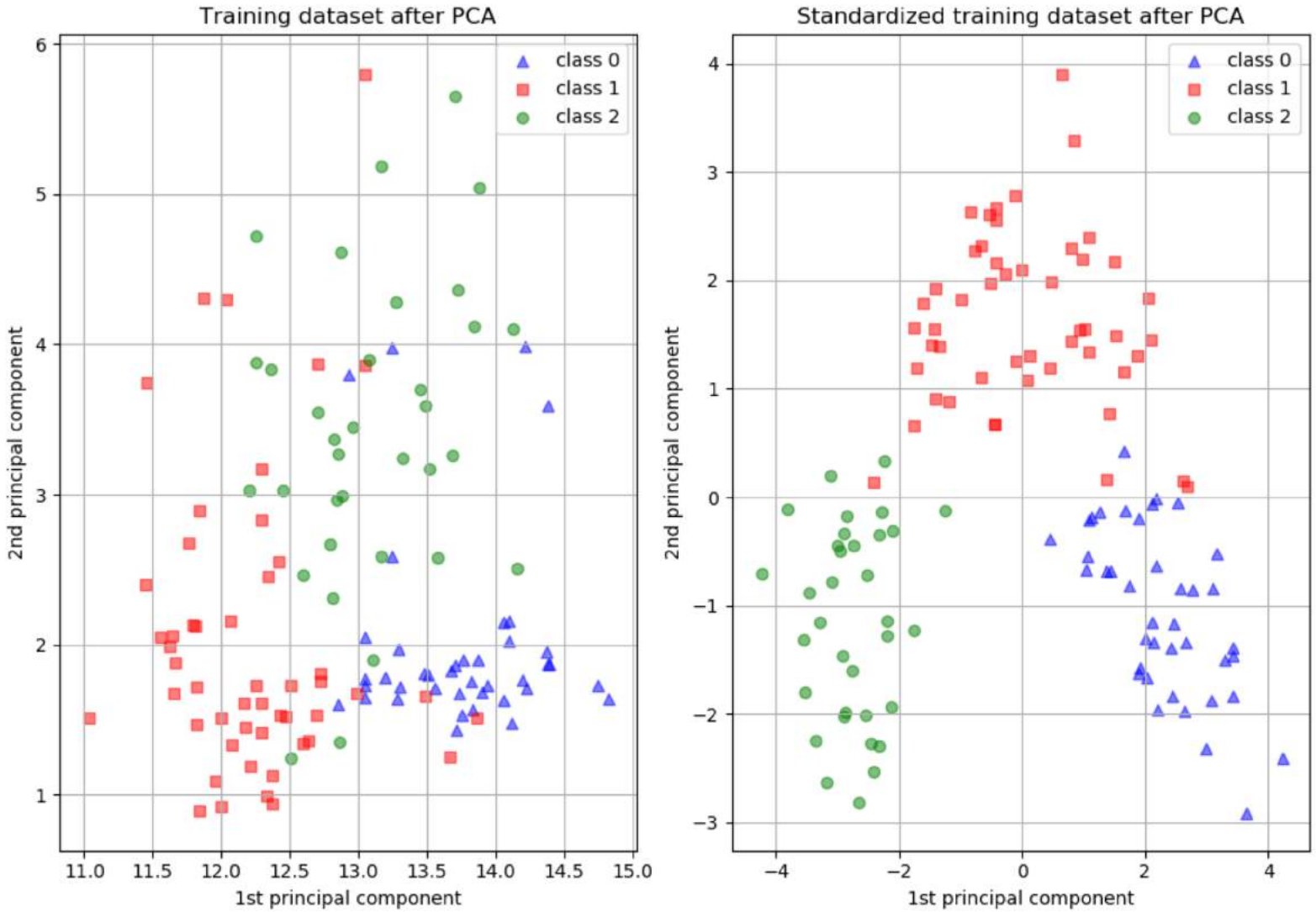
Preprocessing

- Clean data and deal with missing values, etc.
- Feature scaling - rescaling features to be more sensible
- Standardization - getting various features into similar range (e.g. -1 to 1)
  - Square footage of a house (100s of ft) vs # of rooms (1-5)
- Normalization – scaling to some standard (e.g. subtract mean & divide by SD)

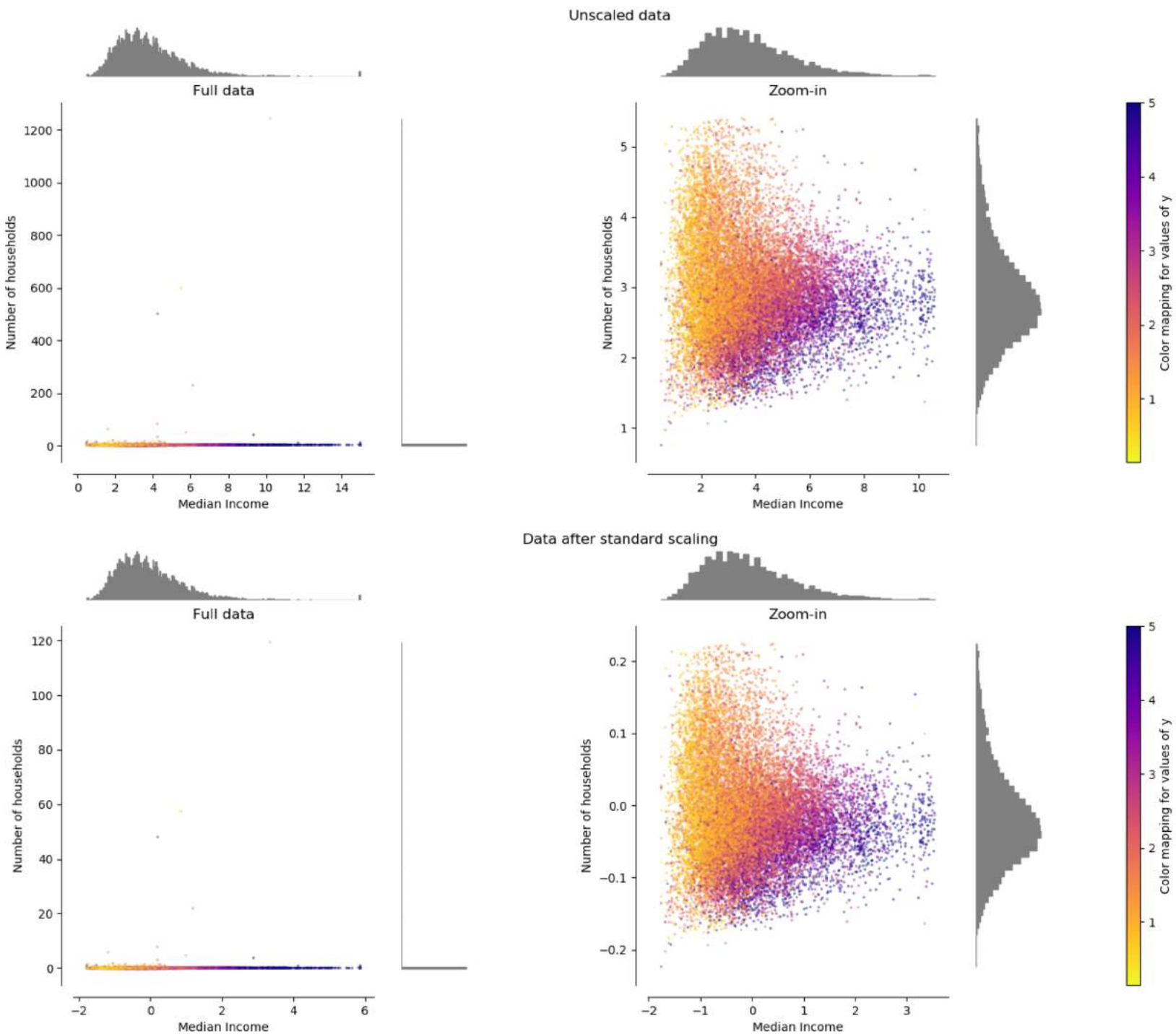
```
>>> X = [[ 1., -1.,  2.],
...      [ 2.,  0.,  0.],
...      [ 0.,  1., -1.]]
>>> X_normalized = preprocessing.normalize(X, norm='l2')

>>> X_normalized
array([[ 0.40..., -0.40...,  0.81...],
       [ 1.  ...,  0.  ...,  0.  ...],
       [ 0.  ...,  0.70..., -0.70...]])
```

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

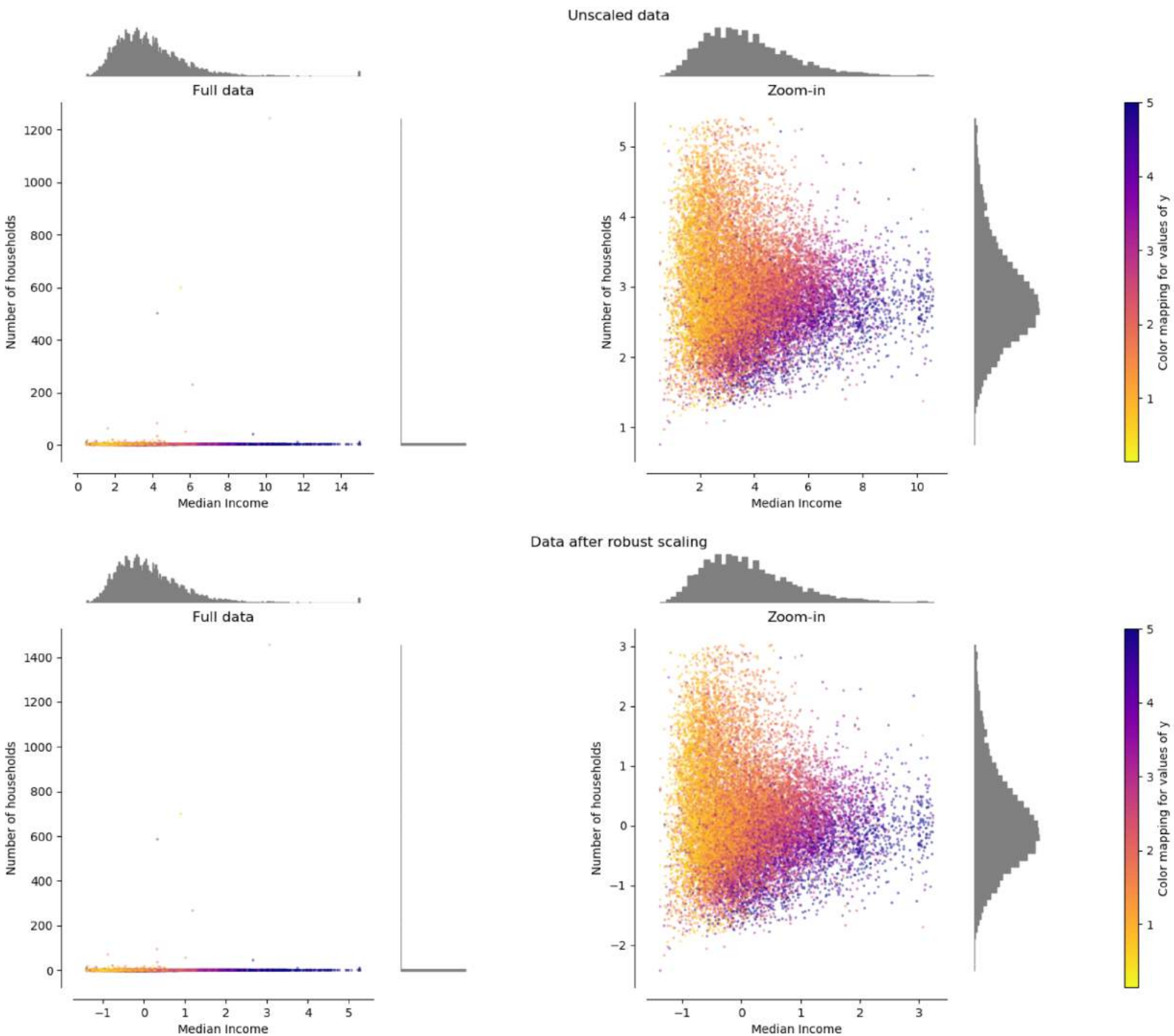
Importance of feature scaling



http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html



StandardScaler

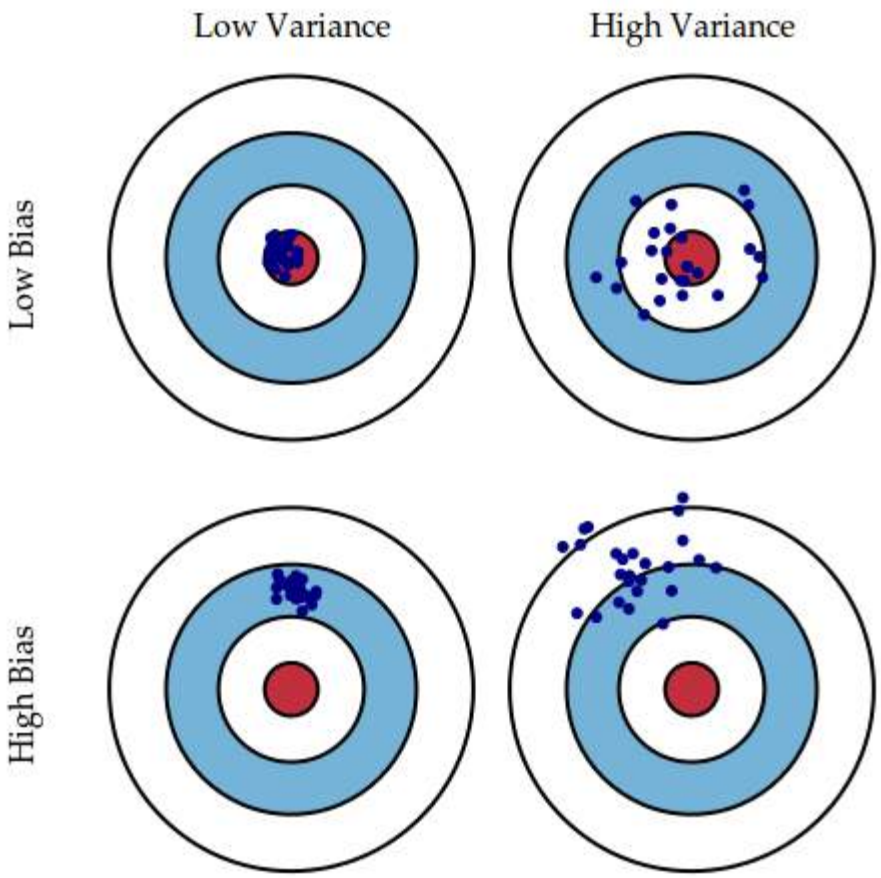


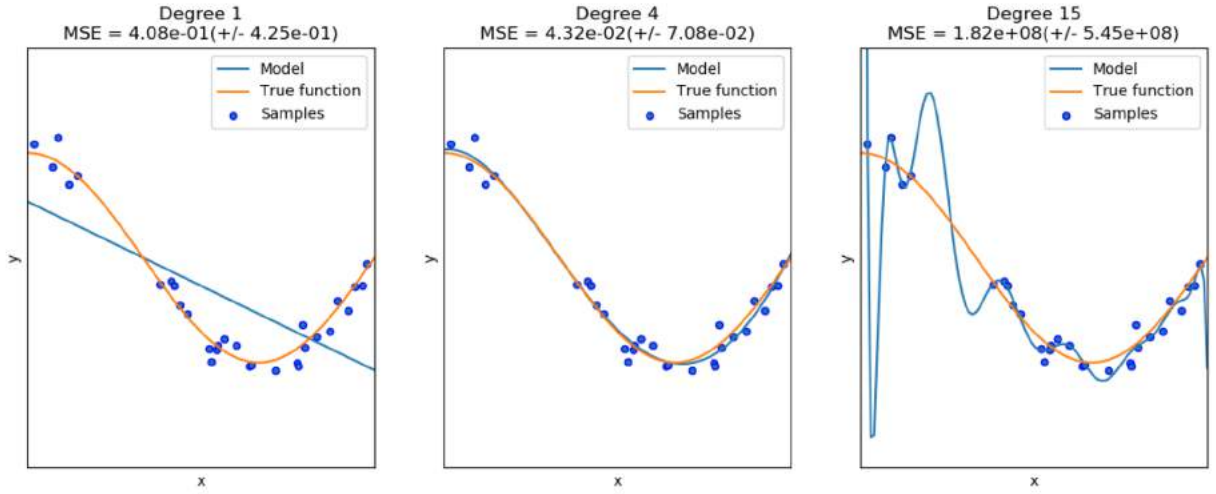
RobustScaler

Train Test (Cross Validate?)

- Why do we need to split up our datasets?
  - Overfitting
- Split dataset
  - Train – for training your model on
  - Test – evaluate performance of model
  - Usually 40% for testing is enough
- Validation set?
- Cross-validation
  - Split up training set into subsets and evaluate performance (can be more computationally expensive but conserves data)
- Hyper-parameter tuning

Bias-variance tradeoff





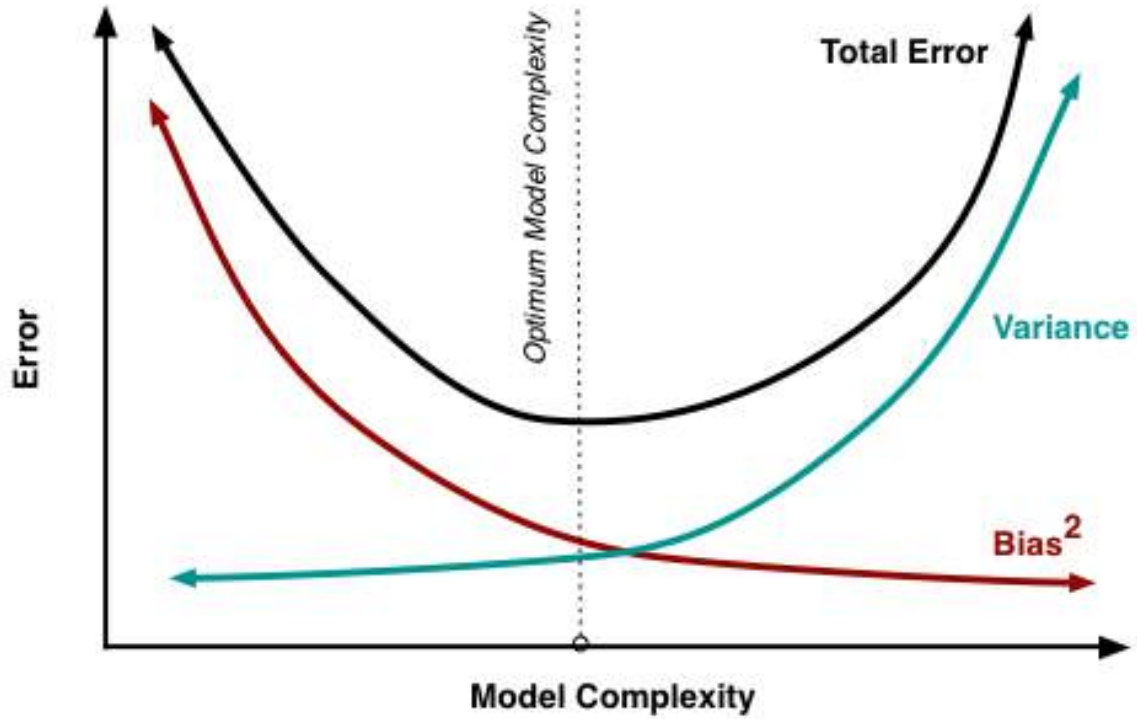
Underfitting
High Bias

Overfitting
High Variance

http://scott.fortmann-roe.com/docs/BiasVariance.html

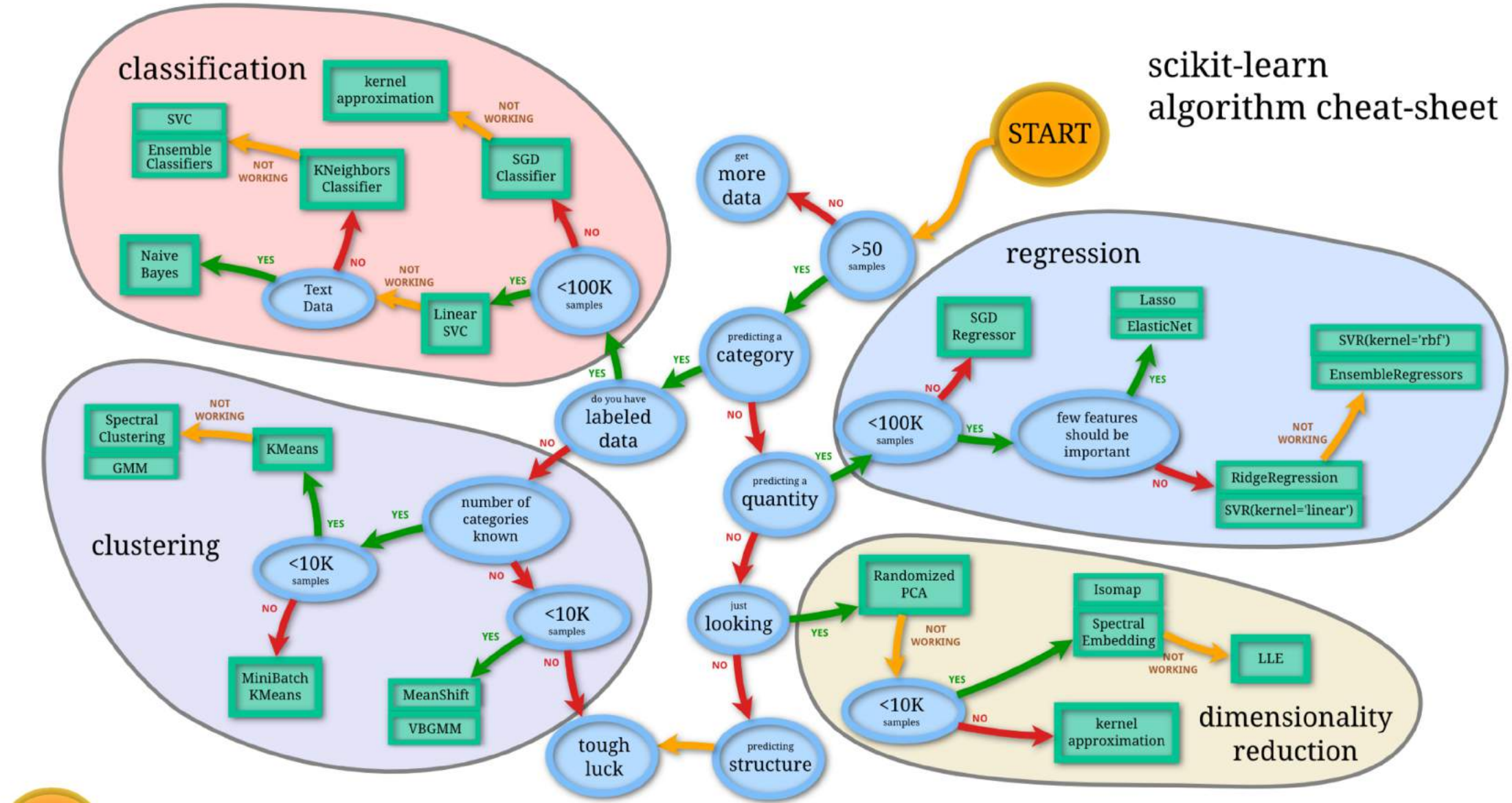
http://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html#sphx-glr-auto-examples-model-selection-plot-underfitting-overfitting-py

Bias-variance tradeoff



http://scott.fortmann-roe.com/docs/BiasVariance.html

How to select a model?



http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Supervised vs Unsupervised Learning

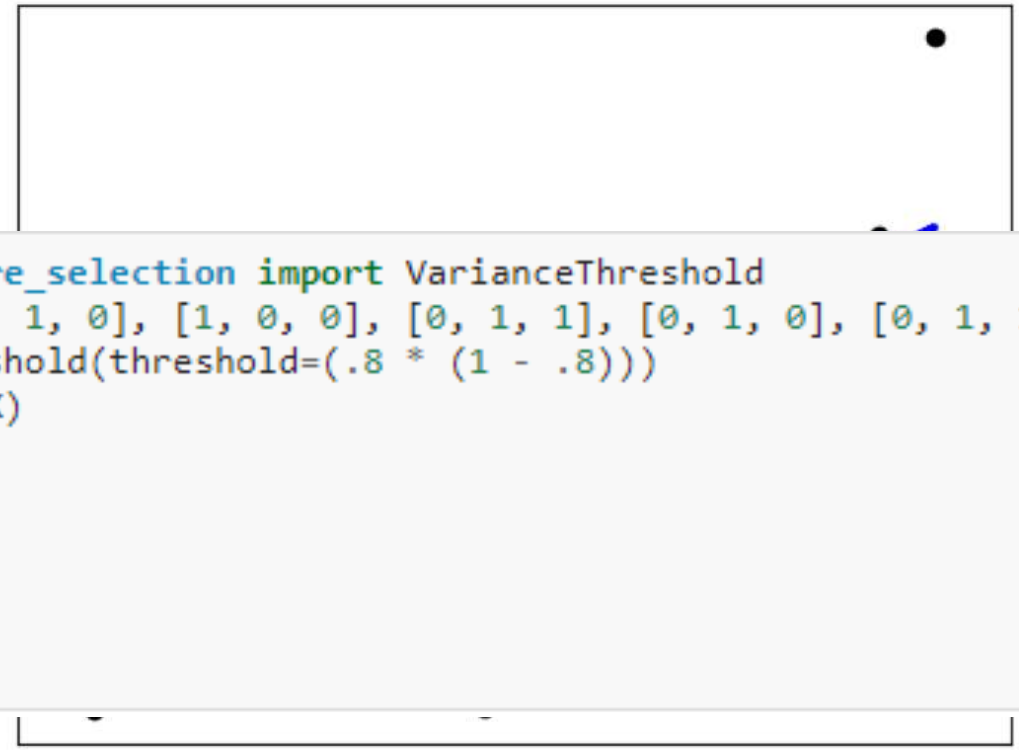
- Supervised
  - Regression, classification
  - Input variables, output variable, learn mapping of input to output
- Unsupervised
  - Clustering, association, etc.
  - No correct answers and no teacher
- Semi-supervised
  - Partially labeled dataset of images
  - Mixing both techniques is what occurs in real-world

Regression

- Linear regression (OLS)

$$Y = \beta_0 + \Sigma_{j=1..p}\, \beta_j X_j + \varepsilon$$

- Prediction
- Multiple variables/f
  - Feature selection

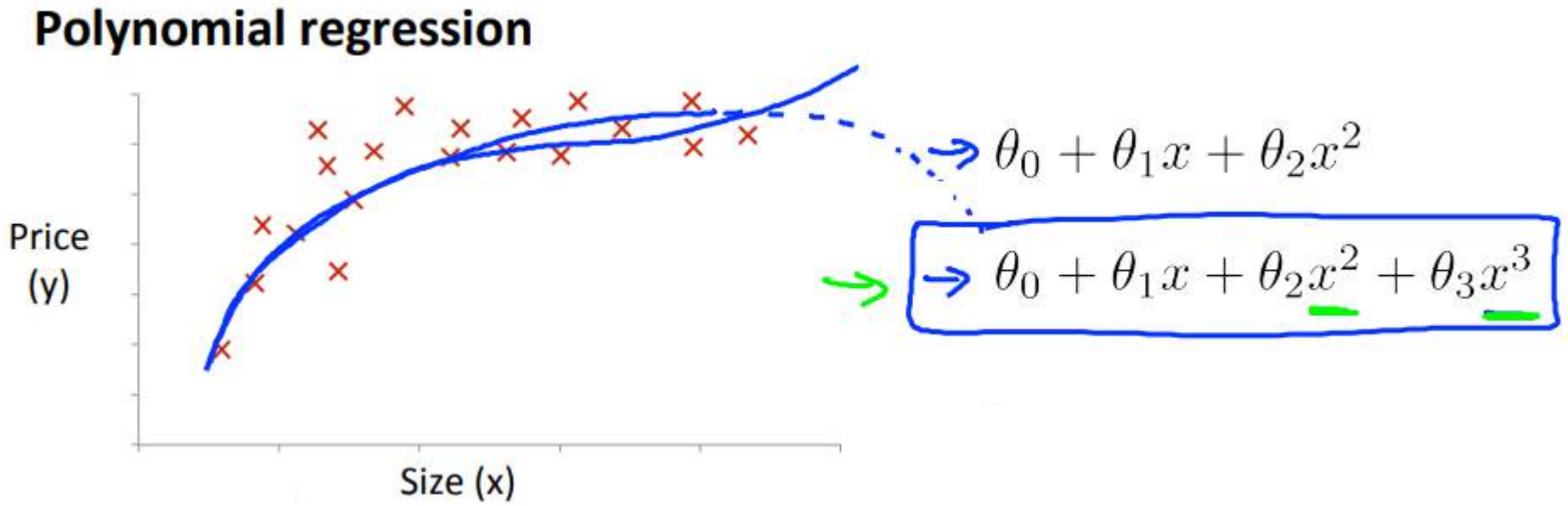


```
>>> from sklearn.feature_selection import VarianceThreshold
>>> X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
>>> sel = VarianceThreshold(threshold=(.8 * (1 - .8)))
>>> sel.fit_transform(X)
array([[0, 1],
       [1, 0],
       [0, 0],
       [1, 1],
       [1, 0],
       [1, 1]])
```

https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols

http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html

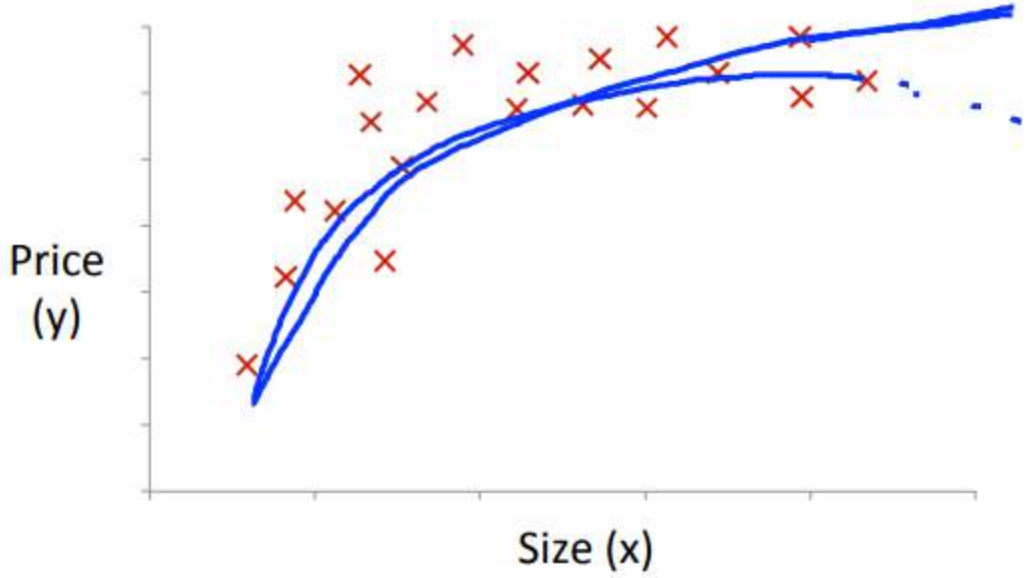
Feature Selection



https://www.coursera.org/learn/machine-learning

Feature Selection

Choice of features



$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2(size)^2$$

$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2\sqrt{(size)}$$

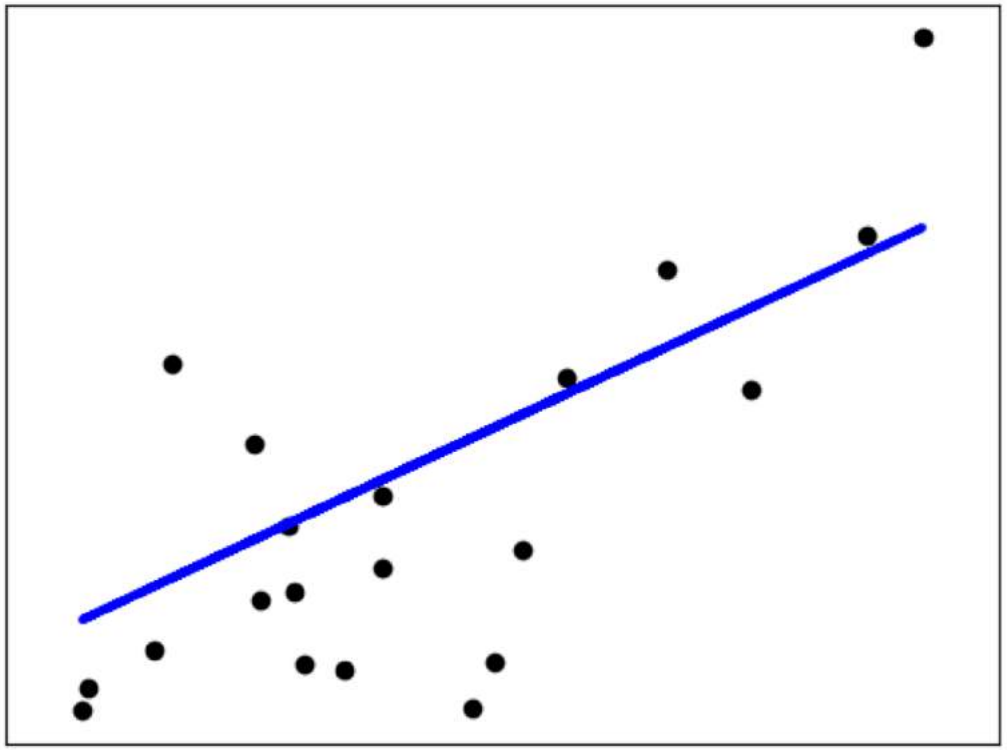
https://www.coursera.org/learn/machine-learning

Regression

- Linear regression (OLS)

$$Y = \beta_0 + \Sigma_{j=1..p}\, \beta_j X_j + \varepsilon$$

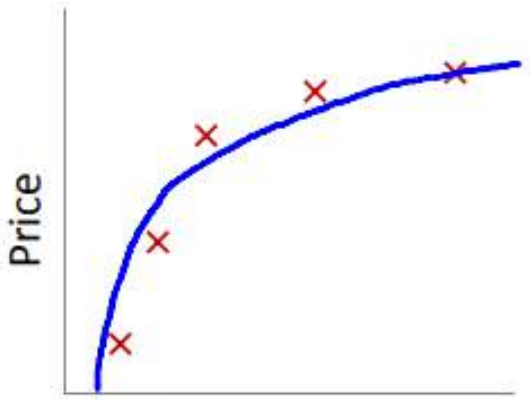
- Prediction
- Multiple variables/features?
  - Feature selection
  - Length, width of a house (area?)
  - Regularization



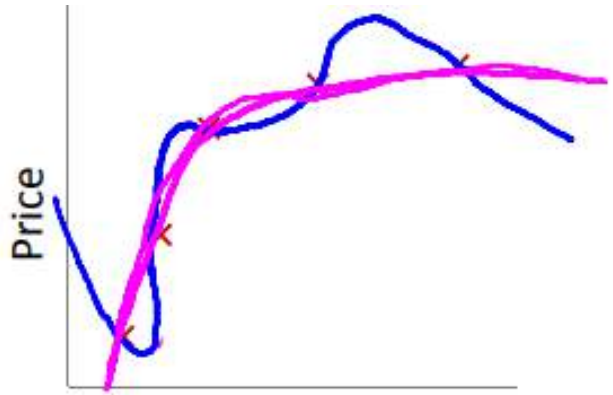
https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols

http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html

Regularization



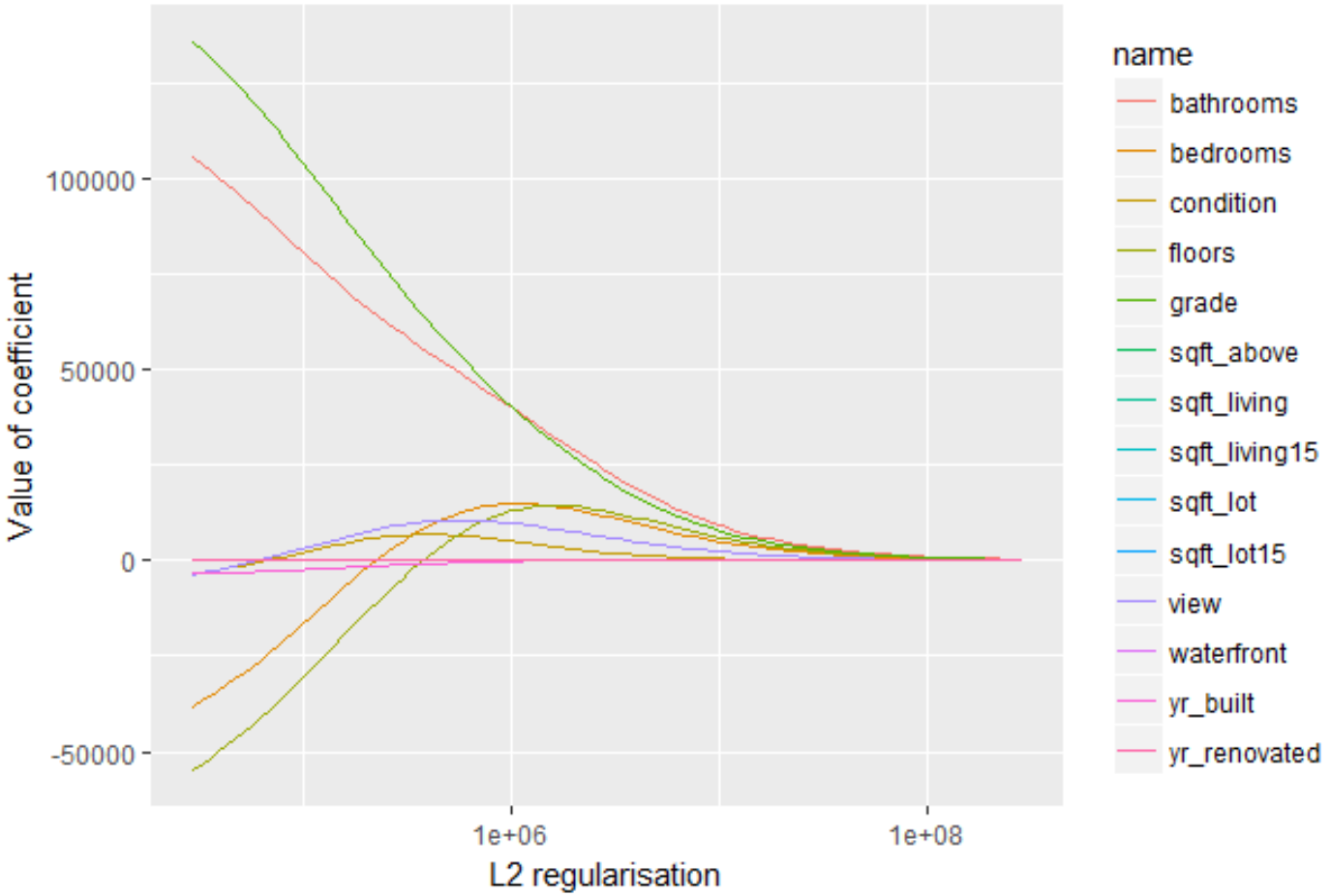
$$\theta_0 + \theta_1 x + \theta_2 x^2$$



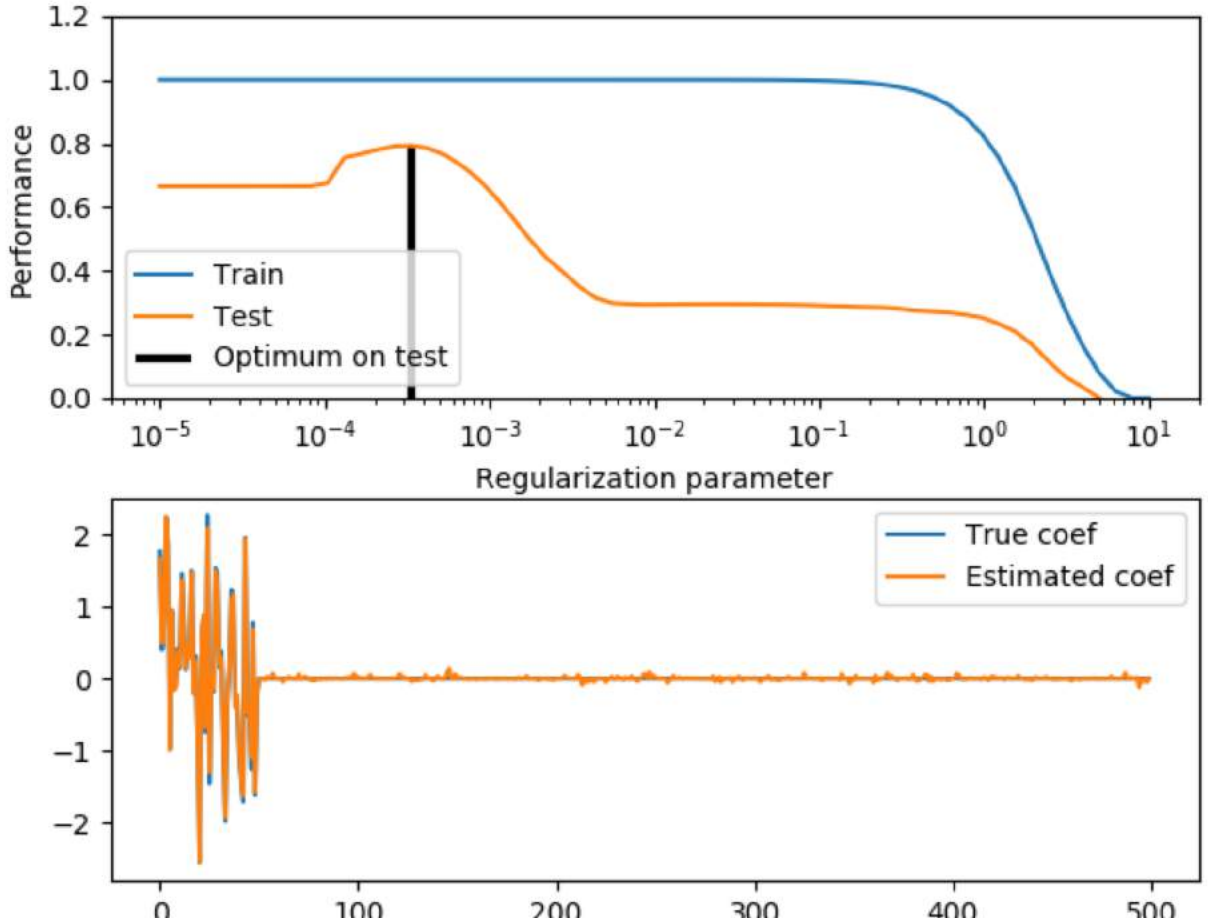
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

https://www.coursera.org/learn/machine-learning

Regularization

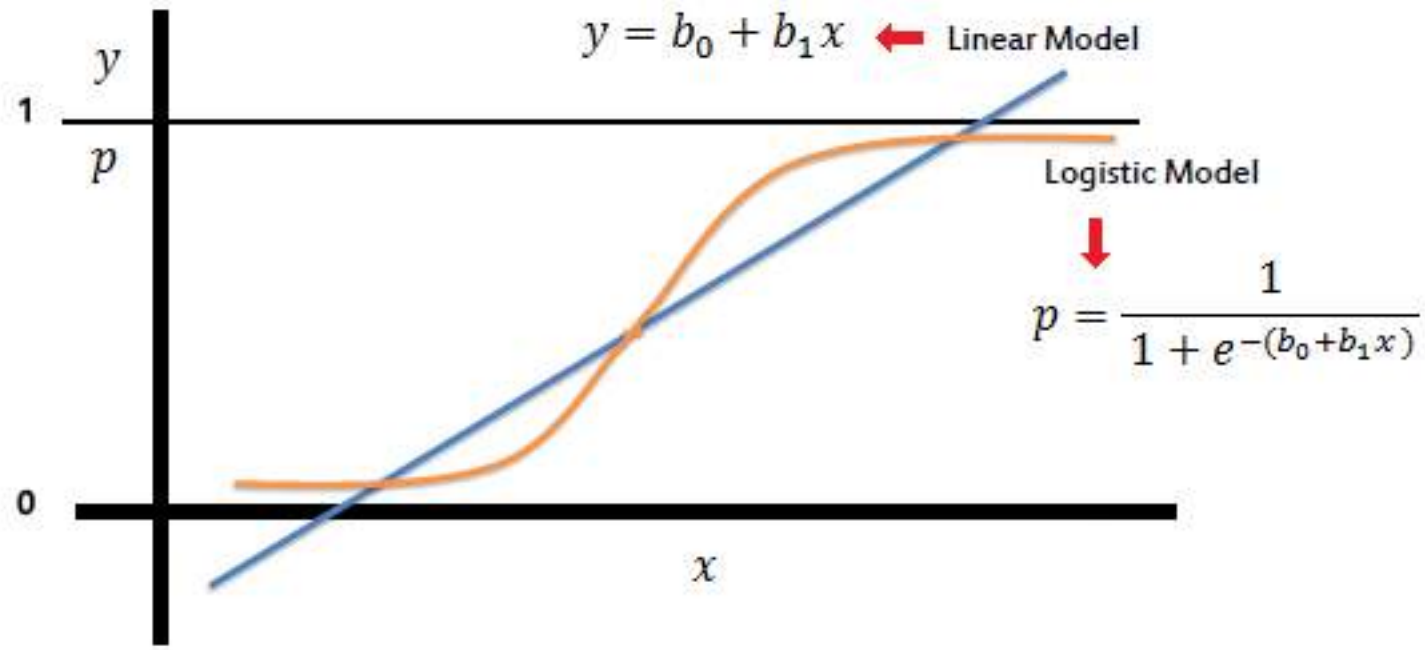


http://enhancedatascience.com/2017/07/04/machine-learning-explained-regularization/



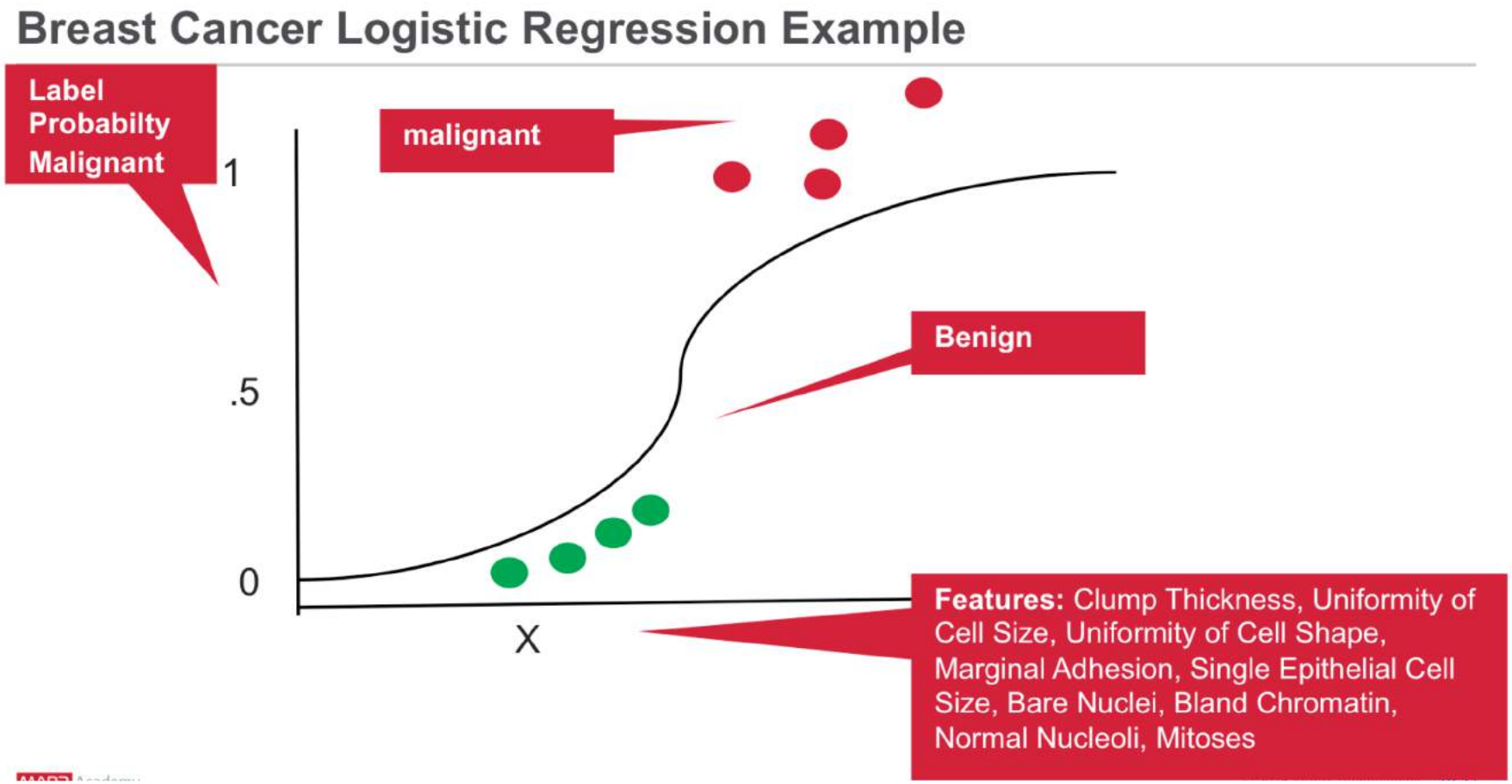
http://scikit-learn.org/stable/auto_examples/model_selection/plot_train_error_vs_test_error.html#sphx-glr-auto-examples-model-selection-plot-train-error-vs-test-error-py

Classification – Logistic Regression



http://www.saedsayad.com/logistic_regression.htm

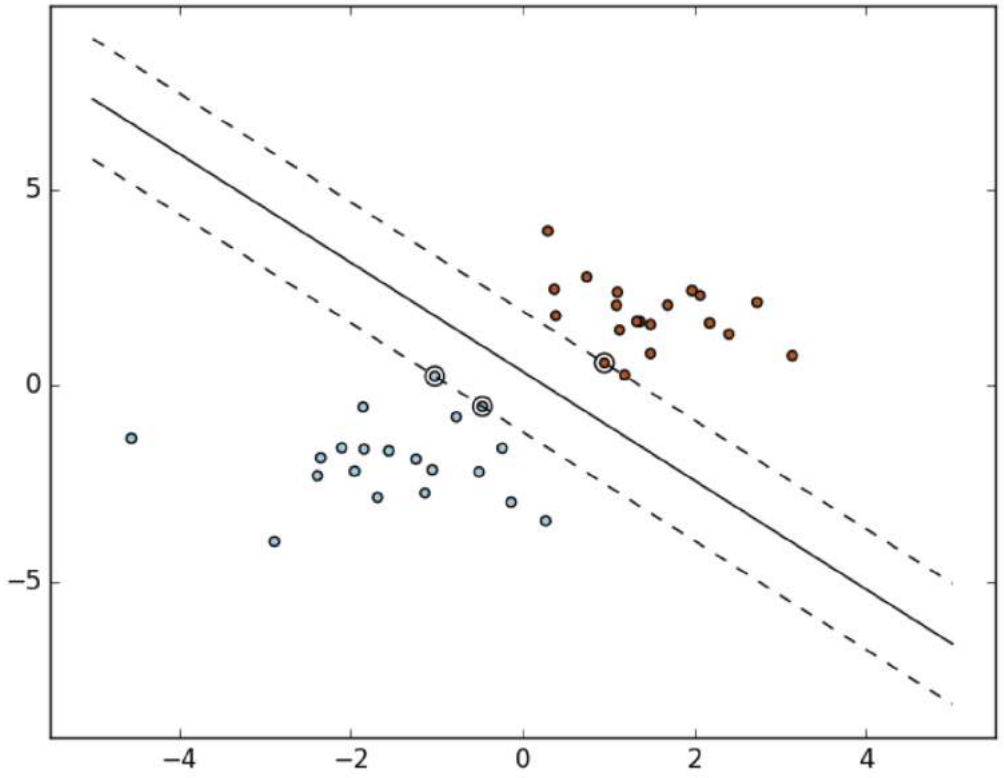
Classification – Logistic Regression



https://medium.com/technology-nineleaps/logistic-regression-bac1db38cb8c

https://mapr.com/blog/predicting-breast-cancer-using-apache-spark-machine-learning-logistic-regression/

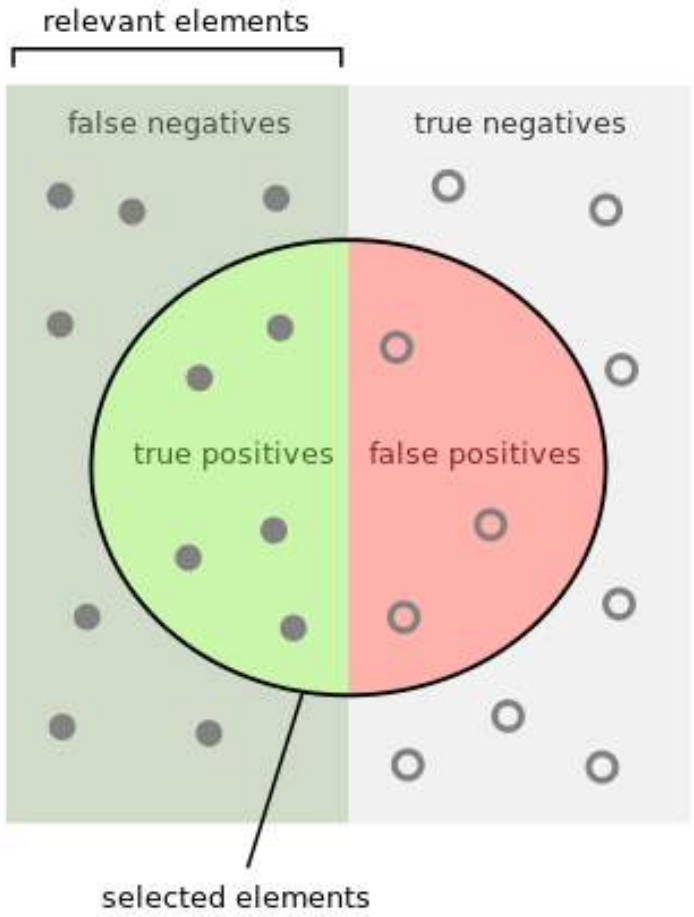
Classification – SVM

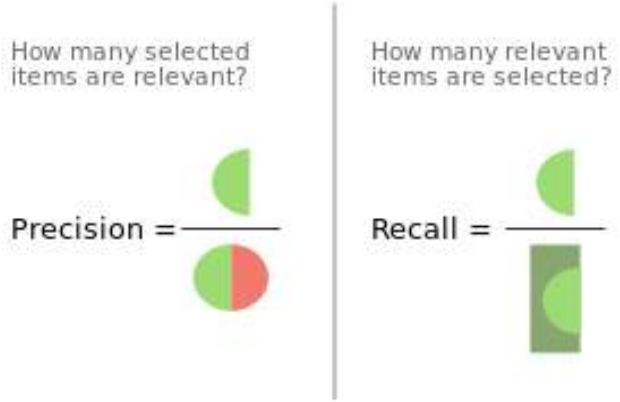


http://scikit-learn.org/0.18/auto_examples/svm/plot_separating_hyperplane.html

Evaluating Performance

- Accuracy – how many predictions are corre dataset?
  - Can be a flawed metric
- Precision and Recall



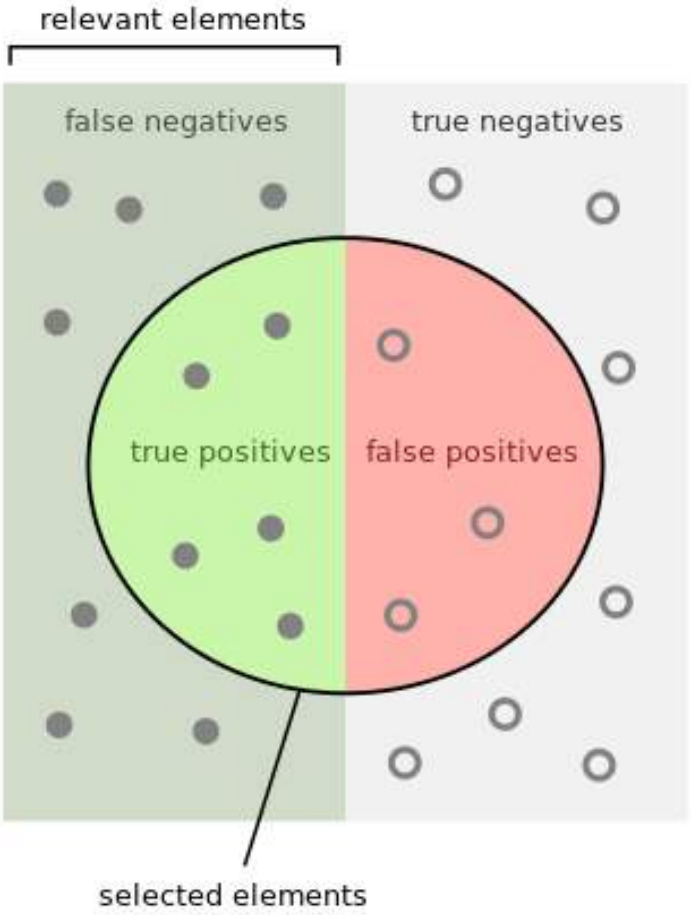


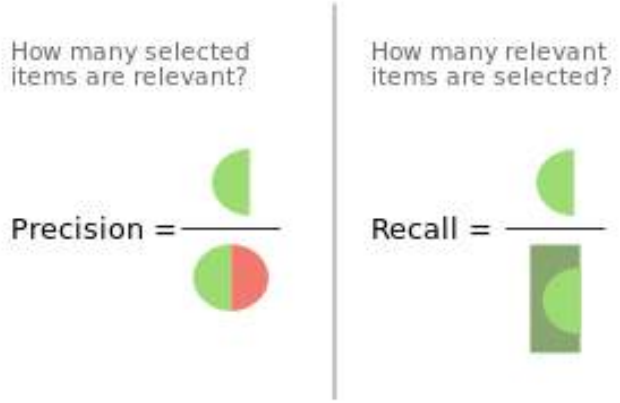
https://en.wikipedia.org/wiki/Precision_and_recall

Evaluating Performance

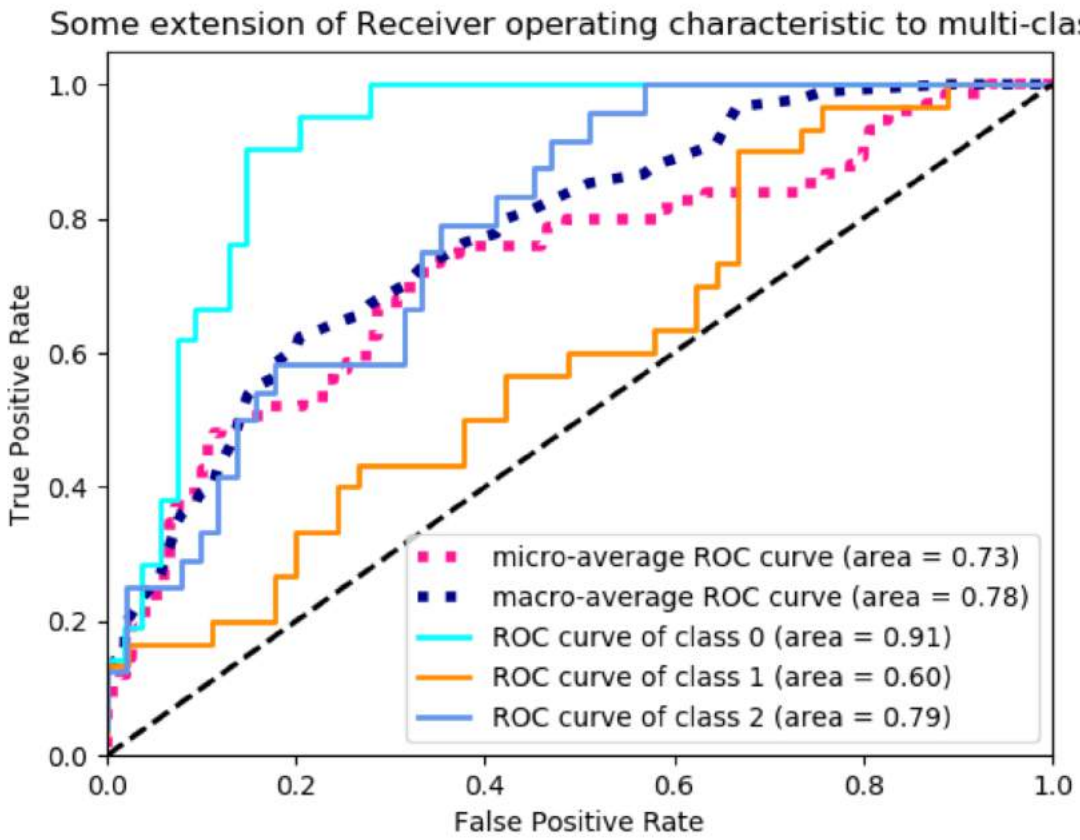
- Accuracy – how many predictions are corre dataset?
  - Can be a flawed metric
- Precision and Recall
- ROC curves
- F1 score

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$



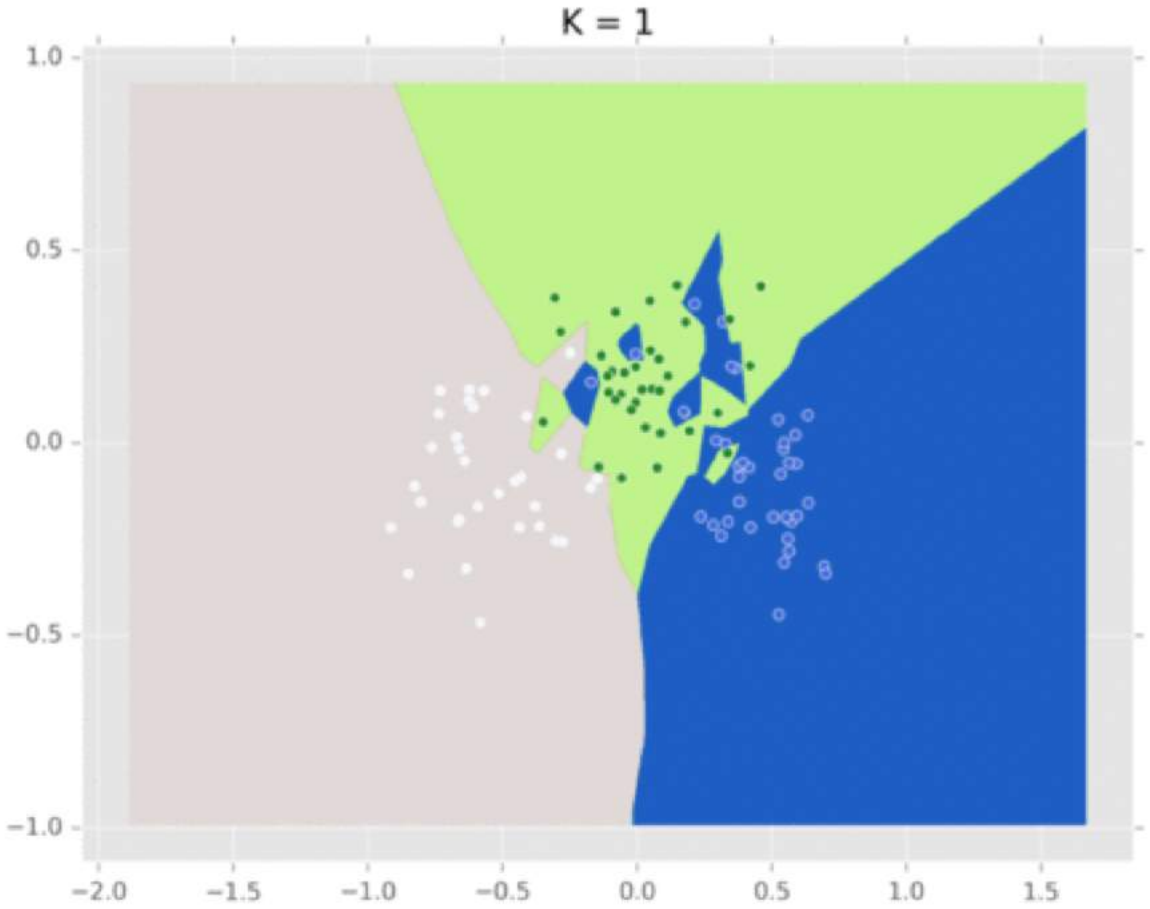


Evaluating Performance

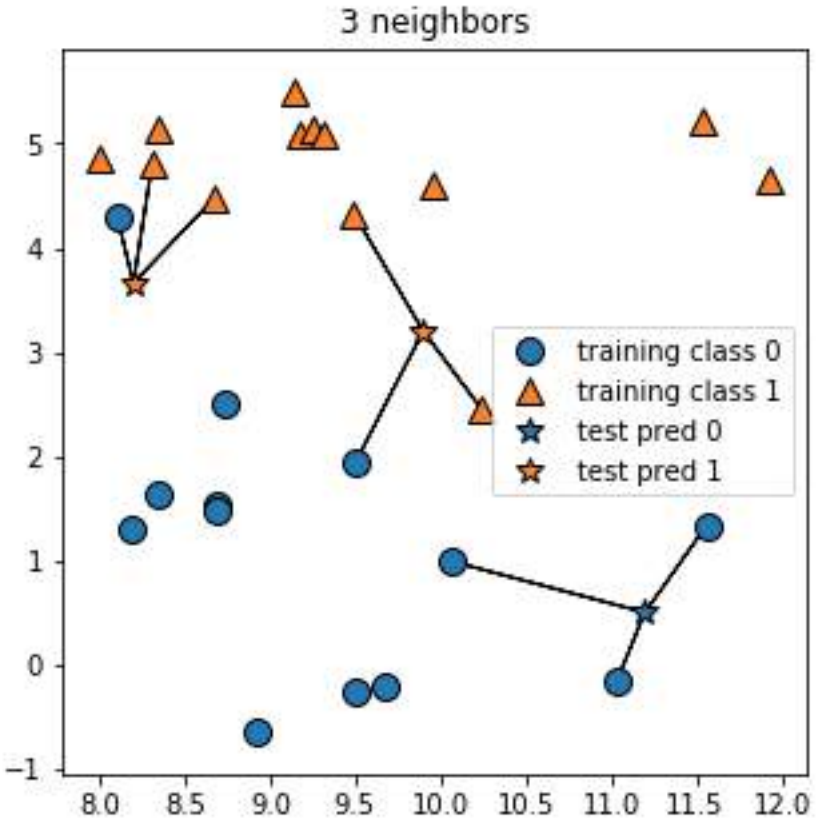


http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html#sphx-glr-auto-examples-model-selection-plot-roc-py

Classification - K-Nearest Neighbors



ber



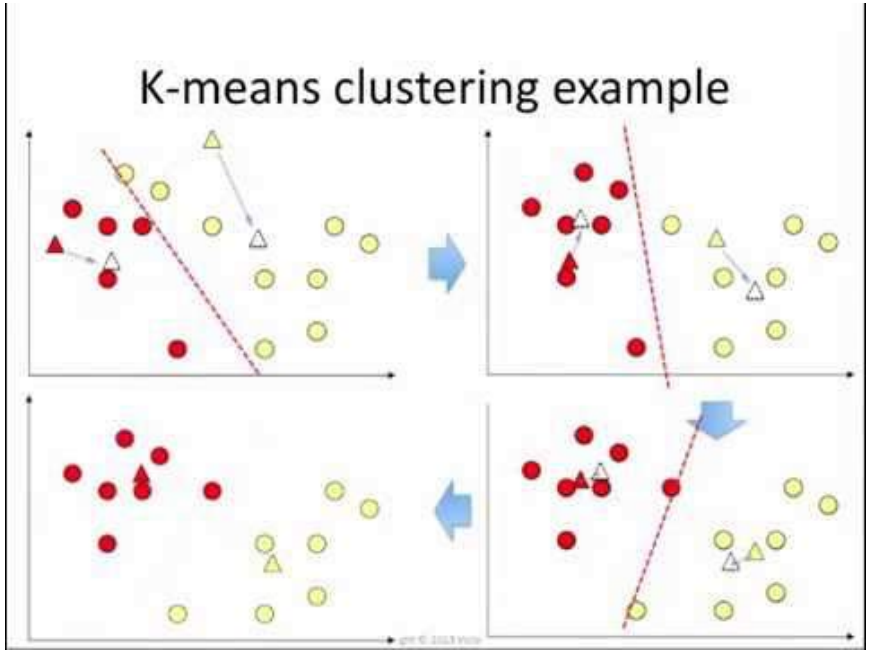
Clustering

- Unsupervised learning
- Can help you understand structure of your data

- Various types of clustering: K-means, Hierarchical, Ward

K-means

- Randomly choose k centroids
- Form clusters around it
- Take mean of cluster to identify new centroid
- Repeat until convergence



https://www.youtube.com/watch?v=_aWzGGNrcic