Practical Applications - A Journey of Self-Study

09/2024 - Overview of Statistical Learning

● Introduction

- **Prediction vs Inference**
  - Prediction = Emphasizes Prediction Accuracy
  - Inference = Emphasizes Model Interpretability
- **Supervised vs Unsupervised Learning**
  - Supervised = Labeled Data w/ Response Variable
    - Ex. Prediction of Response
  - Unsupervised = Unlabeled Data w/ No Response Variable
    - Ex. Clustering, Principal Components Analysis (PCA)
- **Regression vs Classification**
  - Regression = Quantitative (Continuous) Response
  - Classification = Qualitative (Categorical / Discrete) Response
- **Parametric vs Non-Parametric**
  - Parametric = Known Distribution (Only Estimate Params)
    - Assumes correct specification of the model
    - Ex. Linear Regression, Logistic Regression
  - Non-Parametric = Unknown Distribution (No Fixed Params)
    - Ex. K-Nearest Neighbors (KNN)
- **Bias-Variance Trade-Off**
  - Low Flexibility = High Bias, Low Variance
  - High Flexibility = Low Bias, High Variance
  - Goal: Identify the balance between Bias and Variance so that the overall error is minimized (specifically the test error on unseen data)

● Regression

- **Simple Linear Regression**
  - Formula: $Y = \beta 0 + \beta 1X + \epsilon$
  - Assumptions:
    - Linearity: The relationship between X and Y is linear
    - Independence: Observations are independent
    - Homoscedasticity: Constant variance of errors
    - Normality of errors: Residuals (errors) follow a normal distribution
  - Limitations:
    - Fails in non-linear relationships and high-dimensional settings.
- **Multiple Linear Regression**
  - Formula: $Y = \beta 0 + \beta 1X1 + \beta 2X2 + \cdots + \epsilon$
  - Assumptions:
    - Linearity: Linear relationship between predictors and response.
    - Independence: Observations are independent.
    - No multicollinearity: Predictors should not be too highly correlated.

            - Homoscedasticity: Constant variance of residuals.
            - Normality of residuals.
        - Limitations:
            - Struggles in high-dimensional data (i.e., p > n) and non-linear relationships.
    - **K-Nearest Neighbors (KNN)**
        - Lower *k*, Higher Complexity
        - Higher *k*, Lower Complexity

- **Classification**
    - **Logistic Regression**
        - Will always give output probability between 0 and 1
            - Log odds or Logit transformation turns this to a linear model
            - Uses Maximum Likelihood Estimation to estimate parameters
        - Works for Non-Normal Distribution of Observations
        - Discriminative learning (conditional likelihood) to estimate the params
    - **Multinomial Logistic Regression**
        - A linear function is created for each class and weighed against each other using a softmax function
        - Variability of coefficients suffers when classes are well-separated
    - **Bayes Theorem**
    - **Linear Discriminant Analysis (LDA)**
        - Linear Decision Boundary
        - Normal Distribution of Observations
        - Class Specific Mean, Shared Covariance Matrix
        - Additive Fit
        - Generative learning (full likelihood) to estimate the params
    - **Quadratic Discriminant Analysis (QDA)**
        - Non-Linear Decision Boundary
        - Normal Distribution of Observations
        - Class Specific Mean, Class Specific Covariance Matrix
        - Modest *p*, Modest *n*
        - Multiplicative Fit
    - **Naive Bayes**
        - Non-Linear Decision Boundary
        - Independent Predictors
        - Large *p*, Small *n*
        - Additive Fit
    - **K-Nearest Neighbors w/ Cross Validation (KNN-CV)**
        - Non-Parametric
        - Small *p*, Large *n*

- **Generalized Linear Models**
    - **Poisson Regression**

- **Gamma Regression**
- **Negative Binomial Regression**
- **Exponential Family**
  - These each have a characteristic link function, which is the transformation of the mean that is represented by a linear model
  - Each GLE is chosen based on the properties of the response variable
    - Gaussian Distribution
      - Continuous (Modeled by Mean, Variance)
    - Bernoulli Distribution
      - Discrete (Binary Values)
    - Poisson Distribution
      - Counts (Nonnegative Integer Values)
    - Exponential Distribution
      - Time (Events w/ Constant Rate)
    - Gamma Distribution
      - Time (Multiple Events)
    - Negative Binomial Distribution
      - Discrete (Sequence of Independent Bernoulli Trials)

Resampling Methods

- **Cross-Validation**
  - Validation Set
    - High Bias, Low Variance
  - LOOCV
    - Low Bias, High Variance
  - K-Fold CV
    - Balanced Bias & Variance at k = 5,10
- **The Bootstrap**
  - Sampling w/ Replacement

Linear Model Selection

- **Subset Selection**
  - Best Subset Selection
- **Stepwise Selection**
  - Forward Stepwise Selection
  - Backward Stepwise Selection
  - Hybrid Stepwise Selection
- **Metrics for Model Selection**
  - C_p
  - Akaike Information Criterion (AIC)
  - Bayesian Information Criterion (BIC)
    - Heavy penalty on models with many variables
  - Adjusted R^2

        - Validation and Cross Validation
            - SE of MSE
            - One-Standard-Error Rule
    - **Shrinkage (Regularization) Methods**
        - **The Lasso**
            - Performs variable selection of predictors
            - RSS + Shrinking Penalty (towards or equal to 0)
            - L1 Regularization
            - λ Tuning Param
            - Larger λ = Increased Bias, Decreased Variance
            - Yields “Sparse Models”
            - Use when response is a function of few predictors
        - **Ridge Regression**
            - Includes all *p* predictors in the final model
            - RSS + Shrinkage Penalty (towards 0)
            - L2 Regularization
            - λ Tuning Param
            - Larger λ = Increased Bias, Decreased Variance
            - Best used after standardizing the predictors
            - Use when response is a function of many predictors
    - **Dimension Reduction Methods**
        - **Principal Components Analysis (PCA)**
            - One value is assigned to represent multiple variables
            - 1st Principal Component = the direction where observations vary the most OR the line that is as close as possible to the data
                - Minimizes the distance between the projection of each point and its original observation
            - 2nd Principal Component = the linear combination of the variables that is uncorrelated with the 1st Principal Component and has the largest variance subject to this constraint
                - Perpendicular or orthogonal to the 1st Principal Compenent’s direction
        - **Principal Components Regression (PCR)**
            - Construct *M* Principal Components
            - Use these components as predictors in a Linear Regression, fit using Least Squares
            - As the # of *M* increases, Bias decreases and Variance increases
            - Best used after standardizing the predictors
            - Use when few *M* Principal Components capture the majority of the variation in the predictors
        - **Partial Least Squares (PLS)**
            - Supervised alternative to PCR (includes response supervision)
            - Identifies directions for *M* PLS Components that help explain both the response and the predictors

- Prioritizes correlation between Y and Xj as well as variance
- In practice, often performs no better than ridge regression or PCR

- **Considerations in High Dimensions**
  - High-Dimensional Data
    - Data sets containing more features than observations
  - Important Points For High-Dimensional Problems:
    1. Regularization plays a key role in high-dimensional problems
    2. Appropriate tuning-param selection is crucial for good predictive performance
    3. Test error tends to increase as the dimensionality of the problem increases, unless the additional features are truly associated with the response
  - Curse of Dimensionality = The addition of “noise” features will decrease the quality of the prediction
  - Interpreting Results in High Dimensions
    - Risk of multicollinearity is extreme
    - Individual interpretation of predictors is difficult
      - “One of many possible models has been obtained”
  - Never use training set SSE, p-values, R-squared statistics as measures of good model fit in the high-dimensional setting
    - Report results using independent test set or CV errors

● Non-Linear Models

- **Basis Functions**
  - A family of functions or transformations that can be applied to a variable X: b1(X), b2(X),...,bK(X). Instead of fitting a linear model in X, we fit the model yi = β0 + β1b1(xi) + β2b2(xi) + β3b3(xi) + ⋯ + βKbK(xi) + "i.
  - **Polynomial Regression**
    - Extends the least squares or logistic regression to settings where the relationship between the predictors and the response is non-linear\
    - Transformation imposes a global structure
  - **Step Functions**
    - Breaks X into bins, where a continuous variable becomes an ordered categorical variable
    - Includes an indicator function for each step, where 1 is true and 0 otherwise (dummy variables)
    - Also called piecewise-constant regression models
- **Regression Splines**
  - A flexible class of basis functions that extends upon the polynomial regression and piecewise constant regression
  - **Piecewise Polynomials**
    - Fits separate, low-degree polynomials over different regions of X
    - “Knots” = the points where the coefficients change

                - Constraints determine continuity and smoothness
                    - Each constraint frees up one degree of freedom, reducing the overall complexity
                - High variance at the outer range of the predictors (when X is either very small or very large)
                    - Natural Spline = additional boundary constraints where the function is required to be linear at the boundary
            - Regression Splines often give superior results to Polynomial Regression by introducing flexibility at the “knots” without increasing the degree of X
    - **Smoothing Splines**
        - RSS + Penalty Term (penalizes variability)
        - λ Non-Negative Tuning Param
        - The first derivative measures the slope of a function at t, and the second derivative corresponds to the amount by which the slope is changing
        - Shrunken version of a natural cubic spline, where λ controls shrinkage
    - **Local (Weighted) Regression**
        - Computes the fit at a target point x0 using only the nearby training observations (similar to KNN)
        - *span s* = proportion of points used to compute the local regression at x0
            - Smaller *s*, more flexible and “local” the fit
            - Performs poorly when p is greater than 3 or 4 due to sparsity
    - **Generalized Additive Models**
        - Framework for extending a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity
        - Fits non-linear fj to each Xj so that we can model non-linear relationships
        - Smoothness can be summarized via degrees of freedom
        - Restricted to additive models, missing potential interactions (although interaction effects can be added manually)

● Tree-Based Methods

- **Regression Trees**
    - Terminal Nodes == “Leaves”
    - Internal Nodes == Split Points
    - Connections == “Branches”
    - Divide the predictor space into distinct, non-overlapping regions to reduce RSS, then predict the response as the mean of the values in the region
        - **Recursive Binary Splitting**
            - Top-Down, Greedy Approach
            - Identifies the best split at that particular step
    - **Tree Pruning**
        - Without pruning, a decision tree will minimize bias and “overfit”
        - **Cost Complexity Pruning / Weakest Link Pruning**
            - RSS + Penalty Term (penalizes # of terminal nodes)
            - α = Tuning Param

- **Classification Trees**
  - Divide the predictor space into distinct, non-overlapping regions to reduce classification error rate (or more commonly Gini index or entropy), then predict the response as the most commonly occurring class in the region
    - **Gini Index**
      - Measure of total variance across the K classes or “purity”
    - **Entropy**
      - Similar to Gini Index, takes on a small value if the *m*th node is pure
    - **Classification Error Rate** is preferred if prediction accuracy of the final printed tree is the goal
- **Pros (Simple Tree)**
  - Effectively model non-linear relationships
  - Can displayed graphically and easily interpreted
  - Trees can easily handle qualitative predictors without dummy variables
- **Cons (Simple Tree)**
  - Less satisfactory predictive accuracy compared to other models
  - Can be very non-robust (small change in data, large change in tree)
- **Bagging**
  - “Bootstrap Aggregating” reduces the variance of a given method
  - The prediction is performed *B* times using *B* boostrapped training sets
    - The resulting prediction outcomes are averaged (regression) or the majority vote (classification) is taken
    - Each tree is fully grown and not pruned, minimizing bias
  - **Out-of-Bag Error Estimation**
    - Out-of-Bag (OOB) observations are used as a test set
      - Observations not used to fit a bagged tree are OOB, which can be used to predict *i*th observation using each of the models in which that observation was OOB
      - The predictions of those models are averaged (regression) or the majority vote (classification) is taken
    - With B sufficiently large, OOB Error == Leave-One-Out CV
    - Used to obtain OOB MSE (regression) or classification error
  - Bagging improves prediction accuracy at the expense of interpretability
  - Total decrease in RSS over a split for a given predictor averaged over all B trees can show relative importance (Gini index for classification)
- **Random Forests**
  - Improves bagged trees by decorrelating the trees (only a random sample of the *m* predictors are available for each split)
  - *m* = choice of predictor subset size
    - Small *m*, Decreased Variance (especially w/ correlated predictors)
- **Boosting**
  - Trees are grown sequentially using information from previous trees

            - "Learns slowly" using residuals rather than the outcome Y as a response and updating for each iteration
        - *d* = # of splits allowed
        - λ = shrinkage parameter / learning rate
        - *B* = # of trees allowed (Boosting CAN overfit if *B* is too large)
    - **Bayesian Additive Regression Trees (BART)**
        - Combines Bagging with Boosting, building trees from all of the observations and improving iteratively based on experimental perturbations and measuring resulting changes in partial residuals
            - Prediction is made on an average of the after "burn-in" iterations
            - Resistant to overfitting
            - Markov chain Monte Carlo algorithm for fitting the BART model
        - K = # of regression trees
        - B = # of iterations
        - L = # of "burn-in" iterations
    - **Summary of Tree Ensemble Methods**
        - Bagging
            - Trees are grown independently on random samples of obs.
            - Trees tend to be quite similar to each other.
            - Can get caught in local optima and fail to explore model space
        - Random Forests
            - Trees are grown independently on random samples of obs.
            - Each split on each tree is performed using a random subset of features, decorrelating the trees
            - More thorough exploration of model space relative to bagging
        - Boosting
            - Only the original data is used, no random samples drawn
            - Trees are grown successively, using a "slow" learning approach
            - Each new tree is fit to the signal that is left over from previous trees, and shrunken down before it is used
        - BART
            - Only the original data is used, no random samples drawn
            - Trees are grown successively
            - Each tree is perturbed in order to avoid local minima and achieve a more thorough exploration of the model space

● Support Vector Machines

- **Maximal Margin Classifier**
    - Also known as the "Optimal Separating Hyperplane", which is the hyperplane that is the farthest from the training observations
    - Requires data where possible a hyperplane exists that can separate all of the observations into distinct classes
    - **Margin** = distance between the observations to the hyperplane
        - Minimizing margin can lead to overfitting when *p* is large

            - *M* = width of the margin
        - **Support Vectors** = vectors in p-dimensional space (observations) that "support" the maximal margin hyperplane in that if they moved, the maximal margin hyperplane would move as well
            - Can be thought of as observations that "pin down" the hyperplane
            - A movement in any other observations would not affect the separating hyperplane, provided that the observation's movement does not cause it to cross the boundary set by the margin
    - **Support Vector Classifier (SVC)**
        - Also known as the "Soft Margin Classifier", the SVC is more robust to individual observations and best classifies *most* training observations
        - *C* = Nonnegative Tuning Param
            - Bounds the sum of errors and determines the number and severity of the violations to the margin (and hyperplane) tolerated
            - Small *C*, narrow margins (low bias, high variance)
            - Large *C*, wide margins (high bias, low variance)
        - **Support Vectors** = Observations that affect the SVC (lie directly on the margin, or on the wrong side of the margin for their class)
        - Highly resistant to outliers (observations far away from the hyperplane)
    - **Support Vector Machines (SVM)**
        - Extension of the Support Vector Classifier that results from enlarging the feature space using kernels to adapt to non-linearity in the data
            - **Kernel** = function that quantifies the similarity of two observations
        - Simply- combines a Support Vector Classifier with a non-linear kernel
    - **One-Versus-One Classification**
        - Construct (K, 2) SVMs, each of which compares a pair of classes
        - Test observation is assigned to the class where it was most frequently assigned in the pairwise classifications
    - **One-Versus-All Classification**
        - Fit K SVMs, each time comparing one of the K classes to the remaining K - 1 classes
        - Test observation is assigned to the class for which, the parameters that results from fitting an SVM comparing kth class to the others, is largest
    - **Relationship to Logistic Regression**
        - When classes are well separated SVM behaves better
        - When classes have overlapping regimes, logistic regression is preferred

● Deep Learning

- **Single Layer Neural Networks**
    - Features in the input layer pass through K hidden units and are transformed by a non-linear transformation function
    - "Weights" are coefficients, and "Bias" are the intercepts
        - **Sigmoid** = also used in Logistic Regression, this function converts a linear function into probabilities between zero and one

            - **Rectified Linear Unit (ReLU)** = piecewise function with an inflection point at zero, where z < 0 is equal to 0 and z otherwise
        - Values of activations close to 1 are “firing”, values close to 0 are “silent”
        - Sum of nonlinear transformations of linear functions can yield interactions
        - To train the network, we minimize squared error loss
    - **Multilayer Neural Networks**
        - Multiple layers of modest size with K hidden units in each layer makes the learning task of discovering good solutions much easier
            - **One-Hot Encoding** = method of converting categorical variables into binary vectors, where each category is represented by a vector with a single 1 and all other positions set to 0
            - **Softmax** = activation function that ensures outputs behave like probabilities (non-negative and sum to one)
        - To train the network, we minimize the negative multinomial log-likelihood (also known as cross-entropy) if the response is qualitative, squared error loss if the response is quantitative
    - **Convolutional Neural Networks (CNN)**
        - The network first identifies low-level features in the input image, which are combined to form higher-level features, which are used to determine (in the presence or absence of these higher-level features) the probability of a given output class
        - **Convolution Layers**
            - Made up of *convolution filters*, the coefficients of a CNN, each of which is a template that determines whether a particular local feature is present in an image through an operation called a *convolution* (multiplying matrix elements and adding the results)
            - If the *convolution filter* (submatrix) matches the original image, then it will have a large value in the *convolved* image
            - K different convolution filters output K two-dimensional output feature maps, which together are treated a single three-dimensional feature map
        - **Detector Layers**
            - The layer where the ReLU activation function is applied to the convolved image to introduce non-linearity
        - **Pooling Layers**
            - Provides a way to condense a large image into a smaller “summary image”
            - **Max Pooling** = Summarizes each non-overlapping 2x2 block of pixels in an image using the maximum value in the block
        - **Fully Connected Layers**
            - Once the convolution and pooling layers have been repeated so that the feature map is down to just a few pixels in each dimension, the three-dimensional feature maps are flattened

(pixels are treated as separate units) and passed into the the fully connected layers as inputs
        - Within the output layer, the inputs are passed through a softmax activation function and return interpretable probabilities
    - **Data Augmentation**
        - Each training example is replicated many times, each with a replicate randomly distorted in a natural way (zoom, shift, rotate)
            - Increases the training dataset to protect against overfitting
            - Form of regularization (similar to ridge regularization)
- **Document Classification**
    - **Bag-of-Words** = each document is scored based on the presence or absence of each word in a language dictionary- if the dictionary contains M words, then each document has a binary feature vector of length M and scores a 1 for every word present and 0 otherwise
        - This process stores outputs in a **Sparse Matrix Format**
    - **Bag-of-n-Grams** = Records the co-occurence of words in relationship to one another, treating the document as a sequence
    - **Embedding** = each word is 'embedded' as a vector of numerical values that are typically not 0, reducing the vector length and sparsity problem
- **Recurrent Neural Networks (RNN)**
    - The input object X is a sequence where the order and closeness of certain inputs in a sequence convey meaning, much like how CNNs accommodate the spatial structure of image inputs
    - The output Y can also be a sequence, but often is a scalar such as a binary sentiment label
        - As the sequence is processed one vector X(L) at a time, the network updates the activations A(L) in the hidden layer, taking as input the vector X(L) and the activation vector A(L-1) from the previous step
        - Each A(L) feeds into the output layer and produces a prediction O(L) for Y, where the last O(L) is the most relevant
- **Fitting a Neural Network**
    - **Slow Learning** = the model is fit in a slow, iterative fashion, using gradient descent where the process is stopped if overfitting is detected
    - **Regularization** = penalties are imposed on the parameters, which are commonly lasso or ridge regularization techniques
    - **Backpropagation** = during gradient descent, the act of differentiation assigns a fraction of the residual to each parameter via the chain rule
    - **Stochastic Gradient Descent (SGD)** = instead of summing over all *n* observations when *n* is large, we can sample a small fraction or minibatch of them each time we compute a gradient step
    - **Early Stopping** = additional form of regularization to prevent the validation objective from increasing after a number of epochs

        - **Dropout Learning** = relatively new and efficient form of regularization, it removes a fraction of φ, the units in a layer when fitting the model, where surviving weights are scaled by by a factor of 1/(1-φ) to compensate
        - **Fine Tuning**
            - # of Hidden Layers
            - # of Units per Layer
            - Regularization Tuning Params
                - Dropout Rate Param φ
                - Lasso/ Ridge Regularization Param λ
            - Stochastic Gradient Descent Params
                - Batch Size
                - # of Epochs
                - Data Augmentation

Survival Analysis and Censored Data

- Analysis of a unique kind of outcome variable: the time until an event occurs
    - Y = min (T , C) ←- random variable
    - δ = 1 if T is less than or equal to C and 0 otherwise (T > C)
        - δ = 1 if we observe true survival time
        - δ = 0 if we observe the censoring time
- **Survival Time** = also known as *failure time* or *event time*, survival time represents the time at which the event of interest occurs ex. The time at which a patient dies or a customer cancels their subscription
- **Censoring Time** = the time at which censoring occurs, ex. the time at which a patient drops out of a study or the study ends
- We need to assume that the censoring mechanism is independent: conditional on the features, the event time T is independent of the censoring time C
    - **Right Censoring** = T is greater than or equal to Y
    - **Left Censoring** = T is less than or equal to Y
    - **Interval Censoring** = exact event time is unknown, falls within an interval
- **The Kaplan-Meier Survival Curve**
    - **Survival Curve / Survival Function** = S(t) = Pr(T > t)
        - Decreasing function calculates probability of surviving past time t
        - Step-like shape that starts at zero and maps out the observed events as they unfold in time, considering censoring
- **The Log-Rank Test**
    - Examines how the events in each group unfold sequentially in time
- **Regression Models w/ a Survival Response**
    - Hazard Functions
    - Proportional Hazards
    - Cox's Proportional Hazards Model

Unsupervised Learning

- During unsupervised learning, we are not interested in prediction because we do not have an associated response variable Y
- **Principal Components Analysis (PCA)**
  - The process by which principal components are computed, and the subsequent use of these components in understanding the data
  - Finds a low-dimensional representation of a data set that contains as much as possible of the variation
  - **Principal Components** = the normalized linear combination of the features that have the largest variance, where a *loading* constraint ensures that the sum of squares is equal to one (consistent variance)
    - **Principal Component Scores** = data projected on loading vector
    - **Principal Component Loadings** = vector pointing to the max(var)
  - Principal components provide low-dimensional linear surfaces that are closest to the observations, providing a good summary of the data
  - **Proportion of Variance Explained (PVE)** = how much variance in the data can be explained by each principal component out of total variance, mathematically similar to the R-squared interpretation for the approximation for X given by the first M principal components
  - Each variable must be scaled to have a mean of 0 and stdev of 1 (unless all variables are measured in the same units)
    - **Matrix Completion** = used to estimate random missing values
- **Clustering**
  - Partitions observations of a data set into distinct groups so that observations within each group are quite similar to each other, while observations in different groups are quite different from each other
  - **K-Means Clustering**
    - Partitions the data set into K distinct, non-overlapping clusters, where K is specified in advanced
    - Minimizes *within-cluster variation* using *squared Euclidean distance* to determine the distance between all pairwise observations summed over all pairs
      - Assign K random clusters and a cluster *centroid*, then assign each observation to the cluster whose centroid is closest using Euclidean distance
    - Because the K-means algorithm finds a local rather than a global optimum based on the initial (random) cluster assignment, it is important to run the algorithm multiple times from different random initial configurations to determine the best clusters
  - **Hierarchical Clustering**
    - Alternative approach which does not require that we commit to a particular choice of K in advanced, instead creating a *dendrogram* based on the observations to determine the number of clusters
    - **Bottom-Up / Agglomerative Clustering** = a dendrogram is build starting from the leaves and combining clusters up the trunk

        - The hierarchical dendrogram assumes that clusters obtained at one height are nested within clusters at a greater height, although the best results do not always come from breaking into the best 2 groups, and then splitting those groups further (in which case K-Means may be a better alternative)
        - **Linkage** = defines the dissimilarity between two groups of observations
            - **Complete** = maximal intercluster dissimilarity
            - **Average** = mean intercluster dissimilarity
            - **Single** = minimal intercluster dissimilarity
            - **Centroid** = dissimilarity between centroids
        - **Measures of Dissimilarity**
            - **Euclidean Distance** = groups based on distance between obs
            - **Correlation-Based Distance** = groups based on similarity
            - It is also important to consider scaling variables so that "rarer" observations have greater weight in the computations

- # Multiple Testing
    - **Testing a Hypothesis**
        1. Define the null and alternative hypotheses
        2. Construct a test statistic (*T*) that summarizes the strength of evidence against the null hypothesis (based on the theoretical null distribution)
        3. Compute a p-value that quantifies the probability of having obtained a comparable or larger value of the test statistic under the null hypothesis
        4. Based on the p-value, decide whether to reject the null hypothesis
        - **Type I Error Rate** = probability of incorrectly rejecting the H_0
        - **Type II Error Rate** = probability of incorrectly failing to reject H_0
        - **Power** = probability of not making a Type II error given that H_a holds, ie. the probability of correctly rejecting H_0
    - **Family-Wise Error Rate (FWER)**
        - With *m* null hypothesis, FWER determines the probability of making at least one Type I error
        - **The Bonferroni Method**
            - Ensures that we have not falsely rejected too many null hypotheses, but for a price: we reject few null hypotheses, and thus will typically make quite a few Type II errors
        - **Holm's Step-Down Procedure**
            - Controls the FWER, but it is less conservative than Bonferroni, in the sense that it will reject more null hypotheses, typically resulting in fewer Type II errors and hence greater power
            - Makes no independence assumptions about the m hypothesis tests, and is uniformly more powerful than the Bonferroni method
    - **The False Discovery Rate (FDR)**

        - As FWER is controlled and *m* increases, Power decreases significantly.
        - **False Discovery Proportion** = reduce the ratio of false positives (V) to total positives (V + S = R) so that it is sufficiently low, so that most of the rejected null hypothesis are not false positives
            - It is not possible to guarantee a False Discovery Proportion, but we can estimate the FDR, which is the proportion of false positives on average
        - **The Benjamini-Hochberg Procedure**
            - Guarantees that, on average, no more than a fraction q of the rejected null hypotheses are false positives (give m p-values are independent or mildly dependent)
            - Rejects based on an ordered list of p-values, where the threshold is determined by the data
    - **Re-Sampling Approach**
        - If no theoretical null distribution is available due to an unusual null hypothesis H_0 or using an unusual test statistic T
        - If there is a theoretical null distribution available, but assumptions required for its validity do not hold

10/2024 - Applied Statistics

11/2024 - Mathematics for Data Science

12/2024 - Applied Machine Learning

01/2025 - SQL & Database Fundamentals

02/2025 - Advanced SQL

03/2025 - ETL Pipelines

04/2025 - Introduction to Programming

05/2025 - Data Structures & Algorithms

06/2025 - Advanced Programming