Predicting Car Accidents

GROUP H

STATS 101C LECTURE 1

Presentation Contents

01 Introduction

Our team and the overall problem statement

02 Data Cleaning

Dealing with missing values, categorical variables, and multicollinearity

03 Model

How we utilized our imputed data to predict the severity of accidents

04 Limitations & Conclusion

Where we think we could have improved, assumptions made, and ultimate results

01

Introduction

Our Team (GROUP H)

Nishant Jain

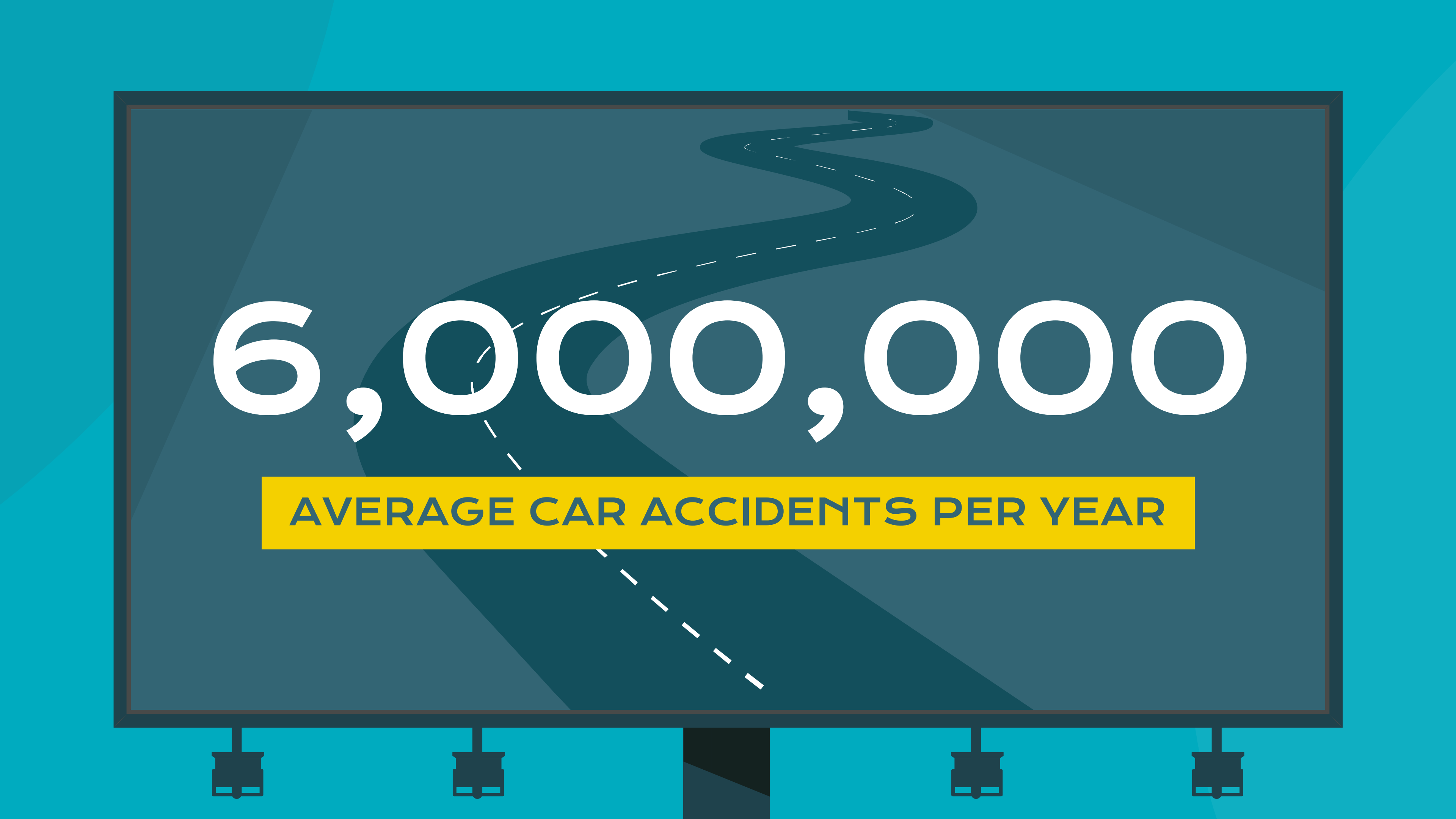
Third Year
Data Theory

Taro Iyadomi

Third Year
Data Theory

Anish Dulla

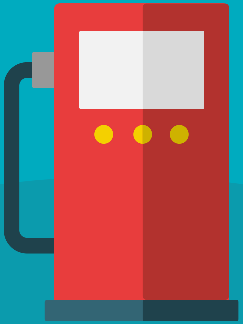
Third Year
Statistics


6,000,000
AVERAGE CAR ACCIDENTS PER YEAR

How can we predict the severity of an accident's impact on traffic?

02

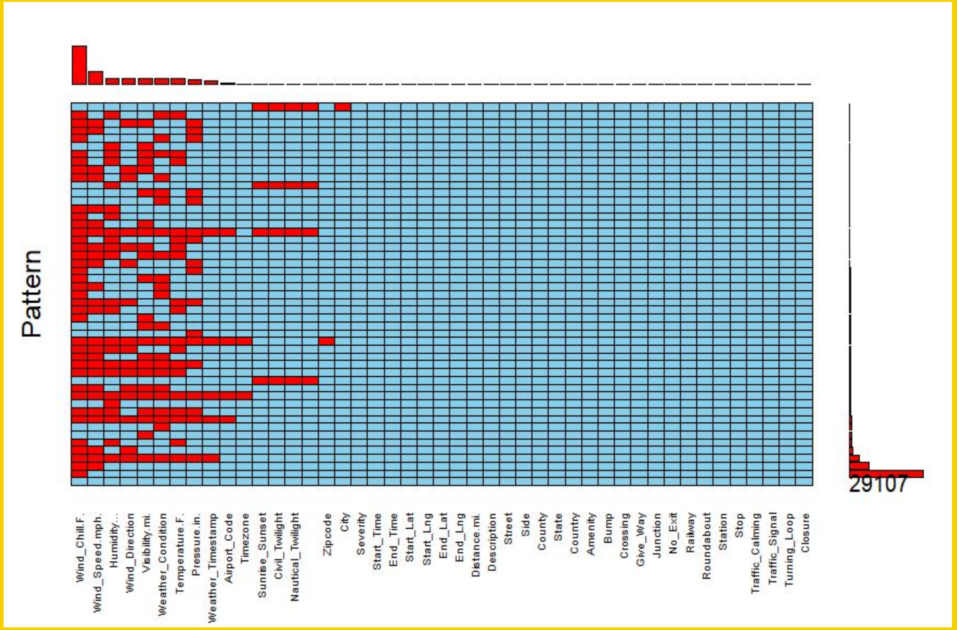
Data Cleaning



Understanding Missing Values
(Part One)

Initial Insights:

- Training data contained **13,211 missing values**
  - *Only 17 predictors accounted for all of the NAs*
- Most of the missing values come from weather related variables
- *This pattern raises questions about the data collection process, as weather variables weren't strictly being recorded*



Understanding Missing Values
(Part One)

Weather Related Variables	
Variable	**% Missing**
Wind Chill	16.19
Wind Speed	5.34
Humidity	2.42
Wind Direction	2.41
Visibility	2.34
Weather Condition	2.31
Temperature	2.29
Pressure	1.91

Further Insights:

- The majority of variables have **less than 5% missing values**
    - *This is ideal for imputation, as it increases the amount of information fed to the model without drastically increasing bias*
- However, we can first clean up the data to eliminate those missing values before proceeding to imputation

Dealing with Categorical Variables

many of the categorical variables contain far too many levels to work with, so we must deal with each of them accordingly to maximize their value.

Start Time, End Time, and Weather Timestamp:

Originally, each of these variables contained nearly 35,000 levels.
This is practically unusable, but there's a lot of useful information to be extracted.

1. Using regular expressions to extract the hour, month, and year of each accident.
2. Converting to POSIXct (seconds since January 1st, 1970)
    a. Creating a Time Lapsed predictor with the new Start Time and End Time predictors
3. Creating a Night predictor (whether the accident took place from 7pm to 7am)

Dealing with Categorical Variables

Night, Sunrise Sunset, Civil Twilight, Nautical Twilight, and Astronomical Twilight:

- Sunrise Sunset, Civil Twilight, Nautical Twilight, and Astronomical Twilight measure the same thing with a different metric, so there is multicollinearity between these predictors.
- What's worse is that there are several observations where all four of these predictors are missing, which can make it more difficult for imputation.
- However, there are **no** missing values for the new Night variable, which means that we can combine all four of these predictors into one predictor without any missing values!

Variable	**% Missing**
Sunrise Sunset	0.0857
Civil Twilight	0.0857
Nautical Twilight	0.0857
Astronomical Twilight	0.0857

Dealing with Categorical Variables

Start Lat, Start Lng, End Lat, and End Lng:

- As expected these four predictors are highly correlated
- By themselves, each of these predictors provide approximately the same information, giving our model unnecessary complexity
- To remedy this, we created a **distance traveled predictor** by calculating the euclidean distance between the start and end points of the accidents
- This differs from the Distance variable, because that measures the extent of the road affected by the accident, not the distance the car traveled during the accident

The “Description” Variable

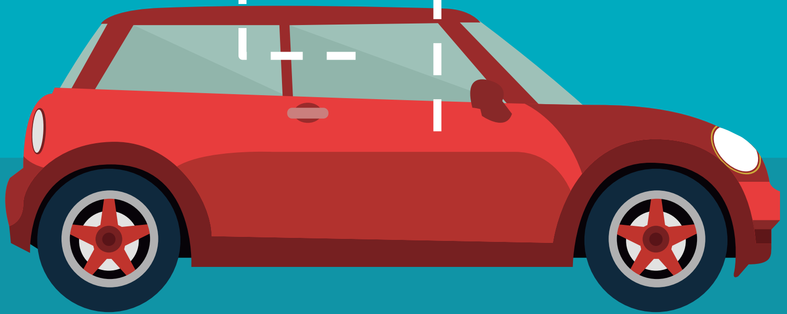
SEVERE Word Cloud

MILD Word Cloud

Top “Description” Terms – Important Predictors

MILD Car Accidents

- “accident”
- “incident”
- “stationary”
- “slow”
- “traffic”
- “caution”



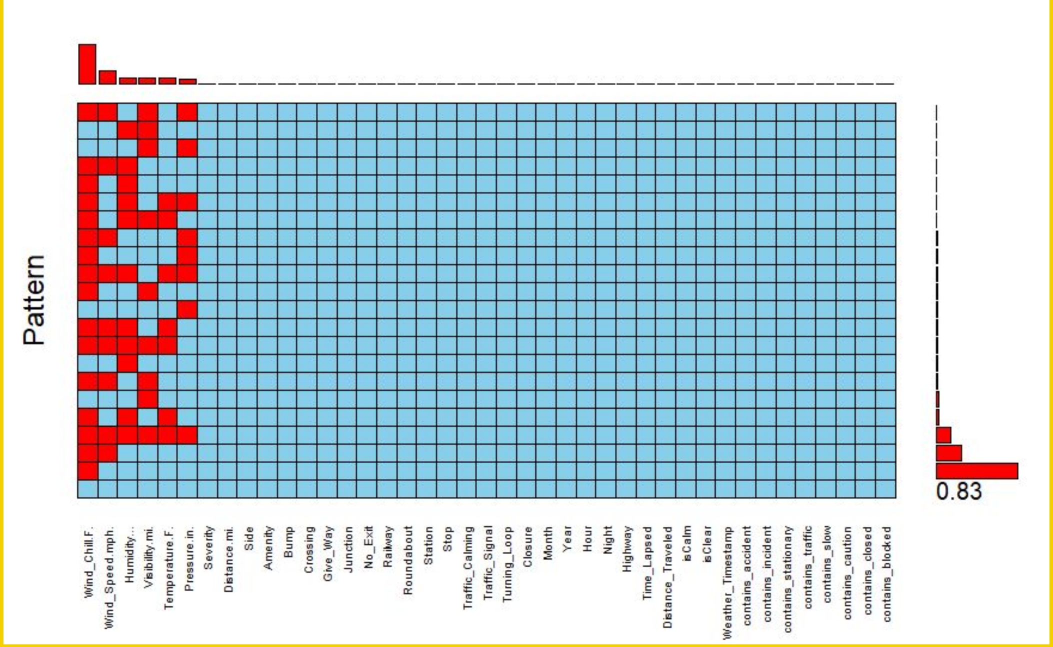
SEVERE Car Accidents

- “road”
- “closed”
- “due”
- “lane”
- “blocked”
- “exit”

“Description” Extracted Variables

Variable	Type	Proportion True (1)
contains_accident	Categorical (Boolean)	27.5%
contains_incident	Categorical (Boolean)	1%
contains_traffic	Categorical (Boolean)	23.8%
contains_slow	Categorical (Boolean)	1%
contains_caution	Categorical (Boolean)	21%
contains_closed	Categorical (Boolean)	8.5%
contains_blocked	Categorical (Boolean)	8.3%
contains_road	Categorical (Boolean)	1%
contains_exit	Categorical (Boolean)	1.68%
contains_lane	Categorical (Boolean)	9.2%
contains_due	Categorical (Boolean)	27.5%

Understanding Missing Values
(Part Two)



10,680

Missing values after cleaning data (2531 NAs removed)

6

Predictors with NAs

Imputation:

Additive Regression, Bootstrapping, and Predictive Mean Matching

- Hmisc Library
- This method accounts for all aspects of uncertainty in the imputations using multiple bootstrap resamples
- A flexible, linear additive regression model is fitted using the data
- Finally, predictive mean matching is performed which works for binary, categorical, and continuous variables
- Resulting dataset has 0 missing values!

03

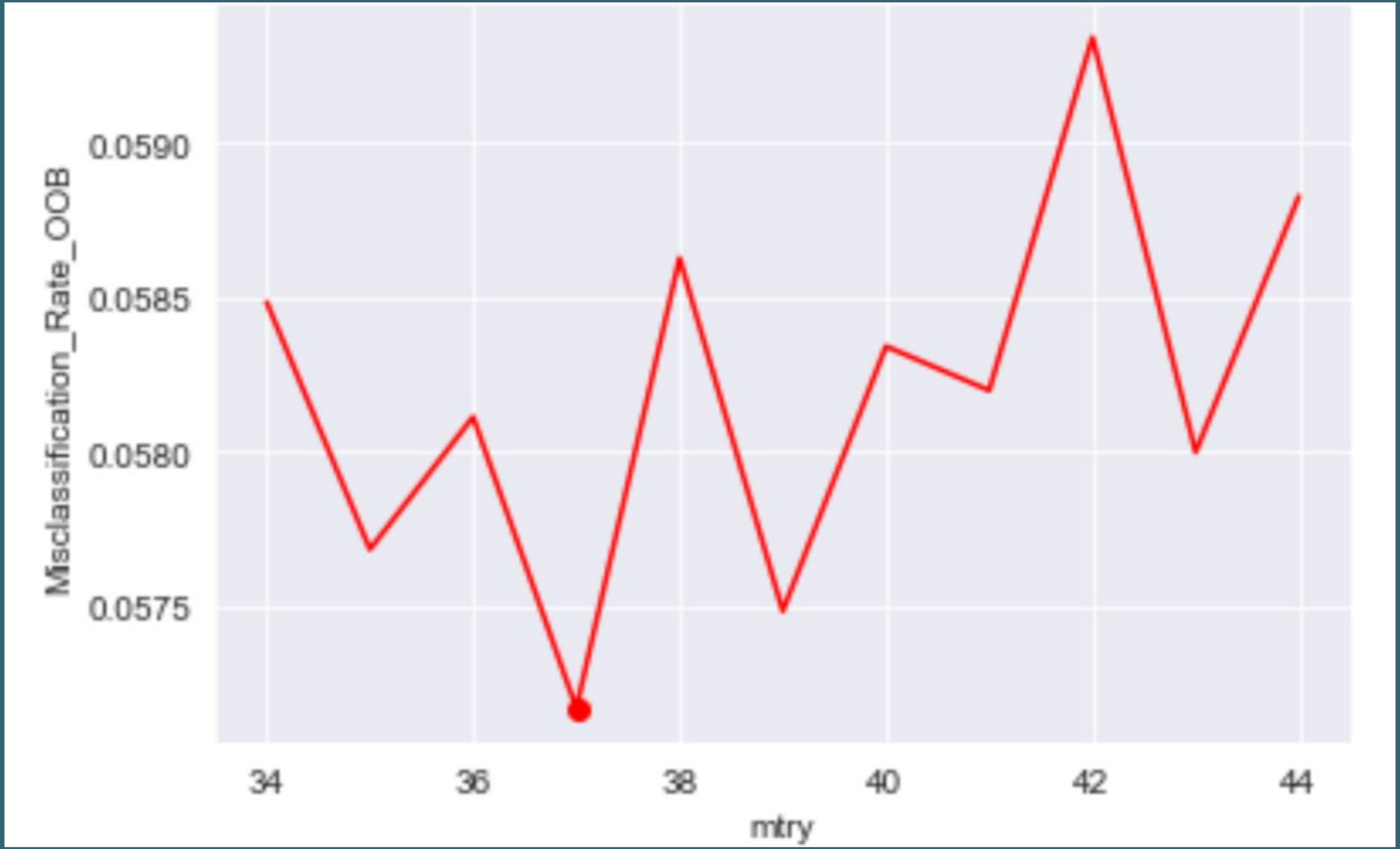
Model

Our Predictive Model - Random Forest

- We decided to use a **Random Forest** model.
  - Our Random Forest performed highest with **MCR: 0.05717** on Test Data

- While trying other classification techniques like Logistic Regression and KNN, Random Forest has the following benefits that make for a better model:
  - High Interpretability (Variance Importance Plot)
  - Versatility to Data (Multiple Decision Trees are Fit)
  - Prediction Performance

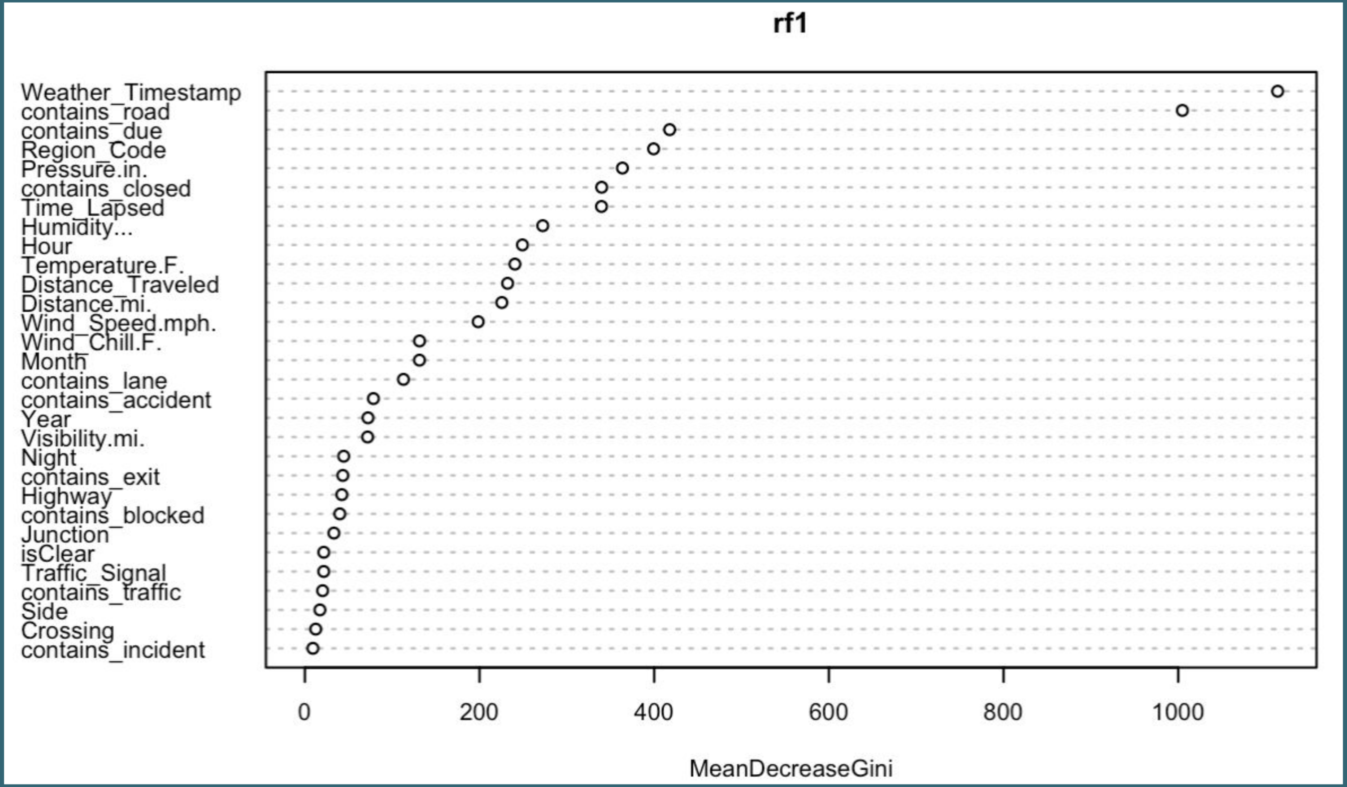
Random Forest Misclassification Rates for Different MTRY



rf1.pred	MILD	SEVERE
MILD	31092	1644
SEVERE	390	1874

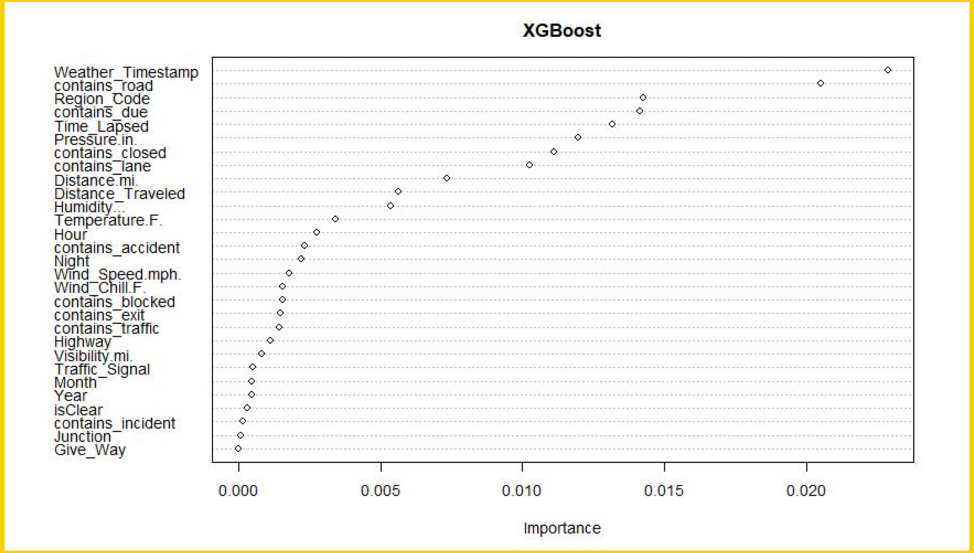
Most Relevant Predictors

for our model "rf1"



eXtreme Gradient Boosting

XGboost is an optimized boosting algorithm that minimizes the loss (cost) function of traditional boosting by combining weak learners.



XGBoost:

- Great for unbalanced, sparse data
- More efficient
- Self tunes

Random Forests:

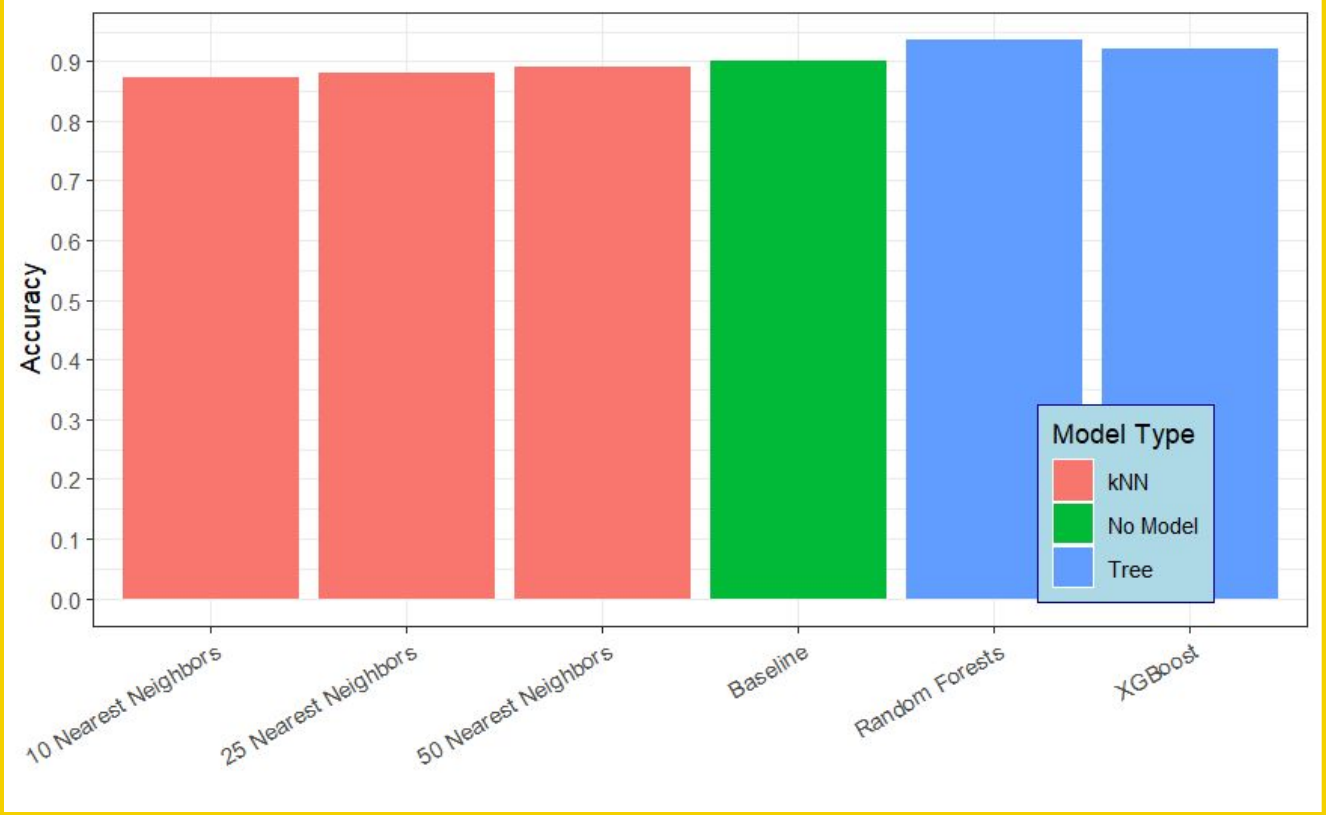
- Easier to tune
- Better for preprocessed, clean data

kNN Classifiers

- Used k-values 10, 25, and 50
- Accuracy increases with higher k-values
- All three models resulted in test accuracy scores less than 90%
- Indicates kNN models are too complex

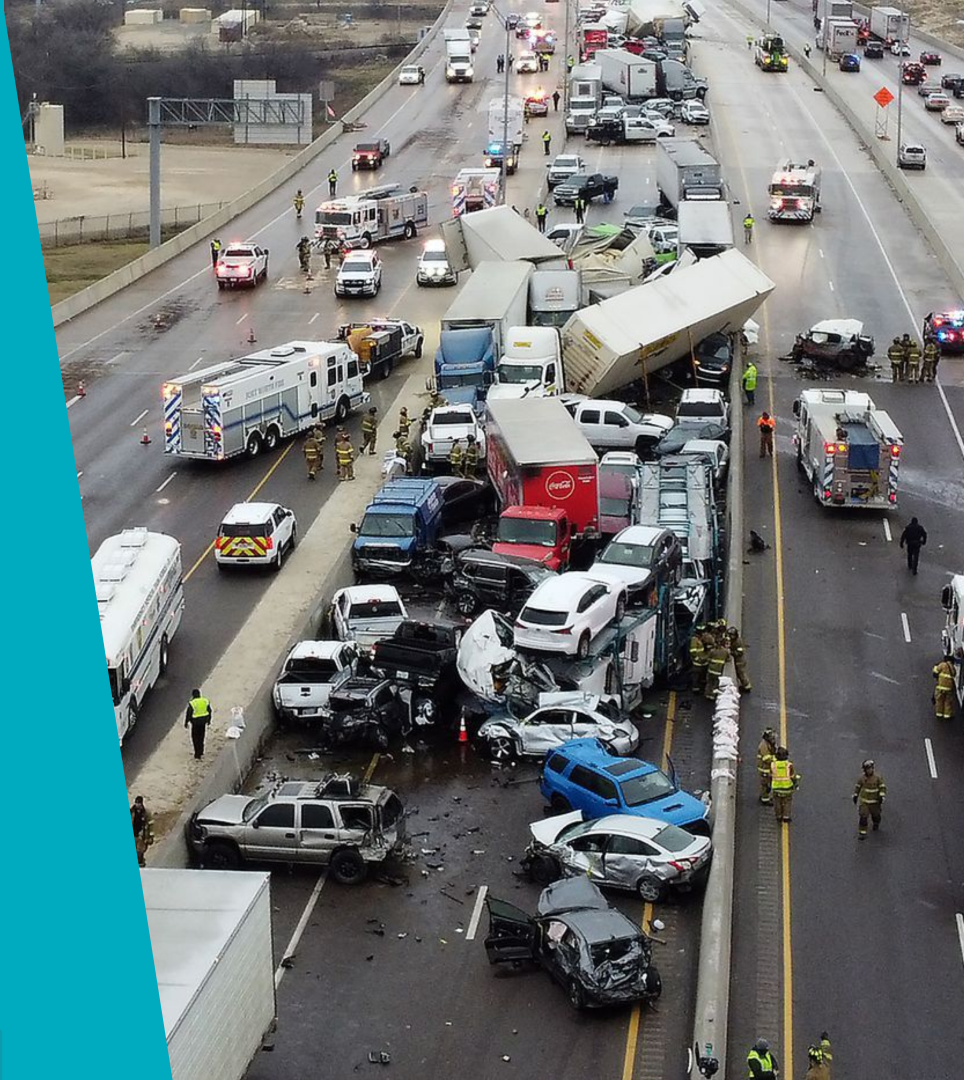
K-Value	Test Accuracy (%)
10	87.41
25	88.08
50	89.28

Model Overview



04

Limitations & Conclusion



Limitations

“DESCRIPTION”

Our 11 added predictors based on terms from “Description” may not be fully representative of the original predictor.

IMPUTATION

Using Hmisc predictive mean matching, the algorithm assumes the variables are linear – which may not be true.

OVERFITTING

Without pruning our model based on the Variable Importance Plot, the RF may have overfit the data using unnecessary predictors.

EXTERNAL DATA

Without using external data, we may have missed an opportunity to add stronger features to better predict severity.

IN CONCLUSION

PUBLIC

SCORE: 0.9354

RANK: 29

PRIVATE

SCORE: 0.9357

RANK: 23

FINAL

SCORE: 0.9355

LECTURE RANK: 14