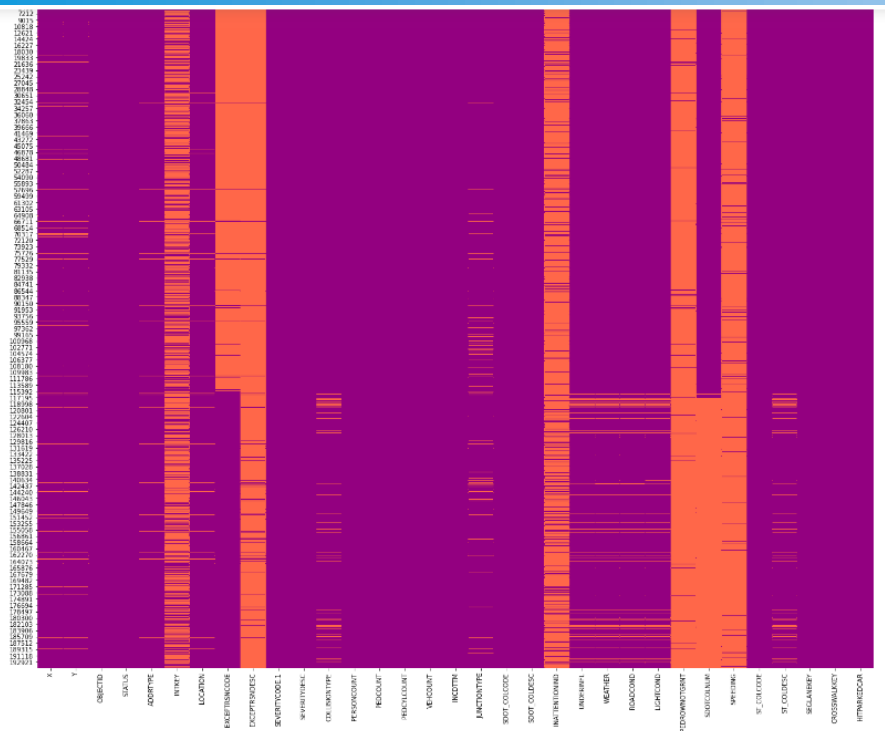
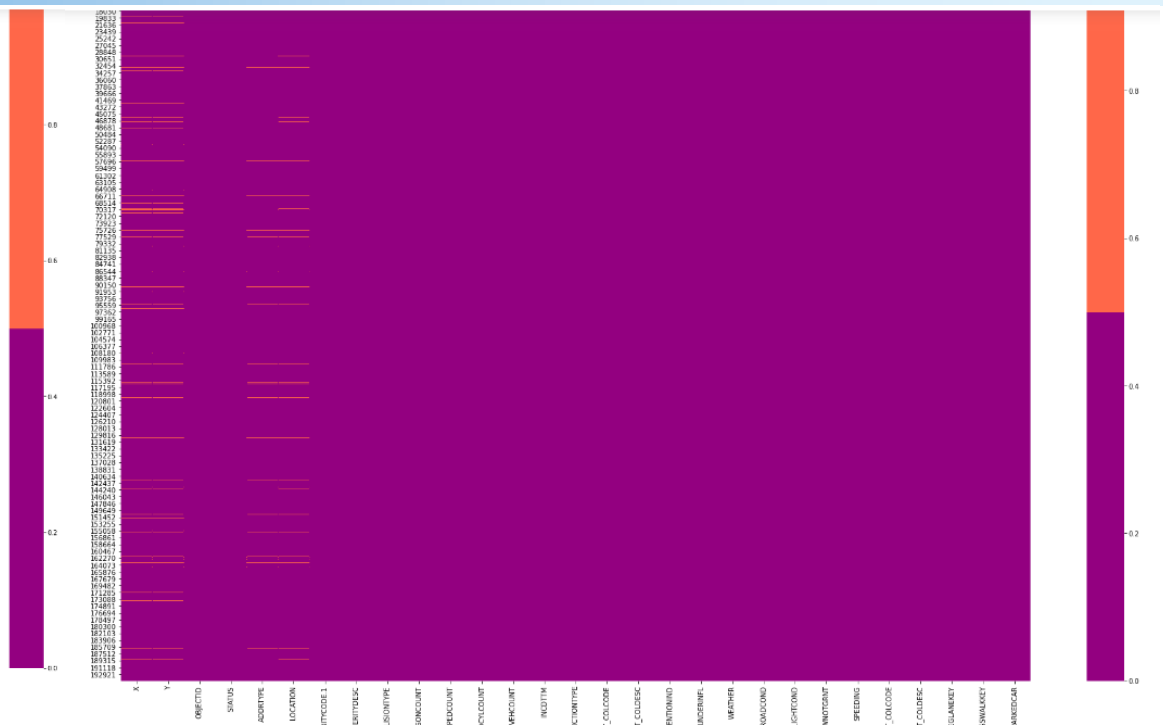
Collisions Data Analysis

Data Information

OBJECTID: ObjectID
SRI: unique identifier
SHAPE: Geometry
ESRI: geometry field
INCKEY: Long, A unique key for the incident
COLDETKEY: Long, Secondary key for the incident
ADDRTYPE: Text, 12 Collision address type:
• Alley
• Block
• Intersection
INTKEY Double, Key that corresponds to the intersection associated with a collision
LOCATION: Text, 255 Description of the general location of the collision
EXCEPTRSNCODE: Text, 10
EXCEPTRSNDESC: Text, 300
SEVERITYCODE: Text, 100 A code that corresponds to the severity of the collision:
• 3—fatality
• 2b—serious injury
• 2—injury
• 1—prop damage
• 0—unknown
SEVERITYDESC: Text A detailed description of the severity of the collision
COLLISIONTYPE: Text, 300 Collision type
PERSONCOUNT: Double, The total number of people involved in the collision
PEDCOUNT: Double, The number of pedestrians involved in the collision. This is entered by the state.
PEDCYLCOUNT: Double, The number of bicycles involved in the collision.This is entered by the state.
VEHCOUNT: Double, The number of vehicles involved in the collision.This is entered by the state.
INJURIES: Double, The number of total injuries in the collision. This is entered by the state.
SERIOUSINJURIES: Double, The number of serious injuries in the collision.This is entered by the state.
FATALITIES: Double, The number of fatalities in the collision. This is entered by the state.
INCDATE: Date, The date of the incident.
INCDTTM: Text, 30 The date and time of the incident.
JUNCTIONTYPE: Text, 300 Category of junction at which collision took place
SDOT_COLCODE: Text, 10 A code given to the collision by SDOT.
SDOT_COLDESC: Text, 300 A description of the collision corresponding to the collision code.
INATTENTIONIND: Text, 1 Whether or not collision was due to inattention.(Y/N)
UNDERINFL: Text, 10 Whether or not a driver involved was under the influence of drugs or alcohol.
WEATHER: Text, 300 A description of the weather conditions during the time of the collision.
ROADCOND: Text, 300 The condition of the road during the collision.
LIGHTCOND: Text, 300 The light conditions during the collision.
PEDROWNOTGRNT: Text, 1 Whether or not the pedestrian right of way was not granted. (Y/N)
SDOTCOLNUM: Text, 10 A number given to the collision by SDOT.
SPEEDING: Text, 1 Whether or not speeding was a factor in the collision. (Y/N)
ST_COLCODE: Text, 10 A code provided by the state that describes the collision. For more information about these codes, please see the State Collision Code Dictionary.
ST_COLDESC: Text, 300 A description that corresponds to the state's coding designation.
SEGLANEKEY: Long, A key for the lane segment in which the collision occurred.
CROSSWALKKEY: Long, A key for the crosswalk at which the collision occurred.
HITPARKEDCAR: Text, 1 Whether or not the collision involved hitting a parked car. (Y/N)



Before



After

Filled missing values with correct values for that independent variables. Methods for filling be varies to variables from variables. ex. “LIGHTCOND” and “WEATHER” is has different parameters.

```
In [34]: df["WEATHER"].replace("Clear", "good", inplace=True)
         df["WEATHER"].replace("Raining", "unbalanced", inplace=True)
         df["WEATHER"].replace("Overcast", "unbalanced", inplace=True)
         df["WEATHER"].replace("Other", "unknown", inplace=True)
         df["WEATHER"].replace("Unknown", "unknown", inplace=True)
         df["WEATHER"].replace("Snowing", "bad", inplace=True)
         df["WEATHER"].replace("Fog/Smog/Smoke", "bad", inplace=True)
         df["WEATHER"].replace("Sleet/Hail/Freezing Rain", "bad", inplace=True)
         df["WEATHER"].replace("Blowing Sand/Dirt", "bad", inplace=True)
         df["WEATHER"].replace("Severe Crosswind", "bad", inplace=True)
         df["WEATHER"].replace("Partly Cloudy", "good", inplace=True)
```

There is weather parameter to fill missing values. 4 condition that have. Good condition, bad condition, unbalanced condition and unknown. I did that because I will use that variable to machine learning as dummy variables. Purpose of that is decrease the categoric variable's number. Same as here I use that to different variables ex. "LIGHTCOND", "ROADCOND" etc.

In [61]:
```
# i looked here that may i fill nan values in ADDRTYPE with COLLISIONTYPE that
# obviously seen high rate diffrentiate.

pd.crosstab(df["ADDRTYPE"], df["COLLISIONTYPE"], normalize="columns")
```

Out[61]:

COLLISIONTYPE / ADDRTYPE	Angles	Cycles	Head On	Left Turn	Other	Parked Car	Pedestrian	Rear Ended	Right Turn	Sideswipe
Alley	0.001647	0.001478	0.002478	0.000000	0.010309	0.006937	0.005603	0.000325	0.000000	0.000866
Block	0.163719	0.425799	0.779485	0.155027	0.803220	0.965313	0.281950	0.875731	0.418367	0.788058
Intersection	0.834634	0.572722	0.218038	0.844973	0.186472	0.027750	0.712447	0.123944	0.581633	0.211076

I looked here that which collision type's associate with which addrtype. Addrtype means collision address type. And we can see here some high rate for example Intersection-Angles about at the rate of 84%. ıf examine the table for cautionsly you can see some high rate differentiate.

In [66]:
```
dms = pd.get_dummies(X[["STATUS", "ADDRTYPE", "SEVERITYCODE.1", "COLLISIONTYPE", "JUNCTIONTYPE", "SDOT_COLCODE",
                        "WEATHER", "ROADCOND", "LIGHTCOND", "UNDERINFL"]])
X = X.drop(["STATUS", "ADDRTYPE", "SEVERITYCODE.1", "COLLISIONTYPE", "JUNCTIONTYPE", "SDOT_COLCODE",
                        "WEATHER", "ROADCOND", "LIGHTCOND", "UNDERINFL", "ST_COLCODE", "SEGLANEKEY", "CROSSWALKKEY"],
           axis=1)
```

In [67]:
```
X = pd.concat([X, dms], axis=1)
y = X["SEVERITYCODE.1"]
X = X.drop("SEVERITYCODE.1", axis=1)
```

In [68]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

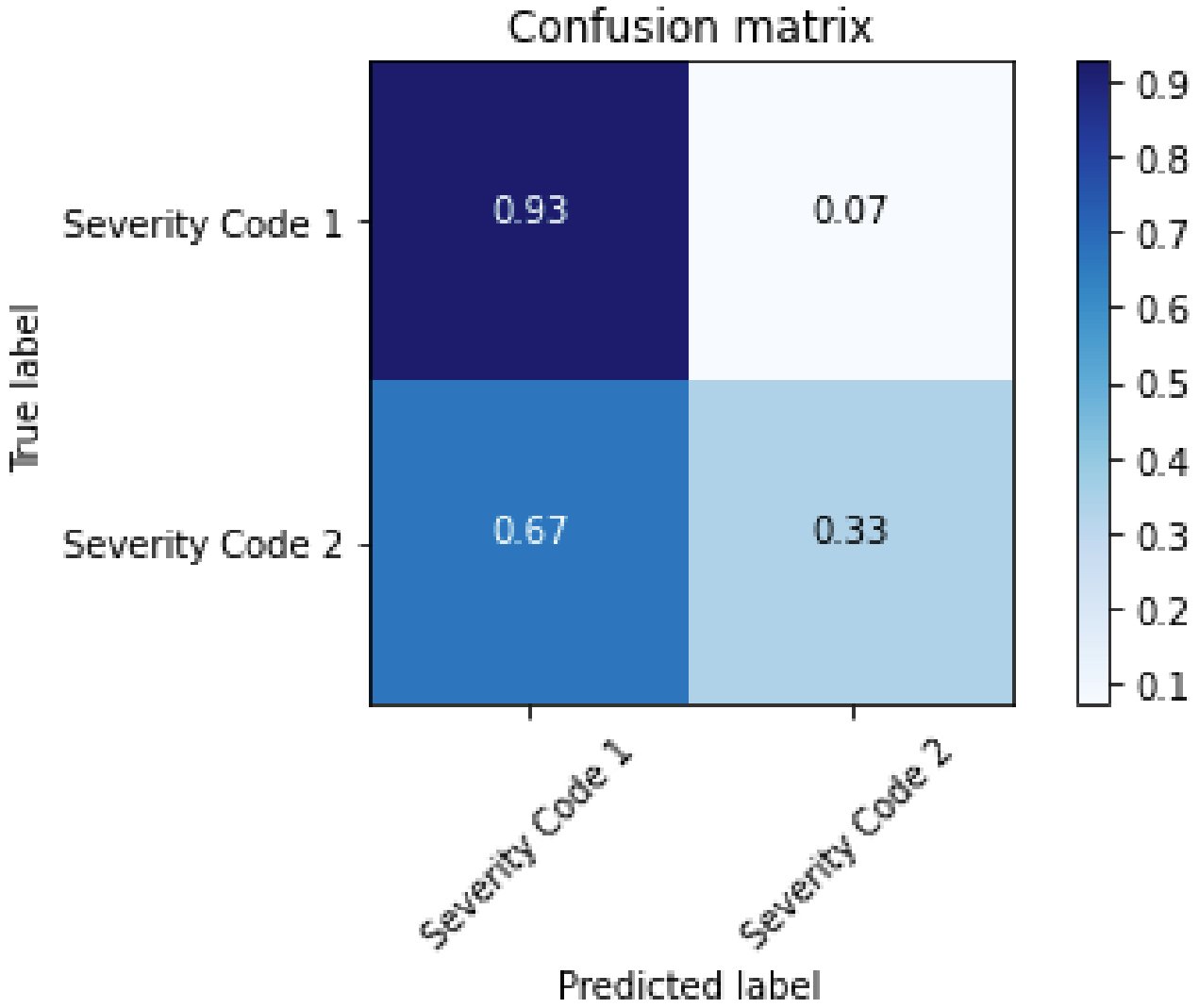
Train and test sets. Dummy variables for modeling and classification reports for decisiton tree.

In [72]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           1       0.77      0.93      0.84     41165
           2       0.66      0.33      0.44     17237

    accuracy                           0.75     58402
   macro avg       0.71      0.63      0.64     58402
weighted avg       0.74      0.75      0.72     58402
```


Confusion matrix
Severity Code 1
Severity Code 2
True label
0.93
0.07
0.67
0.33
Severity Code 1
Severity Code 2
Predicted label
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1