**Mount Rainier Report**

1. **Intro**

This report will include data and information regarding Mount Rainier and will be used to answer a user story. The data utilized for answering the user story consists of 4045 rows and 13 columns all related to Mount Rainier climbing routes and weather data. These ranged from the climbing routes available, date the climb or weather was recorded, if the route was successfully completed, and many weather statistics such as average temperature, average humidity, average wind speed, average wind direction, and average solar radiation.

The question we are tasked with is, “Client would like an interactive dashboard of success rate prediction for the next three months with weather as a factor”. In this report I will be discussing more in-depthly in the following sections about the data, methods used, conclusions which can be drawn, and what can and should be performed in the future. Unfortunately with the time available, I was unable to create a working model and dashboard and I will discuss some of my issues and what has been done up to that point.

1. **Body**

**Data:**

To start off with this user story I initially gathered all of the data from two separate csv files for inspection. I needed to look at the overall data and determine which pieces of data will be most relevant to the story, and to prepare the data set for creating a model for predicting success rate with weather as a factor. After creating the data frames for both the climbing statistics and weather csv files I noticed some immediate data cleaning could be utilized. First off, I checked for any missing values in the data frame and luckily there were none to deal with at this point. I then cleaned the climbing statistics csv and there were some issues in the “Route” column with some naming conventions. One of the routes was marked “Unknown” and needed to be removed in order for better accuracy of a model to be used. Additionally, one of the routes named “Fuhrers Finger” was spelt in two different ways (“Fuhrers Finger” and “Fuhrer’s Finger”) and to make naming conventions similar across the entire data set I removed all apostrophes from the column as this was the only occurrence.

The next step in the cleaning process of this data frame was to remove rows which will not be needed for the model. There are two columns which I decided on removing and these were “Attempts” and “Success” the reason being is there was also a “Success.Percentage” column which utilized both of these columns and was already normalized. The last important part to cleaning this data frame was to rename the “Date” column, as this did not correctly align with the “Date” columns naming convention in the weather statistics data frame. The column was written as “ï..Date”, and in the weather statistics data frame it was written as “Data” so I simply renamed it to that in the climbing statistics data frame.

Eventually the goal is to merge the two data frames in order to use the weather statistics as a factor to predict the success rate of a certain route. Prior to the merge I needed to view the data in the weather statistics csv in order to determine if the data was ready or missing any values, which was not the case. I will discuss the methods used for merging and some additional preparations in order to create the model in the next section.

**Method:**

Now it was time to merge the two data frames in order to create one data frame which I will refer to as “climbing”. Since I renamed the “Date” column in both of the previous data frames I was able to join on the “Date” column in order to have weather statistics for each day a route attempt was recorded. I utilized the setDT method and data.table to perform this along with naming all of the necessary weather columns. Once the two tables were merged I once again checked for any missing values, this time I noticed there were 2151 missing values in the columns which came from the weather data frame. Since this was just over half of the values in the entire merged data frame (2151/4045) I knew simply dropping all of these rows would cause issues further on when trying to create a strong prediction model. I decided to utilize a library called “imputeTS” and a package named na\_interpolation in order to fill the missing values using a linear method.

The main reason I utilized this method was based upon the data available and the end goal of creating a prediction model that a Time Series model would be most appropriate. This package works well with this kind of model based upon research to solve this problem. One thing to note was I would try additional packages from “imputeTS” to fill values if the models prediction accuracy was weak, and at last resort try to create a model after dropping all of the missing values. The last step prior to creating a working Time Series model was to separate all of the unique routes in order to create a model for each of the routes. There were a total of 24 unique routes which needed to be assigned to their own unique variable. After doing so, the “Route” column from each of these tables could then be removed as this was no longer needed for creating the model. Now we are left with the Date, Success Percentage, and Weather Factors for each unique route.

The next step would be to create a working time series model, but based upon the time available to me I was unable to get to this point so I will then discuss the steps which need to still be taken in the next section to complete this user story.

**Next Steps:**

Now that all of the data is prepared and ready for analysis it would be time to create a working model to predict the success rate of a route by using weather as a factor for the next three months. I believe based upon the data available the most applicable method would be a Time Series Model. I tried to create my own models but unfortunately with the time available paired with limited experience with this method I was unable to create a strong enough model to predict the weather. Things that may be able to help with this issue can be to perform a factor analysis on all of the weather factors in the data set to see if some of these can be removed to increase performance. Like previously mentioned, another possibility is to use another method from the “imputeTS” library or one that performs similarly to fill in the missing weather values.

Once a proper solution is found for creating the Time Series Model, you can then apply the model to each of the 24 routes. Once this step is done, all that is left is to create a dashboard utilizing Shiny. My plan if I got to this point would be to use an image of Mount Rainier with all of the route’s being shown, and to have whichever route was selected by the user be the only route which was highlighted on the image. However, this may be an unnecessary step to solve the overall user story, but would be a very strong visualization for the client and users. Additionally, with the dashboard the user should be able to select the date which they plan to attempt a certain route and utilizing the time series model created it will then output a predicted success rate for the user.

1. **Conclusion**

To conclude the report for attempting to create a Time Series model in order to predict the success rate of a given route with weather statistics as a factor for the next three months I will briefly discuss what has been done, what needs to happen, and some limitations to the data. The data was firstly cleaned by checking for missing values, fixing naming conventions, and removing unnecessary or irrelevant rows and columns. Once this was done I was able to merge the data on the “Date” column in order to have proper weather statistics for each Route’s attempt. I then utilized the na\_interpolation package from the “imputeTS” library to fill in missing values from the merged data set from the weather columns, using the linear method. This was done as too much critical data would be unavailable for creating a working model. The last step in preparing this data for the models creation was to separate each unique route into its own variable in order to perform the analysis properly.

The next steps which need to be taken include creating the working Time Series model if the next person to take on this user story feels this is the most applicable method. After extensive research I also believe a multiple logistic regression model may be used, but I still believe that a Time Series model would be the most efficient for the problem in question. After the working model is created, you can then apply the model to each of the 24 unique routes to determine a predicted success rate of the route, on a given date within the next three months. After this is done, the only thing left is to create a dashboard utilizing Shiny that can give a predicted success rate to the user on a given date for any of the unique routes.

After working with this data I believe there are some innate limitations to the models accuracy. While there are 4045 total observations for this data, many of the occurrences are on the same day but for a different route, meaning there is not an extensive amount of total days recorded in the data. The data spans across a two year period, however, not every single day was recorded so if the data was taken every day for a longer period of time the result of the model can perform much stronger in the end. Since this problem has no real solution unless more data can be found, there may be an inevitable issue on creating a strong model for prediction. However, by using a method in which I did not try the models accuracy could be strong enough for the user in the end. I do not intend to make this seem far-fetched, these were just some observations I noticed with my limited time on the user story along with my limited experience in this particular area of expertise.