Predicting Vehicle Accident Severity

IBM Applied Data Science Capstone By Matt Guthrie

Introduction/Business Problem

- Problem: In 2018, more than 36,000 people were killed in vehicle accidents in the United States [1].
- Therefore, understanding the types of vehicle accidents that lead to injury and death is of great interest to a number of parties:
  - First responders and public safety officials (e.g. allocation of ambulances and traffic regulations)
  - Drivers and Pedestrians (e.g. deciding where and when to travel)

[1] https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812826

Introduction/Business Problem

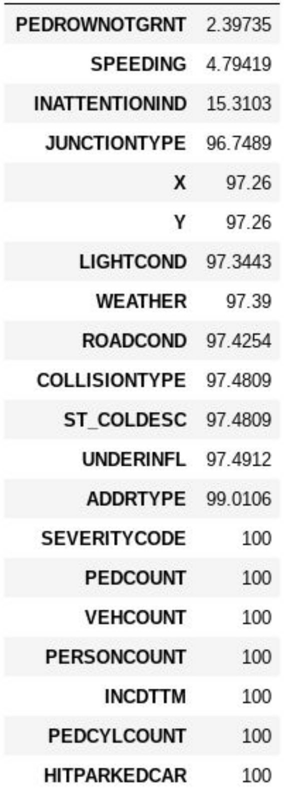
- Determining relationships between certain accident conditions and injury probability would allow interested parties (e.g. local governments) to avoid injuries by affecting input conditions (e.g. imposing traffic restrictions in certain areas or during certain times)

Data

- The data used for the model was obtained from the Seattle Department of Transportation [2]
- It contains 38 features of 194,673 vehicle collisions
- The target variable in this case is SEVERITYCODE, which originally takes values of 1 (no injury) or 2 (injury)

[2] https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf

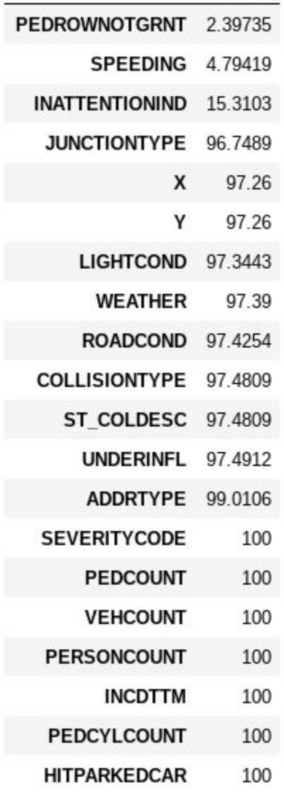
Data



PEDROWNOTGRNT	2.39735
SPEEDING	4.79419
INATTENTIONIND	15.3103
JUNCTIONTYPE	96.7489
X	97.26
Y	97.26
LIGHTCOND	97.3443
WEATHER	97.39
ROADCOND	97.4254
COLLISIONTYPE	97.4809
ST_COLDESC	97.4809
UNDERINFL	97.4912
ADDRTYPE	99.0106
SEVERITYCODE	100
PEDCOUNT	100
VEHCOUNT	100
PERSONCOUNT	100
INCDTTM	100
PEDCYLCOUNT	100
HITPARKEDCAR	100

- Out of the 37 remaining features, there are 20 features that appear useful for a model
- **Location data:** latitude and longitude data for the location of the vehicle accident is available
- **Environmental Conditions:** Weather, Light, Road conditions
- **Collision Type & Vehicles Involved**

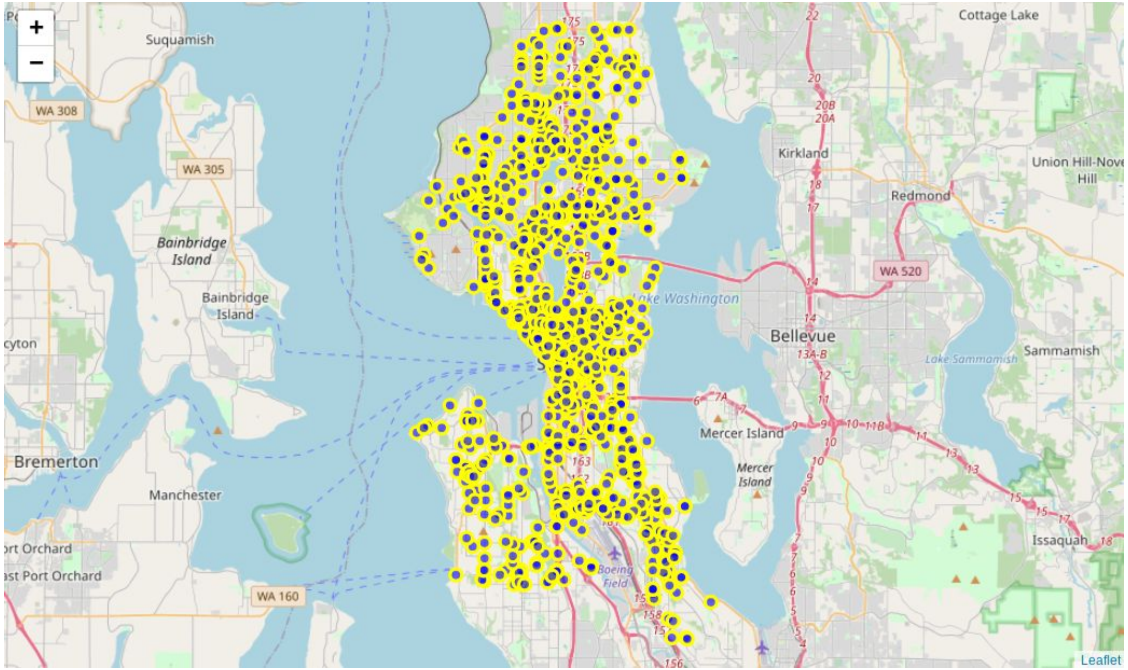
Methods - Missing Values



PEDROWNOTGRNT	2.39735
SPEEDING	4.79419
INATTENTIONIND	15.3103
JUNCTIONTYPE	96.7489
X	97.26
Y	97.26
LIGHTCOND	97.3443
WEATHER	97.39
ROADCOND	97.4254
COLLISIONTYPE	97.4809
ST_COLDESC	97.4809
UNDERINFL	97.4912
ADDRTYPE	99.0106
SEVERITYCODE	100
PEDCOUNT	100
VEHCOUNT	100
PERSONCOUNT	100
INCDTTM	100
PEDCYLCOUNT	100
HITPARKEDCAR	100

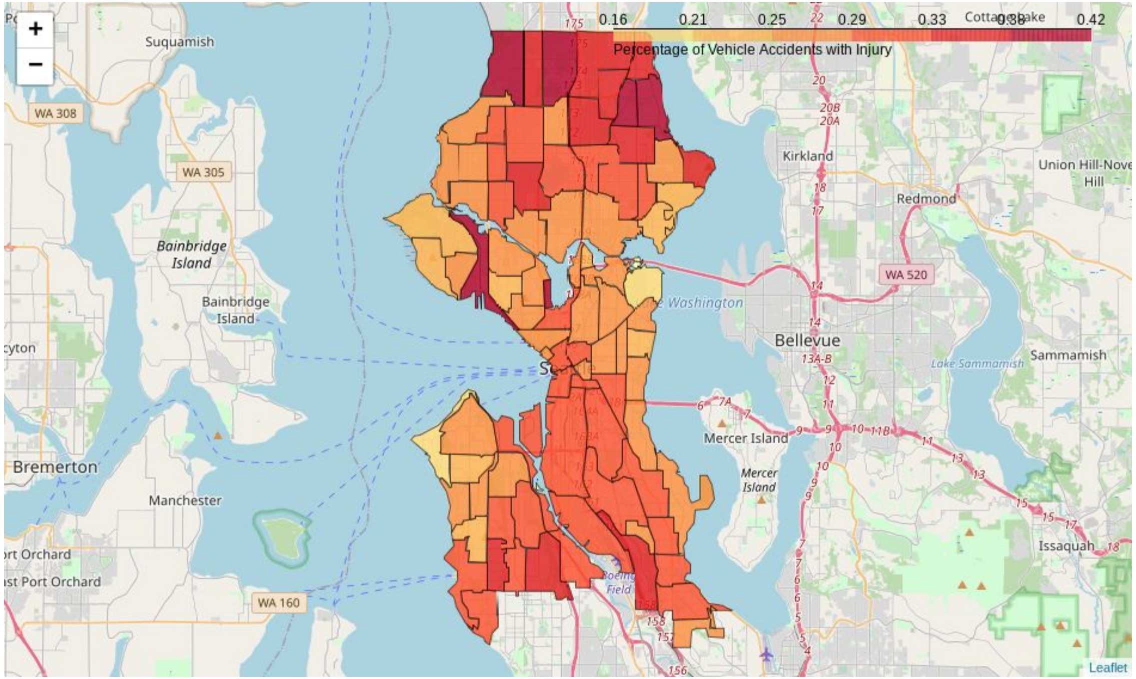
- The first step was to check the degree of missingness for our initial set of features
- Most features are >96% populated, but the first three seem to be mostly missing
- After treating the missing rows as negative values, a chi-squared test shows all three variables to have a significant relationship with SEVERITYCODE. Cramer’s V statistic shows that PEDROWNOTGRNT has most significant relationship

Methods - Location Data



- The X and Y coordinates can be plotted using the Folium package, but individual coordinates aren't of much use.

Methods - Location Data

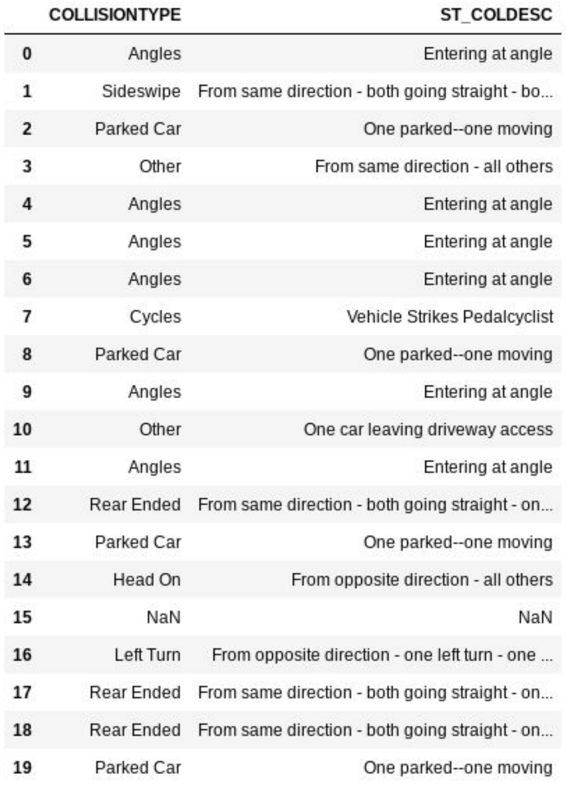


- Using a geojson file from Seattle’s Open Data Portal, we can map the coordinates to neighborhoods

Methods - Categorical Features

- Many of the categorical features that have more than two levels were reduced down to flags (1/0) to remove unnecessary features once we one-hot encoded them.
- For example, the WEATHER feature was mapped to WEATHER_IMPAIRED which indicates whether the weather conditions could impair the driver (e.g. Snowing).

Methods - Categorical Features

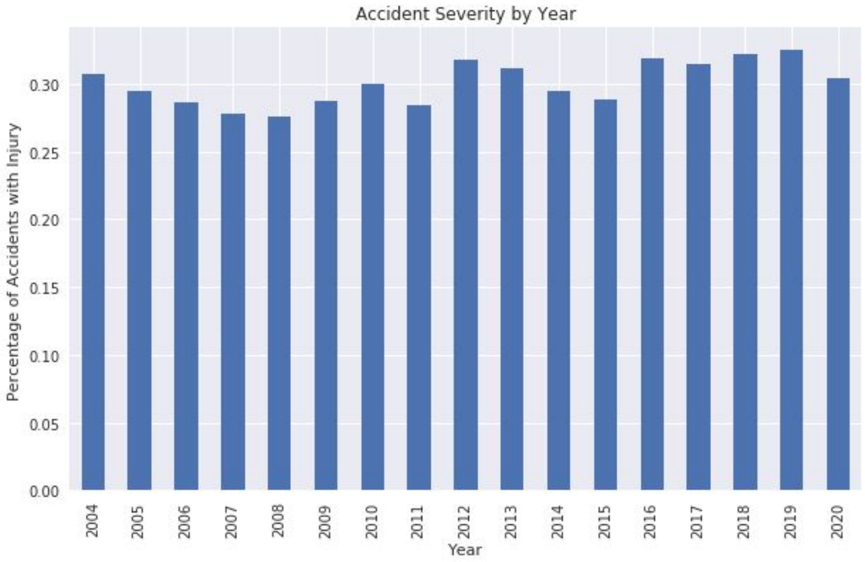


	COLLISIONTYPE	ST_COLDESC
0	Angles	Entering at angle
1	Sideswipe	From same direction - both going straight - bo...
2	Parked Car	One parked--one moving
3	Other	From same direction - all others
4	Angles	Entering at angle
5	Angles	Entering at angle
6	Angles	Entering at angle
7	Cycles	Vehicle Strikes Pedalcyclist
8	Parked Car	One parked--one moving
9	Angles	Entering at angle
10	Other	One car leaving driveway access
11	Angles	Entering at angle
12	Rear Ended	From same direction - both going straight - on...
13	Parked Car	One parked--one moving
14	Head On	From opposite direction - all others
15	NaN	NaN
16	Left Turn	From opposite direction - one left turn - one ...
17	Rear Ended	From same direction - both going straight - on...
18	Rear Ended	From same direction - both going straight - on...
19	Parked Car	One parked--one moving

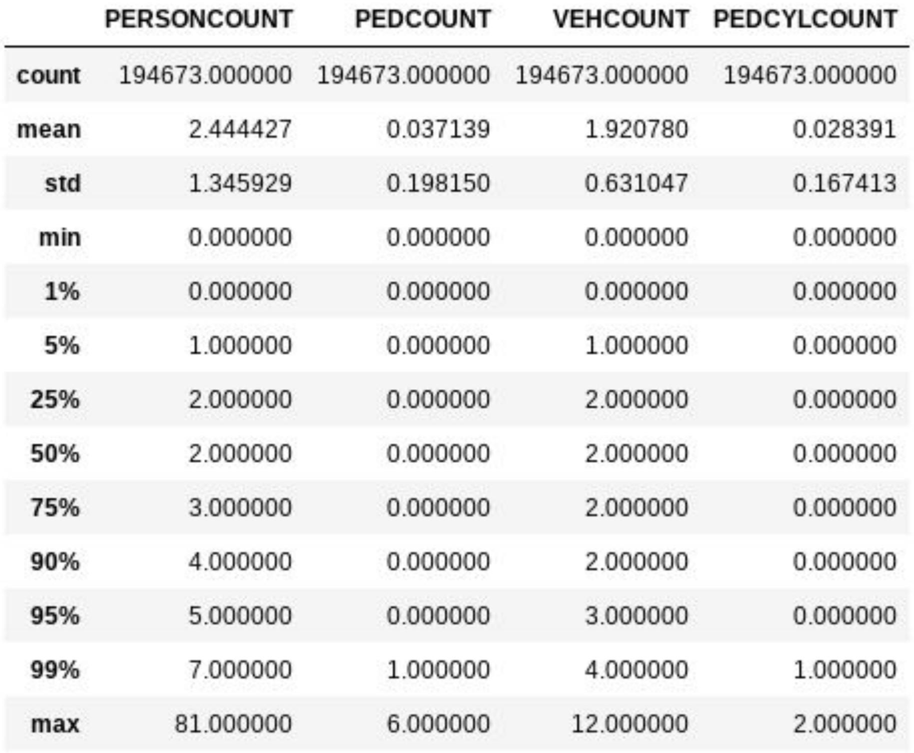
- Other features that appear to give similar information were compared and the most useful features were chosen.
- For example, COLLISIONTYPE and ST_COLDESC both convey information on the type of vehicle collision, but COLLISIONTYPE has less levels while retaining the similar information.

Methods - Categorical Features

- The dataset includes a feature for the date and time of the accident, INCDTTM, but the time is missing from many of the rows.
- From the date, YEAR and MONTH features were extracted



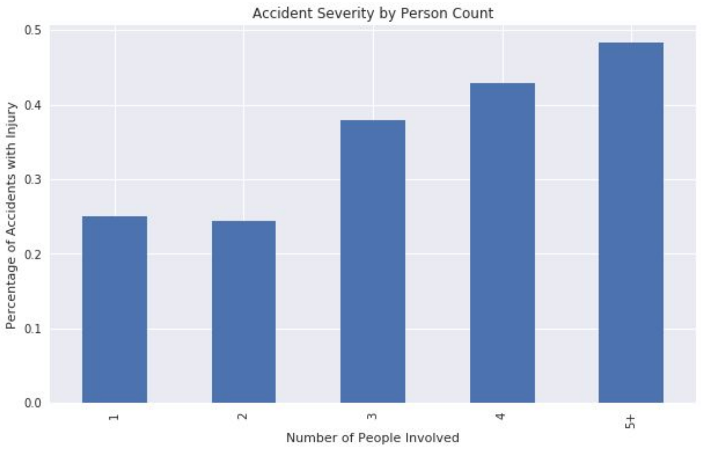
Methods - Numerical Features

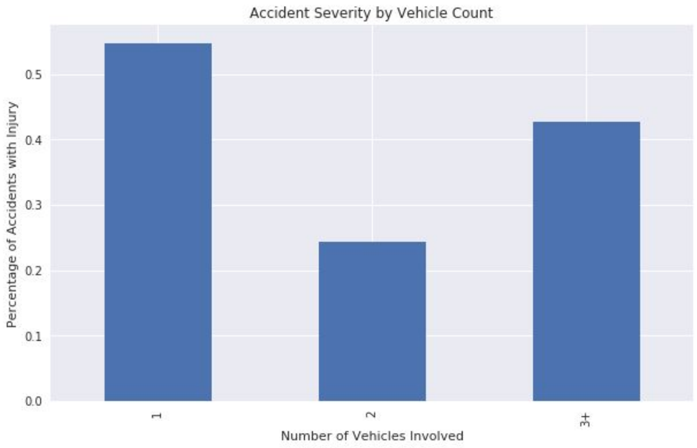


	PERSONCOUNT	PEDCOUNT	VEHCOUNT	PEDCYLCOUNT
count	194673.000000	194673.000000	194673.000000	194673.000000
mean	2.444427	0.037139	1.920780	0.028391
std	1.345929	0.198150	0.631047	0.167413
min	0.000000	0.000000	0.000000	0.000000
1%	0.000000	0.000000	0.000000	0.000000
5%	1.000000	0.000000	1.000000	0.000000
25%	2.000000	0.000000	2.000000	0.000000
50%	2.000000	0.000000	2.000000	0.000000
75%	3.000000	0.000000	2.000000	0.000000
90%	4.000000	0.000000	2.000000	0.000000
95%	5.000000	0.000000	3.000000	0.000000
99%	7.000000	1.000000	4.000000	1.000000
max	81.000000	6.000000	12.000000	2.000000

- PEDCOUNT (# of Pedestrians) and PEDCYLCOUNT (# of Bicycles) are mostly 0 so these features were converted to flags
- Accidents with PERSONCOUNT or VEHCOUNT equal to 0 were dropped since the accidents must include at least one person or vehicle. These features were then transformed to categorical features.

Methods - Numerical Features

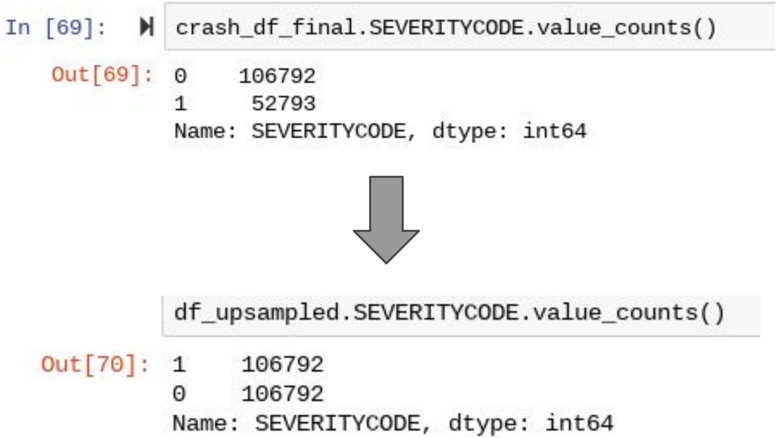




- The proportion of accidents with injury increases with the number of people involves, but the relation is more complex for vehicle count

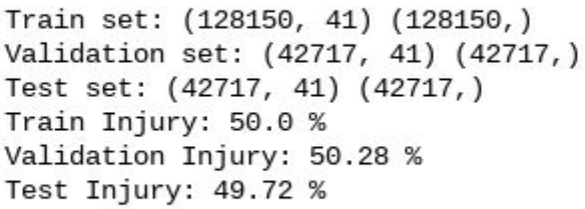
Methods - Dataset Finalization

- After removing rows with unusable or missing data, we retain ~82% of the accidents in the original dataset
- Categorical features are one-hot encoded for model building
- The target variable is originally unbalanced, so we can upsample the minority rows to balance the dataset



```
In [69]:  crash_df_final.SEVERITYCODE.value_counts()

Out[69]:  0    106792
          1     52793
          Name: SEVERITYCODE, dtype: int64
```

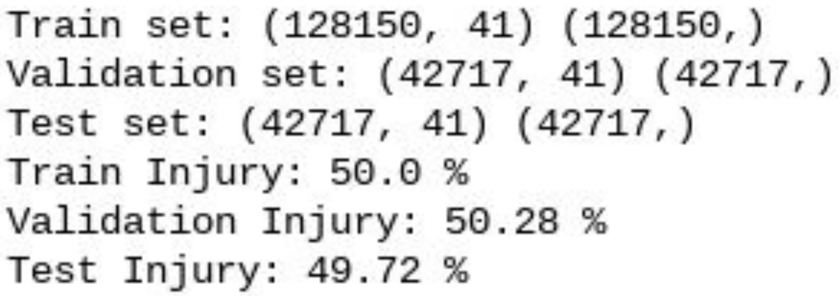
```
          df_upsampled.SEVERITYCODE.value_counts()

Out[70]:  1    106792
          0    106792
          Name: SEVERITYCODE, dtype: int64
```



```
Train set: (128150, 41) (128150,)
Validation set: (42717, 41) (42717,)
Test set: (42717, 41) (42717,)
Train Injury: 50.0 %
Validation Injury: 50.28 %
Test Injury: 49.72 %
```

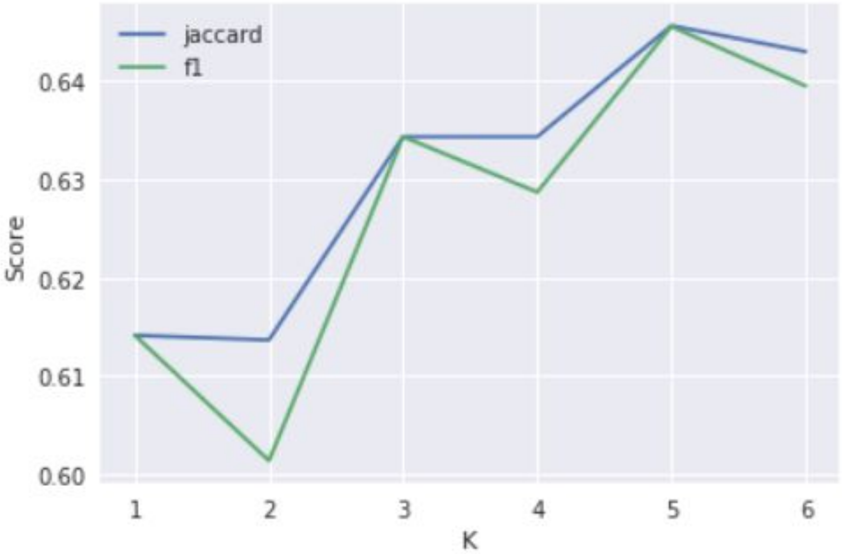
Methods - Dataset Finalization

Finally, we split the data into train, validate, and test sets (60:20:20)



```
Train set: (128150, 41) (128150,)
Validation set: (42717, 41) (42717,)
Test set: (42717, 41) (42717,)
Train Injury: 50.0 %
Validation Injury: 50.28 %
Test Injury: 49.72 %
```

Methods - Model Building



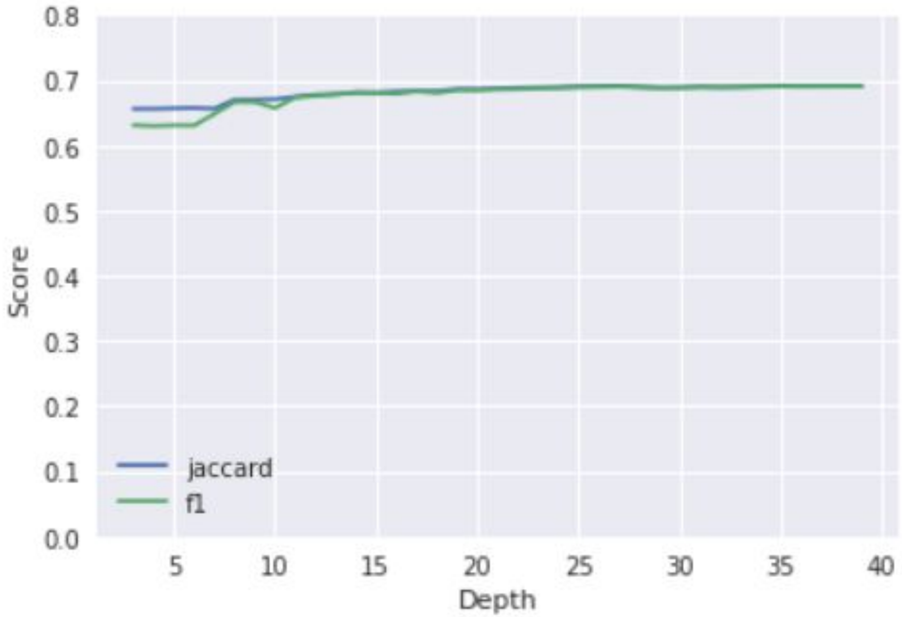
K-Nearest Neighbor (KNN)
Using the validation set we can see that K=5 gets the best performance

Methods - Model Building

Logistic Regression
Using the validation set we can select the best value for C (the inverse of the regularization)

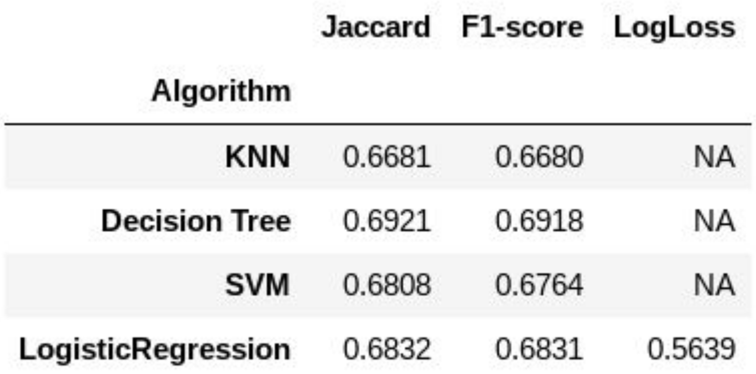
Support Vector Machine (SVM)
Using the validation set we can see that the rbf (radial basis function) kernel gets the best performance

Methods - Model Building



Decision Tree
Using the validation set we can see that the best performance is achieved closer to max depth but only small marginal gains are achieved past a depth of around 15

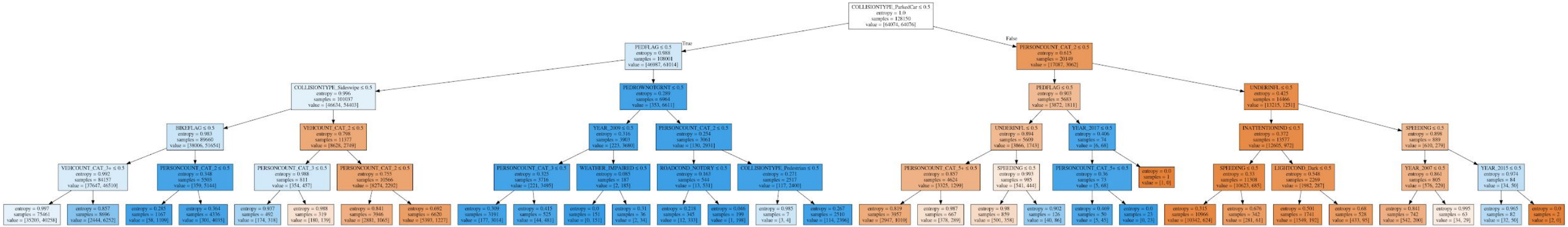
Results



	Jaccard	F1-score	LogLoss
Algorithm			
KNN	0.6681	0.6680	NA
Decision Tree	0.6921	0.6918	NA
SVM	0.6808	0.6764	NA
LogisticRegression	0.6832	0.6831	0.5639

Putting all of the results together, we can see that the Decision Tree model performs best on the test data

Results


COLLISIONTYPE_ParkedCar ≤ 0.5
entropy = 1.0
samples = 128150
value = [64074, 64076]
True
False
PEDFLAG ≤ 0.5
entropy = 0.988
samples = 108001
value = [46987, 61014]
PERSONCOUNT_CAT_2 ≤ 0.5
entropy = 0.615
samples = 20149
value = [17087, 3062]
COLLISIONTYPE_Sideswipe ≤ 0.5
entropy = 0.996
samples = 101037
value = [46634, 54403]
PEDROWNOTGRNT ≤ 0.5
entropy = 0.289
samples = 6964
value = [353, 6611]
PEDFLAG ≤ 0.5
entropy = 0.903
samples = 5683
value = [3872, 1811]
UNDERINFL ≤ 0.5
entropy = 0.425
samples = 14466
value = [13215, 1251]
BIKEFLAG ≤ 0.5
entropy = 0.983
samples = 89660
value = [38006, 51654]
VEHCOUNT_CAT_2 ≤ 0.5
entropy = 0.798
samples = 11377
value = [8628, 2749]
YEAR_2009 ≤ 0.5
entropy = 0.316
samples = 3903
value = [223, 3680]
PERSONCOUNT_CAT_2 ≤ 0.5
entropy = 0.254
samples = 3061
value = [130, 2931]
UNDERINFL ≤ 0.5
entropy = 0.894
samples = 5609
value = [3866, 1743]
YEAR_2017 ≤ 0.5
entropy = 0.406
samples = 74
value = [6, 68]
INATTENTIONIND ≤ 0.5
entropy = 0.372
samples = 13577
value = [12605, 972]
SPEEDING ≤ 0.5
entropy = 0.898
samples = 889
value = [610, 279]
VEHCOUNT_CAT_3+ ≤ 0.5
entropy = 0.992
samples = 84157
value = [37647, 46510]
PERSONCOUNT_CAT_2 ≤ 0.5
entropy = 0.348
samples = 5503
value = [359, 5144]
PERSONCOUNT_CAT_3 ≤ 0.5
entropy = 0.988
samples = 811
value = [354, 457]
PERSONCOUNT_CAT_2 ≤ 0.5
entropy = 0.755
samples = 10566
value = [8274, 2292]
PERSONCOUNT_CAT_3 ≤ 0.5
entropy = 0.325
samples = 3716
value = [221, 3495]
WEATHER_IMPAIRED ≤ 0.5
entropy = 0.085
samples = 187
value = [2, 185]
ROADCOND_NOTDRY ≤ 0.5
entropy = 0.163
samples = 544
value = [13, 531]
COLLISIONTYPE_Pedestrian ≤ 0.5
entropy = 0.271
samples = 2517
value = [117, 2400]
PERSONCOUNT_CAT_5+ ≤ 0.5
entropy = 0.857
samples = 4624
value = [3325, 1299]
SPEEDING ≤ 0.5
entropy = 0.993
samples = 985
value = [541, 444]
PERSONCOUNT_CAT_5+ ≤ 0.5
entropy = 0.36
samples = 73
value = [5, 68]
entropy = 0.0
samples = 1
value = [1, 0]
SPEEDING ≤ 0.5
entropy = 0.33
samples = 11308
value = [10623, 685]
LIGHTCOND_Dark ≤ 0.5
entropy = 0.548
samples = 2269
value = [1982, 287]
YEAR_2007 ≤ 0.5
entropy = 0.861
samples = 805
value = [576, 229]
YEAR_2015 ≤ 0.5
entropy = 0.974
samples = 84
value = [34, 50]
entropy = 0.997
samples = 75461
value = [35203, 40258]
entropy = 0.857
samples = 8696
value = [2444, 6252]
entropy = 0.285
samples = 1167
value = [58, 1109]
entropy = 0.364
samples = 4336
value = [301, 4035]
entropy = 0.937
samples = 492
value = [174, 318]
entropy = 0.988
samples = 319
value = [180, 139]
entropy = 0.841
samples = 3946
value = [2881, 1065]
entropy = 0.692
samples = 6620
value = [5393, 1227]
entropy = 0.309
samples = 3191
value = [177, 3014]
entropy = 0.415
samples = 525
value = [44, 481]
entropy = 0.0
samples = 151
value = [0, 151]
entropy = 0.31
samples = 36
value = [2, 34]
entropy = 0.218
samples = 345
value = [12, 333]
entropy = 0.046
samples = 199
value = [1, 198]
entropy = 0.985
samples = 7
value = [3, 4]
entropy = 0.267
samples = 2510
value = [114, 2396]
entropy = 0.819
samples = 3957
value = [2947, 1010]
entropy = 0.987
samples = 667
value = [378, 289]
entropy = 0.98
samples = 859
value = [501, 358]
entropy = 0.902
samples = 126
value = [40, 86]
entropy = 0.469
samples = 50
value = [5, 45]
entropy = 0.0
samples = 23
value = [0, 23]
entropy = 0.315
samples = 10966
value = [10342, 624]
entropy = 0.676
samples = 342
value = [281, 61]
entropy = 0.501
samples = 1741
value = [1549, 192]
entropy = 0.68
samples = 528
value = [433, 95]
entropy = 0.841
samples = 742
value = [542, 200]
entropy = 0.995
samples = 63
value = [34, 29]
entropy = 0.965
samples = 82
value = [32, 50]
entropy = 0.0
samples = 2
value = [2, 0]

Discussion

- Generally, the choice when working with data is between prediction and inference, i.e. do we simply want to predict the value of the target variable given the feature set or would we like to learn about the relationships between certain features and the target variable?

- This dataset (and this problem) lend themselves to a more inferenced-based approach, however, many of the tools used in machine learning are targeted toward prediction.

- In our case, we were able to develop a model that does fairly well at prediction, but predicting the severity of an accident given all the other features of the accident has fairly limited uses. For example, if you know how many people are involved in the accident, you would presumably already know if any of those people are injured.

- Meanwhile, the best performing model in terms of prediction, a decision tree model, makes it a bit more difficult to make inferences about feature relationships.

Conclusion

In conclusion, there are many things we could improve in future projects:

- More feature engineering/transformation, e.g. some of the features we simplified to 1/0 flags may contain more important info with more levels
- Using more of the available features, e.g. the location data is useful, but there were too many levels to work with without a significant amount of effort
- Adding new features that have less obvious relationships with the target variable, e.g. adding a feature like vehicle model would allow use to check for the significance of the model while controlling for all other features
- More feature selection in models like the Logistic Regression model, e.g. in this case we used all features in the final model even if they were not very significant