Car accident severity

Djibril ISSOUFOU MAMAN

September 25, 2020

1. Introduction

1.1: Background

Driving a car always comes with the risk of an accident. Over 5 million car accidents occur every year in the United States of America. These accidents are in varying degrees of severity, with some only resulting in light injuries whereas others causing death.

Car accidents happen due to a variety of causes, be it traffic, or speeding, or even bad weather. These factors combine with the level of carelessness of the driver in deciding how severe or dangerous a car accident is.

1.2: Problem

The causes of accidents are extremely varied and thus we cannot account for all of them on our own whenever getting on a vehicle. Accidents result not only in a loss on the part of the occupants of the cars in terms of medical bills and injuries, but also involve insurance payments, loss of productivity and time. If an accident could be anticipated for and prevented before it occurs, a large amount of time and money would be saved, mainly for the occupants of the vehicle.

Thus the problem to be solved is: **Can the severity of a car accident be predicted?**

To do this, we need to answer the following questions:

1. What and how much data is needed?
2. What factors must be taken into consideration?
3. How much influence do each of these factors have on the target of prediction?
4. What model is to be used?
5. How accurate is the model used?

1.3: Interest

This problem would be of interest to people who regularly travel in their own vehicles, as learning of the possibility of an accident might induce them to be more careful while driving, avoiding injuries, damage or loss of life. The problem would also be of interest to the city traffic police department, so that they can gauge the possible severity of an accident and respond in kind to prevent it beforehand. Finally, insurance companies would also benefit from not having to pay auto or health insurance for preventable accidents.

2. Data

2.1: Source

The data used for this project is a database of the details of accidents in **Seattle City**, Washington, United States of America, including types and severity.

The organization providing this data is **SDOT Traffic Management Division, Traffic Records Group** and the database is updated weekly.

A description of the data can be found here.
The data can be downloaded here.

2.2: Description

The data consists of 38 columns of which a majority could be dropped. Columns containing incident keys, police codes, lane and street keys were all dropped to narrow it down to relevant columns. Columns with large amounts of missing data, or columns which cannot be adjusted properly were dropped. The number of people involved, weather, speeding status, whether the driver was inattentive, whether the driver was under the influence, and the road and light conditions are among the relevant columns to the problem of predicting severity.

The data was analyzed for all relevant columns using visualization to determine the relationship with the severity. Then the relevant columns were tested, before and after balancing the data, with multiple possible machine learning models. Each model was evaluated using multiple metrics. Then finally the model with the best overall accuracy was considered and used to predict the severity of an accident.

An example of this data would be an accident that happened on 25th April 2020 on NW 65th ST at a block. The dataset tells us it was of severity 1, meaning only property damage occured, and also tells us that it hit a parked car. The vehicle had 2 people in it with no pedestrians involved. The dataset also tells us that the weather was overcast, with the road being wet and that the lighting condition was dark with street lights on. We also know that the vehicle was not speeding.

3. Methodology

3.1: The Dataset

Dropping Columns

First, all columns which were descriptive, had police codes, or had incident keys were removed. These were:

- OBJECTID
- SEVERITYDESC
- INCKEY
- COLDETKEY
- REPORTNO
- STATUS
- INTKEY
- EXCEPTRSNCODE
- EXCEPTRSNDESC
- SDOT_COLCODE
- SDOT_COLDESC
- SDOTCOLNUM
- ST_COLCODE
- ST_COLDESC
- SEGLANEKEY
- CROSSWALKKEY

SEVERITYCODE.1 was removed because the SEVERITYCODE column has the same data.
INCDTTM was removed in favor of INCDATE as INCDATE had more consistent values.
LOCATION was removed as X and Y hold the same information and it was repetitive.
PEDCOUNT and PEDCYLCOUNT have a majority of the data being 0, and thus they do not create much of an impact on the prediction of SEVERITYCODE. PERSONCOUNT also overlaps with these two fields, creating redundancy. Thus both PEDCOUNT and PEDCYLCOUNT were dropped.

Replacing Missing Values

Next, the missing values for each column were tallied, and PEDROWNOTGRNT was dropped because it had a majority of missing values.

The following columns had no missing values: SEVERITYCODE, PERSONCOUNT, VEHCOUNT, INCDATE, HITPARKEDCAR.

Then, the missing values in the remaining columns were replaced: By the value with maximum frequency: ADDRTYPE, UNDERINFL, WEATHER, ROADCOND, LIGHTCOND. COLLISIONTYPE was replaced with 'Other'. JUNCTIONTYPE was replaced with 'Unknown' as data is unknown. In the columns INATTENTIONIND and SPEEDING, the missing value meant 'N' (No) as only values for 'Y' (Yes) were given.

Cleaning Data

The dataset now had columns with no missing values. However, the existing data still had to be cleaned and organized.

The rows with value 'Unknown' were dropped in the columns JUNCTIONTYPE, WEATHER, ROADCOND and LIGHTCOND.

The LIGHTCOND field had repetitive data framed in multiple ways, and thus the repetitive data was replaced with a consistent category.

The UNDERINFL field had values 0,1,N and Y. 0 and 1 were replaced with N and Y respectively.

Converting text to numbers where easily possible

In the columns SPEEDING, UNDERINFL, INATTENTIONIND, and HITPARKEDCAR: the values N and Y were replaced with 0 and 1 respectively.

Converting to proper datatype

Finally, all the columns were converted to proper datatypes (int, float, datetime or object).

List of Fields remaining in Dataset

- SEVERITYCODE
- X
- Y
- ADDRTYPE
- COLLISIONTYPE
- PERSONCOUNT
- VEHCOUNT
- INCDATE
- JUNCTIONTYPE

- INATTENTIONIND
- UNDERINFL
- WEATHER
- ROADCOND
- LIGHTCOND
- SPEEDING
- HITPARKEDCAR

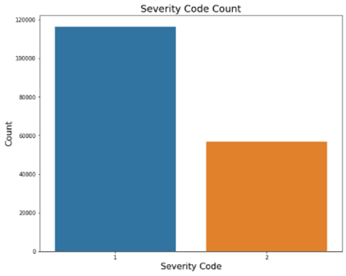
3.2: Creating Alternate Datasets

dfnumbers

This is the original dataset with all categorical variables converted to numerical variables for three purposes.

1. Building a proper correlation matrix to compare correlations.
2. Building a Multiple Linear Regression model to analyse the data and check if the model would work.
3. For effective usage during visualization.

The columns containing categorical variables were converted to numerical, being given whole number values from 1 to 10. These columns were WEATHER, ROADCOND, LIGHTCOND, ADDRTYPE, COLLISIONTYPE and JUNCTIONTYPE. The INCDATE column was dropped as it is not numerical in nature.



Data for Machine Learning

As seen from the visualization, the existing dataset was highly unbalanced, with around double the number of collisions of Severity Code 1, corresponding to property damage, than Severity Code 2, corresponding to Injury. This would result in a flawed or biased model, and so had to be fixed.
The possible methods to fix this would be:

1. Undersampling (choosing equal number of samples of each Severity Code randomly to generate a model)
2. Oversampling (creating random data of samples with Severity Code 2 to match the number of samples with Severity Code 1)
3. Building a biased model

I attempted all three methods to check which one has the best accuracy.

For oversampling, a method called SMOTE was used from the library 'imbalanced-learn'. First a new dataframe titled 'dfoversampled' was created dropping the columns X and Y as location values would not need to be considered in the prediction, and were only used for visualization. Then SMOTE was used to generate samples of severity code 2 from the existing data. This dataframe was called dfoversampled, and was then assigned column names. This dataset had equal number of values

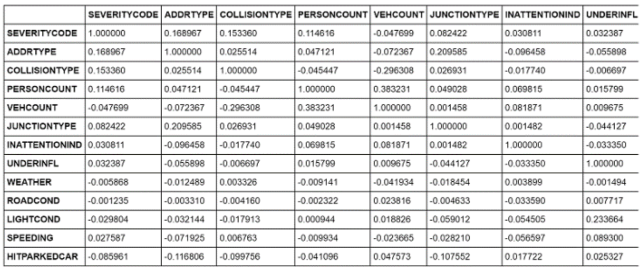
(116458) of SEVERITYCODE 1 and 2. The contents of the dataframe were then rounded to integers to maintain categorical data and the datatype was changed to int64 from float64.

For undersampling, we did the processing before machine learning using the method 'NearMiss' from the library 'imbalanced-learn'. This method directly gave us two numpy arrays with the undersampled data. This results in an equal number of values (56629) of SEVERITYCODE 1 and 2. There is no further modification of the arrays required other than preprocessing.

For building a biased model, we use the existing dataset: dfnumbers.

3.3: Data Analysis

Correlation Matrix



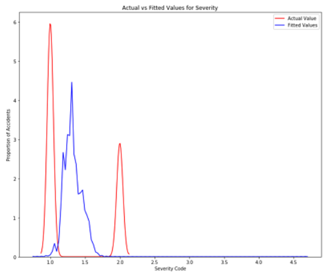
	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	VEHCOUNT	JUNCTIONTYPE	INATTENTIONIND	UNDERINFL
SEVERITYCODE	1.000000	0.168967	0.153360	0.114616	-0.047699	0.082422	0.030811	0.032387
ADDRTYPE	0.168967	1.000000	0.025514	0.047121	-0.072367	0.209585	-0.096458	-0.055898
COLLISIONTYPE	0.153360	0.025514	1.000000	-0.045447	-0.296308	0.026931	-0.017740	-0.006697
PERSONCOUNT	0.114616	0.047121	-0.045447	1.000000	0.383231	0.049028	0.069815	0.015799
VEHCOUNT	-0.047699	-0.072367	-0.296308	0.383231	1.000000	0.001458	0.081871	0.009675
JUNCTIONTYPE	0.082422	0.209585	0.026931	0.049028	0.001458	1.000000	0.001482	-0.044127
INATTENTIONIND	0.030811	-0.096458	-0.017740	0.069815	0.081871	0.001482	1.000000	-0.033350
UNDERINFL	0.032387	-0.055898	-0.006697	0.015799	0.009675	-0.044127	-0.033350	1.000000
WEATHER	-0.005868	-0.012489	0.003326	-0.009141	-0.041934	-0.018454	0.003899	-0.001494
ROADCOND	-0.001235	-0.003310	-0.004160	-0.002322	0.023816	-0.004633	-0.033590	0.007717
LIGHTCOND	-0.029804	-0.032144	-0.017913	0.000944	0.018826	-0.059012	-0.054505	0.233664
SPEEDING	0.027587	-0.071925	0.006763	-0.009934	-0.023665	-0.028210	-0.056597	0.089300
HITPARKEDCAR	-0.085961	-0.116806	-0.099756	-0.041096	0.047573	-0.107552	0.017722	0.025327

This correlation matrix was created on the dfnumbers dataset and shows that no two columns have a high postive or negative correlation. This is to be expected as there are only two possible values for SEVERITYCODE and a limited number of unique values in most columns. As the data is categorical and not continuous, it is very likely that there is no measurable direct correlation between the data.

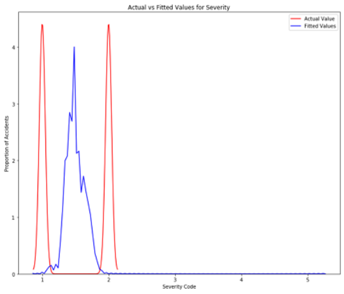
Multiple Linear Regression

A multiple linear regression model was created on the dfnumbers dataset before and after balancing the data. In both cases, the model was extremely inaccurate, owing to the reasons explained above. The R2 score of both models was under 0.1, meaning that a multiple regression model would not effectively fit the data. A visualization of both models is visible.

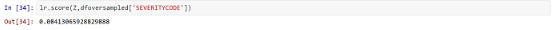
dfnumbers:



```
In [24]: lr.score(Z,dfnumbers['SEVERITYCODE'])
Out[24]: 0.07480986541670931
```



dfoversampled:

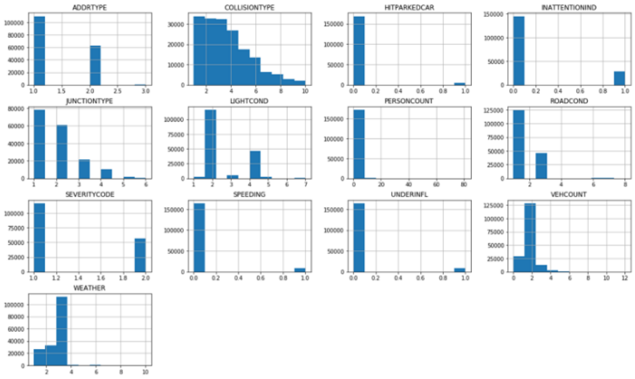


```
In [34]: lr.score(Z,dfoversampled['SEVERITYCODE'])
Out[34]: 0.08413065928829888
```

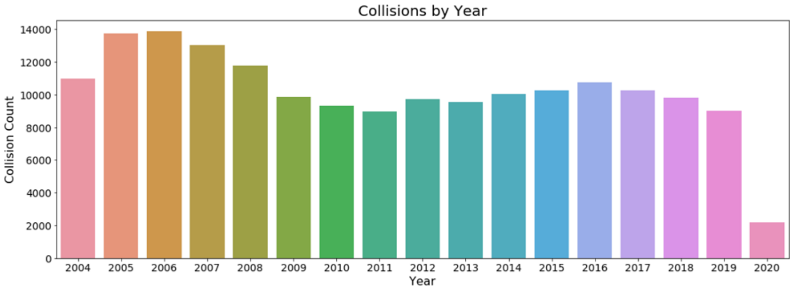
3.4 Data Visualization

Distribution of Data Values in each column

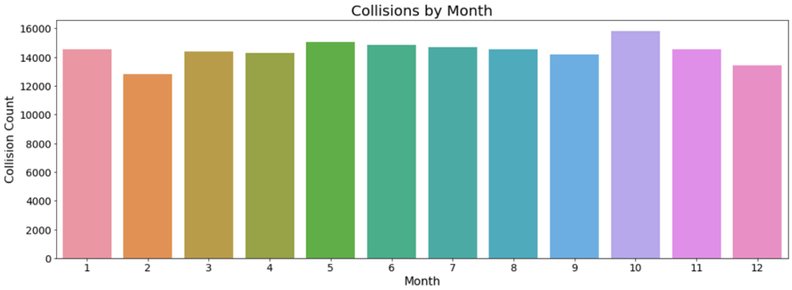
The following histogram matrix shows the numerical data distribution in each category.



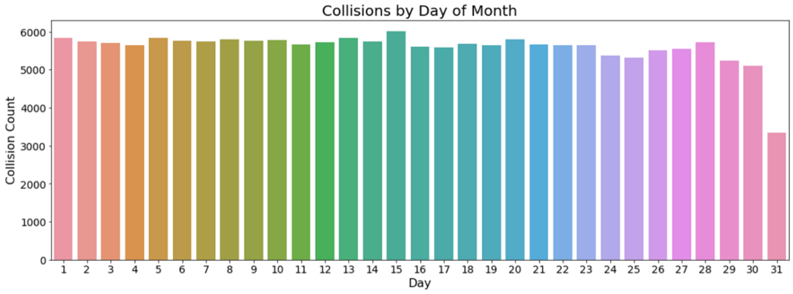
Distributions of Collisions by Year, Month and Day



Year:



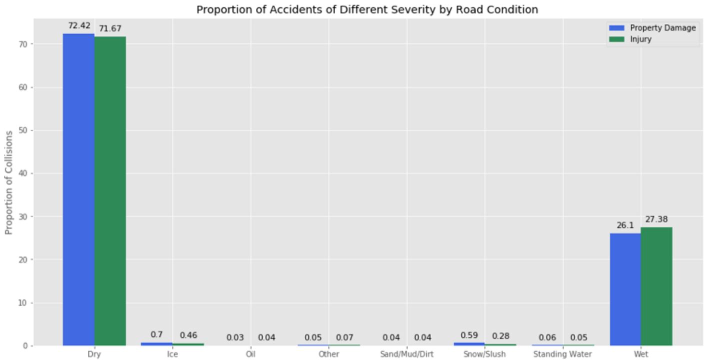
Month:



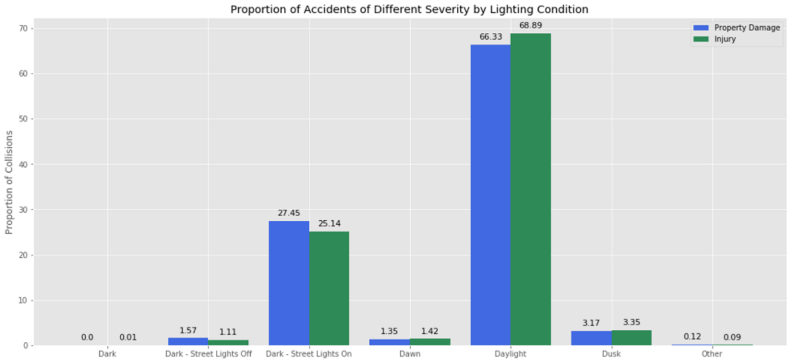
Day:

The bar graph shows an even distribution of collisions throughout every year, with a peak in 2006. However, the collisions in 2020 are considerably lower, possibly owing to the pandemic and a lack of data collection. Similarly, the collisions are almost evenly distributed by month as well, but have a small peak in October. Finally, the distribution by day is also relatively even, with a peak in the middle of the month. The number of collisions on the 31st is noticeable lower, but can be assumed to be due to the fact that not all months have a 31st. However, the distribution shows a lower number of collisions on the 29th and 30th as well, possibly leading to a conclusion that fewer accidents occur at the end of each month.

Proportion of Severity of Collisions by Road, Light and Weather Conditions

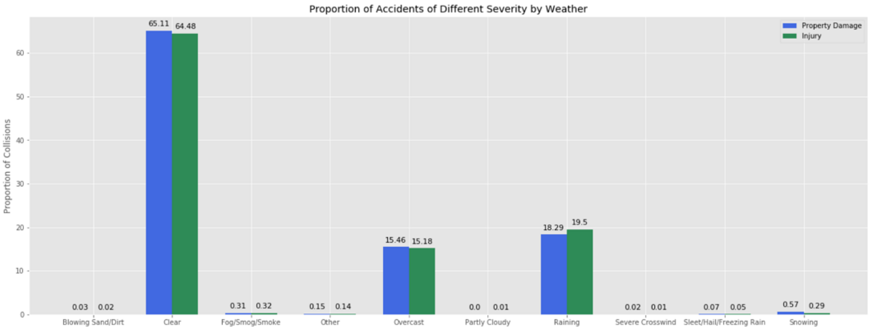


Road Conditions:



Light Conditions:

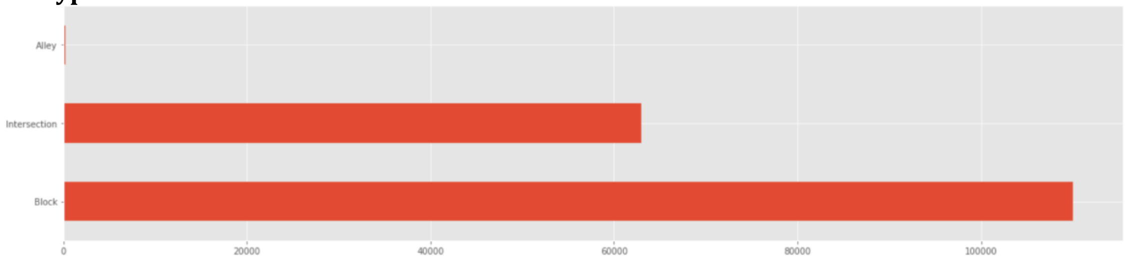
Weather Conditions:



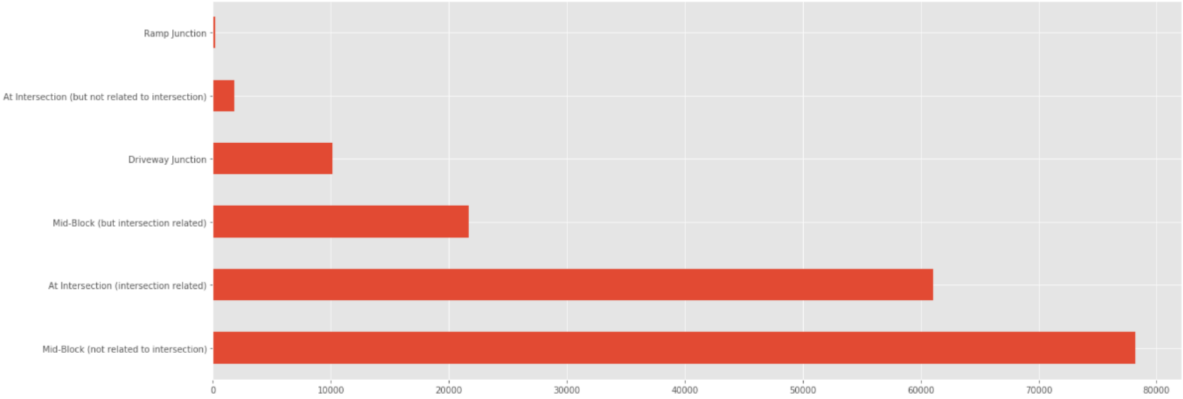
The data has a large number of values in 'normal' conditions, meaning clear weather, dry road and daylight. This results in an extremely low percentage of collisions happening in situations with bad weather, road conditions or lighting conditions. However, the graphs do show us that Injury causing accidents are more likely than Property Damaage when it is raining, or during dawn and dusk, or when the road is wet. We also find out that Property Damage is more likely when the sky is overcast or when it is dark, or when the road is slippery due to ice, snow or standing water.

Collision Count by Address Type and Junction Type

Address Type:

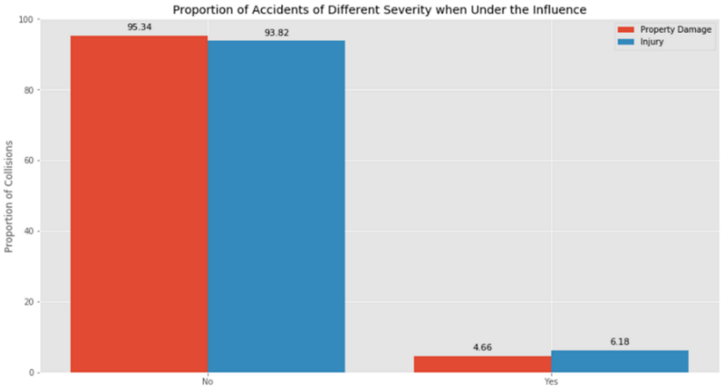


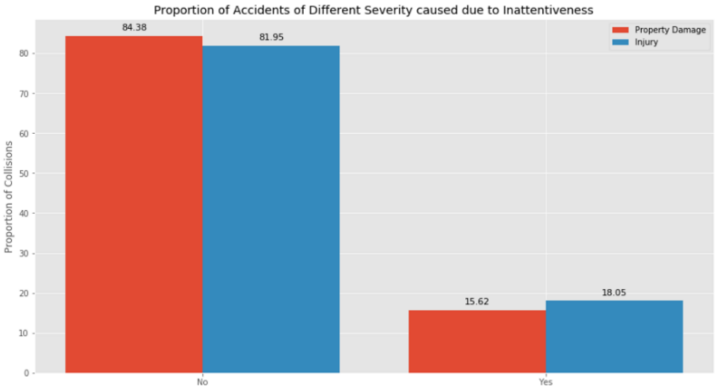
Junction Type:



When organizing collision count by address type and junction type, the data shows that most collisons happpen at a Block, and more specifically, at Mid-Block, but not related to intersection. The second most number of collisions happen at intersections. This leads to the conclusion that more effort should be put in maintaining safety in blocks as well and not only intersections.

Proportion of Severity of Collision when Under the Influence or caused due to Inattentiveness

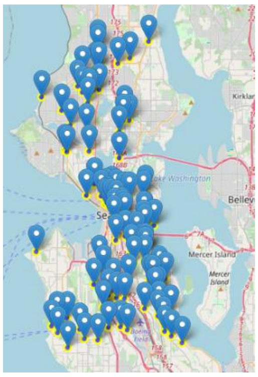
Under the Influence: 

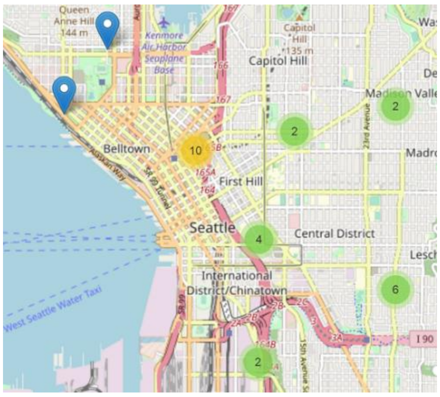
Inattentiveness: 

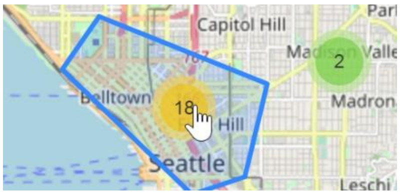
In both cases, we find that a greater percentage of collisions that cause injuries are present when either under the influence or inattentive. However, property damage is more likely when not under the influence and not inattentive. This shows that injuries are more likely to happen in an accident when either under the influence or inattentive.

Mapping Recent Accidents

A map of the 100 most recent accidents, organized by clusters shows that some locations and areas are more susceptible to accidents than other, possibly leading to the conclusion that more effort should be diverted in preventing accidents in these regions.

Pointers: 

Clusters: 



Reason for using proportion

The original dataframe is being used for data visualization due to the presence of text labels and exact data instead of oversampled or undersampled data. However, the original data consists of a large number of collisions of severity code 1, that is: only causing property damage. This would result in a skewed visualization with very few cases of injury every time. Thus for the purposes of visualization when considering both property damage and injury, we consider the count as a proportion of the total number of collisions of that particular severity code.
Proportion =
((number of considered collisions) / (number of collisions of the same severity code)) *100

3.5 Machine Learning

Dropping Unnecessary Data

While ADDRTYPE and JUNCTIONTYPE are useful for the purposes of visualization, they do not have a high impact on the severity of an accident. Thus both the columns are dropped. Our final dataset takes 10 categories into consideration when predicting severity of an accident.

Fields Used: COLLISIONTYPE, PERSONCOUNT, VEHCOUNT, INATTENTIONIND, UNDERINFL, WEATHER, ROADCOND, LIGHTCOND, SPEEDING, HITPARKEDCAR

Preprocessing and Splitting the Data

The dfoversampled dataframe along with the undersampled numpy arrays are used for machine learning as they contains balanced data with equal data points of both Severity Codes. We also check with the original dataset to create a biased model. First we process and normalize the data to avoid any errors with the model using the preprocessing method from the sklearn library. Then we split the data into training or testing sets so that we can accurately gauge the accuracy of the different kinds of machine learning models. Here we use 20% of the data for testing and the remaining 80% for training.

Machine Learning Classification Models

We use three different models: K-Nearest Neighbors, Decision Tree, and Logistic Regression. We train each model with our training data for each case: Original Data, Oversampled Data, Undersampled Data.

Model Evaluation

After training the three models, the three models are tested using the Accuracy, the Jaccard Score and the F1 score and include Log Loss for the Logistic Regression model. This is done for every dataset.

4. Results

The resuls of model evaluation, only showing accuracy are:

1. **Biased Model**
    - K-Nearest Neighbors Accuracy: 0.72
    - Decision Tree Accuracy: 0.73
    - Logistic Regression Accuracy: 0.68

2. **Oversampled Model**
    - K-Nearest Neighbors Accuracy: 0.67
    - Decision Tree Accuracy: 0.68
    - Logistic Regression Accuracy: 0.59

3. **Undersampled Model**
    - K-Nearest Neighbors Accuracy: 0.70
    - Decision Tree Accuracy: 0.70
    - Logistic Regression Accuracy: 0.64

The following are the detailed results of model evaluation, including Best k, Jaccard Index, F1 Score and Log Loss

1. **Biased Model**
    A. K-Nearest Neighbors:
        - Value of best k = 8
        - Jaccard Score = 0.72
        - F1 Score = 0.69

    B. Decision Tree:
        - Jaccard Score = 0.73
        - F1 Score = 0.69

    C. Logistic Regression:
        - Jaccard Score = 0.68
        - F1 Score = 0.60
        - Log Loss = 0.61

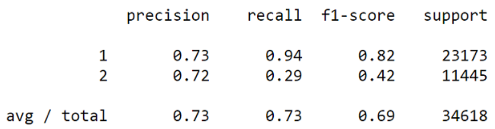
2. **Oversampled Model**
    A. K-Nearest Neighbors:
        - Value of best k = 8

    - Jaccard Score = 0.67
    - F1 Score = 0.67
  - B. Decision Tree:
    - Jaccard Score = 0.68
    - F1 Score = 0.68
  - C. Logistic Regression:
    - Jaccard Score = 0.59
    - F1 Score = 0.59
    - Log Loss = 0.66

3. **Undersampled Model**
   - A. K-Nearest Neighbors:
     - Value of best k = 9
     - Jaccard Score = 0.70
     - F1 Score = 0.70
   - B. Decision Tree:
     - Jaccard Score = 0.70
     - F1 Score = 0.70
   - C. Logistic Regression:
     - Jaccard Score = 0.64
     - F1 Score = 0.63
     - Log Loss = 0.64

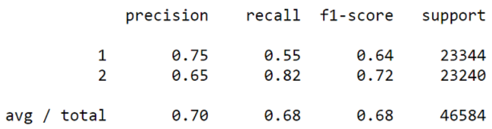
The overall best model for predicting the severity of an accident is a biased Decision Tree with a Jaccard Score of 0.73 and an F1 score of 0.69 .
The accuracy rate of this model is 0.73.

To confirm that the best model is consistent, we look at the classification reports of the Decision Tree models.



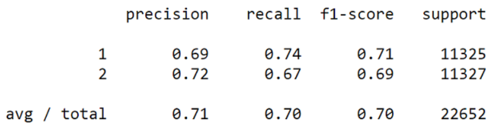
```
             precision    recall  f1-score   support

          1       0.73      0.94      0.82     23173
          2       0.72      0.29      0.42     11445

avg / total       0.73      0.73      0.69     34618
```

Classification Report: Biased Decision Tree -



```
             precision    recall  f1-score   support

          1       0.75      0.55      0.64     23344
          2       0.65      0.82      0.72     23240

avg / total       0.70      0.68      0.68     46584
```

Classification Report: Oversampled Decision Tree -



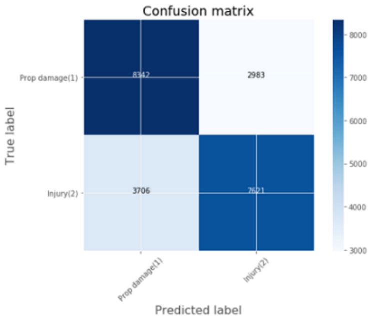
```
             precision    recall  f1-score   support

          1       0.69      0.74      0.71     11325
          2       0.72      0.67      0.69     11327

avg / total       0.71      0.70      0.70     22652
```

Classification Report: Undersampled Decision Tree -

When looking at a detailed classification report, we find that going with a biased decision tree is a bad idea. This is because the classification report shows an accuracy of 0.82 when predicting a SEVERITYCODE of 1 but an accuracy of only 0.42 when predicting 2. The model is extremely biased and cannot accurately predict SEVERITYCODE 2.
So we now compare the decision tree models of the oversampled and undersampled data. A quick

look shows us that the decision tree of the undersampled data gives us the most consistent and accurate model.

Thus the **recommended model is the Decision Tree, with data being undersampled**. This has a Jaccard Score, an F1 Score and an Accuracy of **0.70**.



Confusion Matrix: Undersampled Decision Tree -

5. Discussion

The problem with the dataset is that it is extremely unbalanced. While the method used, or any of the other two methods do somewhat make up for it, some error is still generated when working with unbalanced data. The best way to overcome this error is by getting actually balanced data.
The dataset provided also is not the complete version. It only consists of collisions of severity codes 1 and 2, whereas in actuality there are 3 codes. Using the complete dataset, hopefully a balanced one, would probably give a more accurate model.

We also observe that a decision tree model is faster to run and outperforms both K Nearest Neighbors and Logistic Regression in terms of accuracy, even if the dataset is oversampled or undersampled.

With the existing modified dataset, the following observations can be made:

1. More severe accidents are more likely to happen in cases where road conditions are slippery, during dawn or dusk, and when it is raining.
2. More collisions happen towards the middle of the month than at the end.
3. Most collsions happen mid-block, unrelated to an intersection. The second most number of accidents happen at intersections.
4. A greater proportion of collisions causing injury happen due to being under the influence or inattentive as compared to the ones causing property damage.
5. For a large dataset containing categorical data, a Decision Tree model may be the best classifier.
6. When the data is unbalanced, a biased model is generated, which cannot be put to use in practice. Depending on your dataset, undersampling or oversampling may counter this.
7. K Nearest Neighbors takes an extremely long time to train over a large number of datapoints, whereas Decision Trees and Logistic Regression are much faster.

6. Conclusion

In this project I analyzed the traffic collisions in Seattle from 2004 to 2020 of varying degrees of severity. I looked at multiple factors including Road Conditions, Weather, Speeding, Inattentiveness, Location Type and Number of Vehicles/People Involved. Machine learning models are a good fit for this kind of data as they enable you to predict a particular target when given the values of the other fields. The target of prediction in this case was the severity of an accident using the other data. Using the required features, I built a multiple linear regression model, which gave poor results as expected.

This was because the data was categorical and not continuous. After that, I created multiple classification models and compared their results on the regular data, oversampled data, and undersampled data for the best accuracy, which led to the finding that a Decision Tree model when using undersampled data worked best. The reason for undersampled data having higher accuracy than oversampled data could be due to the large number of data points that had to be generated to make the data oversampled (around 50,000), which left room for error. This model can be useful to regular travellers, traffic management and police, and insurance companies. For example, it could help the traffic police analyze when and in what conditions collisions of higher severity are likely to occur, and take steps to prevent it beforehand.