Predicting Severity of Accidents in the U.S.

Takao Oba, Emily Zhang, Kelsey Lin, Daniel Neufeldt

Abstract

The goal of this kaggle project is to predict the severity of car accidents that occur in the United States utilizing statistical learning models based on given immense, unstructured data. This report provides a clear and detailed description on how we built our classification model from the beginning to the end in a systematic manner. The process includes but is not limited to introduction, exploratory data analysis, data cleansing, feature selection, model construction, evaluating results, and finally discussing limitations and recommendations.

The final model is constructed utilizing a random forest and utilizes the following variables: time_diff, rush_hour, precovid, covid, postcovid, road, altrt, caution, closed, blocked, incident, newzip, online_zip, holiday, insurance, teendrivers, ccft, sdr, Temperature.F., diff_lat, diff_lng. Utilizing the above 21 predictors, we predicted the variable SEVERITY. Overall, the constructed model has a Kaggle public score of 0.93688 and a Kaggle private score of 0.93093 and our final rank is 29th.

1. Introduction

Every year, there are over 6 million passenger car accidents in the United States. This directly translates to approximately 13 car accidents in a single minute. Vehicle accidents is a very broad term and can capture from a slight scratch in the vehicle to fatal incidents. It is definitely unfortunate to have slight dents and marks on a loved vehicle, however, it is even more unfortunate to lose an individual's lives through accidents. In fact, more than 36,000 people die and 3 million people are injured every year in the United States through car accidents. Simply stating the overall number of accidents and fatality does not provide a great picture of what is

actually happening on the roads. Thus, finding an efficient model to successfully analyze vehicle accidents is significant and urgent.

In this kaggle project, the "A Countrywide Traffic Accident Dataset" provided us with 43 variables describing various features of the vehicle accidents. The dataset captures 49 states of the United States and illustrates all accidents that occur through multiple APIs that generate streaming traffic incident data. Currently (December 2022), there are about 2.8 million accident records in this dataset. We will be specifically looking at the observations from February 2016 to December 2021. The 50,000 total observations will be split into the training data, which consists of 35000 rows, and the testing data, which consists of 15000 rows. Each row records one single incident of vehicle accident and consists of data for the 43 given variables. In certain instances, not all variables were captured which introduces NA or missing values within the data. Further, there are both numerical and categorical variables and thus, unique forms of analysis would be conducted to perform the best results. Our overall mission was to introduce a classification model to predict the target variable SEVERITY in the forms of "MILD" and "SEVERE" in the testing data by selecting and adding key variables.

2. Data Analysis

Starting off, we decided to perform exploratory data analysis to help us further understand our training dataset and the predictors in it. First, we sorted the predictors into numerical and categorical variables. There were 11 numerical predictors: Start_Lat, Start_Lng, End_Lat, End_Lng, Distance.mi., Temperature.F., Wind_Chill.F., Humidity..., Pressure.in., Visibility.mi., Wind_Speed.mph. There were 33 categorical predictors: Severity, Start_Time, End_Time, Description, Street, Side, City, County, State, Zipcode, Country, Timezone,

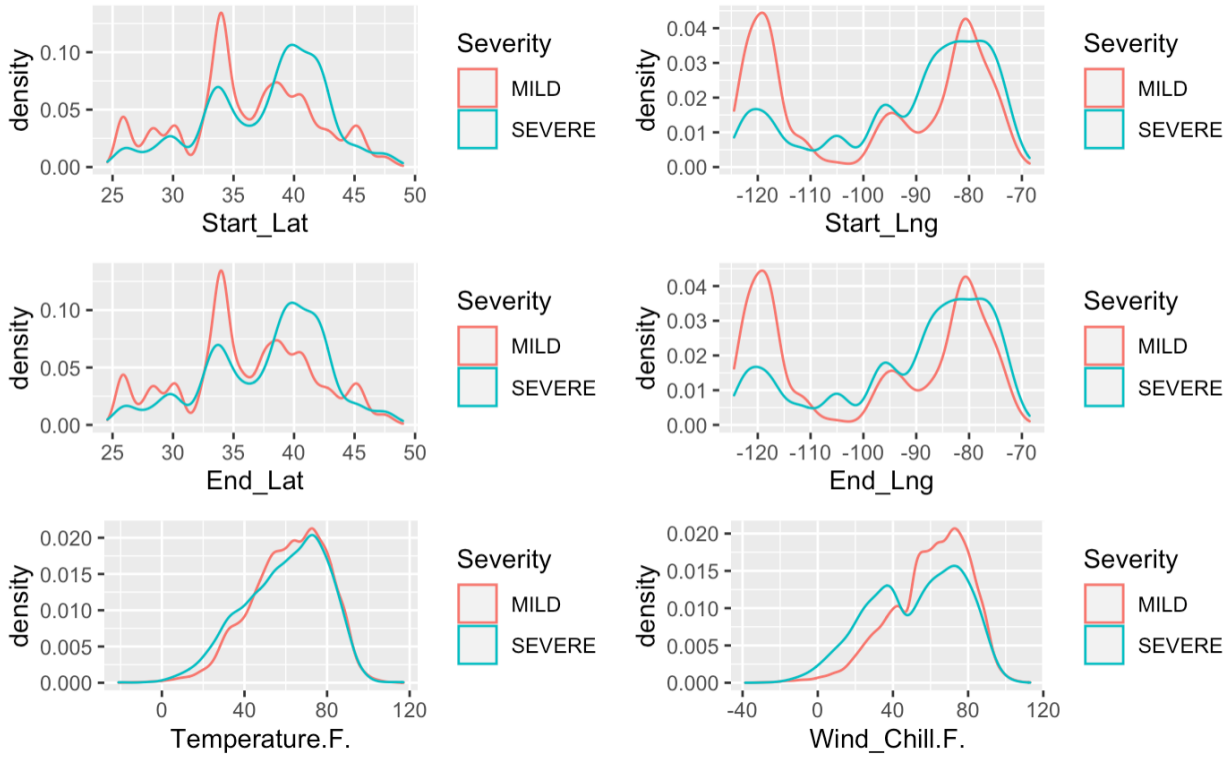
Airport_Code, Weather_Timestamp, Wind_Direction, Weather_Condition, Amenity, Bump, Crossing, Give_Way, Junction, No_Exit, Railway, Roundabout, Station, Stop, Traffic_Calming, Traffic_Signal, Turning_Loop, Sunrise_Sunset, Civil_Twilight, Nautical_Twilight, Astronomical_Twilight.

For the numerical predictors, we tried creating a correlation matrix with our response variable, Severity.

Correlation Matrix between Numerical Predictors and 'Severity'

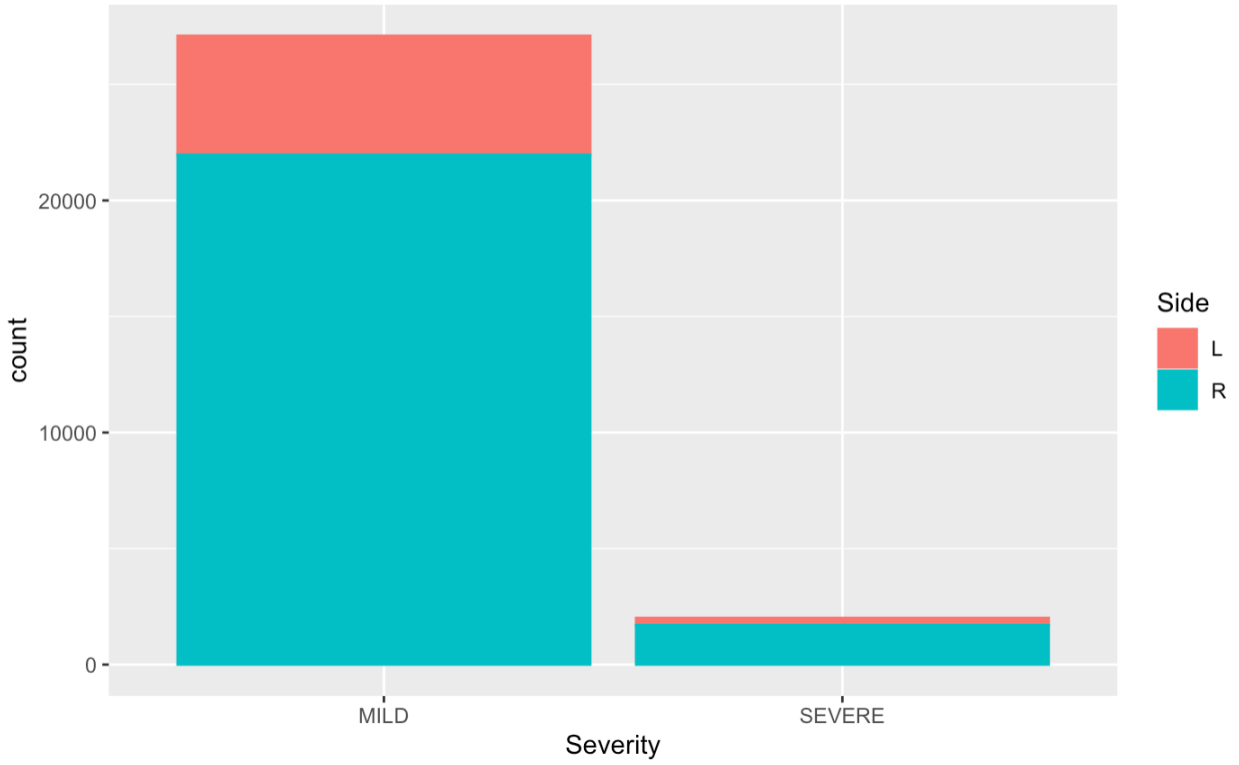
	severity
severity	1.000000000
Start_Lat	0.122739891
Start_Lng	0.102168059
End_Lat	0.122719408
End_Lng	0.102167198
Distance.mi.	0.041639564
Temperature.F.	-0.084979743
Wind_Chill.F.	-0.094599872
Humidity...	0.022847681
Pressure.in.	-0.037244301
Visibility.mi.	0.006555872
Wind_Speed.mph.	0.060132993

As shown above, none of the numerical predictors are strongly associated with Severity. Taking the six most correlated variables from this matrix, we created density plots to dive deeper into their relationship with the response variable.

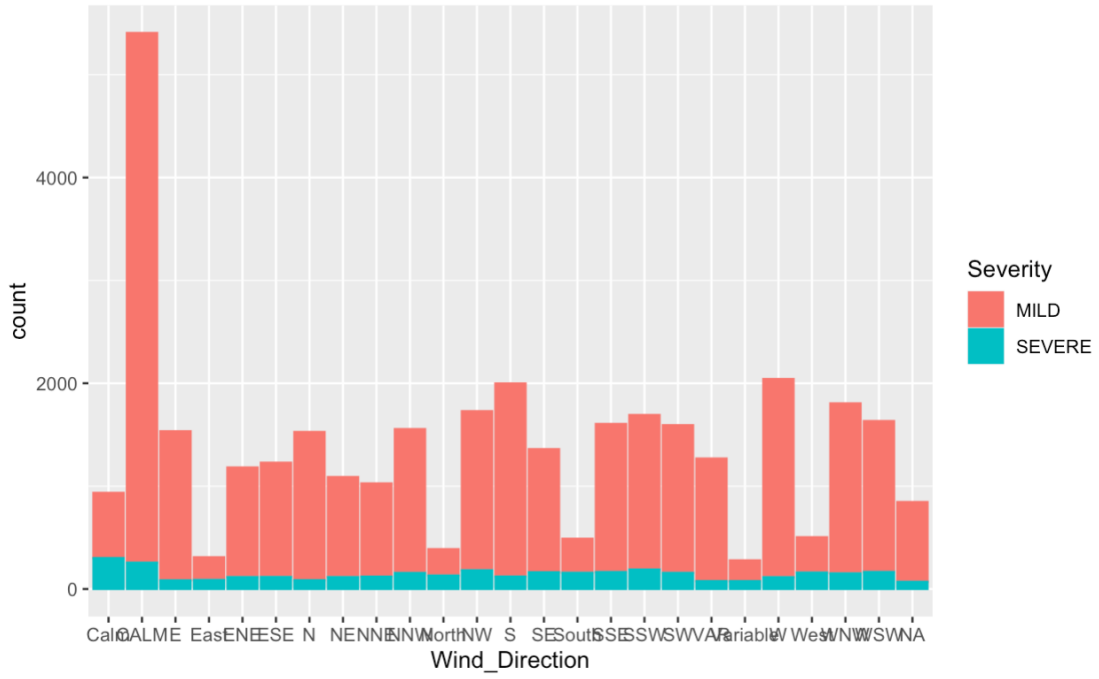


From the plots above, we are able to see that there is not a significant amount of separation between the predictors based on severity. For categorical variables, we tried to plot each predictor separately against “Severity” to try and find a predictor with significant difference between “MILD” and “SEVERE.”

Plot of Side vs. Severity



From the example above, we can see that there is not a significant difference between the "MILD" and "SEVERE" bars. Additionally, there were a few predictors (Street, City, County, State, Zipcode, Airport_Code, wind_Direction, and Weather_Condition) that had too many categories within the predictor itself to make a stacked bar plot that was interpretable. There were also some predictors (Start_Time, End_Time, and Description) that simply did not make sense to include in this style of analysis due to the manner in which the predictors were constructed.

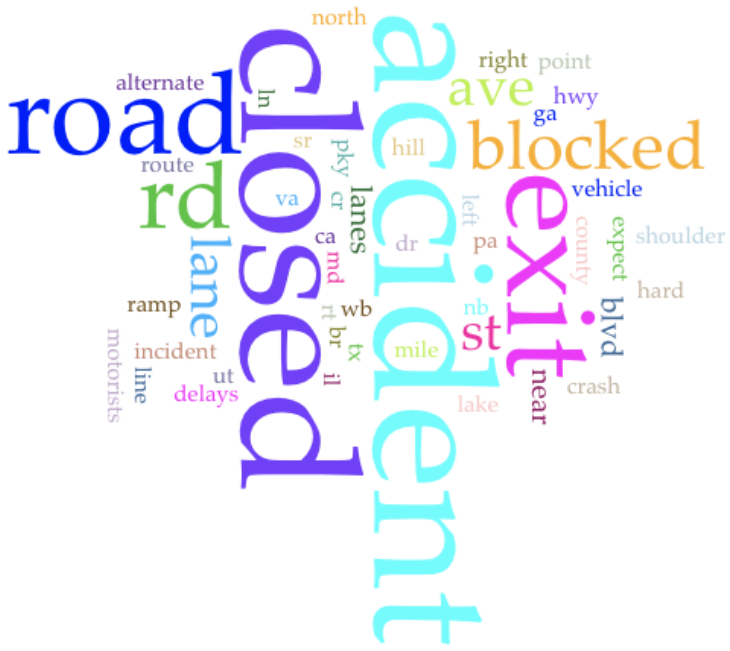


After observing all of our graphs, we came to the conclusion that most of the categorical predictors originally given in the training dataset did not offer a very strong indication of being a good predictor for the severity of a car accident. Therefore, we decided to create new predictors from already existing ones instead of using the existing ones by themselves.

3. Variable Selection

From the previous section, we concluded that the predictors originally given in the training dataset were not fit for creating a model that would predict car accident severity. From this, we decided to create new predictors from the original ones. First, we decided to use description regular expressions to pinpoint the key phrases present in the Description category for both “MILD” and “SEVERE” accidents.

Most Common Words in “SEVERE” Accidents



The most common words appearing for severe car accidents were “accident”, “closed”, and “exit.” The most common phrases appearing were “due to accident”, “road closed due”, “closed due to.” If the accident was “MILD”, then a whole road would not need to be closed. Cars could just move around the accident and would have been told to proceed with caution. Therefore, the word “closed” appearing most of the time makes sense.

Most Common Words in "MILD" Accidents

The most common words appearing for mild car accidents were "accident", "rd", and "traffic." The most common phrases appearing were "due to accident", "drive with caution", and "closed due to."

From examining the two Description columns above for both "MILD" and "SEVERE" car accidents, we were able to create new dummy predictors using text mining. These variables searched the Description column, and if it contained the respective word, the new predictor column would have a '1.' If it did not contain the word, there would be a '0.' These new predictors were road ('road'), altrt ('alternate'), caution ('caution'), closed ('closed'), blocked ('blocked'), and incident ('incident').

We also created new COVID-19 predictors based on the Date column. A precovid column if the date of the accident was before April 1st, 2020. A covid column if the date of the

accident was between April 1st, 2020 and before January 1st, 2021. Finally, a postcovid column was created if the accident occurred after January 1st, 2021.

We also created variables using the numerical differences between certain predictors from the original dataset. First, we used str_split() on the Start_Time and End_Time predictors to properly format the date. We then put these newly formatted dates into the diff_time() function in order to find the duration of the car accident per observation. We found that the mean duration for severe accidents was around 4.26 hours longer than the mean duration for mild car accidents. The other two numerical difference predictors that we created were the difference in latitude and the difference in longitude for each observation. We simply created two new predictor columns that were the absolute value of the End_Lat or End_Lng minus the Start_Lat or Start_Lng respectively. We named these new variables time_diff, diff_lat, and diff_lng.

Utilizing regular expression again, we focused on three predictors: Time, Date, and Location. For Time, we calculated the differences in time to try and determine if the accidents are correlated to rush hour. For Date, we looked at dates during the holiday season and externally sourced which months have the most car accidents. For holidays, we included dates of the most notable holidays in the United States (Christmas, New Years, Fourth of July, Mother's Day, Father's Day). We also found that more people are on the road during the summer months. Pertaining to Location, we took into consideration a number of factors. Using both the training dataset and external data, we found states and zip codes that have the highest rate of car accidents/have the highest number of teen drivers, the highest insurance premiums, and places which have the highest number of fatalities due to car accidents. We created new variables based off of these categories.

States with the Highest Insurance Premiums	States with the Most Number of Teen Drivers	States with the Highest Number of Car Crash Fatalities due to Teen Drivers
Delaware Louisiana New York Georgia Maryland Michigan New Jersey Florida Rhode Island South Carolina	Kentucky Mississippi North Carolina Missouri West Virginia	Wyoming South Dakota Mississippi Missouri Alabama

We also classified states into three categories of deaths per 100,000 population. These included high, mid, and low categories.

States with a high death rate per 100,000 population (> 15)	States with a medium death rate per 100,000 population (between 9 and 15)	States with a low death rate per 100,000 population (< 9)
Alabama Arkansas Florida Georgia Kentucky Louisiana Mississippi Missouri Montana New Mexico Oklahoma South Carolina South Dakota Tennessee Wyoming	Arizona California Colorado Delaware Idaho Illinois Indiana Iowa Kansas Maine Maryland Nebraska Nevada North Carolina North Dakota Ohio Oregon Texas Vermont Virginia West Virginia	Alaska Connecticut District of Columbia Hawaii Minnesota New Hampshire New Jersey New York Pennsylvania Rhode Island Utah Washington

	Wisconsin	

Zip Codes with the Highest Rate of "SEVERE" Car Accidents	Location
75243	Dallas, TX
80229	Denver, CO
33168	Miami, FL
75228	Dallas, TX
80216	Denver, CO
80401	Golden, CO
37210	Nashville, TN
60607	Chicago, IL
20020	Washington, DC
33127	Miami, FL
33150	Miami, FL
60621	Chicago, IL
77092	Houston, TX
80104	Castle Rock, CO
80118	Larkspur, CO

The names of each of these newly created predictors are: rush_hour, crash_months, holiday, teendrivers, ccft (car crashes fatalities teens), insurance, sdr (state death rate), newzip.

For these predictors listed above in the tables, we obtained them by using regular expressions. For the Zip Codes with the Highest Rate of "SEVERE" Car Accidents predictor, we used the training data set and saw which zip codes had the highest frequency of severe car accidents. We then took the top 15 zip codes and made a predictor that gave the observation a

"1" if the zip code was one of the zip codes listed above or a "0" otherwise. For all of the other predictors, we utilized external resources from the internet to determine the states that matched the criteria we were looking for. We then did a similar process where we used regular expression to see if the state from an observation was in one of the groups of states and assigned the appropriate boolean variable to it for each predictor.

One issue that we ran into was missing values. The only predictor we actually ended up using that had missing values was Zipcode. Because zipcodes are non-numerical and in too many groups for us to use a built-in imputation (like mode), we needed to manually impute the missing zip codes. Fortunately, there were only a small amount of missing values, so we simply found the city and county of the accident and looked up all of the zip codes in that area. We noticed that none of these zip codes that we found online were in the group of zip codes that had high rates of severe car accidents from above, so we just imputed a "0" in this column for each of these observations.

4. Methodology

After constructing all of our new predictors, we began our model creation process. For our model, we used the predictors: rush_hour, time_diff, precovid, covid, postcovid, road, altrt, caution, closed, blocked, incident, newzip, hoiday, insurance, teendrivers, ccft, sdr, roadclosed, closeddue, diff_lat, diff_lng, temperature, crash_months. Note that we added the temperature variable (Temperature.F.) because it was one of the best numerical predictors from the original data.

During this process, we needed a way to determine the success of a model. The most obvious way of determining success would be to base it on the Kaggle score that the model

obtains; however, a problem with this is the fact that we were limited to only three Kaggle submissions per day, so we needed to find an alternative option. This alternative option was splitting our training data. The training data consists of 35,000 observations, so we simply used random sampling to randomly split the training data into a 30,000 observation data set to represent our sampled training data and a 5,000 observation data set to represent our sampled testing data. We would create the model using our sampled training data and then make the prediction with the sampled testing data, create a confusion matrix with the predicted severity and the actual severity from the sampled testing data, and then come up with our misclassification rate. This is how we found the "expected misclassification rate" for each model that we tested. Because there were so many observations, we deemed this method an appropriate way to base the success of our models.

The first model that we decided to use was the KNN model due to its ease to implement and understand. The variables that we used were our numerical predictors (difference in time, difference in latitude, difference in longitude, and temperature). We used our split training data to test this model and yielded a misclassification rate of 11.36%. Because of this number, we decided to not use the KNN model as it was not able to utilize the categorical predictors that we created and thus was very limited for this project.

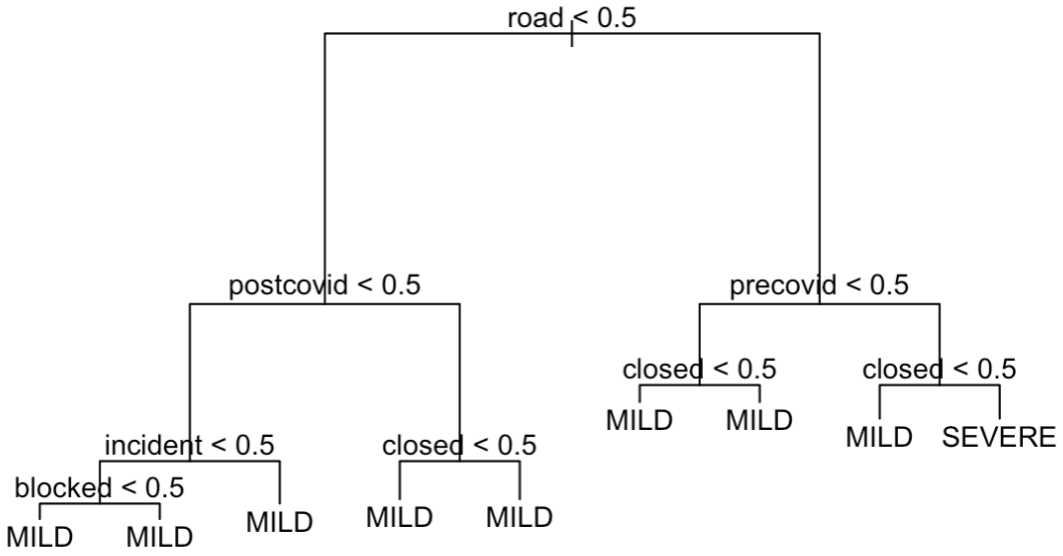
The second model that we tested was the logistic regression model. This model was an appropriate model because we were trying to predict the severity of car accidents by placing them into categories (MILD and SEVERE) and we could also use our categorical predictors. After creating the model, we used our split data to determine an expected misclassification rate

of the logistic regression and found it to be 7.52%. This was significantly better than the KNN model; however, this model did not perform too well in the Kaggle leaderboards, so we decided to explore other models in an effort to achieve greater accuracy.

Actual Sample Testing Severity

		MILD	SEVERE
Logistic Regression	MILD	4438	325
	SEVERE	51	186

The third model that we created was a decision tree. The main reason for attempting to use a decision tree was because of its simplicity and ability to differentiate categories when classes are not well separated. After constructing the decision tree, the issue that we ran into was that there was only one branch that determined if an observation had a severity of severe. Even after pruning the tree, we saw that this did not change anything.



Additionally, when running the split training data on this tree, the expected misclassification rate was over 10% (10.54% to be exact). This meant that we could not use this model and needed to look into random forests in an effort to create a decision tree that yields a higher accuracy.

The final model that we created utilized the random forest method. The main issue with a decision tree was the fact that a small change in the data would result in a significant change in our results, so to overcome this challenge, we decided to use a random forest. Initially, to deal with the missing values for our numerical predictors, we used the missForest() function to do this. After filling our missing values and running the random forest model, we began to experiment with what mtry worked the best in our model. Utilizing our split training data, we saw that even though it is recommended that we use the square root of our number of predictors as our mtry number - around 5 - we obtained better Kaggle results using mtry = 6. In the end, our expected misclassification rate was 6.62%. After seeing this results, we wondered if our method of replacing NA values was truly the best way to do it, so we decided to try one last modification

We kept our idea of using a random forest as this gave us the best results by far in the Kaggle competition; however, we tried a new way of filling our missing values for the numerical predictors. In this case, rather than using the missForest() function, we decided to just replace each NA value with the median value of that predictor. By doing this imputation, along with keeping everything else the same in our model, we had an expected confusion matrix of

Actual Sample Testing Severity

Random Forest Using Median Imputation

	MILD	SEVERE
MILD	4539	289
SEVERE	4	168

We can see that the expected misclassification rate of this model was 5.86%. We experimented with different mtry values again; however, the result stayed the same and we determined that mtry = 6 was the value that yielded the best results. Because this model was the most successful, we deemed our random forest model with median imputation and mtry equaling 6 to be our main Kaggle submission.

5. Discussion and Limitations

Our final model was a random forest (mtry = 6) utilizing median imputation. While we did try libraries such as missForest, what made us go against using mean imputation was that some of the data was skewed and could consequently affect the accuracy of our model. Our model consisted of 21 total predictors with a public Kaggle score of 0.93688 and a private Kaggle score of 0.93093. However, we acknowledge accuracy can still be improved, and there are a few limitations to this model.

While random forest algorithms are efficient in training, one main limitation is that a larger number of trees makes random forests more ineffective for making real-time predictions due to training time (Donges, 2022). This makes random forests a more computationally expensive approach, but it is a tradeoff with its ability to process and handle large datasets.

Our “Description Regular Expression” predictor approach that looked for key words and phrases mostly commonly associated with “MILD” or “SEVERE” was an extremely significant contributor to our model. However, text mining results in higher multicollinearity due to potential overlap in the words and phrases used.

Lastly, when taking a look at the confusion matrix of our final model, we can observe that our model is stronger in accurately predicting “MILD” cases compared to “SEVERE”. We drew

data from external sources that may have contained certain biases that affected our predictor accuracy. In a real-world setting, it is more important to detect “SEVERE” accidents to better understand the factors that contribute to such cases. Thus, our model can further be improved by increasing accuracy for predicting “SEVERE”.

6. Conclusion and Recommendations

From this project, we had the opportunity to explore the strengths and drawbacks of various modeling methods when handling the same dataset. It was extremely crucial to organize our process and plan out how to build new models, test them, and analyze the accuracy. This way, we were able to gauge each model’s performance as well as decide what to continue improving on leading up to creating our final model. In the future with additional time, more focus can be placed on further analyzing the data, creating other variables such as one for holidays, or applying transformations to reduce multicollinearity.

From data cleansing/preparation to feature engineering/model construction to analyzing our results, we have learned how to apply machine learning on real-world issues, and our final classification model was a strong predictor for the severity of U.S. accidents.

References

Donges, N. (2022, September 28). *Random forest classifier: A complete guide to how it works in Machine Learning*. Built In. Retrieved December 9, 2022, from https://builtin.com/data-science/random-forest-algorithm

Horymski, Chris. “10 Most Expensive States for Car Insurance.” *Experian*, Experian, 21 Mar. 2022, https://www.experian.com/blogs/ask-experian/research/most-expensive-states-for-car-insurance/.

Martin, E. (n.d.). *Highest and lowest car insurance rates by ZIP code for 2022*. Carinsurance.com. Retrieved December 9, 2022, from https://www.carinsurance.com/Articles/car-insurance-rate-comparison.aspx

National Safety Council. (2022, March 16). *Motor vehicle - crashes by Month*. Injury Facts. Retrieved December 9, 2022, from https://injuryfacts.nsc.org/motor-vehicle/overview/crashes-by-month/

The Perecman Firm, P.L.L.C. (2022, September 1). *Blog - 8 of the deadliest holidays for driving*. The Perecman Firm, P.L.L.C. Retrieved December 9, 2022, from https://www.perecman.com/blog/8-of-the-deadliest-holidays-for-driving/#:~:text=Memorial%20Day.,likely%20on%20Memorial%20Day%20weekend