Seattle Collision Analysis

By Zhaojie He

Introdution

- This is a dataset of car accidents from 2004 to the present in Seattle. This data set includes 38 attributes such as the severity of the accident, the number of casualties, the weather conditions at the time of the accident, and the latitude and longitude of the accident location. I will use decision tree classifier to predict the severity code of this dataset.

Analysis

- Delete Non-using attributes

```
In [45]: DESC','INCDATE','INCDTTM','SDOT_COLDESC','ST_COLDESC','SEGLANEKEY','CROSSWALKKEY','SEVERITYCODE.1','EXCEPTRSNCODE','SPEEDING','PEDROWNOTGRNT','INATTENTIONIND','LOCATION','SDOTCOLNUM','ST_COLCODE','REPORTNO','SDOT_COLCODE'],axis

In [46]: df_d.head()
```

Out[46]:

	SEVERITYCODE	STATUS	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEHCOUNT	JUNCTIONTYPE	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND	HITPARKEDCAR
0	2	Matched	Intersection	Angles	2	0	0	2	At Intersection (intersection related)	N	Overcast	Wet	Daylight	N
1	1	Matched	Block	Sideswipe	2	0	0	2	Mid-Block (not related to intersection)	0	Raining	Wet	Dark - Street Lights On	N
2	1	Matched	Block	Parked Car	4	0	0	3	Mid-Block (not related to intersection)	0	Overcast	Dry	Daylight	N
3	1	Matched	Block	Other	3	0	0	3	Mid-Block (not related to intersection)	N	Clear	Dry	Daylight	N
4	2	Matched	Intersection	Angles	2	0	0	2	At Intersection (intersection related)	0	Raining	Wet	Daylight	N

- Delete NA

```
In [49]: df_clean=df_d.dropna(axis=0)
         df_clean.isna().sum()
```

Out[49]:

```
SEVERITYCODE    0
STATUS          0
ADDRTYPE        0
COLLISIONTYPE   0
PERSONCOUNT     0
PEDCOUNT        0
PEDCYLCOUNT     0
VEHCOUNT        0
JUNCTIONTYPE    0
UNDERINFL       0
WEATHER         0
ROADCOND        0
LIGHTCOND       0
HITPARKEDCAR    0
dtype: int64
```

Analysis

- Transform categorical variables to numeric variables

```
In [57]: X[1:5]

Out[57]: array([['Matched', 'Block', 'Sideswipe', 2, 0, 0, 2,
                 'Mid-Block (not related to intersection)', 0, 'Raining', 'Wet',
                 'Dark - Street Lights On', 0],
                ['Matched', 'Block', 'Parked Car', 4, 0, 0, 3,
                 'Mid-Block (not related to intersection)', 0, 'Overcast', 'Dry',
                 'Daylight', 0],
                ['Matched', 'Block', 'Other', 3, 0, 0, 3,
                 'Mid-Block (not related to intersection)', 0, 'Clear', 'Dry',
                 'Daylight', 0],
                ['Matched', 'Intersection', 'Angles', 2, 0, 0, 2,
                 'At Intersection (intersection related)', 0, 'Raining', 'Wet',
                 'Daylight', 0]], dtype=object)
```

before

```
In [75]: X[1:5]

Out[75]: array([[0, 1, 9, 2, 0, 0, 2, 4, 0, 6, 8, 2, 0],
                [0, 1, 5, 4, 0, 0, 3, 4, 0, 4, 0, 5, 0],
                [0, 1, 4, 3, 0, 0, 3, 4, 0, 1, 0, 5, 0],
                [0, 2, 0, 2, 0, 0, 2, 1, 0, 6, 8, 5, 0]], dtype=object)
```

after

Analysis

- Build decision tree model

SETTING UP THE DECSION TREE

In [36]:
```
from sklearn.model_selection import train_test_split
```

In [77]:
```
X_trainset, X_testset, y_trainset, y_testset = train_test_split(X, y, test_size=0.3, random_state=3)
```

In [78]:
```
from sklearn.tree import DecisionTreeClassifier
Tree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
Tree
```

Out[78]:
```
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 [79]:
```
Tree.fit(X_trainset,y_trainset)
```

Out[79]:
```
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')
```

Analysis

- Accuracy

Predict

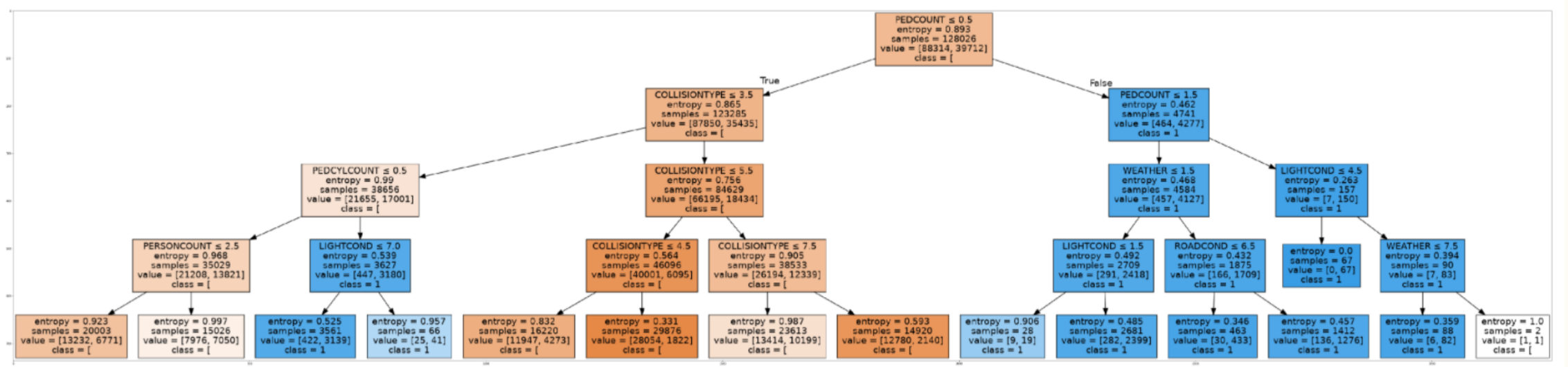
In [80]:
```
predTree = Tree.predict(X_testset)
```

In [81]:
```
from sklearn import metrics
import matplotlib.pyplot as plt
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_testset, predTree))
```

```
DecisionTrees's Accuracy:  0.7446827899178042
```

Analysis

- Visualization



Conclusion

- In conclusion, the decision tree model is qualified, with an accuracy rate of 74.46%. For building a decision tree, there are a total of 13 variables in the input data. But in the end, there are only 6 variables used to build a decision tree. They are PEDCOUNT (The number of pedestrians involved in the collision.), COLLISIONTYPE, PEDCYLCOUNT (The number of bicycles involved in the collision.), LIGHTCOND (The light conditions during the collision.), ROADCOND (The condition of the road during the collision.) and WEATHER. This shows that the road conditions, weather, and light conditions are indeed related to the accident. And through these data, the severity of the accident can be predicted, so that it can be decided in advance whether to dispatch an ambulance.

Thank you!