Predicting Severity of Car Crashes

Balwinder Khakh

October 10, 2020

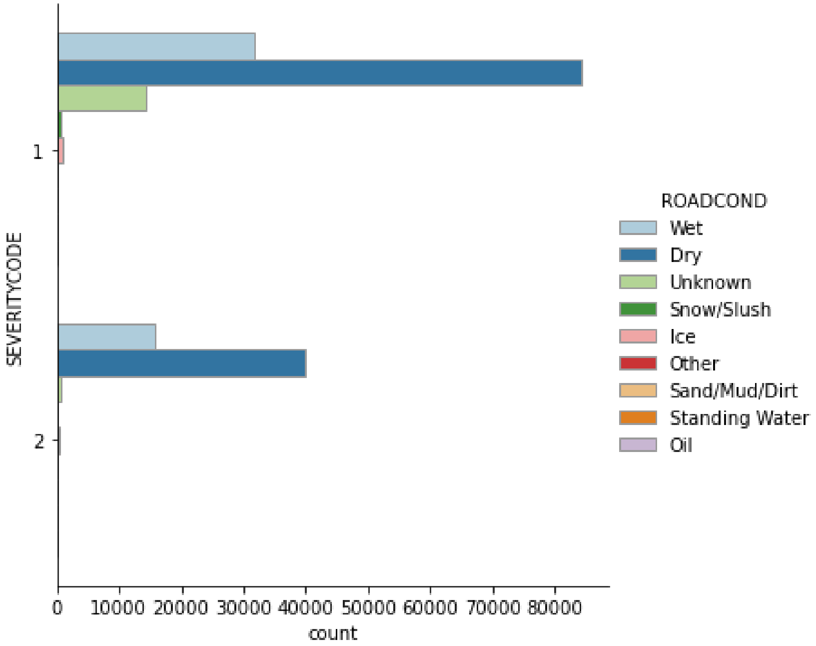
Introduction

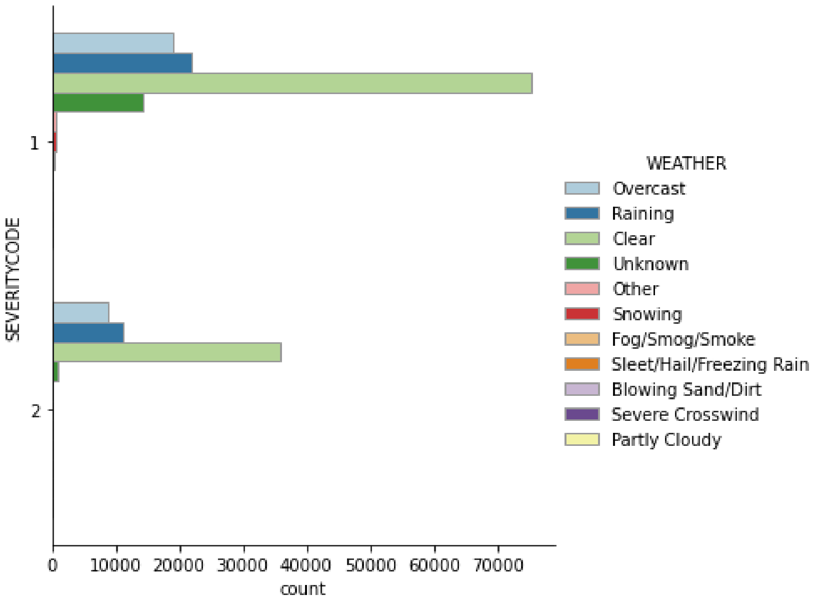
- The city of Seattle in the state of Washington has open, available data on various crashes that have occurred over the years
- Using machine learning algorithms it is possible to analyze this data and find key common factors that can predict how severe a potential accident can be

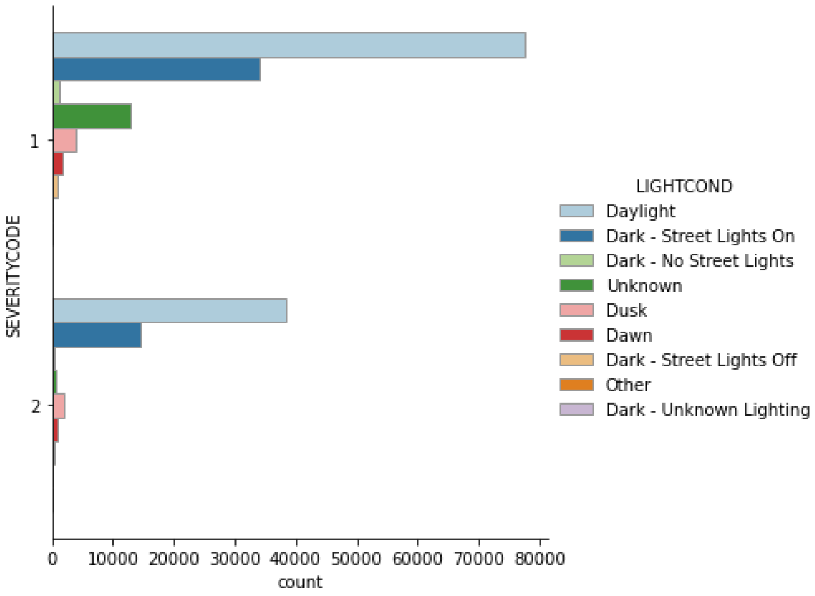
Data Acquisition and Cleaning

```
SEVERITYCODE        0
SEVERITYDESC        0
INCDATE             0
INCDTTM             0
WEATHER          5081
ROADCOND         5012
LIGHTCOND        5170
dtype: int64
```

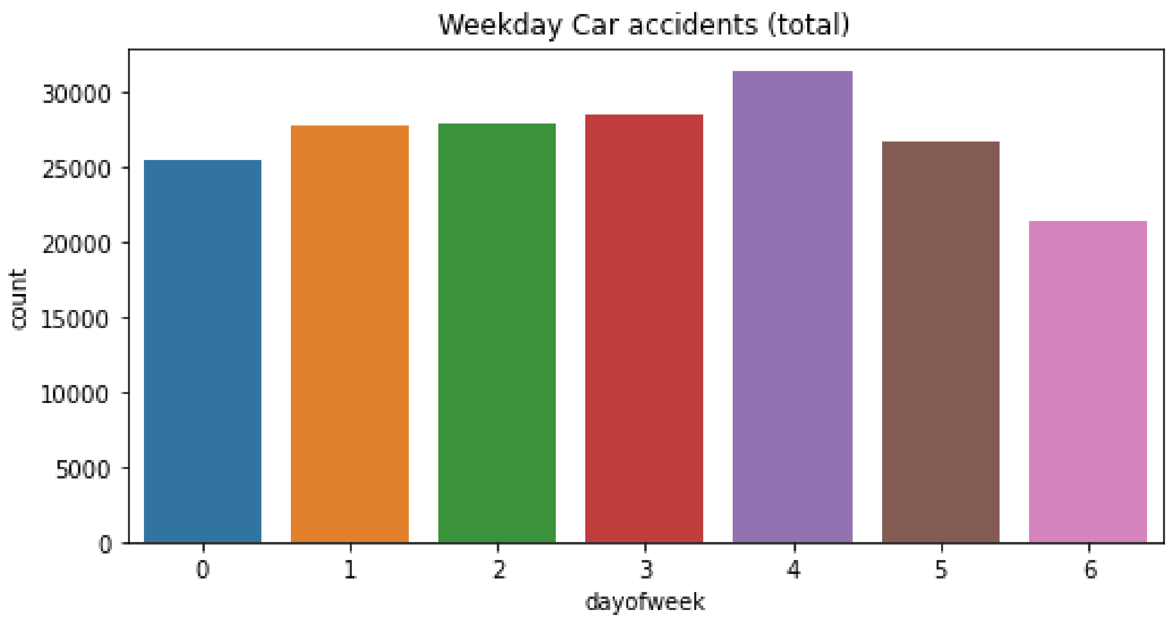
- The data used in this project is hosted on https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv
- The goal is to use more of the universal factors in this analysis
- The variables shown above were ultimately selected

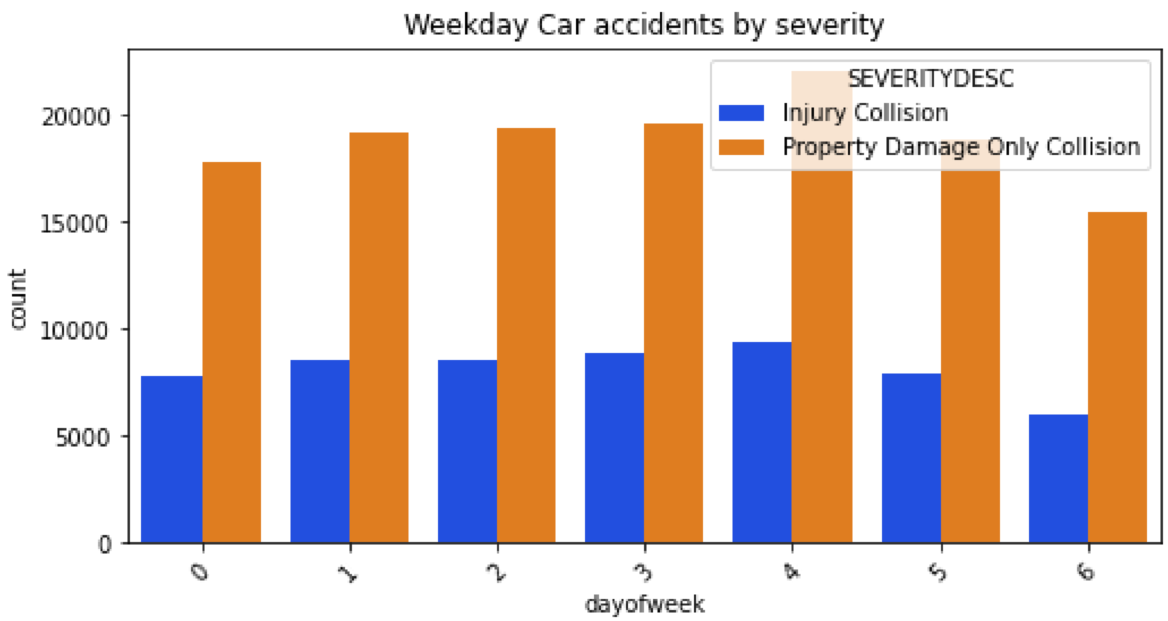




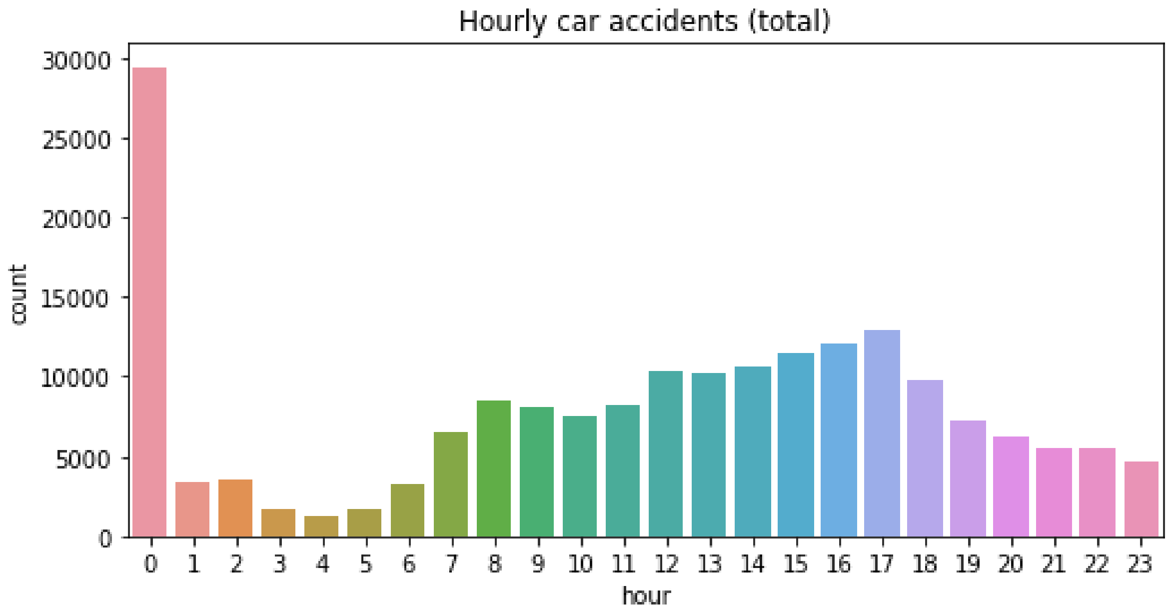


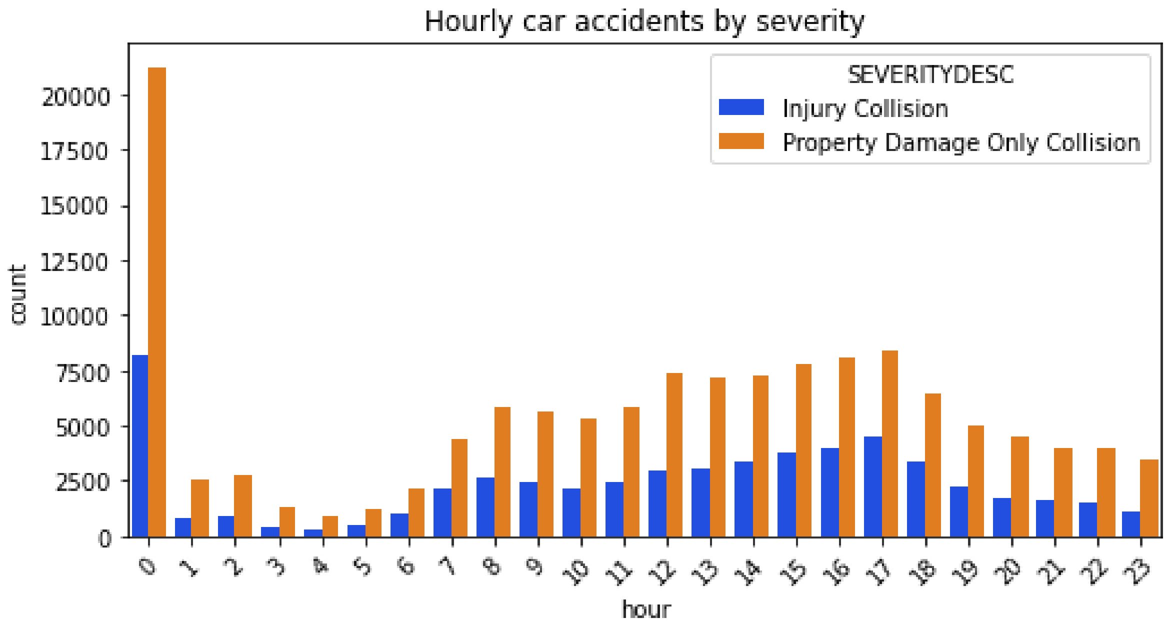
Exploratory Data Analysis-Conditions





Exploratory Data Analysis-Day of the week





Exploratory Data Analysis-Hour

```
[17]: from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
tree.fit(X_train, y_train)
tree
```

```
[17]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=4,
                 max_features=None, max_leaf_nodes=None,
                 min_impurity_decrease=0.0, min_impurity_split=None,
                 min_samples_leaf=1, min_samples_split=2,
                 min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                 splitter='best')
```

```
[16]: from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
lr = LogisticRegression(C=0.0001, solver='liblinear')
lr.fit(X_train, y_train)
lr
```

```
[16]: LogisticRegression(C=0.0001, class_weight=None, dual=False,
          fit_intercept=True, intercept_scaling=1, max_iter=100,
          multi_class='warn', n_jobs=None, penalty='l2', random_state=None,
          solver='liblinear', tol=0.0001, verbose=0, warm_start=False)
```

```
[15]: from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import f1_score
from sklearn.metrics import log_loss
from sklearn.metrics import precision_score

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 8).fit(X_train, y_train)
knn
```

```
[15]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=8, p=2,
           weights='uniform')
```

Predictive Modeling
KNN, Logistic Regression, Decision Tree

Model Results

THE JACCARD RESULTS AND PRECISION

	Algorithm	Jaccard	Precision
0	KNN	0.68	0.6
1	Decision Tree	0.7	0.49
2	Logistic Regression	0.7	0.49

THE OVERALL ACCURACY OF THE MODELS

```
Train set KNN Accuracy:  0.6834421096314182
Test set KNN Accuracy:  0.6789725713883314
Train set Decission Tree Accuracy:  0.6990530803184064
Test set Decission Tree Accuracy:  0.6977923312559416
Train set Logistic regression Accuracy:  0.6990530803184064
Test set Logistic regression Accuracy:  0.6977923312559416
```

Conclusions

- The insight from the exploratory analysis reveals that, at least in the case of Seattle, the majority of accidents are property damage that occur under ideal conditions in the day time
- If the results found in this report apply to other cities, local government might want to focus more effort in accident reduction during peak or normal driving times.