I. Introduction

The US Department of Transportation has reported a total of 6,734,000 vehicle crashes occurring across the country, in 2018, resulting in approximately 1.9 million injuries and 34,000 fatalities. These accidents, and subsequent injuries, can be attributed to several factors, including speeding, distracted driving, number of vehicles and number of people involved in the accident.

By utilizing historical crash data, analyzing each reported incident, and the attributes associated with each crash, a model can be created that can help warn drivers of potential risks, and may help local governments assign additional resources to prevent future accidents.

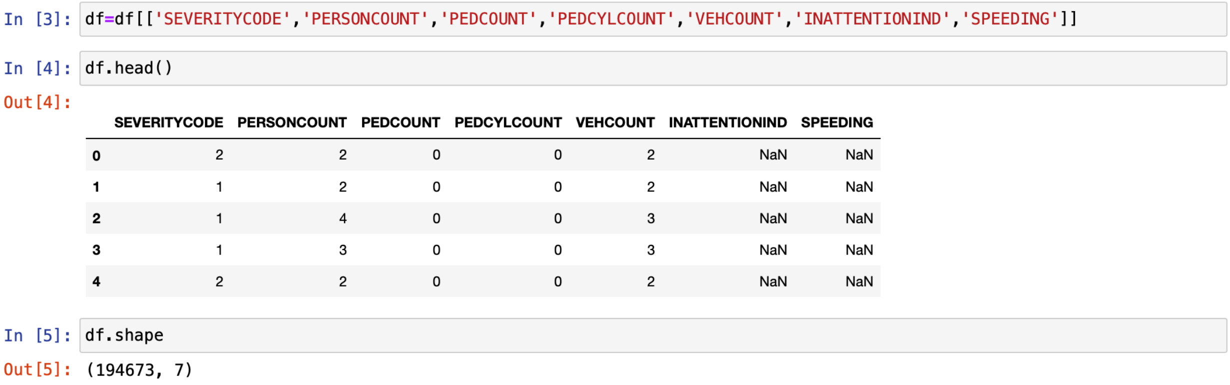
II. Data

The data used for this project include crash reports, from the city of Seattle, Washington, for the years between 2014-2020, to include a total of 194,673 accidents. While there are a number of details associated with each incident, this project will focus only on those factors which are believed to have the most impact on accident severity.

III. Methodology

The dataset contains 194,673 recorded accidents, with 136,485 accidents identified as Severity Code 1, and 58,188 accidents identified as Severity Code 2. Given the difference in number of cases in each category, it is assumed that any number of other factors, also reported in this data set, can be used to determine future accidents' level of severity.

The first step in analyzing this data set is to select the assumed factors that drive severity level. For this project, this researcher assumed the following factors played a major role: person count, pedestrian count, pedestrian/cycle Count, vehicle count, inattention indicators, and speed. It was necessary to pare down the original data set and remove the unstated features, and the result is the following table:

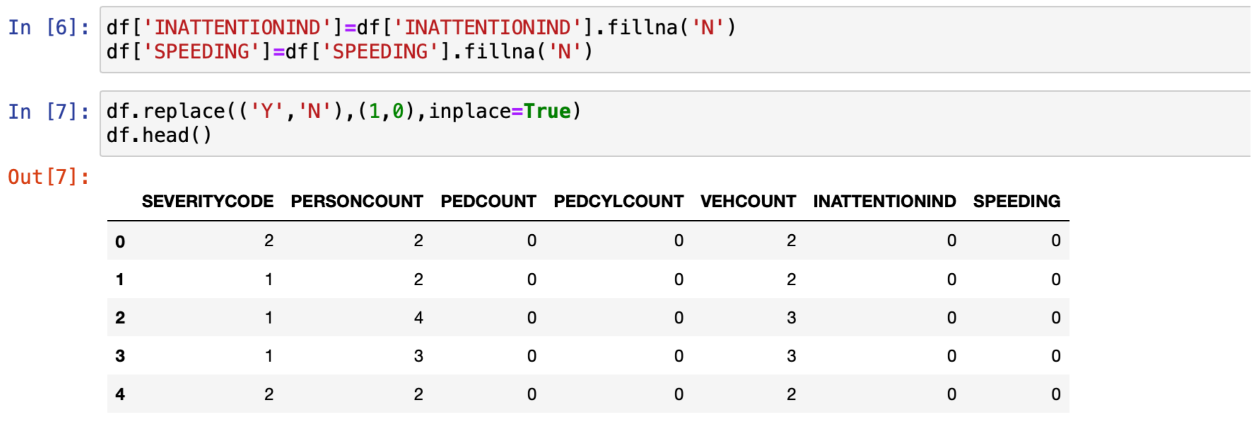


```
In [3]: df=df[['SEVERITYCODE','PERSONCOUNT','PEDCOUNT','PEDCYLCOUNT','VEHCOUNT','INATTENTIONIND','SPEEDING']]

In [4]: df.head()

Out[4]:
```

	SEVERITYCODE	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEHCOUNT	INATTENTIONIND	SPEEDING
0	2	2	0	0	2	NaN	NaN
1	1	2	0	0	2	NaN	NaN
2	1	4	0	0	3	NaN	NaN
3	1	3	0	0	3	NaN	NaN
4	2	2	0	0	2	NaN	NaN

```
In [5]: df.shape

Out[5]: (194673, 7)
```

Data cleaning was necessary, however, as the values for columns Inattention Indicators, and Speeding, were missing for many records (NaN), or were displayed as one of many values ("Y","N","1","0"). Cleaning the values resulted in the following table:



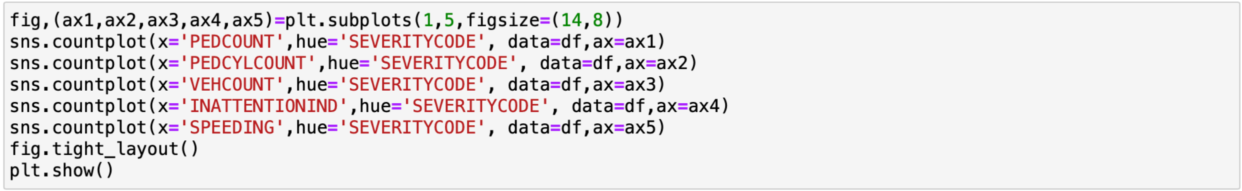
In [6]:
```
df['INATTENTIONIND']=df['INATTENTIONIND'].fillna('N')
df['SPEEDING']=df['SPEEDING'].fillna('N')
```

In [7]:
```
df.replace(('Y','N'),(1,0),inplace=True)
df.head()
```

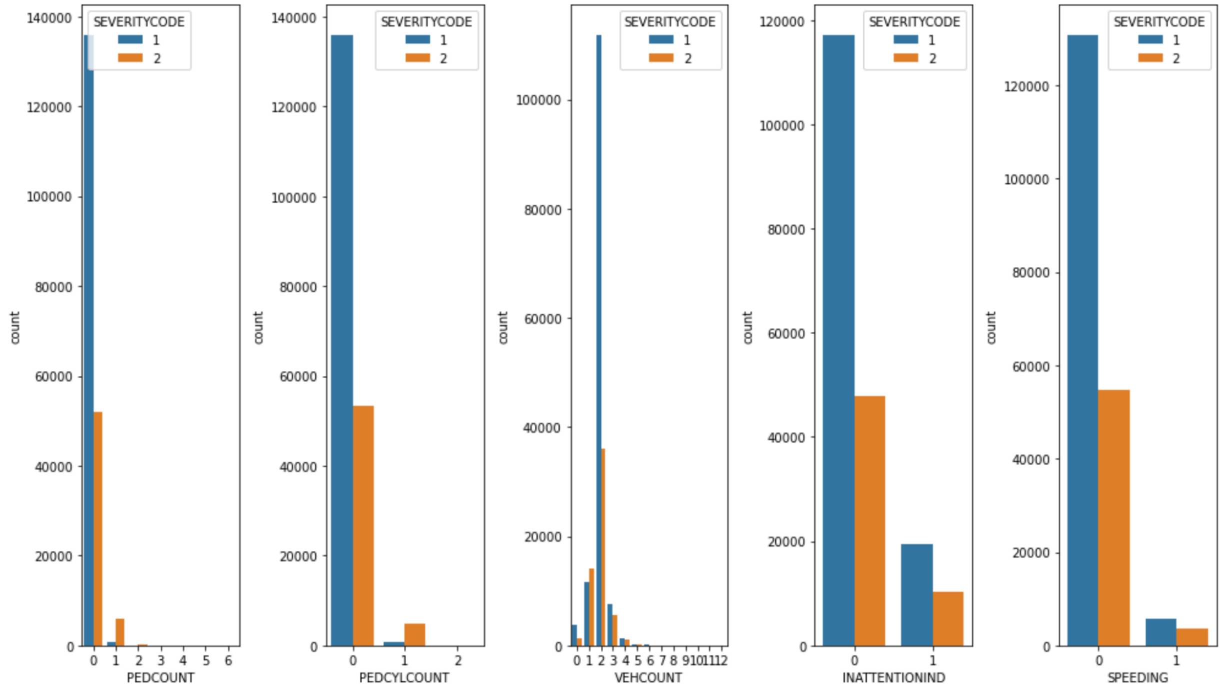
Out[7]:

	SEVERITYCODE	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEHCOUNT	INATTENTIONIND	SPEEDING
0	2	2	0	0	2	0	0
1	1	2	0	0	2	0	0
2	1	4	0	0	3	0	0
3	1	3	0	0	3	0	0
4	2	2	0	0	2	0	0

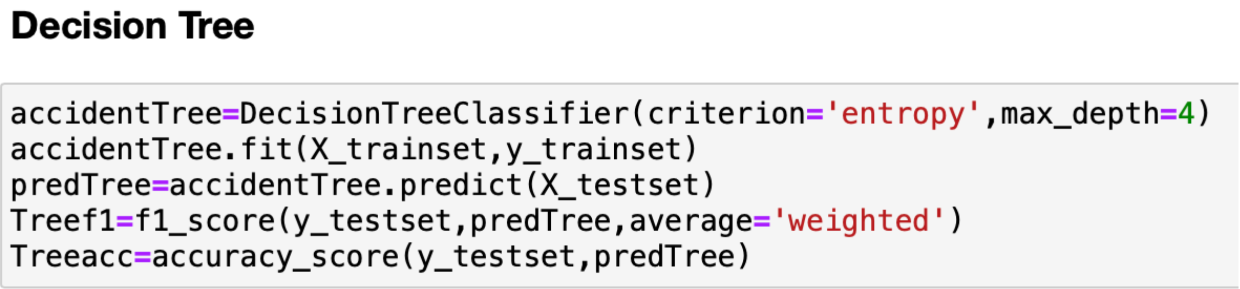
Once the data was clean, analysis could begin.



```
fig,(ax1,ax2,ax3,ax4,ax5)=plt.subplots(1,5,figsize=(14,8))
sns.countplot(x='PEDCOUNT',hue='SEVERITYCODE', data=df,ax=ax1)
sns.countplot(x='PEDCYLCOUNT',hue='SEVERITYCODE', data=df,ax=ax2)
sns.countplot(x='VEHCOUNT',hue='SEVERITYCODE', data=df,ax=ax3)
sns.countplot(x='INATTENTIONIND',hue='SEVERITYCODE', data=df,ax=ax4)
sns.countplot(x='SPEEDING',hue='SEVERITYCODE', data=df,ax=ax5)
fig.tight_layout()
plt.show()
```

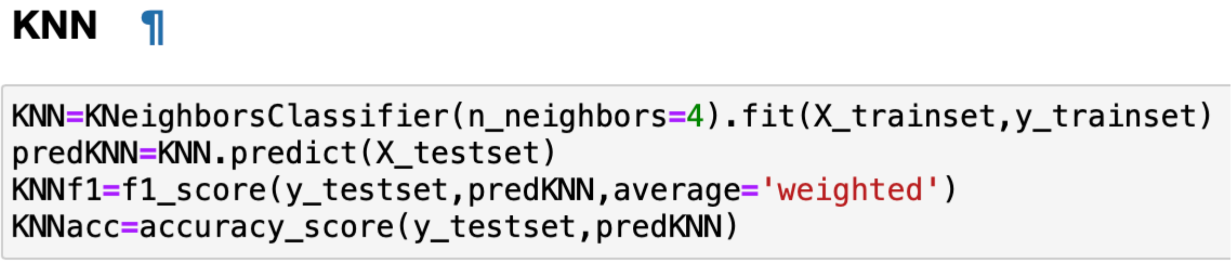


For each contributing factor, the counts of Severity Codes were calculated during exploratory analysis to identify any aggravating factors. Furthermore, machine learning models were created using K-Nearest Neighbors, Decision Tree, and Logistic Regression, once the data was split into training and testing datasets.



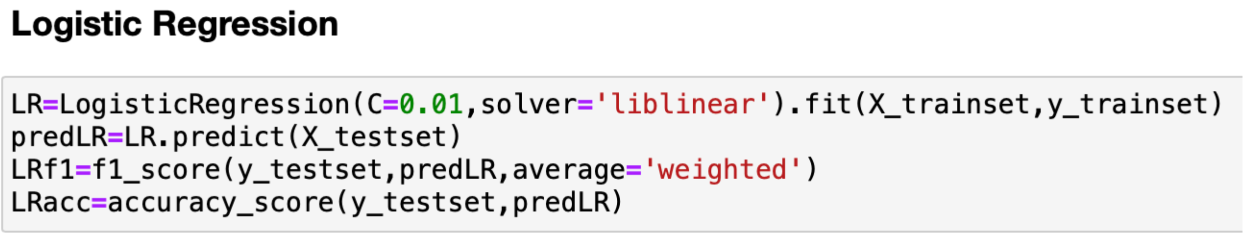
Decision Tree

```
accidentTree=DecisionTreeClassifier(criterion='entropy',max_depth=4)
accidentTree.fit(X_trainset,y_trainset)
predTree=accidentTree.predict(X_testset)
Treef1=f1_score(y_testset,predTree,average='weighted')
Treeacc=accuracy_score(y_testset,predTree)
```



KNN ¶

```
KNN=KNeighborsClassifier(n_neighbors=4).fit(X_trainset,y_trainset)
predKNN=KNN.predict(X_testset)
KNNf1=f1_score(y_testset,predKNN,average='weighted')
KNNacc=accuracy_score(y_testset,predKNN)
```



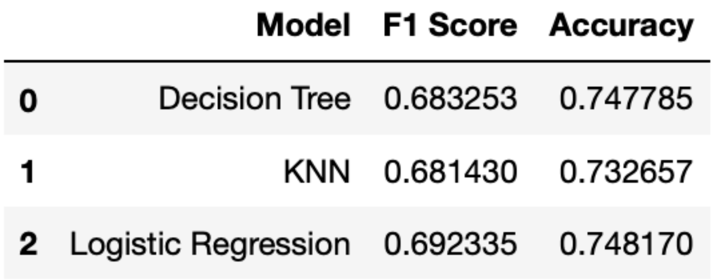
Logistic Regression

```
LR=LogisticRegression(C=0.01,solver='liblinear').fit(X_trainset,y_trainset)
predLR=LR.predict(X_testset)
LRf1=f1_score(y_testset,predLR,average='weighted')
LRacc=accuracy_score(y_testset,predLR)
```

IV. Results

Results did not confirm the hypothesis of this researcher. As can be seen in the above graphs, there was no significant difference in frequency of increased severity among the assumed factors. There were slight differences in factors involving pedestrians (on foot AND on bicycle), however, these results were not significant.

Calculating accuracy and F1 scores resulted in the following:



	Model	F1 Score	Accuracy
0	Decision Tree	0.683253	0.747785
1	KNN	0.681430	0.732657
2	Logistic Regression	0.692335	0.748170

V. Discussion

The Logistic Regression model was most accurate with an accuracy of 74.81%, and F1 score of .692, probably due to the weak correlation between features of dataset and accident severity. Additional models may not provide any significant improvements.

VI. Conclusion

This report address the problem of predicting vehicle accident severity in an effort to provide drivers with advance knowledge of possible accidents during their commute. Unfortunately, correlation could not be identified.