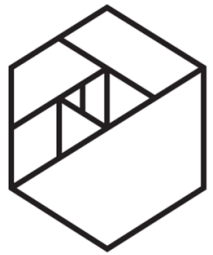


Course wrap up

INTRODUCTION TO DATA SCIENCE – FALL 2018

SESSION 12

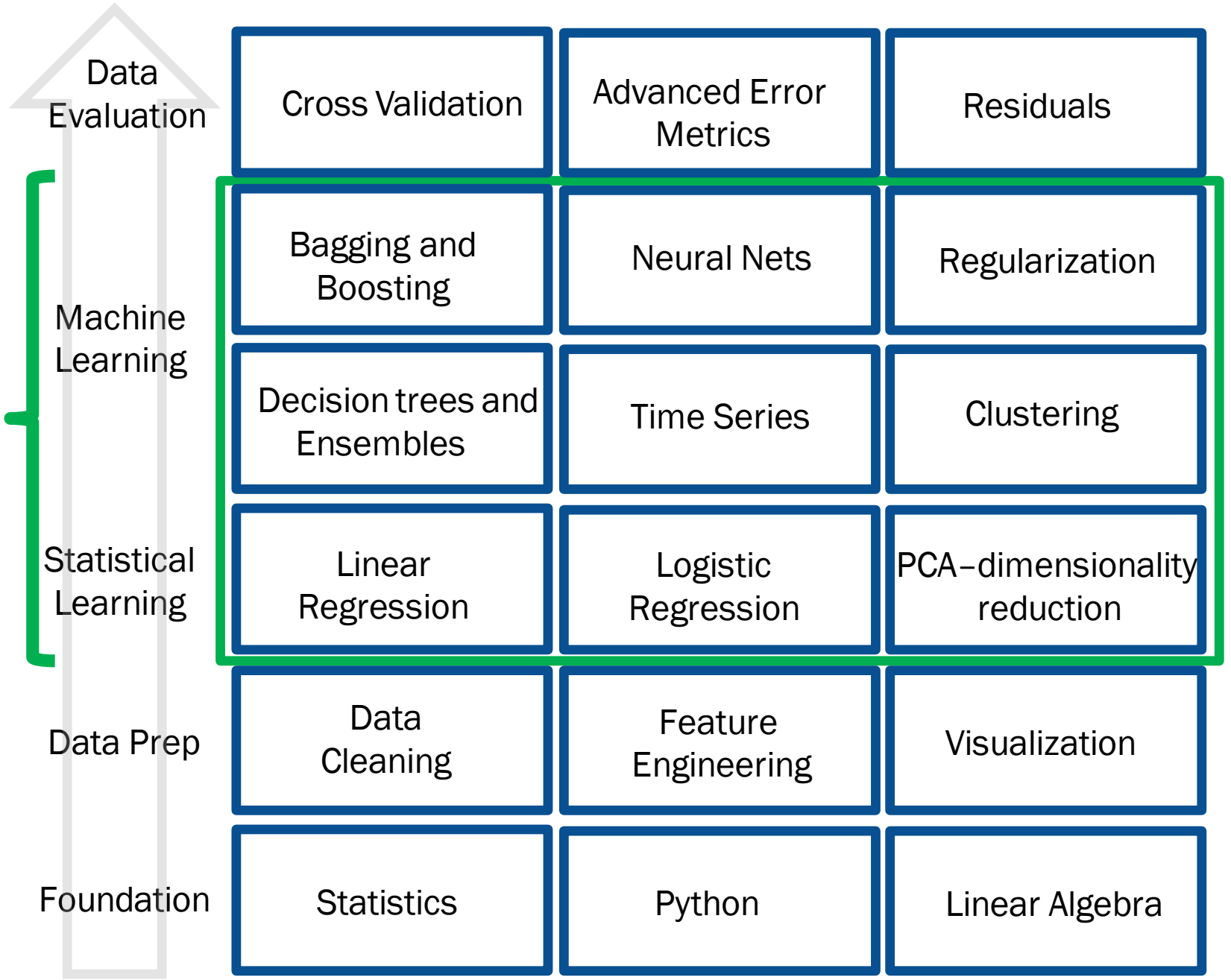
AGENDA

Session 12

1. Data Science Process
2. Supervised Learning
3. Unsupervised Learning
4. Dimensionality Reduction

Introduction to Data Science

- Learning the steps in the Data Science Process
- Learning multiple model methodologies

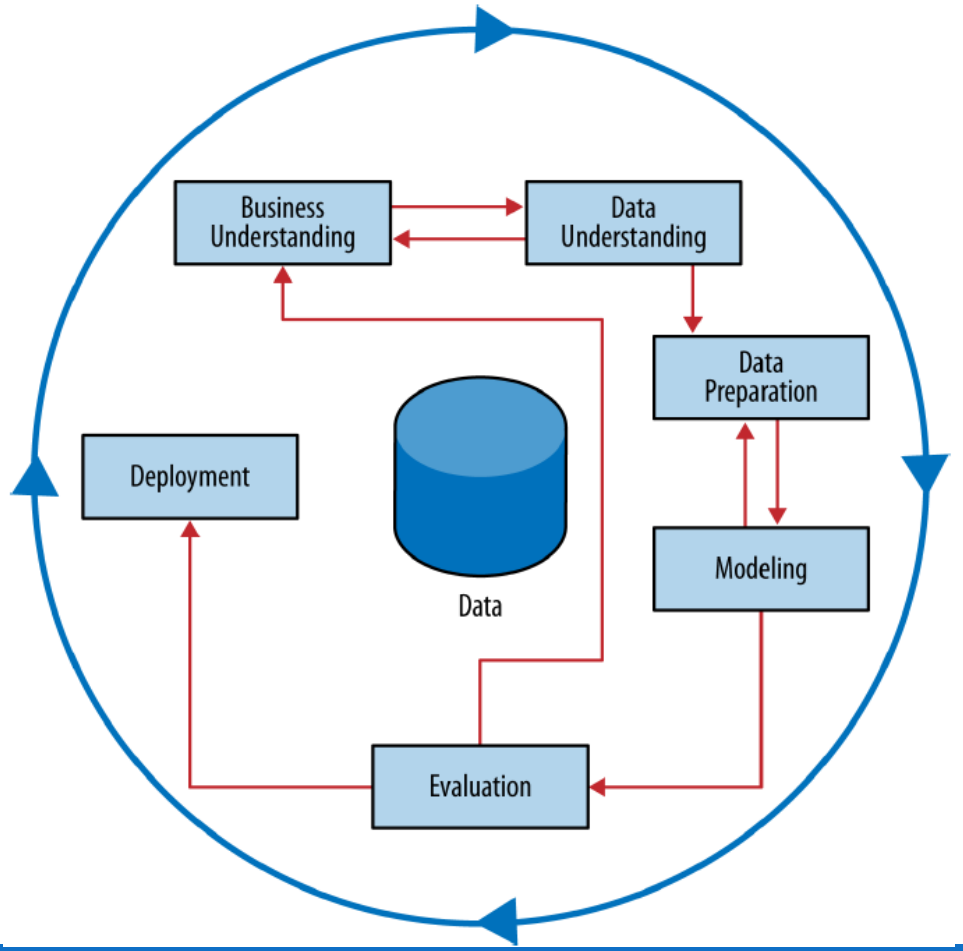


Data science process

ANSWERING THE BUSINESS PROBLEM

Business process

- CRISP-DM
- Cross Industry Standard Process for Data Mining
- Circular process allows for the iterative process of data science
  - Data Preparation
  - Modeling
  - Evaluation



Pre-processing the data

- Standard scaling – mean and unit variance
- Min max scaling – results between 0 and 1
- Feature transformations – reduce skew in feature distributions
- Handling categorical features – using dummy variables
- Handling missing values - imputation

Which models require scaling?

Regularized regression

Linear classifiers

Principle Components Analysis

Clustering Methods

Decision Trees

Random Forest

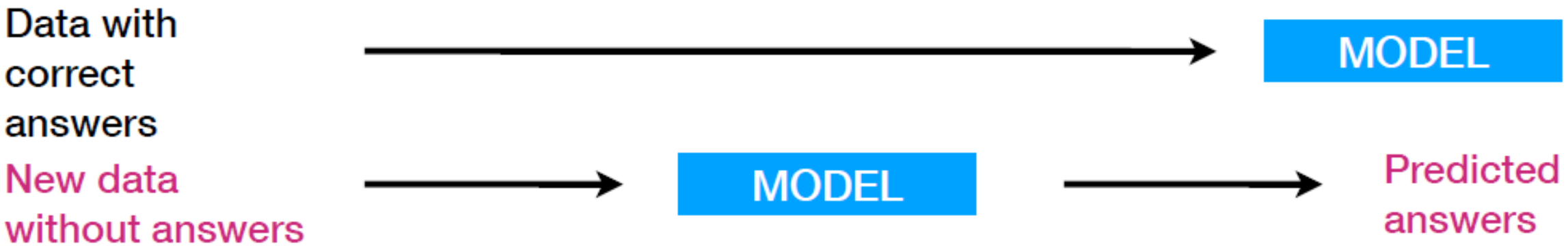
Boosted trees

Modeling

PART 1 – SUPERVISED LEARNING

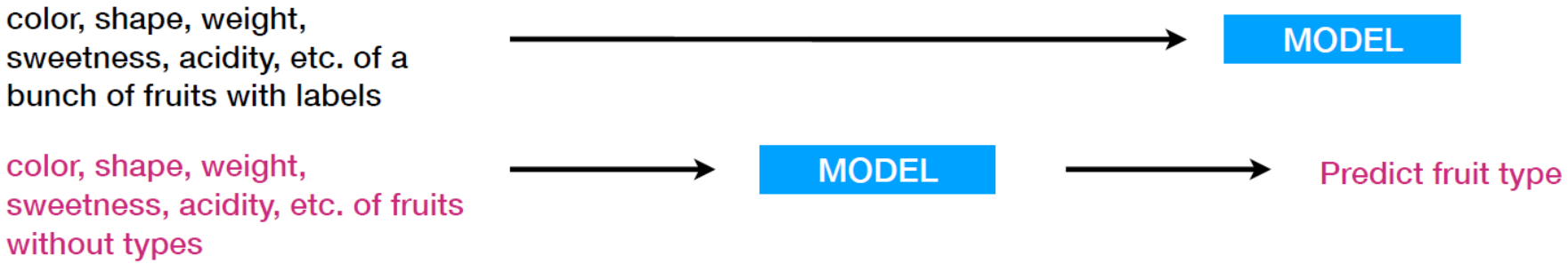
Supervised learning problems

- Involve constructing an accurate model that can predict some kind of an outcome when past data has labels for those outcomes

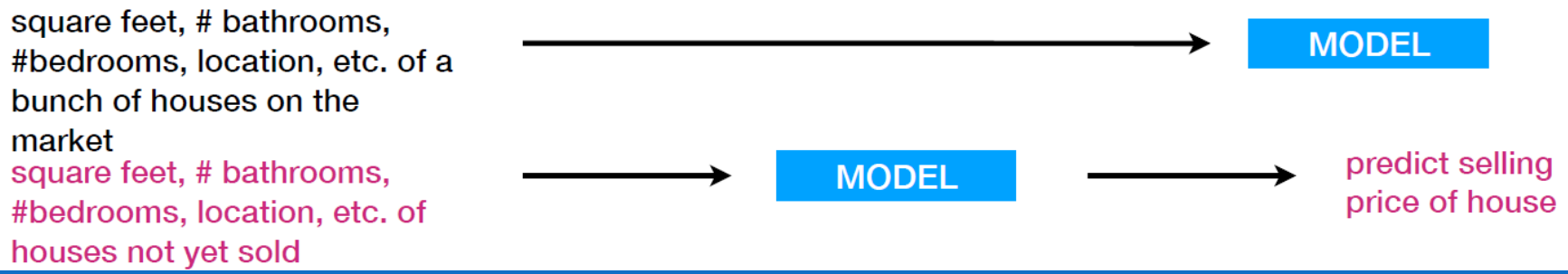


Regression vs Classification

- A **classification problem** is a **supervised learning problem** where the objective is to learn to predict a categorical value.



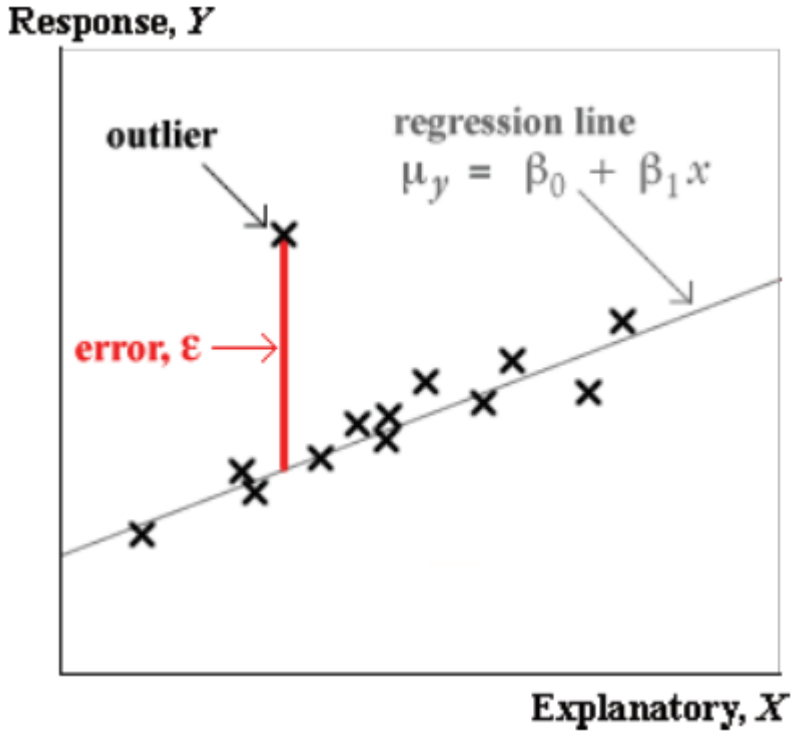
- A **regression problem** is a **supervised learning problem** where the objective is to learn to predict a continuous value.



Linear Model for regression

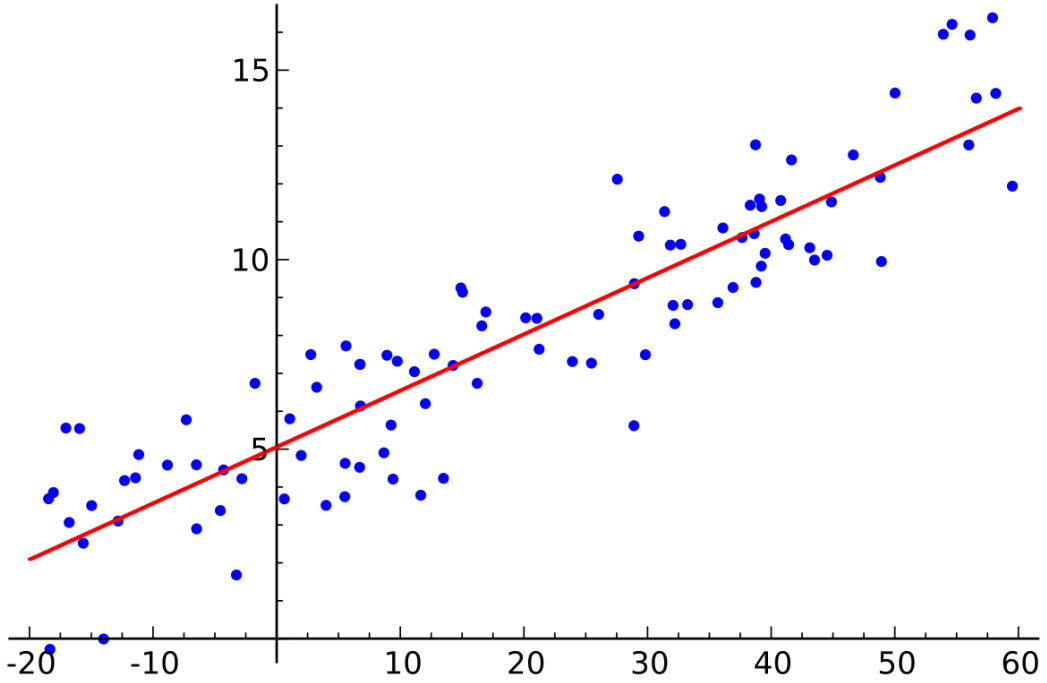
$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

- Finding the line of best fit for our data
- This line can give us predictions for any given set of inputs
- This model is simple and interpretable
- We must abide by the assumptions of the model



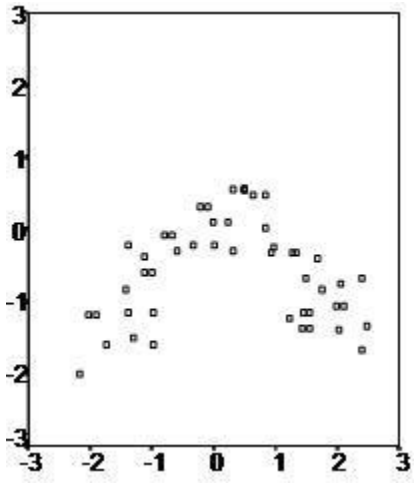
Assumptions of the linear model

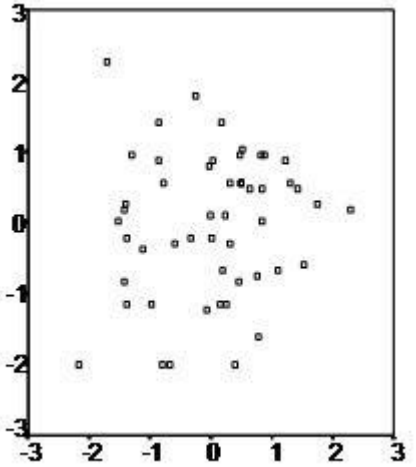
- Linear relationship
- Multivariate normality
- No multicollinearity
- No auto-correlation
- Homoscedasticity



1 – Linear relationship

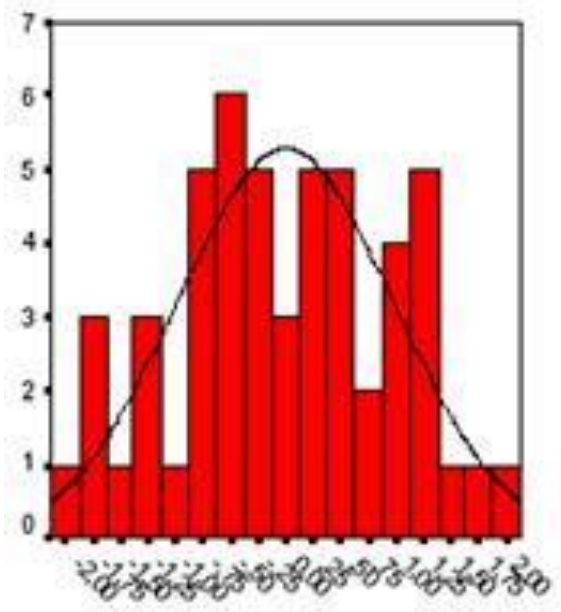
- What - Relationship between the independent and dependent variables has to be linear
- How – Check for linear pattern with scatter plots
- Fix – Using non linear models

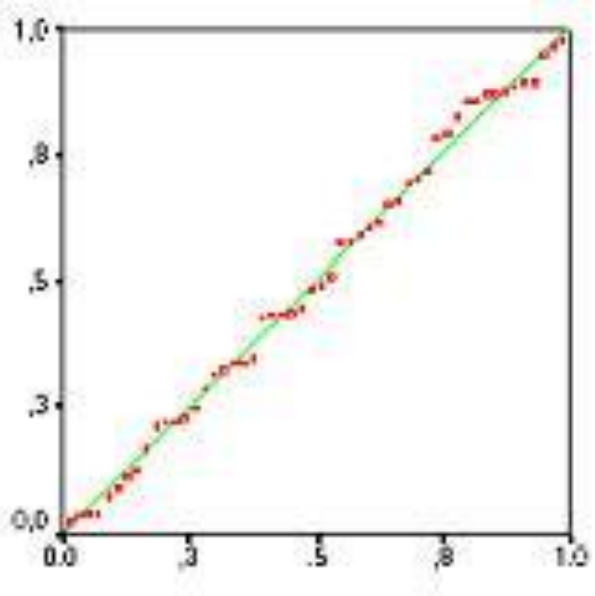




2 – All variables to be normal

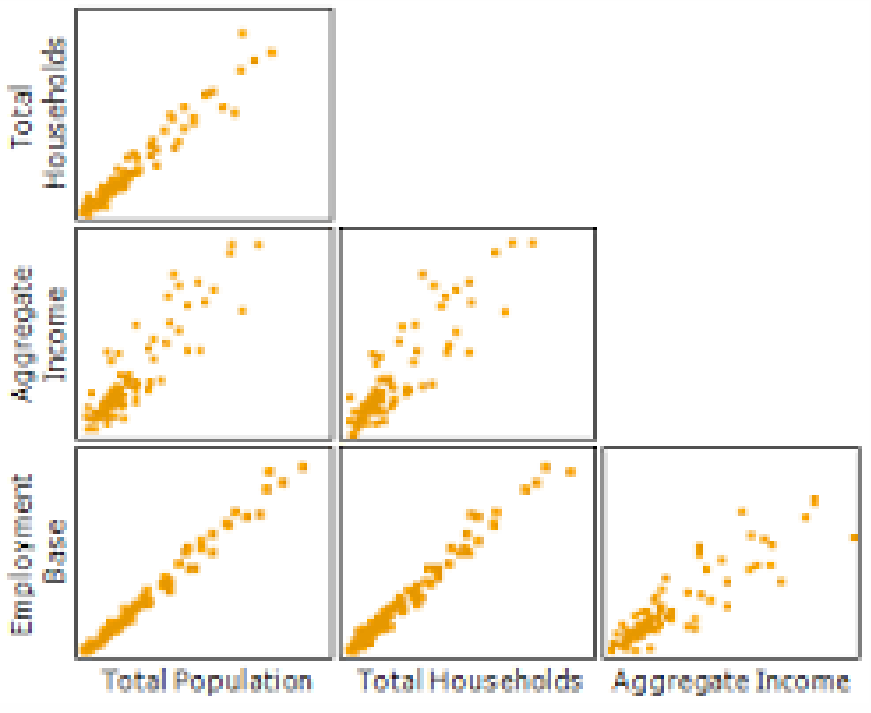
- What – All variables need to be distributed normally
- How – Check for normality using Q-Q plot or goodness of fit test (KS test)
- Fix – Using non linear transformations





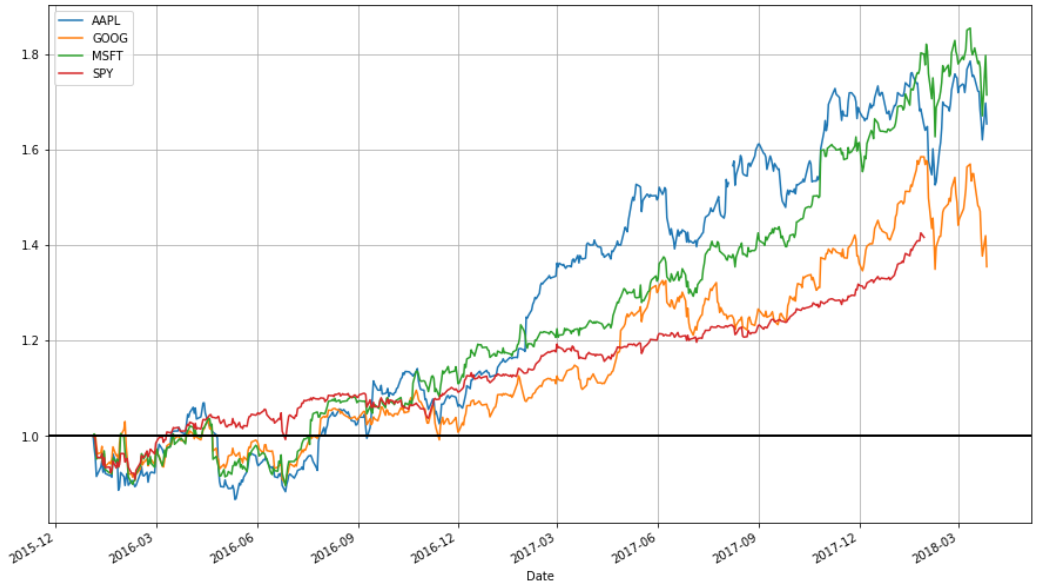
3 – No multicollinearity

- What – Linear model assumes there is little or no multicollinearity in the data
- How –
  - Correlation matrix
  - Tolerance – low values reveal multicollinearity
  - VIF – Values over 100 reveal multicollinearity
- Fix – Center the data or remove variables with high VIF values



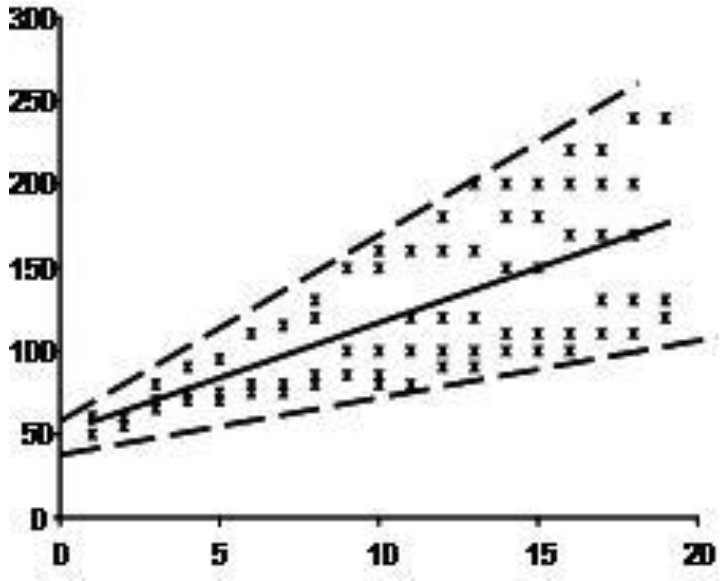
4 – No autocorrelation

- What – Linear model assumes there is little or no autocorrelation in the data
- How – Durbin-Watson test, ACF plots
- Fix – Try differencing the data or use AR models



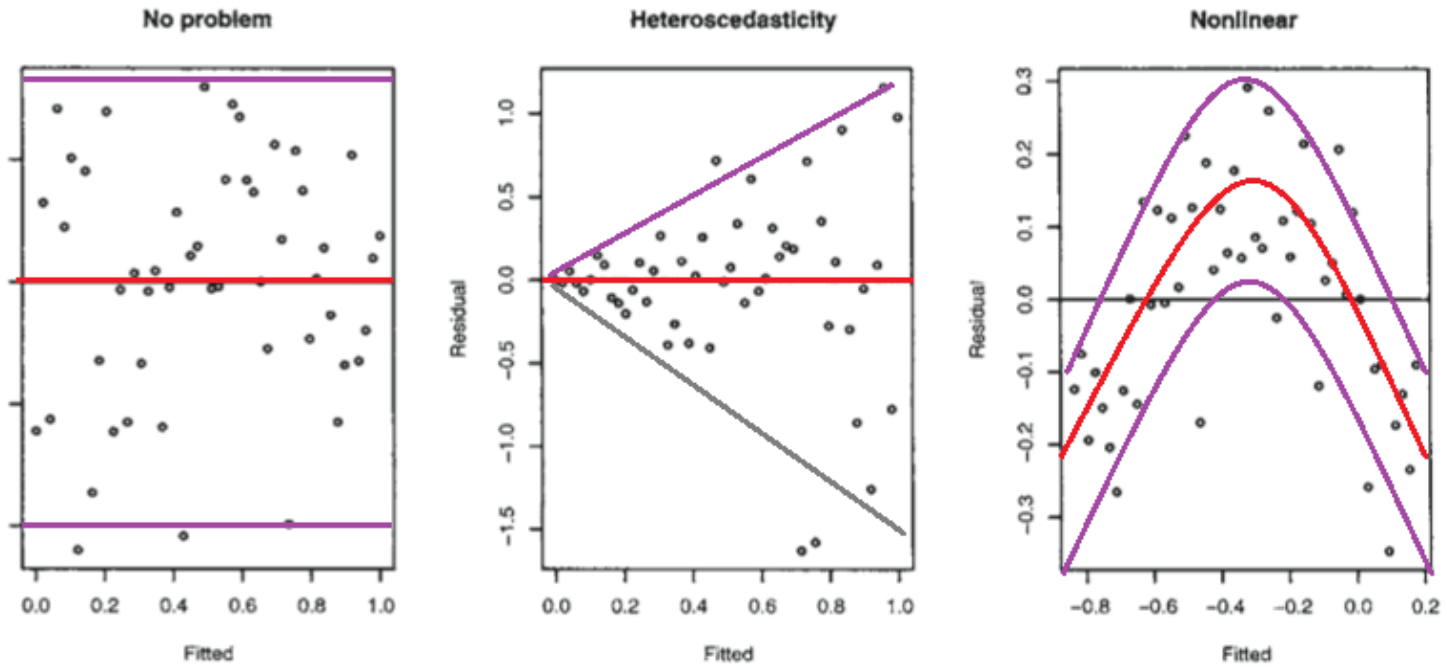
5 – No heteroscedasticity

- What – Linear model assumes there is little or no change in variance of the data
- How – Scatterplot or Goldfeld-Quandt test
- Fix – Box-cox transformation on the dependent variables



Checking the residuals

- Residuals are the actual values minus the predicted values
- Should be small and unstructured, no pattern
- Outliers
- Non-linearity
- Heteroscedasticity



Classification problems

- Classification is the process of predicting the class of given data points
- Classes are sometimes called targets, labels or categories
- What have we learned?
  - K-nearest neighbors
  - Logistic regression
  - Decision Trees
  - Random Forest
  - Boosted Trees
  - Neural Networks

Binary classification

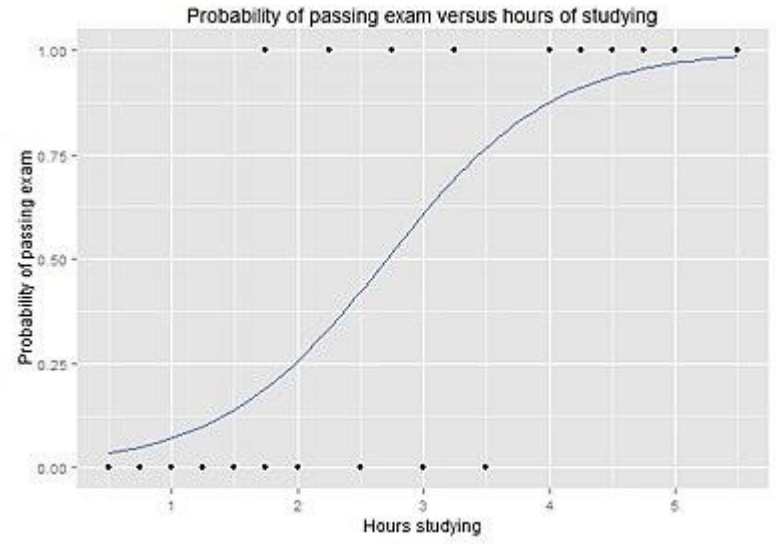
- There are 2 outcome classes
- Usually refer to as positive and negative
- Think of a classifier as sifting through a large population consisting of mostly negative, uninteresting cases, while looking for a small number of rare, positive instances
- Positive: one worthy of attention or alarm
- Negative: uninteresting or benign

- Biological sample, we are testing for disease
  - Test comes back positive – disease is present
  - Test is negative – there is no cause for alarm

Logistic Model

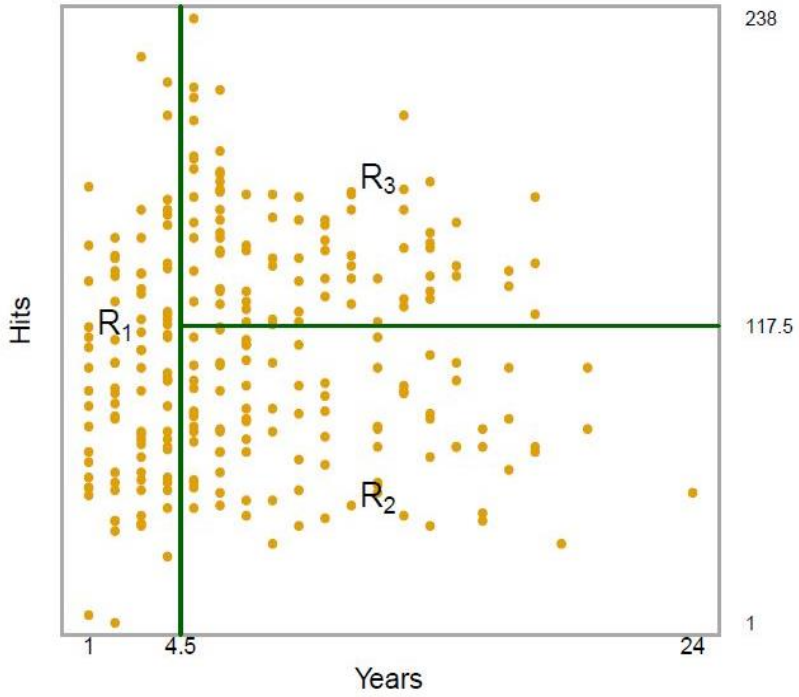
- Used when we are trying to predict a binary outcome
- Uses the Logit function to match both sides of the equation
- Predictions can be in
  - Probabilities
  - Odds
  - Log-odds
- Multinomial logistic regression – multiclass problems

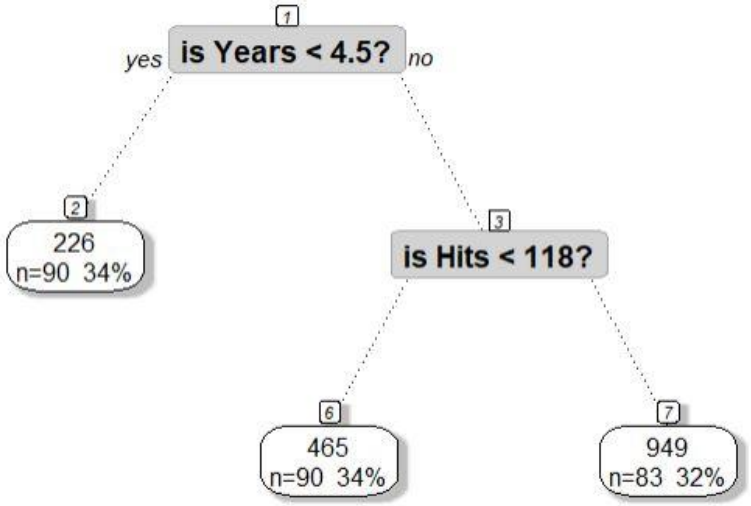
$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x + \ldots \beta_n x$$



Decision Trees

- Repeat the process looking for the best predictor and best cut point
- Minimize the Regional Sum of Squares





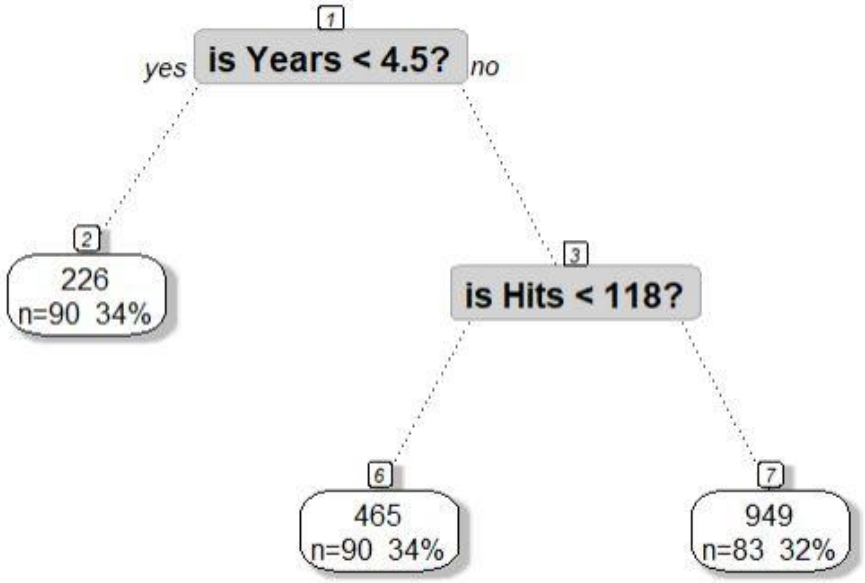
$$R_1(j,s) = \{X|X_j < s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\},$$

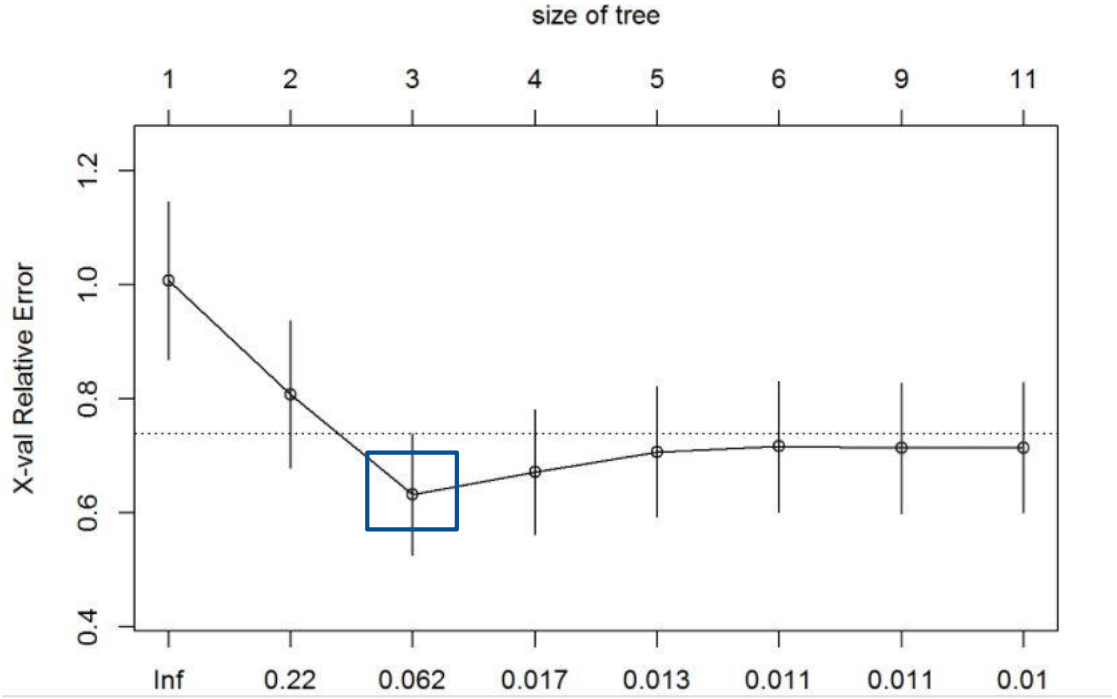
seek the value of j and s that minimize the equation

$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2,$$

Pruning – cross validation

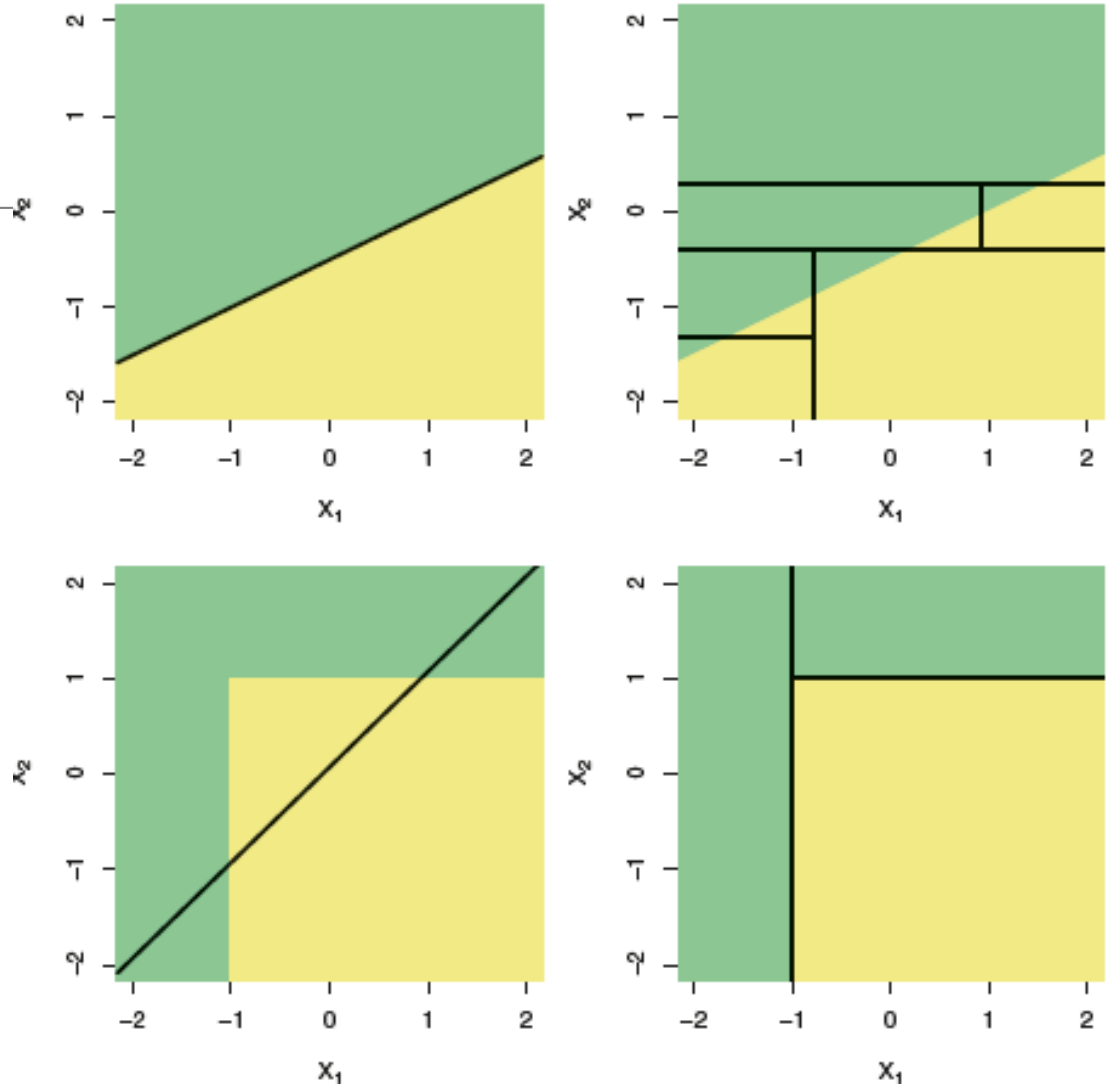
```
##           CP nsplit rel error  xerror    xstd
## 1 0.246750      0   1.00000 1.00686 0.13924
## 2 0.189906      1   0.75325 0.80766 0.12971
## 3 0.020522      2   0.56334 0.63206 0.10662
## 4 0.014281      3   0.54282 0.67086 0.10992
## 5 0.011625      4   0.52854 0.70686 0.11418
## 6 0.010870      5   0.51692 0.71573 0.11457
## 7 0.010267      8   0.48430 0.71287 0.11489
## 8 0.010000     10   0.46377 0.71403 0.11488
```





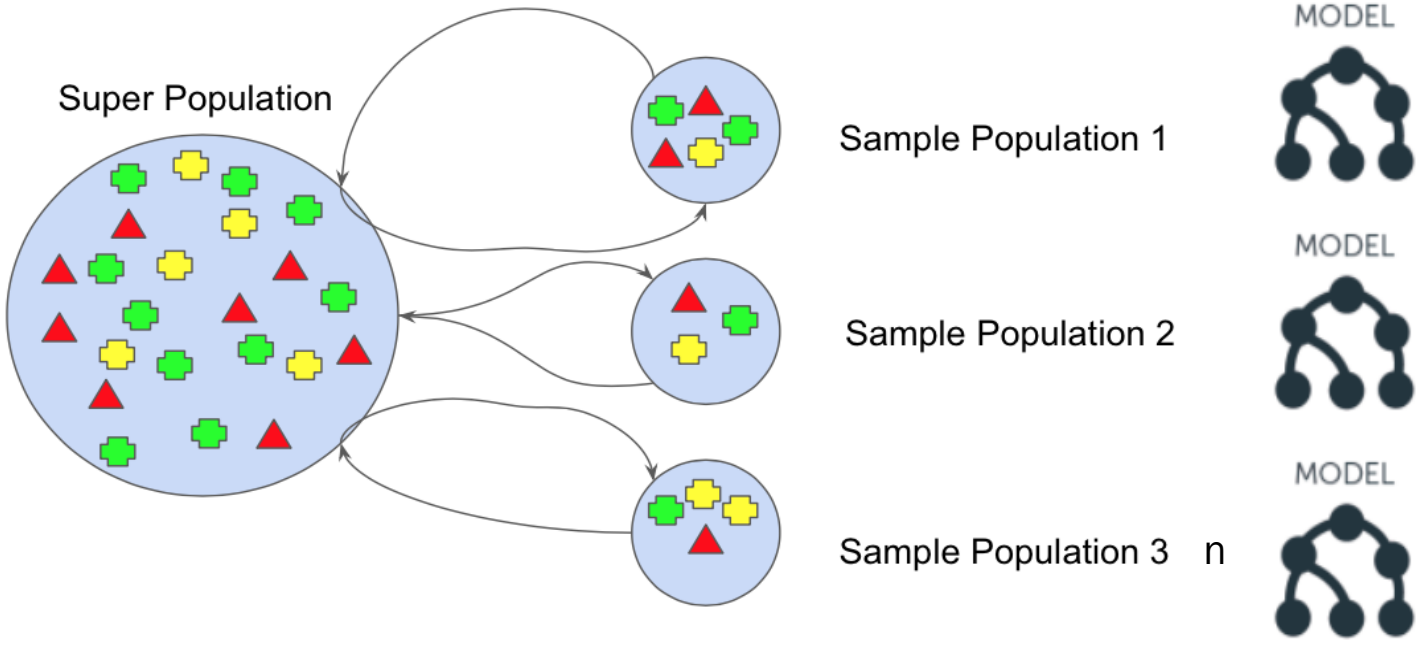
Trees vs Linear Model

- Depends on how the separation of the data is composed
- Top row: Linear classifier provides a better fit than trees for a linear space
- Bottom row: Trees provide a better fit for non linear space



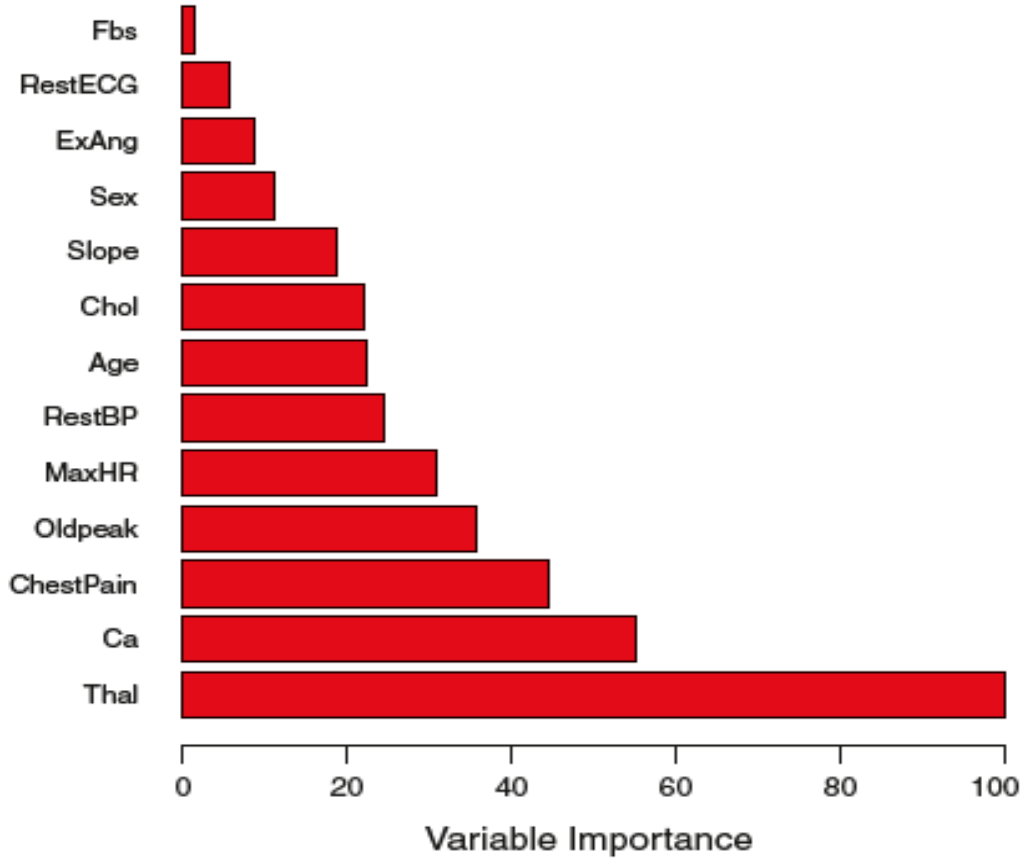
Bagging Bootstrap Aggregation

- Step 1
  - Bootstrap the data & create data set 1
  - Build decision tree 1
- Step 2
  - Bootstrap the data & create data set 2
  - Build decision tree 2
- Step n
  - Bootstrap the data & create data set n
  - Build decision tree n
- Final Step
  - Aggregate predictions
  - Regression – mean
  - Classification - mode



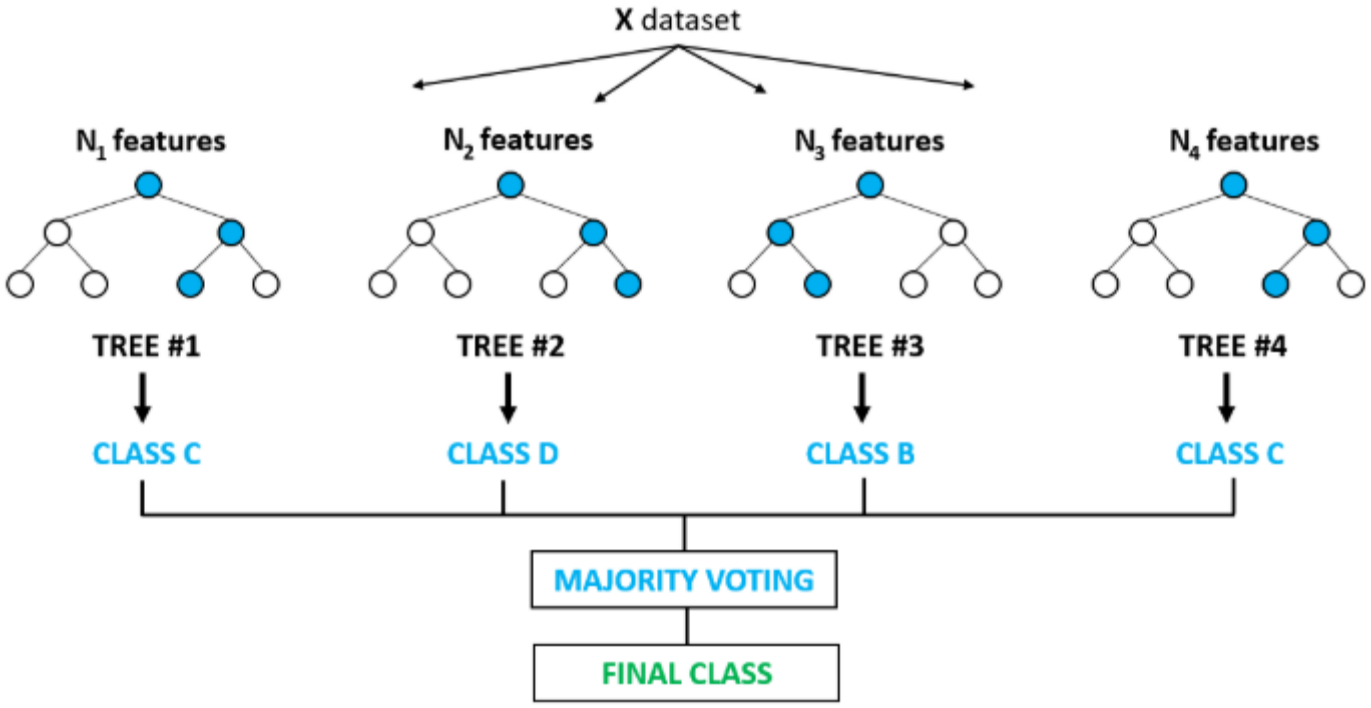
Feature Importance

- Interpretation of the features is lost because we have many trees
- Different trees and different features combine to give the aggregated prediction
- Remove one feature and measure how much error changes
- Importance is relative to the most important predictor



Random Forest

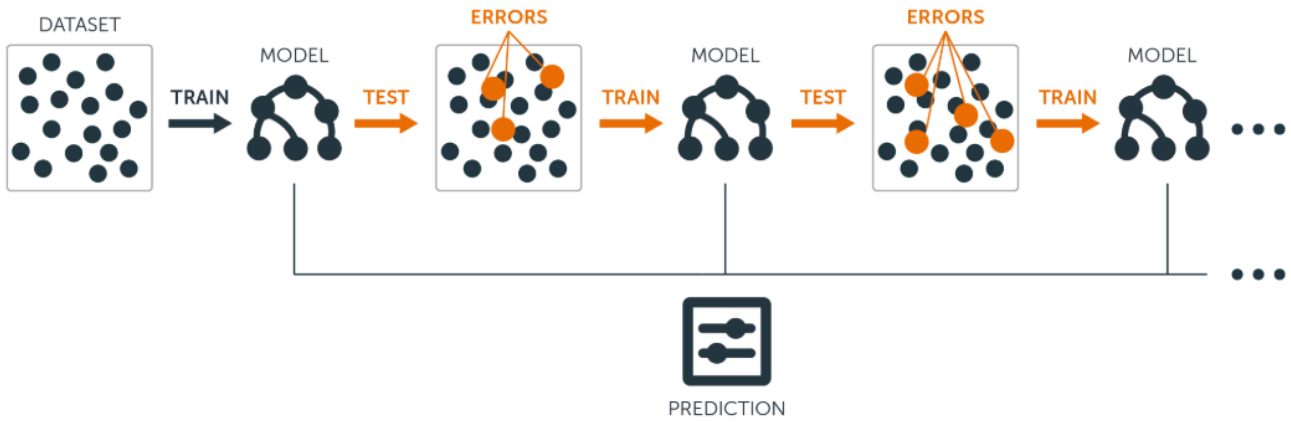
- Random Forest is very similar to Bagging
- Difference is in how we make our splits (which features we consider)
- Every time we make a split, we take a random sample of subset of N features
- Regression subset: $^{N}/_{3}$
- Classification subset: $\sqrt{N}$



Aggregate predictions

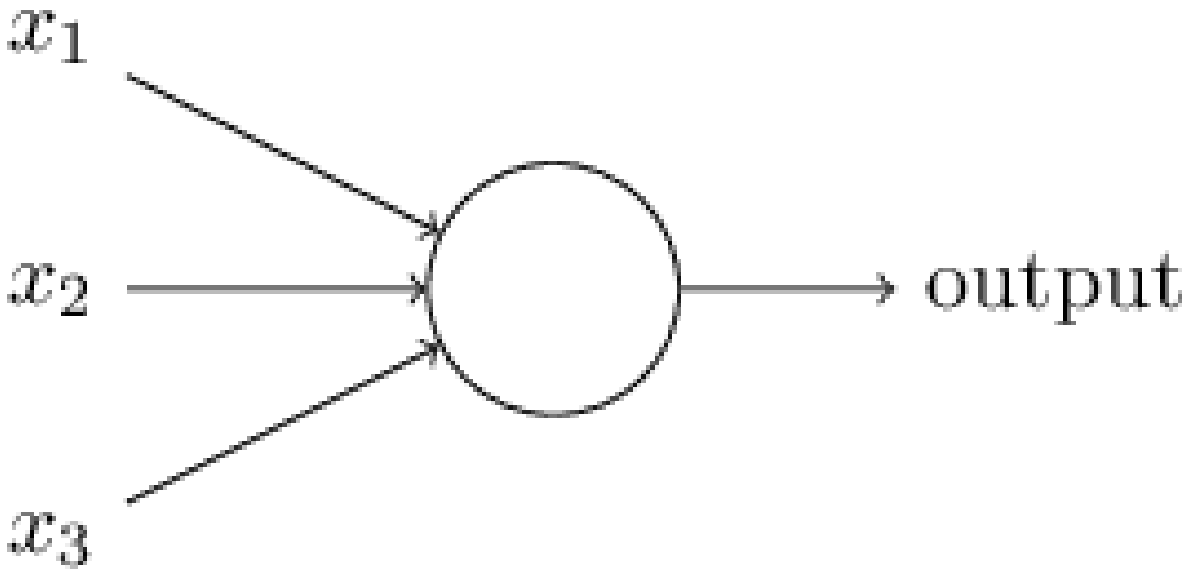
Gradient Boosted

- Difference is trees sequential and dependent
- Residual output of the first tree is the input to the next tree
- Typically use short trees (stumps)
- Slow learner progresses to become powerful
- Learning rate parameter
  - $\lambda = 0.01 \; or \; 0.001$
- Regression or Classification tasks



Single layer perceptron

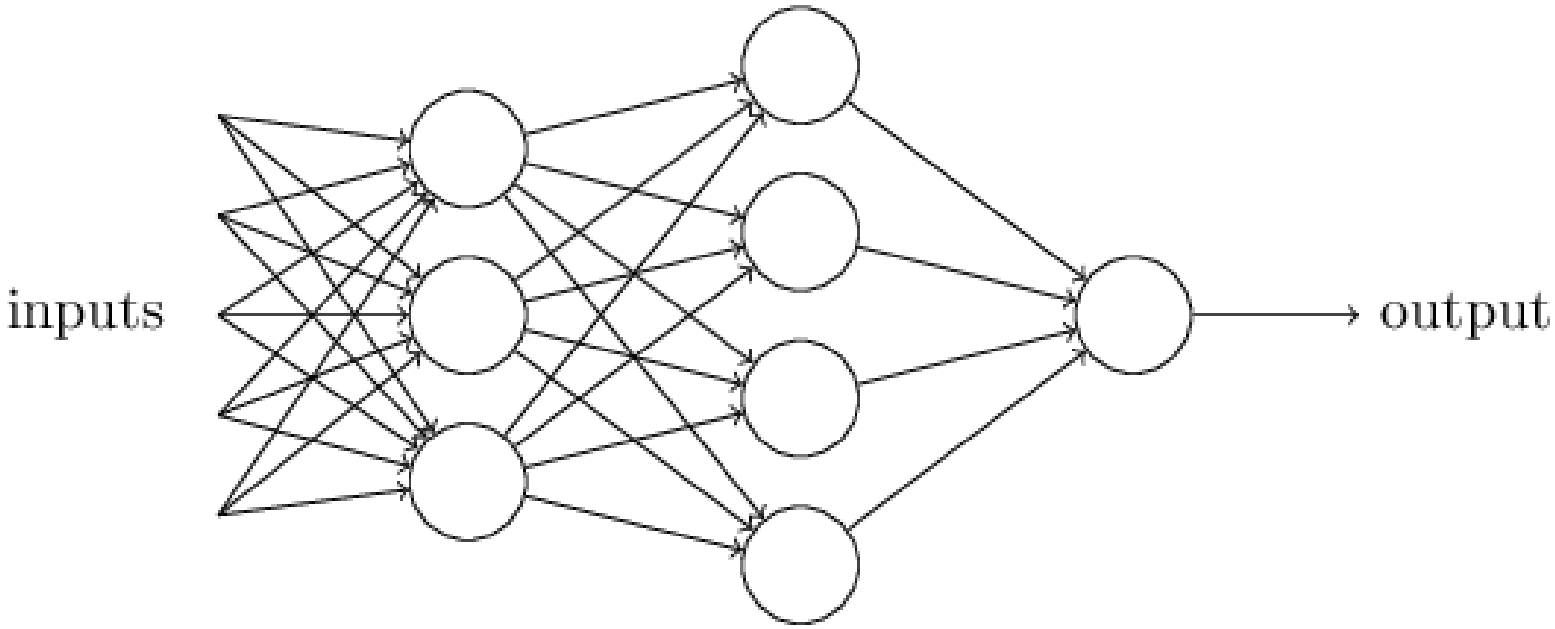
- Developed in 1950s by Frank Rosenblatt
- Takes several binary outputs and produces a single binary output (linear classifier)
- Weights express the importance of inputs to the output
- The neuron's output is 0 or 1



Structure of a single layer perceptron

Multi-layer perceptron

- First column (layer) is making three simple decisions by weighing the inputs
- Second layer is making a decision based on the outputs of the first layer

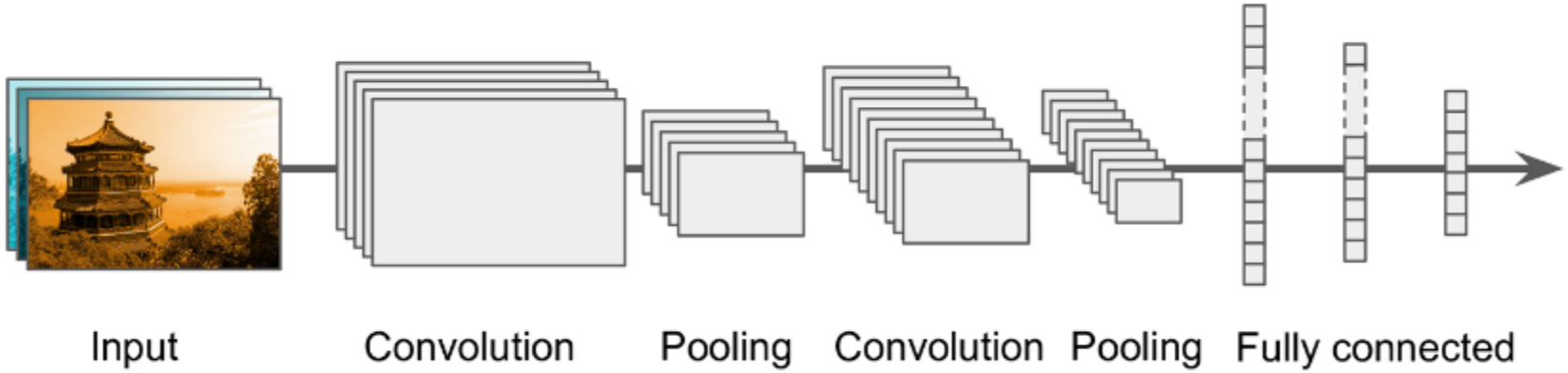


Structure of a multi-layer perceptron

Backpropagation

- Means the backward propagation of errors, this is a method of training multilayer feedforward artificial neural networks
- The method is based on calculation of the gradient of a loss function, with respect to coefficients of the network
- The gradient is returned to the learning process and used by it to find a new iteration of the coefficients that improve the fit
- Backpropagation is a supervised learning method: it requires knowing a output for training sample

CNN architecture



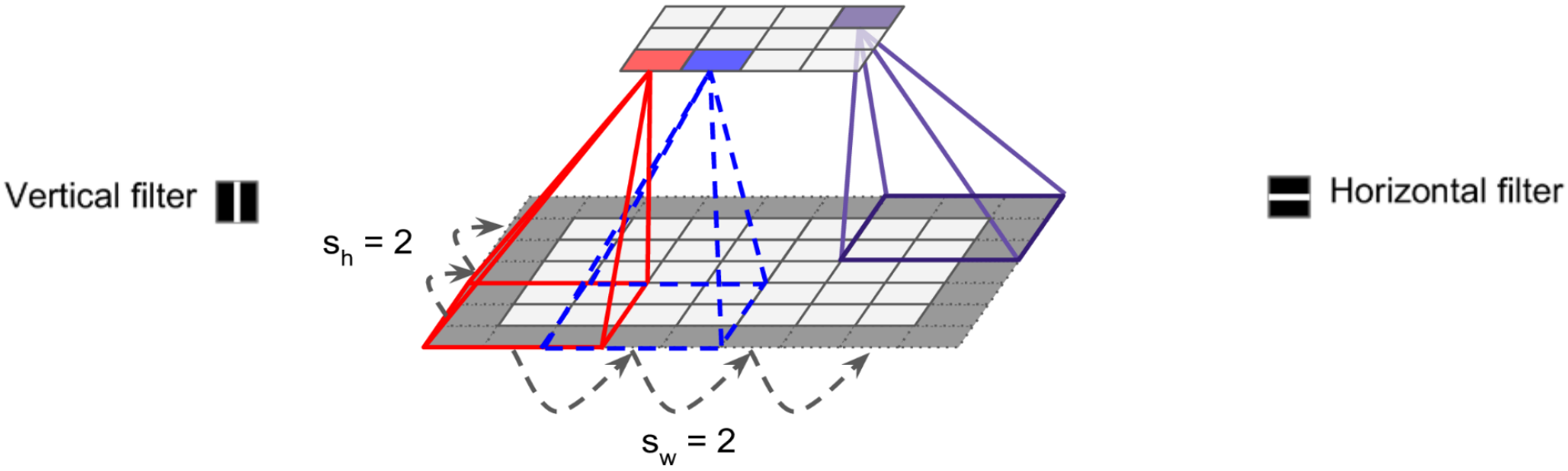
Input: The image is composed of pixels and possibly color channels (RGB). The image gets smaller and smaller as it progresses through the network, also deeper and deeper

Convolution: Creating feature maps using a hierarchical scanning process

Pooling: Downsizing feature maps by one or several pooling kernels, like max or mean

Fully Connected: Regular feedforward neural network, final layer (softmax) estimates class probabilities

Convolutional layer



Stride: Distance between two consecutive receptive fields (horizontal and vertical)

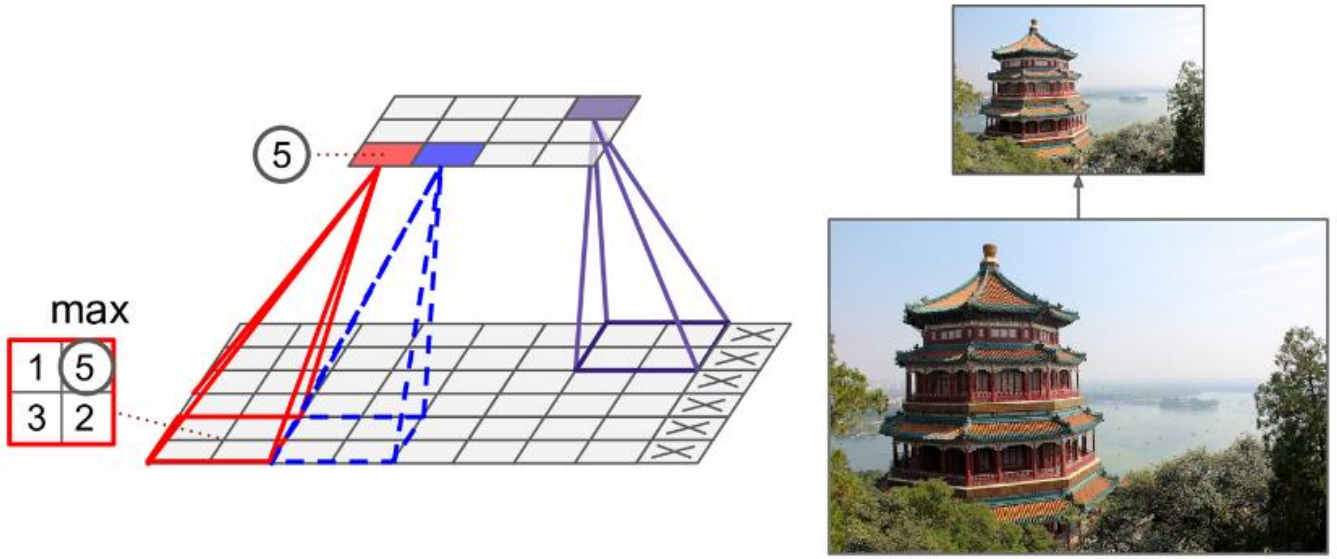
Zero Padding: Adding zeros around the inputs

Neuron: Only connected to pixels in their receptive fields

Filter (convolutional kernel): Neuron's weights, same across the entire filter. Typically learned during training, can be combined into more complex patterns.

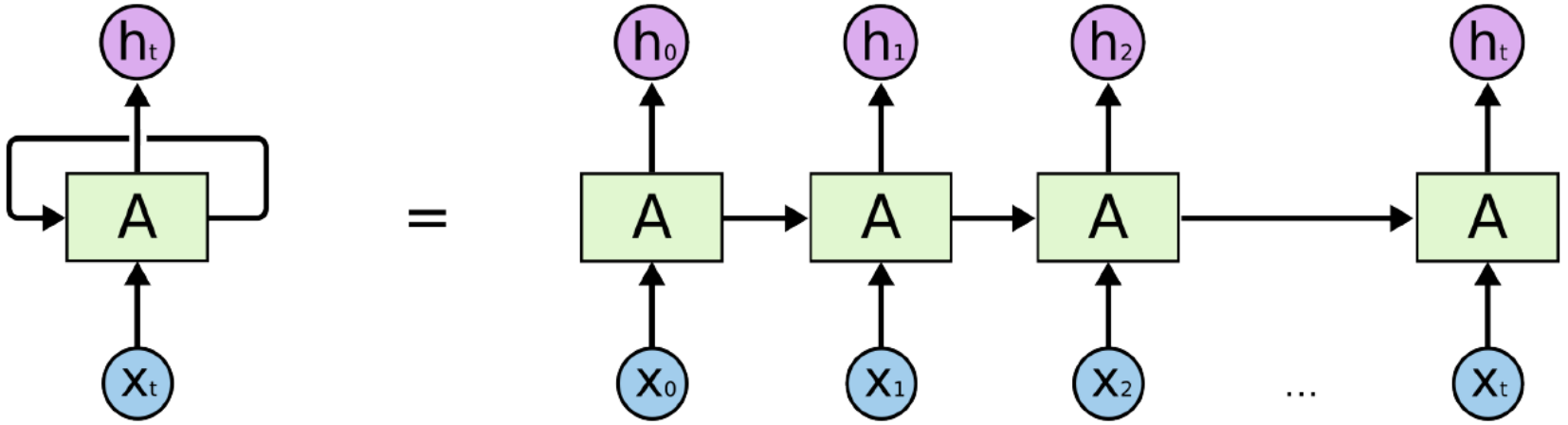
Feature Maps: Resulting data from use of the filter

Pooling layer



- Subsample the input image in order to reduce the computational load, memory usage and number of parameters (reduce risk of overfitting)
- Pooling also requires size and stride parameters.
- Works to aggregate data. Common methods are max or mean pooling

Unrolling the neuron through time



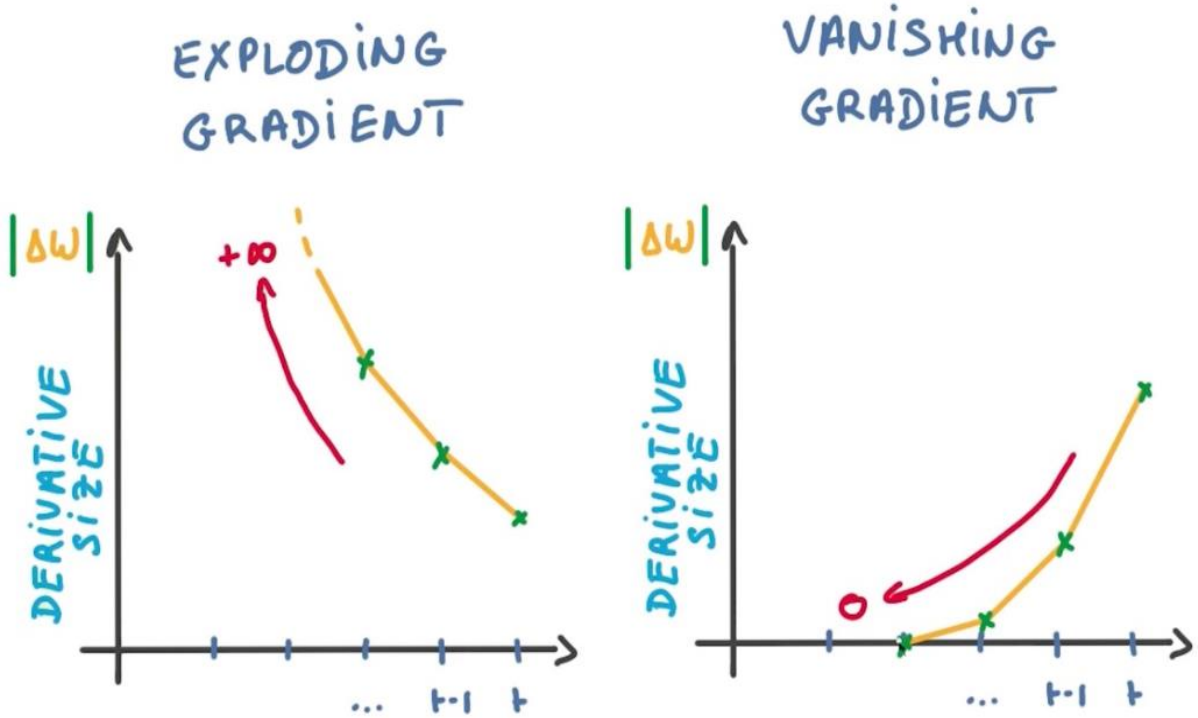
- Recurrent neuron is like multiple copies of the same neuron, which pass information to each other
- The neuron receives two inputs at each time t

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Problems with long-term dependencies

Exploding/vanishing gradients

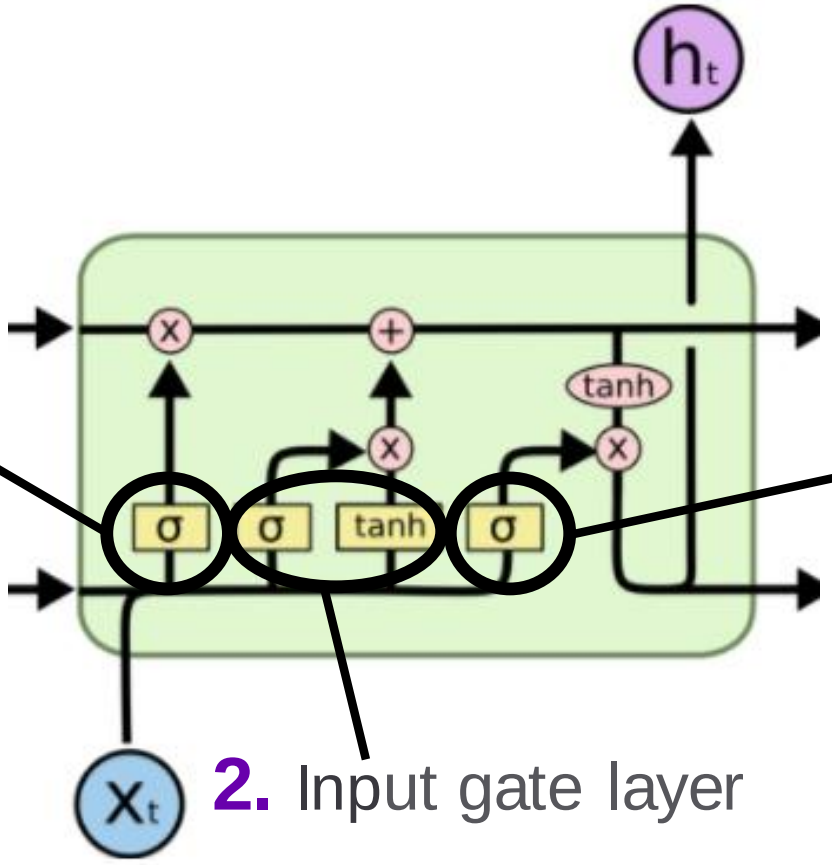
- 1991 Sepp Hochreiter
  - *Fundamental Deep Learning Problem*
- 1997 Sepp Hochreiter
  - *Long Short-Term Memory*



http://people.idsia.ch/~juergen/fundamentaldeeplearningproblem.html
https://www.bioinf.jku.at/publications/older/2604.pdf

Gates regulate information

An LSTM has three gates to affect the cell state



1. Forget gate layer

2. Input gate layer

3. Output gate layer

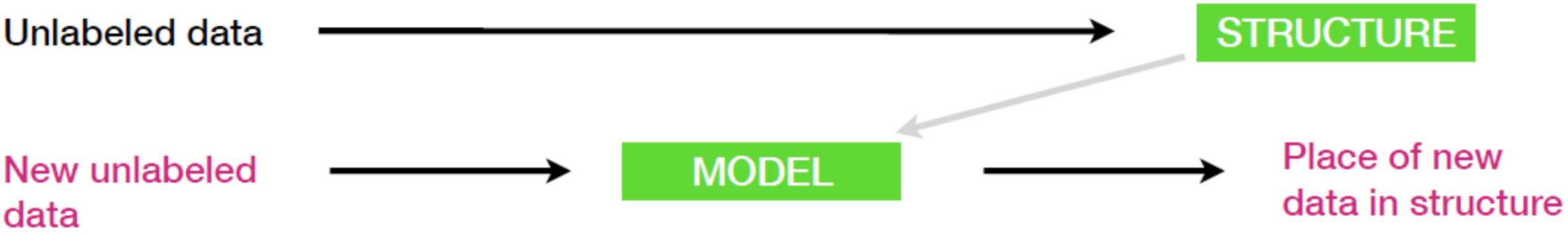
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Modeling

PART 2 – UNSUPERVISED LEARNING

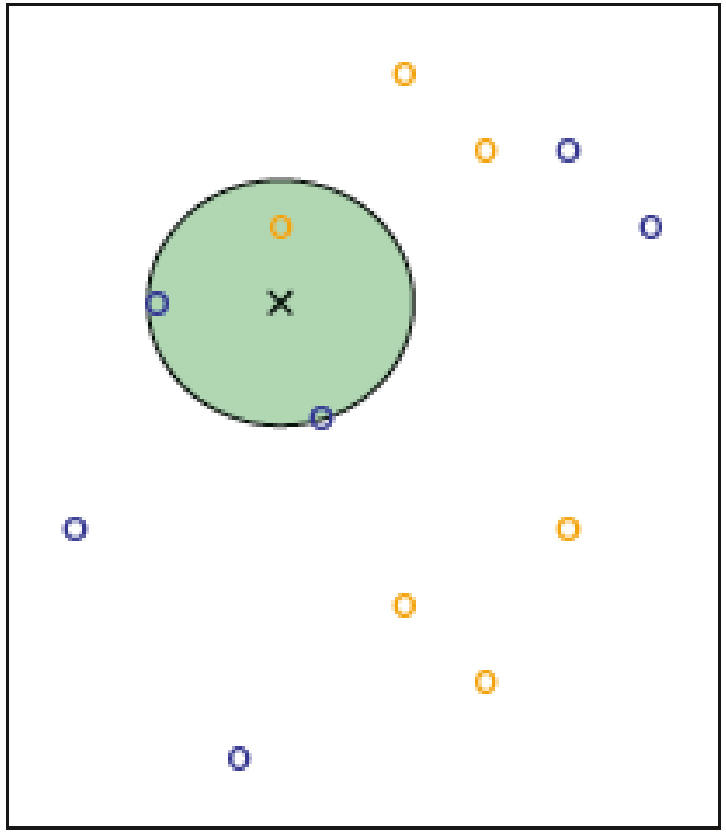
Unsupervised learning problems

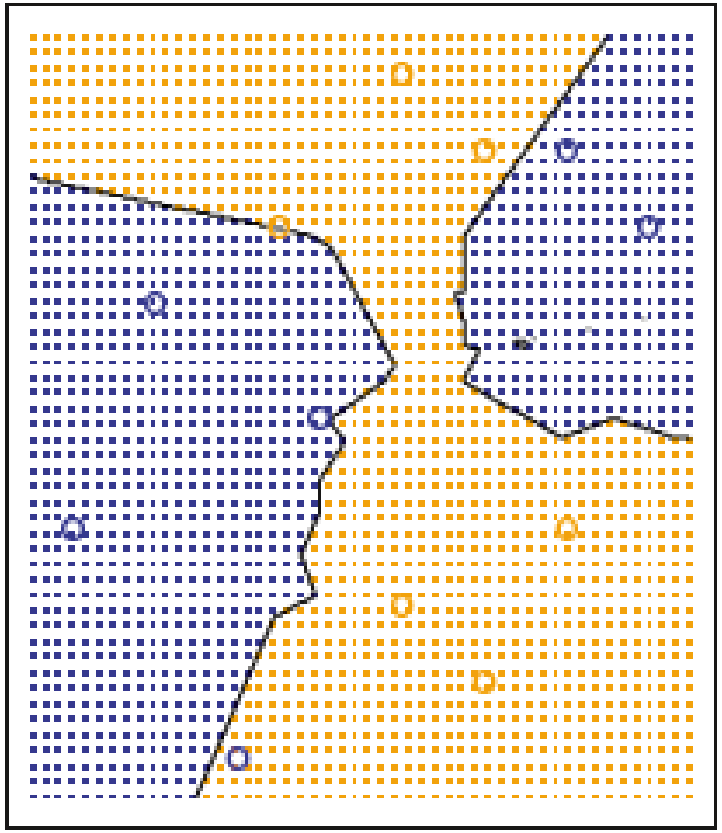
- Involve constructing models where labels on historical data are unavailable



KNN approach where K = 3

- Classification algorithm
- Identify the K (3) **closest points**
- Calculate the probability of classes (blue or orange)
- Apply rule to classify the test observation (black x)



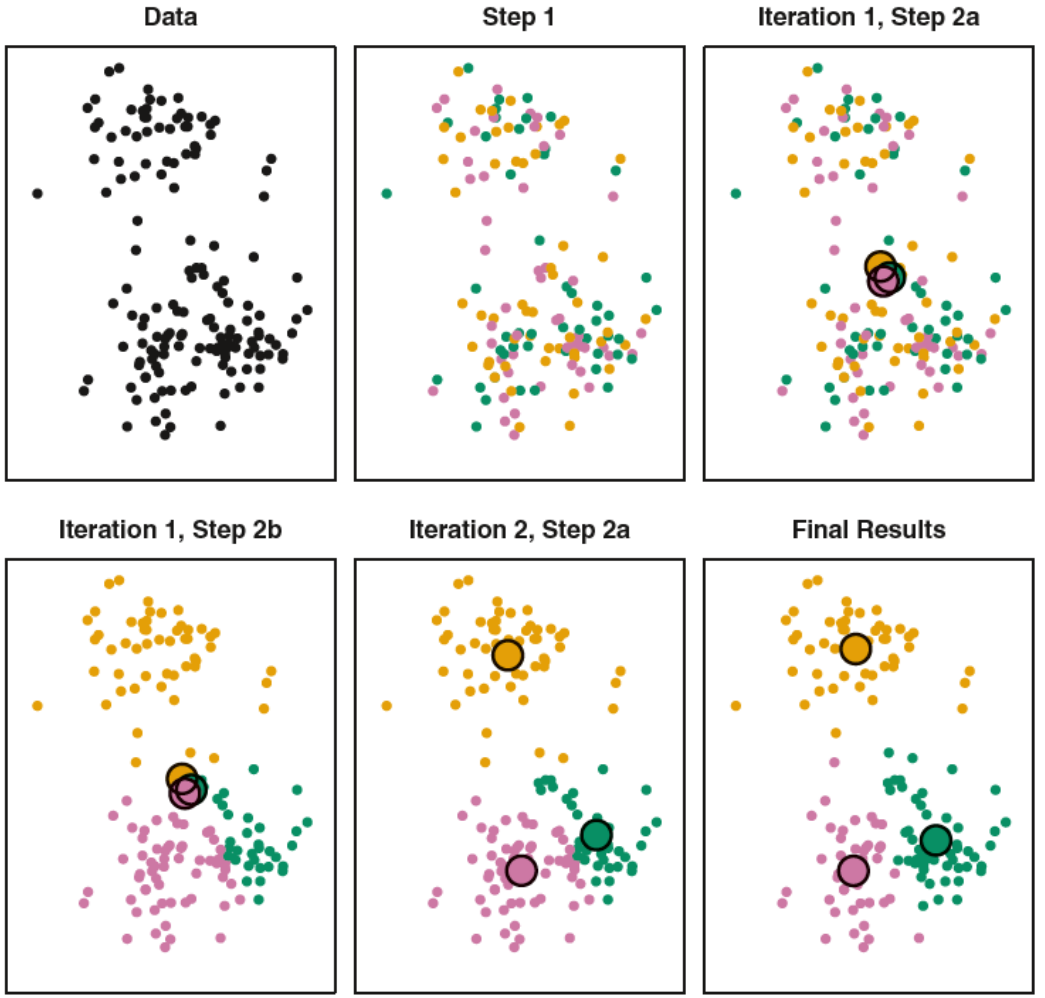


Training data set: 6 blue, 6 orange
Goal: Predict color of black x

Results: Predictions for all possible values of our feature space (X1, X2)
Decision Boundary: Black line

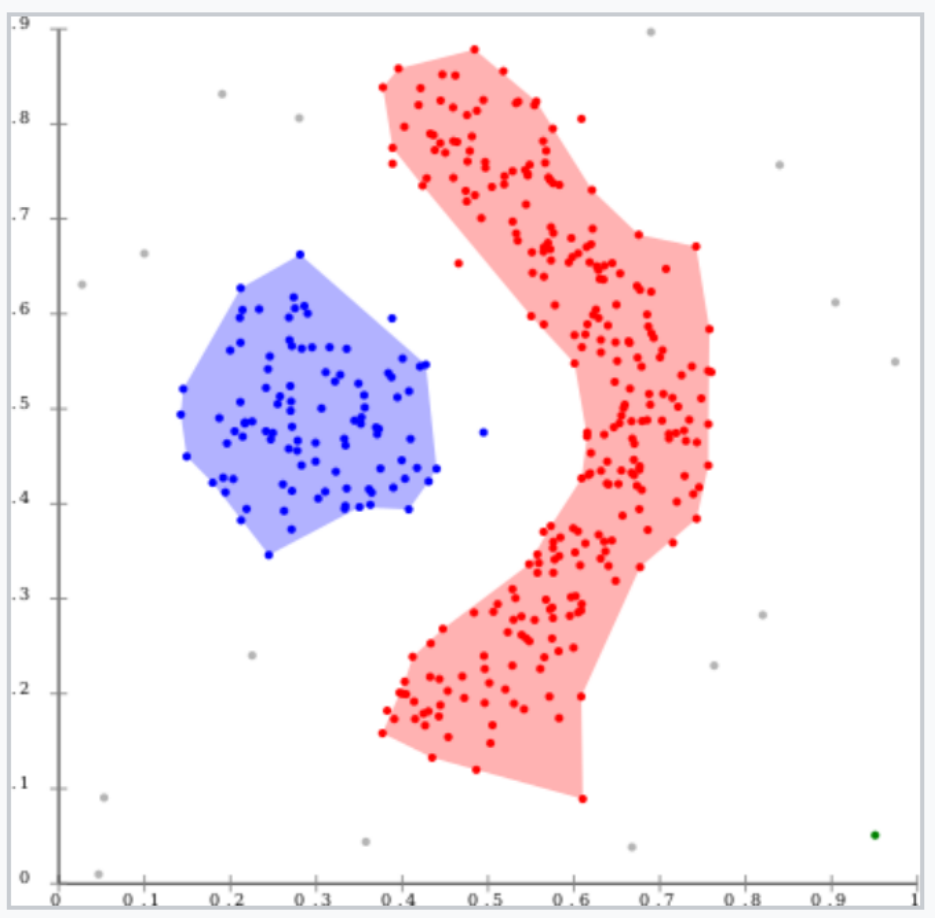
K-means algorithm

1. Randomly assign a number to each observation, initial cluster assignments
2. Iterate until cluster assignments stop changing
   a. For each of the K clusters, compute the centroid.
   b. Assign each observation to the cluster whose centroid is closest using Euclidean distance



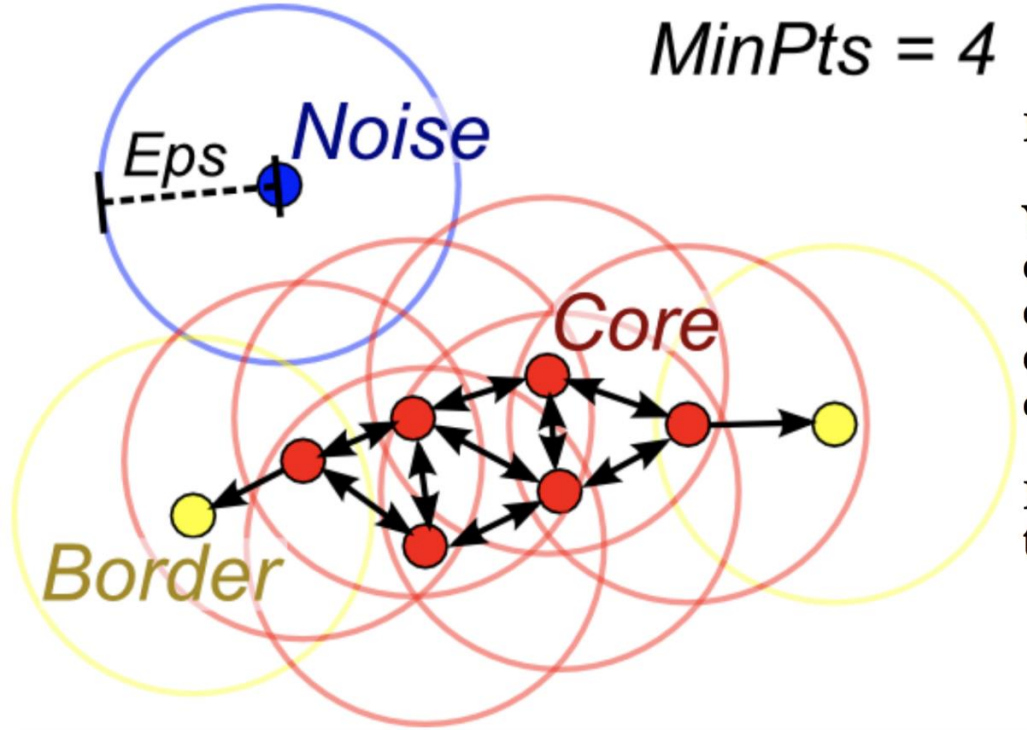
DBSCAN

- Density-Based Spatial Clustering of Applications with Noise
- K-means does not care about the density of data, but DBSCAN does.
- Assumes that regions of high density in your data should be treated as clusters



DBSCAN

- Parameters are
  - Epsilon
  - MinPts



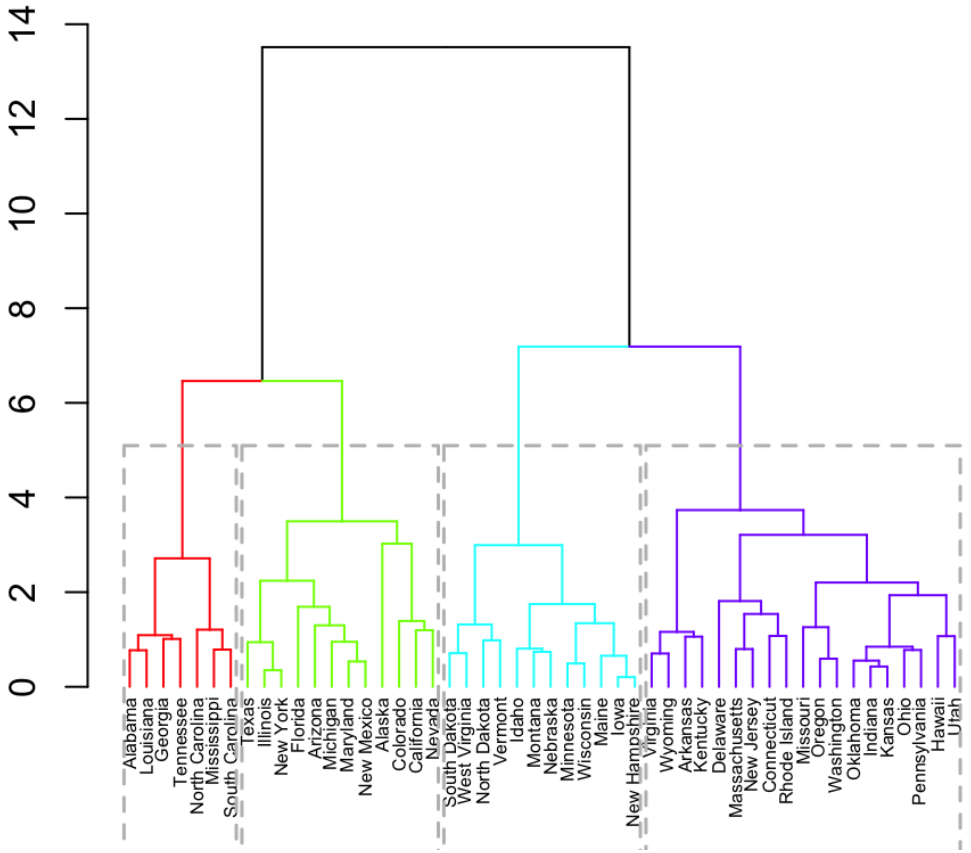
Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

Hierarchical clustering

- Greedy method that pairs the two closest observations
- Results in a structural plot called a dendrogram
- Height of branches represents relative distance of the two sub branches
- Distance – What type of ruler?
  - Euclidean, Manhattan, maximum
- Linkage – Where do we measure from?
  - Single, complete, average, ward

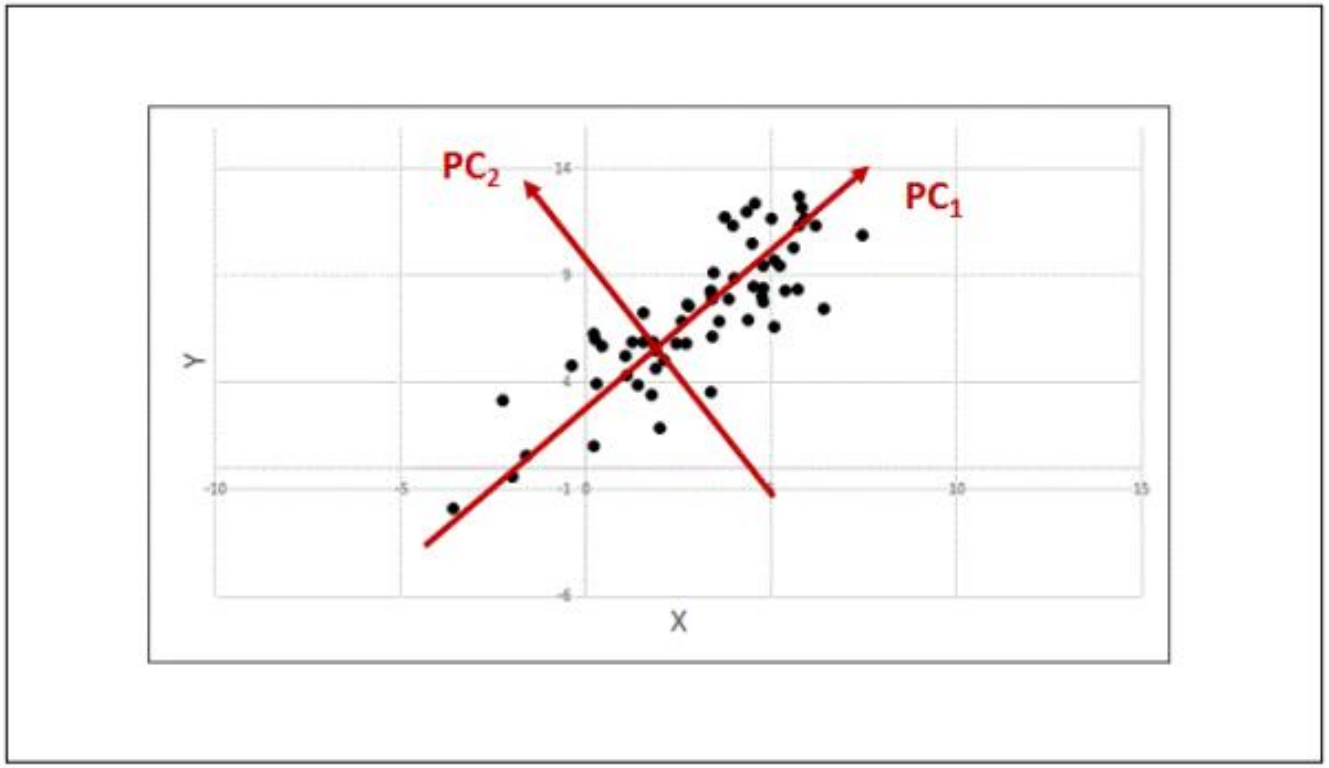


Modeling

PART 3 – DIMENSIONALITY REDUCTION

The real dimension of the data

- PCA finds a new coordinate system obtained from the old one by translation and rotation only and moves center of coordinate system to center of data.
- It moves the x axis onto the principal axis of variation, where you see the most variation relative to other data points
- It moves further axes down the road into orthogonal, less important directions of variation
- PCA finds for you these axes and also tells you how important these axes are. This allows you to use only the most important principal components when building your model.



Outline of the algorithm

1. Create centered matrix

$$\begin{aligned}\mathbf{Y}_0 &= [\mathbf{y}^0_{\cdot 1}, \mathbf{y}^0_{\cdot 2}, \ldots, \mathbf{y}^0_{\cdot n}] \\ &= [\mathbf{y}_{\cdot 1} - \mathbf{1}\bar{y}_{\cdot 1}, \mathbf{y}_{\cdot 2} - \mathbf{1}\bar{y}_{\cdot 2}, \ldots, \mathbf{y}_{\cdot n} - \mathbf{1}\bar{y}_{\cdot n}]\end{aligned}$$

2. Calculate covariance matrix

$$\Theta = cov(\mathbf{Y}_0) = \left\{\mathbb{E}\left[\mathbf{y}^0_{\cdot i}\mathbf{y}^0_{\cdot j}\right]\right\}$$

3. Perform eigenvalue decomposition of Θ. Define

$$\mathbf{L} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m],$$

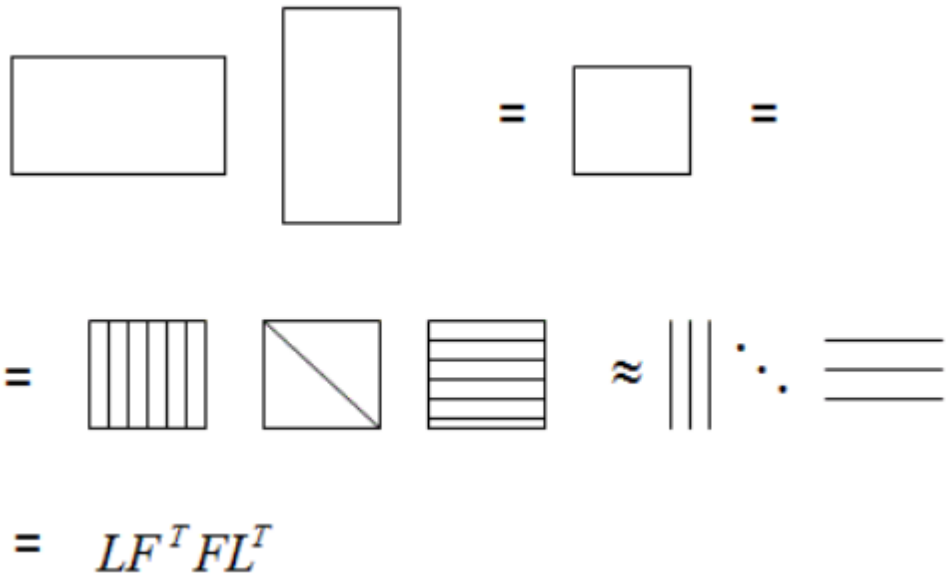
where $\mathbf{u}_i$ is the eigenvector corresponding to the i-s largest eigenvalue of Θ.

4. Define

$$\mathbf{F} = \mathbf{Y}_0\mathbf{L},$$

right-multiplying by $\mathbf{L}$ the definition of the model

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T.$$



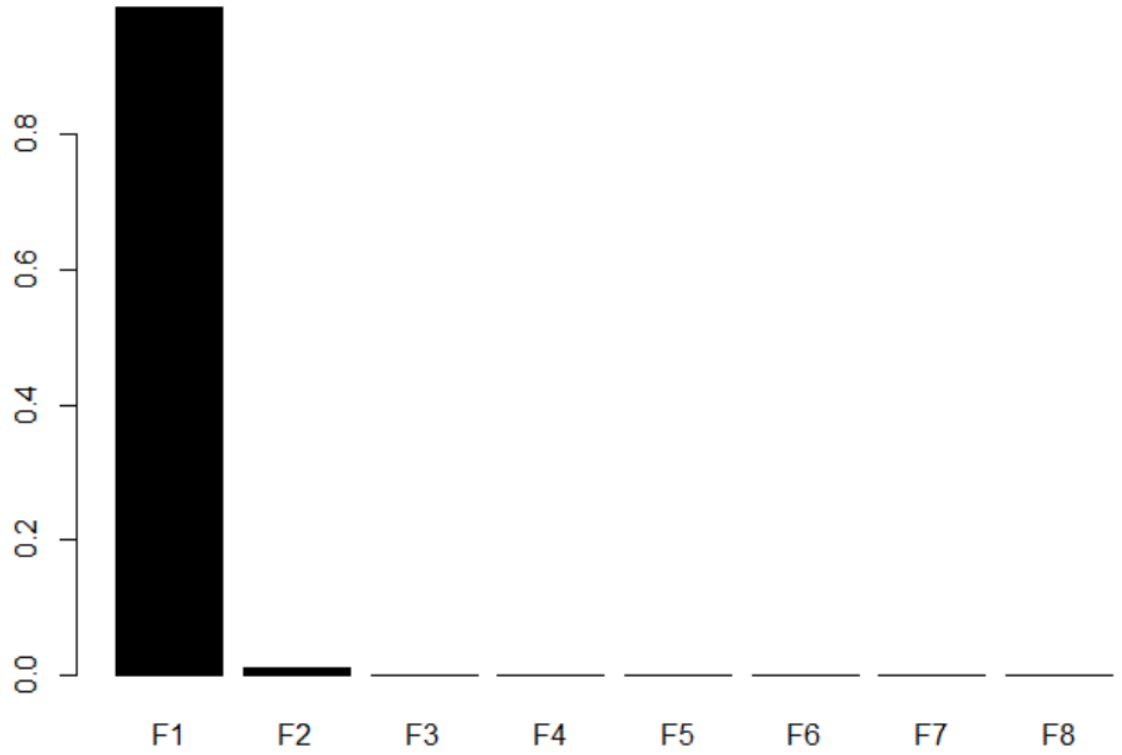
Factors and Loadings

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T$$

- **F** plays the role of inputs. The columns are usually called principal components, factors or factor scores
- L plays the role of slopes. The columns (rows in L^T) are called factor loadings

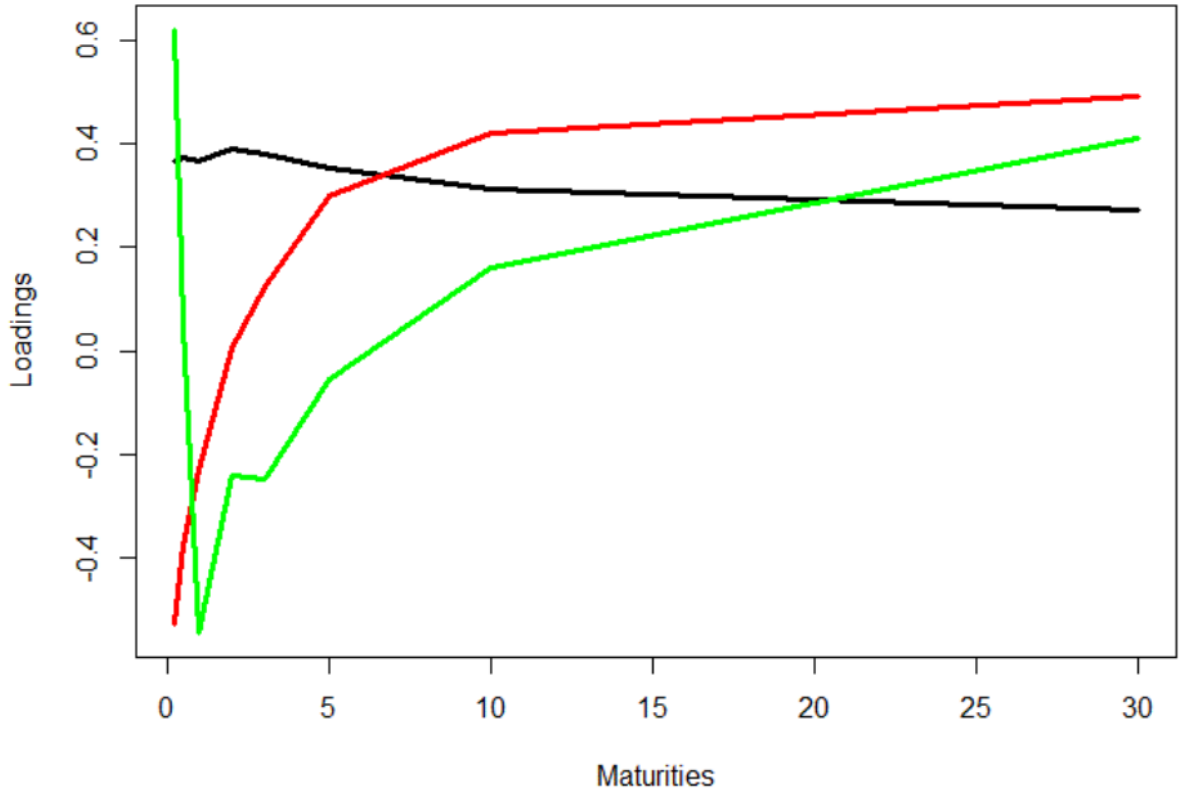
Factors of treasury data

- We only need the first three factors (principal components) to represent 99% of the relative variance in the original data



Loadings of treasury data

- If we plot the loadings, the shapes can let us interpret the data
- Black – Level
  - Parallel shifts of the curve
- Red – Slope
  - Tilting of the yield curve
- Green – Curvature
  - Flexing of the yield curve

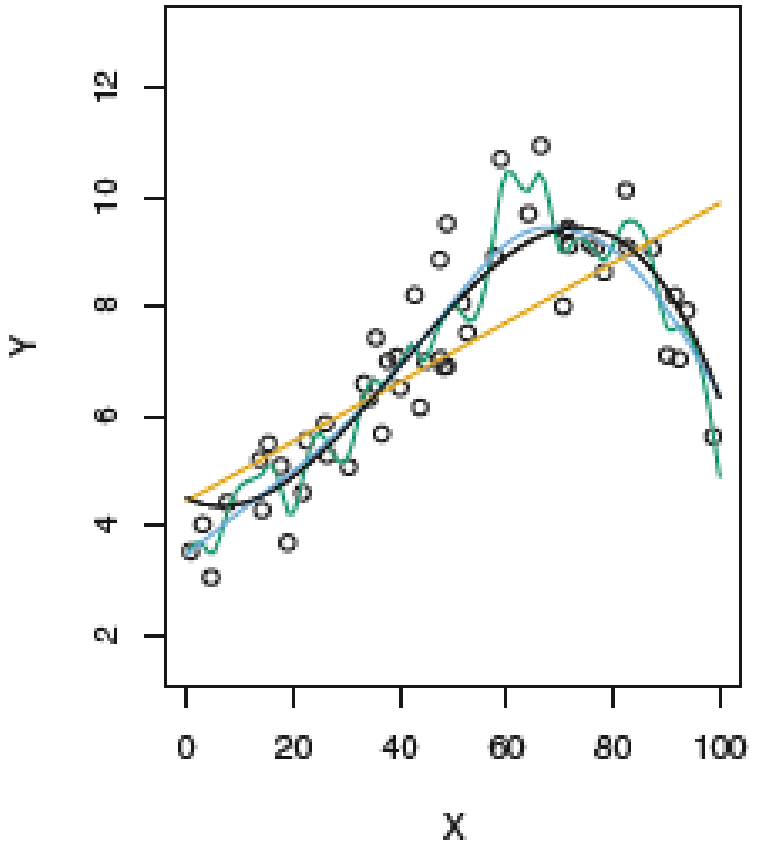


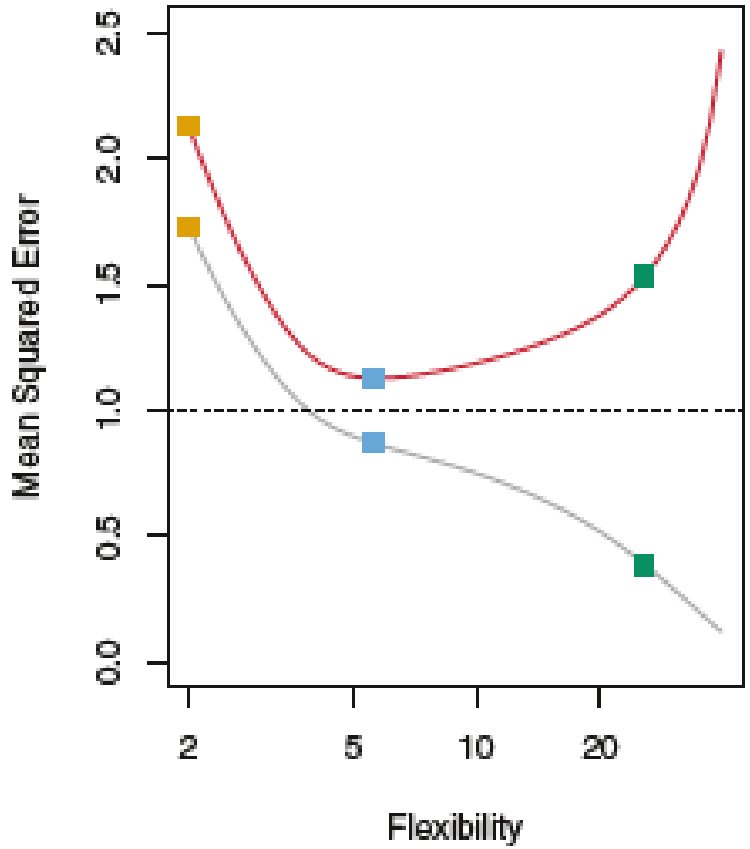
Modeling

PART 4 REGULARIZATION

Examining model fits

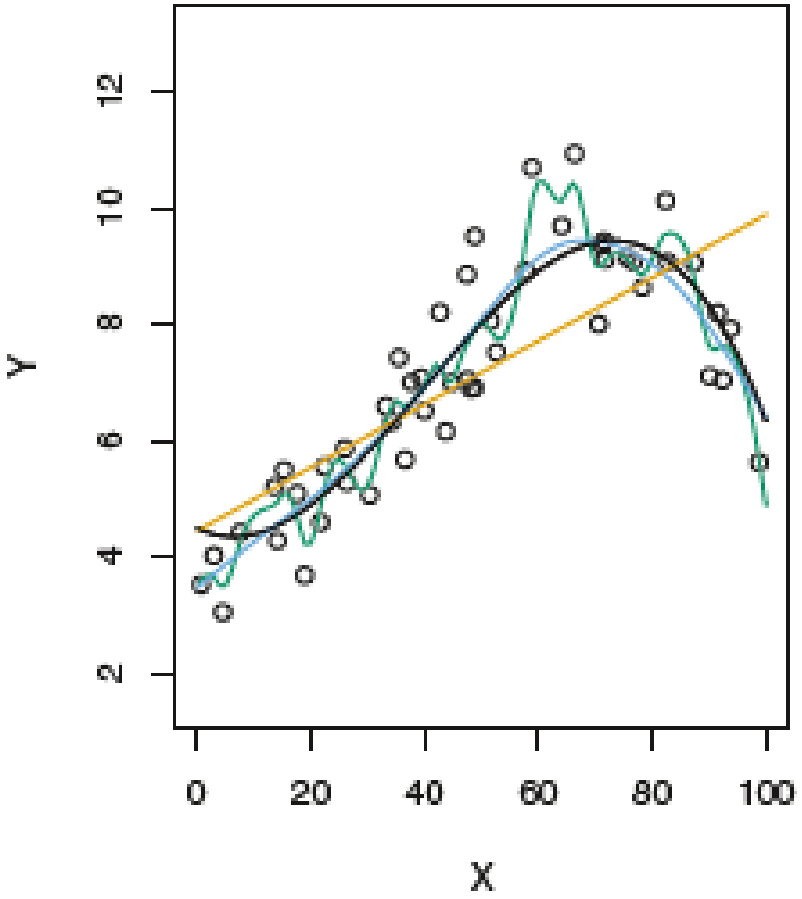
- Left: Three estimates of observed data
- Right:
  - MSE of training set (gray)
  - MSE of test set (red)
  - Colored squares represent MSE of fitted models on left





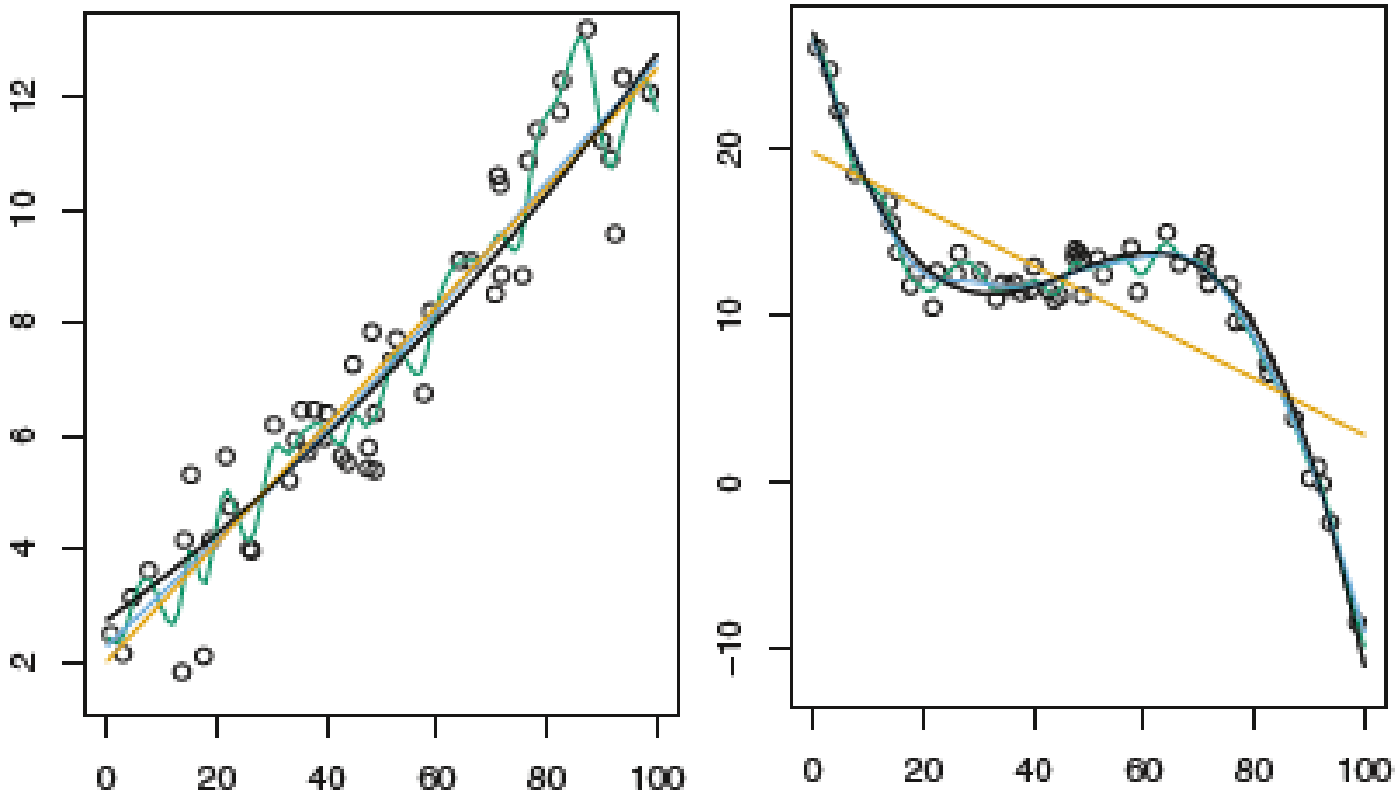
Model variance

- Variance is the amount by which $\hat{f}$ would change if we estimated it using a different train set
  - Since different training data sets are used to fit our models, different sets result in different $\hat{f}$
  - If a method has a high variance, then small changes in data will cause large changes in $\hat{f}$
  - More flexible methods have higher variance
    - Green: High variance
    - Orange: Low variance



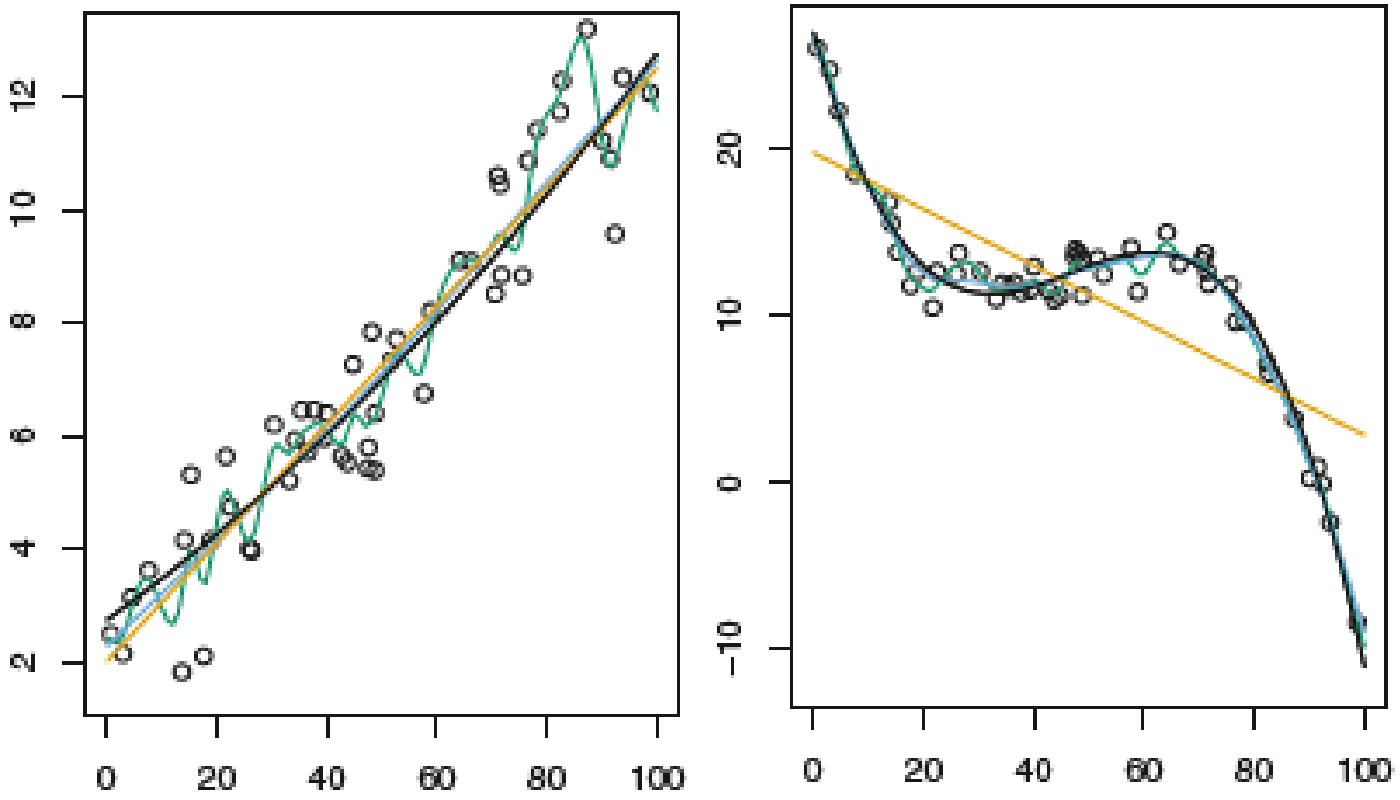
Model bias

- Bias is the error introduced by solving a real-life problem with a simple model
  - No problem is truly linear
  - Linear **data**: Linear model has LOW bias
  - Non-linear **data**: Linear model has HIGH bias

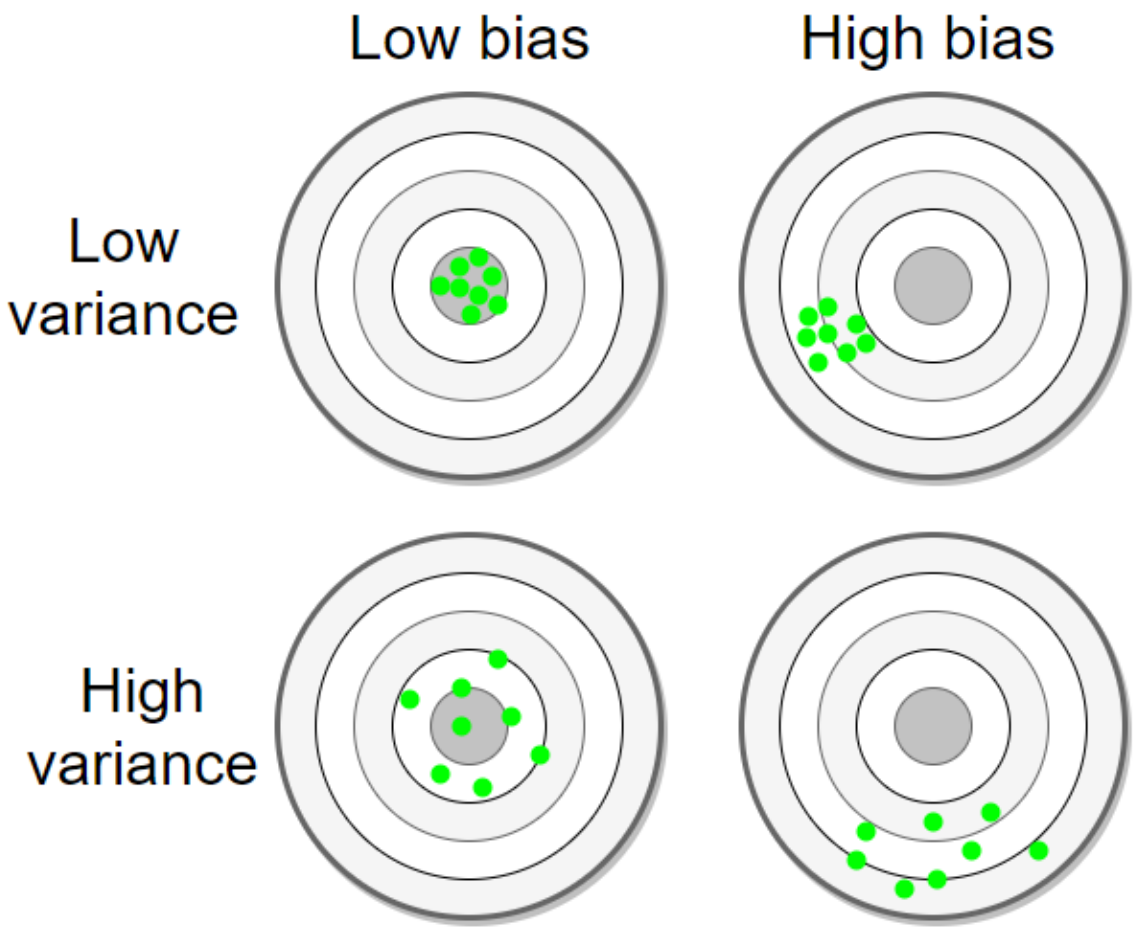


Model bias

- Bias is the error introduced by solving a real-life problem with a simple model
  - No problem is truly linear
  - Linear data: Linear model has LOW bias
  - Non-linear data: Linear model has HIGH bias



Another bias/variance visualization



Ridge regression

- In the process of fitting linear regression model we minimize the sum of squares

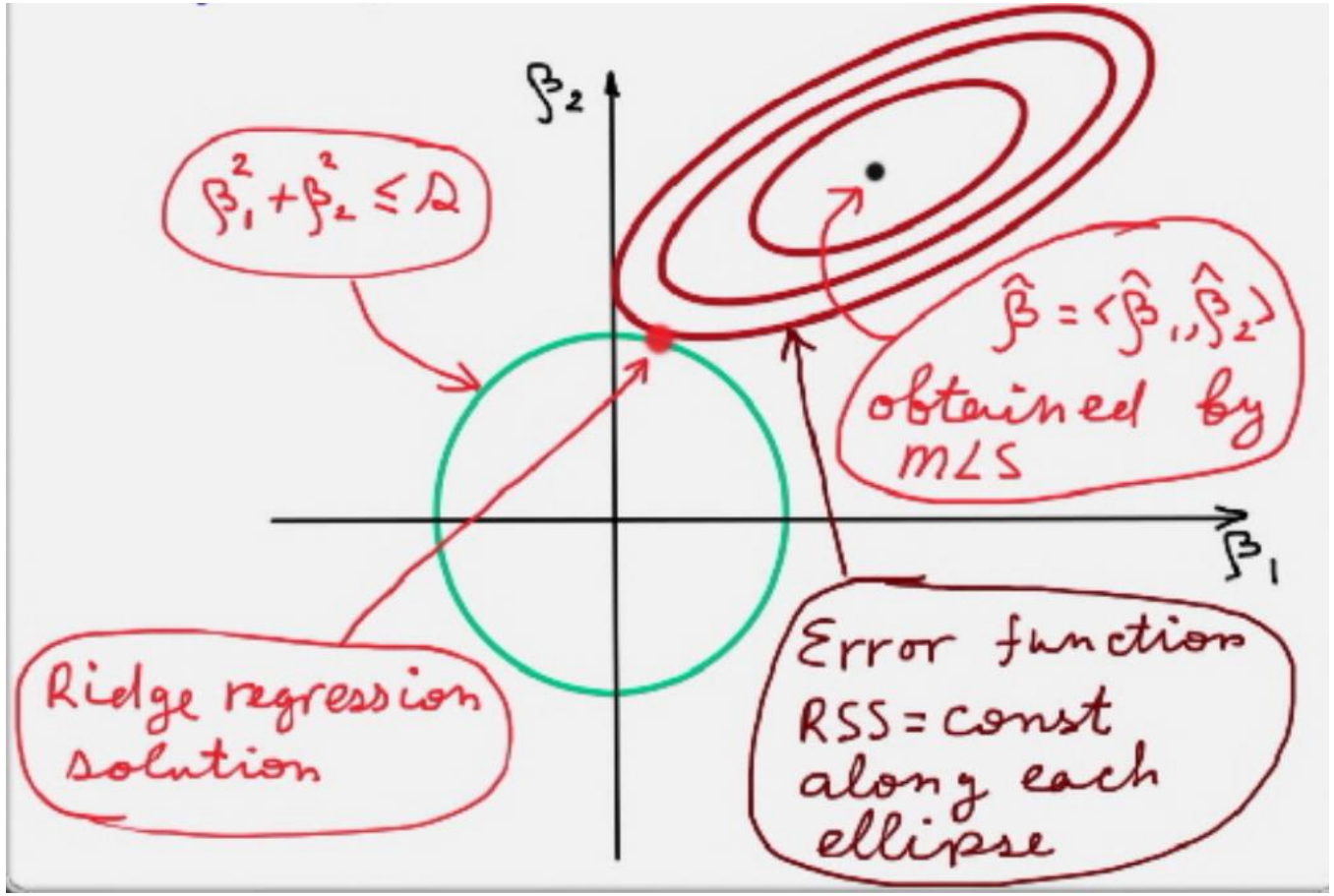
$$RSS\left(\boldsymbol{\beta}; \mathbf{y}, \mathbf{x}\right) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

- In order to "encourage" the coefficients to be closer to zero ridge regression replaces minimization of RSS with minimization of

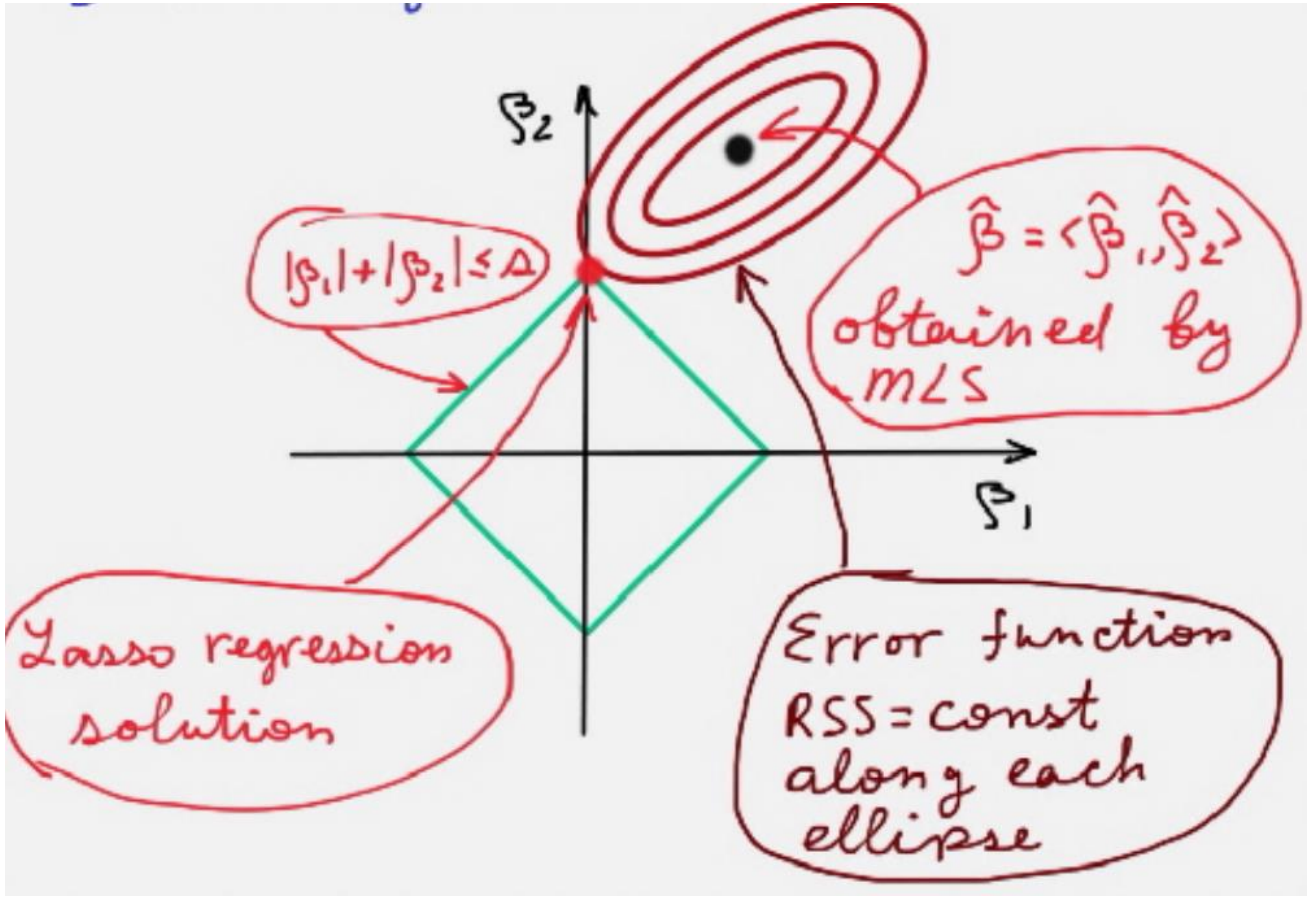
$$RSS\left(\boldsymbol{\beta}; \mathbf{y}, \mathbf{x}\right) + \alpha\left(\boldsymbol{\beta}\right) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

- Function $\alpha\left(\boldsymbol{\beta}\right) = \lambda\sum_{j=1}^{p}\beta_j^2$ is called regularizator. Parameter $\lambda \geq 0$ is a tuning parameter.

Ridge regression iv



Lasso regression ii



Recap

Lasso:

- produces sparse models
- is useful for strong feature selection in order to improve model performance, or to minimize the number of explanatory variables.
- can produce non-unique solutions (when some features are very strongly correlated)
- can produce very different solutions depending on slight changes in features (because of non-uniqueness)

Ridge:

- produces stable models with smooth non-zero coefficients across features.
- is useful for data interpretation, understanding what features, even when correlated, may be used in combination to predict the response.
- may tell you something about how the data itself was generated.

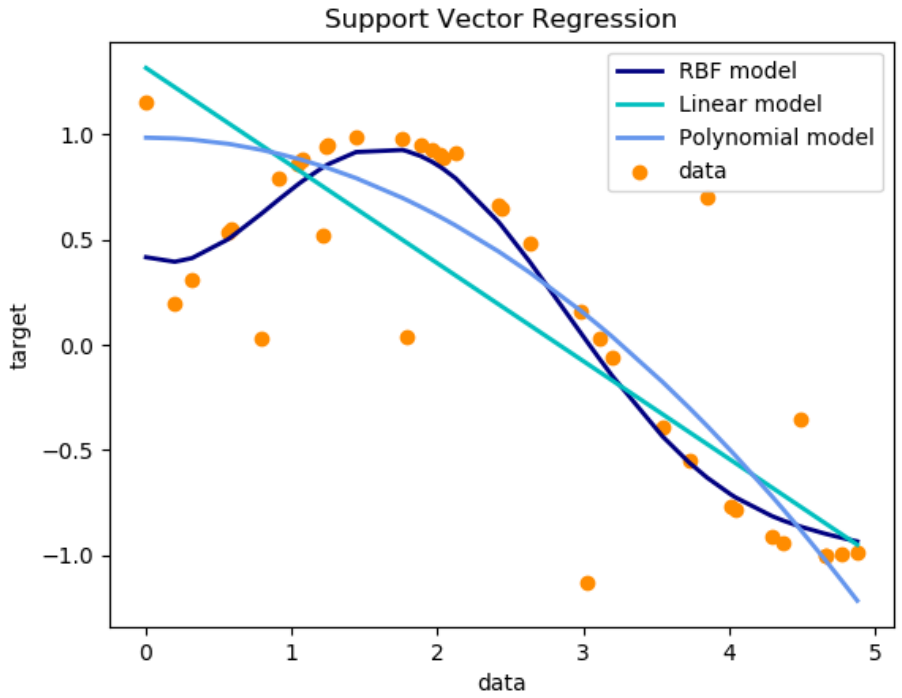
You can combine both Lasso and Ridge models into a single penalized model (that uses a weighted combination of Lasso and Ridge regression). This is called the `ElasticNet`.

Model evaluation

ERRORS AND APPROACHES

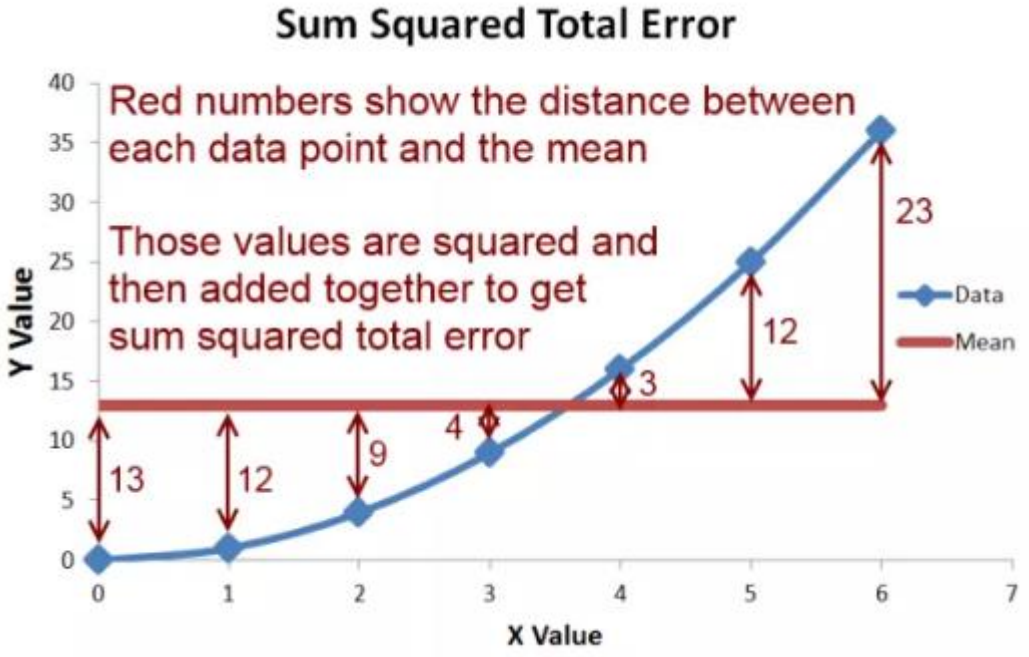
Improving on a base model

- Start with a baseline model
  - Logistic regression - Classification
  - Linear regression - Regression
  - Arima model – Time Series
- Create more complex models to try to improve the model fit and select the model with the highest predictive power



Calculating error for regression

- MAE – Mean Absolute Error
- MSE – Mean Squared Error
- RMSE – Root Mean Squared Error
- MAPE – Mean Absolute Percentage Error
- R-squared – proportion of variance explained



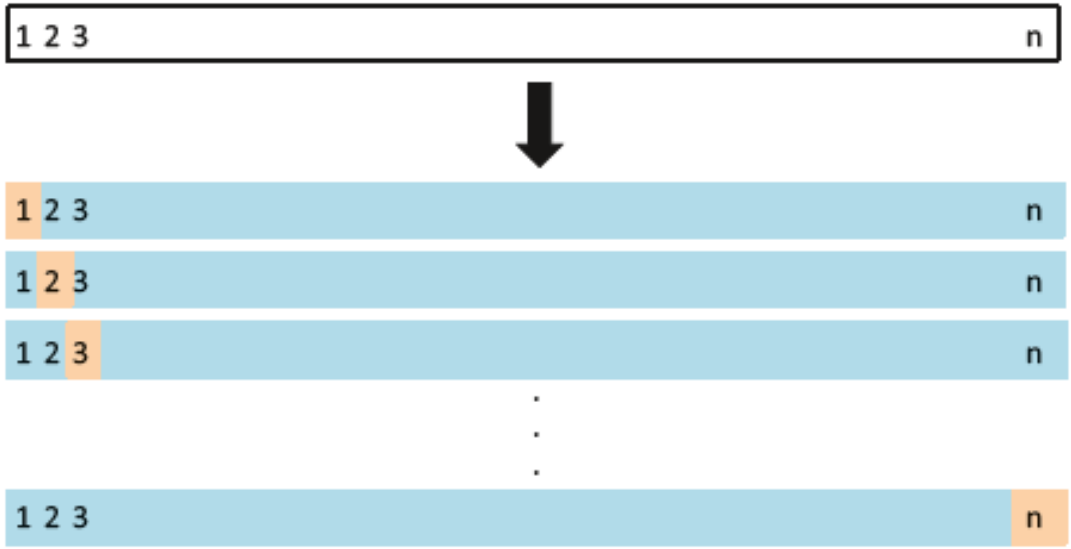
Validation set approach

- Divide the entire set into two parts
- Fit the model on the training set
- Use that fitted model to predict on the test set
  - aka validation set, hold-out set
- Examine error metric for fitted model

LOOCV - Leave-one-out cross-validation

- Split a data set with n observations into a test set of 1 and a train set of n-1
- Calculate the MSE
- Repeat procedure n times
- Aggregate the error metrics

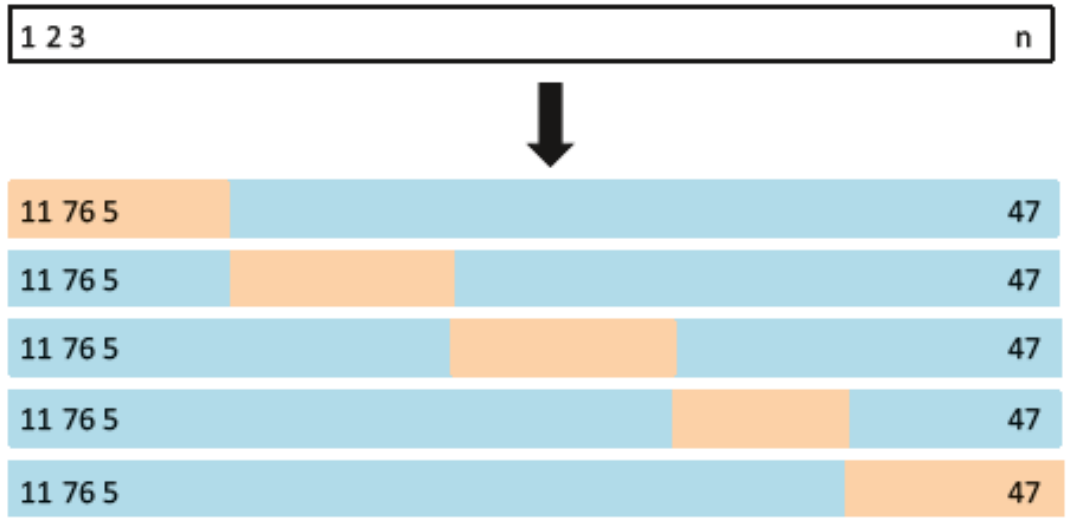
$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i.$$



K-fold cross-validation

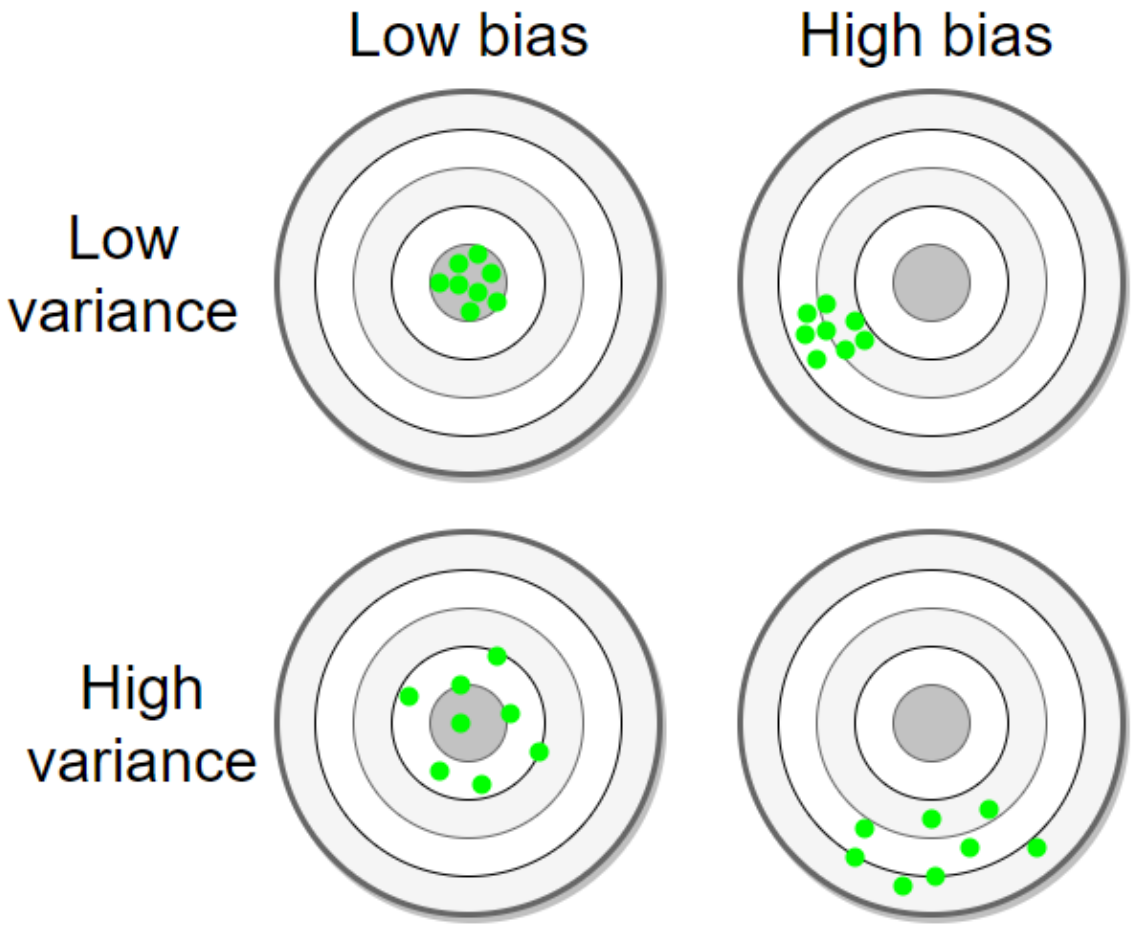
- Randomly divide the set into k groups or folds of equal size
- The first fold is a hold out, method is fit on remaining k-1 folds
- MSE is calculated on the hold olut fold
- Repeat process k times and average MSEs

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$



Bias-variance tradeoff for K-folds CV

- LOOCV gives unbiased predictions
- KFCV will give more biased predictions
- LOOCV gives a higher variance
  - Each of the n fitted models are trained on an almost identical set of data
- KFCV is fit with less correlated sets
- The mean of many highly correlated quantities has **higher variance** than the man of many quantities that are not as highly correlated
- K=5 or k=10 have been shown to yield test error estimate rates that don't suffer from very high bias nor from very high variance



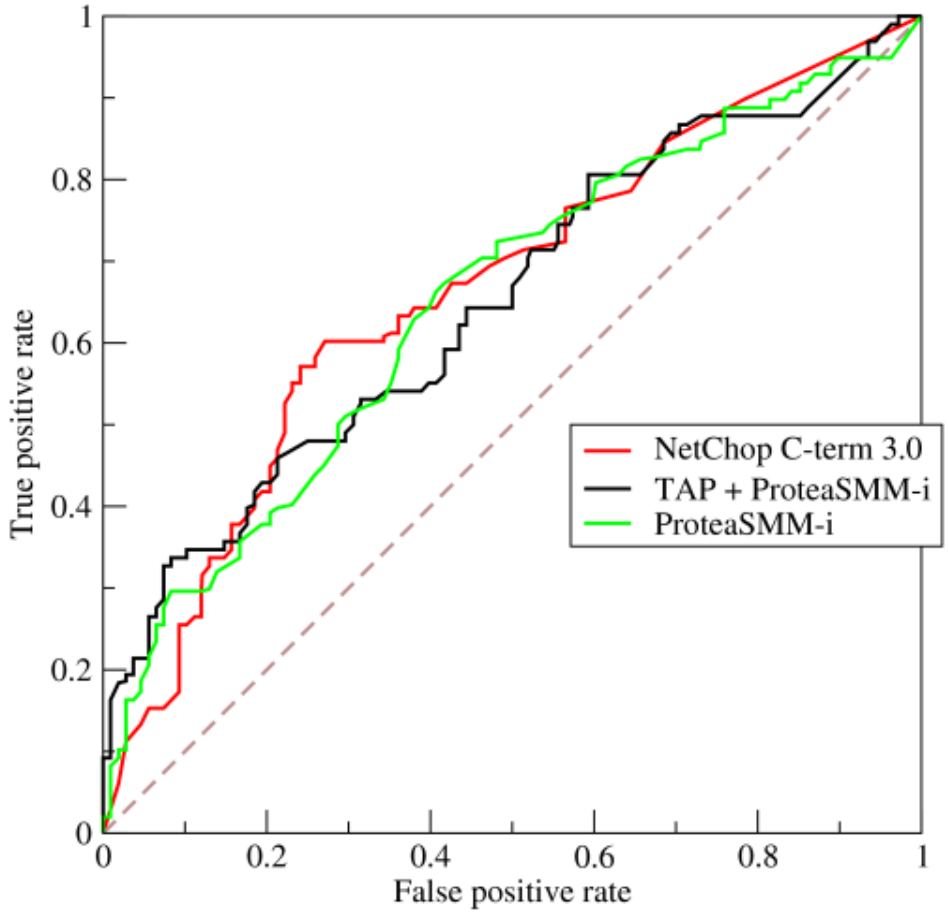
Calculating error for classification

- Columns are labeled with actuals
  - Positive or negative
- Rows are labeled with predictions
  - Yes or No
- Main diagonal are correct counts
- False positives are negative instances classified as Yes
- False negatives are positive instances classified as No

Predictions	Actuals: p	Actuals: n
Y	True positives	False positives
N	False negatives	True negatives

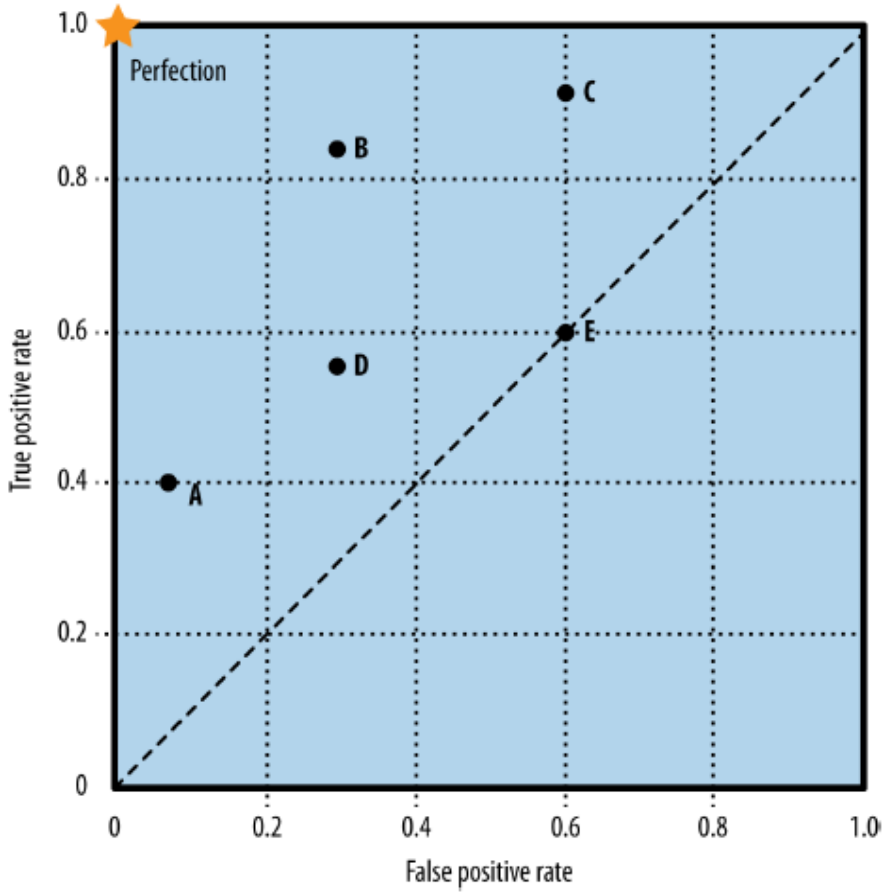
ROC graph

- Used to show the entire space of performance possibilities
- 2 dimensional plot of a classifier
- X axis: False positive rate
- Y axis: True positive rate
- ROC depicts relative tradeoffs that a classifier makes



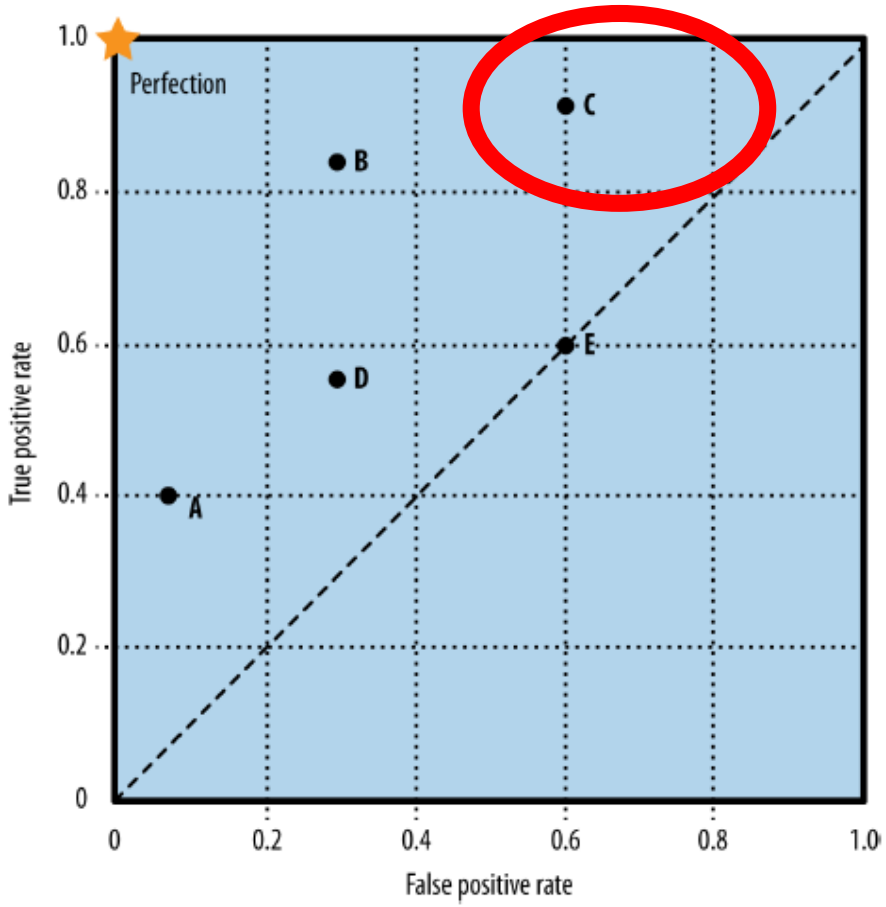
TPR vs FPR

- (0,0) – never issuing a positive classification
  - Commits no False positives, but also gains no True positives
- (1,1) – unconditionally issuing positive classifications
- (0,1) – perfect classification – Gold star
- (0.5, 0.5) - random guessing
- In order to move away from the diagonal, the classifier must exploit some info in the data



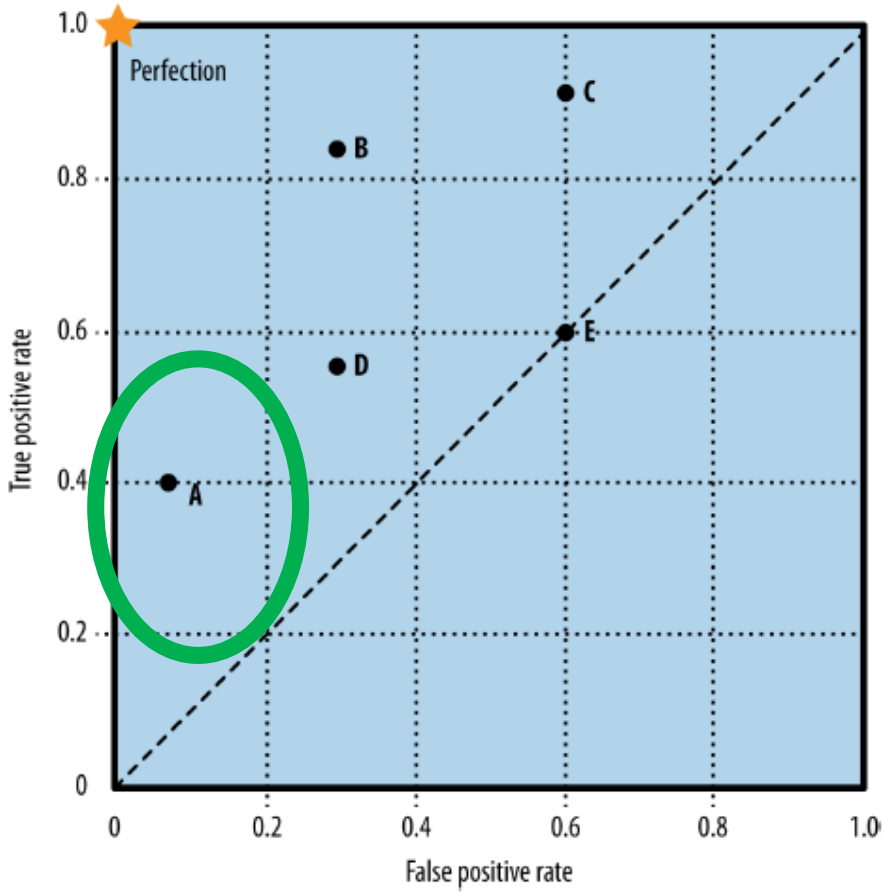
What do the sections mean?

- Considered permissive classifiers
- They make positive classifications with weak evidence
- Classify nearly all positives correctly
- High FPR



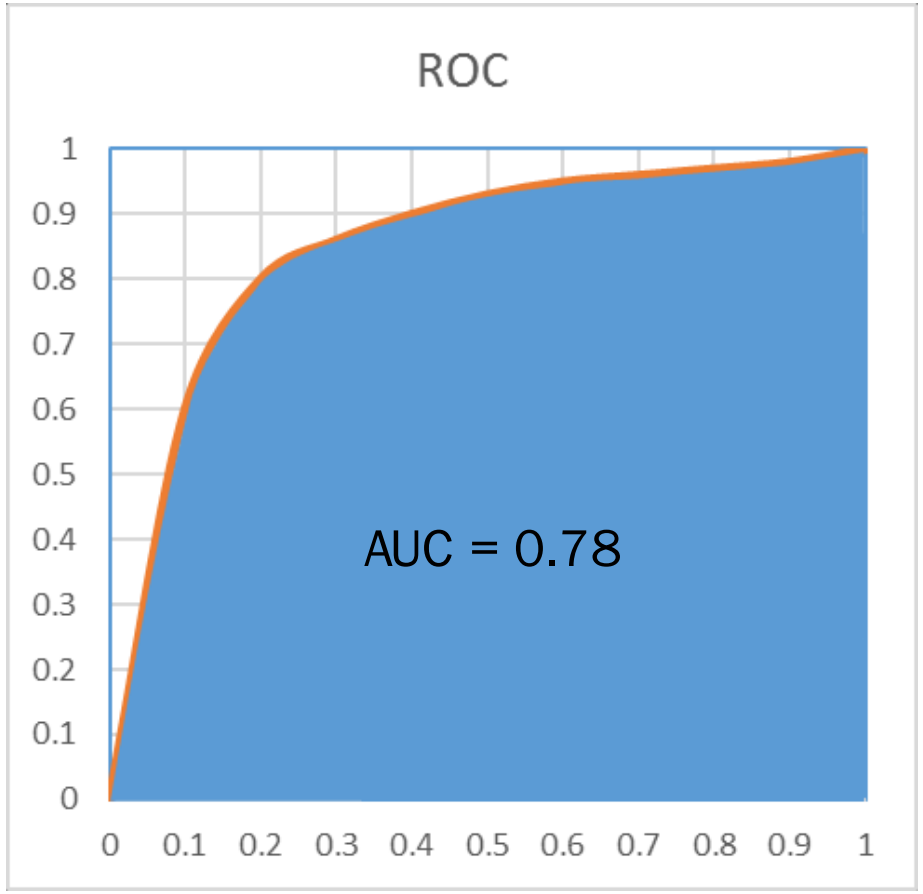
What do the sections mean?

- Considered conservative classifiers
- Raise alarms only with strong evidence
- Make few false positives
- Have low TPR as well
- Imagine if there are many negative examples
  - Then even a moderate false alarm



AUC is a summary statistic

- AUC ranges from zero to one
- Useful as a single number to summarize performance
- Common tool for visualizing model performance for classification
- However, this is not intuitive for business stakeholders



Summary of ROC curves

Valuable visualization for data scientists

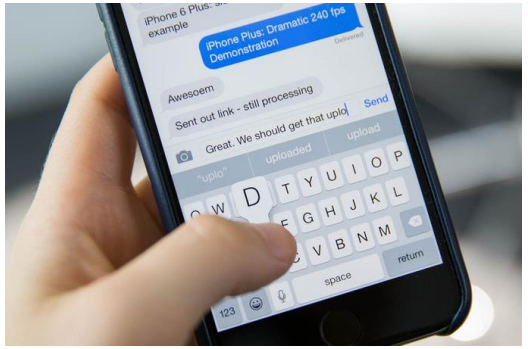
Display the trade-offs that each model is making

AUC is a summary statistic for comparison

Business stakeholders don't understand

Value in Lift curves or cumulative return curves

Predicting with our model



Acknowledgments

Thanks to all you!