Will It Rain in Seattle?

DSC 630 Final Project

Isabella Sturm

2021 November 18

Abstract

Seattle, the Emerald City, surrounded by lush green trees is also known to be a very rainy city, raining nearly half the days in the year. Despite the summer months being warm and sunny, void of intense humidity, there tends to be a negative associated of Seattle with rain. There is a decent wave of tourism in the summer, but the question is whether tourists (and residents) of Seattle really be limiting themselves to activities only in the summer.

Realistically, the rainy days in Seattle are not full days of rain or consisting of heavy rain. The chances of being able to enjoy a cold fall or winter day outdoors in Seattle is probable. Using Seattle weather data from Kaggle, I trained several models to predict whether or not it will rain in Seattle.

Of the four models I trained and evaluated, the Logistic Regression model gave me the best accuracy, of 76.35%. The random forest and k-neighbors classifiers gave me very close results to the logistic regression model, resulting in accuracies of 75.65% and 74.58% respectively. The decision tree model gave the lowest accuracy of 69.35%. Moving forward with predictions I would use the logistic regression model, but all models evaluated gave very fair and comparable results.

Background

Situated between Puget Sound and Lake Washington, Seattle is a city with a reputation for being rainy. Often, when people ask Seattleites about the weather, they expect to hear rain. While the reality is that there are more rainy days in Seattle compared to other cities, averaging at 152 rainy days per year, it is also the reality that roughly only five days per year produce more than an inch of precipitation.

From personal conversations with friends, even with little rain fall per rainy day, people will still avoid outdoor activities or even flee the city for sunnier weather if given the chance.

Using historical precipitation data from Seattle dating back to 1948, the goal of this predictive analytics project is to predict a rainy day in Seattle so both tourists and people who call Seattle home can plan for their city, beach, or mountain activities free of rain. At the very least, expecting rain may allow for a change in mindset approaching a rainy day, opposed to plans being ruined by one.

Using both R (RStudio) and Python (Jupyter Notebook), I am going to analyze data from Kaggle and training four different classification models to predict if it will rain in Seattle. RStudio is used to perform the exploratory data analysis, and Python is used to further prepare the data for analysis and create and train the models.

Methods

Exploratory Analysis/Data Preparation

Exploratory analysis was performed using to get an understanding of the data being used and what may need to be done to clean and further prepare the data for analysis. The dataset being used from Kaggle, is a relatively clean data set with few missing or inconsistent values.

With that said, as part of the data preparation step, I calculated a few new columns from the Date field to get the Day of the Week, Week in the Year, Month of Year, Day of Month, and Day of Year for use in data analysis.

Once adding the new variables to the data frame, next was to explore the dataset further with summary statistics and graphs. Primarily, I created histograms and bar charts to get a better understanding of the data and potential factors for determining if it will rain. I will discuss this more in the Conclusion section under Preliminary/Exploratory Discussion.

After the data was cleaned and some preliminary analysis was done to understand the data, I exported the results to a new CSV to be used in further analysis and modelling in Python.

Modeling

For the dataset I am using, the target variable is a binary (TRUE/FALSE) variable, leading me to choose classification models. There are four classifiers I have decided to train in order to find a good model for the dataset: Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest.

In Python, I loaded the data from the new CSV generated from R, and I split the data into testing and training datasets using a 66 (train):33 (test) split. Then using sklearn, I trained each classifier, returned the accuracy, and created a confusion matrix of the results for each one.

Evaluation & Deployment

Using the accuracy metric (or the built-in classifier score), I determined the model I would deploy is the Logistic Regression.

Classifier	**Accuracy**
Logistic Regression	0.7635418171313014
K-Nearest Neighbors	0.7457749530550339
Decision Tree	0.6934854831720353
Random Forest	0.7564639607106746

If I were to deploy this model, I would need to get live data to run through the model and make regular predictions on the data. Deploying the model would also require maintenance of the model by tuning parts of the model, such as the features, to keep the model making valid and accurate predictions over time.

Results

Preliminary/Exploratory Results

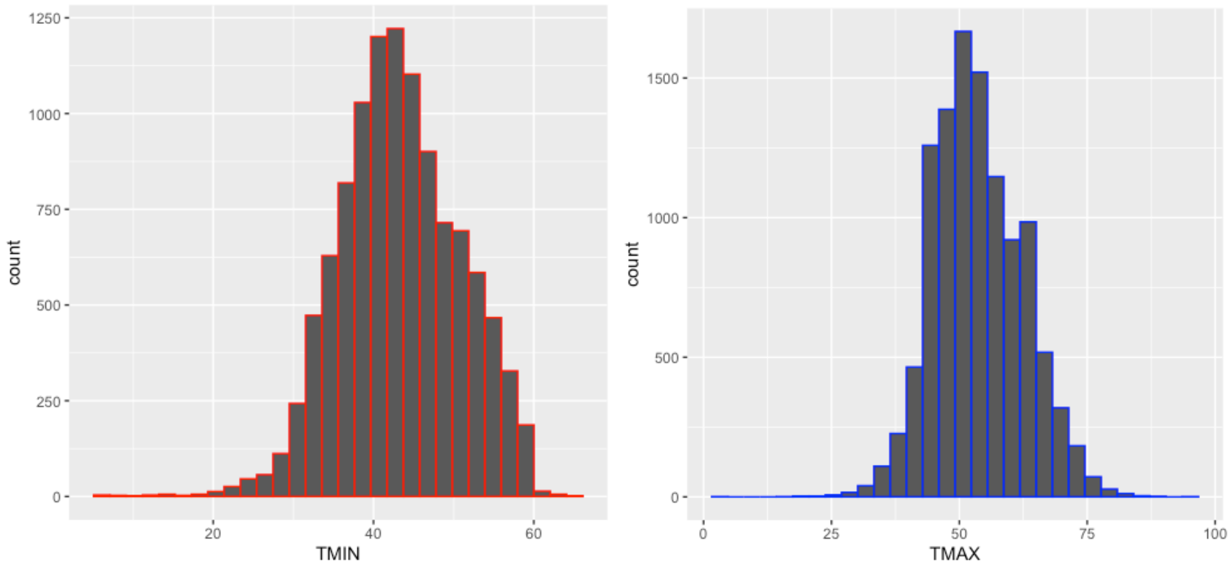

Figure 1 Histograms of Minimum and Maximum temperature for days it rained.

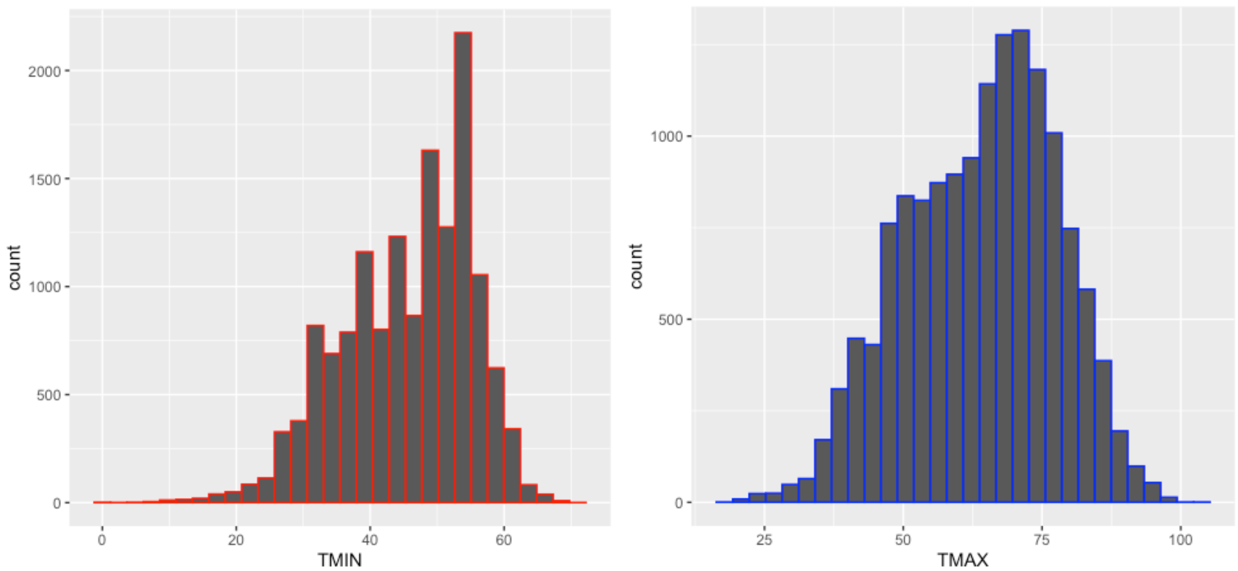

Figure 2 Histograms of Minimum and Maximum Temperatures for days it didn't rain.

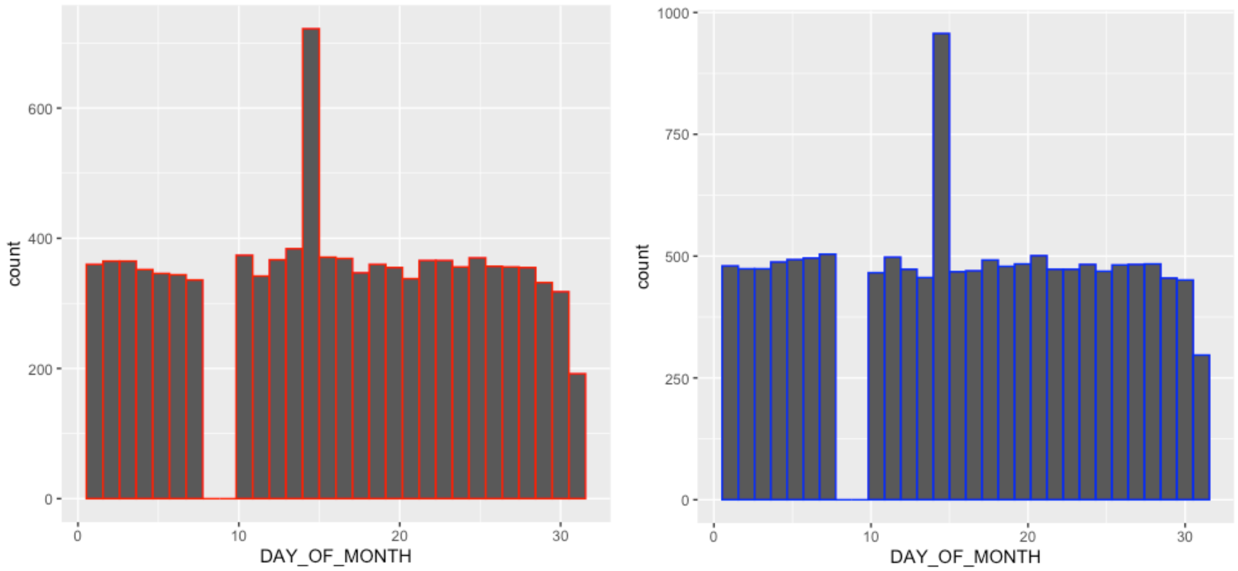

Figure 3 Historgrams of the days of the month. On the left, it represents days it rained. On the right, it represents days it didn't rain.

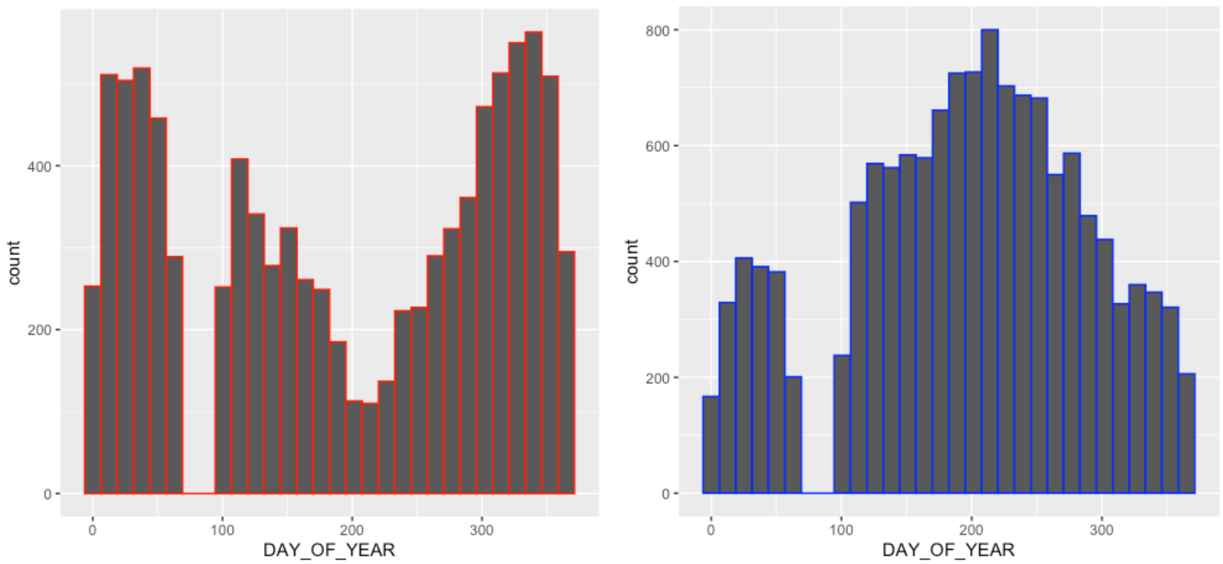

Figure 4 Histograms of the Day of the Year for rain. Left: There was rain. Right: There was no rain.

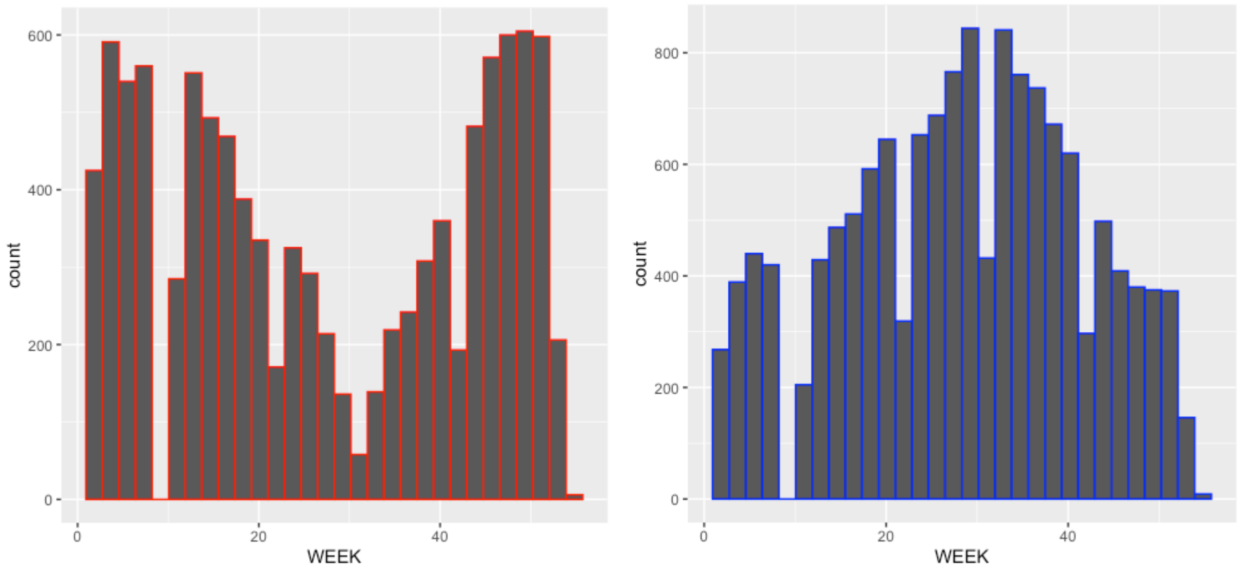

Figure 5 On the left, a historygram of the week in the year there was rain. On the right, a histogram of the week in the year there wasn't rain.

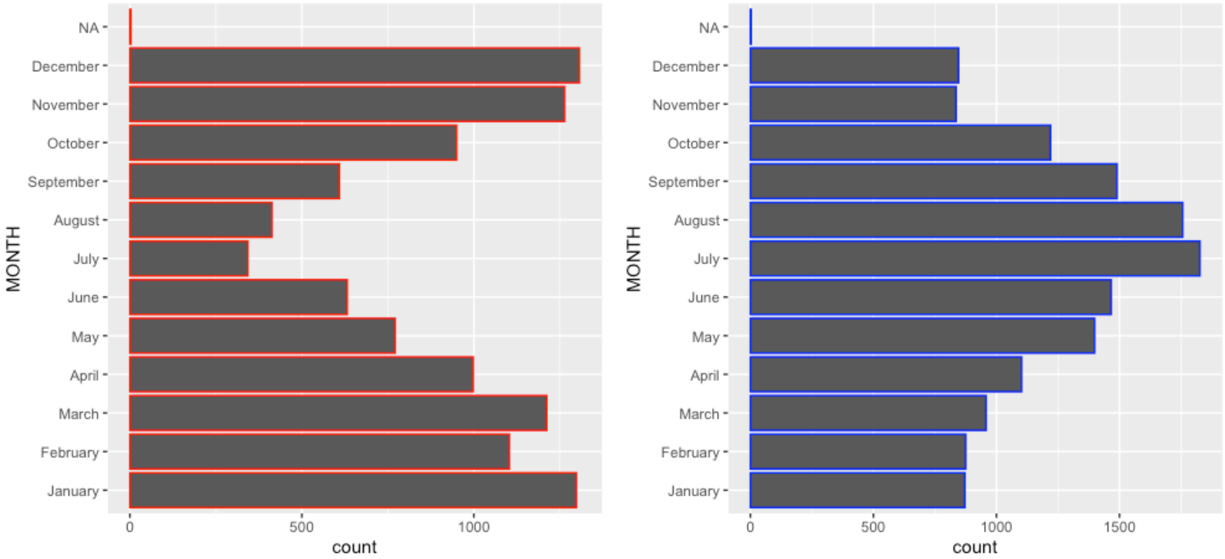

Figure 6 Bar charts showing counts of days by Month there was rain (left) and no rain (right).

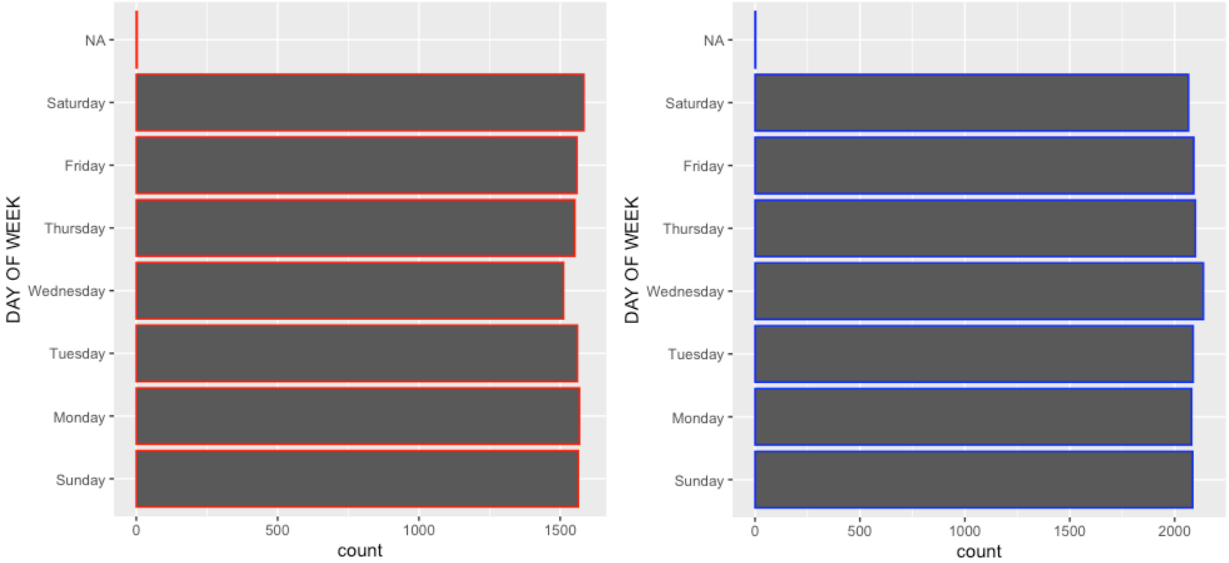

Figure 7 Bar charts showing counts of days by Day of the Week there was rain (left) and no rain (right).

Final Results

Logistic Regression – Deployment Model

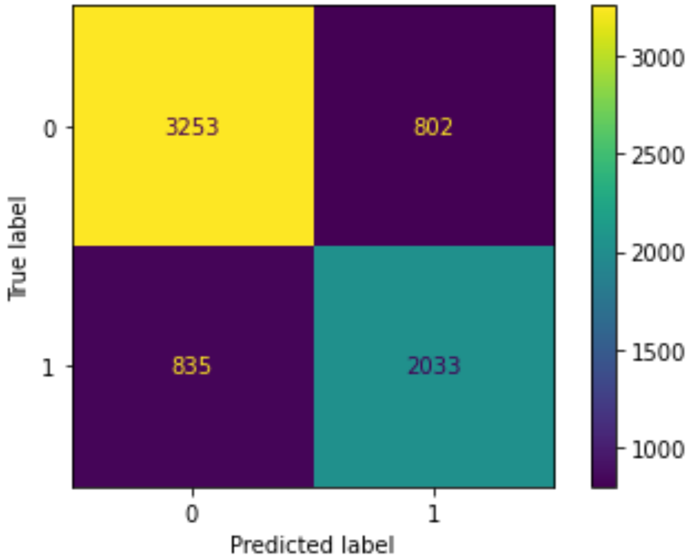

Figure 8 Confusion Matrix for Logistic Regression Model, Accuracy: 0.7635418171313014

Random Forest

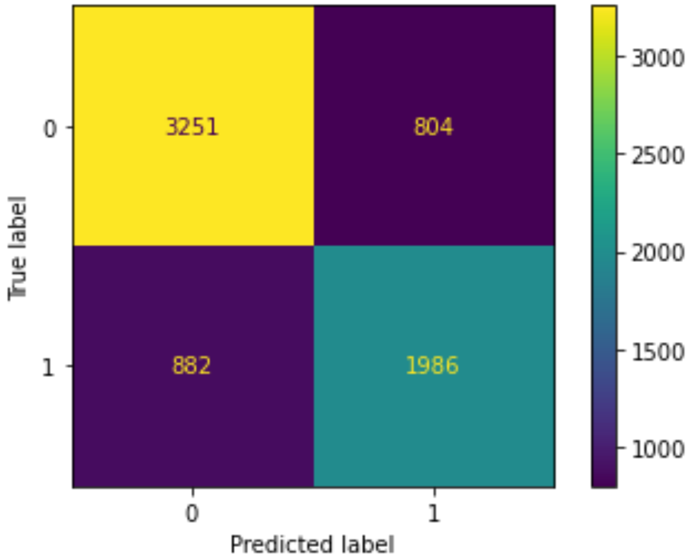

Figure 9 Confusion Matrix for Random Forest Classifier, Accuracy: 0.7564639607106746

K-Nearest Neighbors

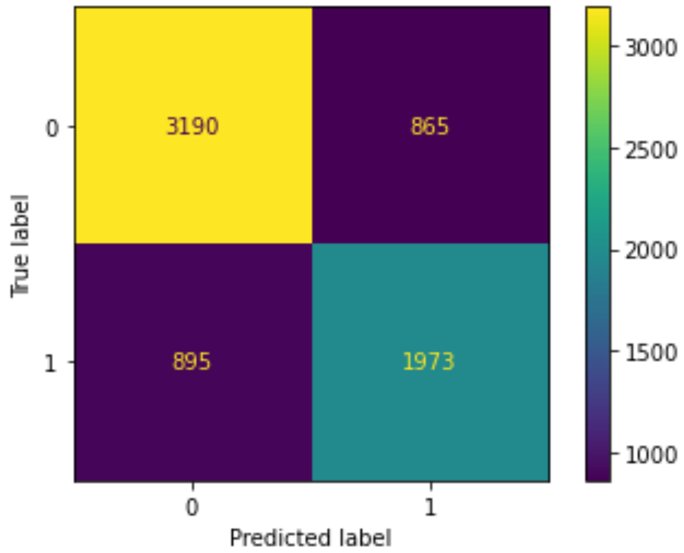

Figure 10 Confusion Matrix for KNN Classifier, Accuracy: 0.7457749530550339

Decision Tree

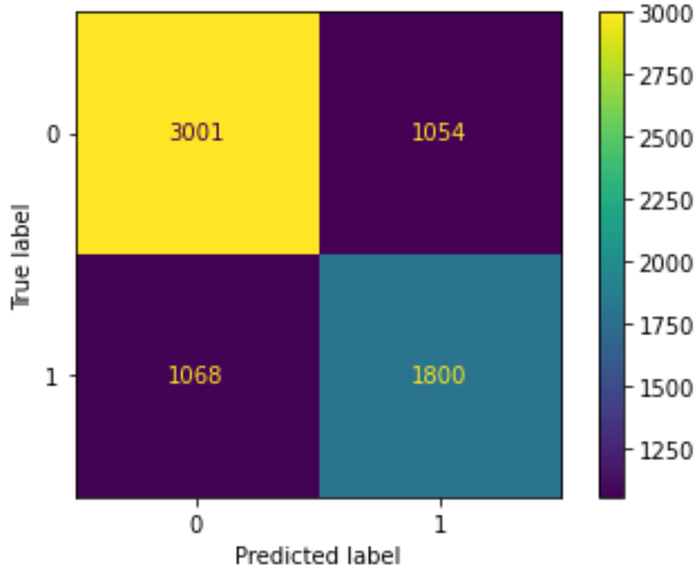

Figure 11 Confusion Matrix for Decision Tree Classifier, Accuracy: 0.6934854831720353

Conclusion

Preliminary/Exploratory Discussion

Based on the preliminary analysis of the dataset, there are a few conclusions that will drive the next steps in the process.

First, Day of Week and Day of Month do not seem to have any significance in determining whether or not it will rain in Seattle.

Second, Month, Week and Day of Year seem like they could be used to determine whether or not it will rain in Seattle. Based on the histograms and bar charts, it seems like the charts for when it will rain are nearly inverses of the ones for when it did not rain. This suggests there is a distinct time of year when it is most likely to rain in Seattle.

Lastly, the histograms for minimum and maximum temperatures and if it rained do show a clear pattern, however from the histograms alone, it is hard to distinguish how temperature affects the weather. Further analysis into these factors is needed to determine their significance in predicting Seattle rain.

Final Discussion

All four models I trained and evaluated – Logistic Regression, K-Nearest Neighbors, Decision Tree and Random Forest – turned out to be pretty good models for the data. With decision trees being lowest with 69% accuracy and logistic regression being the best with 76% accuracy, any of the models could be deployed and give fair predictions for whether it will rain in Seattle.

When I first created the models, I had used Precipitation (PRCP) as feature. This ended up overfitting my models, which was especially apparent on the decision tree model which gave

100% accuracy. When investigated further, I saw the tree was only 1 level – the model was only looking at precipitation to determine the output. After seeing this, I realized the problem with including precipitation is that it was the continuous/numeric equivalent to the target variable RAIN. Removing this feature helped mitigate the overfitting I had introduced to the model.

For a future improvement or implementation of the models, I would add more weather data such as humidity and whether it rained the previous day, likely collected from a weather API. Connecting to the Weather API would also enable me to deploy the model and continuous train, tweak, and test the model in real-time.

References

Lam, Linda. "Seattle's Rainy Reputation Is Well-Deserved: The Weather Channel - Articles from The Weather Channel." *The Weather Channel*, 15 Oct. 2016, https://weather.com/science/weather-explainers/news/seattle-rainy-reputation.

Tatman, Rachael. "Did It Rain in Seattle? (1948-2017)." *Kaggle*, 20 Dec. 2017, https://www.kaggle.com/rtatman/did-it-rain-in-seattle-19482017.