Analysis of Seattle Collision Data

a presentation by Miranda Childs for Coursera's IBM Data Science Professional Certificate capstone project

Collision reduction starts with awareness

- In Seattle, WA, from 2004 to present there have been 194,673 collisions reported by the Seattle Police Department (SPD) to the Seattle Department of Transportation (SDOT). 58,188 of those collisions involved an injury.

- By determining and focusing on the factors that contribute most heavily to **severe collisions**, we can create meaningful strategies to reduce the number of accidents, especially those with injuries, in order to increase the wellbeing and longevity of our community.

- Vision Zero Network is "a collaborative campaign helping communities reach their goals of Vision Zero -- eliminating all traffic fatalities and severe injuries -- while increasing safe, healthy, equitable mobility for all." (Vision Zero Network). Through our thorough analysis we will make recommendations for the next campaigns and strategies that Vision Zero can execute in collaboration with SDOT.

Data Acquisition

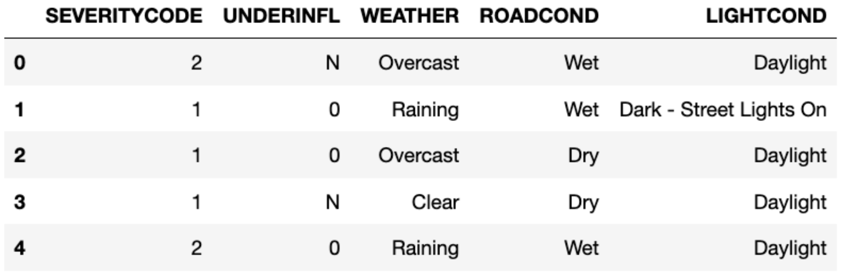
- Collision data from 2004 to present, provided by the Traffic Records Group in conjunction with the Seattle Police Department and Seattle Department of Transportation. (collision data set, and metadata)

- There are 194,673 rows and 37 features in the raw dataset

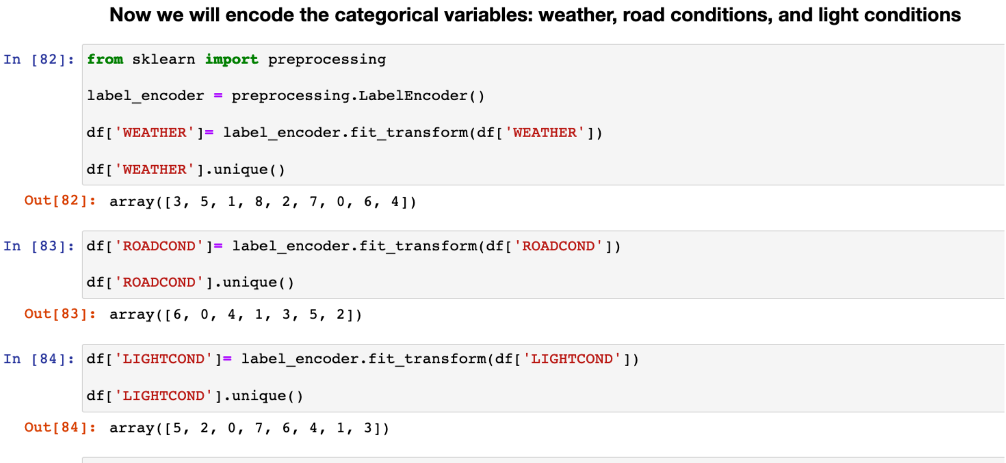
Data Cleaning

- Duplicate columns and null values were dropped
- Features were transformed for consistency and accuracy
- 4 features were chosen to predict the severity of collisions: Under The Influence, Weather, Road Condition, and Light Condition
- The cleaned data contains 172,262 rows and 4 features

Successful transformation of the categorical variables



	SEVERITYCODE	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND
0	2	N	Overcast	Wet	Daylight
1	1	0	Raining	Wet	Dark - Street Lights On
2	1	0	Overcast	Dry	Daylight
3	1	N	Clear	Dry	Daylight
4	2	0	Raining	Wet	Daylight



Now we will encode the categorical variables: weather, road conditions, and light conditions

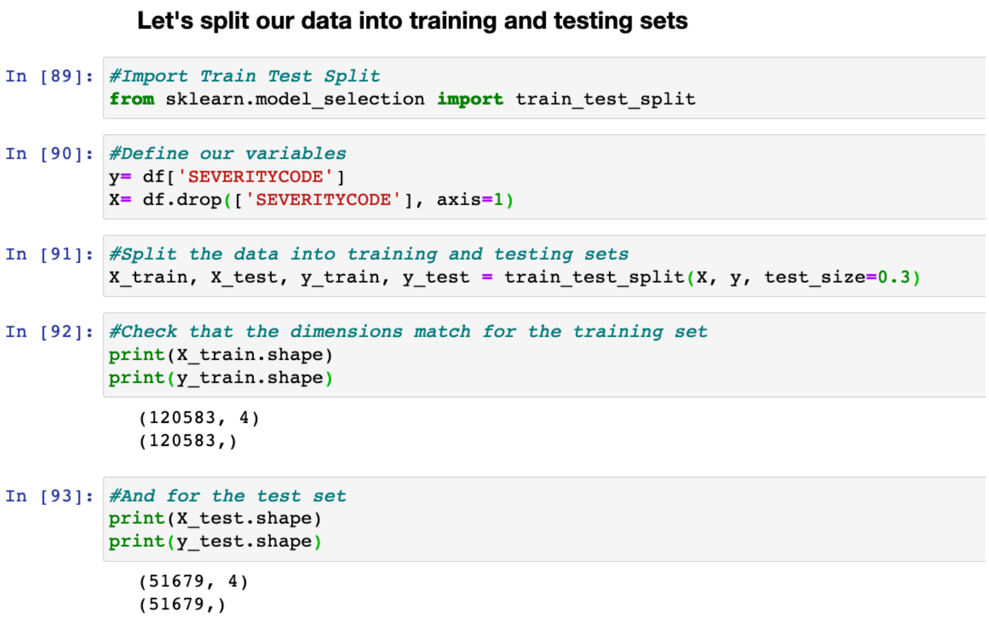
```
In [82]: from sklearn import preprocessing

         label_encoder = preprocessing.LabelEncoder()

         df['WEATHER']= label_encoder.fit_transform(df['WEATHER'])

         df['WEATHER'].unique()

Out[82]: array([3, 5, 1, 8, 2, 7, 0, 6, 4])

In [83]: df['ROADCOND']= label_encoder.fit_transform(df['ROADCOND'])

         df['ROADCOND'].unique()

Out[83]: array([6, 0, 4, 1, 3, 5, 2])

In [84]: df['LIGHTCOND']= label_encoder.fit_transform(df['LIGHTCOND'])

         df['LIGHTCOND'].unique()

Out[84]: array([5, 2, 0, 7, 6, 4, 1, 3])
```

Methodology

- Classification techniques are ideal for predicting the category of the collision: either severe (resulting in injury), or less severe (property damage only)

- A decision tree is a good algorithm for this dataset as it is a classification algorithm, and because it can work well with imbalanced data.

First the data was split into separate training and testing sets



Let's split our data into training and testing sets

```
In [89]: #Import Train Test Split
         from sklearn.model_selection import train_test_split
```

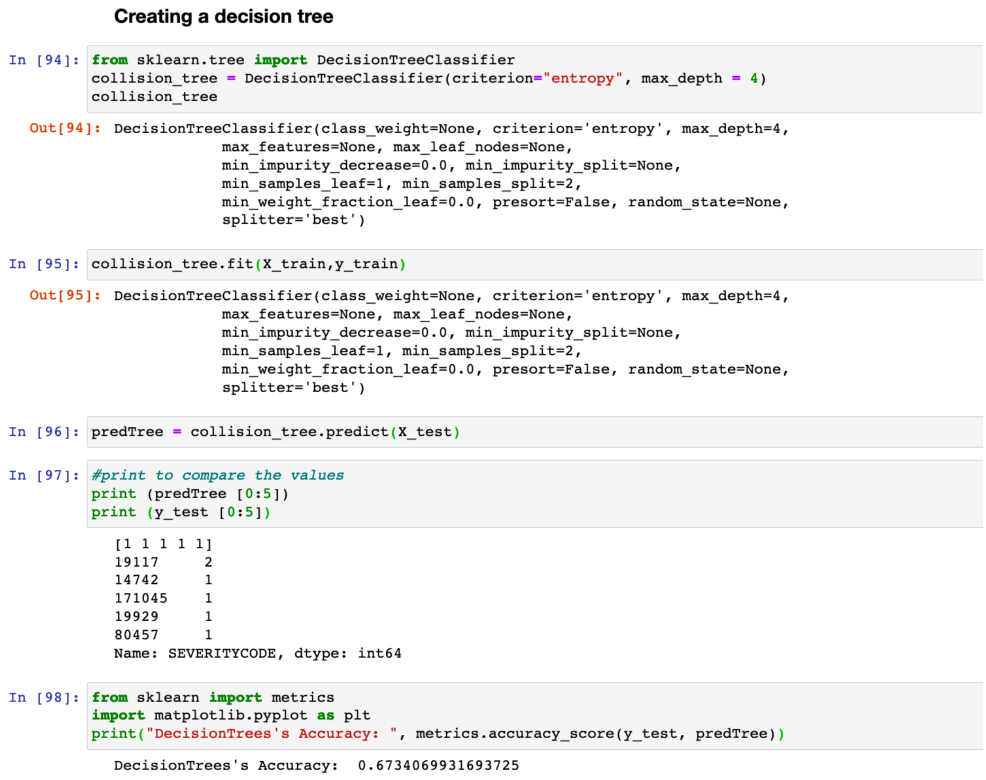
```
In [90]: #Define our variables
         y= df['SEVERITYCODE']
         X= df.drop(['SEVERITYCODE'], axis=1)
```

```
In [91]: #Split the data into training and testing sets
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
In [92]: #Check that the dimensions match for the training set
         print(X_train.shape)
         print(y_train.shape)

            (120583, 4)
            (120583,)
```

```
In [93]: #And for the test set
         print(X_test.shape)
         print(y_test.shape)

            (51679, 4)
            (51679,)
```

Then a decision tree was created

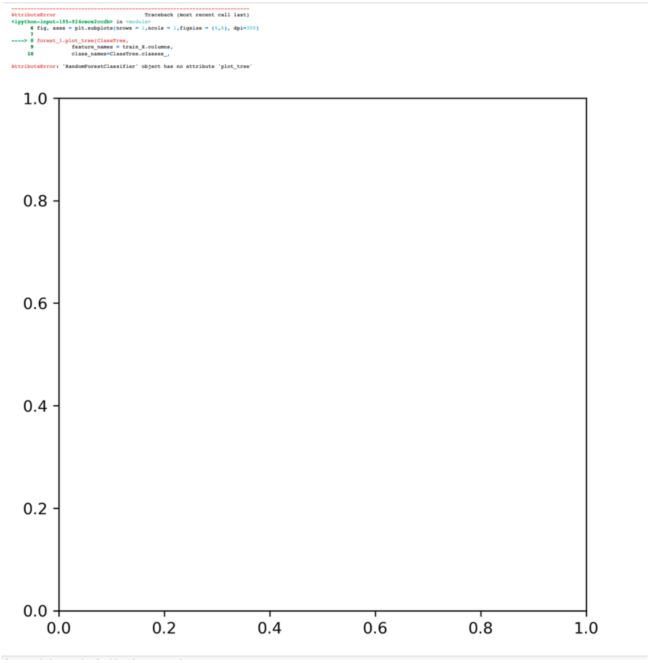


Creating a decision tree

```
In [94]: from sklearn.tree import DecisionTreeClassifier
         collision_tree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
         collision_tree

Out[94]: 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')

In [95]: collision_tree.fit(X_train,y_train)

Out[95]: 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')

In [96]: predTree = collision_tree.predict(X_test)

In [97]: #print to compare the values
         print (predTree [0:5])
         print (y_test [0:5])

          [1 1 1 1 1]
          19117     2
          14742     1
          171045    1
          19929     1
          80457     1
          Name: SEVERITYCODE, dtype: int64

In [98]: from sklearn import metrics
         import matplotlib.pyplot as plt
         print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test, predTree))

          DecisionTrees's Accuracy:  0.6734069931693725
```

67 % accuracy

Unfortunately, modeling of the decision tree was not successful



```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-195-924cece2ccdb> in <module>
      6 fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (6,6), dpi=300)
      7
----> 8 forest_1.plot_tree(ClassTree,
      9                feature_names = train_X.columns,
     10                class_names=ClassTree.classes_,

AttributeError: 'RandomForestClassifier' object has no attribute 'plot_tree'
```

Additional methods were attempted without success

Results

- Sadly, since none of the methods are worked for me in my notebook, I do not have real results to report.

- I have been working on this all day, every day, for 3 days now and would be happy to put in more work if I thought that would yield a superior result.

- I am completely befuddled by the absence of instructors in this course (I asked multiple questions and only received answers from other students).

- Perhaps my mistake was choosing the shared data set? I would be interested in any feedback.

Discussion

- Let’s discuss! If you would be interested in forming a study-group or discussing these topics, please email me at mirandacchilds@gmail.com

- I would still like to learn these subjects despite having put so much time into this program, yet learning very little. I will definitely be looking into other programs, people to study with, and hoping to find a mentor.

Conclusion and recommendations

- A more experienced data science can provide a superior analysis of the data
- Accuracy of the models can increase with better data collection. For example ‘Unknown’ or ‘Other’ should never be listed as weather conditions.
- Perhaps a citywide campaign can emphasize the hazards of driving during poor weather, light, and road conditions. Up-to-date digital signage can also be considered.