In [ ]:
```python
# imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

In [ ]:
```python
# read in the data
data = pd.read_csv('../data/processed/Cleaned_Data.csv')
```

In [ ]:
```python
# print the number of 0s in the new column
print(data['Severe_crimes'].value_counts())
```

```
Severe_crimes
0    75556
1     5373
Name: count, dtype: int64
```

In [ ]:
```python
# use the 2024 data as testing data
test_data = data[data["YEAR"] == 2024]

# use train test split to split the data into training and validation data
train_data, val_data = train_test_split(data[data["YEAR"] != 2024], test_size=0.2)

# save the data
train_data.to_csv("../data/processed/train_data.csv", index=False)
val_data.to_csv("../data/processed/val_data.csv", index=False)
test_data.to_csv("../data/processed/test_data.csv", index=False)
```

In [ ]:
```python
# describe the data
print("Training Data")
print(train_data.describe())
```

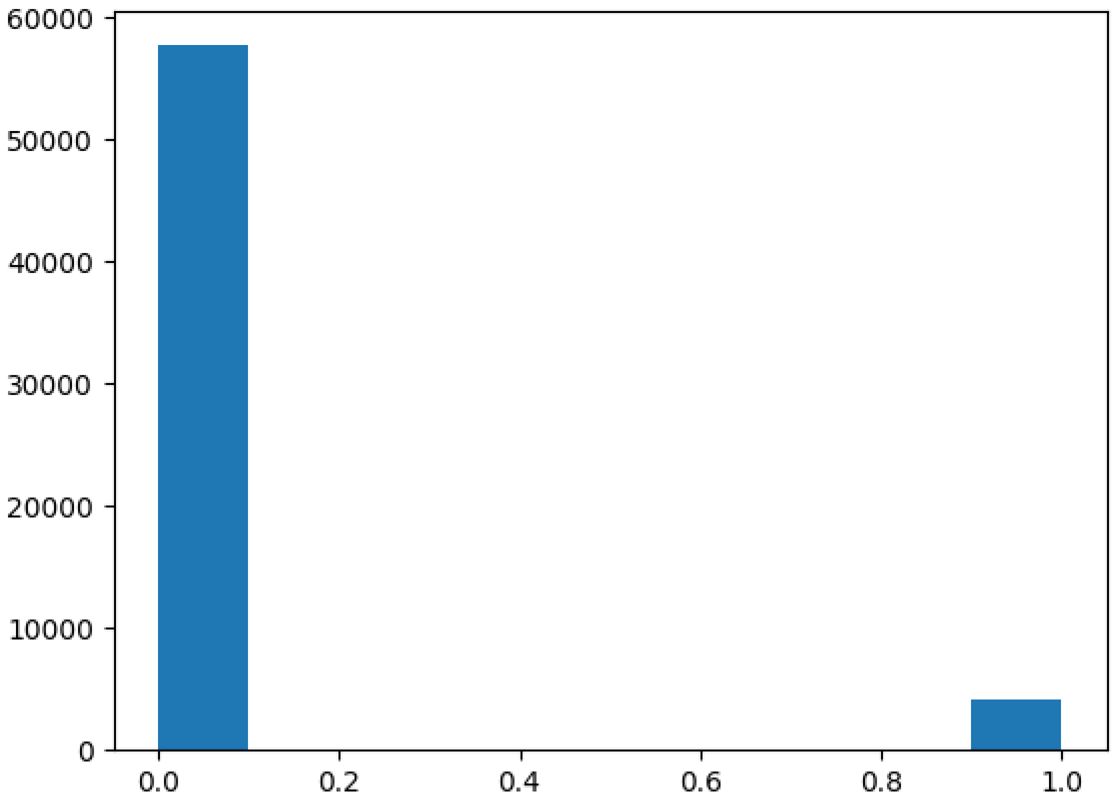
```
Training Data
                _id  OFFENSE_CODE      SHOOTING     YEAR         MONTH  \
count  61877.000000  61877.000000  61877.000000  61877.0  61877.000000
mean   39098.792152   2336.128933      0.008081   2023.0      6.625127
std    22521.612610   1177.259331      0.089529      0.0      3.416435
min        1.000000    111.000000      0.000000   2023.0      1.000000
25%    19505.000000   1106.000000      0.000000   2023.0      4.000000
50%    39223.000000   2907.000000      0.000000   2023.0      7.000000
75%    58572.000000   3201.000000      0.000000   2023.0     10.000000
max    81133.000000   3831.000000      1.000000   2023.0     12.000000

               HOUR  Severe_crimes
count  61877.000000   61877.000000
mean      12.503483       0.067246
std        6.571160       0.250450
min        0.000000       0.000000
25%        8.000000       0.000000
50%       13.000000       0.000000
75%       18.000000       0.000000
max       23.000000       1.000000
```

In [ ]:
```python
# print number of 1 and 0 s in the trainning data Severe_crimes column
print("Number of 1s and 0s in the training data")
print(train_data["Severe_crimes"].value_counts())
```

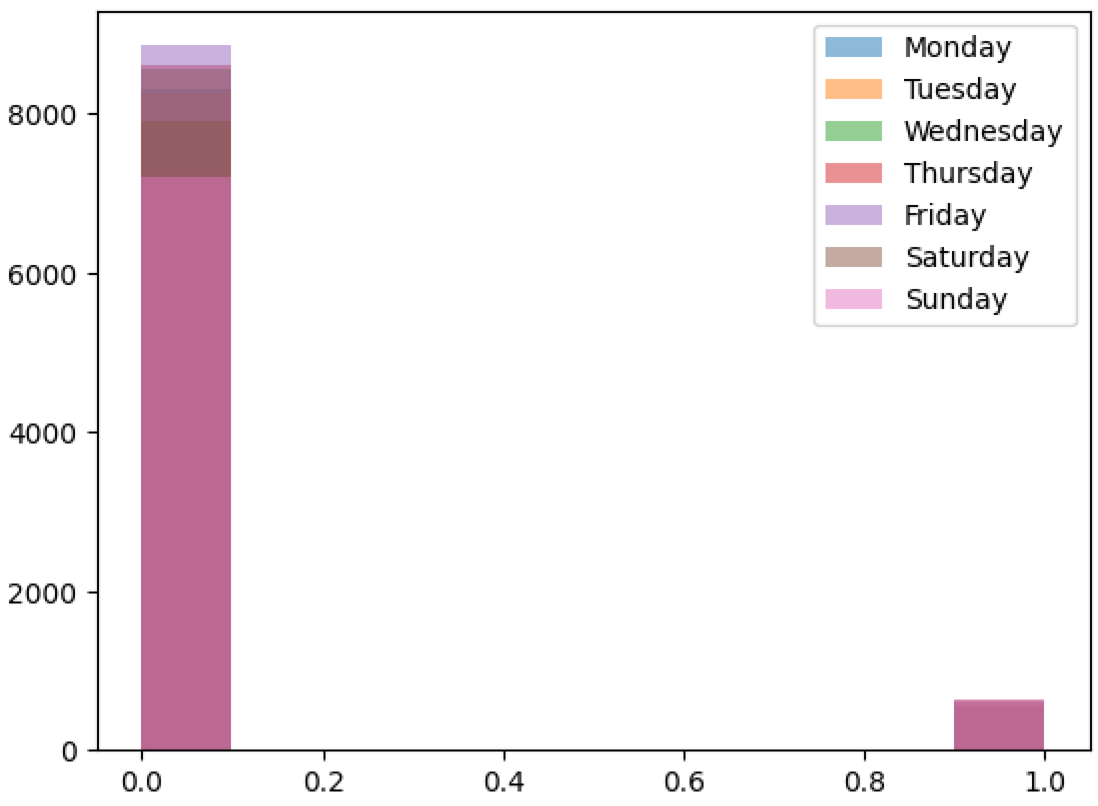
```
Number of 1s and 0s in the training data
Severe_crimes
0    57716
1     4161
Name: count, dtype: int64
```

In [ ]:
```
# do a EDA on the data
# plot the distribution of the target variable
plt.hist(train_data["Severe_crimes"])
```

Out[ ]:
```
(array([57716.,     0.,     0.,     0.,     0.,     0.,     0.,     0.,
            0.,  4161.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```



In [ ]:
```
# more EDA
# plot the distribution of the target variable by day of the week
plt.hist(train_data[train_data["DAY_OF_WEEK"] == "Monday"]["Severe_crimes"], alpha=
plt.hist(train_data[train_data["DAY_OF_WEEK"] == "Tuesday"]["Severe_crimes"], alpha
plt.hist(train_data[train_data["DAY_OF_WEEK"] == "Wednesday"]["Severe_crimes"], alp
plt.hist(train_data[train_data["DAY_OF_WEEK"] == "Thursday"]["Severe_crimes"], alph
plt.hist(train_data[train_data["DAY_OF_WEEK"] == "Friday"]["Severe_crimes"], alpha=
plt.hist(train_data[train_data["DAY_OF_WEEK"] == "Saturday"]["Severe_crimes"], alph
plt.hist(train_data[train_data["DAY_OF_WEEK"] == "Sunday"]["Severe_crimes"], alpha=
plt.legend()
# save the plot
plt.savefig("../reports/figures/Severe_crimes_by_day_of_week.png")

plt.show()
```



```
In [ ]: # print all the column names
        print(train_data.columns)
```

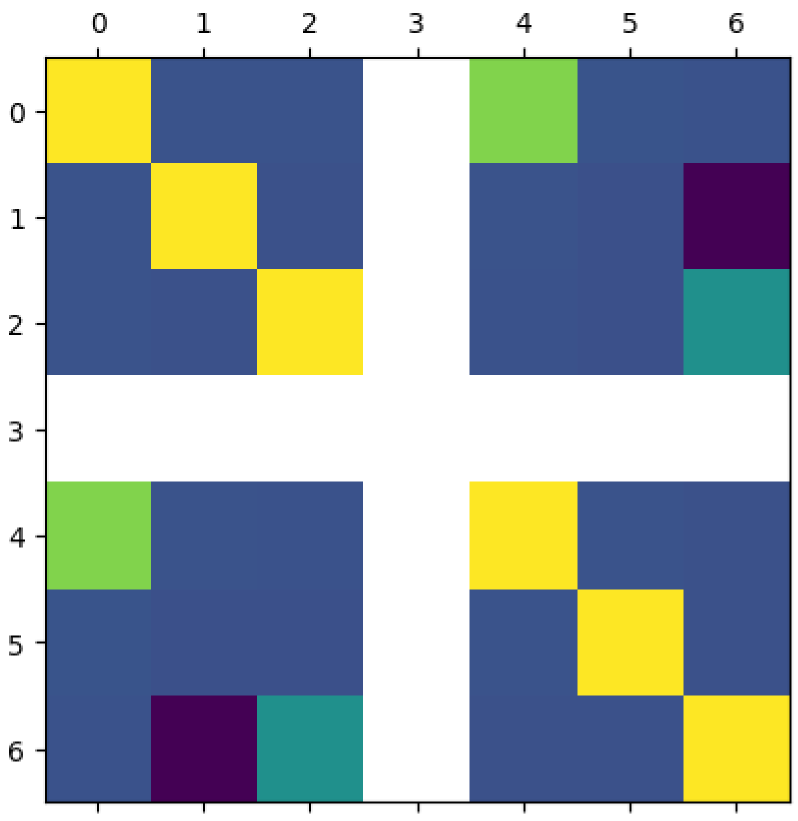
```
Index(['_id', 'OFFENSE_CODE', 'OFFENSE_DESCRIPTION', 'DISTRICT',
       'REPORTING_AREA', 'SHOOTING', 'OCCURRED_ON_DATE', 'YEAR', 'MONTH',
       'DAY_OF_WEEK', 'HOUR', 'STREET', 'Severe_crimes'],
      dtype='object')
```

```
In [ ]: # more EDA
        # analyze the relationship between the target variable and district
        districts = train_data["DISTRICT"].unique()
        for district in districts:
            print(district)
            print(train_data[train_data["DISTRICT"] == district]["Severe_crimes"].value_cou

        # more EDA
```

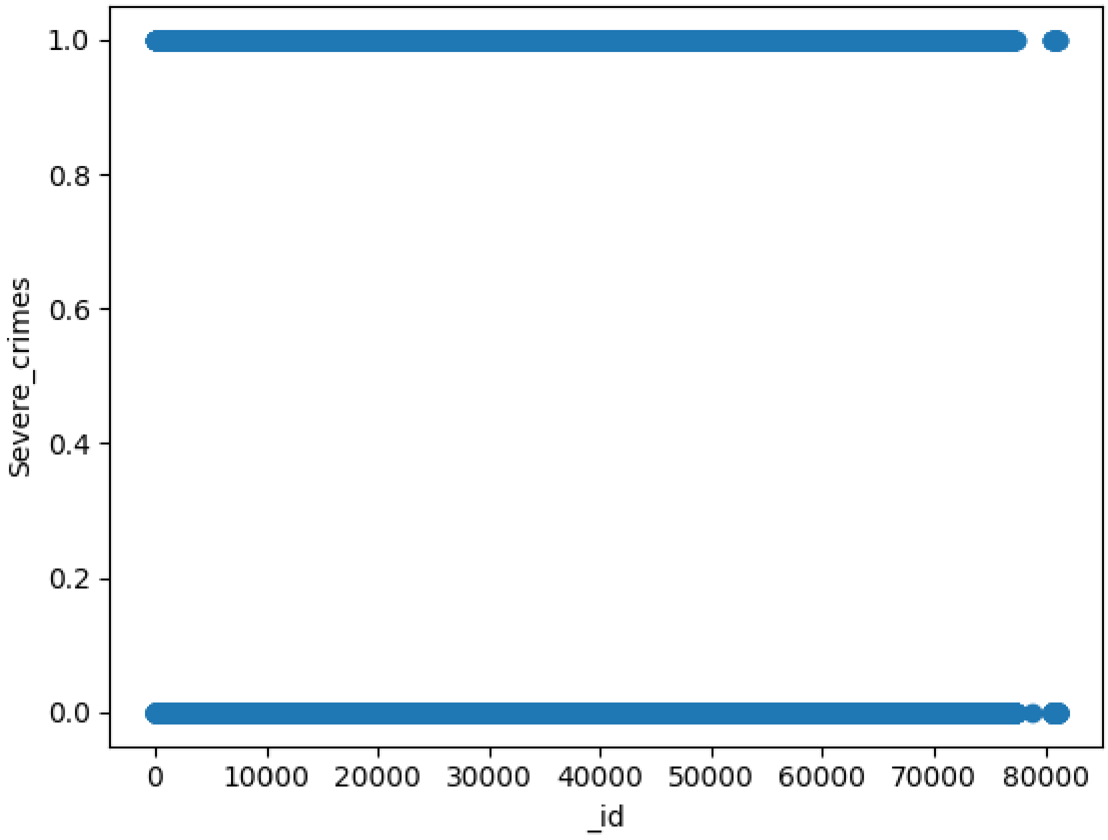
```
B3
Severe_crimes
0    5821
1     463
Name: count, dtype: int64
B2
Severe_crimes
0    7639
1     638
Name: count, dtype: int64
A15
Severe_crimes
0    1040
1      49
Name: count, dtype: int64
E18
Severe_crimes
0    3119
1     189
Name: count, dtype: int64
E5
Severe_crimes
0    2822
1     115
Name: count, dtype: int64
E13
Severe_crimes
0    3556
1     206
Name: count, dtype: int64
D4
Severe_crimes
0    7735
1     595
Name: count, dtype: int64
A7
Severe_crimes
0    3073
1     170
Name: count, dtype: int64
A1
Severe_crimes
0    6957
1     644
Name: count, dtype: int64
C6
Severe_crimes
0    4869
1     387
Name: count, dtype: int64
C11
Severe_crimes
0    6720
1     493
Name: count, dtype: int64
D14
Severe_crimes
0    4307
1     207
Name: count, dtype: int64
External
Severe_crimes
0    57
1     4
```

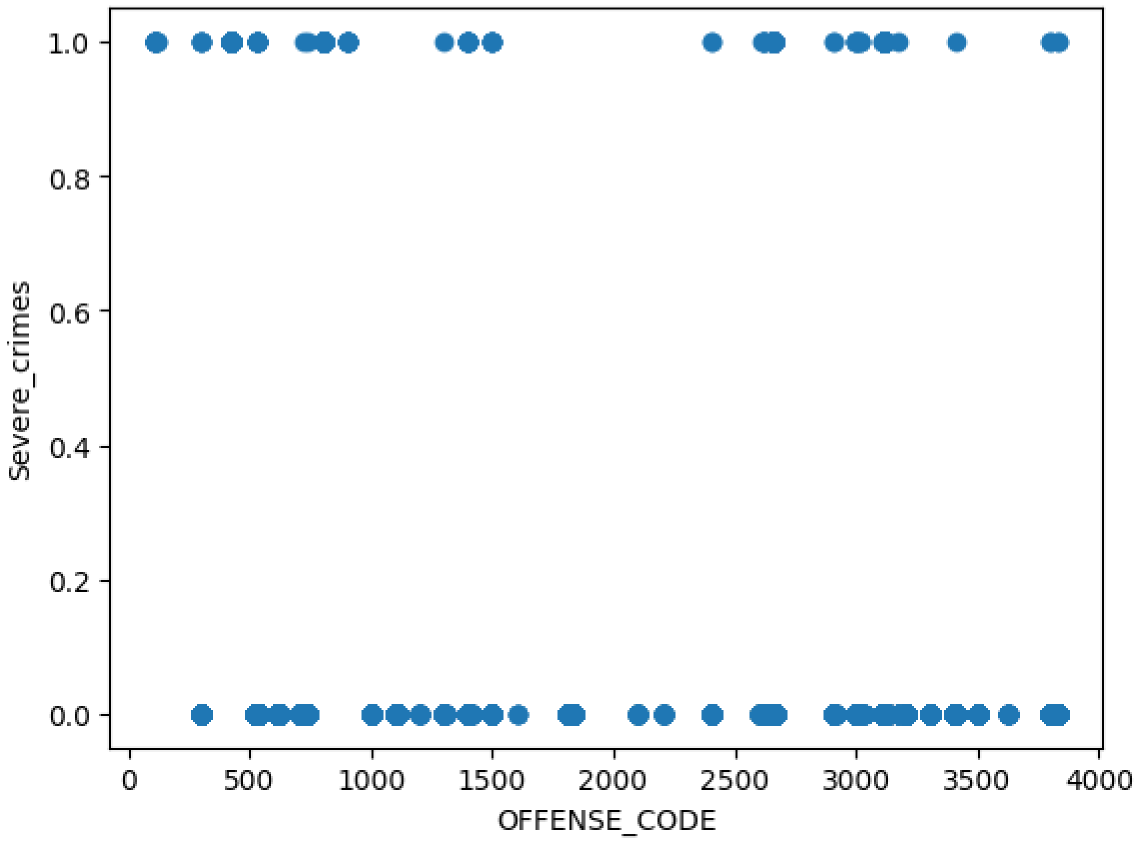
```
Name: count, dtype: int64
Outside of
Severe_crimes
1    1
0    1
Name: count, dtype: int64
```

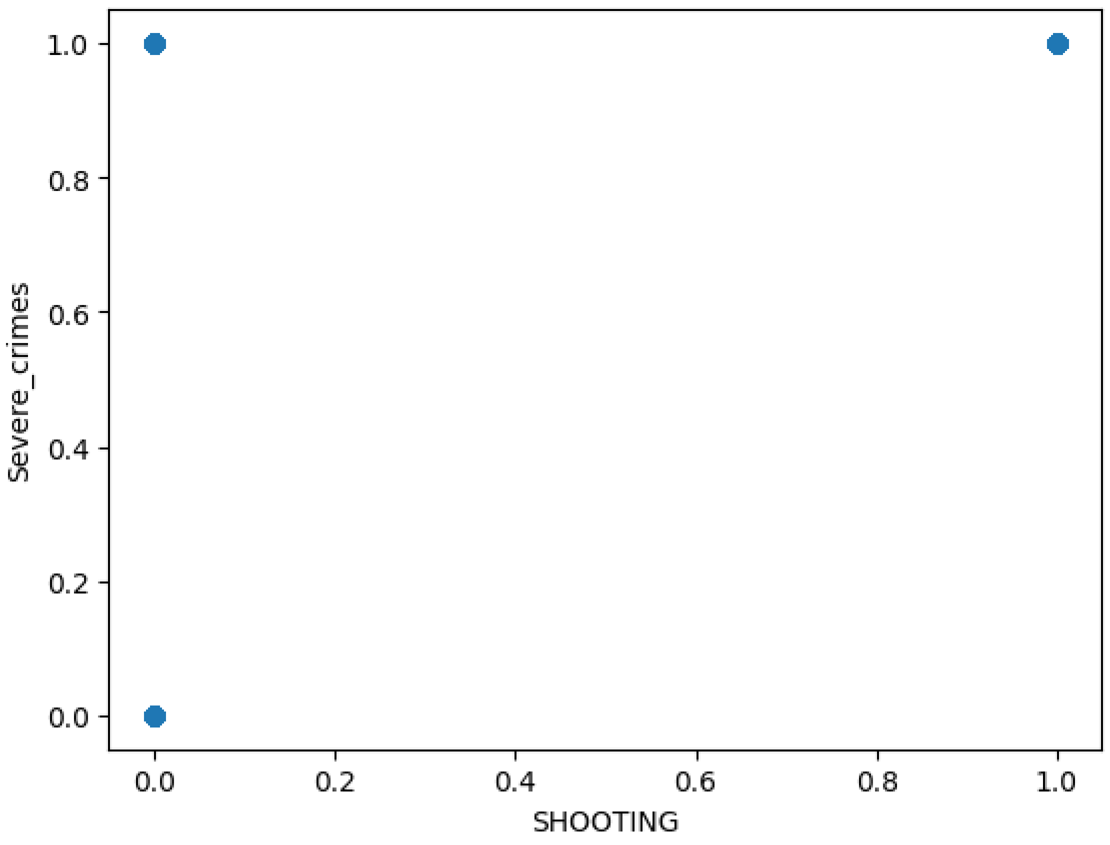
In [ ]:
```python
# draw a correlation matrix for the data
# draw all column names that are not strings

# get the columns that are numbers
numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
numeric_columns = train_data.select_dtypes(include=numerics).columns
# draw the correlation matrix
correlation_matrix = train_data[numeric_columns].corr()
plt.matshow(correlation_matrix)

# save the plot
plt.savefig("../reports/figures/correlation_matrix.png")

plt.show()
```



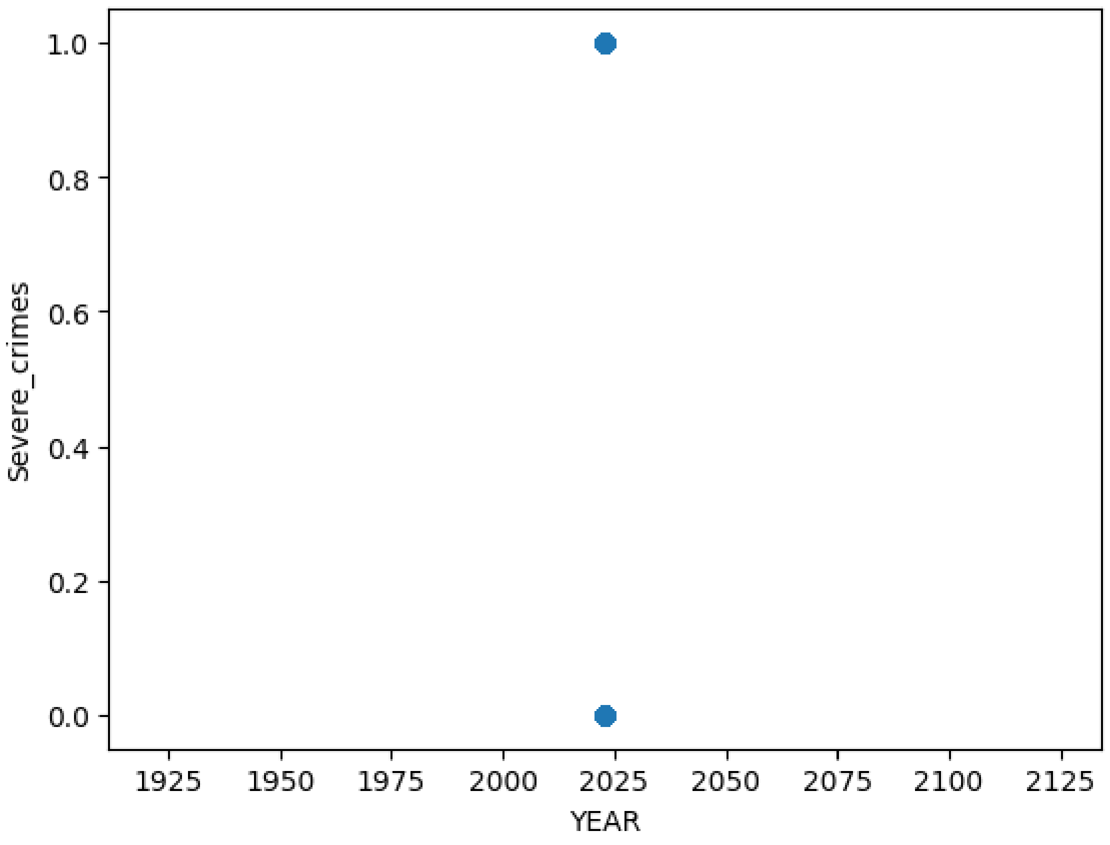
In [ ]:
```python
# Bivariate analysis of the data
# plot the relationship between the target variable and the other variables
# plot the relationship between the target variable and the district
for column in numeric_columns:
    if column != "Severe_crimes":
        plt.scatter(train_data[column], train_data["Severe_crimes"])
        plt.xlabel(column)
        plt.ylabel("Severe_crimes")
        plt.show()
```


Severe_crimes
1.0
0.8
0.6
0.4
0.2
0.0
0
10000
20000
30000
40000
50000
60000
70000
80000
_id

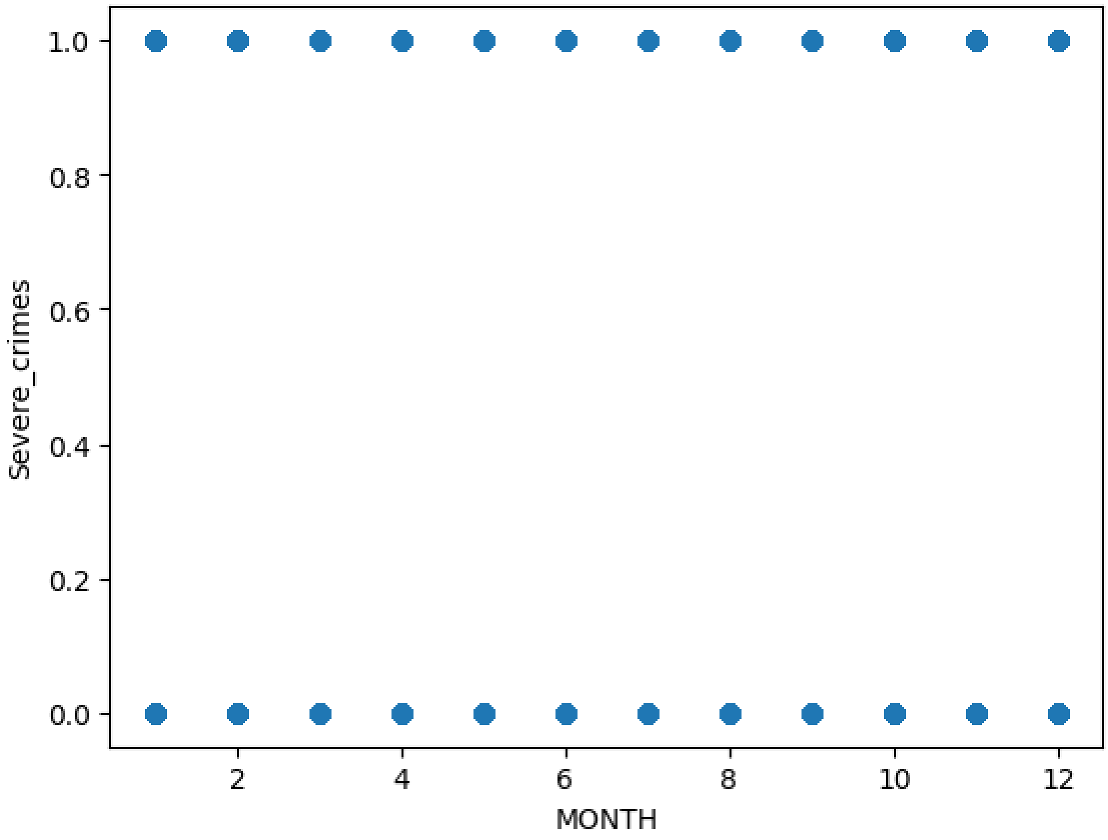

Severe_crimes
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
3000
3500
4000
OFFENSE_CODE

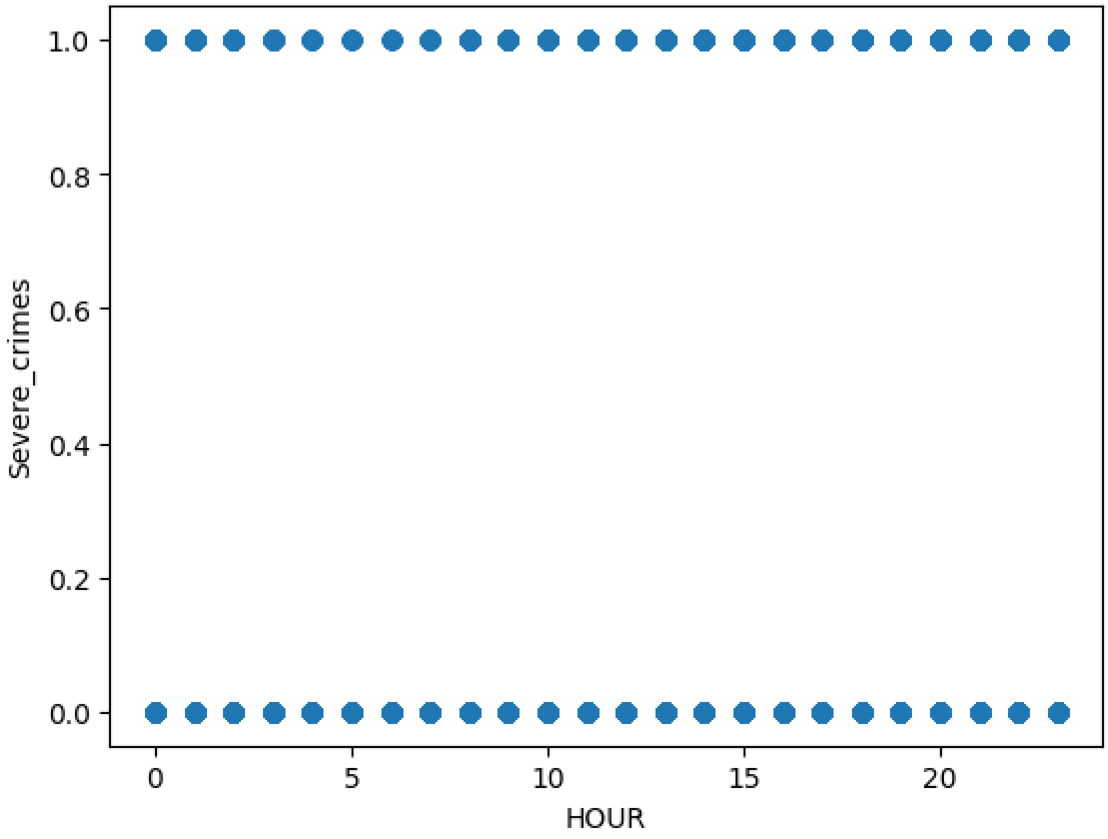


Severe_crimes
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
SHOOTING



Severe_crimes
1.0
0.8
0.6
0.4
0.2
0.0
1925
1950
1975
2000
2025
2050
2075
2100
2125
YEAR



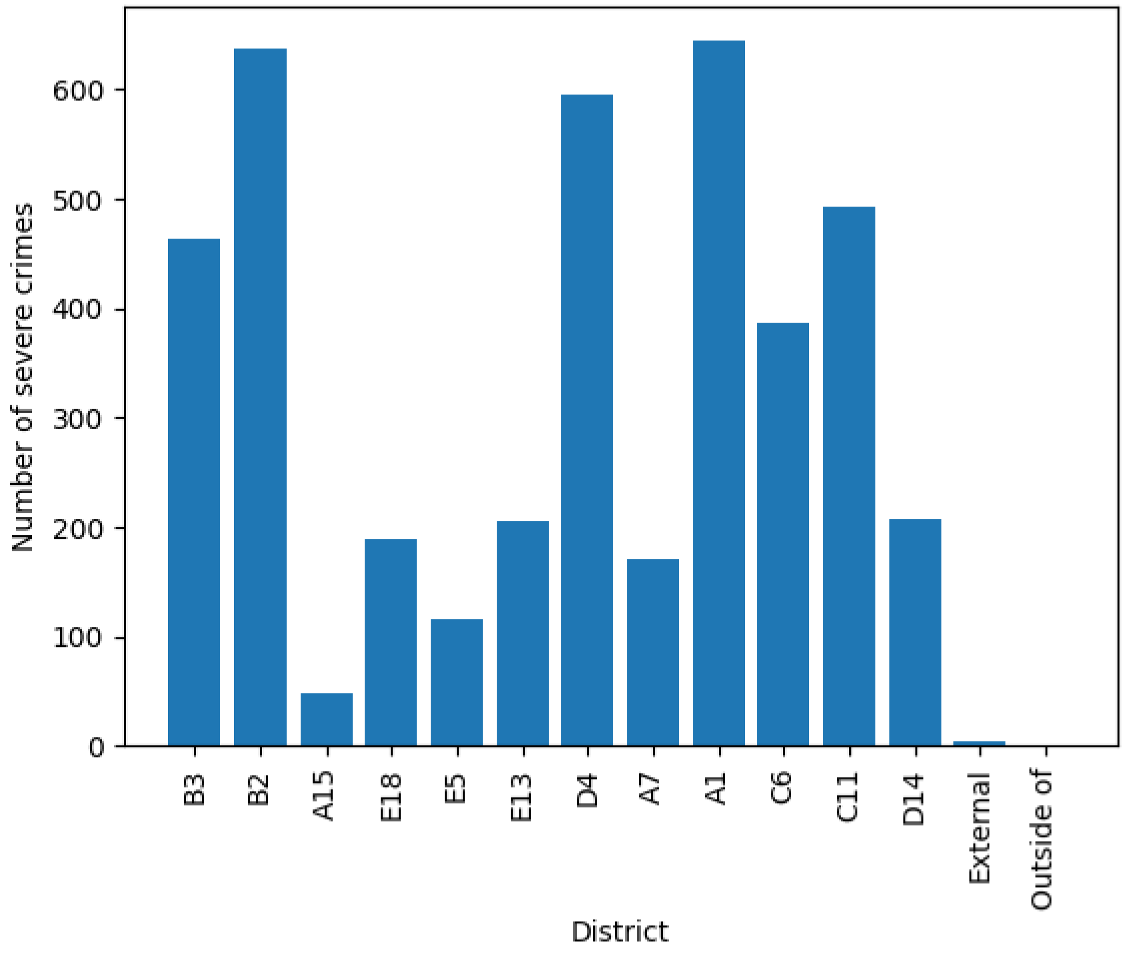


```
In [ ]: # EDA
        # plot the number of severe crimes by district
        districts = train_data["DISTRICT"].unique()
        for district in districts:
            print(district)
            print(train_data[train_data["DISTRICT"] == district]["Severe_crimes"].value_cou

        # plot both the number of severe crimes by district in one plot
        severe_crimes = []
```

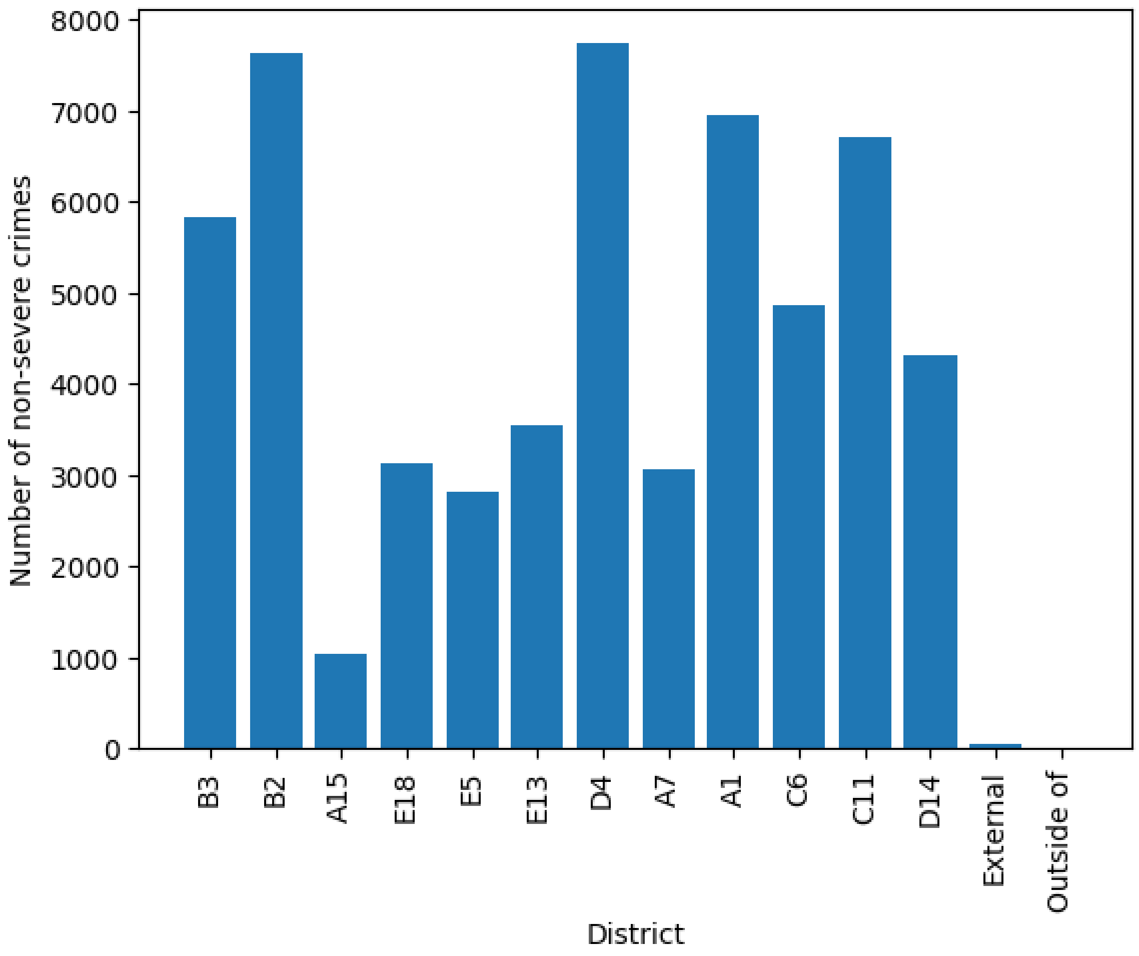
```
for district in districts:
    severe_crimes.append(train_data[train_data["DISTRICT"] == district]["Severe_cri

plt.bar(districts, severe_crimes)
plt.xlabel("District")
plt.ylabel("Number of severe crimes")
plt.xticks(rotation=90)
# save the plot
plt.savefig("../reports/figures/severe_crimes_by_district.png")

plt.show()
```

```
B3
Severe_crimes
0    5821
1     463
Name: count, dtype: int64
B2
Severe_crimes
0    7639
1     638
Name: count, dtype: int64
A15
Severe_crimes
0    1040
1      49
Name: count, dtype: int64
E18
Severe_crimes
0    3119
1     189
Name: count, dtype: int64
E5
Severe_crimes
0    2822
1     115
Name: count, dtype: int64
E13
Severe_crimes
0    3556
1     206
Name: count, dtype: int64
D4
Severe_crimes
0    7735
1     595
Name: count, dtype: int64
A7
Severe_crimes
0    3073
1     170
Name: count, dtype: int64
A1
Severe_crimes
0    6957
1     644
Name: count, dtype: int64
C6
Severe_crimes
0    4869
1     387
Name: count, dtype: int64
C11
Severe_crimes
0    6720
1     493
Name: count, dtype: int64
D14
Severe_crimes
0    4307
1     207
Name: count, dtype: int64
External
Severe_crimes
0    57
1     4
```

```
Name: count, dtype: int64
Outside of
Severe_crimes
1    1
0    1
Name: count, dtype: int64
```

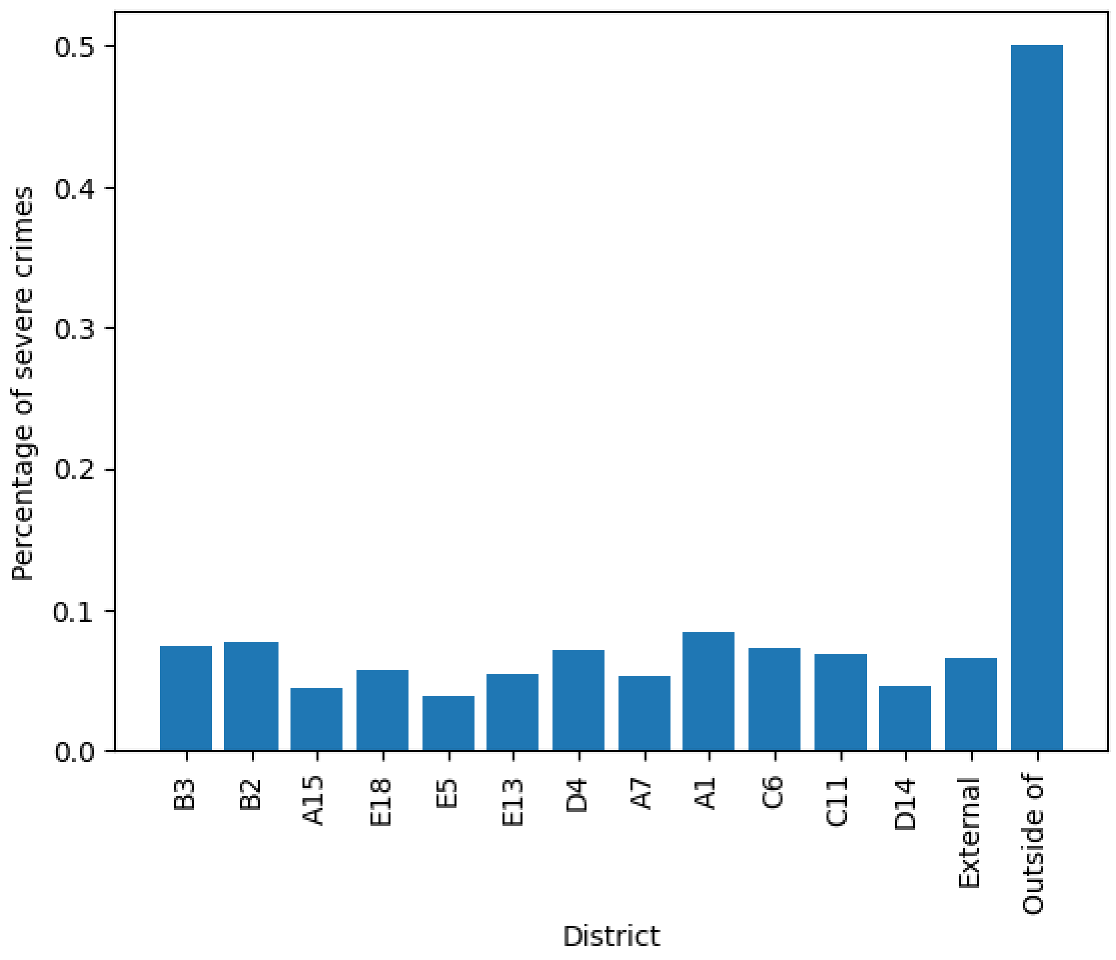


```
In [ ]: # plot the number of non-severe crimes by district
        non_severe_crimes = []
        for district in districts:
            non_severe_crimes.append(train_data[train_data["DISTRICT"] == district]["Severe

        plt.bar(districts, non_severe_crimes)
        plt.xlabel("District")
        plt.ylabel("Number of non-severe crimes")
        plt.xticks(rotation=90)

        # save the plot
        plt.savefig("../reports/figures/non_severe_crimes_by_district.png")
        plt.show()
```



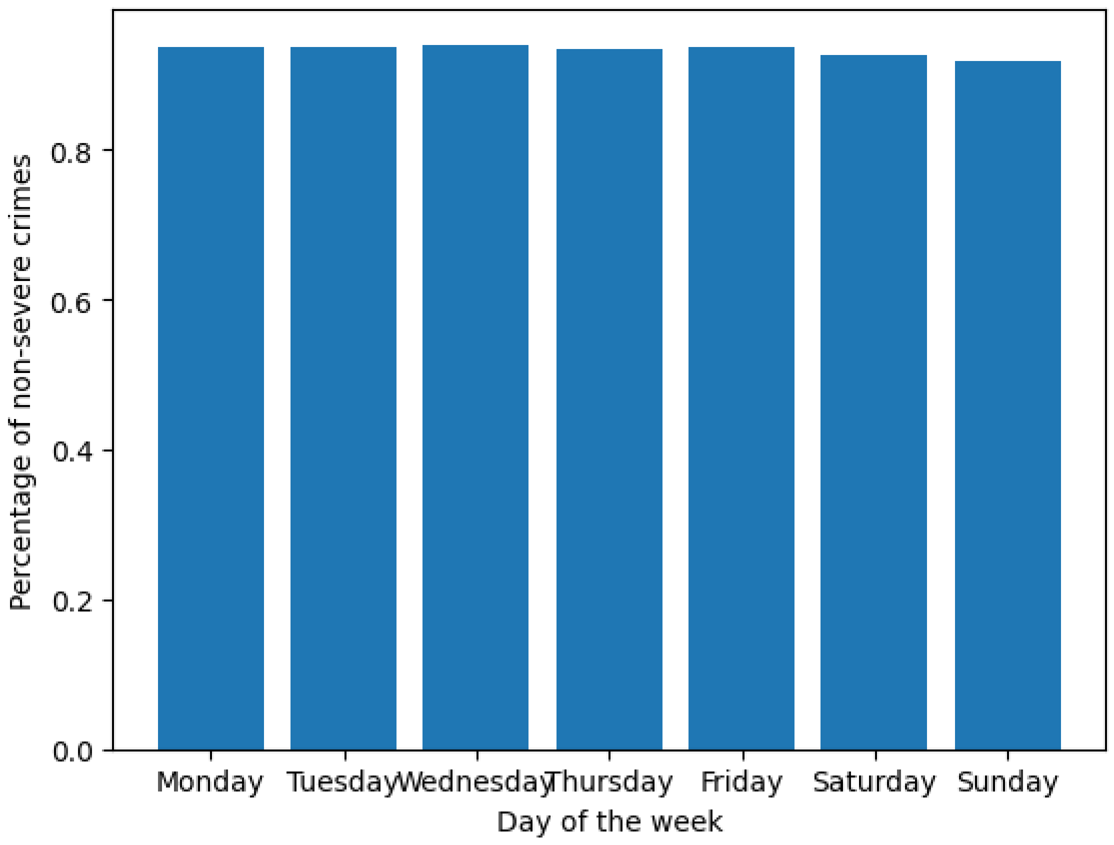
In [ ]:

```
# do a percentage of severe crimes by district
percentage_severe_crimes = []
for i in range(len(districts)):
    percentage_severe_crimes.append(severe_crimes[i] / (severe_crimes[i] + non_seve

plt.bar(districts, percentage_severe_crimes)
plt.xlabel("District")
plt.ylabel("Percentage of severe crimes")
plt.xticks(rotation=90)
# save the plot
plt.savefig("../reports/figures/percentage_severe_crimes_by_district.png")

plt.show()
```



In [ ]:

```
# do a percentage of non-severe crimes by day of the week
percentage_non_severe_crimes = []

for day in ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Su
    percentage_non_severe_crimes.append(train_data[train_data["DAY_OF_WEEK"] == day

plt.bar(["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunda
plt.xlabel("Day of the week")
plt.ylabel("Percentage of non-severe crimes")
# save the plot
plt.savefig("../reports/figures/percentage_non_severe_crimes_by_day_of_week.png")

plt.show()
```



Percentage of non-severe crimes
0.0
0.2
0.4
0.6
0.8
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of the week