Predicting Severity of Car Accidents

Team：GaveProudSharkTracer
Gavin Chen, Proud Jiao, Yun Lin, Keying Zhang

Overview

- Introduction
- Data cleaning methods and outside resource
- Method: Random Forest
- Conclusion
- Drawbacks and possible improvements

Project objective:
Classifying Car Accidents as Severe/Mild in the United States

Introduction

Description

Countrywide accident data from February 2016 to December 2021

Each observation represents a car accident recorded in the US territory.

Background

The most common cause of death of teenagers is a vehicle accident.

More than 90 people die in car accidents every day.

3 million people in the U.S. are injured every year in car accidents.

Data Set

35,000 Observations

43 predictors (Wind speed, Humidity… …)

Response variable: Severe / Mild

Data Cleaning & Transformation

- Handling missing values
- Feature engineering

Handling NAs

KNN:

- Filled missing **Zipcode** using Start/Ending Latitude/Longitude with k = 1
- Basically Zipcode of the nearest location.

MICE:

- Reduced missing values from 13000 to 1500
- Used to train RandomForest model

Iterative Imputer:

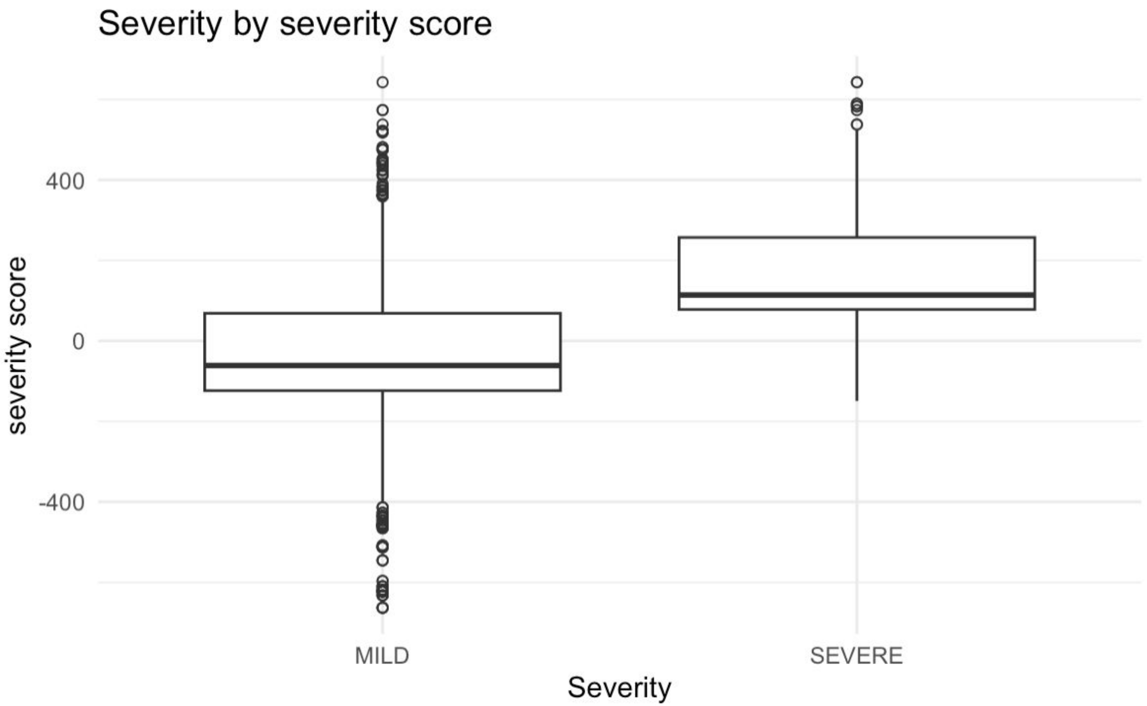
- Pro: Reduced missing values to 0 while MICE could only reduce it to 1500
- Con: Needs to change categorical variable into numeric first
- Used to train ANN model.

Feature engineering

- Generated
    a. **Duration, Month, Week** from Start_ and End_Time.
    b. **Description length**, **Severity Score, Detected Severe, Detected Mild** from Description using Text Mining Technique
    c. **Population**, **population density** from Zipcode and 2022 US census (References: https://simplemaps.com/data/us-zips)
    d. **Speed** by Duration and Distance.mi.
- All generated features significant
- **Possible improvements**
    - Accidents spans from 2016-2021
    - But we used 2022 US census

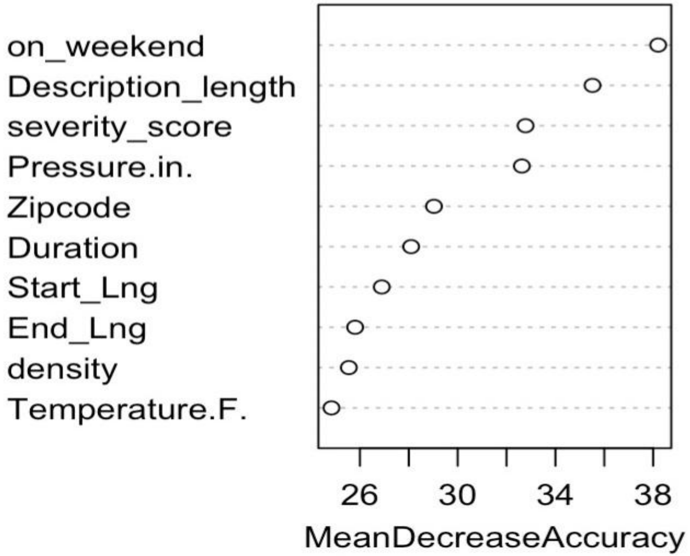
Text Mining Technique

- Assign weight to words according to frequency and severity
- Calculate severity & mild score for each Description
- Final score = severity score - mild score



Importance of predictors

- Used RF to select/visualize important predictors



Methodology

Try different models and determine the best model based on the performance.

Our models with Kaggle Score

Models	Score
Logistic Regression	0.93288
K-Nearest-Neighbors	0.9328
XGBoost	0.93484
Artificial Neural Networks (ANN)	0.92897
Random Forest	0.94373

Random Forest

Abstract: using the idea of ensemble method Bagging to train the trees by randomly select the data with replacement with limited features.

Performance: 94.37% CV accuracy with the training data set.

Max depth: 21

Variables: 'Start_Lat', ' Start_Lng', 'End_Lat', 'End_Lng', 'Distance.mi.', 'Side', 'City', 'County', 'State', 'Zipcode', 'Country', 'Timezone', 'Temperature.F.', 'Wind_Chill.F.', 'Humidity...', 'Pressure.in.', 'Visibility.mi.', 'Wind_Speed.mph.', 'Weather_Condition', 'Amenity', 'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway', 'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal', 'Turning_Loop', 'Sunrise_Suns' . . .

Conclusion

Conclusion based on random forest model with performance 0.94373 reported by kaggle.

Conclusion

Final Kaggle Rank: 6

Better performance on **nonparametric model,** discovered non-linear relationship among predictions,

Description is so important that building a model with descriptions only can reach about 93% accuracy in the testing set.

Drawbacks and Future Improvements

Drawbacks based on our model and possible improvement for future analysis.

Drawbacks & Possible Improvements

Timezone

- For all the time-related value, we assume that they are Universal Time Coordinated (UTC) but it’s not the case. We could manage them with their corresponding region.

Collinearity

- There is a chance of multicollinear features getting picked up together

Overfitting

- Max leave for max depth of 21 is 2^21

Thank You!