
IBM Data Science
Predicting Severity of Car Accidents
PRIYESH SAINI

Predicting Severity of accidents is crucial for safety of citizens

- Seattle, a seaport city on the west coast of the US, is home to around 8 lac people.
- Due to population density and other factors, city is witnessing number of accidents every day.
- According to 2019 Annual Traffic Collison report from WSDOT, there were a total 10,315 cases in Seattle alone.
- Out of which, 22 - fatal, 190 - serious injury collisions & 834 - minor injury collisions.
- Clearly city needs some strict measures to counteract the current situation.
- This project is an attempt to do the same.

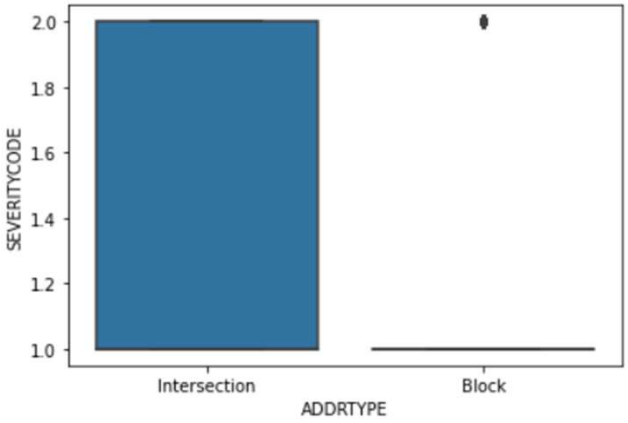
Data Source & Data Pre-processing

- Dataset is provided by SPD & SDOT (under IBM Data Science Specialization from Coursera).
- This dataset contains a total of 1,94,673 records & 37 attributes.
- Following things are dropped from the dataset:
  - ❖ Attributes having missing values more than 40 %.
  - ❖ Attributes which don't have any impact on prediction of target variable.
  - ❖ Latitude and Longitude attribute values, as they need special libraries for imputation.
  - ❖ Attributes which represented same information & were redundant.

Cleaned Data contains 10 attributes.

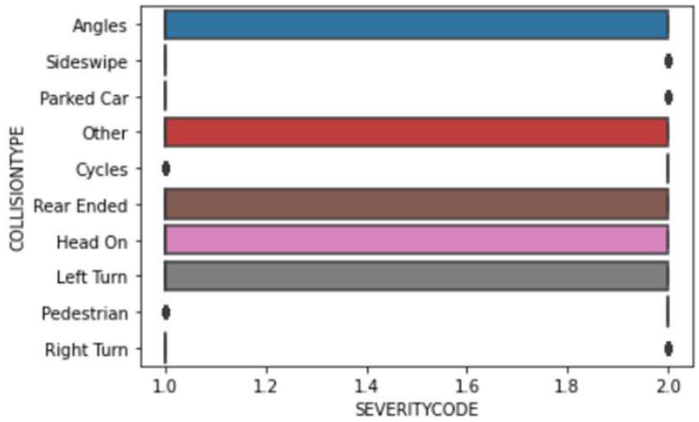
Exploratory Data Analysis

- In this section we will analyse each input attributes and its effect on target variable, visually.
- ADDRTYPE & SEVERITYCODE
  - ❖ Most crashes happened in Block area (65.2%).
  - ❖ Most of the accidents that happened in the Intersection area belonged to severity level 2.
  - ❖ There is no overlap between two variables. Hence it is a good predictor variable for our model.



Exploratory Data Analysis Contd…

- COLLISIONTYPE & SEVERITYCODE
  - ❖ Crashes were more severe where type of collision was at Angles, Rear Ended, Head On, Left Turn and Others.
  - ❖ There were rare cases of severity 2 accidents when the collisions were of type Sideswipe, Parked Car, & Right Turn.
  - ❖ There were collisions involving Cycle and Pedestrian that was of severit level 1.



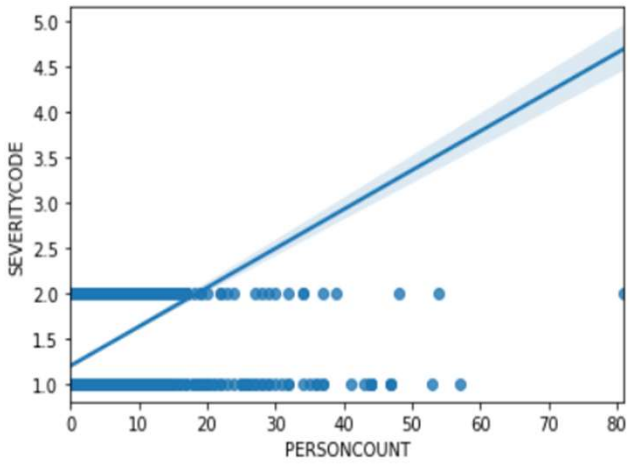
Exploratory Data Analysis Contd…

- PERSONCOUNT & SEVERITYCODE

❖ There is a direct linear relationship between the number of casualties and severity of crashes.

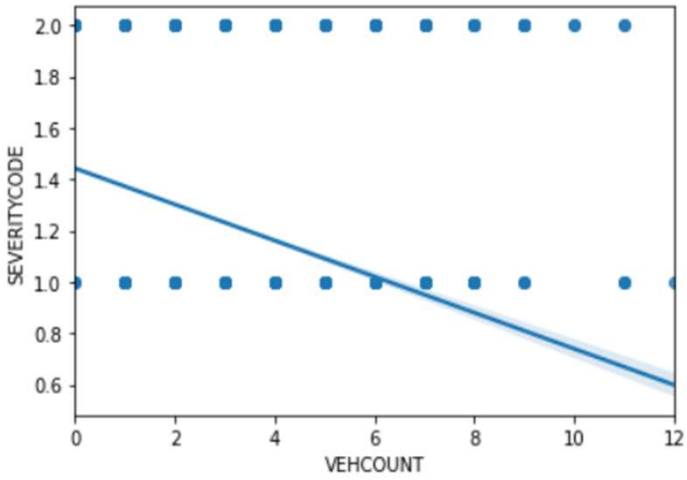
❖ Since the regression line have positive slope there exists a positive correlation between the two variables.

❖ Although, there is not much difference between the number of casualties in accidents of severity level 1 and 2.



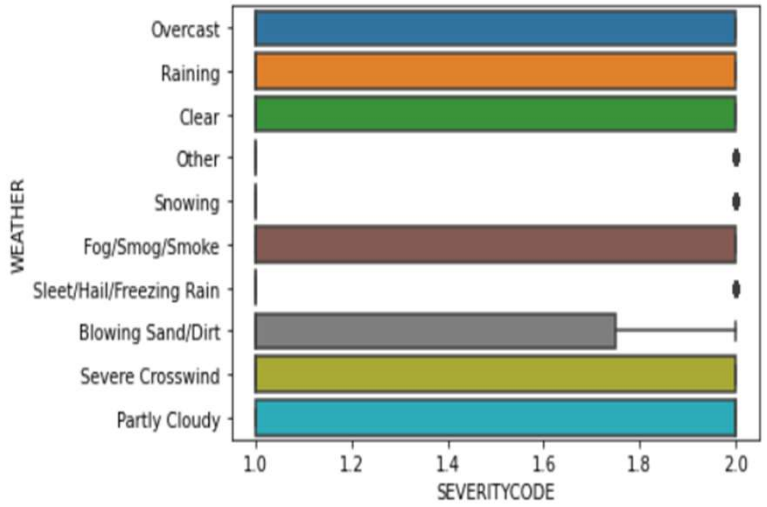
Exploratory Data Analysis Contd…

- VEHCOUNT & SEVERITYCODE
  - ❖ There is a negative correlation between two variables since slope of regression line is negative.
  - ❖ Vehicle count is almost similar in both cases of severity.
  - ❖ Interestingly the rate of increase in vehicle count is more in accidents of severity 1.



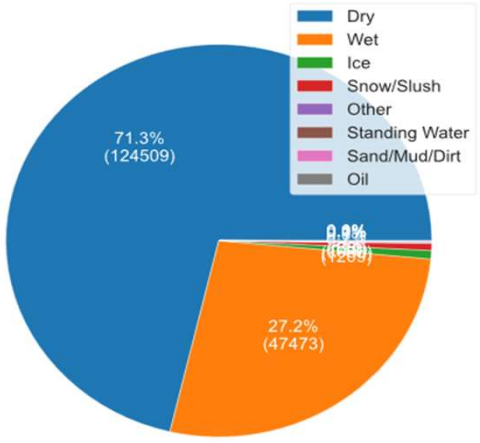
Exploratory Data Analysis Contd…

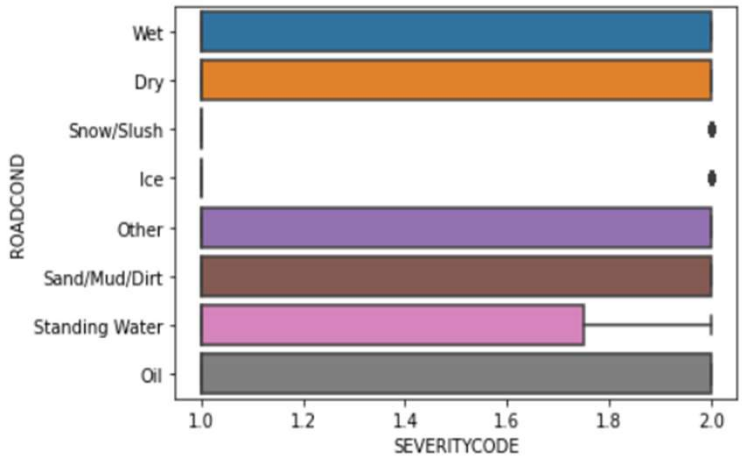
- WEATHER & SEVERITYCODE
  - ❖ On weather conditions like Clear, Rainy, Overcast and Severe Crosswind, most of the crashes belongs to severity level 2.
  - ❖ There are some extreme outliers for level 2 crashes for weather conditions like Snowing, Sleet/Hail/Freezing Rain and others.



Exploratory Data Analysis Contd…

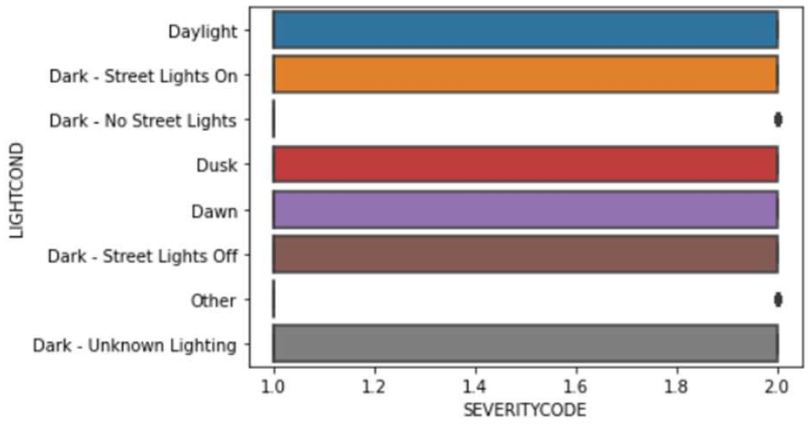
- ROADCOND & SEVERITYCODE
  - ❖ As expected, accidents were more severe when the roads were wet, oily, covered with sand/mud/dirt and other situations.
  - ❖ Surprisingly, severity was high even for Dry conditions.



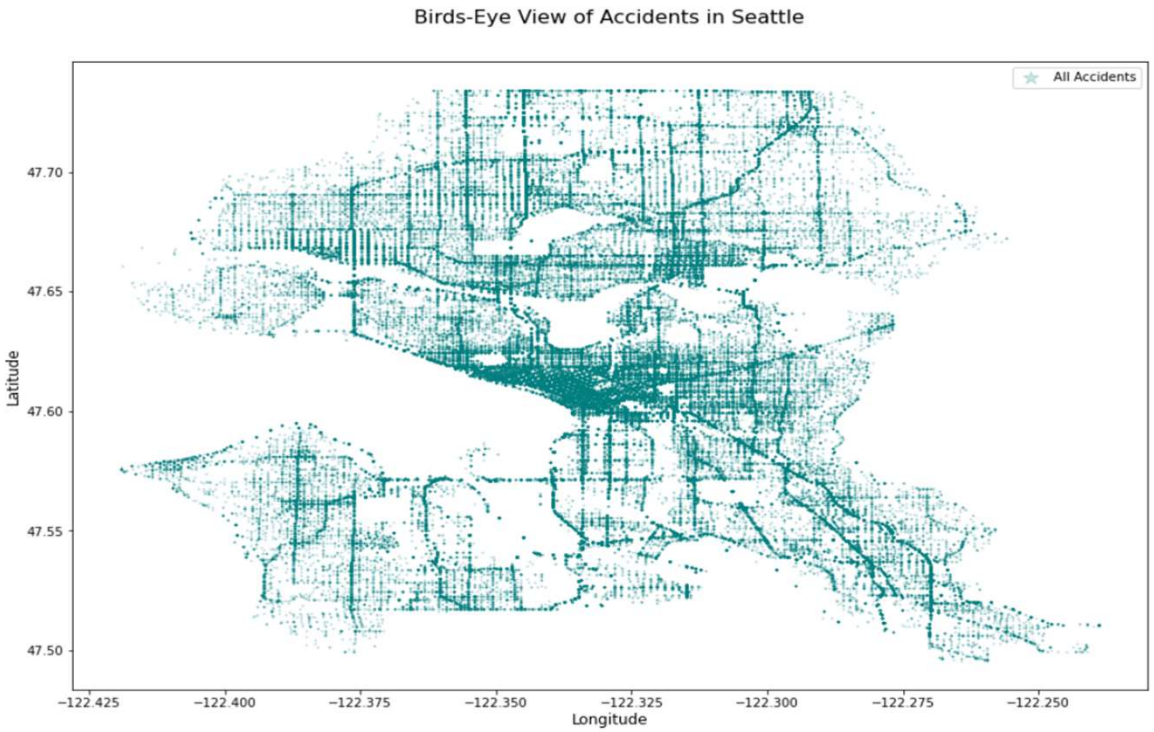


Exploratory Data Analysis Contd…

- LIGHTCOND & SEVERITYCODE
  - ❖ In the absence of proper lit environment, severity of accidents is high. For e.g. at conditions like Dusk, Dawn, Dark – Street lights off, Dark – Unknown Lightning the number of crashes are less but the probability of each crash being of high severity is quite high in these situations.
  - ❖ Surprisingly, severe crashes also happened during Daylight furthermore, the number of accidents was also highest during daylight.
  - ❖ There are some extreme outliers in situation like Dark – No street lights and others.



Exploratory Data Analysis Contd…



Exploratory Data Analysis Contd…

```
# X is input features

X=df[['LATITUDE','LONGITUDE','ADDRTYPE','COLLISIONTYPE','PERSONCOUNT','VEHCOUNT','WEATHER','ROADCOND','LIGHTCOND','HITPA
X
```

	LATITUDE	LONGITUDE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	VEHCOUNT	WEATHER	ROADCOND	LIGHTCOND	HITPARKEDCAR
0	47.703140	-122.323148	1	0	2	2	4	7	5	0
1	47.647172	-122.347294	0	9	2	2	6	7	2	0
2	47.607871	-122.334540	0	5	4	3	4	0	5	0
3	47.604803	-122.334803	0	4	3	3	1	0	5	0
4	47.545739	-122.306426	1	0	2	2	6	7	5	0
...	...	...	...	...	...	...	...	...	...	...
184577	47.565408	-122.290826	0	2	3	2	1	0	5	0
184578	47.690924	-122.344526	0	7	2	2	6	7	5	0
184579	47.683047	-122.306689	1	3	3	2	1	0	5	0
184580	47.678734	-122.355317	1	1	2	1	1	0	6	0
184581	47.611017	-122.289360	0	7	2	2	1	7	5	0

184582 rows × 10 columns

```
y=df['SEVERITYCODE']
```

Exploratory Data Analysis Contd…

- Perfroming Normalization on input feature dataset and splitting it into Training & Testing dataset.

```
X= preprocessing.StandardScaler().fit(X).transform(X)
X[0:5]
```

```
array([[ 1.48739777,  0.24501881,  1.38540755, -1.60088434, -0.34423279,
         0.04747219,  0.80212158,  1.71821177,  0.57629845, -0.19415094],
       [ 0.49174574, -0.55927011, -0.72180926,  1.6151249 , -0.34423279,
         0.04747219,  1.78209098,  1.71821177, -1.54999725, -0.19415094],
       [-0.20740912, -0.13443019, -0.72180926,  0.18578746,  1.12835405,
         1.84491063,  0.80212158, -0.58893996,  0.57629845, -0.19415094],
       [-0.26199758, -0.1432006 , -0.72180926, -0.1715469 ,  0.39206063,
         1.84491063, -0.66783251, -0.58893996,  0.57629845, -0.19415094],
       [-1.31271947,  0.80202434,  1.38540755, -1.60088434, -0.34423279,
         0.04747219,  1.78209098,  1.71821177,  0.57629845, -0.19415094]])
```

```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2, random_state=1)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (147665, 10) (147665,)
Test set: (36917, 10) (36917,)
```

Model Development & Evaluation

At a fundamental level, there are two types of Supervised Machine Learning Algorithm.

- Regression Model -Used to predict the continuous values.
- Classification Model - Used to predict the categorical values.

Reason: Since our target variable (SEVERITYCODE) is a categorical variable i.e. it has two values '1' and '2', our problem falls to the category of classification model. We are going to use two machine learning algorithms to predict the value of target variable.

- ❖ Logistic Regression Model
- ❖ Decision Tree Model

Finally, we will find out-of-sample accuracy for both models using test data and compare the accuracy of both in the result section.

Model Development & Evaluation Contd…

- Logistic Regression Model

Creating a Logistic Regression Object:

```
# Fitting the model using Logistic Regression object, LR
LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train)
LR
```

```
LogisticRegression(C=0.01, solver='liblinear')
```

Predicting the target variable using test data and its probability:

```
# Predicting the target variable using Test Data
yhat = LR.predict(X_test)
yhat
```

```
array([2, 1, 1, ..., 1, 1, 1], dtype=int64)
```

```
yhat_prob = LR.predict_proba(X_test)
yhat_prob
```

```
array([[0.47016722, 0.52983278],
       [0.78106286, 0.21893714],
       [0.74269633, 0.25730367],
       ...,
       [0.92957679, 0.07042321],
       [0.59542303, 0.40457697],
       [0.75388773, 0.24611227]])
```

Model Development & Evaluation Contd…

- Decision Tree Model

Creating a Decision Tree Classifier Object and predicting the target variable:

```
# Creating a Decision Tree Classifier Instance
severity = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
severity.fit(X_train,y_train)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=4)
```

```
# Predicting the target variable using Test Data
pred_severity = severity.predict(X_test)
pred_severity
```

```
array([2, 1, 1, ..., 1, 1, 1], dtype=int64)
```

Results

Logistic Regression Model

```
print('Jaccard Smimilarity Score is: ',jaccard_score(y_test, yhat))
```

```
Jaccard Smimilarity Score is:  0.7008705052344105
```

```
# Creating a Confusion Matrix
cnf_matrix = confusion_matrix(y_test, yhat, labels=[1,0])
np.set_printoptions(precision=2)
```

```
print(classification_report(y_test, yhat))
```

```
              precision    recall  f1-score   support

           1       0.73      0.95      0.82     25828
           2       0.60      0.16      0.25     11089

    accuracy                           0.72     36917
   macro avg       0.66      0.56      0.54     36917
weighted avg       0.69      0.72      0.65     36917
```

```
# Calculating Log Loss
print('Log Loss is: ',log_loss(y_test, yhat_prob))
```

```
Log Loss is:  0.570318380953288
```

Results

Decision Tree Model

```
# Evaluating the model using F1 Score and Jaccard Similarity Score
d=f1_score(y_test,pred_severity, average='weighted')
print("F1 Score is: ",d)
print("DecisionTree's Jaccard Smilarity Score Score is: ", jaccard_score(y_test, pred_severity))
```

```
F1 Score is:  0.7201209036124941
DecisionTree's Jaccard Smilarity Score Score is:  0.7214827752348831
```

Discussion

- Now, we are ready for prediction of severity of the accidents.
- This model can also help us in finding the key factors and situations that led to accidents.
- We can find out answer to following crucial questions:
  - ❖ Which areas in Seattle are more prone to severe collisions?
  - ❖ What road and light conditions are responsible for accidents?
  - ❖ Up to what extent weather can have impact on severity of crashes?

Conclusion

- In this project we tried to solve the given problem using two classification models and is now ready to be deployed and predict the severity of accidents provided the sufficient input.
- I hope that the SDOT will get sufficient help from this project to overcome the problem of fatal crashes.
- This project was a step in the direction of keeping the citizens safe from accidents so that they can return to their homes safe and sound where their family is waiting for them.