Predicting the Severity of a Car Accident through Classification Models

By: Emilio Padron Molina

Introduction

- Now a days, car accidents are one of the most common types of accidents, and even though there are global and local efforts to decrease this statistic, the numbers are still alarming.
- Question: *Is there a way to predict a car accident before it happens?*
- Solving this question benefits companies that use navigation systems (ex. Uber, Google, and Rappi).
    - To optimize traveling time.
    - To prevent a potential accident.

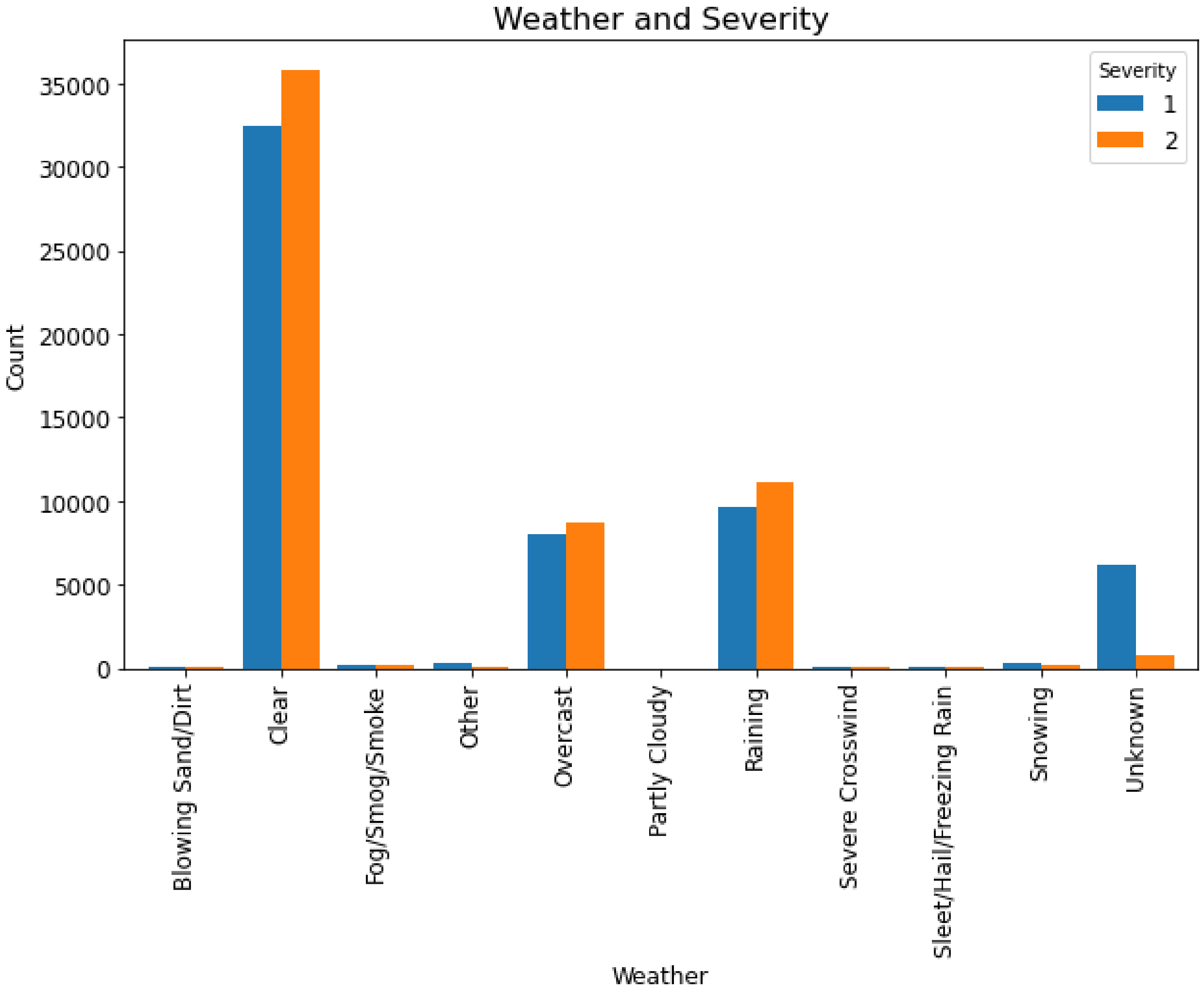
The Data Set

- Source: Seattle Police Department (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv)
- Data Description: reports from car collisions in the city in a tabular form, from January 1st, 2004- May 20th, 2020.
- Data shape: Rows (incidents) → 194,673; Columns (incident's characteristics) → 38.
- After initial data cleaning (including null values and duplicates):
  - Features: 9- collision type, alcohol/substance influences, weather, road conditions, address type, junction type, lack of attention, light conditions, and speeding
  - Target: Severity- values are 1 of 2 (2 being more severe).

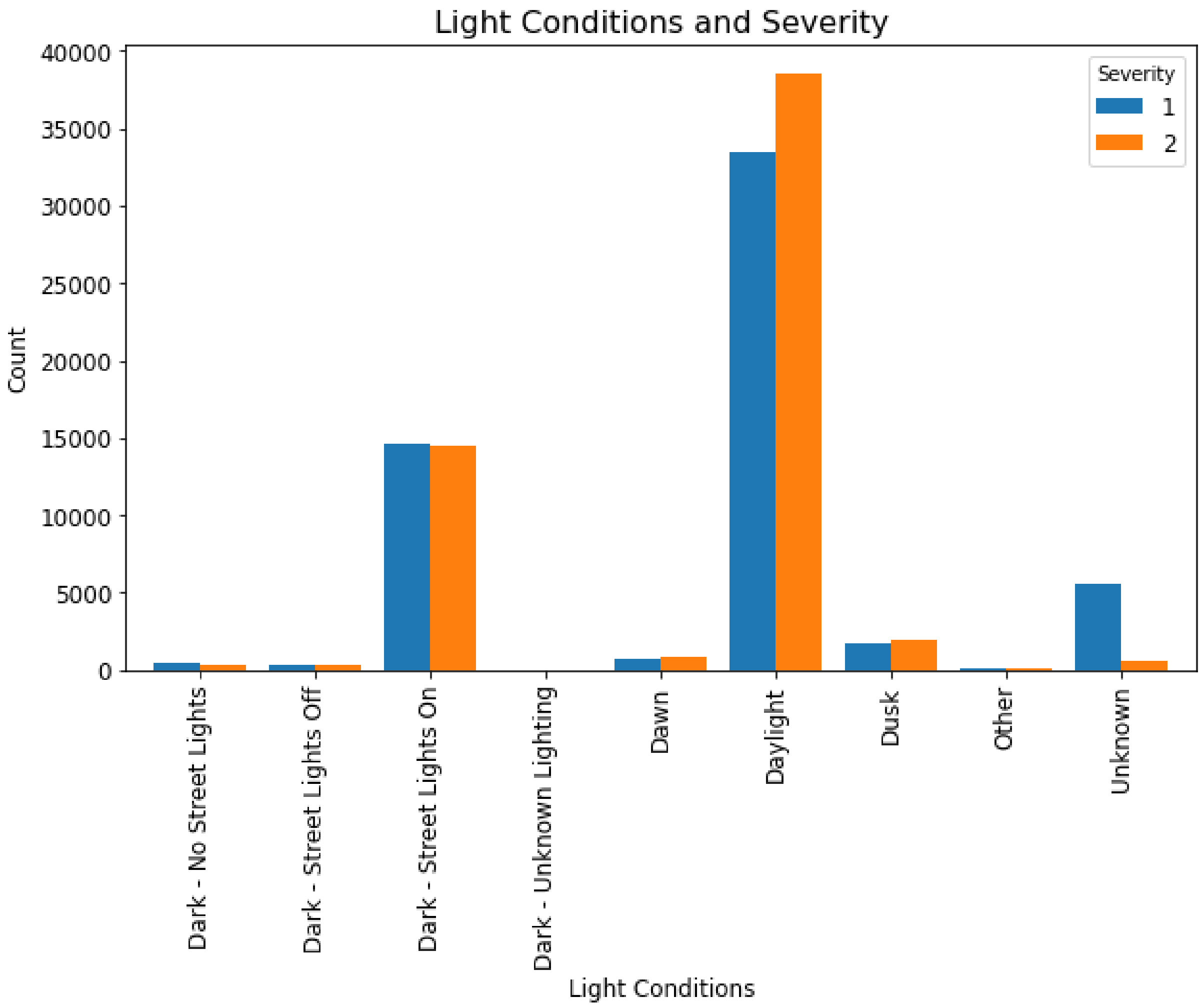
Exploratory Analysis

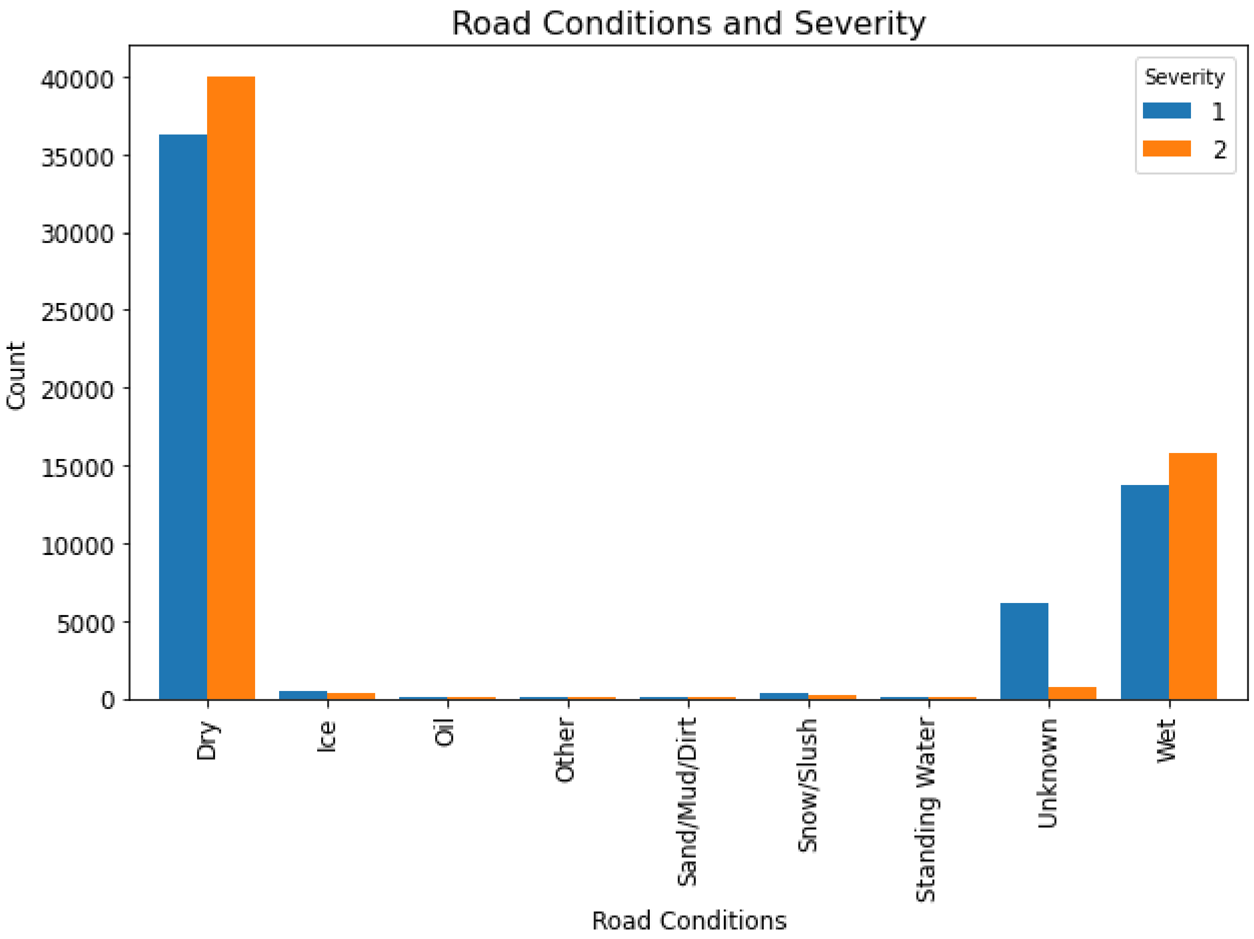
- Data Balancing: making the count of labels in the Severity column the same.

- Data Visualization: counts of the Severity labels for specific feature categories.



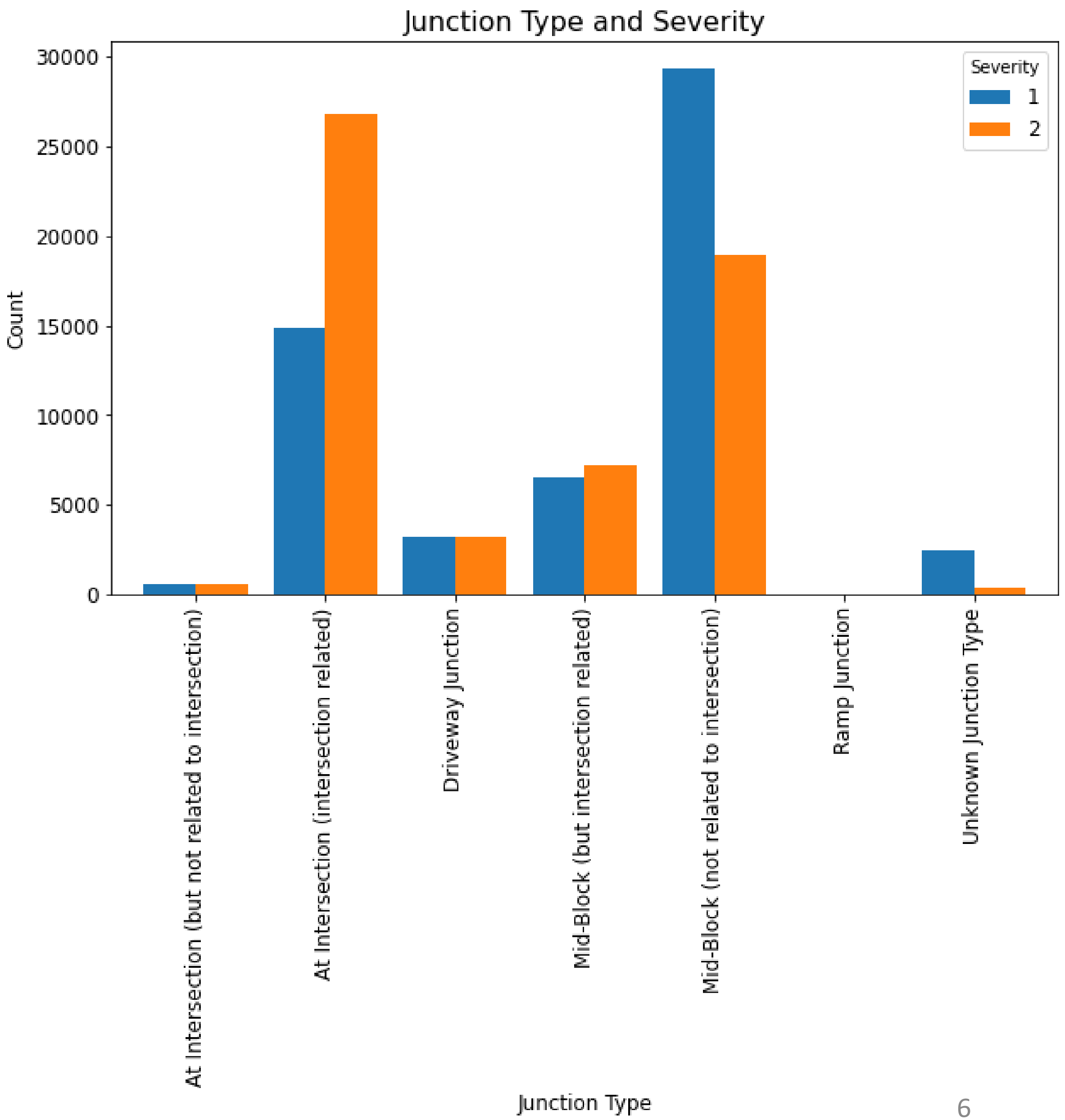
Exploratory Analysis



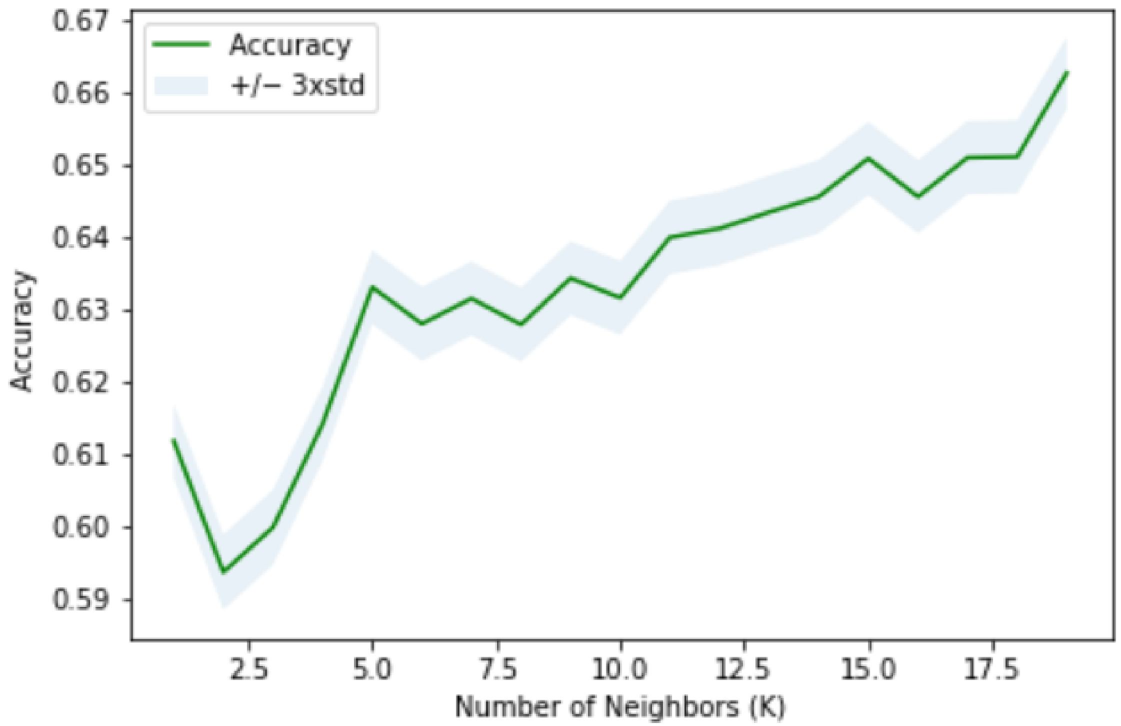


Exploratory Analysis

- Problem: unknow values favor a specific label in the junction type, weather, road conditions, and light conditions features.
- Solution: delete unknowns before balancing the data set.
- Given that we are dealing with categorical variables → get dummies



Classification Models



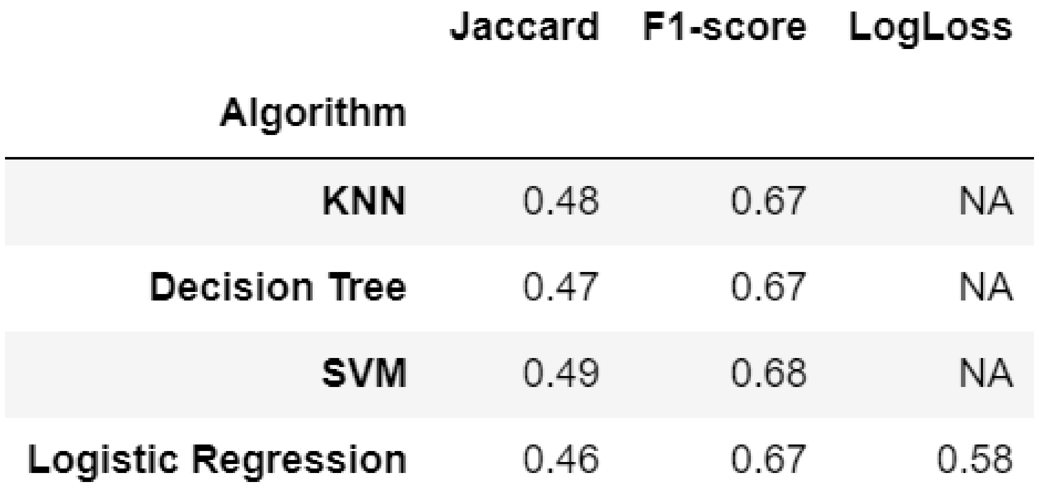
```
The best accuracy was with 0.6625759963972079 with k= 19
```

- **KNN**:
  - k=19 is selected.
  - For k>19 the accuracy doesn’t grow significantly → risk of overfitting.

Classification Models

- **Decision Tree**:
  - Criterion: gini
- **SVM**:
  - Kernel: RBF
  - C: 0.01
- **Logistic Regression**:
  - Solver: liblinear
  - C: 1

- Scores:



	Jaccard	F1-score	LogLoss
Algorithm			
KNN	0.48	0.67	NA
Decision Tree	0.47	0.67	NA
SVM	0.49	0.68	NA
Logistic Regression	0.46	0.67	0.58

Conclusion

- The SVM model proved to best model of all with a Jaccard score of 0.49 and F1-score of 0.68.
  - *Note*: the accuracy scores of this model were neither significantly different from the others nor large enough say it is performs well.
- Recommendations for improving the model:
  - Acquire more data with a target label of 2
  - Acquire more information from the incidents (ex. age of the driver, traffic conditions, type of the car).
  - Transform and analyze the time of the incident from the original set.