In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
df = pd.read_csv("last_two_years_accidents.csv")
```

In [4]:

```
df.columns
```

Out[4]:

```
Index(['ID', 'Source', 'Severity', 'Start_Time', 'End_Time', 'Start_Lat',
       'Start_Lng', 'Distance(mi)', 'Street', 'City', 'County', 'State',
       'Zipcode', 'Country', 'Timezone', 'Airport_Code', 'Weather_Timestamp',
       'Temperature(F)', 'Humidity(%)', 'Pressure(in)', 'Visibility(mi)',
       'Wind_Direction', 'Wind_Speed(mph)', 'Weather_Condition', 'Amenity',
       'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway',
       'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal',
       'Turning_Loop', 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilight',
       'Astronomical_Twilight'],
      dtype='object')
```

In [5]:

```
df['Severity'].head()
```

Out[5]:

```
0    1
1    1
2    1
3    1
4    2
Name: Severity, dtype: int64
```

In [6]:

```
severity_labels = {
    1: 'Not Severe',
    2: 'Not Severe',
    3: 'Severe',
    4: 'Severe'
}
```

In [7]:

```
df['Severity'] = df['Severity'].map(severity_labels)
```

In [8]:

```
df['Severity']
```

Out[8]:

```
0          Not Severe
1          Not Severe
2          Not Severe
3          Not Severe
4          Not Severe
              ...
2009080    Not Severe
2009081    Not Severe
2009082    Not Severe
2009083    Not Severe
2009084    Not Severe
Name: Severity, Length: 2009085, dtype: object
```

In [9]:

```
df['Severity'].value_counts()
```

Out[9]:

```
Severity
Not Severe    1880793
Severe         128292
Name: count, dtype: int64
```

In [10]:

```
# huge unbalanced data
# to make it balanced we have to use a sampling technique with same number of samples for each of the category.
```

In [11]:

```
df['Severity'].value_counts()['Severe']
```

Out[11]:

```
128292
```

In [12]:

```
size = df['Severity'].value_counts()['Severe']
```

In [13]:
```
size
```
Out[13]:
```
128292
```
In [14]:
```
df_balanced_severity = pd.DataFrame()
```
In [15]:
```
df_balanced_severity
```
Out[15]:
```
_
```
In [16]:
```
df_balanced_severity = df.groupby('Severity', group_keys = False).apply(lambda x: x.sample(size, random_state = 30)
```
In [17]:
```
df_balanced_severity['Severity'].value_counts()
```
Out[17]:
```
Severity
Not Severe    128292
Severe        128292
Name: count, dtype: int64
```
In [18]:
```
df_balanced_severity.info()
```
```
<class 'pandas.core.frame.DataFrame'>
Index: 256584 entries, 461355 to 109670
Data columns (total 41 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   ID                     256584 non-null  object 
 1   Source                 256584 non-null  object 
 2   Severity               256584 non-null  object 
 3   Start_Time             256584 non-null  object 
 4   End_Time               256584 non-null  object 
 5   Start_Lat              256584 non-null  float64
 6   Start_Lng              256584 non-null  float64
 7   Distance(mi)           256584 non-null  float64
 8   Street                 256584 non-null  object 
 9   City                   256584 non-null  object 
 10  County                 256584 non-null  object 
 11  State                  256584 non-null  object 
 12  Zipcode                256584 non-null  object 
 13  Country                256584 non-null  object 
 14  Timezone               256584 non-null  object 
 15  Airport_Code           256584 non-null  object 
 16  Weather_Timestamp      256584 non-null  object 
 17  Temperature(F)         256584 non-null  float64
 18  Humidity(%)            256584 non-null  float64
 19  Pressure(in)           256584 non-null  float64
 20  Visibility(mi)         256584 non-null  float64
 21  Wind_Direction         256584 non-null  object 
 22  Wind_Speed(mph)        256584 non-null  float64
 23  Weather_Condition      256584 non-null  object 
 24  Amenity                256584 non-null  bool   
 25  Bump                   256584 non-null  bool   
 26  Crossing               256584 non-null  bool   
 27  Give_Way               256584 non-null  bool   
 28  Junction               256584 non-null  bool   
 29  No_Exit                256584 non-null  bool   
 30  Railway                256584 non-null  bool   
 31  Roundabout             256584 non-null  bool   
 32  Station                256584 non-null  bool   
 33  Stop                   256584 non-null  bool   
 34  Traffic_Calming        256584 non-null  bool   
 35  Traffic_Signal         256584 non-null  bool   
 36  Turning_Loop           256584 non-null  bool   
 37  Sunrise_Sunset         256584 non-null  object 
 38  Civil_Twilight         256584 non-null  object 
 39  Nautical_Twilight      256584 non-null  object 
 40  Astronomical_Twilight  256584 non-null  object 
dtypes: bool(13), float64(8), object(20)
memory usage: 60.0+ MB
```

In [19]:

```
df_balanced_severity['Wind_Speed(mph)']
```

Out[19]:

```
461355      8.00000
1340271    12.00000
1291327     3.00000
170756     10.00000
1675489     3.00000
             ...
935219     10.00000
1363937    15.00000
139521      9.00000
207978      6.00000
109670      7.68549
Name: Wind_Speed(mph), Length: 256584, dtype: float64
```

In [20]:

```
categorical_features = set(["Weather_Condition", "Civil_Twilight", 'Wind_Direction'])
```

In [21]:

```
for feature in categorical_features:
    df_balanced_severity[feature] = df_balanced_severity[feature].astype("category")
```

In [22]:

```
for cat in categorical_features:
    print(cat,'-', len(df_balanced_severity[cat].unique()))
```

```
Civil_Twilight - 2
Weather_Condition - 79
Wind_Direction - 18
```

In [23]:

```
df_balanced_severity['No_Exit'].head()
```

Out[23]:

```
461355     False
1340271    False
1291327    False
170756     False
1675489    False
Name: No_Exit, dtype: bool
```

In [24]:

```
bool_columns = df_balanced_severity.select_dtypes(include='bool').columns
```

In [25]:

```
bool_columns
```

Out[25]:

```
Index(['Amenity', 'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit',
       'Railway', 'Roundabout', 'Station', 'Stop', 'Traffic_Calming',
       'Traffic_Signal', 'Turning_Loop'],
      dtype='object')
```

In [26]:

```
df_balanced_severity[bool_columns] = df_balanced_severity[bool_columns].replace({True:1, False:0})
```

In [27]:

```
# One hot encoding

df2= df_balanced_severity[['Start_Lat','Start_Lng','Distance(mi)', 'Temperature(F)', 'Humidity(%)', 'Pressure(in)',
           'Visibility(mi)', 'Wind_Speed(mph)','Amenity','Bump','Crossing','Give_Way',
           'Junction','No_Exit','Railway','Roundabout','Station','Stop','Traffic_Calming','Traffic_Signal',
           'Civil_Twilight','Weather_Condition','Civil_Twilight',
           'Wind_Direction','Severity']]
```

In [28]:

```
df2 = pd.get_dummies(df2, columns=list(categorical_features), drop_first=True)
```

In [29]:

```
df2.head()
```

Out[29]:

	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Amenity	Bump	...
461355	33.921625	-84.189911	2.264	55.0	55.0	29.06	10.0	8.0	0	0	...
1340271	25.619409	-80.378894	4.503	93.0	56.0	29.98	10.0	12.0	0	0	...
1291327	39.096284	-94.593196	0.961	70.0	93.0	29.05	10.0	3.0	0	0	...
170756	34.743759	-82.621170	0.000	63.0	27.0	29.03	10.0	10.0	0	0	...
1675489	45.457275	-123.841022	0.053	41.0	100.0	30.28	10.0	3.0	0	0	...

5 rows × 118 columns

In [30]:

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import MultinomialNB
```

In [31]:

```
Y = df2['Severity'] # target column
X = df2.drop(columns = ['Severity']) # features
```

In [32]:

```
X_train, X_test, y_train, y_test = train_test_split(X, Y,test_size=0.2, random_state=30)
```

In [33]:

```
X_train
```

Out[33]:

	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Amenity	Bump	...
707893	33.902266	-118.061911	0.024	50.0	77.0	29.81	10.0	7.0	0	0	...
1054842	44.952500	-93.070125	1.045	33.0	48.0	29.21	10.0	17.0	0	0	...
883004	41.279425	-76.405730	2.440	31.0	69.0	29.32	6.0	16.0	0	0	...
627204	45.445172	-122.736672	0.446	70.0	42.0	29.94	10.0	5.0	0	0	...
100847	33.766865	-86.633484	0.000	77.0	64.0	29.28	10.0	8.0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
1057146	35.239259	-119.015526	1.663	61.0	32.0	29.75	10.0	5.0	0	0	...
76592	39.047817	-94.460548	0.000	95.0	47.0	28.68	10.0	17.0	0	0	...
634783	34.136431	-117.560211	0.100	60.0	42.0	29.06	10.0	3.0	0	0	...
86808	34.211655	-118.228027	0.000	72.0	49.0	29.01	10.0	10.0	0	0	...
325825	34.146441	-84.742452	0.689	45.0	71.0	29.49	10.0	3.0	0	0	...

205267 rows × 117 columns

In [34]:
```
X_test
```

Out[34]:

	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Amenity	Bump	...
63223	34.183231	-81.332870	0.000	68.0	100.0	29.59	5.0	0.0	0	0	...
445110	43.045431	-75.951265	1.077	57.0	67.0	29.50	10.0	8.0	0	0	...
1026617	30.014352	-90.013456	0.803	91.0	47.0	29.89	10.0	5.0	0	0	...
79219	39.774441	-105.143425	0.000	70.0	35.0	24.45	10.0	7.0	0	0	...
795597	43.015394	-83.432057	3.578	18.0	86.0	28.75	1.0	7.0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
865587	40.779766	-73.661731	1.027	25.0	41.0	29.69	10.0	30.0	0	0	...
1923669	38.228103	-81.576429	0.288	49.0	97.0	28.78	1.0	0.0	0	0	...
908859	33.891413	-79.721048	0.438	54.0	94.0	30.05	10.0	0.0	0	0	...
1306410	25.850362	-80.322286	1.970	80.0	56.0	30.17	10.0	10.0	0	0	...
674181	41.670756	-73.813543	0.557	64.0	93.0	29.64	10.0	3.0	0	0	...

51317 rows × 117 columns

In [35]:
```
y_train
```

Out[35]:
```
707893      Not Severe
1054842     Not Severe
883004          Severe
627204      Not Severe
100847      Not Severe
               ...    
1057146     Not Severe
76592           Severe
634783      Not Severe
86808           Severe
325825          Severe
Name: Severity, Length: 205267, dtype: object
```

In [36]:
```
y_test
```

Out[36]:
```
63223           Severe
445110          Severe
1026617     Not Severe
79219           Severe
795597          Severe
               ...    
865587          Severe
1923669         Severe
908859      Not Severe
1306410     Not Severe
674181          Severe
Name: Severity, Length: 51317, dtype: object
```

In [37]:
```
naive_bayes =GaussianNB()
naive_bayes.fit(X_train, y_train)
```

Out[37]:

▾ GaussianNB ⓘ ?

(https://scikit-learn.org/1.4/modules/generated/sklearn.naive_bayes.GaussianNB.html)

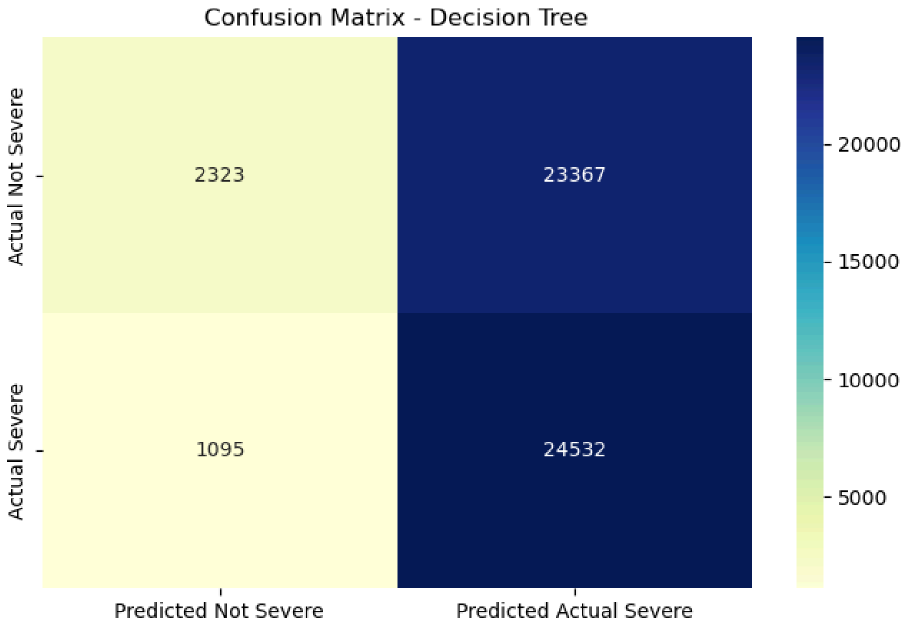
```
GaussianNB()
```

In [38]:
```
# Make predictions
y_pred = naive_bayes.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
print("Accuracy:", accuracy)
print("Confusion Matrix:")
print(conf_matrix)
```

```
Accuracy: 0.5233158602412455
Confusion Matrix:
[[ 2323 23367]
 [ 1095 24532]]
```

In [39]:

```
confmat=confusion_matrix(y_test, y_pred)

index = ["Actual Not Severe", "Actual Severe"]
columns = ["Predicted Not Severe", "Predicted Actual Severe"]
conf_matrix = pd.DataFrame(data=confmat, columns=columns, index=index)
plt.figure(figsize=(8, 5))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="YlGnBu")
plt.title("Confusion Matrix - Decision Tree")
plt.show()
```



In [ ]: