Car Crash Data Modeling Presentation

Project for Flatiron's Data Science Bootcamp
by Henry Alpert
January 4, 2021

FLATIRON SCHOOL

Data Overview

City of Chicago Car Crash Data

Problem: Look to limit accidents that lead to serious injuries or fatalities

Data Source: City of Chicago accident data, mostly 2017 to present. This data set merged with another data set of people involved in the car crashes, filtered for only drivers.

- 461,315 entries. Pared to 70,000 and ultimately to 55,766 after removing outliers
- Final columns included speed limit, weather, lighting, roadway conditions, number of vehicles involved, accident time (hour, day of week, and month), and age and sex of driver

Target

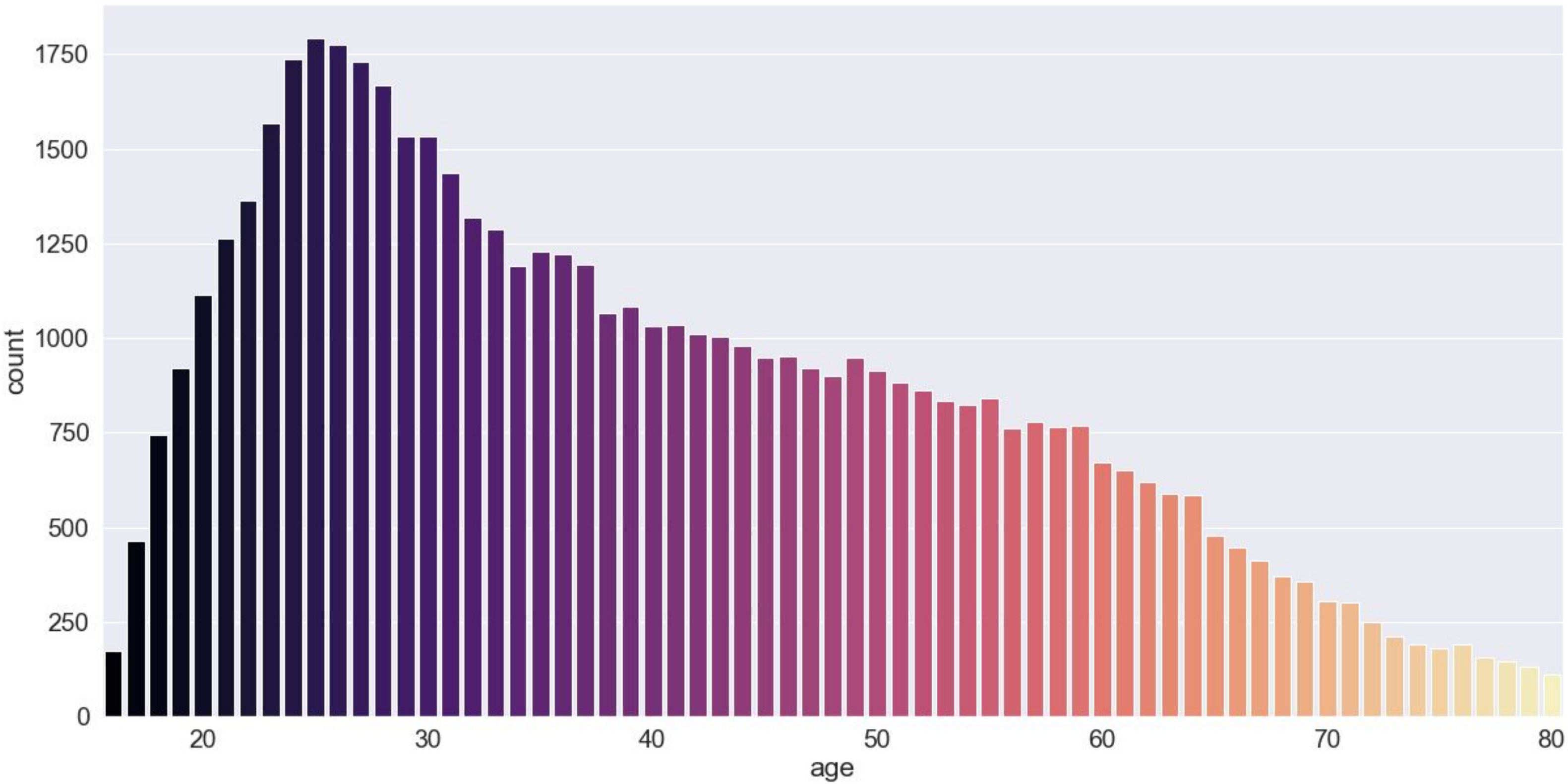
- **Serious Accident -** Combined crashes with “fatal” and “incapacitating” injuries.

Imbalanced Dataset

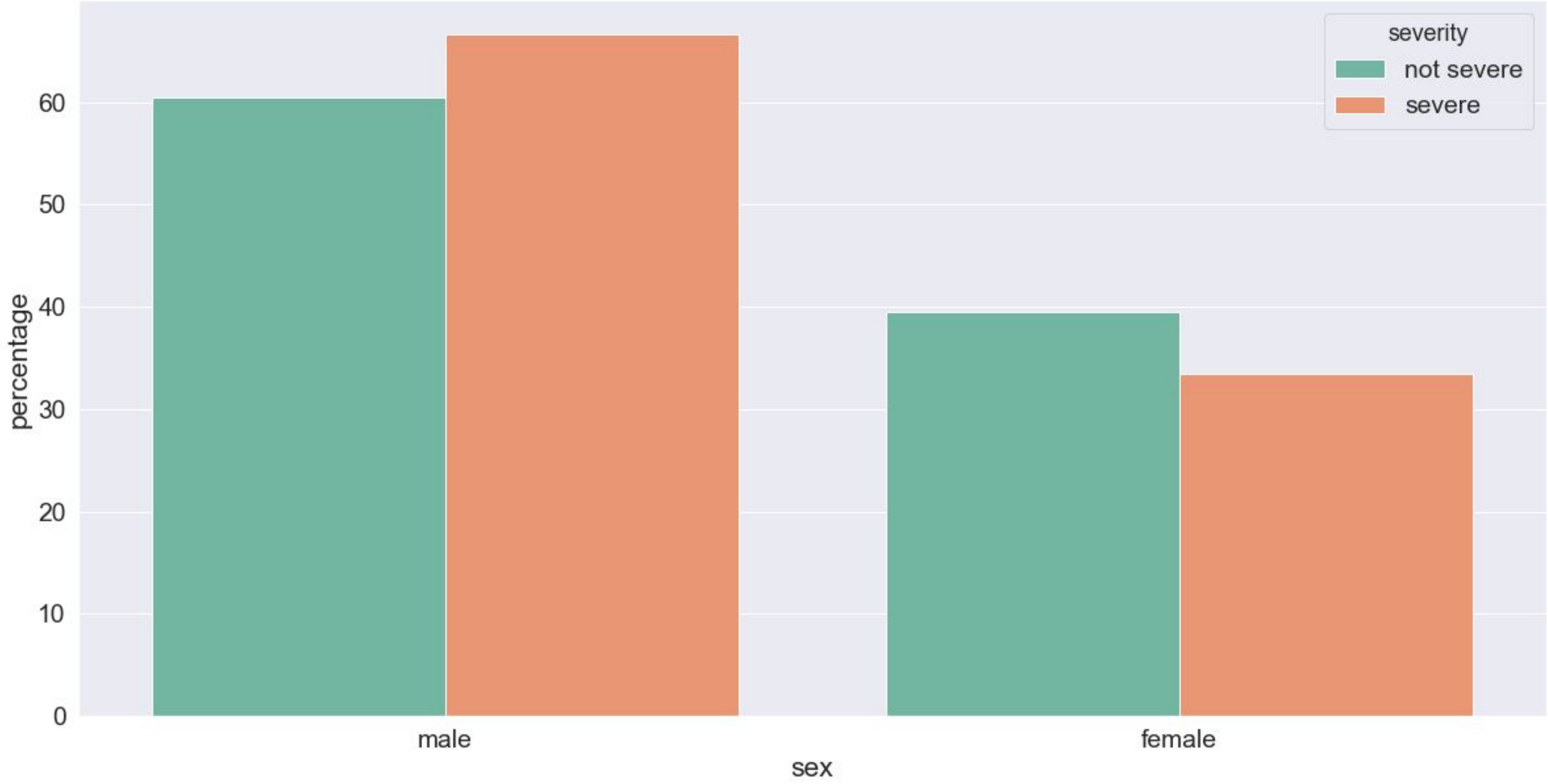
- 1.85% of accidents were serious.
- Split data into training and testing data.
- Used SMOTE to balance the training data

Exploratory Data Analysis

Age of Drivers in All Accidents



Sex of Drivers, Serious vs. Not Serious Accidents



Modeling

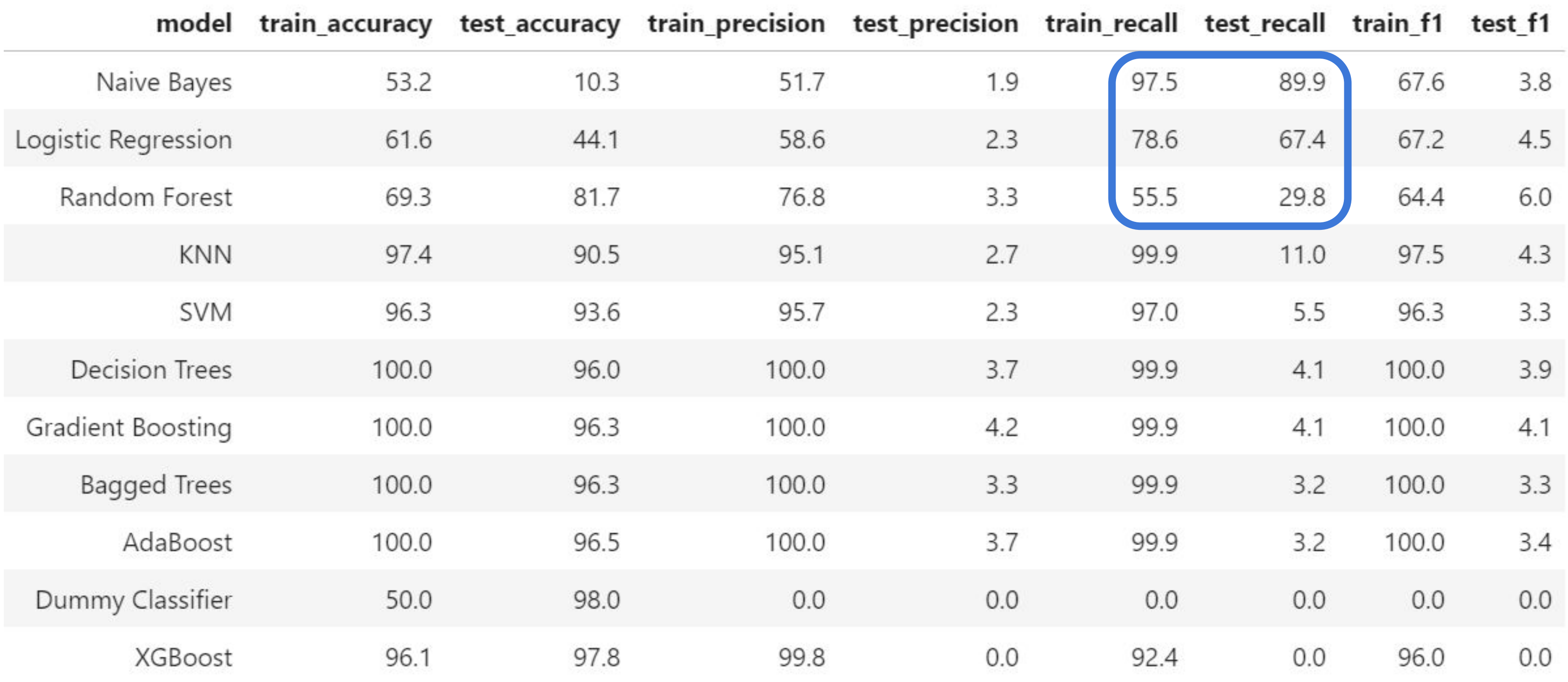
Initial Modeling

Ran 11 models with their default parameters

- Focus on **recall** instead of accuracy to prioritize serious accidents even if some non-serious accidents are misclassified.
- Looked to avoid overfitting the training data

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives}$$

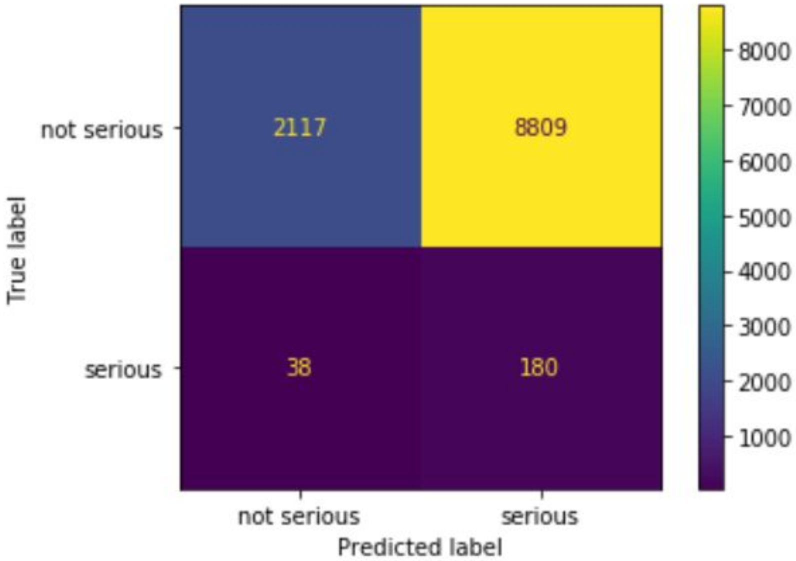
11 Models with Default Parameters

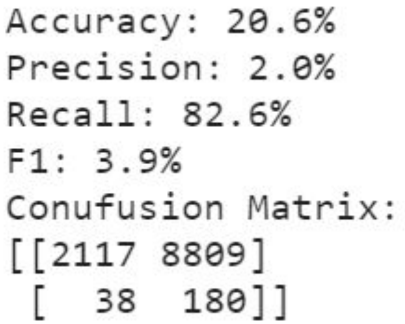


model	train_accuracy	test_accuracy	train_precision	test_precision	train_recall	test_recall	train_f1	test_f1
Naive Bayes	53.2	10.3	51.7	1.9	97.5	89.9	67.6	3.8
Logistic Regression	61.6	44.1	58.6	2.3	78.6	67.4	67.2	4.5
Random Forest	69.3	81.7	76.8	3.3	55.5	29.8	64.4	6.0
KNN	97.4	90.5	95.1	2.7	99.9	11.0	97.5	4.3
SVM	96.3	93.6	95.7	2.3	97.0	5.5	96.3	3.3
Decision Trees	100.0	96.0	100.0	3.7	99.9	4.1	100.0	3.9
Gradient Boosting	100.0	96.3	100.0	4.2	99.9	4.1	100.0	4.1
Bagged Trees	100.0	96.3	100.0	3.3	99.9	3.2	100.0	3.3
AdaBoost	100.0	96.5	100.0	3.7	99.9	3.2	100.0	3.4
Dummy Classifier	50.0	98.0	0.0	0.0	0.0	0.0	0.0	0.0
XGBoost	96.1	97.8	99.8	0.0	92.4	0.0	96.0	0.0

Naive Bayes

- Ran several models and used grid search
- `var_smoothing: 0.1` was best model but had too many accidents predicted as serious that were not serious to be useful.

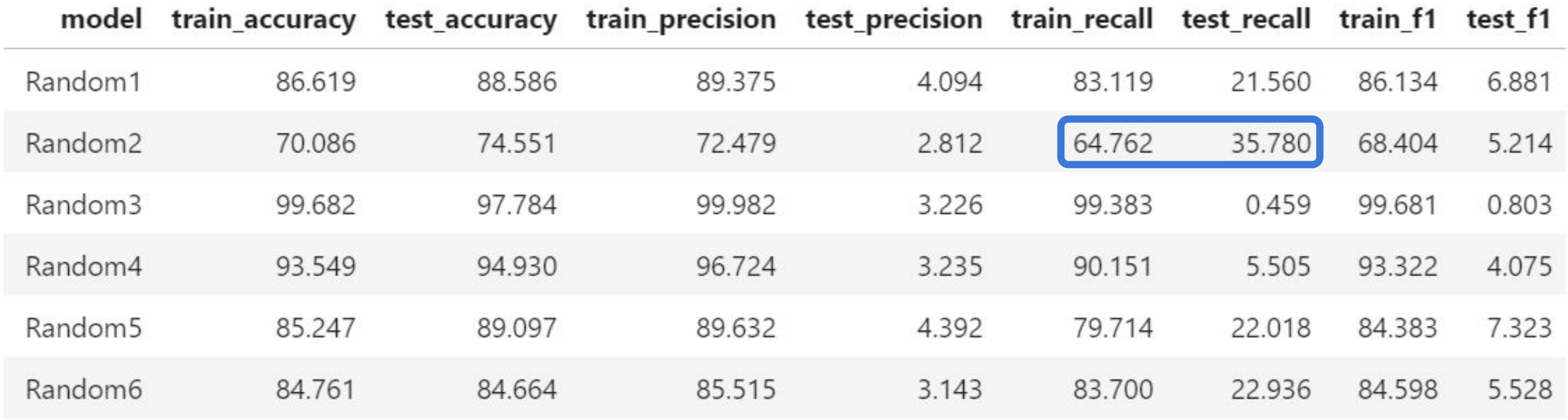




```
Accuracy: 20.6%
Precision: 2.0%
Recall: 82.6%
F1: 3.9%
Conufusion Matrix:
[[2117 8809]
 [  38  180]]
```

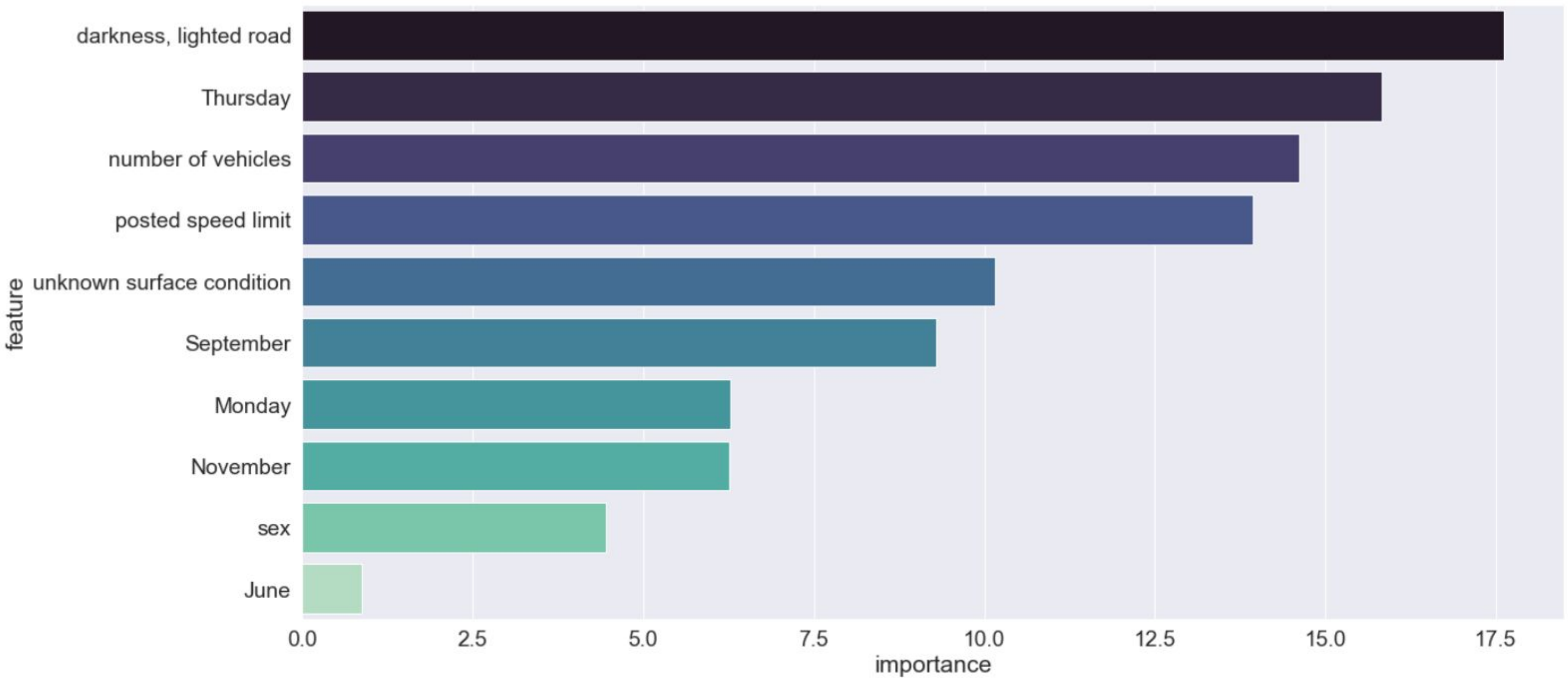
Random Forest

- Again, ran several models and used grid search.
- Model with params `max_depth=2, max_features=10, n_estimators=5` had best recall with least overfitting. Still, it had low recall score and overfitting problem.

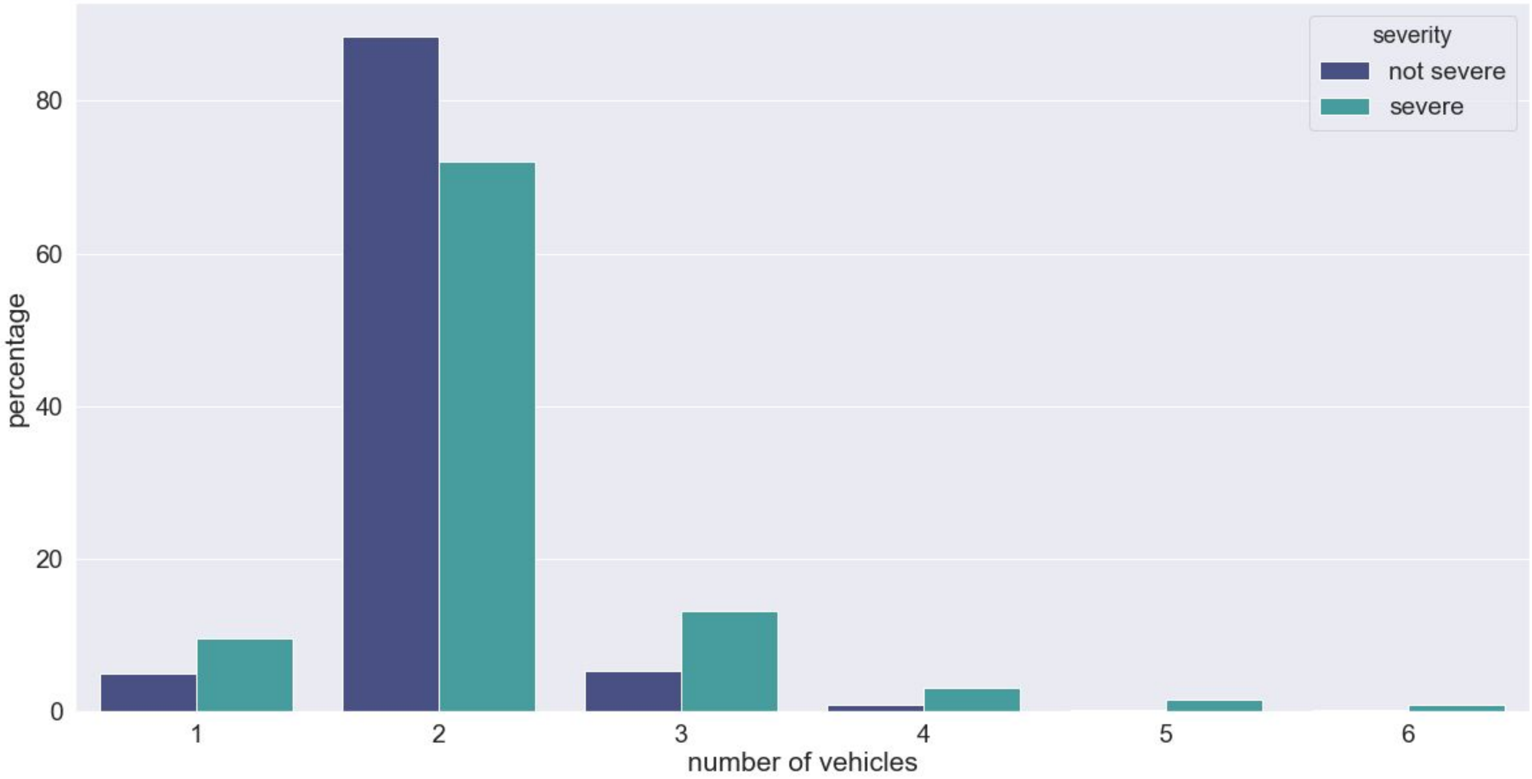


model	train_accuracy	test_accuracy	train_precision	test_precision	train_recall	test_recall	train_f1	test_f1
Random1	86.619	88.586	89.375	4.094	83.119	21.560	86.134	6.881
Random2	70.086	74.551	72.479	2.812	64.762	35.780	68.404	5.214
Random3	99.682	97.784	99.982	3.226	99.383	0.459	99.681	0.803
Random4	93.549	94.930	96.724	3.235	90.151	5.505	93.322	4.075
Random5	85.247	89.097	89.632	4.392	79.714	22.018	84.383	7.323
Random6	84.761	84.664	85.515	3.143	83.700	22.936	84.598	5.528

Important Features of Decision Tree Model



Number of Vehicles, Serious vs. Not Serious Accidents



Possible Additional Steps

- Models emphasized recall too much to be useful.
    - I would experiment with re-running models that took other metrics into account.
- Models had a problem with overfitting. When I sought to get the training data and the testing data closer for the Random Forest model, key metric decreased.
    - Again, I would experiment with re-running models to avoid overfitting and improve metrics.
- Use the entire dataset (with a more powerful computer)

Sources

- The City of Chicago’s Car Crash Data
    - https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3if
- The City of Chicago’s Data of People Involved in the Car Crashes
    - https://data.cityofchicago.org/Transportation/Traffic-Crashes-People/u6pd-qa9d

Contact

Henry Alpert
halpert3@gmail.com

LinkedIn: https://www.linkedin.com/in/henryalpert/

GitHub Project Repo:
https://github.com/halpert3/flatiron-mod2-project