

PREDICTING THE SEVERITY
OF COLLISION IN SEATIFLE
CITY

# **INTRODUCTION:**

- ▶ 1.Car collision problem harass Seattle City
- > 2.Government and City police awareness
- > 3.Citizen alert.

# DATA ACQUISITION AND CLEANING:

```
In [7]: df_raw['ROADCOND'].value counts()
In [6]: df_raw['WEATHER'].value_counts()
                                                                 Out[7]: Dry
                                                                                          124510
   Out[6]: Clear
                                          111135
                                                                                           47474
            Raining
                                           33145
            Overcast
                                           27714
                                                                         Unknown
                                                                                           15078
            Unknown
                                           15091
                                                                         Ice
                                                                                            1209
            Snowing
                                             907
                                                                         Snow/Slush
                                                                                            1004
            Other
                                             832
                                                                         Other
                                                                                             132
            Fog/Smog/Smoke
                                             569
            Sleet/Hail/Freezing Rain
                                             113
                                                                         Standing Water
                                                                                             115
            Blowing Sand/Dirt
                                              56
                                                                         Sand/Mud/Dirt
                                                                                              75
            Severe Crosswind
                                              25
                                                                         0il
            Partly Cloudy
                                                                         Name: ROADCOND, dtype: int64
            Name: WEATHER, dtype: int64
In [8]: df raw['LIGHTCOND'].value counts()
                                                              In [9]: df_raw['SPEEDING'].value_counts()
  Out[8]: Daylight
                                         116137
            Dark - Street Lights On
                                           48507
            Unknown
                                           13473
            Dusk
                                            5902
                                            2502
            Dawn
            Dark - No Street Lights
                                            1537
                                                                        Name: SPEEDING, dtype: int64
            Dark - Street Lights Off
                                            1199
            Other
                                             235
            Dark - Unknown Lighting
                                              11
            Name: LIGHTCOND, dtype: int64
```

Name: SEVERITYCODE, dtype: int64

- Our target severity is imbalance, with 132,285 severity 1 and only 57,052 severity 2.
- Severity 1 size almost three times than severity 2. So we decide use de-sampling method down size severity 1 to match severity 2.

### **EXPLORATORY DATA ANALYSIS**

### PREDICTIVE MODELING

#### **DECISION TREE**

```
#Decison Tree Test
yhat tree = CollisionTree.predict(X test)
from sklearn.metrics import f1_score
f1_score_tree = f1_score(y_test, yhat_tree, average='weighted')
print(f1 score tree)
from sklearn.metrics import jaccard similarity score
jaccard_similarity_score_tree = jaccard_similarity_score(y_test, yhat_tree)
print(jaccard_similarity_score_tree)
   0.537266541918114
   0.5619385653564699
```

### PREDICTIVE MODELING

#### LOGISTIC REGRESSION

```
#logistic regression Test
from sklearn.metrics import f1_score
f1_score_LR = f1_score(y_test, yhat_log, average='weighted')
print(f1_score_LR)

from sklearn.metrics import jaccard_similarity_score
jaccard_similarity_score_LR = jaccard_similarity_score(y_test, yhat_log)
print(jaccard_similarity_score_LR)

from sklearn.metrics import log_loss
log_loss_LR = log_loss(y_test, yhat_prob)
print(log_loss_LR)

0.5314368139211159
0.5611059988606985
0.6654749127580274
```

## CONCLUSIONS

- ► Weather, road condition, lighting condition and severity connection
- ▶ Citizen driving alert
- ▶ Government and police improvement