Data Science Book

Ivan Bizberg

2023-07-17

Table of contents

Preface

Welcome to the world of data science, a captivating realm where art, mathematics, and technology converge to unlock the hidden insights lying dormant within vast troves of information. In this book, we embark on an exhilarating journey through the most important aspects of data science techniques, exploring their foundations, applications, and transformative potential.

The rapid advancement of technology and the explosive growth of data have ushered in an era of unprecedented opportunities. Every click, every transaction, and every interaction generates a data footprint that can be harnessed to drive innovation, solve complex problems, and shape the future. Data science equips us with the tools to make sense of this digital universe, enabling us to extract meaning from seemingly chaotic data and turn it into actionable knowledge.

At its core, data science is a multidisciplinary field that draws from various domains, including statistics, mathematics, computer science, and domain expertise. It encompasses a wide range of techniques, algorithms, and methodologies designed to collect, analyze, interpret, and visualize data to uncover patterns, make predictions, and generate insights.

> **Note**
>
> This book is in the process of being made and is intended to serve as a comprehensive compilation of the most important aspects of data science techniques and tools. However, it is important to note that the content presented herein reflects the author's personal understanding and interpretation of these concepts, which may occasionally deviate from the precise and mathematical definitions of certain data science principles.

1 Terms & definitions

1.1 Correlation

- **Correlation** measures the strength and direction of the linear relationship between two variables, but it standardizes the measure to fall between -1 and 1.

When to use it:

Correlation is a more useful measure than covariance when comparing relationships across different datasets or variables, as it removes the influence of the scales of the variables.

1.2 Correlation vs covariance

- **Covariance** measures the direction and magnitude of the linear relationship between two variables. It calculates how changes in one variable are related to changes in another variable. Covariance can take any value, positive or negative, depending on the nature of the relationship.

When to use it:

- Scaling and Interpretation: If you are primarily interested in the magnitude of the relationship between two variables, without the need for standardized interpretation, covariance can be used. Since covariance is not standardized, it preserves the original scale of the variables. This can be helpful when the units of measurement carry important information or when you want to maintain the original context of the data.
- Non-linear Relationships: If you suspect a non-linear relationship between variables, covariance can still provide insights into the direction and magnitude of the relationship, albeit without quantifying the strength in a standardized manner.

Example:

By calculating the covariance between height and weight, you can obtain a measure of how the two variables vary together.

Example:

If you were interested in comparing the relationship between height and weight with other datasets or variables, or if you wanted a standardized measure of the strength of the relationship, then calculating the correlation coefficient would be more appropriate. The correlation coefficient would provide a standardized measure between -1 and 1, allowing for easier comparison and interpretation across different contexts.

1.3 Correlation vs cointegration

Both are statistical concepts used to measure the relationship between variables.

- **Cointegration** measures whether the variables tend to move together over time, despite possibly having short-term fluctuations.

Example:

A man leaves the pub with his dog.
When the man and the dog first leave the pub, their paths are **correlated**. They generally move in the same direction, but the distance between the dog and the man has no actual limit. It increases at times, decreases at times, but is generally random and poorly defined. The direction of the two, however, is generally the same.
When the man leashes his dog to cross the road, they become **cointegrated**. Now, while their direction is still the same, their distance from one another is finite. The dog cannot move beyond the length of the leash from the man. ("What Is the Difference Between Correlation and Cointegration? Is Cointegration a Good Measure of Risk?" n.d.)

1.4 P-value

How surprised should we be by a result if the null hypothesis is true

Example:

A p-value of 0.001 indicates that if the null hypothesis tested were indeed true, then there would be a one-in-1,000 chance of observing results at least as extreme. It suggests that the observed data is unlikely to have occurred by random chance alone, and you may reject the null hypothesis in favor of the alternative/new hypothesis

1.5 Bias and Variance

We search a model with low bias and low variance

Bias: The inability for a machine learning method to capture the true relationship is called bias (can't capture underlying patterns = underfit)
Variance: The difference in fits between train set and test set (can't generalize on unseen data = overfit)

Red model: low variance: This model has low variance if a new point it would not substantially change the model fit (leads to under-fitting) high bias: Ineffective at modeling the data

Blue model: high variance: Small perturbations in the data will significantly change the model fit (leads to over-fitting) low bias: More complex and flexible allowing to model very good the data

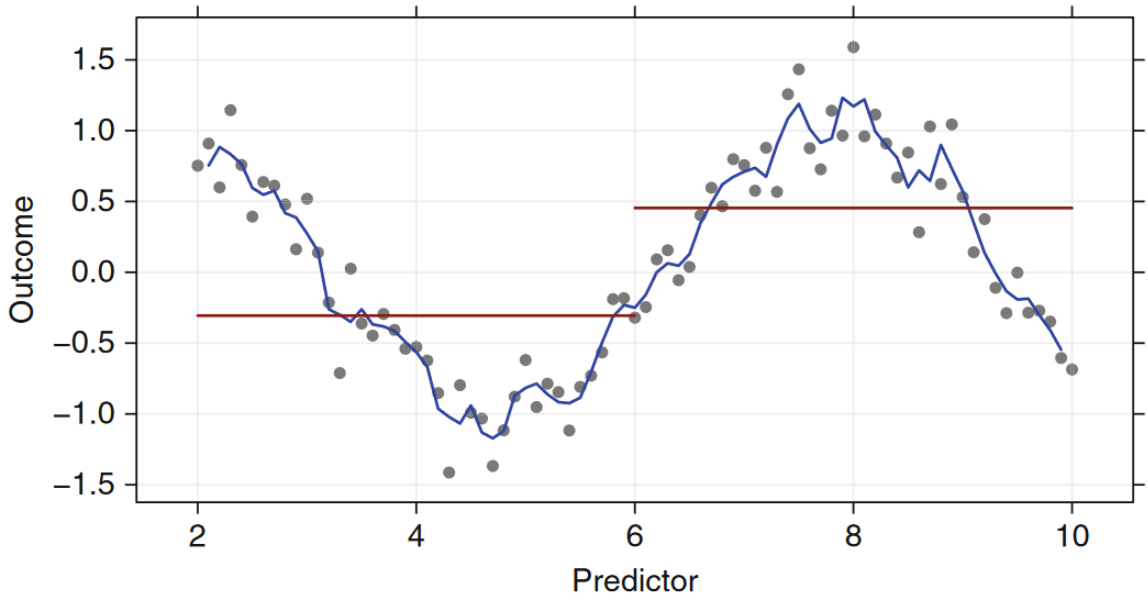


Figure 1.1: variance-bias trade-off

1.6 Parametric / Non-Parametric

ML models can be divided into two types:

– Parametric : uses a fixed number of parameters with respect to sample size – Non-Parametric : uses a flexible number of parameters and doesn't make particular assumptions on the data

1.7 Residual

Difference between the observed value and the predicted value

1.8 Regularization

It make the prediction less sensitive to the training data and reduces overfitting.

1.9 Quantiles

Median: The 50th percentile, which divides the data into two equal halves. Half of the data points are below the median, and half are above it.

Quartiles: The 25th, 50th, and 75th percentiles are called the first quartile (Q1), the median (Q2), and the third quartile (Q3), respectively. They divide the data into four equal parts, each containing 25% of the data.

Percentiles: These are any values that divide the data into 100 equal parts. For example, the 20th percentile is the value below which 20% of the data falls, and the 80th percentile is the value below which 80% of the data falls.

1.10 Greedy algorithms

When a procedure is greedy, it means that it does not reevaluate past solutions, it makes locally optimal choices at each step with the hope of finding a global optimum. Advantages: simplify the decision-making process and lead to a solution that is close to optimal. Disadvantages: not always guarantee finding the globally optimal solution

1.11 Extrapolation

Extrapolation is commonly defined as using a model to predict samples that are outside the range of the training data. To know if we can trust our model we need to compare the predictor space between the training data and new data.

- This can be done using **PCA**. If the training data and new data are generated from the same mechanism, then the projection of these data will overlap in the scatter plot. However, if the training data and new data occupy different parts of the scatter plot, then the data may not be generated by the same mechanism and predictions for the new data should be used with caution.

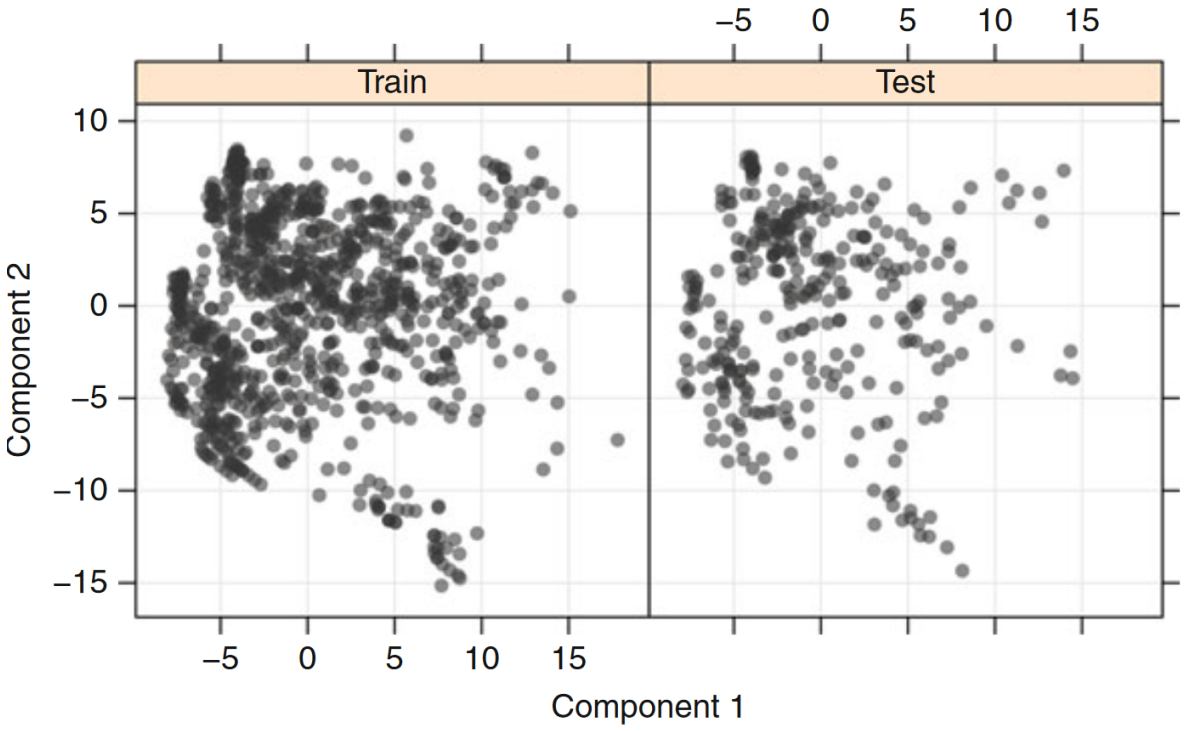


Fig. 20.7: PCA plots for the solubility data

Figure 1.2: PCA and Extrapolation: the training and testing data appear to occupy the same space as determined by these components.

- This can also be done using the following **algorithm**

1. Compute the variable importance for the original model and identify the top 20 predictors
2. Randomly permute these predictors from the training set
3. Row-wise concatenate the original training set's top predictors and the randomly permuted version of these predictors
4. Create a classification vector that identifies the rows of the original training set and the rows of the permuted training set
5. Train a classification model on the newly created data
6. Use the classification model to predict the probability of new data being in the class of the training set.

Algorithm 20.1: Algorithm for determining similarity to the training set

Figure 1.3: Algorithm and Extrapolation

2 Choose the machine learning model

2.1 Supervised models

Model	Mode	Allows n < p	
Linear regression	regression	no	centering and scaling, remove near-zer
Partial least squares	regression	depends	cente
Logistic regression	classification	no	centering and scaling, remove near-zer
Ridge regression	regression & classification	no	centering and scalin
Elastic net/Lasso	regression & classification	yes	ce
Support vector machines	regression & classification	yes	cente
MARS/FDA	regression & classification	yes	
K-nearest neighbors	regression & classification	yes	ce
Random forest	regression & classification	yes	
Boosted trees	regression & classification	yes	
Neural networks	regression & classification	yes	centering and scaling, remove near-zer
LDA	classification	no	
Naive Bayes	classification	yes	
C5.0	classification	yes	

(Kuhn and Johnson 2013)

2.2 Model Process for supervised models

0) Exploratory data analysis (evaluating simple summary measures or identifying predictors that have strong correlations with the outcome / how the predictors will be represented)

1) Pre-processing the predictor data

2) Estimating model parameters

3) Selecting predictors for the model

4) Evaluating model performance

5) Fine tuning class prediction rules (via ROC curves, etc.)

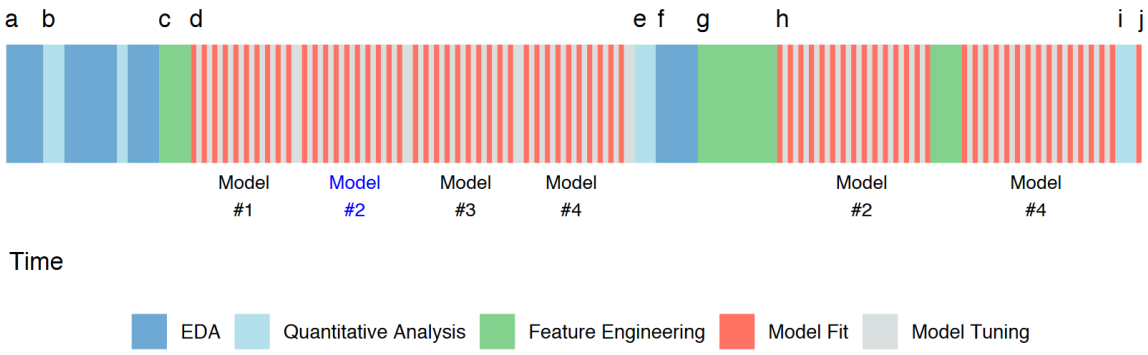


Figure 1.4: A schematic for the typical modeling process.

Figure 2.1: Modeling Process: four distinct models are being evaluated. When modeling data, there is almost never a single model fit or feature set that will immediately solve the problem. The process is more likely to be a campaign of trial and error to achieve the best results. The effect of feature sets can be much larger than the effect of different models. The interplay between models and features is complex and somewhat unpredictable. With the right set of predictors, is it common that many different types of models can achieve the same level of performance.

6) optimization routines (e.g., Nelder–Mead simplex method = *direct methods*) can later be use to search the optimal key value (e.g., determine possible mixtures with improved compressive strength)

7) EDA can be conducted on the model results (e.g., residual analysis)

2.3 Unsupervised models

2.3.1 Clustering models

Groups similar data points together based on distance

k-Means

https://www.youtube.com/watch?v=4b5d3muPQmA

Hierarchical Clustering

Hierarchical clustering focuses on the similarities between the individual instances and how similarities link them together. Tells you pairwise what two things are most similar.

https://www.youtube.com/watch?v=7xHsRkOdVwo

2.3.2 Autoencoders

Neural networks that can denoise or smooth the predictor values and focus on capturing important features in the data.

Use case

Anomaly Detection: The decoder struggles to capture anomalous patterns, and the reconstruction error acts as a score to detect anomalies. Identifies unusual patterns that differ from the majority of the data. Assumes that anomalies are:

- **Rare** : the minority class that occurs rarely in the data
- **Different** : have feature values that are very different from normal observations

Image processing dimension reduction information retrieval

3 Pre-processing

3.1 Transform the predictor/independent variables

Important to transform independent variables that are skewn or containt outliers for models sensitive to them See Table

3.1.1 Centering and scaling

Advantages: Improve the numerical stability to minimize potential numerical errors
Disadvantage: Loss of interpretability

3.1.2 Resolve Skewness

Skewed data: Ratio of the highest value to the lowest value is greater than 20 have significant skewness. (e.g., if the lowest income is $1,000 and the highest income is $20,000 or more, the dataset exhibits significant skewness due to the large ratio between the highest and lowest values.)

- Left Skew: Mean < Median Mode (value that appears most frequently)
- Right Skew: Mean > Median Mode

Techniques:

- log
- square root
- inverse
- Box and Cox (can only be applied to data that is strictly positive)

3.1.3 Resolve Outliers

Techniques:

- Spatial sign

> ⚠ Warning
>
> - it is important to **center and scale** the predictor data prior to using this transformation
> - Avoid removing predictor variables after applying this technique as spatial sign transformation transforms predictors as a group.

3.1.4 Data Reduction and Signal/Feature Extraction

These methods reduce the data by generating a smaller set of predictors that seek to capture a majority of the information in the original variables.

3.1.4.1 PCA (*unsupervised technique*)

The number of components to retain is choosen by creating a scree plot (Fig 1)

For most data sets, the first few PCs will summarize a majority of the variability, and the plot will show a steep descent; variation will then taper off for the remaining components. Generally, the component number prior to the tapering off of variation is the maximal component that is retained. In an automated model building process, the optimal number of components can be determined by cross-validation (see Resampling Techniques).

Advantages: The primary advantage is that it creates components that are uncorrelated. The smaller dataset may be easier to deal with or to process. Moreover, the smaller dataset may better reveal the information.

Disadvantage:

- Loss of interpretability.
- Unsupervised technique which means that PCA it does not consider the modeling objective or response variable when summarizing variability. PCA can generate components that summarize characteristics of the data that are irrelevant to the underlying structure of the data and also to the ultimate modeling objective. **If the predictive relationship between the predictors and response is not connected to the predictors' variability, then the derived PCs will not provide a suitable relationship with the response.** In this case, a supervised technique, like PLS, will derive components while simultaneously considering the corresponding response.
- Data should be linearly related

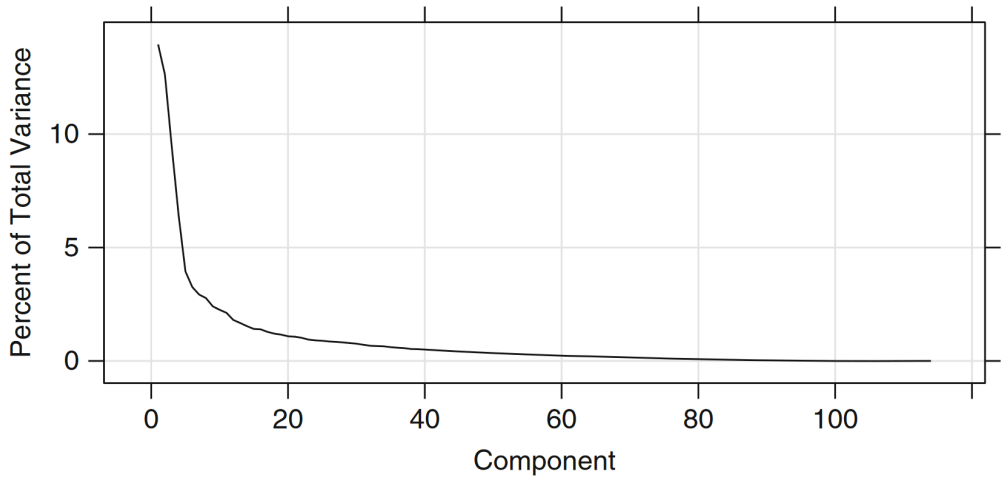


Fig. 3.6: A "scree plot" where the percentage of the total variance explained by each component is shown

Figure 3.1: **Figure 1**: The variation tapers off at component 5. Using this rule of thumb, four PCs would be retained

> ⚠ Warning
>
> First transform skewed predictors and then center and scale the predictors prior to performing PCA. Centering and scaling enables PCA to find the underlying relationships in the data without being influenced by the original measurement scales.

> Pro tips
>
> - If PCA has captured a sufficient amount of information in the data. **Visually examining the principal components** is a critical step for assessing data quality and gaining intuition for the problem. To do this, the first few principal components can be plotted against each other and the plot symbols can be colored by relevant characteristics, such as the class labels.
>
> – Check for blatant **outliers** that may prompt a closer examination of the individual data points
>
> – Check for **clusters** of samples (for classification problems; Try other models that could better accommodate the data to have a final conclusion)
>
> – **Checks loadings to characterize which predictors are associated with each component** (Loadings close to zero indicate that the predictor variable

did not contribute much to that component; Fig 2)

- **Check for multicollinearity** (substantial correlation between multiple predictors): For example, if the first principal component accounts for a large percentage of the variance, this implies that there is at least one group of predictors that represent the same information. For example, Fig 1 indicates that the first 3–4 components have relative contributions to the total variance. This would indicate that there are at least 3–4 significant relationships between the predictors. Colinearity can increase the model variance

- If the percentages of variation explained are not large (e.g., less than 48 %) for the first three components, it is **important not to over-interpret** the resulting image.

3.1.4.2 PLS (*supervised technique*)

Derive components while simultaneously considering the corresponding response.

3.1.5 Dealing with Missing Values

Before proceeding: It is important to understand why the values are missing to check for *informative missing* using visualization such as heatmap or cooccurrence plot (for smaller data) or by plotting the first two scores from a PCA model of the missing data indicator matrix (for larger data sets). Informative missingness can induce significant bias in the model.

Examples: - people are more compelled to rate products when they have strong opinions (good or bad) - The tested drug was extremely ineffective or had significant side effects. The patient may be likely to miss doctor visits or to drop out of the study.

- **1) Remove predictors** from models if the percentage of missing data is substantial
- **2)** For large data sets, **removal of samples** based on missing values is not a problem
- **3** If we **do not remove** the missing data:
  - Use tree-based techniques, which account for missing data
  - code missing data as "missing"
  - impute missing data using information in the training set predictors (for small to moderate amounts of missingness). This amounts to a predictive model within a predictive model. If we are using resampling to select tuning parameter values or to estimate performance, the imputation should be incorporated within the resampling. This will increase the computational time for building models, but it will also provide honest estimates of model performance.

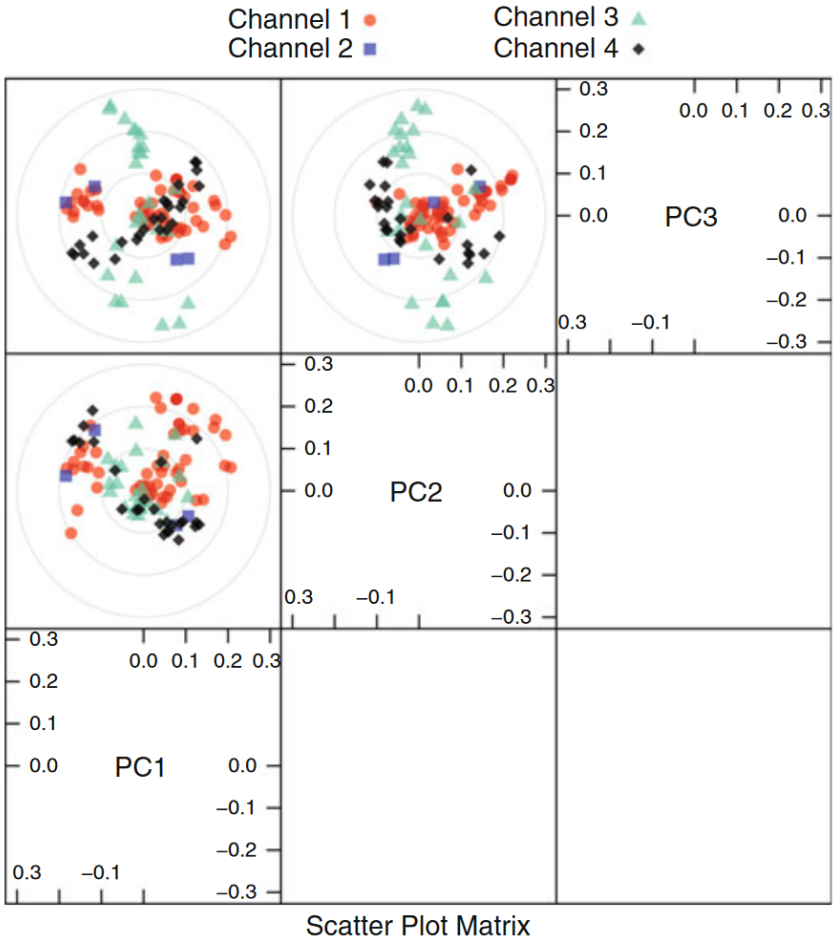


Figure 3.2: **Figure 2**: loadings for the first three components in the cell segmentation data. For the first principal component, the loadings for the first channel are on the extremes. This indicates that channel 1 have the largest effect on the first principal component and by extension the predictor values. Also note that the majority of the loadings for the third channel are closer to zero for the first component. Conversely, the third principal component is mostly associated with the third channel while the first channel plays a minor role here.

    * K-nearest neighbor model: *Advantages*: The imputed data are confined to be within the range of the training set values. *Disadvantages*: The entire training set is required every time a missing value needs to be imputed **and** The number of neighbors is a tuning parameter
    * Linear regression model: between a predictor with few missing points strongly associated with the predictor with missing data (correlation / visualizations / PCA.
    * Bagged trees

NOTES Censored data Missing data: The exact value is missing but something is known about its value. For example: If a customer has not yet returned a movie to blockbuster, we do not know the actual time span, only that it is as least as long as the current duration.

- For inference models: the censoring is usually taken into account in a formal manner by making assumptions about the censoring mechanism
- For predictive models, it is more common to treat these data as simple missing data or use the censored value as the observed value.

3.1.6 Removing Predictors

Advantages: Does not compromise the performance and stability of the model. Decreased computational time and complexity. Lead to a more parsimonious and interpretable model

- Remove near-zero predictors (e.g., predictor variable where the percentage of unique values is low $< 10\%$ = unique values/total values and The ratio of the frequency of the most prevalent value to the frequency of the second most prevalent value is large > 20).
- Remove problematic predictors with degenerate distributions as some models can be crippled by them
- Remove highly correlated predictors as both mesure the same underlaying information. **For linear regressions use VIF for other sensitive models ensure that all pairwise correlations are below 0.75 threshold**
- Feature Selection (see Feature Selection section)

3.1.7 Adding Predictors

See Feature_engineering section

4 Model Tuning

Identify settings for the model's parameters that yield the best and most realistic predictive performance

4.1 Data splitting method

4.1.1 Nonrandom approaches to splitting the data

Use case

- In **time-series data**, it's crucial to maintain the temporal order. Randomly shuffling the data can break the temporal dependencies and result in unrealistic evaluations of a model's performance.
- If your data has **spatial dependencies**, such as in geospatial analysis, maintaining the spatial structure during data split becomes essential. Random splitting might scatter spatially related data points across different sets, leading to poor model generalization.
- In situations where **data privacy** is a concern, you might want to ensure that certain sensitive records or individuals are not included in the training set. A nonrandom approach allows for more control over the inclusion or exclusion of specific data points.
- **Specific Use Cases**:
  - In spam filtering; it is more important for the model to catch the new spamming techniques rather than prior spamming schemes.
  - If a model was being used to predict patient outcomes, the model may be created using certain patient sets (e.g., from the same clinical site or disease stage), and then tested on a different sample population to understand how well the model generalizes.
  - In chemical modeling for drug discovery, new "chemical space" is constantly being explored. We are most interested in accurate predictions in the chemical space that is currently being investigated rather than the space that was evaluated years prior.

4.1.2 Random sampling methods

4.1.2.1 Simple random sample

The simplest way to split the data randomly into a training and test.

Disadvantage: limited ability to characterize the uncertainty in the results.

4.1.2.2 Simple k-Fold Cross-Validation:

The samples are randomly partitioned into k sets of roughly equal size. A model is fit using the all samples except one subset. The held-out samples are predicted by this model and used to estimate performance measures. The first subset is returned to the training set an procedure

repeats with the next subset held out, and so on. Performance estimates, are calculated from each set of held-out samples and then averaged.

NOTES: The choice of k is usually **5** or **10**, but there is no formal rule. **The bias is smaller for k = 10 than k = 5.** As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. But larger values of k are more computationally burdensome.

Advantage: Low computational costs. **Disadvantage**: k-fold cross-validation generally has high variance compared to other methods (only for small training sets). **USE**: **If sample sizes are large (> 10 000)** and we want to choose tuning parameters

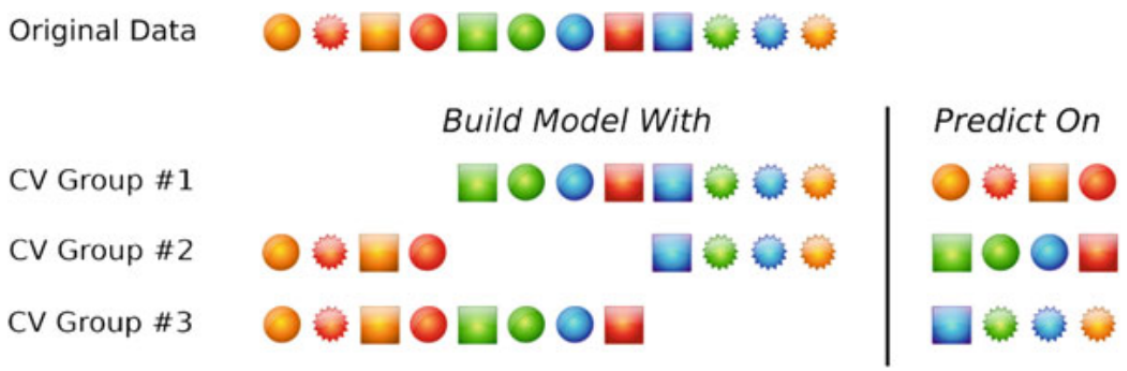


Figure 4.1: 3-Fold Cross-Validation

4.1.2.3 Repeated k-Fold Cross-Validation

Repeated k-fold cross-validation replicates Simple k-Fold Cross-Validation multiple times.

Advantage: Increase the precision of the estimates while still maintaining a small bias. The bias and variance properties are good **Disdvantage**: Larger computational costs if large sample size.
USE: with k = 10; **If the samples size is small (< 1000 obs)** and we want to choose tuning parameters. For example, if 10-fold cross-validation was repeated five times, 50 different held-out sets would be used to estimate model efficacy.

4.1.2.4 leave-one-out Cross-Validation / LOOCV:

Fits as many models as there are samples in the training set, should only be considered when the **number of samples is very small**.

NOTE: leave-one-out and k =10-fold cross-validation yielded similar results, indicating that k = 10 is more attractive from the perspective of computational efficiency.

4.1.2.5 leave-group-out Cross-Validation / Repeated training/test splits / Monte Carlo cross-validation:

Same as k-fold cross-validation except that samples can be represented in multiple held-out subsets. Also, the number of repetitions is usually larger than in k-fold cross-validation. the bias of the resampling technique decreases as the amount of data in the subset approaches the amount in the modeling set. A good rule of thumb is about 75–80 %. Higher proportions are a good idea if the number of repetitions is large.

NOTES: Increase the number of repetition can allow to increase the proportion of data in the train set and decreasing the uncertainty of the performance estimates. To get stable estimates of performance, it is suggested to choose a larger number of repetitions **(say 50–200)**

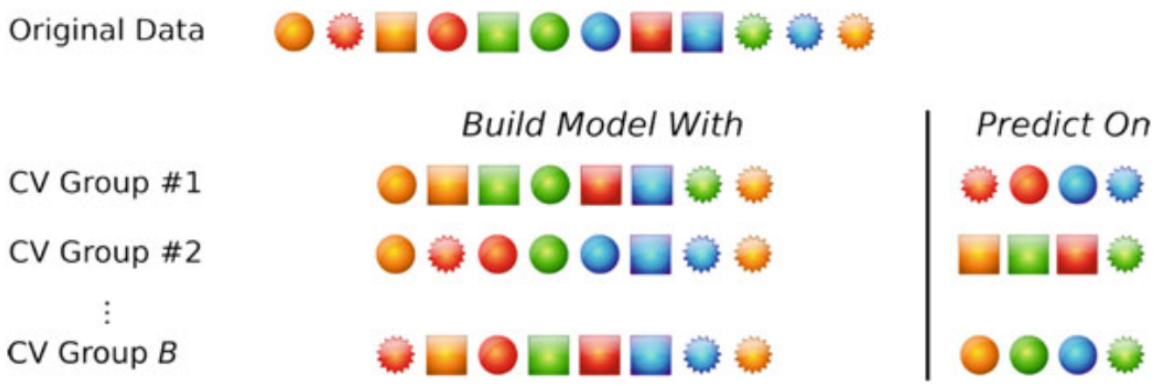


Figure 4.2: leave-group-out Cross-Validation

4.1.2.6 The Bootstrap:

Each train subset is the same size as the original and can contain multiple instances of the same data point (taken with replacement). Samples not selected by the bootstrap ("out-of-bag" samples) are predicted and used to estimate model performance

Advantage: error rates have less uncertainty than k-fold cross-validation. Very low variance. **Disadvantage**: On average, 63.2 % of the data points the bootstrap sample are represented at least once, so this technique has bias. similar to k-fold cross-validation when k 2. If the training set size is small, this bias may be problematic, but will decrease as the training set sample size becomes larger. **USE**: **If the goal is to choose between models** (boosted trees vs support vector machines…), as opposed to getting the best indicator of performance

- The Bootstrap 632 method

  Advantage: The modified bootstrap estimate reduces the bias.

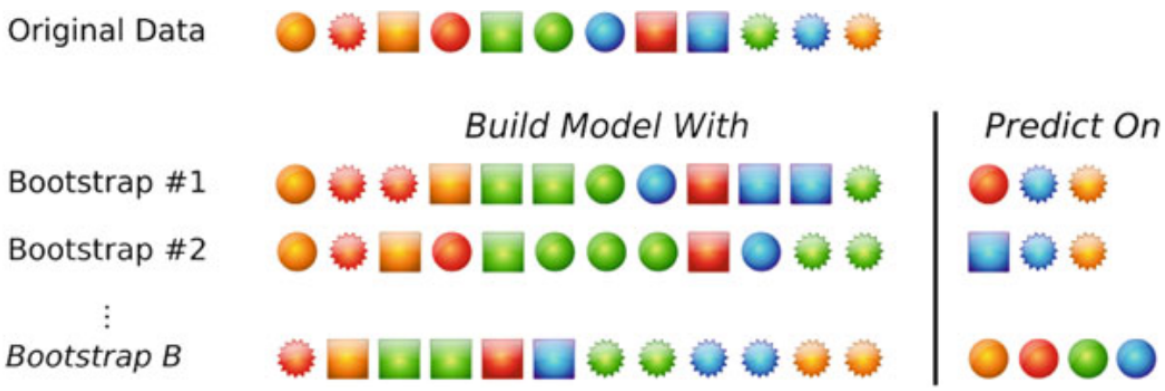


Figure 4.3: Bootstrap

Disadvantage: The estimate is unstable with small samples sizes. This estimate can also result in unduly optimistic results when the model severely over-fits the data, since the apparent error rate will be close to zero.

- The Bootstrap 632+ method **Advantage**: Allows to adjust the bootstrap 632 method estimates

4.1.2.7 Stratified random

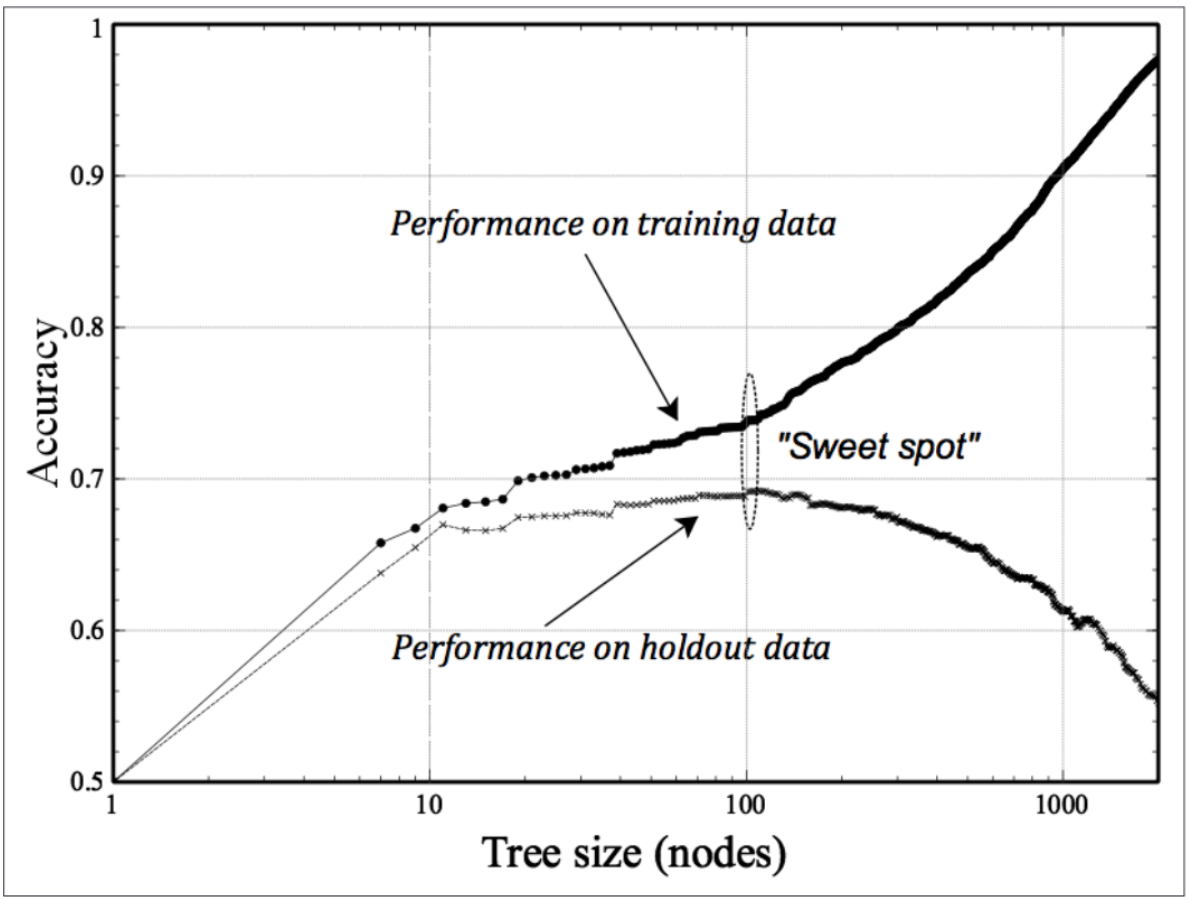
To account for the outcome when splitting the data. Applies random sampling within subgroups (such as the classes or is outcomes are numbers the numeric values are broken into similar groups (e.g., low, medium, and high)).

4.1.2.8 Maximum dissimilarity sampling

The data is split on the basis of the predictor values.

4.2 Choosing the best tuning parameters

- Pick the settings associated with the numerically best performance estimates but **Disadvantage**: lead to models that are overly complicated (see Figure fitting graph)



- Pick simpler models that provide acceptable performance (relative to the numerically optimal settings)

- The "one-standard error" method: pick the simpler model within a single standard error of the numerically best value. In table below we would pick cost value of 2.
- the "percent decrease in performance" method: pick the simplier model that is within a certain tolerance of the numerically best value. (e.g., The percent decrease in performance could be quantified by $(X - O)/O$ where X is the performance value and O is the numerically optimal value. For example, in Figure below the best accuracy value across the profile was 75 %. If a 4 % loss in accuracy was acceptable as a trade-off for a simpler model, accuracy values greater than 71.2 % would be acceptable. For the profile in Figure below, a cost value of 1 would be chosen using this approach.)

Resampled accuracy (%)			
Cost	Mean	Std. error	% Tolerance
0.25	70.0	0.0	−6.67
0.50	71.3	0.2	−4.90
1.00	74.0	0.5	−1.33
2.00	74.5	0.7	−0.63
4.00	74.1	0.7	−1.20
8.00	75.0	0.7	0.00
16.00	74.9	0.8	−0.13
32.00	72.5	0.7	−3.40
64.00	72.0	0.8	−4.07
128.00	72.0	0.8	−4.07

Figure 4.4: Cross-validation accuracy

4.3 Metrics / performance measures

4.3.1 For models predicting a categorical outcome

4.3.1.1 Accuracy based metrics

A good model has generally a metric above 0.7 / 70%

1) **Accuracy**:

- *Higher Value*: Better performance (values form 0 to 1)
- *When to use*: Use when the classes are balanced (e.g., Suppose the rate of this disorder1 in fetuses is approximately 1 in 800 or about one-tenth of one percent. A predictive model can achieve almost perfect accuracy by predicting all samples to be negative for Down syndrome.), and misclassification of different classes has similar consequences.
- *Advantage*: Simple and easy to interpret.
- *Disadvantage*: Can be misleading when classes are imbalanced / make no distinction about the type of errors being made
- *Description*: Accuracy measures the proportion of correct predictions out of all predictions made by the model.

- *Example*: Suppose you have a binary classification problem to identify whether an email is spam or not. If your model has an accuracy of 90%, it means it correctly classified 90% of the emails.
- *Calculation*: (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives) (Number of Correct Predictions) / (Total Number of Predictions)
- *Notes*:
  - When evaluating the accuracy of a model, the baseline accuracy rate to beat would be the percentage which could be achieve by simply predicting all samples to the dominant category (e.g., In the data set, 70 % were rated as having good, accuracy rate to beat would be 70 % which is the no-information rate).
  - **error rate**: (Number of Incorrect Predictions) / (Total Number of Predictions)

1.5) **Kappa**:

- *Higher Value*: Better performance (values from -1 to 1; 0.30 to 0.50 indicate reasonable agreement)
- *When to use*: Rather than calculate the overall accuracy and compare it to the no-information rate, *Kappa* can be used that take into account the class distributions of the training set samples.
- *Advantage*: Takes into account the accuracy that would be generated simply by chance.
- *Disadvantage*: NA
- *Description*: assess the agreement between two raters
- *Example*: 0 means there is no agreement between the observed and predicted classes, while a value of 1 indicates perfect concordance of the model prediction and the observed classes. Negative values indicate that the prediction is in the opposite direction of the truth, but large negative values seldom occur, if ever, when working with predictive models.
- *Calculation*: Kappa = O − E / 1−E: O is the observed accuracy and E is the expected accuracy based on the marginal totals of the confusion matrix.
- *Note*: The Kappa statistic can also be extended to evaluate concordance in problems with more than two classes. When there is a natural ordering to the classes (e.g., "low,""medium," and "high"), an alternate form of the statistic called weighted Kappa can be used to enact more substantial penalties on errors that are further away from the true result. For example, a "low" sample erroneously predicted as "high" would reduce the Kappa statistic more than an error were "low" was predicted to be "medium." See (Agresti 2002)for more details.

2) **Precision**:

- *Higher Value*: Better performance (good = 0.7)

- *When to use*: Use when the cost of false positives is high (e.g., medical diagnosis, fraud detection).
- *Advantage*: Focuses on the relevance of positive predictions.
- *Disadvantage*: Ignores true negatives and may not be suitable for imbalanced datasets.
- *Description*: Precision is the proportion of true positive predictions (correctly predicted positive class) out of all positive predictions made by the model.
- *Example*: In the spam email example, if your model has a precision of 80%, it means that out of all the emails it predicted as spam, 80% of them were actually spam.
- *Calculation*: True Positives / (True Positives + False Positives)

3) **Sensitivity / True positive rate / Recall**:

- *Higher Value*: Better performance
- *When to use*: Use when the cost of false negatives is high (e.g., medical diagnosis, safety-critical applications).
- *Advantage*: Focuses on the completeness of positive predictions (includes true positive and false negatives).
- *Disadvantage*: Ignores true negatives. For many classification problems, sensitivity may be misleading specially under class imbalance. Since a better cutoff may be possible, an analysis of the ROC curve can lead to improvements in these metrics. Consequently, performance metrics that are independent of probability cutoffs are likely to produce more meaningful contrasts between models.
- *Description*: Is the proportion of true positive predictions out of all actual positive instances in the dataset.
- *Example*: % of people *with* heart diseases were correctly identify by the model
- *Calculation*: TP / (TP + FN)
- *Notes*:
  - If the data set includes more events than nonevents, the sensitivity can be estimated with greater precision than the specificity and sensitivity shouls be use to choose between models.
  - When we want to make unconditional evaluations of the data: know for example what are the chances that … (e.g., If PPV = 0.75 this means that out of all the individuals who tested positive for Disease X, 75% of them actually have the disease, while the remaining 25% are false positives) we can use positive predicted value (PPV = Sensitivity × Prevalence / (Sensitivity × Prevalence) + ((1 − Specificity) × (1 − Prevalence))) IMPOTANT: Predictive values are not often used to characterize the model. There are several reasons why, most of which are related to prevalence. First, prevalence is hard to quantify.

4) **Specificity / True Negative Rate**:

- *Higher Value*: Better performance

- *When to use*: Use when you want to focus on correctly identifying negative cases and the cost of false positives is high.
- *Advantage*: Focuses on the negative class and avoids false positives. Can be misleading specially under class imbalance.
- *Disadvantage*: Ignores true positives.
- *Description*: Specificity measures the proportion of true negative predictions out of all actual negative samples.
- *Example*: % of people *without* heart diseases were correctly identify by the model
- *Calculation*: True Negatives / (True Negatives + False Positives);
- *Notes*:
  - When we want to make unconditional evaluations of the data: know for example what are the chances that … (If NPV = 0.966, this means that out of all the individuals who tested negative for Disease X, 96.6% of them truly do not have the disease, while the remaining 3.4% are false negatives (individuals who have the disease but were incorrectly identified as negative).) we can use negative predicted value (NPV = Specificity × (1 − Prevalence) / (Prevalence × (1 − Sensitivity)) + (Specificity × (1 − Prevalence))). IMPORTANT: *idem*
  - *False-positive rate* : one minus the specificity

4.3) **Youden's J Index**

- *Higher Value*: Better performance
- *When to use*: Use when you want a measure that reflects the false-positive and false-negative rates and summarize the magnitude of both types of errors.
- *Advantage*: Focuses on the negative class and avoids false positives.
- *Disadvantage*: Ignores true positives and may not be suitable for imbalanced datasets.
- *Description*: measures the proportions of correctly predicted samples for both the event and nonevent groups.
- *Example*: % of people *without* heart diseases were correctly identify by the model
- *Calculation*: J = Sensitivity + Specificity − 1

5) **F1 Score**:

- *Higher Value*: Better performance
- *When to use*: Use when classes are imbalanced and there is a trade-off between precision and recall.
- *Advantage*: Incorporates both precision and recall into a single metric.
- *Disadvantage*: Ignores true negatives, which can be important in some cases. May not be ideal for highly imbalanced datasets.
- *Description*: F1 score is the harmonic mean of precision and recall, providing a balance between the two.
- *Example*: Let's say your model has an F1 Score of 0.75, it means there is a balanced trade-off between correctly identifying positive samples and minimizing false positives.

- *Calculation*: 2 * (Precision * Recall) / (Precision + Recall)

4.3.1.2 Class probabilities

Class probabilities potentially offer more information about model predictions than the simple class value. This

4.7) **ROC**:

- *Higher Value*: Better performance (A perfect model that completely separates the two classes would have 100 % sensitivity and specificity / A completely ineffective model would result in an ROC curve that closely follows the 45 ◦ diagonal line and would have an area under the ROC curve of approximately 0.50.) Area under the curve can be used as a quantitative measure of performance
- *When to use*: Helpful tool for choosing a threshold that appropriately maximizes the trade-off between sensitivity and specificity (how many false positive we are willing to accept) (e.g., Lowering the threshold (aka 50%) can we improve the sensitivity to capture more true positives). Make a quantitative assessment of the model
- *Advantage*: the curve is insensitive to disparities in the class proportions. Metrics that is independent of probability cutoffs
- *Disadvantage*: disadvantage of using the area under the curve to evaluate models is that it obscures information (i.e., the curves cross both AUC can be the same).
- *Description*: AUC-ROC measures the area under the receiver operating characteristic curve, which plots the **true positive rate (recall) against the false positive rate (1 - Specificity)** at various classification thresholds (10%, 20%… 50% = commonly used).
- *Example*: An AUC-ROC score of 0.85 indicates that the model has an 85% chance of correctly ranking a randomly chosen positive instance higher than a randomly chosen negative instance.
- *Calculation*: AUC-ROC can be calculated using various methods, such as the trapezoidal rule or Mann-Whitney U statistic.
- *Notes*:
  - We can use the partial area under the ROC curve as a technique to summarize these curves that focuses on specific parts of the curve.
  - ROC technique can be extended to fit three or more classes problems

6) **Recall - Precision Curve**:

- *When to use*: When data is imbalanced as it focuses on the correct prediction of the minority class
- *Description*: Plot precision in X axis and recall in Y axis

7) **Lift Charts**:

- *Higher Value*: Better performance (Figure)
- *When to use*: To assess the ability of a model to detect events in a data set with two classes and allow us to choose a quasithreshold for a model.
- *Advantage*: Easy connect the model to the buisness value: Using the lift plot, the expected profit can be calculated for each point on the curve to determine if the lift is sufficient to beat the baseline profit
- *Disadvantage*: Bad for comparing different models
- *Description*: The lift chart plots the cumulative gain/lift against the cumulative percentage of samples that have been screened
- *Example*: Figure shows the best and worse case lift curves for a data set with a 50 % event rate. The non-informative model has a curve that is close to the 45 ◦ reference line, meaning that the model has no benefit for ranking samples. The other curve is indicative of a model that can perfectly separate two classes. At the 50 % point on the x-axis, all of the events have been captured by the model.
- *Calculation*: *NA*
- *Notes*:
  - The section of the curve associated with the highest-ranked samples should have an enriched true-positive rate and is likely to be the most important part of the curve.

NOTES:

- It is important to test whether the estimated class probabilities are reflective of the true underlying probability of the sample (well-calibrated Probabilities) using a *calibration plot*. This plot shows some measure of the observed probability of an event versus the predicted class probability. One approach for creating this visualization is to score a collection of samples with known outcomes (preferably a test set) using a classification model. The next step is to bin the data into groups based on their class probabilities. For example, a set of bins might be [0, 10 %], (10 %, 20 %], ..., (90 %, 100 %]. For each bin, determine the observed event rate. Suppose that 50 samples fell into the bin for class probabilities less than 10 % and there was a single event. The midpoint of the bin is 5 % and the observed event rate would be 2 %. The calibration plot would display the midpoint of the bin on the x-axis and the observed event rate on the y-axis. If the points fall along a 45 ◦ line, the model has produced well-calibrated probabilities.

- If there are three or more classes, a heat map of the class probabilities can help gauge the confidence in the predictions.

- An approach to improving classification performance is to create an *equivocal* or *indeterminate zone* where the class is not formally predicted when the confidence is not high. (e.g., For a two-class problem that is nearly balanced in the response, the equivocal zone could be defined as 0.50 ± z.Ifz were 0.10, then samples with prediction probabilities between 0.40 and 0.60 would be called "equivocal." In this case, model performance would be calculated excluding the samples in the indeterminate zone.)

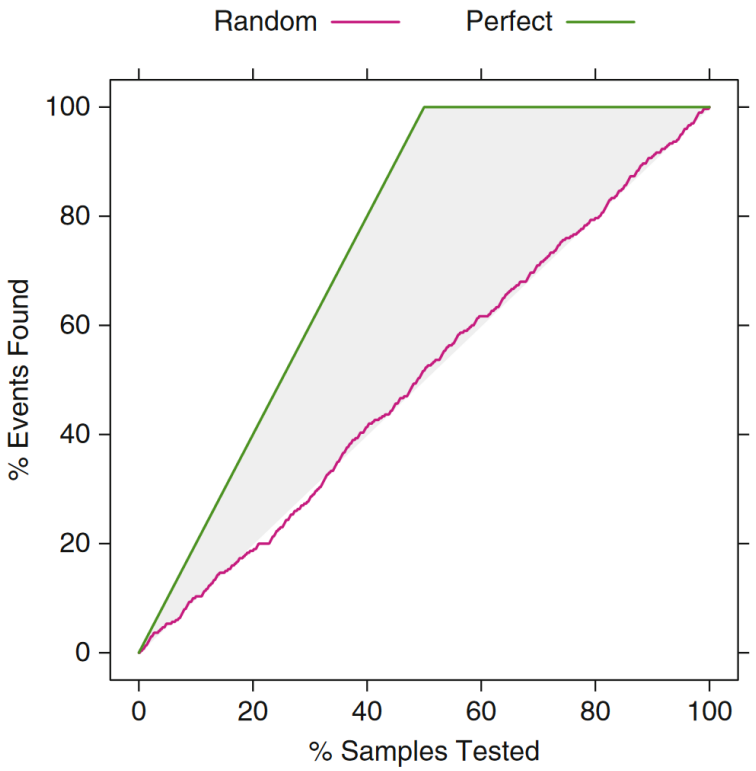


Fig. 11.7: An example lift plot with two models: one that perfectly separates two classes and another that is completely non-informative

Figure 4.5: Lift Charts

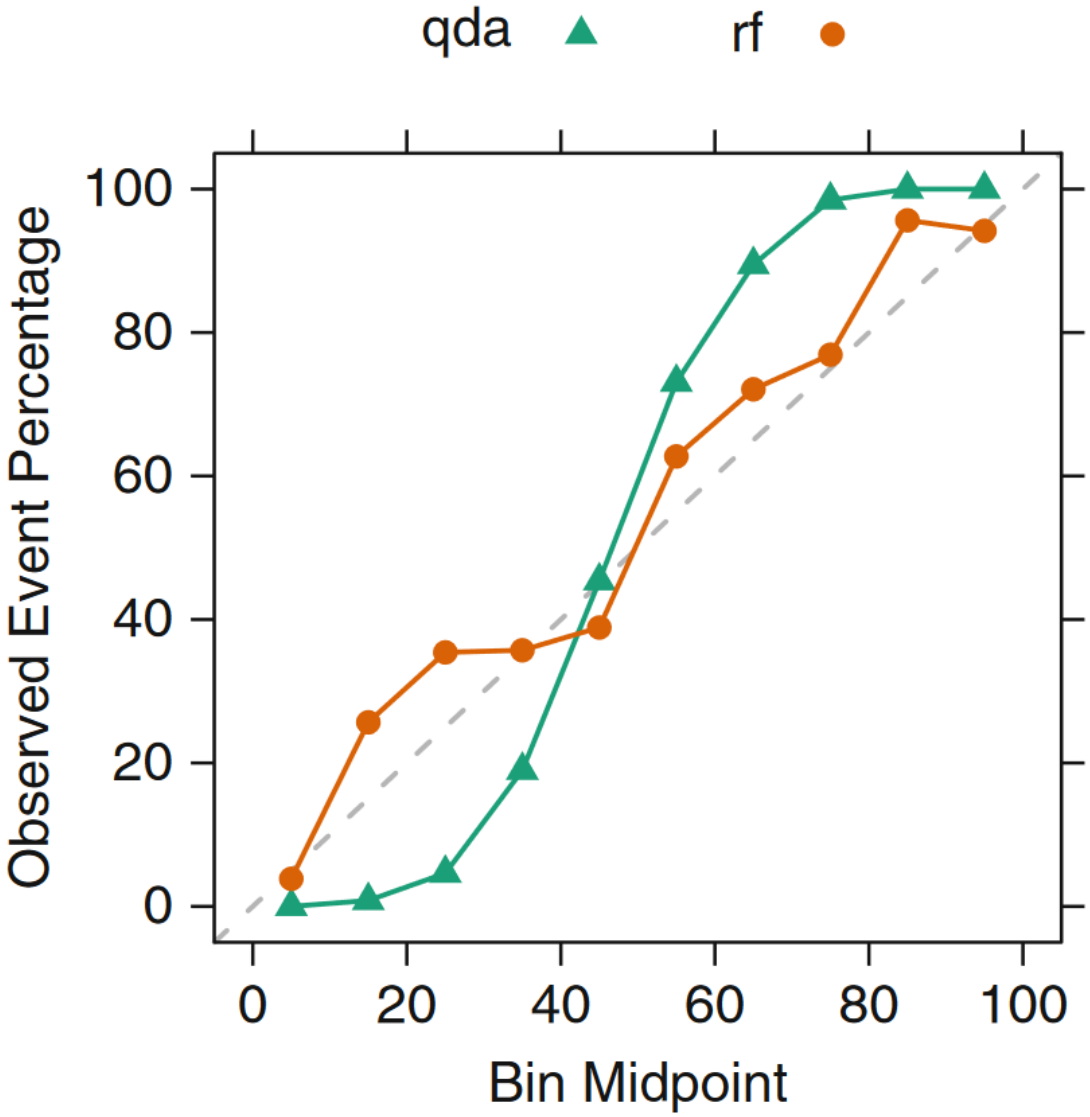


Figure 4.6: A calibration plot of the test set probabilities for random forest and quadratic discriminant analysis models

4.3.1.3 Non-Accuracy-Based Criteria

When accuracy is not the primary goal for the predictive model and we want to quantify the consequences of correct and incorrect predictions (i.e., the benefits and costs)

Examples:

- Predict investment opportunities that maximize return
- Improve customer satisfaction by market segmentation
- Lower inventory costs by improving product demand forecasts
- Reduce costs associated with fraudulent transactions: For example, in fraud detection, a model might be used to quantify the likelihood that a transaction is fraudulent. Suppose that fraud is the event of interest. Any model predictions of fraud (correct or not) have an associated cost for a more in-depth review of the case. For true positives, there is also a quantifiable benefit to catching bad transactions. Likewise, a false negative results in a loss of income.

1) **profit** = *Cost/Benefit* * TP − *Cost/Benefit* FP − *Cost/Benefit* FN
2) **NEC** (normalized expected cost / classification_cost_penalized) = PCF × (1 − TP)+(1− PCF) × FP (between 0 and 1)

4.3.2 For models predicting a numeric outcome

1) **RMSE**:

- *Higher Value*: Worse performance
- *When to use*: Commonly used to measure the average magnitude of prediction errors.
- *Advantage*: Penalizes larger errors more heavily, sensitive to outliers and unit is the same as the target variable, making it more interpretable.
- *Disadvantage*: Sensitive to outliers.
- *Description*: The average distance between the observed values and the model predictions.
- *Example*: Continuing with the house price prediction example, an RMSE of 100 means that, on average, the predicted house prices deviate from the actual prices by $100.
- *Calculation*: Squared root of the (sum the residuals (the observed values minus the model predictions) and dividing by the number of samples) For example, if we have actual values [5, 10, 15] and predicted values [6, 12, 10], the MSE would be calculated as ((1^2) + (2^2) + (5^2)) / 3 = 10.

2) **MAE**:

- *Higher Value*: Worse performance
- *When to use*: Suitable when you want to avoid the influence of outliers.

- *Advantage*: Not sensitive to outliers as it uses the absolute error.
- *Disadvantage*: It does not penalize large errors as heavily as RMSE.
- *Description*: The Mean Absolute Error measures the average of the absolute differences between predicted and actual values.
- *Example*: For the house price prediction, an MAE of $50 means that, on average, the predicted house prices deviate from the actual prices by $50.
- *Calculation*: For example, with the same actual and predicted values, the MAE would be calculated as (|1| + |2| + |5|) / 3 = 2.67.

3) **R2**:

- *Higher Value*: Better performance
- *When to use*: Commonly used to measure the average magnitude of prediction errors. It is a measure of correlation, not accuracy. **Bad** for predicting a **number** (accuracy) but **good** for determining the **rank** correlation between the observed and predicted values (e.g., pharmaceutical scientists want to find the compounds predicted to be the most biologically active).
- *Advantage*: Represents the proportion of the variance in the dependent variable that is predictable from the independent variables.
- *Disadvantage*: It can be misleading when used with complex models or when the number of predictors is large. It is dependent on the variation in the outcome (e.g., If the range of the houses in the test set was large, say from $60K to $2M, the variance of the sale price would also be very large. One might view a model with a 90 % R2 positively, but the RMSE may be in the tens of thousands of dollars—poor predictive accuracy for anyone selling a moderately priced property)
- *Description*: The proportion of the information in the data that is explained by the model
- *Example*: An R-squared of 0.75 means that 75% of the variance in the house prices can be explained by the model, and the remaining 25% is due to random variation.
- *Calculation*: Correlation coefficient between the observed and predicted values
- *Note*: By plotting R2 we can see where the model is overpredict (e.g., low values) and underpredict (e.g., higher values). If this happend depending on the context, this systematic bias in the predictions may be acceptable if the model otherwise works well.

4) **R2 adjusted**:

- *Higher Value*: Better performance
- *When to use*: Helpful when you have multiple predictors and want to account for model complexity.
- *Advantage*: It adjusts R-squared for the number of predictors, giving a more reliable assessment of model performance when compared to R-squared.
- *Disadvantage*: It might not penalize overfitting adequately with large numbers of predictors.

- *Description*: R-squared adjusted is similar to R-squared but takes into account the number of predictors in the model. It penalizes models with more predictors if they don't contribute significantly to the variance explained.
- *Example*:
- *Calculation*:

5) **MAPE**:

- *Higher Value*: Worse performance
- *When to use*: Useful when you want to evaluate the performance in percentage terms.
- *Advantage*: Represents the percentage difference between predicted and actual values, making it interpretable and independent of the scale of the data.
- *Disadvantage*: It can be problematic when actual values are close to zero.
- *Description*: The Mean Absolute Percentage Error calculates the mean percentage difference between predicted and actual values.
- *Example*: An MAPE of 10 means that, on average, the predicted house prices deviate from the actual prices by 10%.
- *Calculation*: For example, if we have actual values [100, 50, 75] and predicted values [90, 40, 70], the MAPE would be calculated as (|(100-90)/100| + |(50-40)/50| + |(75-70)/75|) / 3 0.16.

6) **EV**:

- *Higher Value*: Better performance (good value > 0.6)
- *When to use*: Useful to understand how well the model explains the variance in the target variable.
- *Advantage*: Measures the proportion of variance explained by the model, similar to R-squared.
- *Disadvantage*: It might not penalize the model adequately for underfitting or overfitting.
- *Description*: The Explained Variance Score quantifies the proportion of variance in the target variable that is explained by the model. It ranges from 0 to 1, with 1 indicating a perfect fit. For example, an EV of 0.85 means that 85% of the variance is explained by the model.
- *Example*: An EV of 0.9 means that the model explains 90% of the variance in the house prices, leaving 10% unexplained by the model.
- *Calculation*:

7) **MSLE**:

- *Higher Value*: Worse performance
- *When to use*: Suitable when you want to focus on the ratio of errors rather than their absolute differences. It can be useful when predictions are on a large scale.
- *Advantage*: Penalizes underestimation and overestimation proportionally and is less sensitive to large errors.

- *Disadvantage*: The logarithmic transformation can be problematic for data containing zero or negative values.
- *Description*: The Mean Squared Logarithmic Error calculates the mean of the squared logarithmic differences between predicted and actual values.
- *Example*: For the house price prediction, an MSLE of 0.1 means that, on average, the predicted house prices deviate from the actual prices by 10% when measured on a logarithmic scale.
- *Calculation*: For instance, if we have actual values [100, 50, 75] and predicted values [110, 40, 80], the MSLE would be calculated as ((log(110) - log(100))^2 + (log(40) - log(50))^2 + (log(80) - log(75))^2) / 3 0.015.

4.3.3 Calculate expected value

See Data science for Business chapter

4.3.4 Visualizing Model Performance

It is often revealing to visualize model behavior under a broad range of conditions.

- **Profit curves**:
  - **Use Case**: Basic profit graph can be useful to compare models of interest under a range of conditions. These graphs may be easy to comprehend for stakeholders who are not data scientists, since they reduce model performance to their basic "bottom line" cost or profit.
  - **Disadvantages**: The disadvantage of a profit graph is that it requires that operating conditions be known and specified exactly. With many real-world problems, the operating conditions are imprecise or change over time, and the data scientist must contend with uncertainty. In such cases other graphs may be more useful.

- **Cumulative response curves**:
  - **Use Case**: To understand what percentage of the population has to be targeted. When costs and benefits cannot be specified with confidence, but the class mix will likely not change, a cumulative response or lift graph is useful. Both show the relative advantages of classifiers, independent of the value (monetary or otherwise) of the advantages.

- **Lift curves**:
  - **Use Case**: Same as *Cumulative response curves*.

- **ROC/AUC curves**:

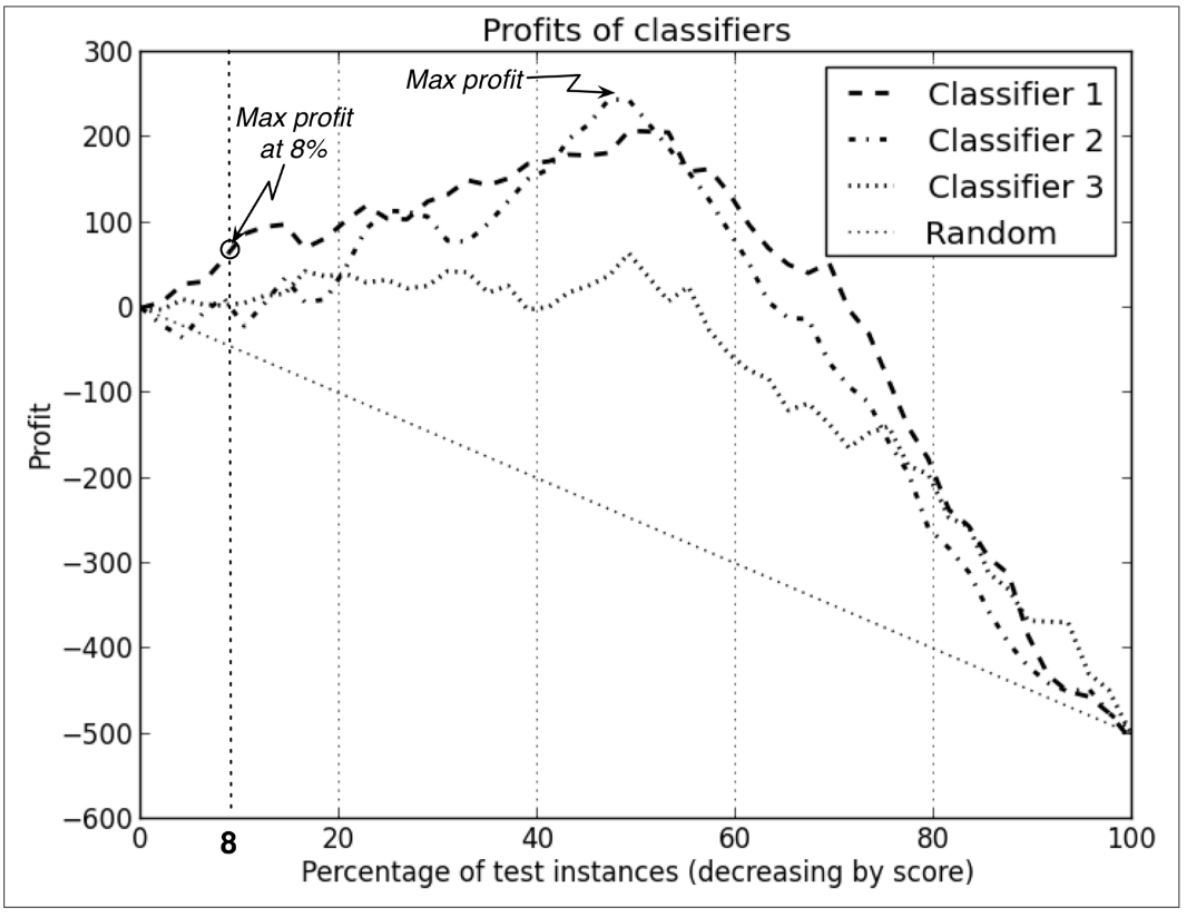


Figure 8-2. Profit curves of three classifiers. Each curve shows the expected cumulative profit for that classifier as progressively larger proportions of the consumer base are targeted.

Figure 4.7: Profit curves

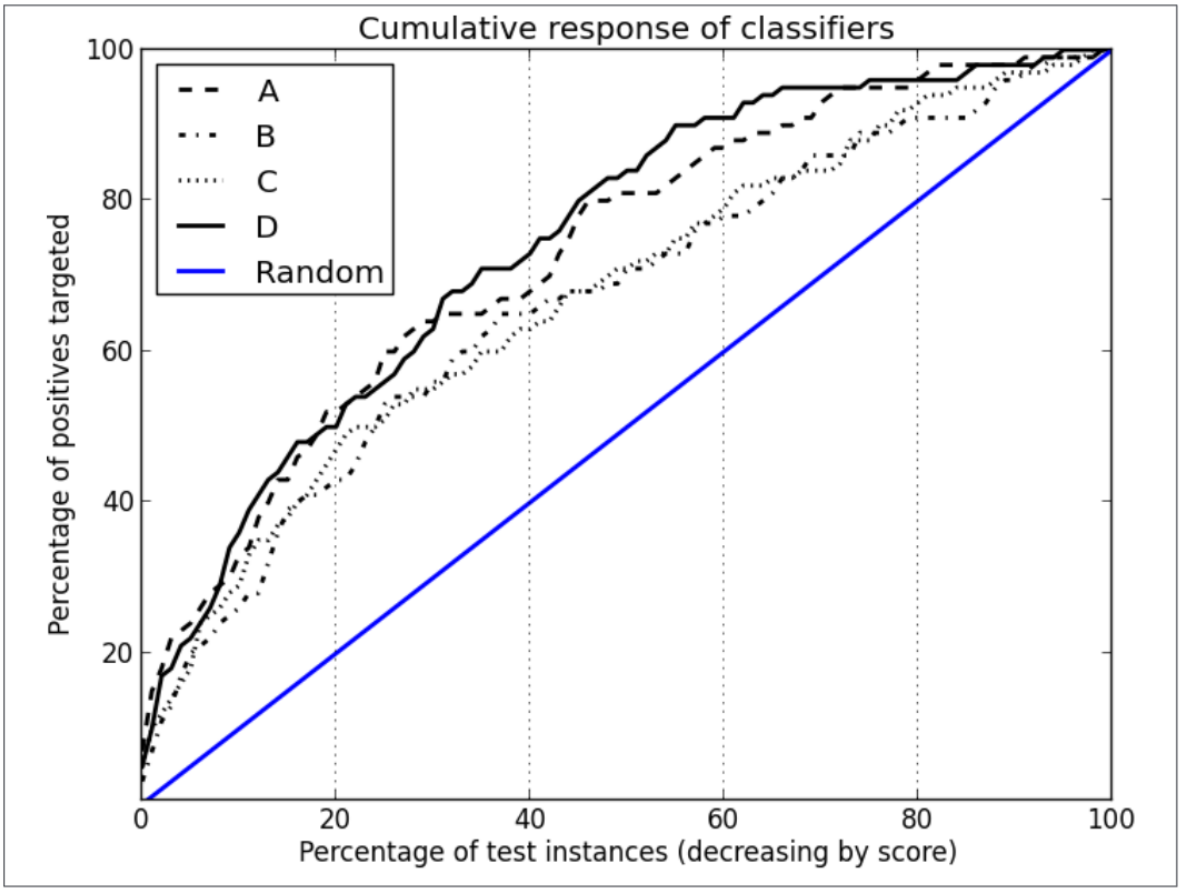


Figure 8-6. Four example classifiers (A–D) and their cumulative response curves.

Figure 4.8: Cumulative response curves.

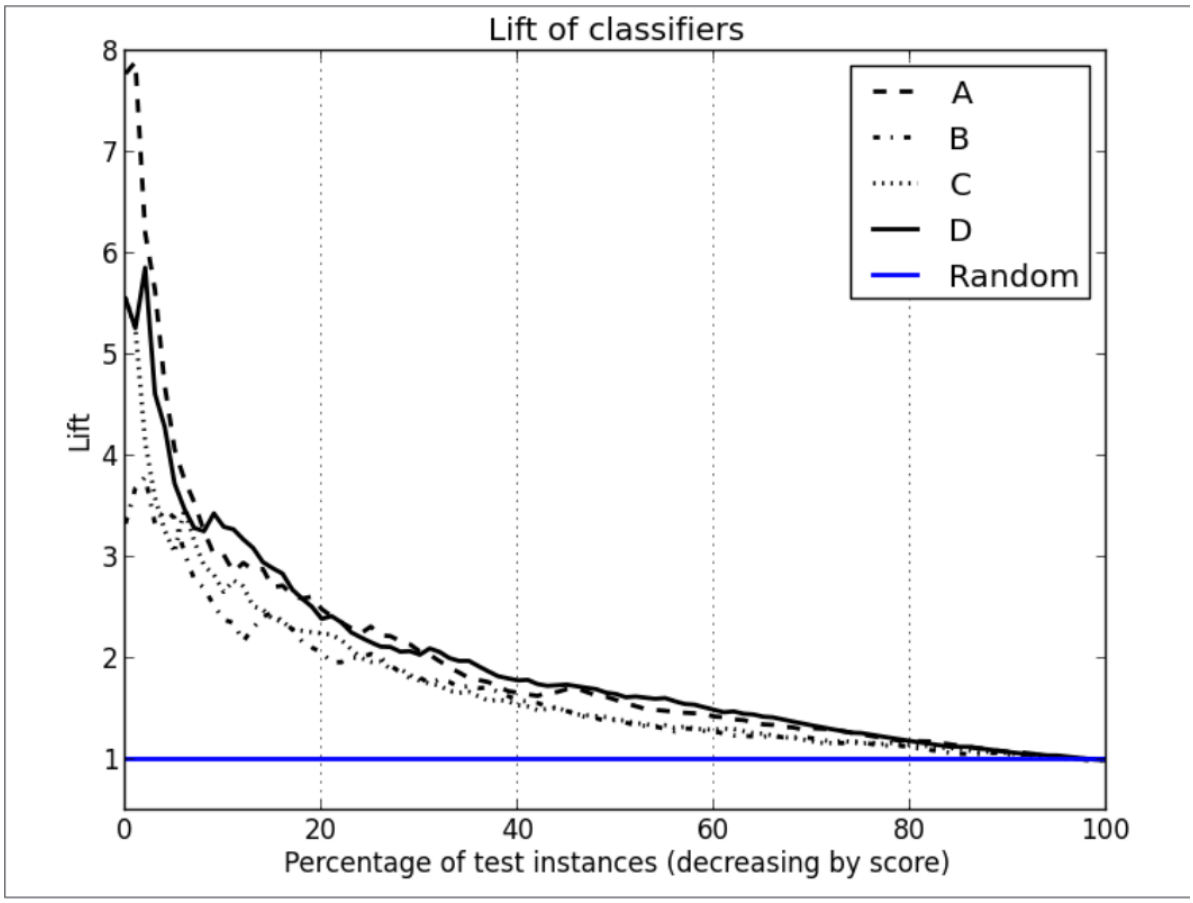


Figure 8-7. The four classifiers (A–D) of Figure 8-6 and their lift curves.

Figure 4.9: Lift curves

– **Use Case**: Finally, ROC curves are a valuable visualization tool for the data scientist. Though they take some practice to interpret readily, they separate out performance from operating conditions. In doing so they convey the fundamental trade-offs that each model is making.

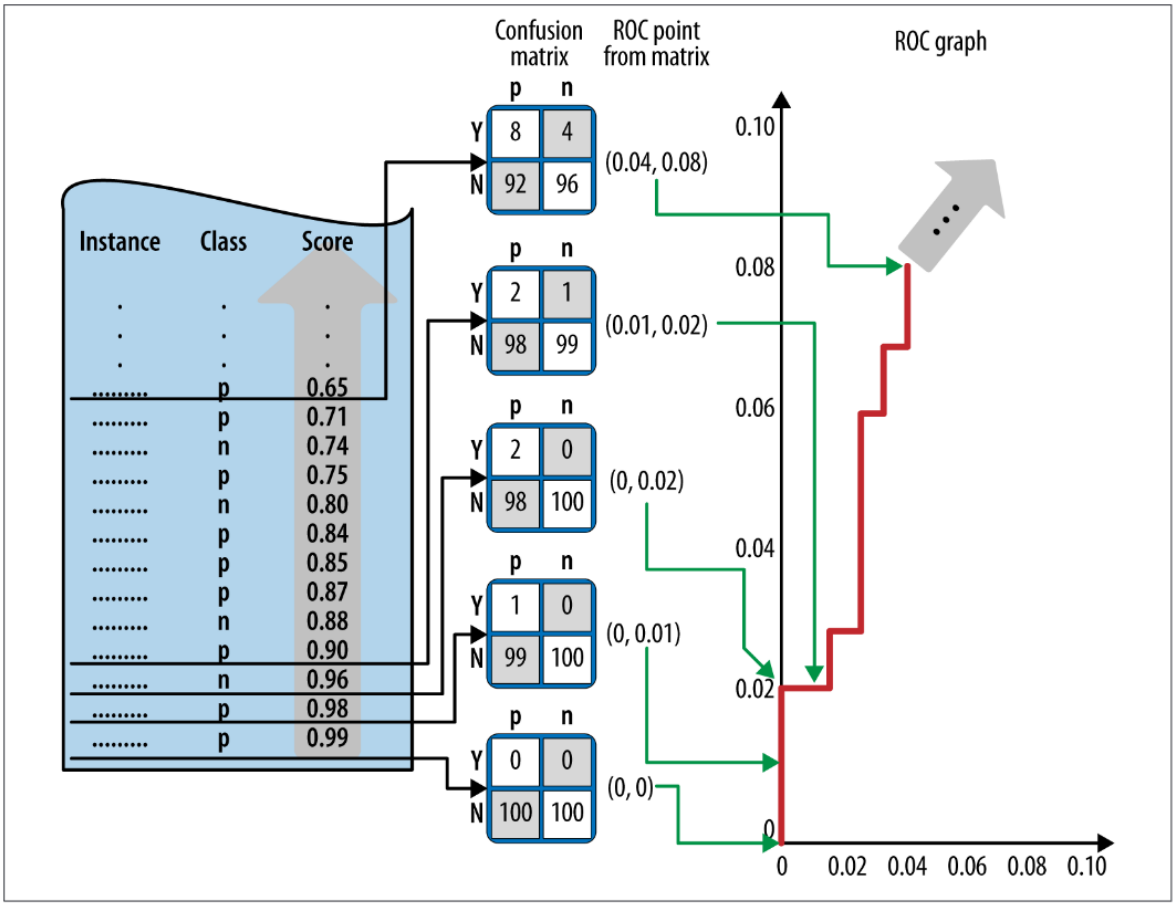


Figure 8-5. An illustration of how a ROC "curve" (really, a stepwise graph) is constructed from a test set. The example set, at left, consists of 100 positives and 100 negatives. The model assigns a score to each instance and the instances are ordered decreasing from bottom to top. To construct the curve, start at the bottom with an initial confusion matrix where everything is classified as **N***. Moving upward, every instance moves a count of 1 from the* **N** *row to the* **Y** *row, resulting in a new confusion matrix. Each confusion matrix maps to a (fp rate, tp rate) pair in ROC space.*

Figure 4.10: ROC

5 Choosing between models

Once the settings for the tuning parameters have been determined for each model, the question remains: how do we choose between multiple models?

0) Comparison against a random model or a simpler model (naive classifier that always chooses the majority class of the training dataset; for regression problems we have a directly analogous baseline: predict the average value over the population (usually the mean or median); decision stumps often produce quite good baseline performance) establishes that there is some information to be extracted from the data.
1) Start with several models that are the least interpretable and most flexible, such as boosted trees or support vector machines. Across many problem domains, these models have a high likelihood of producing the empirically optimum results (i.e., most accurate).
2) Investigate simpler models that are less opaque (e.g., not complete black boxes), such as multivariate adaptive regression splines (MARS), partial least squares, generalized additive models, or näıve Bayes models.
3) Consider using the simplest model that reasonably approximates the performance of the more complex methods.
4) Visualizing Model Performance to visualize model behavior under a broad range of conditions

5.1 Tools

5.1.1 Paired t-test

To evaluate if the differences between models are statistically significant. It is also recommended to plot confidence intervals that were derived using the bootstrap (Figure) for two reasons.

- The interval quantifies the variation in the model but is also reflective of the data. For example, smaller test sets or noise (or mislabeling) in the response can lead to wider intervals.
- Facilitate trade-offs between models. If the confidence intervals for two models significantly overlap, this is an indication of (statistical) equivalence between the two and might provide a reason to favor the less complex or more interpretable model.

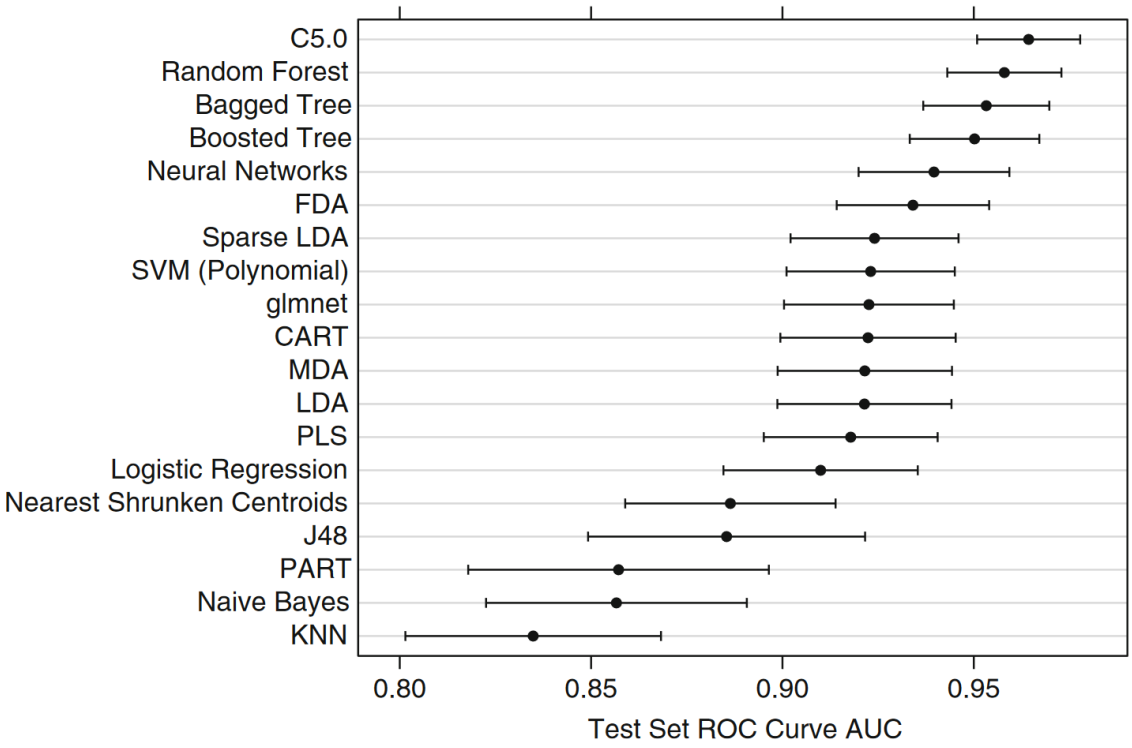


Figure 5.1: A plot of the test set ROC curve AUCs and their associated 95 % confidence intervals

5.1.2 Learning curves

- The learning curve may show that generalization performance has leveled off so investing in more training data is probably not worthwhile; instead, one should accept the current performance or look for another way to improve the model, such as by devising better features

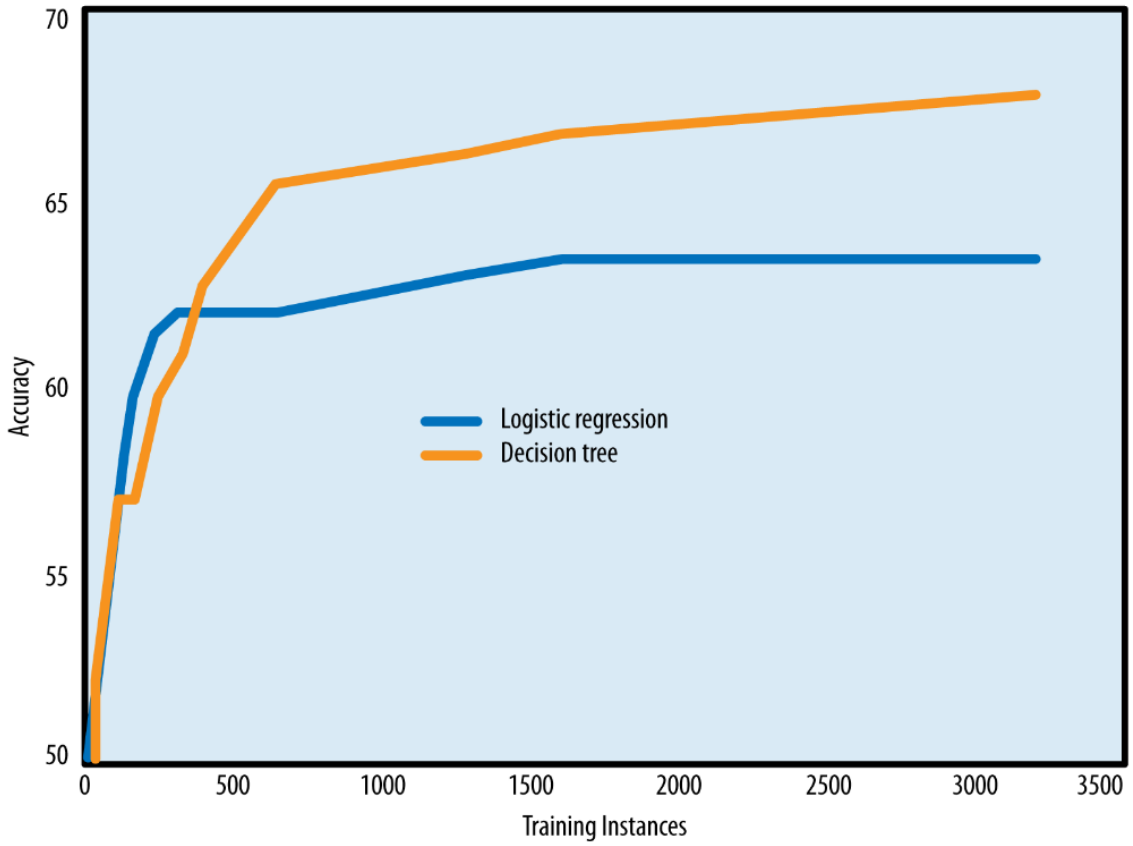


Figure 5.2: Learning curves for tree induction and logistic regression for the churn problem. As the training size grows (x axis), generalization performance (y axis) improves. Importantly, the improvement rates are different for the two induction technique, and change over time. Logistic regression has less flexibility, which allows it to overfit less with small data, but keeps it from modeling the full complexity of the data. Tree induction is much more flexible, leading it to overfit more with small data, but to model more complex regularities with larger training sets.

6 Linear Regression and Its Cousins

They all seek to find estimates of the parameters to minimize the sum-of-squared errors

6.1 Ordinary linear regression

Finds parameter estimates that have minimum bias using the NIPALS approach **Advantages**:

- highly interpretable
- enables us to compute standard errors of the coefficients allowing to assess the statistical significance of each predictor

Assumptions:

- Linear relationship and independent observations
- Homoscedasticity error terms have constant variance
- Errors are uncorrelated and normally distributed
- Low multicollinearity

6.2 Penalized models

For classification and regressions: Finds parameter estimates that have lower variance. We introduce bias to reduce variance and avoid overfitting

USE: When sample size are small

Advantages: reduce variance and increases prediction on the long term

6.2.1 Ridge regression

https://www.youtube.com/watch?v=Q81RR3yKn30

Advantages: better at reducing variance in models that contain usefull variables

6.2.2 Lasso regression

https://www.youtube.com/watch?v=NGf0voTMlcs

Advantages: - better at reducing variance in models that contain useless variables - simplify model

6.2.3 Elastic net

https://www.youtube.com/watch?v=1dKRdX9bfIo

USE: When you don't know if you have useless variables **Advantages**: Best of both ridge and lasso

6.3 Logistic Regression

For classification only https://www.youtube.com/watch?v=yIYKR4sgzI8&list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe

Assumptions: - Linear relationship between X and log-odds of Y - Independent observations - Low multicollinearity

6.4 Linear Discriminant Analysis

For classification only

It is like PCA but it focuses on maximizing the separability among the known categories.

Create an axis (create two axis for three of more categories) that maximizes the distance between the means for the two categories while minimizing the scatter.

As in PCA the first axis created (LDA1) by LDA accounts for the most variation between the categories. LDA2 does the second better job, LDA 3 the third best job etc etc...

NOTE: We can see which variables correlates the most with each LDA

https://www.youtube.com/watch?v=azXCzI57Yfc

6.5 Partial least squares (PLS)

For regression and classification:

supervised dimension reduction procedure while PCR (PCA + linear regression) is **unsupervised**

USE: when there are correlated predictors and a linear regression-type solution is desired instead of PCA then linear regression (AKA PCR; If, the variability in the predictor space is not related to the variability of the response, then PCR can have difficulty identifying a predictive relationship when one might actually exist).

Efficiently for data sets of small-to moderate size (e.g., $< 2{,}500$ samples and < 30 predictors)

Pre-prossesing:

- centered and scaled predictors.
- Remove predictors with small PLS regression coefficients and small VIP (<1)
- *To include nonlinear relationships add squared or cubic predictors*
- *To include nonlinear relationships splits each predictor into two or more bins for those predictors that are thought to have a nonlinear relationship with the response. Cut points for the bins are selected by the user and are based on either prior knowledge or characteristics of the data. The original predictors that were binned are then excluded from the data set that includes the binned versions of the predictors. (GIFI approach)*

Tuning: Cross-validation was used to determine the optimal number of PLS components to retain that minimize RMSE *# of tuning parameter*: PLS has one tuning parameter: the number of components to retain

For classification: **NOTES**: Produce continuous predictions that do not follow the definition of a probability-the predicted values are not necessarily between 0 and 1 and do not sum to 1. Therefore, a transformation (e.g., *softmax transformation*) must be used to coerce the predictions into "probability-like" values so that they can be interpreted and used for classification.

6.5.1 Algorithmic Variations of PLS

- SIMPLS approach

USE: for data large data sets (e.g., $> 2{,}500$ samples and > 30 predictors)

- Rannar et al. (1994) kernel

USE: when there are more predictors than samples.

6.5.2 Partial Least Squares Discriminant Analysis

For classification only

6.5.3 Nearest Shrunken Centroids

For classification only

7 Nonlinear Models

In these models, the exact form of the nonlinearity does not need to be known explicitly or specified prior to model training.

7.1 Tree-based models

Advantages:

- Handles high-dimensional data well.
- Robust to outliers and noise.
- Provides feature importance measures.
- Requires minimal data preprocessing and is relatively easy to implement.

Disadvantages:

- Can be computationally expensive for large datasets.
- May not perform well on imbalanced datasets.
- Lacks interpretability compared to single decision trees.

USE:

- When you have a large dataset with a high number of features.
- When interpretability is not a top priority.
- When you want to build a model that is robust to overfitting.

Tuning parameters:

- The number of trees in the forest. Higher values generally improve performance but increase computational time (Common values are between 50 to 500).
- The maximum depth of each decision tree: Controls the tree's complexity and potential overfitting (Common values are between 5 to 50).
- The minimum number of samples required to split an internal node: Higher values prevent overfitting Common values are between 2 to 20).
- The number of features to consider when looking for the best split: Common values are 'sqrt' (square root of total features) or 'log2'.

For regression:

https://www.youtube.com/watch?v=g9c66TUylZ4

For classification:

https://www.youtube.com/watch?v=_L39rN6gz7Y

7.1.1 Random forests

https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

7.1.2 Boosting trees

Sequentially fits many simple models that account for the previous model's errors. As opposed to bagging, boosting trains on all the data and combines models using the learning rate .

7.1.2.1 AdaBoost

Advantages:

- Can achieve high accuracy by combining multiple weak learners.
- Can handle both classification and regression problems.
- Less susceptible to overfitting compared to individual decision trees.
- Can be combined with any base estimator that accepts sample weights.

Disadvantages:

- Sensitive to noisy data and outliers.
- Can be computationally expensive as it requires sequentially training multiple learners.
- May not perform well on highly imbalanced datasets.

USE:

- When you have a moderately sized dataset and you want to improve the accuracy of weak learners.
- When you want to create a powerful ensemble with different weak learners.

Tuning parameters:

- n_estimators: The number of boosting stages (weak learners) to be run. Common values are between 50 to 500.
- learning_rate: The contribution of each weak learner to the final combination. Common values are between 0.01 to 1.0.

- base_estimator: The base estimator used for boosting. Common choices are decision trees with max_depth set or linear models.

https://www.youtube.com/watch?v=LsK-xG1cLYA

7.1.2.2 XgBoost

Advantages:

- Highly efficient and scalable, making it suitable for large datasets.
- Can handle missing data and supports regularization to prevent overfitting.
- Provides built-in cross-validation, early stopping, and feature importance.
- Often performs well even with default hyperparameters.

Disadvantages:

- Can be sensitive to hyperparameter tuning.
- Requires more careful tuning and validation compared to Random Forest and AdaBoost.
- The interpretation of feature importance may not be as straightforward as in Random Forest.

USE:

- When you have a large dataset and computational efficiency is crucial.
- When you need better performance compared to other algorithms on a wide range of problems.
- When you can invest time in tuning hyperparameters.

Tuning parameters:

- n_estimators: The number of boosting rounds. Common values are between 50 to 500.
- learning_rate: The step size shrinkage used to prevent overfitting. Common values are between 0.01 to 0.3.
- max_depth: The maximum depth of each tree. Common values are between 3 to 10.
- subsample: The fraction of samples used for training each tree. Common values are between 0.5 to 1.0.
- colsample_bytree: The fraction of features used for training each tree. Common values are between 0.5 to 1.0.

For regression: https://www.youtube.com/watch?v=3CC4N4z3GJc https://www.youtube.com/watch?v=OtD8wVaFm6E

For classification: https://www.youtube.com/watch?v=jxuNLH5dXCs https://www.youtube.com/watch?v=8b1JEDvenQU

7.1.2.3 C5.0

Tuning parameters:

- maximum number of leaves: (generally between 8 and 32)
- learning rate: Value between 0 and 1 (generally 0.1)

7.2 Multivariate Adaptive Regression Splines (MARS)

MARS uses surrogate features (usually a function of only one or two predictors (second degree) at a time which are broken into two groups and models linear relationships between the predictor and the outcome in each group) instead of the original predictors (like pls and neural networks).

NOTES - GCV statistic is use to determine the contribution of each feature to the model.

Tuning parameters:

- the degree of the features that are added to the model (number of interaction; e.g., 0-4)
- the number of retained terms

Advantages:

- the model automatically conducts feature selection
- interpretability is high
- requires very little pre-processing of the data (Correlated predictors do not drastically affect model performance, but they can complicate model interpretation.).

Disadvantages: For MARS models that can include two or more terms at a time, we have observed occasional instabilities in the model predictions where a few sample predictions are wildly inaccurate (perhaps an order of magnitude off of the true value). This problem has not been observed with *additive MARS* models (models with degree of 1).

USE: When there is a clear indication that the relationship between the dependent variable and independent variables is non-linear and interpretability is impportante

7.3 Support Vector Machines (SVM)

USE: When we seek to minimize the effect of outliers

https://www.youtube.com/watch?v=efR1C6CvhmE

NOTE: This principle also apply for regression. However in this case the svm will search for hyperplane that holds the maximum of the observation within the margin (tolerance level)

Tuning parameters: - Kernel: SVM can use different kernel functions to transform the input data into a higher-dimensional space, where it becomes easier to find a separating hyperplane. Common kernel functions include: - Linear Kernel (If regression line is truly linear, the linear kernel function will be a better choice) - Polynomial Kernel (In general, quadratic models have smaller error rates than the linear models) (tuning parameters: degree and scale factor (coef0) and c) - Radial Basis Function (RBF) Kernel (radial basis function has been shown to be very effective overall and easier to tune than polynomial one less tuning parameter) (tuning parameters: (sigma) that controls the scale and c) - hyperbolic tangent

- Threshold (epsilon) (called margin in tidymodels?) (If the threshold is set to a relatively large value, then the outliers are the only points that define the regression line) (e.g., = 0.01; the cost parameter provides more flexibility for tuning the model. So it is suggested to fix a value for and tune over the other kernel parameters)
- C parameter (Cost; e.g., values between 0.25 and 2048):
  - *For classification*: It controls the trade-off between maximizing the margin and minimizing the classification error. A smaller C value creates a wider margin but may allow some misclassifications, while a larger C value creates a narrower margin but may result in fewer misclassifications on the training set.
  - *For regression*: The cost parameter is the main tool for adjusting the complexity of the model. When the cost is large, the model becomes very flexible since the effect of errors is amplified. When the cost is small, the model will "stiffen" and become less likely to over-fit (but more likely to underfit)

Pre-processing: Center and scale the predictors prior to building an SVM model since the predictors enter into the model as the sum of cross products, differences in the predictor scales can affect the model.

7.4 K-Nearest Neighbors (KNN)

For classification: https://www.youtube.com/watch?v=HVXime0nQeI

For regression: https://www.youtube.com/watch?v=3lp5CmSwrHI

Tuning parameters: K number of neighbors.

Advantages: The KNN method can have poor predictive performance when local predictor structure is not relevant to the response.

Pre-processing: - Remove irrelevant, noise-laden predictors is a key pre-processing step for KNN, since these can cause similar samples to be driven away from each other in the predictor space

NOTE: to enhance KNN predictability weight the neighbors' contribution to the prediction of a new sample based on their distance to the new sample.

7.5 Neural Networks

Neural Networks uses surrogate features instead of the original predictors (like pls and MARS)

For classification: **NOTES**: Produce continuous predictions that do not follow the definition of a probability-the predicted values are not necessarily between 0 and 1 and do not sum to 1. Therefore, a transformation (e.g., *softmax transformation*) must be used to coerce the predictions into "probability-like" values so that they can be interpreted and used for classification.

For regression: *To be continued*

7.6 Nonlinear Discriminant Analysis

For classification only:

7.7 Flexible Discriminant Analysis

For classification only:

7.8 Naive Bayes

Naive Bayes is included in nearly every data mining toolkit and serves as a common baseline classifier against which more sophisticated methods can be compared

Advantages:

For classification only:

- It is very simple and efficient in terms of storage space and computation time. Training consists only of storing counts of classes and feature occurrences as each example is seen.

- Naive Bayes is that it is naturally an "incremental learner." An incremental learner is an induction technique that can update its model one training example at a time. It does not need to reprocess all past training examples when new training data become available. Incremental learning is especially advantageous in applications where training labels are revealed in the course of the application, and we would like the model to reflect this new information as quickly as possible.

> ⚠ Warning
>
> Naive Bayes assume independence between variables although violating this rule don't hurt classification performance it becomes a problem if we're going to be using the probability estimates themselves (e.g., ranking if email 1 has a bigger likelihood of been spam than email 2).

https://www.youtube.com/watch?v=O2L2Uv9pdDA

7.8.1 Multinomial Naive Bayes

Gaussian Naive Bayes https://www.youtube.com/watch?v=H3EjCKtlVog

8 Clustering

Advantages:

- Pattern Discovery: Clustering helps uncover hidden patterns or structures within data. It can reveal insights and relationships that might not be apparent when examining individual data points.
- Data Reduction: Clustering can reduce the dimensionality of data by grouping similar data points together. This simplification can make it easier to visualize and understand large datasets.
- Anomaly Detection: Clustering can help identify outliers or anomalies in data. Data points that do not belong to any cluster may indicate unusual or unexpected behavior.
- Segmentation: Clustering is used extensively in marketing and customer segmentation. It helps businesses target specific customer groups with tailored marketing strategies.
- Recommendation Systems: Clustering can be used to group users or items with similar preferences, making it valuable for recommendation systems. For example, suggesting products or content to users with similar tastes.
- Unsupervised Learning: Clustering is a form of unsupervised learning, which means it does not require labeled data.

Disadvantages:

- Subjectivity: Results should not be taken as the absolute truth about a data set: not very robust to perturbations to the data; Choosing number of clusters (k) and defining similarity or distance metrics can change drastically the results. *RECOMENDATION: clustering subsets of the data in order to get a sense of the robustness of the clusters obtained.*
- Scalability: Overall computational heavy
- Evaluation Challenges: Evaluating the quality of clusters can be challenging, as there is no definitive metric for assessing clustering results. Different evaluation measures may lead to conflicting interpretations (see Section 8.3).
- Curse of Dimensionality: Clustering can become less effective as the dimensionality of the data increases. High-dimensional spaces may require specialized techniques or preprocessing to achieve meaningful clustering.
- Interpretability: While clustering can reveal patterns, it does not provide direct explanations for why certain data points are grouped together. Additional analysis is often needed for interpretation.

8.1 Algorithms

8.1.1 K-means

We seek to partition the observations hierarchical clustering into a pre-specified number of clusters

Advantages:

- Ease of Interpretation: K-means produces non-overlapping clusters, which can be easier to interpret and analyze compared to the nested structure of hierarchical clustering.
- Efficiency: K-means is computationally efficient, especially when dealing with a large dataset. It can handle large datasets much better than hierarchical clustering.
- Scalability: K-means can work well with high-dimensional data, making it suitable for a wide range of applications, including text mining and image segmentation.

Disadvantages:

- it requires us to pre-specify the number of clusters K
- Assumes Equal Sized Clusters: K-means assumes that clusters are spherical, equally sized, and have similar densities, which may not hold true in real-world datasets.

https://www.youtube.com/watch?v=4b5d3muPQmA

8.1.2 Hierarchical (bottom-up or agglomerative: dendrogram is build starting from the leaves)

We do not know in advance how many clusters we want and the clusters are later choose base on the generated dendrogram.

Advantages:

- Does not requires us to pre-specify the number of clusters K and the results; thus more flexible
- Hierarchy Visualization: Attractive tree-based representation of the observations.
- Flexibility: You can cut the hierarchical tree at different levels to obtain clusters of different sizes and shapes, making it adaptable to various scenarios.
- Robustness: less sensitive to the initial conditions compared to K-means.

Disadvantages:

- Computationally Intensive and may not perform well with high-dimensional data or extremely large datasets due to the computational burden.

https://www.youtube.com/watch?v=7xHsRkOdVwo

8.2 Preprocessing and Tuning

We try several different choices, and look for the one with the most useful or interpretable solution.

Preprocessing: - standardized observations or features - Outlier handling

Tuning: - For K-means: - n clusters

- For hierarchical clustering:
  - dissimilarity measure
  - type of linkage
  - cutting points in the dendrogram

8.3 Validating the Clusters Obtained

Assign a p-value to a cluster in order to assess whether there is more evidence for the cluster than one would expect due to chance. (see Hastie, Tibshirani, and Friedman 2009)

8.4 Deal with outliers

Mixture models (soft version of K-means clustering) are an attractive approach for accommodating the presence of small subset of the observations are quite different from each other and from all other observations (outliers that do not belong to any cluster) (see Hastie, Tibshirani, and Friedman 2009).

8.5 References

data from (James et al. 2021)

9 Remedies for suboptimal data

9.1 Class Imbalance

An imbalance occurs when one or more classes have very low proportions in the training data as compared to the other classes.

- Online advertising: Ad clicked or not (2.4%)
- Pharmaceutical research: Molecules with activity (vwry few) or not
- Insurance claims: Fraud (only 22%) or not fraud
- Spam detection: Spam or not spam
- Selling buisness: Buy (6%) or not buy

9.1.1 The Effect of Class Imbalance

- The models achieve good specificity (since almost every customer is predicted no insurance) but have poor sensitivity (Figure).

Table 16.1: Results for three predictive models using

Model	Accuracy	Kappa	Sensitivity	Spec
Random forest	93.5	0.091	6.78	
FDA (MARS)	93.8	0.024	1.69	
Logistic regression	93.9	0.027	1.69	

- The imbalance also had a severe effect on the predicted class probabilities. (e.g., In the random forest model, for example, 82 % of the customers have a predicted probability of having insurance of 10 % or less. This highly left-skewed predicted probability distribution also occurs for the other two models. This means that the models are not very confident in predicting that most customers have insurance; they tend to assign low probabilities of having insurance to a significant portion of the customers.)
- Imbalance cause that lift charts and ROC curves have similar patterns

9.1.2 Strategies for overcoming class imbalances

9.1.2.1 Hyperparameters selection

- **Model tuning strategy**: tune the model to maximize the accuracy or sensitivity of the minority class(es)

9.1.2.2 Post-processing techniques (use model outputs)

- **Alternate probability Cutoffs** to improve the prediction accuracy of the minority class samples (i.e., post-processing the model predictions to redefine the class predictions). The most straightforward approach is to use the *ROC curve* since it calculates the sensitivity and specificity across a continuum of cutoffs. Using this curve, an appropriate balance between sensitivity and specificity can be determined.
  - Several techniques exist for determining a new cutoff:
    1) First, if there is a particular target that must be met for the sensitivity or specificity, this point can be found on the ROC curve and the corresponding cutoff can be determined.
    2) Another approach is to find the point on the ROC curve that is closest (i.e., the shortest distance) to the perfect model (with 100 % sensitivity and 100 % specificity), which is associated with the upper left corner of the plot. In Figure, a cutoff value of 0.064 would be the closest to the perfect model.
    3) The cutoff associated with the largest value of the Youden index (measures the proportion of correctly predicted samples for both the event and nonevent groups / can be computed for each cutoff that is used to create the ROC curve): show superior performance relative to the default 50 % value. For the random forest ROC curve, the cutoff that maximizes the Youden index (0.021) is similar to the point closest to the optimal model.

NOTE: In our analysis, the alternate cutoff for the model was not derived from the training or test sets. It is important, especially for small samples sizes, to use an independent data (small evaluation set used for developing post-processing techniques ~ 10% training set used to tune model) set to derive the cutoff. If the training set predictions are used, there is likely a large optimistic bias in the class probabilities that will lead to inaccurate assessments of the sensitivity and specificity. If the test set is used, it is no longer an unbiased source to judge model performance.

- **Adjusting Prior Probabilities**: For models that use prior probabilities naiıve Bayes and discriminant analysis classifiers. Unless specified manually, these models typically derive the value of the priors from the training data. Weiss and Provost (2001a) suggest that priors that reflect the natural class imbalance will materially bias predictions to the majority class. **Using more balanced priors or a balanced training set may help**

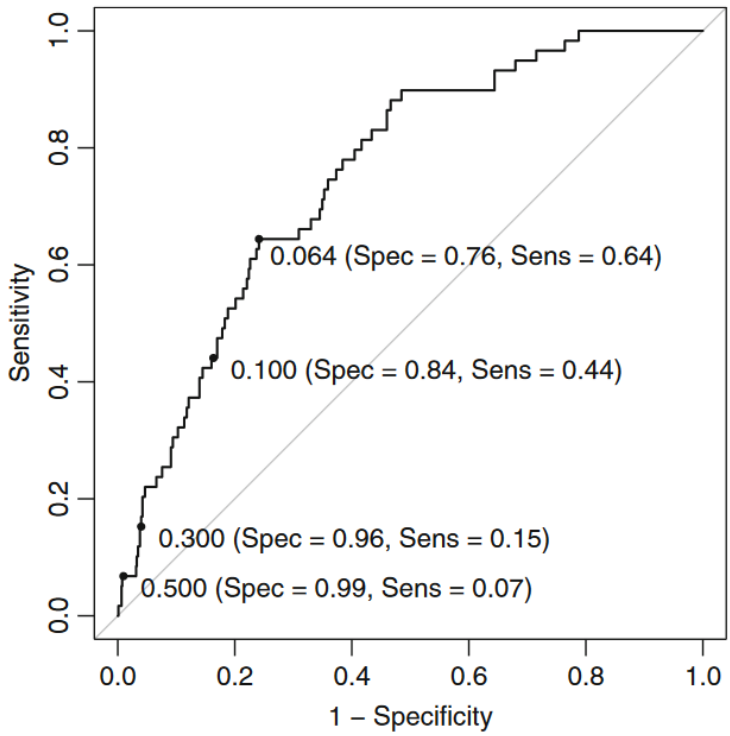


Fig. 16.2: The random forest ROC curve for predicting the classes using the evaluation set. The number on the left represents the probability cutoff, and the numbers in the parentheses are the specificity and sensitivity, respectively. Several possible probability cutoffs are used, including the threshold geometrically closest to the perfect model (0.064)

Figure 9.1: ROC: The predicted sensitivity for the new cutoff of 0.064 is 64.4 %, which is a significant improvement over the value generated by the default cutoff. The consequence of the new cutoff is that the specificity is estimated to drop from 99 % to 75.9 %.

deal with a class imbalance. (e.g., when classes are 6 % and 94 % for the insured and uninsured it is better to use 60 % for the insured and 40 % for the uninsured). This strategy did not change the model (same ROC) but allows for different trade-offs between sensitivity and specificity.

9.1.2.3 Alter training data prior to model training

- **Adjust Sampling Methods**: Non of them is a clear winner, it depends of the case study

  1) *priori sampling approach*: select a training set sample to have roughly equal event rates during the initial data collection. However, the test set should be sampled to be more consistent with the state of nature and should reflect the imbalance so that honest estimates of future performance can be computed.
  2) *post hoc sampling approach*:

  – up-sampling:
    * adding random samples with replacement from minority classes)
  – down-sampling:
    * randomly sample the majority classes so that all classes have approximately the same size
    * bootstrap sample across all cases such that the classes are balanced in the bootstrap set (advatange of bootstrap is that we can obtain the estimate of variation about the down-sampling)
  – SMOTE: adds new/synthetic samples to the minority class and down-sample cases from the majority class via random sampling

  NOTE: Adjusting sampling can bias model performance (e.g., up-sampling: the same sample can be use to predict and tune model)

9.1.2.4 Alter model training process (model parameters are being modified)

- **Unequal Case Weights**: Many of the predictive models for classification (boosting trees which apply different case weights at each iteration) have the ability to use case weights where each individual data point can be given more emphasis in the model training phase. Increase the weights for the samples in the minority classes
- **Cost-Sensitive Training**: Some models (SVM, CART trees, C5.0 trees) can alternatively optimize a cost or loss function that differentially weights specific types of errors of specific classes (it can cause that class probabilities cannot be generated and ROC cannot be use). For example, it may be appropriate to believe that misclassifying true events (false negatives) is X times as costly as incorrectly predicting nonevents (false positives). Correctly classify class A is more importante than correctly classify class B

9.2 Big sample size

An increase in the number of samples can have less positive consequences:

- Computational burdens as the number of samples (and predictors) grows
- There are diminishing returns on adding more of the same data from the same population. Since models stabilize with a sufficiently large number of samples, garnering more samples is less likely to change the model fit.

10 Measuring Predictor Importance

Measurement of predictor relevance is derived by permuting each predictor individually and assessing the loss in performance when the effect of the predictor is negated. Useful for guiding the user to focus more closely on specific predictors via visualizations and other means.

Many predictive models have built-in or intrinsic measurements of predictor importance.

- MARS
- Tree-based models

When models have not built-in measurements of predictor importance we can apply some other techniques

10.1 For Numeric Outcomes

- LOESS
- t-statistic
- ANOVA
- Relief

10.2 For Categorical Outcomes

- area under the ROC curve
- t-statistics
- MIC
- Relief

11 Feature Selection

To get a more interpretable model but not a more accurate model.

11.1 Unsupervised methods

When the outcome is ignored during the elimination of predictors, the technique is unsupervised.

- removing predictors that have high correlations with other predictors
- removing near-zero variance predictors

11.2 Supervised methods

When, the outcome is typically used to quantify the importance of the predictors. Predictors are specifically selected for the purpose of increasing accuracy or to find a subset of predictors to reduce the complexity of the model

11.3 Consequences of Using Non-informative Predictors

The presence of non-informative variables can add **uncertainty/noise** to the predictions and reduce the overall effectiveness of the model (linear regression, partial least squares, neural networks, svm). Regression trees, MARS models and Random forests are not affected or very slightly in the case of random forests

11.4 Approaches for Reducing the Number of Predictors

- **Wrapper** methods evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance. In essence, wrapper methods are search algorithms that treat the predictors as the inputs and utilize model performance as the output to be optimized.

- **Forward, Backward, and Stepwise Selection**: Add, remove predictors or both to find the model that results in the smallest model RMSE/AIC (Used in linear regressions)
- **Correlation-based feature selection**: find the best subset of predictors that have strong correlations with the outcome but weak between-predictor correlations
- **Simulated annealing**: Starting from an initial solution, the method iteratively explores neighboring solutions with the ability to accept worse solutions initially, gradually decreasing this acceptance as the process continues. This balance between exploration and exploitation helps the algorithm escape local optima and search for global optima in the solution space.
- **Genetic Algorithms**: GAs start with a population of potential solutions encoded as "genetic" representations. Through multiple generations, solutions are selected based on their fitness, which measures how well they solve the problem. These selected solutions then undergo crossover (combination of genetic material) and mutation (random changes) to produce a new population. Over successive generations, the algorithm converges towards better solutions as traits from successful solutions propagate and refine in the population.

Advantages: **Disadvantages**:

- many models are evaluated (which may also require parameter tuning) and thus an increase in computation time
- increased risk of over-fitting

- **Filter** methods evaluate the relevance of the predictors outside of the predictive models and subsequently model only the predictors that pass some criterion. For example, for classification problems, each predictor could be individually evaluated to check if there is a plausible relationship between it and the observed classes. Only predictors with important relationships would then be included in a classification model.

  Advantages: more computationally efficient **Disadvantages**:

  - the selection criterion is not directly related to the effectiveness of the model
  - methods evaluate each predictor separately, and, consequently, redundant (i.e., highly-correlated) predictors may be selected and important interactions between variables will not be able to be quantified
  - subjective nature to the procedure. Most scoring methods have no obvious cut point to declare which predictors are important enough to go into the model (In practice, finding an appropriate value for the confidence value may require several evaluations until acceptable performance is achieved.)

- **Embedded** methods are models where the feature selection procedure occurs naturally in the course of the model fitting process. Here an example would be a simple decision tree where variables are selected when the model uses them in a split. If a predictor is

never used in a split, the prediction equation is functionally independent of this variable and it has been selected out.

Warning

When using other search procedures or filters for reducing the number of predictors, there is still a risk. The following situations increase the likelihood of selection bias:

- The data set is small.
- The number of predictors is large (since the probability of a non-informative predictor being falsely declared to be important increases).
- The predictive model is powerful (e.g., black-box models), which is more likely to over-fit the data.
- No independent test set is available

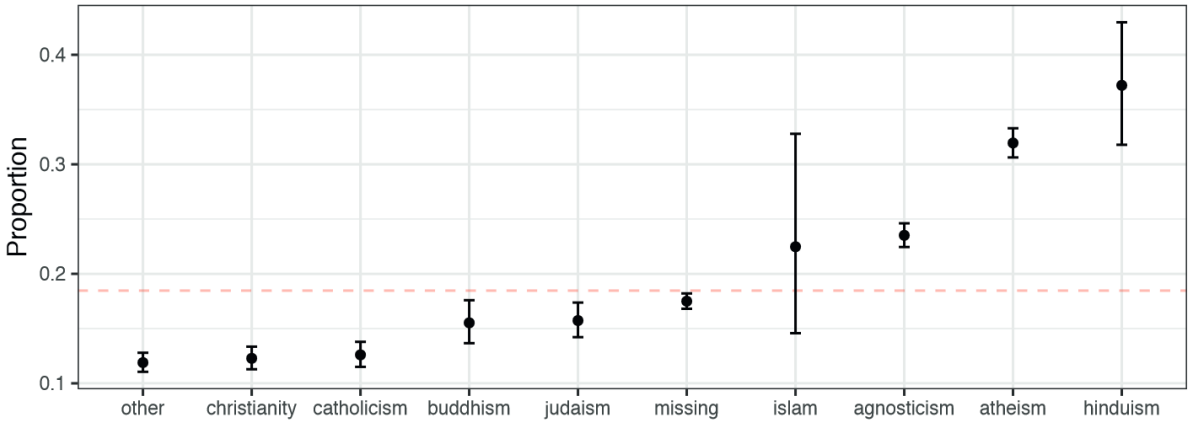
tips

- When the data set is large, it is recommended separate data sets for selecting features, tuning models, and validating the final model (and feature set).
- When training sets are small, proper resampling is critical.
- When the amount of data is not too small (333 obs), it is recommended setting aside a small test set to double check that no gross errors have been committed.

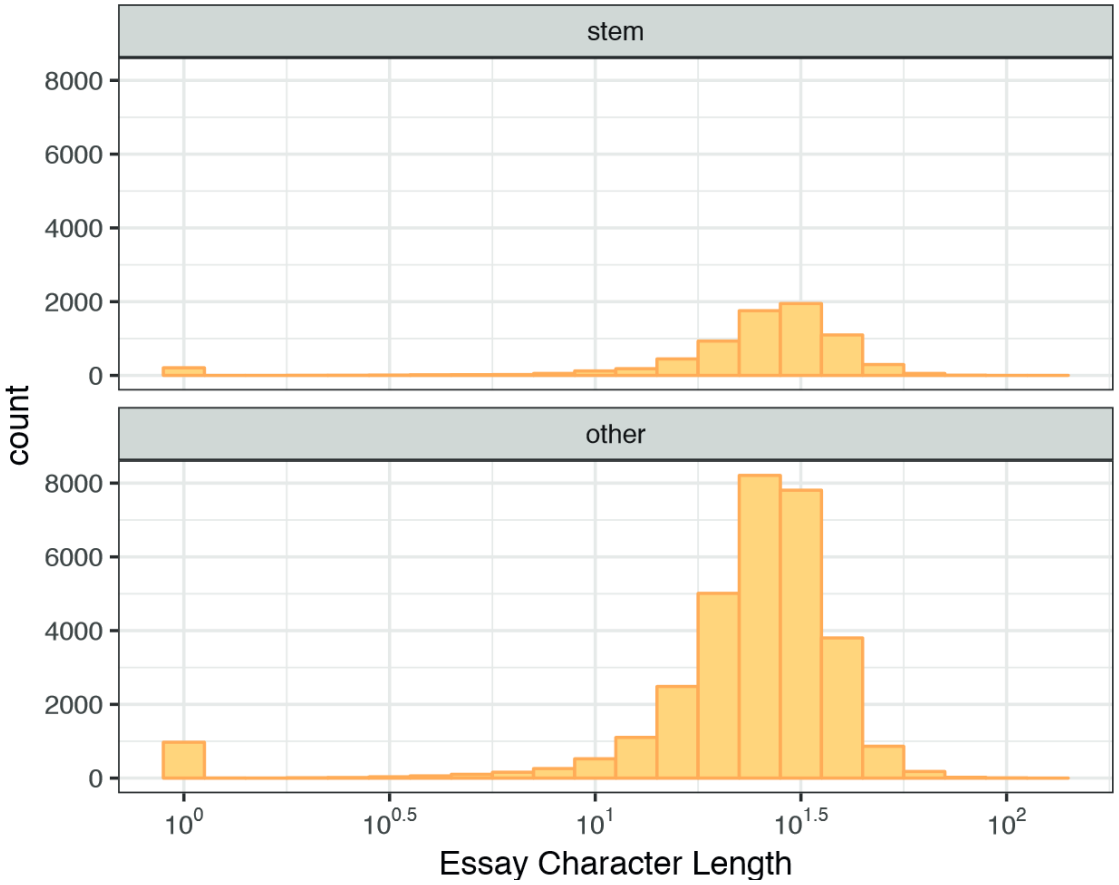
12 Feature Engineering

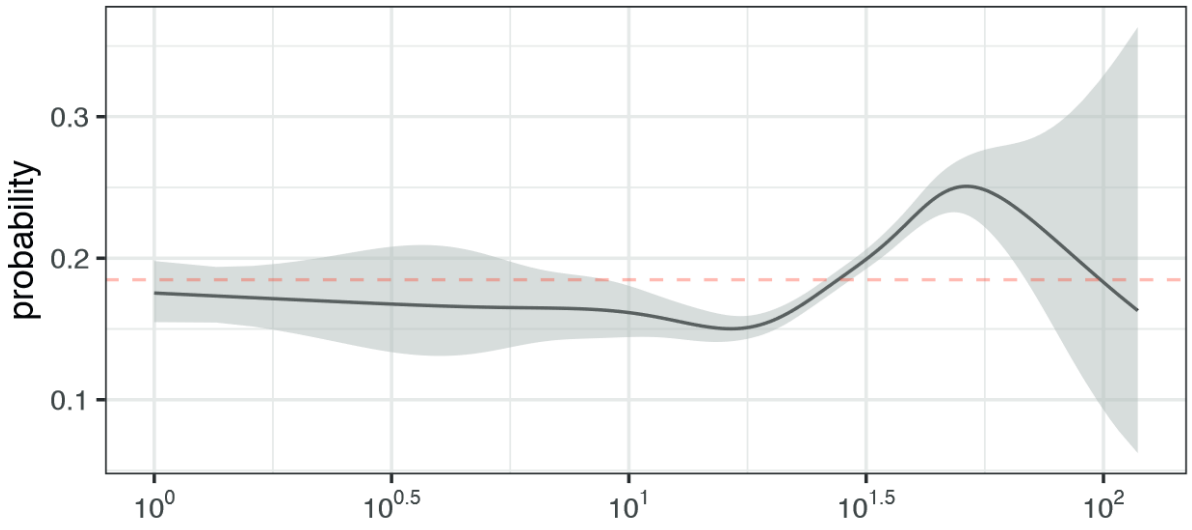
12.1 Exploratory Visualizations

- **Univariate visualizations** (Box Plots, Violin Plots, and Histograms) to understand the distribution of a single variable.
  - understand the distribution of the response variable (symmetric distribution / skewed distribution / distribution has multiple peaks or modes / outliers) to:
    - know its variation and provide a lower bound of the expectations of model performance (the residuals from a model that contains these predictors should have less variation than the variation of the response)
    - know if it should be transformed to a normal distribution prior to analysis to have better predictive performance
    - provide clues for including or creating features that help explain the response
  - understand the distribution of the predictors (when moderate number of predictors ($<$ ~100) / if not examine a subset of predictors that are thought to be important)
    - scatter plots or mosaic plots to see if any of the responses "cluster" with others
      - *correspondence analysis* can answered the question: linear regression for scatter plot or X^2 for mosaic plot. And *principal coordinates* can be computed to create new variables
- **scatter plots / bar charts + 95% confidence intervals / smoother** of individual predictors and the response outcome:
  - to easly test for new crucial predictor (e.g., if we find a strong linear relationship)
  - try to understand the relation between predictors and if some points don't follow the overall pattern trying to understand them could lead to a new feature.


Proportion
0.4
0.3
0.2
0.1
other
christianity
catholicism
buddhism
judaism
missing
islam
agnosticism
atheism
hinduism

(a)


stem
other
count
8000
6000
4000
2000
0
10^0
$10^{0.5}$
10^1
$10^{1.5}$
10^2
Essay Character Length



- a **heatmap** across the samples and predictors (e.g., to see when these unusual values occur)
- a pairwise **correlation** plot among the predictors (e.g., high degree of correlation is a clear indicator that the information present across the stations is redundant and could be eliminated or reduced.)
  - hierarchical cluster analysis to arrange samples in a way that those that are 'close' in the measurement space are also nearby in their location on the axis.
- **line plots** for time-based predictors (e.g., trends or patterns associated with time to know if variable's current value is more related to recent values than to values further apart in time)
- **PCA / PLS / MDS (multidimensional scaling)** : to engineer features that effectively condense the original predictors' information while retaining crucial predictive information. (if the first and first and second component captures 76.7% and 83.1% respectively the information is redundant and can likely be summarized in a more condensed fashion)
  - cumulative amount of variation summarized: how many components are required to summarize a sufficient amount of variation in the data
  - scatter plot of the first two components to detect clusters
  - violin plot of the first and second components against the underlying variables that appear to affect them the most

12.2 Postmodeling Exploratory Visualizations

To understand the next set of improvements.

- **Multiple linear regression (lm)**: identify relationships that may be useful to include in the model
  - *partial regression plot*:
- the first few levels of a regression or classification tree

12.3 Encoding Categorical Predictors

12.3.1 Creating Dummy Variables for Unordered Categories

Advatanges: Lead to zero-variance predictor which can be remove and omitting rarely occurring values and propagates this noise into the resampling estimates of performance. **Disadvatanges**: However if dummy zero-variance predictor is remove the model will not be able to predict **USE**: - When categories are small and it does not lead to zero-variance predictors **NOTE**: when model can support categorical data it is very difficult to predict if dummy variable will improve the model. Start without dummy variables and, if the model appears promising, to also try refitting using dummy variables.

12.3.2 Encoding Predictors with Many Categories

- *Creating Dummy Variables* and remove zero-variance predictor
- *Hashing function* to combine categories to create feature hashing
- *Create an "other" category*
- *Supervised Encoding Methods* to encode categorical predictors to numeric columns using the outcome data as a guide
  - *USE*: When the predictor has many possible values
  - *Technics*:
    - effect or likelihood encoding: (e.g., mean or median sale price of a house for each neighborhood from the training data and use this statistic to represent the factor level in the model)
    - logistic regression model (for classification problems):
    - linear regression model (for regression problems):
    - word/entity embedding: estimate a smaller set of numeric features that can be used to adequately represent the categorical predictors
    - hidden layers
  - *Disadvantage*:
    - generate error when a factor level has a single value (to solve the issue use *shrinking methods* such as Bayesian analysis)
    - increases the possibility of overfitting

    - ∗ can drastically underestimate the variation in the data and might give a falsely optimistic opinion of the utility of the new encoding column
  - *NOTES*: It is strongly recommended that either different data sets be used to estimate the encodings and the predictive model or that their derivation is conducted inside resampling so that the assessment set can measure the overfitting (if it exists).

12.3.3 Approaches for Novel Categories to enable the original model to be applied to new data without completely refitting it

- *Create a "other" category* and asign the new category no other
- *Create a zero-variance dummy* variable in the training or test set or both.
- *Supervised Encoding Methods* to encode categorical predictors to numeric columns using the outcome data as a guide
  - *USE*: When new levels appear after model training
- *Encodings for Ordered Data* (e.g., "low", "medium", and "high.")
  - *Technics*: polynomial contrast
    - ∗ *Advatanges*: By employing polynomial contrasts, we can investigate multiple relationships (linear, quadratic, etc.) simultaneously by including these in the same model.
    - ∗ *Disadvantage*:
      - · polynomial contrasts may not effectively relate a predictor to the response (For example, in some cases, one might expect a trend where "low" and "middle" samples have a roughly equivalent response but "high" samples have a much different response.)
      - · **Not recommended when there are moderate to high number of categories**
  - *Technics*: Translate the ordered categories into a single set of numeric scores based on context-specific information.

12.3.4 Approaches for Text Data

- Transform the text data into the *odds-ratio* of containing a keywords/link this can be extended to the odds-ratio of containing a text/link for each response variable (The rate of hyperlinks in the STEM profiles (response variable) was 21%, while this rate was 12.4% in the non-STEM profiles. For the STEM profiles, the odds of containing a hyperlink are relatively small with a value of 0.21/1-0.21 = 0.27. For the non-STEM profiles, it is even smaller (0.142).)
- *create "text-related" features*: (e.g., number of commas, hashtags, mentions, exclamation points)

- code *sentiment values*
- code *language used* (e.g., first-, second-, or third-person text and other language elements)

13 Engineering Numeric Predictors converting continuous predictors into a form that a model can better utilize.

13.1 Transformation:

- centering
- scaling
- transforming a distribution to symmetry

13.2 Feature engeneering

- transforming predictors in its original scale to nonlinear scales that may be informative.
  - *Techincs*:
    - basis expansions (e.g., Squared predictors for simplistic models such as regressions)
    - splines
    - combination of kernel function and PCA
  - *Disadvantage*: computational cost
- reduce the dimension of the predictors
  - *Techincs*:
    - PCA (unsupervised approach)
    - ICA (unsupervised approach)
    - NNMF (unsupervised approach)
    - PLS (supervised approach)
    - categorizing the response (only appropriate when the response is bimodal (or multimodal)).
- harness information in unlabeled data or dampen the effect of extreme samples
  - *Techincs*:
    - autoencoders

* spatial sign transformation
* distance and depth measures (e.g., class centroids for classification models: centers of the predictor data for each class. For each predictor, the distance to each class centroid can be calculated and these distances can be added to the model)

14 Working with Profile Data (*To Do*)

This type of data can occur if a sample is measured repeatedly over time, if a sample has many highly related/correlated predictors, or if sample measurements occur through a hierarchical structure.

Basic preprocessing steps for profiled data can include estimating and adjusting the baseline effect, reducing noise across the profile, and harnessing the information contained in the correlation among predictors. The latter in order to remove the characteristics that prevent this type of data from being used with most predictive models while simultaneously preserving the predictive signal between the profiles and the outcome.

15 SQL

15.1 SQL vs dplyr

SQL	dplyr
SELECT table1.column_name1	select
FROM	df %>%
UNION(remove duplicates)/UNION ALL (SELECT FROM WHERE … **UNION** SELECT FROM WHERE …)	bind_rows()
INNER JOIN table2 ON table1.column_name1 = table2.column_name1 AND table1.column_name2 = table2.column_name2	inner_join(., table2, by = c("column_name1" = "column_name1", "column_name2" = "column_name2"))
LEFT JOIN table2 ON table1.column_name = table2.column_name	left_join(., table2, by = c("column_name" = "column_name")
RIGHT JOIN table2 ON table1.column_name = table2.column_name	right_join(., table2, by = c("column_name" = "column_name")
FULL OUTER JOIN table2 ON table1.column_name = table2.column_name	full_join(., table2, by = c("column_name" = "column_name")
WHERE (IN [num/char]; BETWEEN [num] AND [num]; LIKE [char]; REGEXP/NOT REGEXP regex[char]; AND	filter (%in% ; %in% c(1:10); ==; grepl/!grepl)
GROUP BY (all columns in select)	group_by
HAVING (filter results of aggregate functions applied to grouped data)	filter
ORDER BY column1 ASC, column2 DESC	arrange(column1, desc(column2))
LIMIT [num]	slice_head(n = [num])

15.2 functions

- *MOD* Returns the remainder (number after the point 67 in 3.67) of a number divided by another number
- *LENGTH* Returns the length of a string (in bytes)
- *CONCAT*(col1, col2) Combine two or more strings into a single string
- *RIGHT/LEFT* Extracts a number of characters from a string (starting from RIGHT/LEFT)
- *ROUND*(VALUE, number of decimals) Rounds a number to a specified number of decimal places
- *CEILING* round it up to the next integer.
- *FLOOR* round it up to the last integer.
- *AVG* Average
- *CAST*(column AS data_type): convert the data type of a column
- *REPLACE*(original_string, old_substring, new_substring): replace occurrences of a specified substring with another substring in a given string. eg: SELECT REPLACE('Hello World', 'World', 'Universe') AS Result;
- *LAG/LEAD*(column OVER (ODER BY column_date))
- *MONTH* extract month
- *LPAD*(MONTH(trans_date), 2, '0') stands for "Left PADding," and it is a string function used in SQL to add characters to the left side of a string until it reaches a specified length.
- *DATE_SUB*('2019-07-27', INTERVAL 28 DAY) function used to subtract 28 days from the reference date
- *LOWER/UPPER* (column1) convert all characters in a string to lowercase/uppercase ex: CONCAT(UPPER(RIGHT(name, 1)), LOWER(LEFT(name, LENGTH(name) - 1))) AS name

15.3 Regex

^[char] start with
[char]$ end with
.* any number of characters

15.4 Match / regex used with the LIKE

% is a wildcard character to represent zero or more characters. When used in a LIKE pattern, % matches any sequence of characters.

15.5 Subqueries

WHERE column = (SELECT MAX(column) FROM data WHERE …);
FROM (SELECT MAX(column) FROM data);

15.6 Joins

When JOIN is use we need to include table1.column_name1 to select column

15.7 Mutate

15.7.1 Case when

CASE
WHEN condition1 THEN “result1”
WHEN condition2 THEN “result2”
ELSE “result”
END AS new_column_name

15.7.2 IF (only in MYSQL)

SELECT if(column=“confirmed”,1,0) | mutate(action = if_else(column=“confirmed”,1,0))

15.8 Exist

The EXISTS operator is used to test for the existence of any record in a subquery.
The EXISTS operator returns TRUE if the subquery returns one or more records.

WHERE EXISTS (SELECT column_name FROM table_name WHERE condition)
THEN “result1” # (if condition is TRUE return “result1”)
ELSE “result2”

CASE WHEN EXISTS (SELECT column_name FROM table_name WHERE condition)
THEN result

15.9 Joins

15.9.1 SELF JOIN

SELECT w1.*, w2.* FROM Weather w1 JOIN Weather w2 ON w1.recordDate = w2.recordDate + 1

w1 and *w2* are different table aliases for the same table.

15.9.2 CROSS JOIN

Used generally when merging a column with only one columns to avoid creating a huge table with all possible combinations keyword returns all matching records from both tables whether the other table matches or not.

15.10 WITH df AS

The WITH clause in SQL is used to define a Common Table Expression (CTE). A CTE is a temporary result set that can be referenced within the context of a SELECT, INSERT, UPDATE, or DELETE statement. The purpose of a CTE is to simplify complex queries, make the code more readable, and avoid repeating the same subquery multiple times.

WITH df AS (SELECT Signups.user_id, action, COUNT(Confirmations.action) AS conf FROM Signups LEFT JOIN Confirmations ON Signups.user_id = Confirmations.user_id GROUP BY action, Signups.user_id)

SELECT * FROM df

15.11 PIVOT tables

CASE WHEN … THEN … END

SELECT user_id, CASE WHEN action = “timeout” THEN conf END AS timeout, CASE WHEN action = “confirmed” THEN conf END AS confirmed, CASE WHEN action = “confirmed” THEN conf END / (CASE WHEN action = “timeout” THEN conf END + CASE WHEN action = “confirmed” THEN conf END) AS confirmation_rate FROM df

15.12 Windows functions

- OVER (PARTITION BY)

SELECT , *COUNT()* OVER(PARTITION BY column1) AS name1, COUNT(*) OVER(PARTITION BY column2, column3) AS name2 FROM data

SELECT *, SUM(weight) OVER (ORDER BY turn) # cumulative sum FROM Queue

see website SQL Window Functions

15.13 Common Table Expressions (CTEs)

CTEs in SQL provide a way to create temporary result sets that can be referenced within a SELECT, INSERT, UPDATE, or DELETE statement.

- WITH tablename AS (SELECT column1, column2 FROM your_table WHERE some_condition)

-CREATE, ALTER, DROP, RENAME, TRUNCATE, COMMENT

15.14 Data manipulation

Using DML statements to retrieve and manipulate data.

-INSERT, UPDATE, DELETE, MERGE, CALL, EXPLAIN PLAN, LOCK TABLE

15.15 Data security

Using DCL (data control language) commands to manage database security.

- GRANT
- REVOKE

15.16 Stored procedures

CREATE PROCEDURE procedure_name
AS
sql_statement (e.g., SELECT * FROM Customers)
GO;

EXEC procedure_name;

15.17 Index

To accelerate SQL performance

- clustered
- non clustered

15.18 Schema design

15.19 Query efficiency

- indexes
- avoiding subqueries
- optimizing the database schema

15.20 Database Normalization

Structure the table to eliminate redundant information, improve understanding, and make it easy to enhance, extend, and protect against insertion, update, and deletion anomalies

- 1NF : Don't use row order to convey information / don't mix data types within a column / don't have primary column (also call key) / don't store a repeating group of data on a single row
- 2NF: each columns (which are non-key atributes) must depend on all the column(s) that are considered primary key (e.g. player ID)
  - Example: player_rating depend on player_ID but not player_inventory column which is also a primary key in this table

- 3NF: every attributes (column) in a table should depend on the key column(s) the hole key and nothing but the key
  - Example: player_ID is the primary key/column. Non-key atribute player_rating depend on player_ID but non-key atribute player_skill depend on player_rating which is not a primary key

see Video for more detail about 4NF and 5NF

16 SQL in R

16.1 Create connection

```
con <- DBI::dbConnect(RSQLite::SQLite(), ":memory:")
copy_to(con, mtcars)
copy_to(con, iris)
DBI::dbListTables(con)
```

```
[1] "iris"         "mtcars"       "sqlite_stat1" "sqlite_stat4"
```

```
DBI::dbRemoveTable(con, "sqlite_stat1")
```

```
CREATE TABLE OccupationData (
    PersonName VARCHAR(50),
    OccupationType VARCHAR(20),
    Occupation VARCHAR(50)
);
```

```
SELECT *
FROM OccupationData;
```

Table 16.1: 0 records

PersonName	OccupationType	Occupation

```
INSERT INTO OccupationData (PersonName, OccupationType, Occupation)
VALUES
('John', 'Actor', 'JohnActor'),
('John', 'Doctor', 'JohnDoctor'),
('Jane', 'Actor', 'JaneActor'),
('Jane', 'Doctor', 'JaneDoctor');
```

```
SELECT *
FROM OccupationData;
```

Table 16.2: 4 records

PersonName	OccupationType	Occupation
John	Actor	JohnActor
John	Doctor	JohnDoctor
Jane	Actor	JaneActor
Jane	Doctor	JaneDoctor

```
SELECT *
FROM iris
WHERE Species LIKE "virginica"
UNION
SELECT *
FROM iris
WHERE Species LIKE "setosa"
```

Table 16.3: Displaying records 1 - 10

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4.3	3.0	1.1	0.1	setosa
4.4	2.9	1.4	0.2	setosa
4.4	3.0	1.3	0.2	setosa
4.4	3.2	1.3	0.2	setosa
4.5	2.3	1.3	0.3	setosa
4.6	3.1	1.5	0.2	setosa
4.6	3.2	1.4	0.2	setosa
4.6	3.4	1.4	0.3	setosa
4.6	3.6	1.0	0.2	setosa
4.7	3.2	1.3	0.2	setosa

```
SELECT COUNT(cyl)
FROM mtcars
WHERE cyl > 6
```

Table 16.4: 1 records

COUNT(cyl)
14

```
mtcarsSQL <- tbl(con, "mtcars")
query <- mtcarsSQL %>% filter(across(everything(), ~!is.na(.)))
query %>% show_query()
```

```
<SQL>
SELECT `mtcars`.*
FROM `mtcars`
WHERE
  (NOT((`mpg` IS NULL))) AND
  (NOT((`cyl` IS NULL))) AND
  (NOT((`disp` IS NULL))) AND
  (NOT((`hp` IS NULL))) AND
  (NOT((`drat` IS NULL))) AND
  (NOT((`wt` IS NULL))) AND
  (NOT((`qsec` IS NULL))) AND
  (NOT((`vs` IS NULL))) AND
  (NOT((`am` IS NULL))) AND
  (NOT((`gear` IS NULL))) AND
  (NOT((`carb` IS NULL)))
```

data from ("SQL Tutorial" n.d.)

17 Big Data

17.1 DataBases vs Warehouse vs Data Lake

Data Lake: Store all types of data DataBases: Raw data stored in tables Warehousen: Generally subset of data importe from the database through ETL used to perform analysis

17.2 Connect to databricks

17.2.1 Libraries

```
library(sparklyr)
library(pysparklyr)
```

17.2.2 Connection

```
sc <- spark_connect(
  master = "",
  cluster_id = "1026-175310-7cpsh3g8",
  token = "",
  method = "databricks_connect"
)
```

17.3 Save data from dplyr to databricks

Create a notebook in databricks

```
library(tidyverse)
spark_available_versions()
sc <- spark_connect(
  master = "local",
```

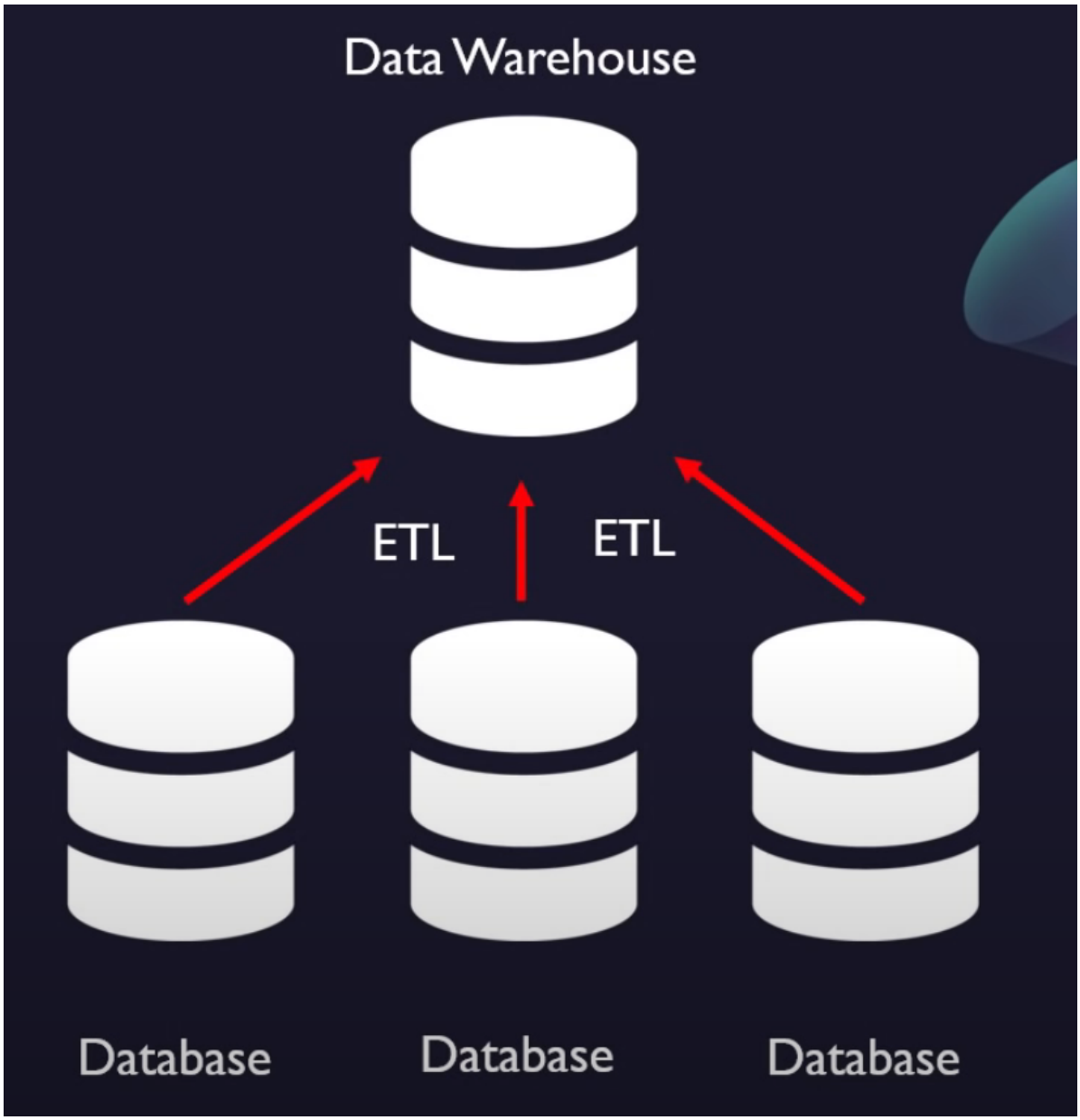


Figure 17.1: DataBases, Warehouse Data Lake

```
  version = "3.5"
)

df <- mtcars
df_spark <- copy_to(sc, df, "df_spark")
sdf_register(df_spark, "temp_table")
SparkR::registerTempTable(df_sparkr, "temp_df")
```

see more on (Ruiz n.d.)

18 Data Science for Business

ML is generally associated with making predictions at the individual level (inferences without attempting to understand the entire population), **while statistics** is used for inference, emphasizing the analysis of a sample from the population to draw conclusions about the population.

18.1 ML process

1) What is the goal of the analysis? (It is never to maximize accuracy.)
2) Think in context: What are the baseline benefits, and is it worth trying to improve them? (Understand the time cost of complex systems.)
3) Expected value framework:

- Decompose the problem into subtasks that we think we can solve, usually starting with existing tools.
- Calculate expected benefit
- Estimate each probabilities and values using data (statistics or ml, business values often need to be acquired from other sources)
- We may discover knowledge that will help us to solve the problem we had set out to solve, or we may discover something unexpected that leads us to other important successes
- Acknowledge the biases

4) Plan a way to evaluate/measure the benefits of the system (e.g., offline metrics like A/B tests / cost and benefit framework and online metrics like accuracy).
5) Communicate the results using explainable methods.
6) Ethical considerations: Be careful of biased data used to test the model, as the model will likely exacerbate those biases.

Equation 7-1. The general form of an expected value calculation

$$EV = p(o_1) \cdot v(o_1) + p(o_2) \cdot v(o_2) + p(o_3) \cdot v(o_3) \ldots$$

Figure 18.1: Each oi is a possible decision outcome; $p(o\text{i})$ is its probability and $v(o\ \text{i})$ is its value.

18.1.1 Expected value framework:

Expected value example:

- probability of response * (response benefit - expenses) - (1 - probability of response) * (benefit of not responding (zero) - the cost of the solicitation)
- probability of response * (response benefit - expenses) - (1 - probability of response) * (benefit of not responding (zero) - the cost of the solicitation) > 0
- probability of response * (response benefit - expenses) > (1 - probability of response) * (benefit of not responding (zero) - the cost of the solicitation)
- probability of response > 1 / ((benefit of not responding (zero) - the cost of the solicitation) + (response benefit - expenses))

With these example values, we should target the consumer as long as the estimated probability of responding is greater than 1 / ((benefit of not responding (zero) - the cost of the solicitation) + (response benefit - expenses))%.

A common way of expressing expected profit is to factor out the probabilities of seeing each class, often referred to as the class priors. The class priors, p(p) and p(n), specify the likelihood of seeing positive and negative instances, respectively. Factoring these out allows us to separate the influence of class imbalance from the fundamental predictive power of the model, as we will discuss in more detail in Chapter 8.

*Equation 7-2. Expected profit equation with priors p(**p**) and p(**n**) factored.*

$$\begin{aligned}\text{Expected profit} = \quad & p(\mathbf{p}) \cdot [p(\mathbf{Y} \mid \mathbf{p}) \cdot b(\mathbf{Y}, \mathbf{p}) + p(\mathbf{N} \mid \mathbf{p}) \cdot c(\mathbf{N}, \mathbf{p})] + \\ & p(\mathbf{n}) \cdot [p(\mathbf{N} \mid \mathbf{n}) \cdot b(\mathbf{N}, \mathbf{n}) + p(\mathbf{Y} \mid \mathbf{n}) \cdot c(\mathbf{Y}, \mathbf{n})]\end{aligned}$$

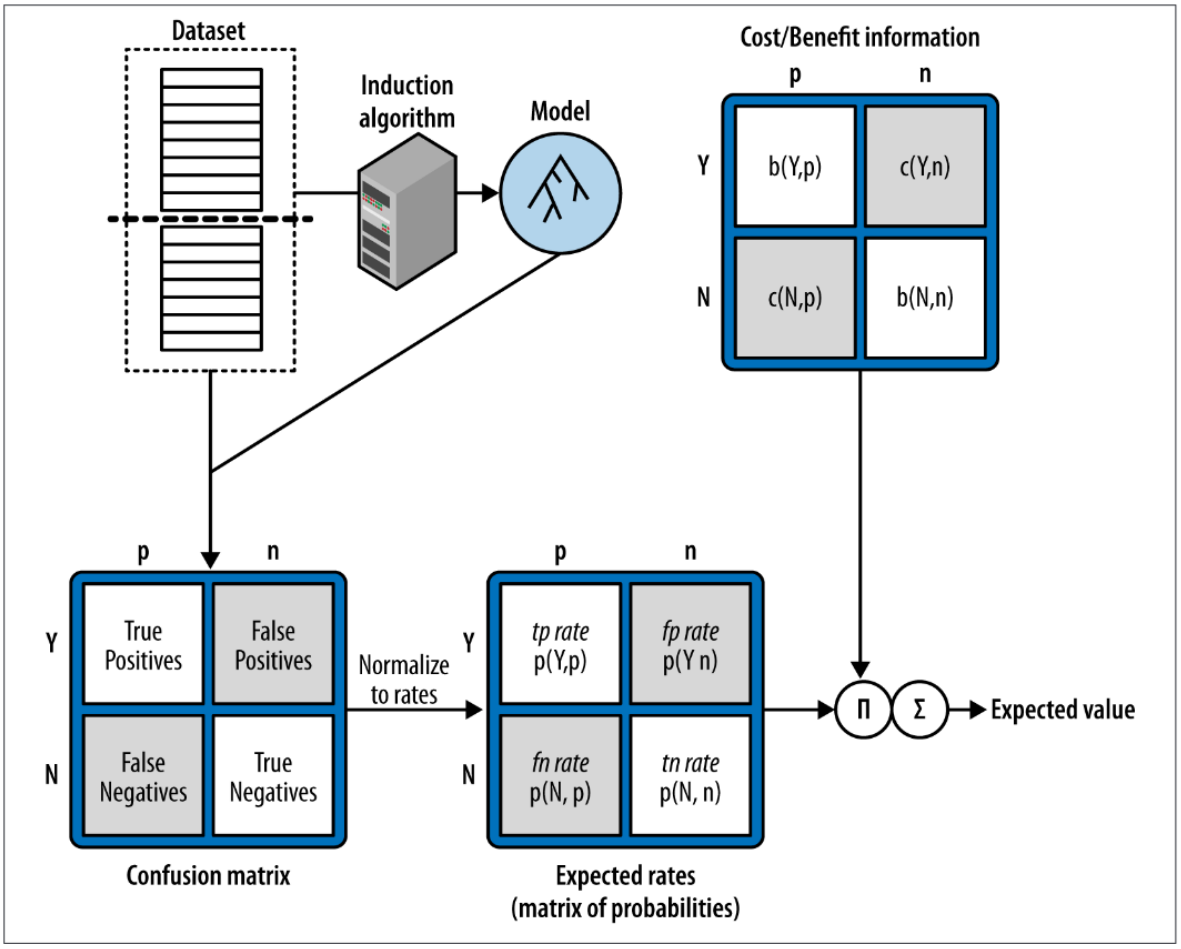


Figure 7-2. A diagram of the expected value calculation. The Π and Σ refer to the multiplication and summation in the expected value calculation.

Figure 18.2: Expected value calculation: Each *o*i corresponds to one cell of the confusion matrix. For example, what is the probability associated with the particular combination of a consumer being predicted to churn and actually does not churn? That would be estimated by the number of test-set consumers who fell into the confusion matrix cell (Y,n), divided by the total number of test-set consumers.

$$
\begin{aligned}
\text{expected profit} &= p(\mathbf{p}) \cdot [p(\mathbf{Y} \mid \mathbf{p}) \cdot b(\mathbf{Y}, \mathbf{p}) + p(\mathbf{N} \mid \mathbf{p}) \cdot c(\mathbf{N}, \mathbf{p})] + \\
&\quad p(\mathbf{n}) \cdot [p(\mathbf{N} \mid \mathbf{n}) \cdot b(\mathbf{N}, \mathbf{n}) + p(\mathbf{Y} \mid \mathbf{p}) \cdot c(\mathbf{Y}, \mathbf{n})] \\
&= 0.55 \cdot [0.92 \cdot b(\mathbf{Y}, \mathbf{p}) + 0.08 \cdot b(\mathbf{N}, \mathbf{p})] + \\
&\quad 0.45 \cdot [0.86 \cdot b(\mathbf{N}, \mathbf{n}) + 0.14 \cdot p(\mathbf{Y}, \mathbf{n})] \\
&= 0.55 \cdot [0.92 \cdot 99 + 0.08 \cdot 0] + \\
&\quad 0.45 \cdot [0.86 \cdot 0 + 0.14 \cdot \text{ - } 1] \\
&= 50.1 \text{ - } 0.063 \\
&\approx \mathbf{\$50.04}
\end{aligned}
$$

This expected value means that if we apply this model to a population of prospective customers and mail offers to those it classifies as positive, we can expect to make an average of about $50 profit per consumer.

Two pitfalls that are common when formulating cost-benefit matrices:

- It is important to make sure the signs of quantities in the costbenefit matrix are consistent. In this book we take benefits to be positive and costs to be negative. In many data mining studies, the focus is on minimizing cost rather than maximizing profit, so the signs are reversed. Mathematically, there is no difference. However, it is important to pick one view and be consistent.
- An easy mistake in formulating cost-benefit matrices is to "double count" by putting a benefit in one cell and a negative cost for the same thing in another cell (or vice versa). A useful practical test is to compute the benefit improvement for changing the decision on an example test instance.
  For example, say you've built a model to predict which accounts have been defrauded. You've determined that a fraud case costs $1,000 on average. If you decide that the benefit of catching fraud is therefore +$1,000/case on average, and the cost of missing fraud is -$1,000/case, then what would be the improvement in benefit for catching a case of fraud? You would calculate: b(Y,p) - b(N,p) = $1000 - (-$1000) = $2000 But intuitively you know that this improvement should only be about $1,000, so this error indicates double counting. The solution is to specify either that the benefit of catching fraud is $1,000 or that the cost of missing fraud is -$1,000, but not both. One should be zero.

18.2 Include costs of aquiring data

Different data sources may have different associated costs, and careful evaluation may show which can be chosen to maximize the return on investment.

18.3 CRISP-DM

Extracting useful knowledge from data to solve business problems can be treated systematically by following a process with reasonably well-defined stages.

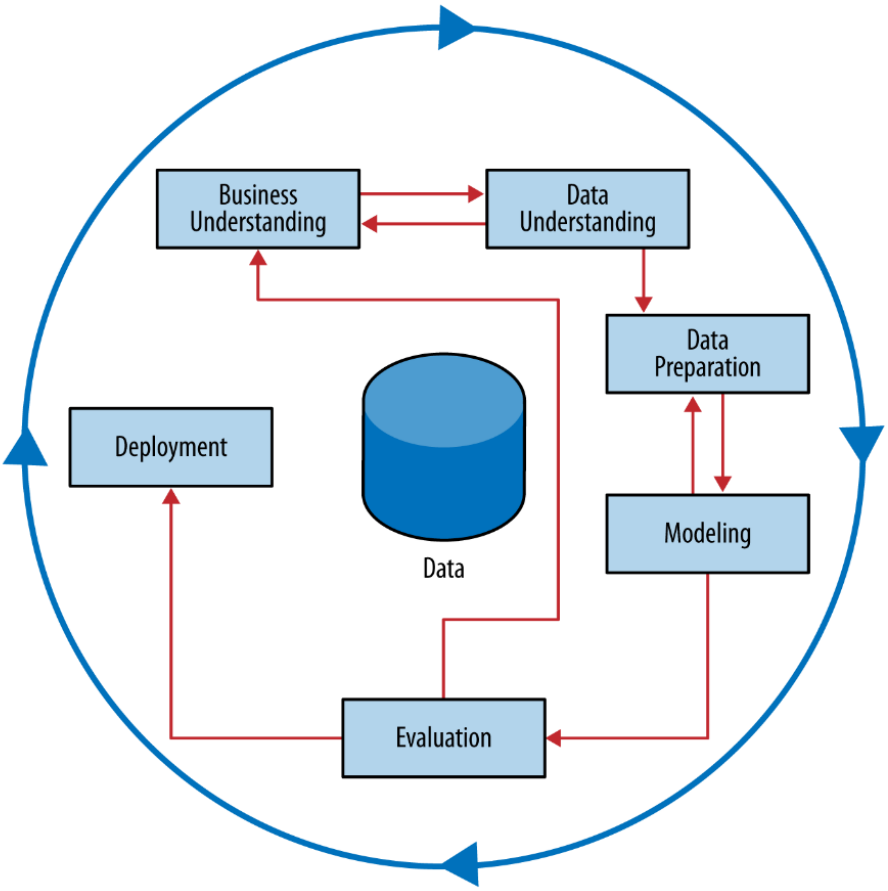


Figure 18.3: CRISP

18.4 ML list of data science tasks and tools

Tasks	Description	Tools
Prediction	Estimate or predict, for each individual, which of a (small) set of classes this individual belongs to or the numerical value of some variable for that individual	**Supervised**: Classification/Regression models
Causal modeling	Understand what events or actions actually influence other	**Supervised**: A/B tests
Similarity matching	Identify similar individuals based on data known about them	**Generally unsupervised**: clustering (also Classification, regression)
Clustering	Group individuals in a population together by their similarity, but **not driven by any specific purpose** / Exploratory analysis	**Unsupervised**: K-means, Hierarchical Clustering
Co-occurrence grouping	Find associations between entities based on transactions involving them: e.g., What items are commonly purchased together? While clustering looks at similarity between objects based on the objects' attributes, co-occurrence grouping considers similarity of objects based on their appearing together in transactions.	**Unsupervised**: cluster algorithms, Hidden Markov Models
Profiling	characterize the typical behavior of an individual, group, or population e.g. "What is the typical cell phone usage of this customer segment?"	**Generally Unsupervised**: Cluster analysis, Anomalies detection; (also NLP, descriptive statistics)

18.5 Data Science process questions

Business and Data Understanding

- What exactly is the business problem to be solved?
- Is the data science solution formulated appropriately to solve this business problem? NB: sometimes we have to make judicious approximations.
- What business entity does an instance/example correspond to?
- the problem a supervised or unsupervised problem? — If supervised — Is a target variable defined? — If so, is it defined precisely? — Think about the values it can take.
- Are the attributes defined precisely? — Think about the values they can take.
- For supervised problems: will modeling this target variable improve the stated business problem? An important subproblem? If the latter, is the rest of the business problem addressed?
- Does framing the problem in terms of expected value help to structure the subtasks that need to be solved?
- If unsupervised, is there an "exploratory data analysis" path well defined? (That is, where is the analysis going?)

Data Preparation

- Will it be practical to get values for attributes and create feature vectors, and put them into a single table?
- If not, is an alternative data format defined clearly and precisely? Is this taken into account in the later stages of the project? (Many of the later methods/techniques assume the dataset is in feature vector format.)
- If the modeling will be supervised, is the target variable well defined? Is it clear how to get values for the target variable (for training and testing) and put them into the table?
- How exactly will the values for the target variable be acquired? Are there any costs involved? If so, are the costs taken into account in the proposal?
- Are the data being drawn from the similar population to which the model will be applied? If there are discrepancies, are the selection biases noted clearly? Is there a plan for how to compensate for them?

Modeling

- Is the choice of model appropriate for the choice of target variable?
  - Classification, class probability estimation, ranking, regression, clustering, etc.
- Does the model/modeling technique meet the other requirements of the task?
  - Generalization performance, comprehensibility, speed of learning, speed of application, amount of data required, type of data, missing values?

  - Is the choice of modeling technique compatible with prior knowledge of problem (e.g., is a linear model being proposed for a definitely nonlinear problem)?

- Should various models be tried and compared (in evaluation)?
- For clustering, is there a similarity metric defined? Does it make sense for the business problem?

Evaluation and Deployment

- Is there a plan for domain-knowledge validation?
  - Will domain experts or stakeholders want to vet the model before deployment? If so, will the model be in a form they can understand?
- Is the evaluation setup and metric appropriate for the business task? Recall the original formulation.
  - Are business costs and benefits taken into account?
  - For classification, how is a classification threshold chosen?
  - Are probability estimates used directly?
  - Is ranking more appropriate (e.g., for a fixed budget)?
  - For regression, how will you evaluate the quality of numeric predictions? Why is this the right way in the context of the problem?
- Does the evaluation use holdout data?
  - Cross-validation is one technique.
- Against what baselines will the results be compared?
  - Why do these make sense in the context of the actual problem to be solved?
  - Is there a plan to evaluate the baseline methods objectively as well?
- For clustering, how will the clustering be understood?
- Will deployment as planned actually (best) address the stated business problem?
- If the project expense has to be justified to stake

18.6 Proposal Example

1) Explain the problem.
2) Define the goal (pay attention to the KPI used and report if the problem has slighlty change due to data limitations).
3) Outline the approach to achieving the goal:
   - Specify the data to be used/collected.
   - What attributes are going to be used

- Justify the choice of model(s). Think about the comprehensibility of the model to stakeholders

4) Document the assumptions made during modeling.
5) Describe the method for testing model performance.
6) If pilot study already has been conducted and learning curves having been produced on data samples report and estimate of model performance
7) Identify individuals responsible for peer reviewing the project (keep in mind that favour a simple model to allow other people to understand the model).
8) Detail the strategy for tracking model performance.

18.7 DataBricks

Create a notebook in databricks

19 Supply Chain

20 A/B Testing

Examines user experience through randomized tests with two variants.

Typical steps

1) Determine the evaluation metric and experiment goals
2) Select a significance level and power threshold 1 -
3) Calculate the required sample size per variation
4) Randomly assign users into control and treatment groups
5) Measure and analyze results using the appropriate test

The required sample size depends on , , and the MDE Minimum Detectable Effect - the target relative minimum increase over the baseline that should be observed from a test Overall Evaluation Criterion - quantitative measure of the test's objective, commonly used when short and long-term metrics have inverse relationships

Tools

Univariate Testing:

Statistical Test	Description	Use case
Z-Test	Test differences between two means	**When large sample size and known population variance (>30)**
T-Test	Test differences between two means	**When small sample size and unknown population variance (<30)**
Welch's T-Test	Test differences between two means	**Adaptation of the t-test that does not assume equal variances (homoscedasticity: spread or dispersion of data points around the mean is consistent across all groups) offering more flexibility**

Statistical Test	Description	Use case
Mann-Whitney U Test	Comparing two independent groups	**When data is not normally distributed**
ANOVA	Test differences between three or more means	
Chi-Squarde Test	Test if there is a significant association between two categorical variables	**e.g., sexe and energy drinks**
Fisher's Exact Test	Test if there is a significant association between two categorical variables	**When small sample size <30**
Multivariate Testing	compares 3+ variants or combinations, but requires larger sample sizes	

Bonferroni Correction - when conducting n tests, run each test at the n significance level, which lowers the false positive rate of| finding effects by chance

20.0.1 Network Effects

Changes that occur due to effect spillover from other groups.

Typical steps To detect group interference:

1) Split the population into distinct clusters
2) Randomly assign half the clusters to the control and treatment groups A1 and B1
3) Randomize the other half at the user-level and assign to control and treatment groups A2 and B2
4) Intuitively, if there are network effects, then the tests will have different results To account for network effects, randomize users based on time, cluster, or location

###Sequential Testing

Allows for early experiment stopping by drawing statistical borders based on the Type I Error rate. If the effect reaches a border, the test can be stopped. Used to combat peeking (preliminarily checking results of a test), which can inflate p-values and lead to incorrect conclusions.

###Cohort Analysis

Examines specific groups of users based on behavior or time and can help identify whether novelty or primacy effects are present

(**wangDataScienceCheatsheet2021?**)

References

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer. https://doi.org/10.1007/978-0-387-84858-7.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2021. *An Introduction to Statistical Learning: With Applications in R*. Springer Texts in Statistics. New York, NY: Springer US. https://doi.org/10.1007/978-1-0716-1418-1.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. New York, NY: Springer. https://doi.org/10.1007/978-1-4614-6849-3.

Ruiz, Edgar, Kevin Kuo. n.d. *Mastering Spark with R*. Accessed September 22, 2023.

"SQL Tutorial." n.d. https://www.w3schools.com/sql/default.asp. Accessed August 26, 2023.

"What Is the Difference Between Correlation and Cointegration? Is Cointegration a Good Measure of Risk?" n.d. *Quora*. https://www.quora.com/What-is-the-difference-between-correlation-and-cointegration-Is-cointegration-a-good-measure-of-risk. Accessed July 17, 2023.