**MGMT 590: Using R for Analytics**

**R-Shiny DSS App Final Project**

**Objective**

Over the term you have developed the ability to interpret and modify R code, build shiny applications, and apply R libraries to perform descriptive, predictive, and prescriptive analytics. Your final hands-on activity in this course is to develop a shiny application with specificized requirements that ties all the learning components together.

**Problem and Data**

Your class team is required to create a shiny application that can be used to help identify manufacturing process settings, suppliers, or customer window specifications that can help reduce the window breakage rate. You are using the provided data to help create a decision-support system (DSS) prototype that will help you remember after the course is over how descriptive, predictive, and prescriptive analytics could be integrated together to support decision-making. Also, to show how you used popular R libraries and functions to do something useful.

A DSS usually contains three components (1) a data component such as database or data file, (2) a model component that helps to support the problem, and (3) a user interface (UI) component that allows the decision-maker to interact with the model or data, such as specifying parameters, and doing some specific task(s). You will develop a shiny app using the *Window\_Manufacturing.xlsx* data to provide the window manufacturing technician end-user, insights or actionable recommendations about their problem via interactive data analytics.

**Video Presentation**

Every team must present a short demonstration of their DSS via a recorded video that discusses the business problem, the need for analytics support, and demonstrate the functionality of the app to support the problem (items you address below in the project requirements rubric). You may use one of the 12 available [video express studios](https://mediaspace.itap.purdue.edu/media/Video+Express+101/1_sz5hnwg1/) on campus to create a video, but you do not have to. You can schedule one these rooms (BRNG 2294, BRNG 2296, HAMP 4162, KRAN 774, MRGN 233, RAWL B098, STEW G53, STEW G62A, WALC B059, YONG 529) [here](https://videoexpress.purdue.edu/Portal/Login). I imagine most of you will use Zoom or similar conferencing software to create your video, which is fine.

**Due Date**

Only one team member should submit your team’s work in the Purdue Box folder that is assigned to your team by **Saturday, October 14th, 2023 by 11:59pm EST.** In the folder you will have (1) *Window\_Manufacturing.xlsx*, (2) finalized code(s), and (3) your video.

**Peer Evaluations**

By Saturday night when the project is due, every student must also upload their peer evaluation to Brightspace. There you will provide scores for yourself on homework assignments and the final project. You need not be the best performer to receive 10/10. Rather you just need to make an honest and fairly equal share of the workload, and do what you said you would do. This information will be confidential and will not be shared with your teammates. Do not complete peer evaluations together as a team, nor share the scores you give with your teammate(s) if asked. This will be considered an academic integrity violation and will be reported.

**Project Requirements & Grading Rubric**

Read the grading rubric below carefully as it shows exactly what I am looking for and how points are associated to those items. Your DSS prototype need not be sophisticated. However, teams that provide functionality that exceeds others or have higher quality presentations will be ranked higher than other teams.

R cheat sheets you might useful refreshers can be found [here](https://www.rstudio.com/resources/cheatsheets/).

|  |  |
| --- | --- |
| **Business Problem Framing** | Points |
| On the shiny app, provide a tab that has a short paragraph describing the made-up problem scenario.   * A clear and concise description of the business problem * Identification of key stakeholder app end-user * Refinement of the problem to identify any possible constraints and how you’ve addressed those possibly via assumptions made * Define the initial set of business benefits (e.g., tool is expected to improve efficiency by x%, increase sales by y%, etc.) | 4 |
| **Analytics Problem Framing** |  |
| In the same shiny tab as the one where you framed the problem, create a new section called ‘Analytics Problem Framing’ that describes how you decomposed the business scenario into and decision supported design using descriptive, predictive, and prescriptive analytics methods.   * Reformulate problem statement as an analytics problem (e.g., Predicted sales is a regression-type problem, thus we used linear regression and estimated a model using the caret library). * Develop a proposed set of drivers and relationships to outputs (e.g., name at least one relationship the end-user can expect from your data from previous relationships). * State the key set of assumptions related to the problem * Define key metrics of success | 4 |
| **Data** |  |
| In the same shiny tab as the one where you framed the problem, create a new section called ‘Data’ that describes the data used in your app.   * A link provided where the data came from. If data was artificially created to prototype your scenario say that. * A data dictionary which the following columns (variable name, R data type, short variable description) | 2 |
| **Functions Used in Shiny Application** |  |
| To ensure you apply the library and functional learnings we covered, you must incorporate these somewhere into your code:   1. One of is., as., class(), or str() functions      1. Use of a data.frame() or data.table() √      1. Use of a list()      1. Use of at least one of read.table() or readxl library. √      1. Creation and use of your own custom function using function() and source()      1. Use of conditional logic somewhere in your code (e.g., else, else if, ifelse, etc.)      1. Use of at least one for() or while() loop        1. Use of at least one apply family of functions      1. Use of one among aggregate(), plyr library, dplyr library, merge(), or sqldf library      1. Use of a ‘Go’ button to demonstrate non-reactivity in your shiny app | 10 |
| **Descriptive Analytics** |  |
| Create a shiny tab in your app called ‘Descriptive Analytics’. Your app should provide the following functionality:   * Use of ggplot2 library to create at least 4 unique and visually appealing visuals that highlight the key relationships in your data | At most 20 (ranked) |
| **Predictive Analytics** |  |
| Create a shiny tab in your app called ‘Predictive Analytics’. Your app should provide the following functionality:   * Use of lm(), glm(), caret library, or h2o library to estimate a linear regression or logistic regression model (determined based on your problem-type). * Have a table generated that provides the end-user the estimated parameter coefficients and their estimate p-values * Provide the end-user a figure that provides some model fit evaluation statistics based on the training data and testing data | At most 20 (ranked) |
| **Prescriptive Analytics** |  |
| Create a shiny tab in your app called ‘Prescriptive Analytics’. Your app should provide the following functionality:   * Take your final predictive model and use this as the objective function of an optimization model. Here you will either try to maximize or minimize your function based on what makes sense. * Identify which decision variables in your objective function the end-user would have control over and which ones they would not. For those that are non-controllable (e.g., weather/temperature), have the end-user set values for them using sliders. For those that are controllable (e.g., 10% off coupon), your model will solve for those. * Add constraints to your optimization model that make business/problem sense. For example, sometimes modelers will add constraints where the minimize and maximum values that the decision variables could take are no smaller or larger than what is realized in the original data set used (to keep them within scope). * Show the optimal actions to take to the end-user. | At most 20 (ranked) |
| **Video Presentation** |  |
| Create a video presentation of your app following these specifications:   * Total video length does not exceed 6 minutes. * Well-balanced presentation time among team members (e.g., no one person do most of the presenting) * Professional demeanor * Appears as though it was a well-prepared demonstration of app. * Focuses on the business problem and app’s functionality to support decisions and avoids discussing or showing the code * Discusses each section of Descriptive, Predictive, and Prescriptive Analytics * Does the tool work without errors? | At most 20 (ranked) |
| **Total Score** | / 100 |