2019 Predictive Analytics Symposium

Session 24: B/I - How Can an Actuary Become a Data Scientist?

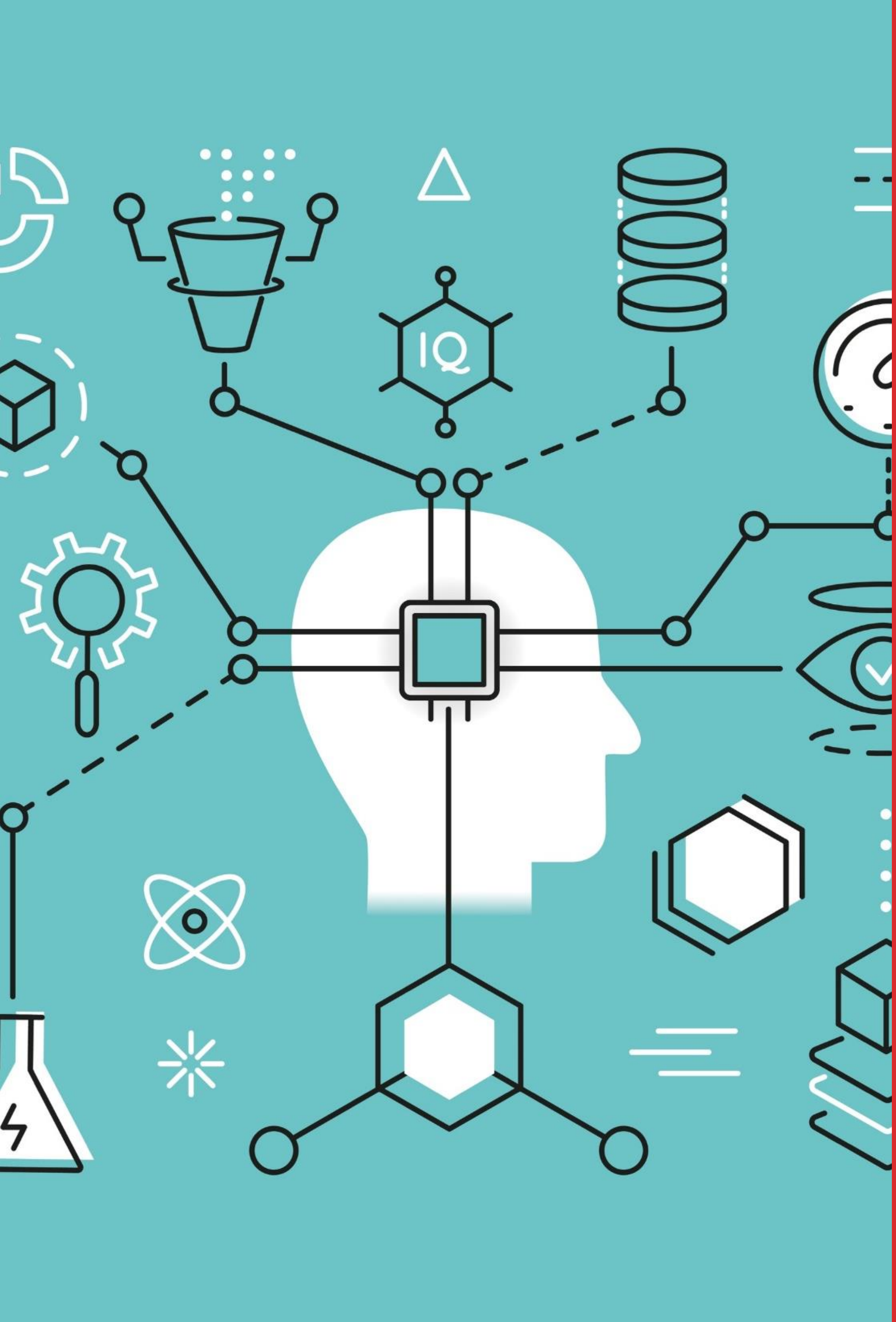
2019 PREDICTIVE ANALYTICS SYMPOSIUM

How Actuary can become Data Scientist

Richard Xu, PhD, FSA

September 2019

SOCIETY OF ACTUARIES®



RGA

How Actuary can become Data Scientist

Richard Xu, PhD, FSA

VP & Actuary

Head of Data Science

RGA

September 2019

Agenda

- ➢ What actuary already knew
- ➢ What actuary may not know
- ➢ Basic models beyond OLS
  - Generalized Linear Model
  - Decision tree & more
  - Clustering & more
- ➢ What is next?
- ➢ Conclusion, Q & A

RGA

What actuary already knew

- Are you familiar with the following terms?
  - Ordinary Least Square (OLS)
  - Time Series
- Linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon = \Sigma_i \beta_i X_i + \varepsilon = \overline{\boldsymbol{X}}\overline{\boldsymbol{\beta}} + \varepsilon$$

  - Y target variable, X_i predictor variable, ε error term/noise
  - β_i parameters to be estimated
- Underlying Assumptions for a valid LM
  - Y: homogeneity, representative of population, independence between observations
  - X: fixed, error-free
  - Normality, $\varepsilon \sim N(0,\sigma^2)$
  - (linearity)

Ordinary Least Squares

- Ordinary Least Squares(OLS)

$$\hat{\beta} = arg\min(RSS) = arg\min(\Sigma_i(\hat{y}_i - y_i)^2) = arg\min\left(\Sigma_i(\Sigma_j \beta_j X_{ij} - y_i)^2\right)$$

  - For a simple regression

$$\widehat{\beta_1} = (\Sigma x_i y_i - \tfrac{1}{n}\Sigma x_i \Sigma y_i)/(\Sigma x_i^2 - \tfrac{1}{n}(\Sigma x_i)^2), \quad \widehat{\beta_0} = \bar{y} - \widehat{\beta_1}\bar{x}$$

- Identical to Maximum likelihood estimator
  - More robust and consistent approach

$$\hat{\beta} = arg\max(L(X,Y,\beta)) = arg\min(-\ln(L(X,Y,\beta))) = arg\min(\Sigma_i(y_i - \hat{y}_i(\mu_i))^2)$$

if normal distribution

- Use adj R^2 to compare fitness of models

$$1 = \frac{RSS}{TSS} + \frac{ESS}{TSS}$$

- portion that has been explained by OLS model
- portion of TSS for the error

Define $R^2 = \frac{RSS}{TSS} = 1 - \frac{ESS}{TSS} = \frac{\Sigma_i(Y_i - \widehat{Y}_i)^2}{\Sigma_i(Y_i - \bar{Y})^2}$, but it is biased

Adjusted $R^2 = 1 - \frac{ESS}{TSS} * \frac{n-1}{n-k} = 1 - (1-R^2) * \frac{n-1}{n-k}$

Why actuary did not use OLS

- Processes are inherently linear, or can be well-approximated by LM
- Effectiveness & Completeness
  - OLS makes very efficient use of the data; good results with relatively small data sets
  - Identical to maximum likelihood estimation
- Easy to understand and communicate
  - theory is well-understood; Results are easy to communicate
- Great! but wait …
- There are several issues with OLS
  - Validation of assumptions - Normal w/ constant σ^2, independent, homogeneous
  - Unbounded data, non-negative value
- How about insurance application? Distribution of data, variance structure
  - Binomial for rate (mortality/lapse/UW, etc.), $\sigma^2 \sim r(1-r)$
  - Poisson for claim count, ~ mean
- <u>OLS may not be applicable in insurance, but you know lots about modeling</u>

What actuary may not know

Machine Learning & Statistical Techniques

- Generalized Linear Model (GLM)
- Random Forest
- XG-boost machine
- Gradient Boosting
- Ada Boosting
- Support vector machine
- Ensemble method
- Survey Data Analysis
- Genetic Algorithms
- Sentiment Analysis
- Markov chain Monte Carlo (MCMC)
- Optimization Methods
- Feature engineering
- Decision Trees (CART/MARS)
- Neural Networks / Deep learning
- Bayesian Analysis
- Classification/Association
- Analysis of Variance
- Mixed Models
- Categorical Data Analysis
- Multivariate Analysis
- Survival Analysis
- Cluster Analysis (e.g. K-Means)
- Non-Parametric Analysis
- Text mining

PM terminology

Supervised vs. Unsupervised

- Supervised: estimate expected value of *Y* given values of *X*
  - Example: OLS/LM, GLM, Cox, NN, etc.
- Unsupervised: find interesting patterns amongst X; no target *Y*
  - Example: Clustering, Correlation / Principal Components

Classification vs. Regression

- Classification: segment observations into 2 or more categories
  - Example: fraud vs. legitimate, lapsed vs. retained, UW class
- Regression: predict a continuous amount,
  - Example: dollars of loss for a policy, ultimate size of claim

Parametric vs. Non-Parametric

- Parametric Statistics: probabilistic model of data
  - Example: Poisson Regression(claims count), Gamma (claim amount)
- Non-Parametric Statistics: no probability model specified
  - Example: classification trees, NN

8

Generalized Linear Model

- Generalized Linear Model(GLM)
  - Major focus of PM in insurance industry
  - Include most distributions related to insurance
  - Great flexibility in variance structure
  - OLS model is a special case of GLM
  - (Relatively) Easy to understand and communicate
  - Multiplicative model intuitive & consistent with insurance practice
- 3 components
  - *Random component*
  - *Systematic component*
  - *link function*

Generalized Linear Model

Random component

Observations $Y_1, . . . , Y_n$ are independent w/ density from the exponential family

$$f_i(y_i;\theta_i,\phi) = exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i,\phi)\right\}$$

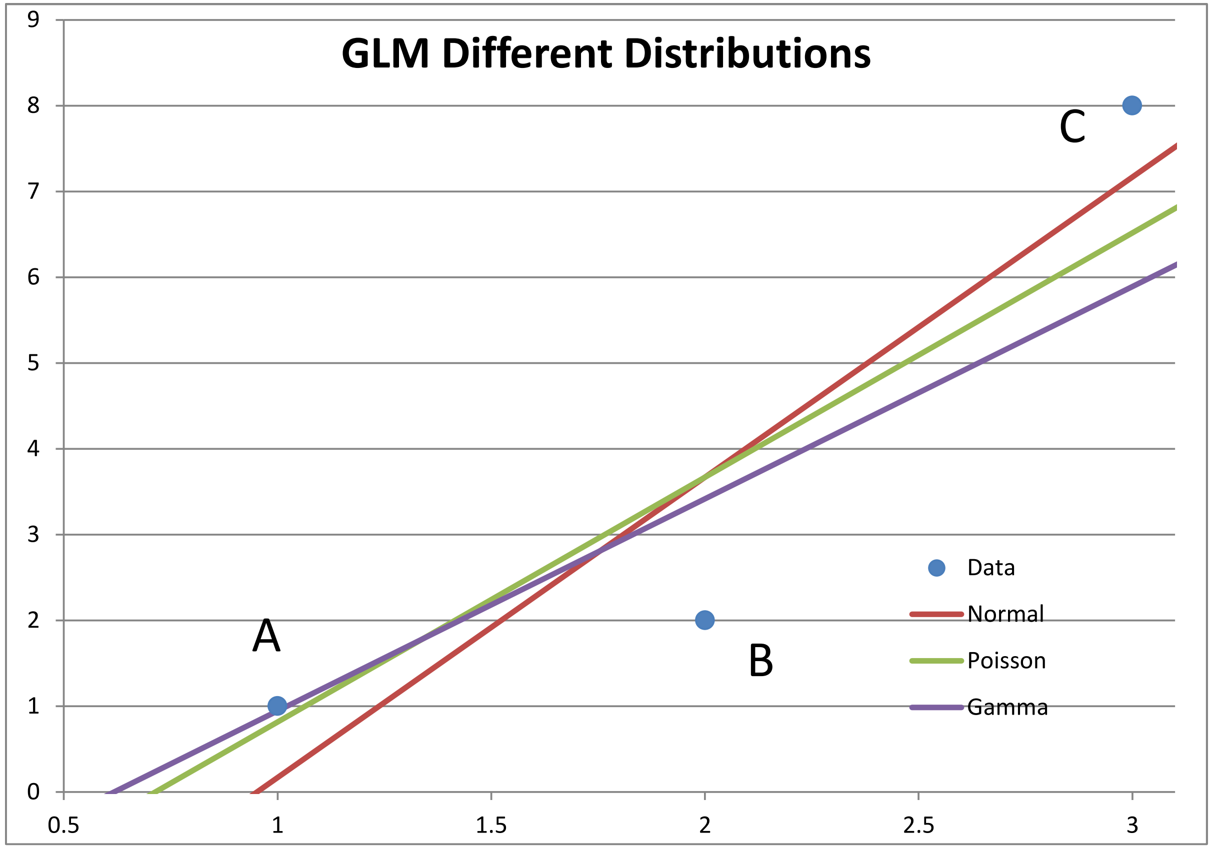
From maximum likelihood theory,

$$E(Y) = \mu = b'(\theta), \qquad var(Y) = b''(\theta)a(\phi) = a(\phi)V(\mu)$$

- Each distribution is specified in terms of mean & variance
- Variance is a function of mean

	Normal	Poisson	Binomial	Gamma	InverseGaussian
Name	$N(\mu,\sigma^2)$	$P(\mu)$	$B(m,\pi)/m$	$G(\mu,\nu)$	$IG(\mu,\sigma^2)$
Range	$(-\infty,+\infty)$	$(0,+\infty)$	$(0,1)$	$(0,+\infty)$	$(0,+\infty)$
$b(\theta)$	θ^2	e^θ	$\ln(1+e^\theta)$	$-\ln(-\theta)$	$-(-2\theta)^{1/2}$
$\mu(\theta)$	θ	e^θ	$e^\theta/(1+e^\theta)$	$-1/\theta$	$(-2\theta)^{-1/2}$
$V(\mu)$	1	μ	$\mu(1-\mu)$	μ^2	μ^3

Why distribution will affect results



Variance of different distributions

- Gaussian, constant
- Poisson, ~ mean
- Gamma, ~ mean^2

Generalized Linear Model

- *Systematic* component

  A linear predictor $\eta_i = \sum_j x_{ij}\beta_j = X\beta$ for observation *i*

- *link function*

  $\eta_i = g(\mu_i)$, random & systematic are connected by a smooth & invertible function

	Identity	Log	Logit	Reciprocal
$g(\mu_i)$	x	$\ln(x)$	$\ln(\frac{x}{1-x})$	$1/x$
$g^{-1}(\eta_i)$	x	e^x	$\frac{e^x}{1+e^x}$	$1/x$

Log is unique in insurance application s.t. all parameters are multiplicative

- $y = \exp(\sum_j x_{ij}\beta_j) = \prod_j \exp(x_{ij}\beta_j) = \prod_j \exp(\beta_j)^{x_{ij}} = \prod_j {f_j}^{x_{ij}}$
- Consistent with most insurance practices
- Intuitively easy to understand and communicate

RGA

Generalized Linear Model

- Solve for parameters (β) by maximum likelihood
  - Closed form for small data and simple model
  - Iterative numerical techniques for large data set & complex model
    - $\beta_{n+1}=\beta_n - \boldsymbol{H}^{-1} \cdot s$, similar to Newton's method $x_{n+1}=x_n - f(x_n)/f'(x_n)$
  - Use statistical analysis application, such as *R*
- Compare OSL and GLM

	Random	Systematic	Link
OLS	Normal only	$\eta_i = \sum_j x_{ij}\beta_j$	$E(y_i) = \eta_i$
GLM	Various distribution		$g(E(y_i)) = \eta_i$

- Great flexibility
  - Various distribution, variance structure
  - Prior weight and the credibility of data
  - Offset of data

13

Where we go from GLM

- More regression models
  - Survival Models (Cox Proportional Hazard)
  - Generalized Additive Models (GAM)
  - Multilevel/Hierarchical Linear Model(HLM)
- Support vector machine
  - Instead of a linear boundary that are affected by all data points to separate classes, an optimal boundary is selected to maximize the gap between classes
- Neural network / Deep Learning
  - Logistic model is the simplest neural network model

Decision Tree Model

- Decision Tree model, (Classification And Regression Tree - CART)
  - Both classification and regression
  - Non-parametric approach (no requirement on data structure)

- CART tree is generated by repeated partitioning of data set
  - Data is split into two partitions (binary partition)
  - Partitions can also be split into sub-partitions (recursive)
  - Until data in end node(leaf) is homogeneous (more or less)

- Results are very intuitive
  - Identify specific groups that deviate in target variable
  - Yet, algorithm is very sophisticated

15

Decision Tree Model

Recursive Partitioning

- Take all data points
- Consider *all* possible values of *all* variables
- Select the variable/value **($X=t_1$)** that produces the greatest "separation"
  - ($X=t_1$) is called a "split".
  - If $X< t_1$ then send the data to the "left"; otherwise, to the "right"
- Repeat same process on these two "nodes"
  - Result is a "tree"; uses binary splits
- Stop split data until certain criteria are meet

Two Core Questions

- How to find split points
  - Which variable among all, at which value or category, what criterion to use
- When to stop splitting
  - Avoid saturated model

RGA

Decision Tree Model

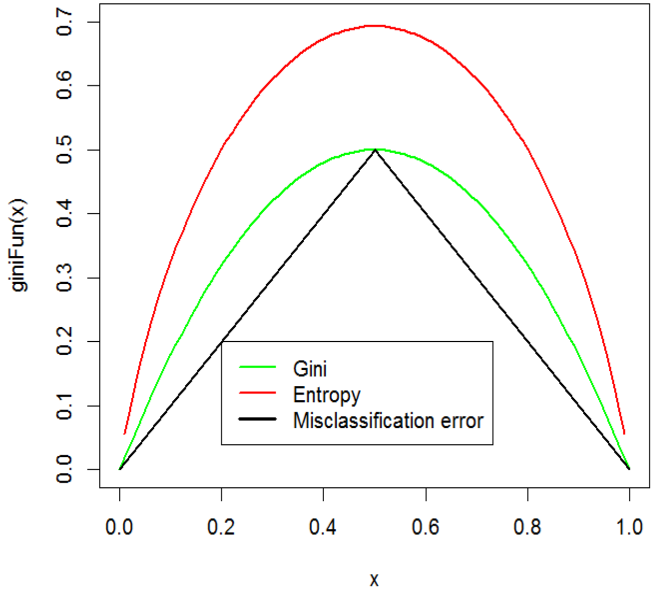
Splitting Point

- "Separation" defined in many ways; different for regression & classification
- Regression Trees: use sum of squared errors
  - $SSE_p = \sum_i (y_i - \mu)^2$
  - $SSE_c = \sum_i (y_i^L - \mu^L)^2 + \sum_i (y_i^R - \mu^R)^2$
  - Select $X=t_1$ such that $\max\limits_{x_i,t}(SSE_P - SSE_C)$
- Classification Trees: use measures of purity/impurity
  - Intuition: an ideal tree model would produce completely pure nodes
  - *Gini Index* - purity of a node $f(p) = p(1-p)$

$f(p) = \sum_i p_i(1-p_i) = 1 - \sum_i p_i^2,\ \ p_i = \text{freq of class i}$

  - *Entropy* – information index $f(p) = -plog(p)$

$f(p) = \sum_i -p_i\, log(p_i) = -plog(p) - (1-p)log(1-p)$

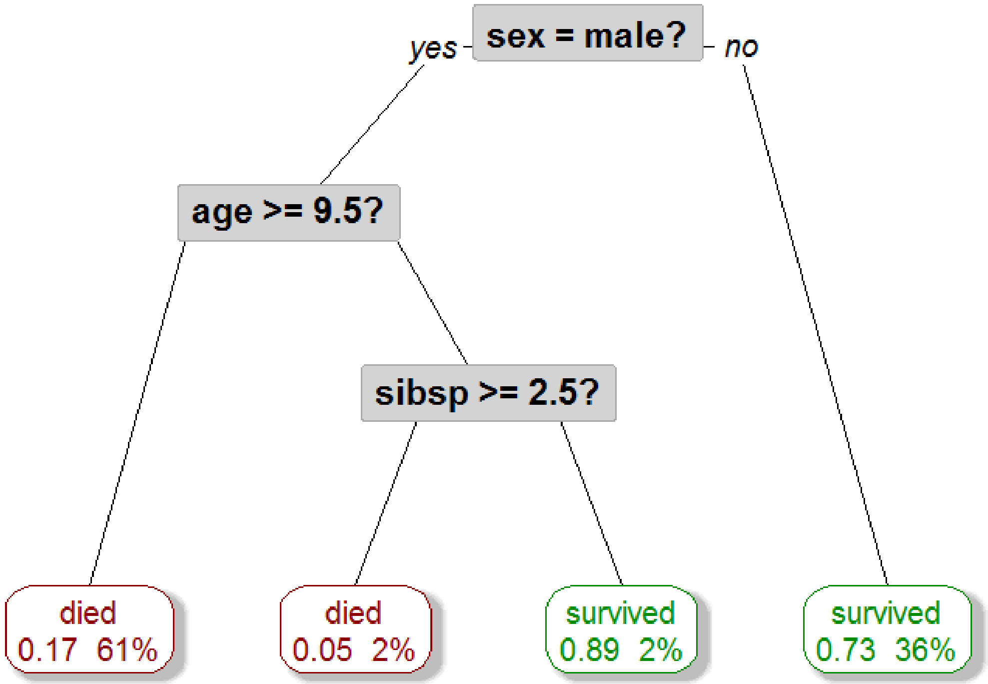


Decision Tree Model

Surrogate Splits

- Problem: if missing data on x_i, we don't know how to assign the object
- Solution: we can use a similar split on another variable that is associated (correlated); we use these (surrogate) splits to assign the object to the class
  - Missing value can be solved in algorithm level

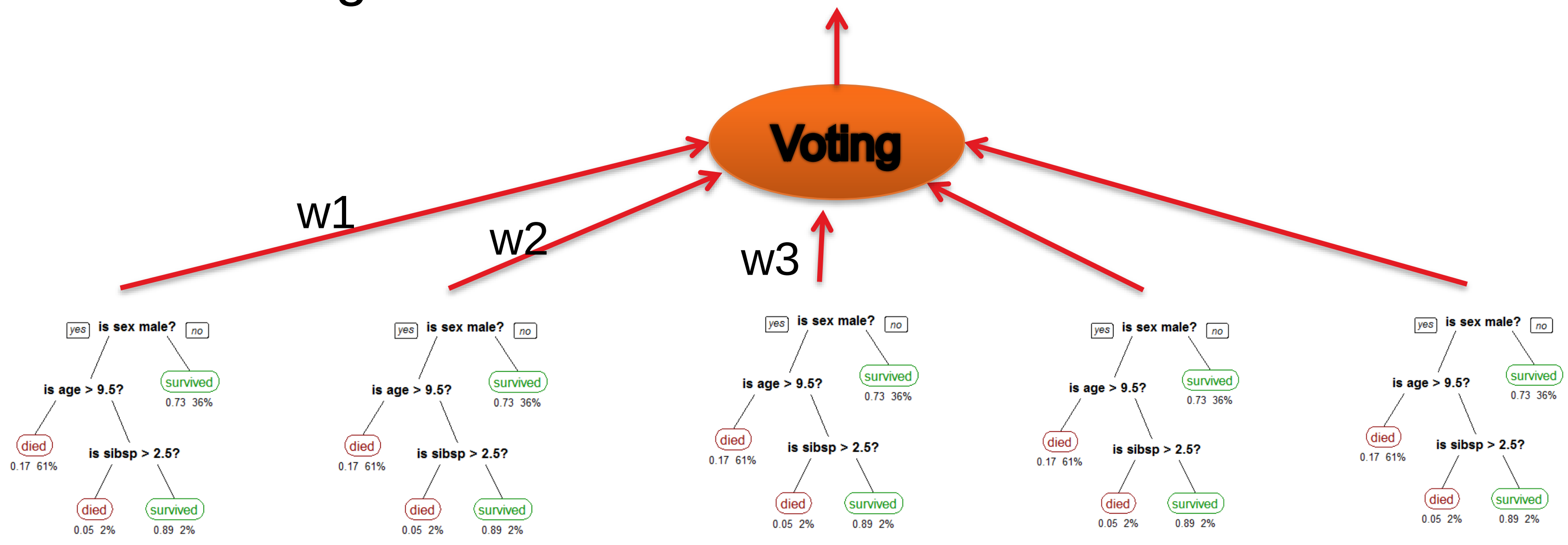
A simple example - Titanic survivor model



Where we go from Decision Tree

Decision tree based model

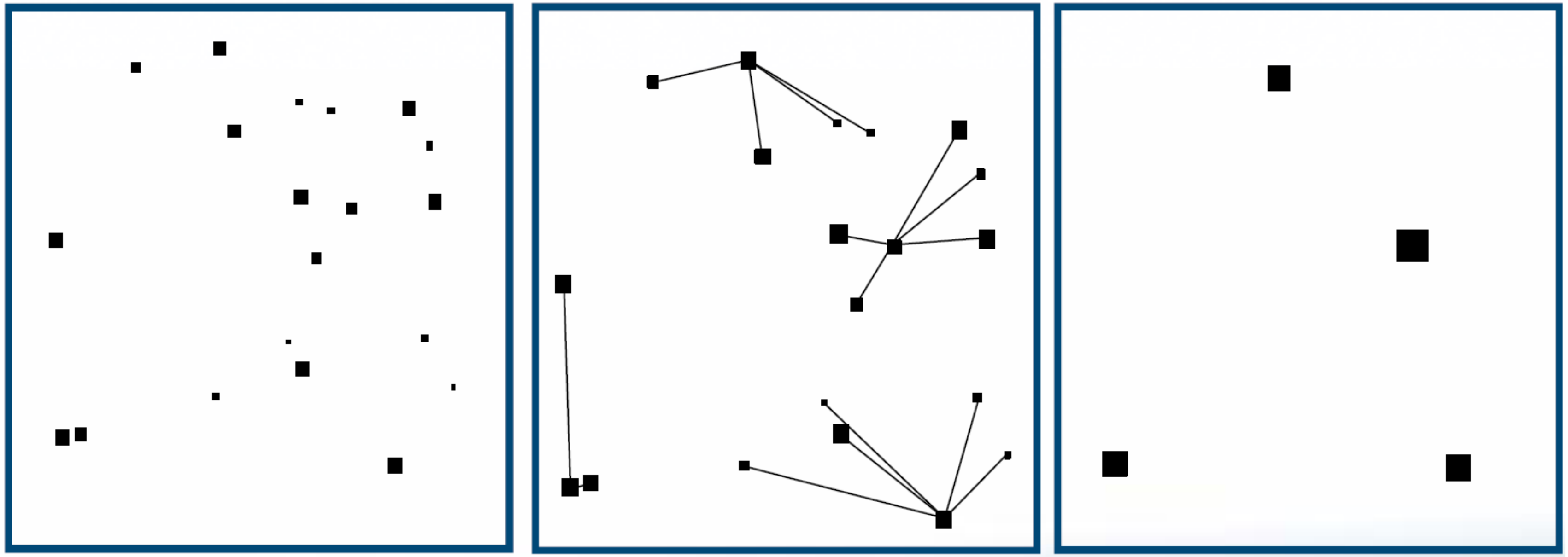
- Random forest
- XG-boost machine
- Gradient Boosting
- Ada Boosting



Data Clustering

Clustering algorithm

- ✓ Find similarities in data according to features in data & group similar objects into clusters
- ✓ Unsurprised (no pre-defined), classification, non-parametric
- ✓ How to measure similarities/dissimilarities, e.g. distance
  - Numeric, categorical, and ordinal variables
- ✓ Partitioning (k-means), Hierarchical, Density-based, etc.



Data Clustering

Algorithm

- Partitioning algorithms - K-measn/k-medoids
  - Maintain k clusters with k known; place points into their “nearest” cluster
- Hierarchical (Agglomerative)
  - Objects are more related to nearby objects than to objects farther away; objects are connected by distance; how to define “nearby” object

K-Means Algorithm

1. Select K points as initial centroids, with a given k
2. Repeat
3. Form K clusters by assign each points to its nearest centroid
4. Re-compute the centroids of each cluster
5. Until centroids do not change

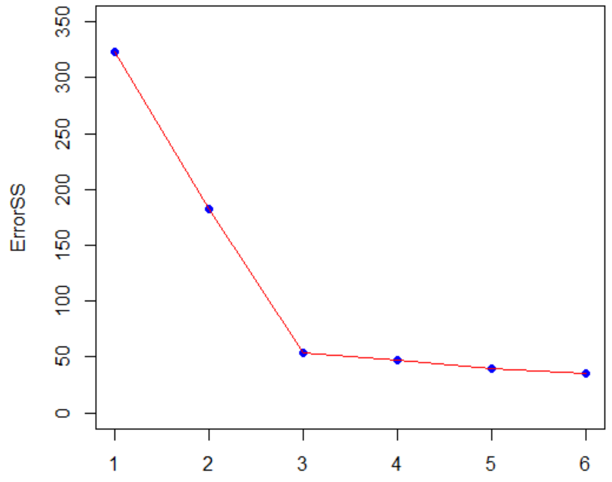
Data Clustering

Define Distance

- Euclidean: $d(x_i, x_j) = (\sum_k (x_{ik} - x_{jk})^2)^{1/2}$ (p=2), easy to understand, but not scale invariant
- Manhattan: $d(x_i, x_j) = \sum_k |x_{ik} - x_{jk}|$ (p=1), city-block distance
- Chebychev: $d(x_i, x_j) = max_k |x_{ik} - x_{jk}|$ (p→∞),
- Minkowski: $d(x_i, x_j) = (\sum_k (x_{ik} - x_{jk})^p)^{1/p}$
- Others like Pearson correlation, Spearman,Canberra, Jaccard, binary, ...

Standardization / Normalization

- Values of variables may have different units
- Variable with high variability/range will dominate metric, & lead to bias



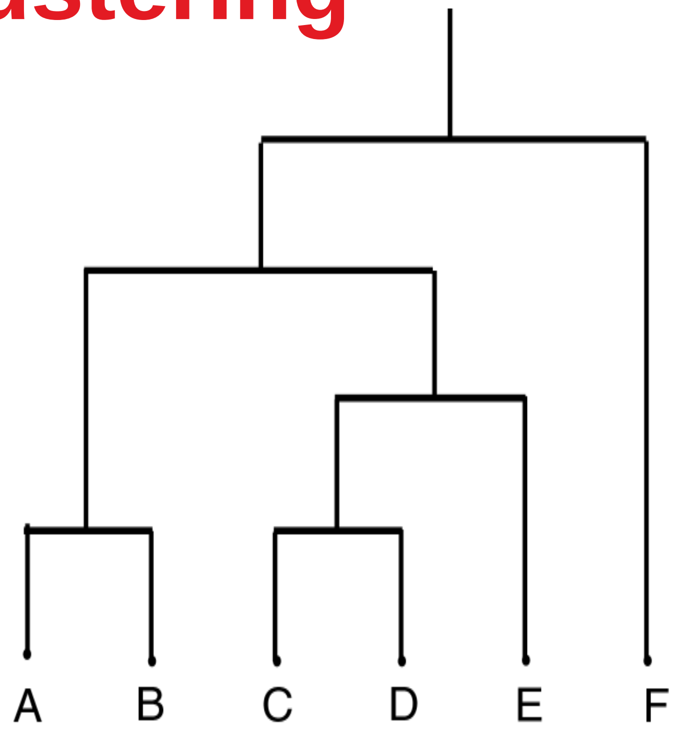
How to determine K

- Business reasons could dictate k
- Try different k, looking at the change in the average distance to centroid, as k increases; error falls rapidly until right k, then changes little

Data Clustering

Comments on K-Means

- Strength: simple, very efficient, & fast
- Weakness
  - Applicable only when *mean* is defined, (categorical?)
  - Need to know *k* in advance
  - Unable to handle noisy data & *outliers;* sensitive to outliers
  - Not suitable for clusters with *non-convex shapes*
  - *Maybe sensitive to initialization*
- There are variants of *k-means*



Hierarchical clustering

- Bottom up (aglomerative) or top down (divisive/deglomerative) produce a dendrogram
- Important questions - how to represent a cluster of more than one point, & how to determine the "nearness" of clusters?
  - ***Single Link***: smallest distance between points
  - ***Complete Link:*** largest distance between points
  - ***Average Link:*** average distance between points
  - ***Centroid:*** distance between centroids

What is next for Actuary?

- You have solid education background in statistics
- You already have the business knowledge
- Pick up the new skills of data analytics
  - Refresh yourself with the basics of modeling
  - Learn a modeling application / language & practice with examples
  - Attend seminar, conference, training program, etc.
  - Start a project to apply the new skills
  - Link your new skills with your job & practice if possible

RGA

Conclusion

Actuary unique position

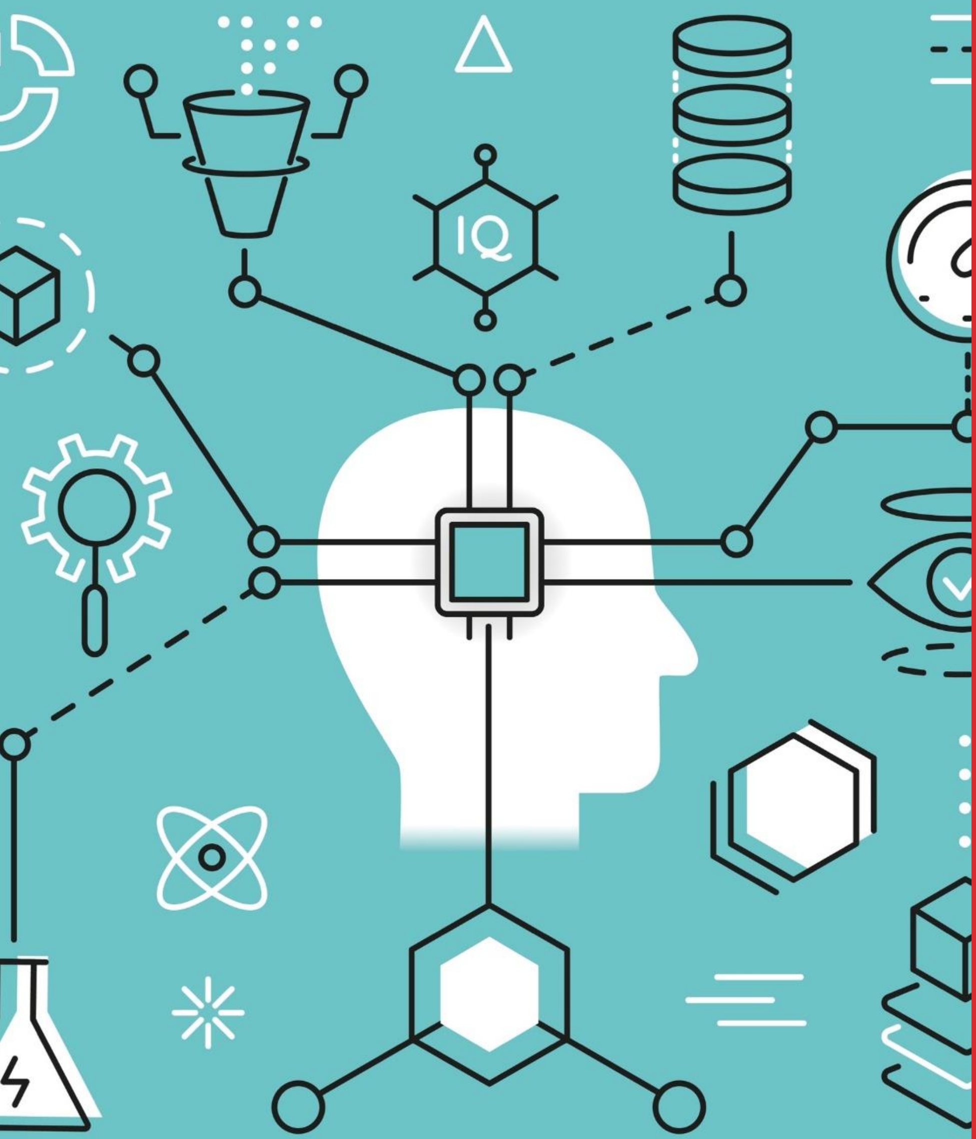
- Industry knowledge: domain knowledge is key in the predictive modelling process
- Data expertise: data is always the largest issue in data-driven applications

Challenge for actuary

- Solid math foundation, but need to learn modeling skills and new technology
- Combine the new skills with domain knowledge

Opportunity for actuary

- Data science is changing the insurance & will revolutionize how we run business
- Actuaries should lead the transforming by becoming data scientist or leading DS



RGA

How Actuary can become Data Scientist

Richard Xu, PhD, FSA

VP & Actuary

Head of Data Science

RGA

September 2019