Course Notes - Introduction to Analytics Modeling

Nolan MacDonald

Fall 2024

Contents

Chapter 1

Module 1 - Introduction

1.1 M1L1 - Course Overview

1.2 M1L2 - Course Structure

1.3 M1L2a - Homework Grading and Q&A

1.3.1 Homework Format

The main focus of the homework will be your analysis of your results, **not your code.** For a top score, you shouldn't just run some code and display some results; rather, you should also discuss the results qualitatively, point out anything surprising, and comment on possible explanations. The top recommended forms for your analysis, in order, are: pdf, html, txt, and then word doc.

We recommend **not** including your name / e-mail in your homework PDF. We (TAs) and the system will know who you are, and submitting anonymously will not cause any problems assigning your grade later. It will also avoid potential problems of bias.

1.3.2 Grading Homeworks

Once you've submitted your homework, you will have to complete 3-Peer Reviews. You are given a rubric with **scoring guidelines with possible values of 100, 90, 75, 50, and 0.** When doing the grading, Canvas does allow you to give options outside of the scale, but please stick to it. I can see who gave the off-scale score, and it only frustrates your fellow students and makes my job a little harder. Also, if you have a homework marked as "late" in your grading queue, please grade it as normal. We have a very small tolerance between when peer reviews are assigned and homeworks are due to allow for students who had technical issues to submit.

- 100 is "All correct (perhaps except a few details) with a deeper solution than expected"
  - This means that if students gave answers that delved deeper into the analytical methods and modeling techniques with appropriate solutions and explanations etc, then that constitutes a 100. All questions are completed. **When giving this grade, please comment what you believe the person did well and how they went above and beyond.** I cannot force you to leave a comment, but it will help stay consistent with the regrading policy that will be mentioned later.
- 90 is for "most or all correct."

  - This means that student provided code, answered the questions asked with code output, and provided reasonable but basic explanations for their answers. There can be minor errors like slight inconsistencies or minor misunderstandings. All questions are answered. **This is the "default" grade. If a student does what is asked, and provides reasonable explanation for their work, but doesn't go above and beyond, this is their grade.** While comments here can be nice, if you have nothing else to add other than "they did everything right," you do not have to leave a comment.

- 75 is "not correct, but a reasonable attempt."
  - There is some code, solutions, and explanation, but the explanation is faulty, incorrect, or non-existent or the solution values are completely different than outlined in the homework solutions or solutions and explanation do not make sense. At least half of the questions are answered/attempted including the coding questions. **When giving this grade, please comment what you believe the person did that was fundamentally incorrect.** This will help the student learn and will also help me in case of a regrade request.
- 50 is "Not correct, insufficient effort."
  - There is a distinct lack of effort on homework like no code, no answers, no explanations, or little of any of these. Particularly, if a student only answers a descriptive problem (one which involves answering a prompt) and does none of the coding problems, then this would also be considered insufficient effort. **When giving this grade, please comment what you believe the person did that was fundamentally incorrect and explain what part of the homework they did not do.** This will help the student learn and will also help me in case of a regrade request.
- 0 is "Not Submitted"
  - There is nothing of merit submitted. The student did not attempt the problems, just submitted random or unrelated work to try and obtain a 50. **When giving this grade, it is because the student literally had nothing there, was completely unrelated (a picture of a ham-sandwich), or basically un-attempted.** If you feel like the student was just being lazy, you most likely should give them a zero.
- Other info: Optional truly means optional. Doing an optional part does not guarantee a 100. On the other end, You can still get a 100 even if you don't do an optional part. An optional part can contribute to a deeper analysis, but isn't required nor does it guarantee it.
- Please try to be helpful with your comments if you provide any.
  - Also, try to provide appropriate feedback or comments based on the homework questions being asked.
  - For example, I could say that "exploring more C values for Q2.2.1 would have shown a wider range/change in the model accuracy" or "using different kernels would result in higher model accuracy"
    - These comments are appropriate but do not result in a drop in the student's grade since exploring different kernels was optional and we did not give a specific range of C values to look at.

1.4 M1L3 - Modeling

Chapter 2

Module 2 - Classification

Chapter 3

Module 3 - Validation

3.1 Overview

Module 3 will continue with basic machine learning algorithms. The modules will cover couple of cross-cutting concepts and the important topic of model validation.

Additional References:

[1]. A Survey of Cross-Validation Procedures for Model Selection

3.2 M3L1 - Introduction to Validation

Validation

- How good is the model?

Data has two types of patterns

- **Real Effect** - Real relationship between attributes and response
- **Random Effect** - Random, but looks like a real effect

Fitting matches both real and random effects

- Real effects - Same in all data sets
- Random effects - Different in all data sets

Example: What day of the month were you born?

- Training Data: 3, 21, 24, 24, 25, 26, 27, 30, 30, 31
- Best Predictor: You were born on the 26th
  - Right in the middle of 9/10 data points
- **This is a random effect!**
- This model using 9/10 from 21-31 doesn't have a large error
- If new data showed 2, 9, 11, 12, 14, 21, 24, 24, 29, 31

  - Much larger error due to the uniform spread over the month
- Was this just luck? (3, 21, 24, etc.)
  - No, some random pattern would have shown up
    * Early in month
    * Middle of month
    * Even/odd numbered day
    * Day is multiple of 3
    * Day is close to one of my kids birthdays
    * Etc.

3.2.1 M3L1 - Summary

- The example proves we can't measure the model's effectiveness on data it was trained on
- Model fit captures real and random effects
- Only real effects are duplicated in other data

Don't judge a model based on how well it fits the training data.

- Validation is crucial to determine how good a model is and how accurately it performs on new data
- Measuring a model's performance on the same training data used to create it is not a good approach, as it will be too optimistic
- Any dataset contains both real effects (true relationships) and random effects (patterns that occur by chance)
- When fitting a model to training data, it captures both real and random effects
- However, when using the model on new data, only the real effects will persist, while the random effects will be different
- An example is given of a silly model that predicts people's birth dates based on a random pattern in the training data, which would not generalize well
- The key takeaway is that we cannot rely on training data performance to evaluate a model - we need a separate validation process to get an accurate assessment of its effectiveness

3.3 M3L2 - Validation and Test Data Sets

Measure a model's performance:

- A larger set of data to fit the model
- A smaller set of data to measure the model's effectiveness

Splitting Data:

- Training set (larger) to fit model
- Validation set (smaller) to estimate effectiveness

Training and Validation Sets:

- Observed performance = real quality + random effects
  - High-performing models more likely to have *above-average random effects*
- Observed performance of chosen model is *probably too optimistic*

Test Sets:

- Training data set to fit the models
- Validation data set to choose best model
- Test data set to estimate performance of chosen model

Overall:

- Training Set - Building models
- Validation Set - Picking a model
- Test Set - Estimating performance of chosen model

3.3.1 Summary

- Using only training data to evaluate a model's performance is often too optimistic, as the model overfits to random patterns in the training data.
- **To get a better measure of performance, we use separate validation and test sets:**
  - Training set: Used to fit/build the model
  - Validation set: Used to evaluate and compare different models
  - Test set: Used to get an unbiased final estimate of the chosen model's performance
- The validation set helps select the best model, but its performance estimate may still be inflated due to random chance when choosing the "best" model.
- The test set provides a final unbiased performance estimate for the selected model.
- **General process:**
  - Train multiple models on training data
  - Evaluate models on validation set and select best one
  - Estimate final performance of chosen model on test set
- This three-way split helps avoid overfitting and provides a more realistic assessment of how well the model will generalize to new data.
- There are different ways to split data into training, validation and test sets, which will be covered in a future lesson.

The key takeaway is that using separate datasets for training, model selection, and final evaluation helps produce more reliable and generalizable machine learning models.

3.4 M3L3 - Splitting Data

- Training data too optimistic so we need to use a test set
- Training data set to build model
- Validation set to compare models
- Test set to estimate performance of chosen model

How do we split data into traning, validation, and test sets?

- Method 1: Random - Randomly choose data points
- Method 2: Rotation - Take turns selecting points

Training-Validation-Training-Test-Training

- Randomness could give one set more early or late data
  - Rotation equally separates data
- Rotation may introduce bias

3.4.1 Summary

- Data needs to be split into training, validation, and test sets to properly evaluate model performance.
- **Recommended splits:**
  - For two sets: 70-90% training, 10-30% testing (Rule of thumb)
  - For three sets:
    - 50% training, 25% validation, 25% testing
    - 60% training, 20% validation, 20% testing
    - 70% training, 15% validation, 15% testing
- **Two main approaches for splitting data:**
  - Simple randomness: Randomly assign data points to each set
  - Rotation: Systematically rotate through assigning points to each set
- **Advantages of rotation:**
  - Ensures equal representation of data across time periods
  - Avoids potential biases from random sampling
- **Disadvantages of rotation:**
  - May introduce other biases if not done carefully (e.g. only certain days of week in each set)
- A hybrid approach combining randomness and rotation can be used to avoid biases.
- Cross-validation is another technique for using data, which will be covered in a future lesson.

It is important to properly split data to get accurate model evaluations, while there are tradeoffs between different splitting approaches.

3.5 M3L4 - Cross-Validations

- What if important data only appears in validation or test sets? *Use cross-validation!*
- Use of `k` in analytics
  - k-means
  - k-nearest neighbor
  - k-fold cross validation

k-fold Cross-Validation

- For each of the `k` parts:
  - Train the model on all the other parts
  - Evaluate it on the one remaining part
- *Average the k evaluations to estimate the model's quality*
- No standard number for `k`, but `k=10` is common

ANSWER: NONE!

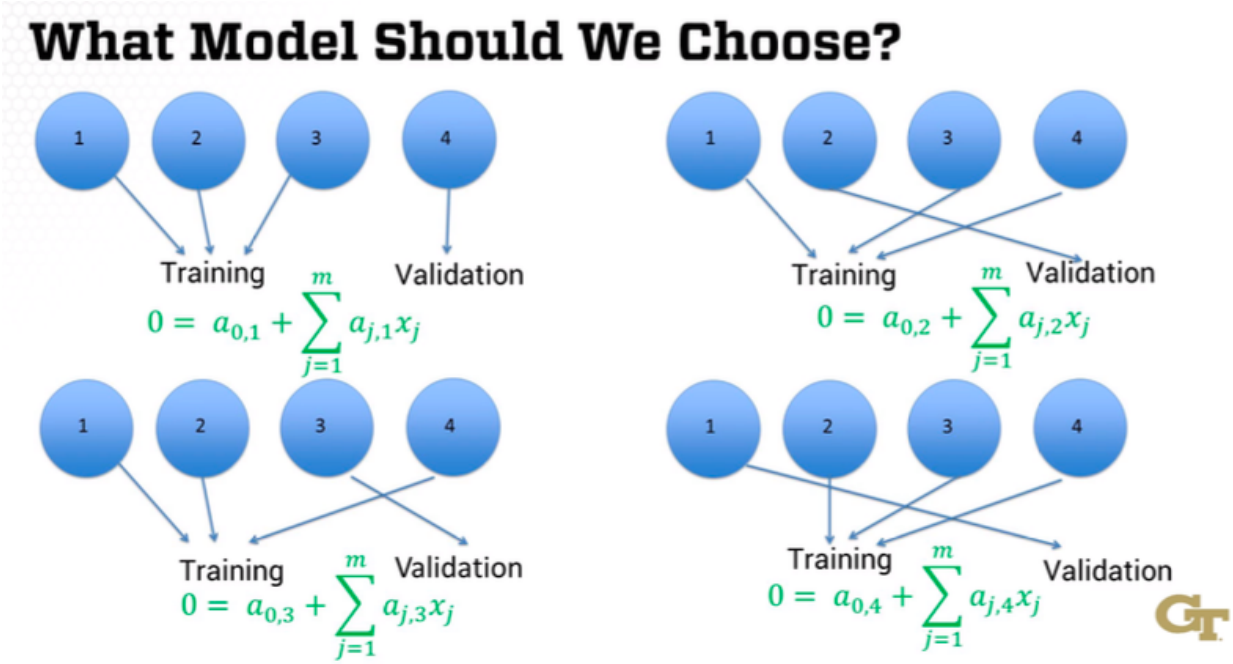

Figure 3.1: k-Fold Cross Validation

- Do not average the coefficients over four splits
- Train the model again using all the data

k-Fold Cross-Validation provides:

- Better use of data
- Better estimate of model quality
- Choose model more effectively

3.5.1 Summary

- **Purpose of Cross-Validation:** Cross-validation is introduced as a technique to ensure that important data points are not excluded from the training set, which can happen if they only appear in the validation or test sets. This method helps in making better use of the data available.
- **Types of Cross-Validation:** The lecture specifically discusses k-fold cross-validation, a popular method in analytics. The 'k' in k-fold cross-validation indicates the number of parts the data is split into for training and validation purposes.
- **Process of K-Fold Cross-Validation:**
  - The data is divided into k parts. For example, if k=4, the data is split into four parts.
  - The model is trained on k-1 parts and validated on the remaining part. This process is repeated k times, with each part being used as the validation set once.
  - Every data point is used for training in k-1 models, ensuring no important data is left out.
- **Model Evaluation:** The performance of the model is evaluated by averaging the results from the k different validation sets. This average provides an estimate of the model's quality.
- **Choosing the Final Model:** After using cross-validation to select a model, the final model is retrained using all the data parts together. This ensures the model benefits from the full dataset.
- **Common Practice:** While there is no standard number for k, using k=10 is common in practice.

Overall, cross-validation is emphasized as a crucial step in model selection and evaluation, helping to improve the reliability of the analytics process.

Chapter 4

Module 4 - Clustering

4.1 Overview

Module 4 will continue with basic machine learning algorithms. The focus will be on clustering models. The modules will cover couple of cross-cutting concepts, including distance norms, and k-means clustering.

Additional References:

[2]. Data Mining Algorithms In R/Clustering/K-Means

4.2 M4L1 - Introduction to Clustering

Clustering: An unsupervised machine learning technique designed to group unlabeled examples based on their similarity to each other.

- Grouping data points
- *Important to note: If the examples are labeled, the grouping is called classification*

Examples of Clustering:

- Targeted marketing/market segmentation
  - Potential customers need a message that would be most likely to encourage them to buy
- For example, if we were selling a SUV:
  - Size
  - Price
  - Versatility
  - Coolness
- Each set of people would be a cluster
- We would try to use data to split consumers into sets to discover what marketing they should be shown
- You can examine a cluster and it may not always be correct, which can help you find a meaningful cluster in your data
  - For example, we did not consider gas mileage
- Other examples:

- Targeted marketing/market segmentation
- Personalized medicine
- Locating facilities - Look at where people live and provide a police station for each cluster
- Image analysis - CAPCHA
- Initial data investigation

4.3 M4L2 - Distance Norms

The choice of distance measures is important in clustering. Distance measures define how the similarity of the two element are calculated and influences the shape of the clusters.

Euclidean (straight-line) distance:

- Distance in Euclidean space by length of straight line segment between two points

$$distance = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Rectilinear (Manhattan) Distance:

- Commonly used in city planning with a grid, hence Manhattan term

$$distance = \sum_{i=1}^{n}|x_i - y_i| = |x_1 - y_1| + |x_2 - y_2|$$

Minkowski (p-norm) Distance:

- We can describe both euclidean (p=2) and rectilinear (p=1) distance with p-norm (or Minkowski) distance

$$distance = \sqrt[p]{\sum_{i=1}^{n}|x_i - y_i|^p} = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p}$$

The most common values for p include 1, 2, and ∞ (∞-norm distance).

Infinity-Norm Distance (∞-norm):

The infinity norm simply measures how large the vector is by the magnitude of its largest entry. Simply put, it is the largest of a set of numbers in an absolute of values (the biggest).

$$distance = \lim_{p\to\infty} \sqrt[\infty]{\sum_{i=1}^{n}|x_i - y_i|^\infty}$$

The largest value for ∞ (assume p is 8) is dominated by the largest power. The term to the 7th power would be small compared to the 8th power. This means considering distance, the sum of terms is equal to the largest $|x_i - y_i|$ to the infinity power.

Note: Should include the limit to infinity, but for simplicity the equations do not all include the limit.

$$distance = \lim_{p\to\infty} \sqrt[\infty]{\sum_{i=1}^{n}|x_i - y_i|^\infty} = \sqrt[\infty]{\max_i |x_i - y_i|^\infty} = \max_i |x_i - y_i|$$

4.4 M4L3 - K-Means Clustering

The K-Means algorithm is a popular technique of representative-based clustering. K-Means is a simple learning algorithm for clustering analysis. The goal of K-Means algorithm is to find the best division of n entities in k groups, so that the total distance between the group's members and its corresponding centroid, representative of the group, is minimized [2].

Consider the K-Means algorithm defined as:

$$\min_{y,z} \sum_i \sum_k y_{ik} \sqrt{\sum_j (x_{ij} - z_{jk})^2}$$

The algorithm is subject to $\sum_k y_{ik} = 1$ for each `i` where:

- x_{ij} is attribute `j` of data point `i`
- y_{ik} is 1 if data point `i` is in cluster k, 0 if not
- z_{jk} is coordinate `j` of cluster center `k`

Adds up all data points to cluster centers but only when the data point is in the cluster.

0. Pick `k` cluster centers within range of data (e.g., Pick 3 points, `k=3`, where each point is a cluster center)
1. Assign each data point to nearest cluster center
2. Recalculate cluster centers (centroids of the data points in the cluster)

This could result in new cluster centers with cluster points that are more applicable for another cluster. There is an iterative process for 1 and 2 that are repeated until there are no changes. Stops when no data points change clusters.

K-Means Algorithm Overview:

- Machine Learning
- Heuristic - Fast, good, but not guaranteed to find absolute best solution
- Expectation-Maximization (EM) Algorithm
  - Maximizing the negative distance to a cluster center

4.5 M4L4 - Practical Details for K-Means

Using the K-Means algorithm in practice

4.5.1 Summary

- **Handling Outliers:** K-means will assign outliers to the nearest cluster, but this can distort results. While one option is to remove outliers, a more thoughtful approach is to investigate their significance and implications for your analysis.
- **Algorithm Limitations:** K-means is a heuristic, meaning it's not guaranteed to find the best clustering but is efficient and often finds good solutions. To improve results, it's advised to run k-means multiple times with different initial cluster centers and compare the outcomes.

- **Determining the Number of Clusters:** The number of clusters (k) can be optimized by running the algorithm with different k values and using an "elbow diagram" to identify where increasing the number of clusters no longer significantly improves the solution. However, practical considerations should also guide this choice, depending on the context.
- **Balancing Science and Art:** The lecture emphasizes the importance of blending data science with the "art" of analytics, where understanding the situation and making informed decisions can provide greater value than merely running algorithms.

4.6 M4L5 - Clustering for Prediction

4.6.1 Summary

- **Clustering Recap:** Clustering involves grouping data points based on their similarity and proximity. The k-means heuristic is a common method for finding good clusterings.
- **Predictive Clustering:** K-means clustering can be used predictively by determining which cluster a new data point should belong to, typically by finding the closest cluster center.
- **Handling New Data Points:** If a new data point falls within an existing cluster, it is straightforward to assign it to that cluster. If not, the point is assigned to the nearest cluster center.
- **Voronoi Diagrams:** The space around each cluster center can be divided into regions, where each region represents the area closer to that center than to any other. This is visualized using a Voronoi diagram.
- **Historical Context:** Voronoi diagrams have been used historically, including in the analysis of a cholera outbreak in London over 150 years ago, and by mathematicians like Rene Descartes in the 1600s.
- **Old Ideas in Analytics:** Some effective analytical techniques, like Voronoi diagrams, are not new but have been around for a long time and remain valuable.

4.7 M4L6 - Clustering v. Classification

4.7.1 Summary

- **Classification Models:** These involve a set of data points where both their attributes and correct groupings (responses) are known. For example, in loan application data, we know whether applicants repaid their loans (blue) or not (red). Classification models use both attributes and known responses to classify new data points. This process is known as supervised learning because it uses observed responses to guide the model.

- **Clustering Models:** In contrast, clustering models start with a set of data points where only the attributes are known, and the correct groupings are not known. The model must determine how to group the data points based solely on their attributes. This is known as unsupervised learning because there are no observed responses to guide the model.

Supervised learning is more common in analytics (such as classification) but unsupervised learning (such as clustering) is also a valuable tool.

Chapter 5

Appendix A: Glossary

5.1 Basic Machine Learning

Lessons 2.1-2.2, 2.4-2.6, 2.8, 4.1, 4.3-4.6, 6.1-6.3, 16.4

Algorithm: Step-by-step procedure designed to carry out a task.

Change detection: Identifying when a significant change has taken place in a process.

Classification: The separation of data into two or more categories, or (a point's classification) the category a data point is put into.

Classifier: A boundary that separates the data into two or more categories. Also (more generally) an algorithm that performs classification.

Cluster: A group of points identified as near/similar to each other.

Cluster center: In some clustering algorithms (like -means clustering), the central point (often the centroid) of a cluster of data points.

Clustering: Separation of data points into groups ("clusters") based on nearness/similarity to each other. A common form of unsupervised learning.

CUSUM: Change detection method that compares observed distribution mean with a threshold level of change. Short for "cumulative sum".

Deep learning: Neural network-type model with many hidden layers.

Dimension: A feature of the data points (for example, height or credit score). (Note that there is also a mathematical definition for this word.)

EM algorithm: Expectation-maximization algorithm.

Expectation-maximization algorithm (EM algorithm): General description of an algorithm with two steps (often iterated), one that finds the function for the expected likelihood of getting the response given current parameters, and one that finds new parameter values to maximize that probability.

Heuristic: Algorithm that is not guaranteed to find the absolute best (optimal) solution.

k-means algorithm: Clustering algorithm that defines clusters of data points, each corresponding to one of cluster centers selected by the algorithm.

k-Nearest-Neighbor (KNN): Classification algorithm that defines a data point's category as a function of the nearest data points to it.

Kernel: A type of function that computes the similarity between two inputs; thanks to what's (really!) sometimes known as the "kernel trick", nonlinear classifiers can be found almost as easily as linear ones.

Learning: Finding/discovering patterns (or rules) in data, often that can be applied to new data.

Machine: Apparatus that can do something; in "machine learning", it often refers to both an algorithm and the computer it's run on. (Fun fact: before computers were developed, the term "computers" referred to people who did calculations quickly in their heads or on paper!)

Margin: For a single point, the distance between the point and the classification boundary; for a set of points, the minimum distance between a point in the set and the classification boundary. Also called the separation.

Machine learning: Use of computer algorithms to learn and discover patterns or structure in data, without being programmed specifically for them.

Misclassified: Put into the wrong category by a classifier.

Neural network: A machine learning model that itself is modeled after the workings of neurons in the brain.

Supervised learning: Machine learning where the "correct" answer is known for each data point in the training set.

Support vector: In SVM models, the closest point to the classifier, among those in a category. (Note that there is a more-technical mathematical definition too.)

Support vector machine (SVM): Classification algorithm that uses a boundary to separate the data into two or more categories ("classes").

SVM: Support vector machine.

Unsupervised learning: Machine learning where the "correct" answer is not known for the data points in the training set.

Voronoi diagram: Graphical representation of splitting a plane with two or more special points into regions with one special point each, where each region's points are closest to that special point.

[1] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," 2010.
[2] Wikibooks, "Data mining algorithms in r/clustering/k-means." http://en.wikipedia.org/w/index.php?title=K-means%20clustering&oldid=1243054475, 2024.