Accidents-Severity-Prediction-Analysis

Yanying Zhou

Introduction

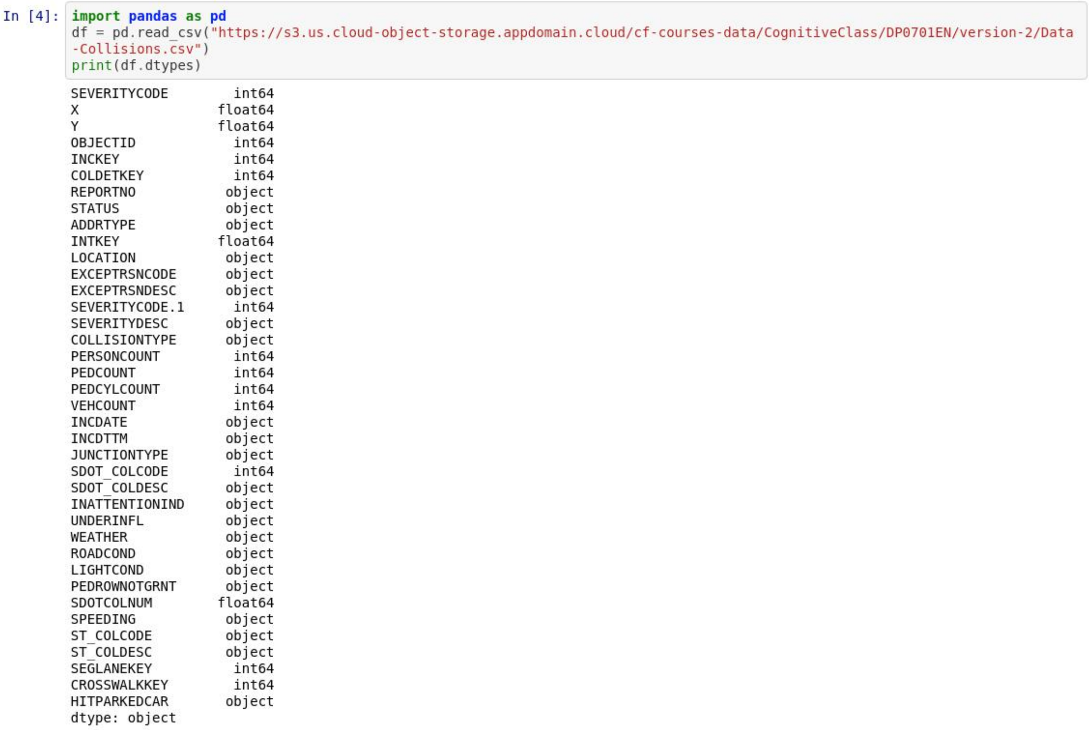
In order to reduce the frequency of car accidents, I would like to use the existing dataset to predict the severity of the accident with the current weather, vehicle speed, road conditions and light conditions. When the prediction are bad, an alarm system will be activated to remind drivers to increase their vigilance or remind local police to make adequate preparations in advance.

Datasource

The dataset is the Example Dataset in Week1 on Applied Data Science Capstone. This dataset provides collisions from 2004 to the present in Seattle.

Data Analysis

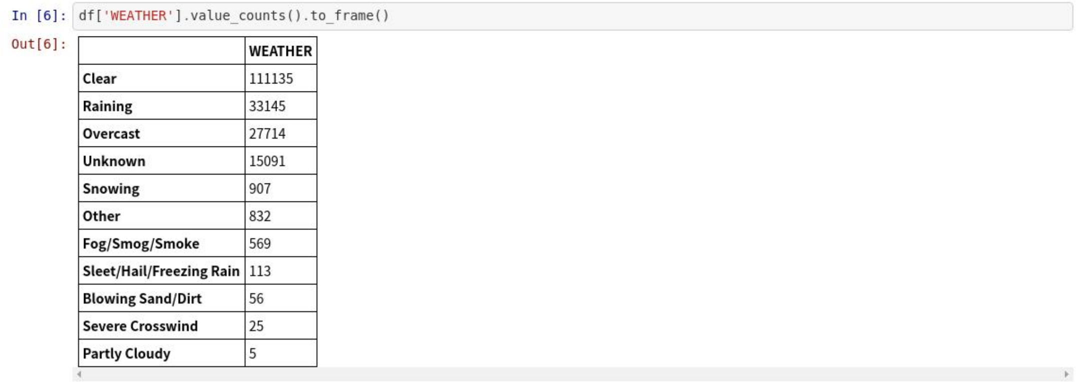
First, I read the data and find the attributes related to car accidents, for example, SEVERITYCODE,SEVERITYDESC, WEATHER, ROADCOND and LIGHTCOND.



```
In [4]: import pandas as pd
        df = pd.read_csv("https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data
        -Collisions.csv")
        print(df.dtypes)
```

```
SEVERITYCODE        int64
X                 float64
Y                 float64
OBJECTID            int64
INCKEY              int64
COLDETKEY           int64
REPORTNO           object
STATUS             object
ADDRTYPE           object
INTKEY            float64
LOCATION           object
EXCEPTRSNCODE      object
EXCEPTRSNDESC      object
SEVERITYCODE.1      int64
SEVERITYDESC       object
COLLISIONTYPE      object
PERSONCOUNT         int64
PEDCOUNT            int64
PEDCYLCOUNT         int64
VEHCOUNT            int64
INCDATE            object
INCDTTM            object
JUNCTIONTYPE       object
SDOT_COLCODE        int64
SDOT_COLDESC       object
INATTENTIONIND     object
UNDERINFL          object
WEATHER            object
ROADCOND           object
LIGHTCOND          object
PEDROWNOTGRNT      object
SDOTCOLNUM        float64
SPEEDING           object
ST_COLCODE         object
ST_COLDESC         object
SEGLANEKEY          int64
CROSSWALKKEY        int64
HITPARKEDCAR       object
dtype: object
```

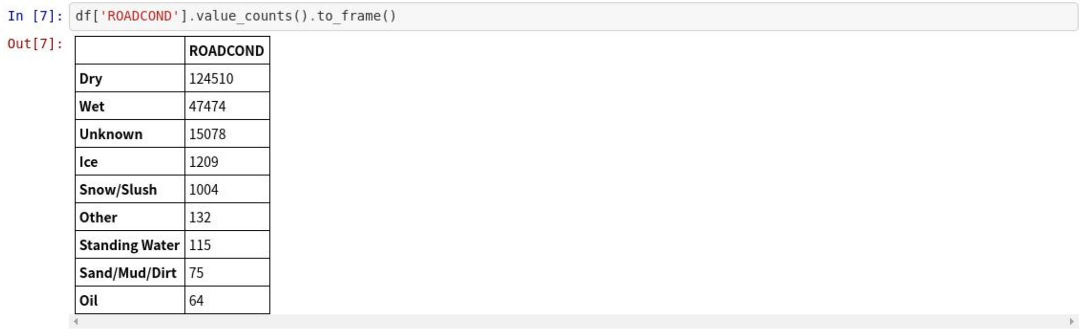
Then, run the value count on WEATHER, ROADCOND and LIGHTCOND to see which type of roads had more accidents.



```
In [6]: df['WEATHER'].value_counts().to_frame()
```

Out[6]:

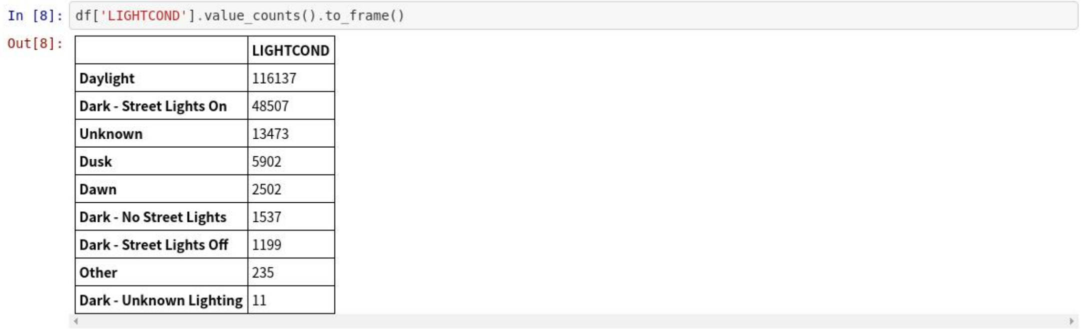
	WEATHER
Clear	111135
Raining	33145
Overcast	27714
Unknown	15091
Snowing	907
Other	832
Fog/Smog/Smoke	569
Sleet/Hail/Freezing Rain	113
Blowing Sand/Dirt	56
Severe Crosswind	25
Partly Cloudy	5



```
In [7]: df['ROADCOND'].value_counts().to_frame()
```

Out[7]:

	ROADCOND
Dry	124510
Wet	47474
Unknown	15078
Ice	1209
Snow/Slush	1004
Other	132
Standing Water	115
Sand/Mud/Dirt	75
Oil	64

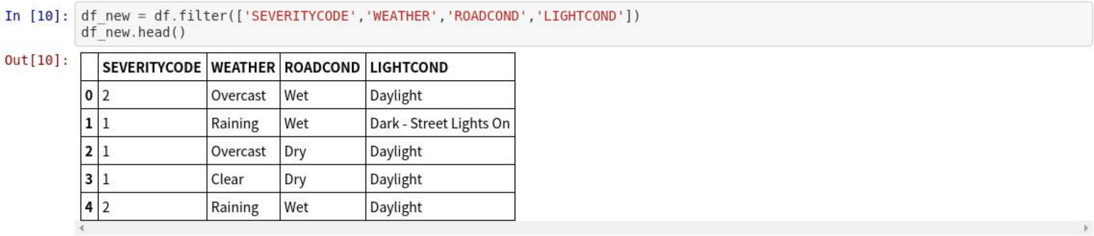


```
In [8]: df['LIGHTCOND'].value_counts().to_frame()
```

Out[8]:

	LIGHTCOND
Daylight	116137
Dark - Street Lights On	48507
Unknown	13473
Dusk	5902
Dawn	2502
Dark - No Street Lights	1537
Dark - Street Lights Off	1199
Other	235
Dark - Unknown Lighting	11

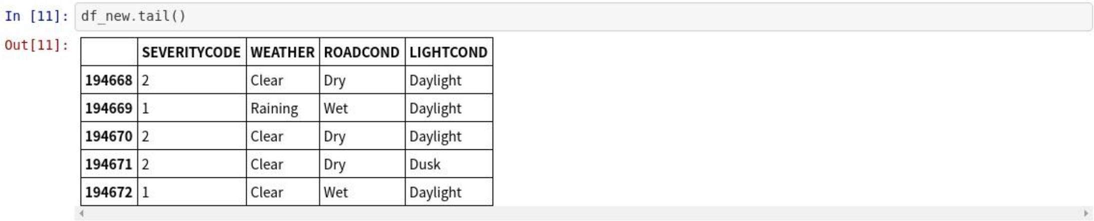
Obviously, clear weather with dry road had the most accidents in day time. So, I create a new dataframe.



```
In [10]: df_new = df.filter(['SEVERITYCODE','WEATHER','ROADCOND','LIGHTCOND'])
         df_new.head()
```

Out[10]:

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



```
In [11]: df_new.tail()
```

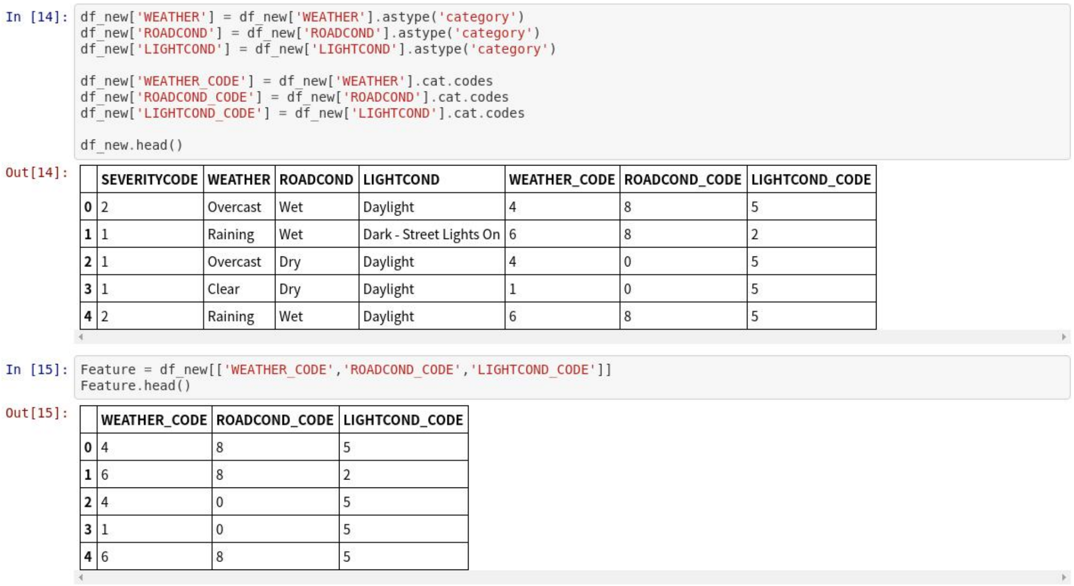
Out[11]:

	SEVERITYCODE	WEATHER	ROADCOND	LIGHTCOND
194668	2	Clear	Dry	Daylight
194669	1	Raining	Wet	Daylight
194670	2	Clear	Dry	Daylight
194671	2	Clear	Dry	Dusk
194672	1	Clear	Wet	Daylight

Methodology

I try to use machine learning model to analysis.

KNN



In [14]:

```
df_new['WEATHER'] = df_new['WEATHER'].astype('category')
df_new['ROADCOND'] = df_new['ROADCOND'].astype('category')
df_new['LIGHTCOND'] = df_new['LIGHTCOND'].astype('category')

df_new['WEATHER_CODE'] = df_new['WEATHER'].cat.codes
df_new['ROADCOND_CODE'] = df_new['ROADCOND'].cat.codes
df_new['LIGHTCOND_CODE'] = df_new['LIGHTCOND'].cat.codes

df_new.head()
```

Out[14]:

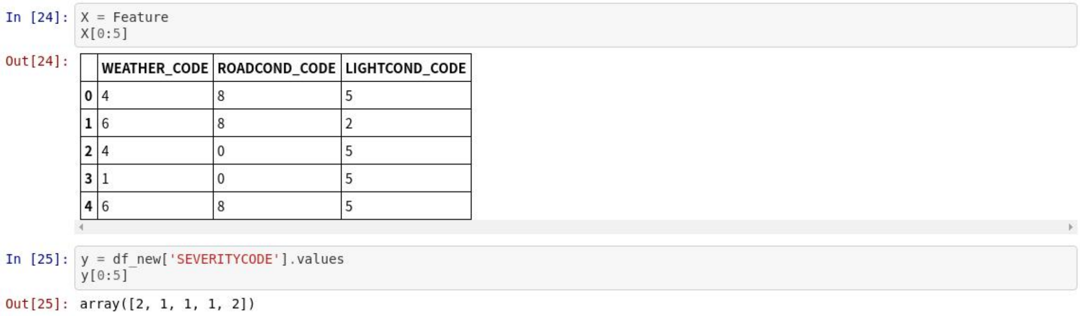
	SEVERITYCODE	WEATHER	ROADCOND	LIGHTCOND	WEATHER_CODE	ROADCOND_CODE	LIGHTCOND_CODE
0	2	Overcast	Wet	Daylight	4	8	5
1	1	Raining	Wet	Dark - Street Lights On	6	8	2
2	1	Overcast	Dry	Daylight	4	0	5
3	1	Clear	Dry	Daylight	1	0	5
4	2	Raining	Wet	Daylight	6	8	5

In [15]:

```
Feature = df_new[['WEATHER_CODE','ROADCOND_CODE','LIGHTCOND_CODE']]
Feature.head()
```

Out[15]:

	WEATHER_CODE	ROADCOND_CODE	LIGHTCOND_CODE
0	4	8	5
1	6	8	2
2	4	0	5
3	1	0	5
4	6	8	5



In [24]:

```
X = Feature
X[0:5]
```

Out[24]:

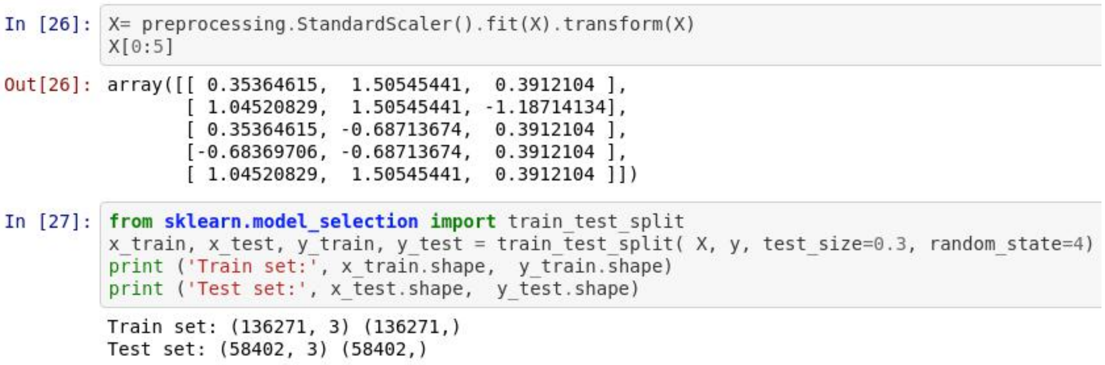
	WEATHER_CODE	ROADCOND_CODE	LIGHTCOND_CODE
0	4	8	5
1	6	8	2
2	4	0	5
3	1	0	5
4	6	8	5

In [25]:

```
y = df_new['SEVERITYCODE'].values
y[0:5]
```

Out[25]:

```
array([2, 1, 1, 1, 2])
```



In [26]:

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

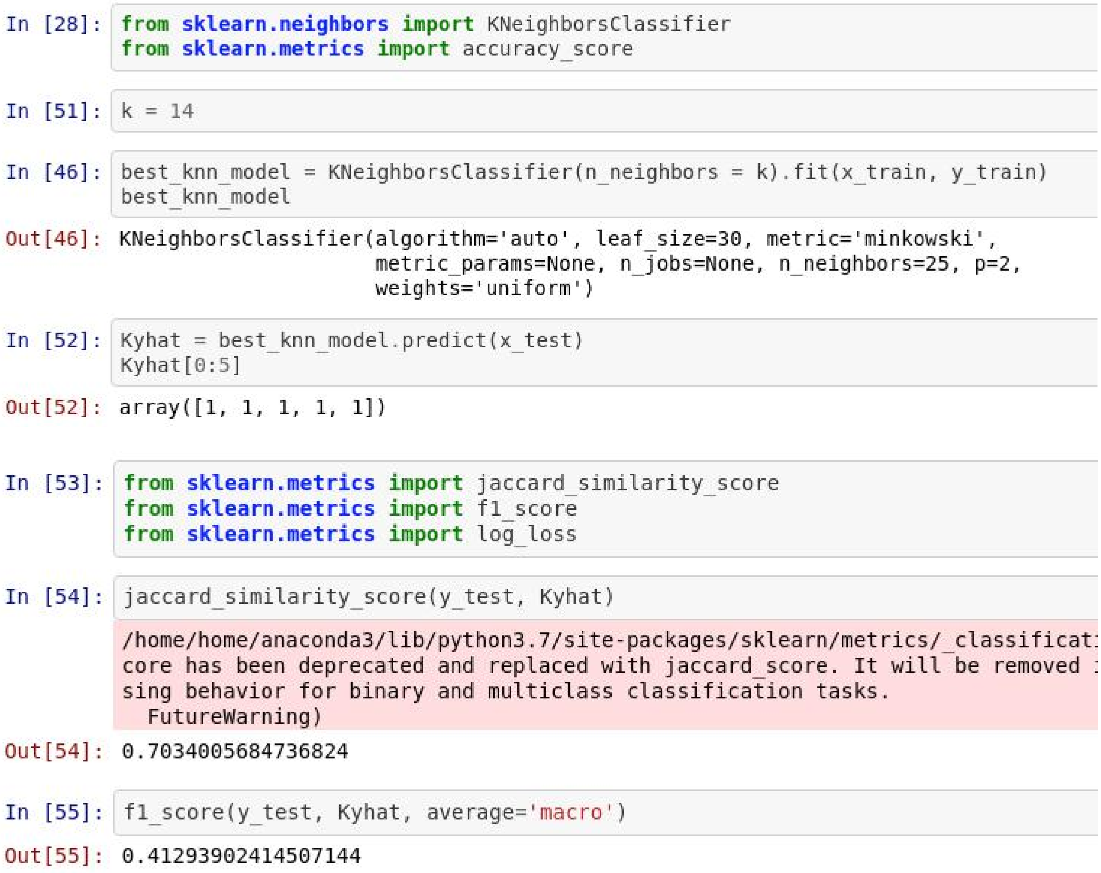
Out[26]:

```
array([[ 0.35364615,  1.50545441,  0.3912104 ],
       [ 1.04520829,  1.50545441, -1.18714134],
       [ 0.35364615, -0.68713674,  0.3912104 ],
       [-0.68369706, -0.68713674,  0.3912104 ],
       [ 1.04520829,  1.50545441,  0.3912104 ]])
```

In [27]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state=4)
print ('Train set:', x_train.shape,  y_train.shape)
print ('Test set:', x_test.shape,  y_test.shape)
```

```
Train set: (136271, 3) (136271,)
Test set: (58402, 3) (58402,)
```



```
In [28]: from sklearn.neighbors import KNeighborsClassifier
         from sklearn.metrics import accuracy_score

In [51]: k = 14

In [46]: best_knn_model = KNeighborsClassifier(n_neighbors = k).fit(x_train, y_train)
         best_knn_model

Out[46]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                              metric_params=None, n_jobs=None, n_neighbors=25, p=2,
                              weights='uniform')

In [52]: Kyhat = best_knn_model.predict(x_test)
         Kyhat[0:5]

Out[52]: array([1, 1, 1, 1, 1])

In [53]: from sklearn.metrics import jaccard_similarity_score
         from sklearn.metrics import f1_score
         from sklearn.metrics import log_loss

In [54]: jaccard_similarity_score(y_test, Kyhat)

         /home/home/anaconda3/lib/python3.7/site-packages/sklearn/metrics/_classificati
         core has been deprecated and replaced with jaccard_score. It will be removed i
         sing behavior for binary and multiclass classification tasks.
           FutureWarning)
Out[54]: 0.7034005684736824

In [55]: f1_score(y_test, Kyhat, average='macro')

Out[55]: 0.41293902414507144
```

Discussion

According to result, we can see there had much more accidents on Clear Days with dry road in day time. There had much less collisions on raining days with wet roads with dark light. There may two reason: one is that people will be more careful when conditions are bad, and the other is that there will be much more clear days which enlarge the count.

Conclusion

Based on historical data related to weather conditions, we can conclude the relationship between the probability of accidents and the special weather conditions.

Thanks for your reading!