Capstone Project:
Traffic Crashes in Chicago

Group 5:
Min-Chi Tsai
Fatima Nurmakhamadova

Prof. Daya Rudhramoorthi ALY 6140 - 80784
Analytics Systems Technology
College of Professional Study

May 17, 2022

AGENDA

1. Project Introduction
2. Exploratory Data Analysis (EDA)
3. Modeling
4. Conclusion & Recommendations

Introduction

The goal of the project is to help Chicago Police Department to

- predict the traffic crash type, and
- understand the causes that lead to it.

The data set is about daily traffic crash reports in Chicago city from 2015 to the present provided by Chicago Police Department.

Questions we want to solve

1. **What factors affect the severity of the traffic crash?**
2. What are the common types of traffic crashes and their causes?
3. How environmental factors affect crashes?
4. What time of the day has the most traffic crashes?

EDA: Data Cleaning

Checking the wrong data types

```
#Show the variables type
crash22.dtypes
```

```
CRASH_DATE                  datetime64[ns]
POSTED_SPEED_LIMIT                   int64
TRAFFIC_CONTROL_DEVICE              object
DEVICE_CONDITION                    object
WEATHER_CONDITION                   object
LIGHTING_CONDITION                  object
FIRST_CRASH_TYPE                    object
TRAFFICWAY_TYPE                     object
LANE_CNT                           float64
ROADWAY_SURFACE_COND                object
ROAD_DEFECT                         object
REPORT_TYPE                         object
CRASH_TYPE                          object
INTERSECTION_RELATED_I              object
NOT_RIGHT_OF_WAY_I                  object
HIT_AND_RUN_I                       object
DAMAGE                              object
PRIM_CONTRIBUTORY_CAUSE             object
SEC_CONTRIBUTORY_CAUSE              object
WORK_ZONE_TYPE                      object
NUM_UNITS                            int64
MOST_SEVERE_INJURY                  object
INJURIES_TOTAL                     float64
CRASH_HOUR                           int64
CRASH_DAY_OF_WEEK                    int64
CRASH_MONTH                          int64
dtype: object
```

```
#Show the unique values in each column
crash22.nunique()
```

```
CRASH_DATE                  20207
POSTED_SPEED_LIMIT             20
TRAFFIC_CONTROL_DEVICE         17
DEVICE_CONDITION                8
WEATHER_CONDITION              12
LIGHTING_CONDITION              6
FIRST_CRASH_TYPE               17
TRAFFICWAY_TYPE                20
LANE_CNT                        2
ROADWAY_SURFACE_COND            7
ROAD_DEFECT                     7
REPORT_TYPE                     2
CRASH_TYPE                      2
INTERSECTION_RELATED_I          2
NOT_RIGHT_OF_WAY_I              2
HIT_AND_RUN_I                   2
DAMAGE                          3
PRIM_CONTRIBUTORY_CAUSE        36
SEC_CONTRIBUTORY_CAUSE         38
WORK_ZONE_TYPE                  4
NUM_UNITS                       9
MOST_SEVERE_INJURY              5
INJURIES_TOTAL                  9
CRASH_HOUR                     24
CRASH_DAY_OF_WEEK               7
CRASH_MONTH                     4
dtype: int64
```

```
#for PRIM_CONTRIBUTORY_CAUSE with 36 unique values
transformed_column,new_category_list=cumulatively_categorise(crash22['PRIM_CONTRIBUTORY_CAUSE'],return_categories_list=

#Check the unique values
transformed_column.value_counts()
```

```
UNABLE TO DETERMINE                    12151
Other                                   3919
FAILING TO YIELD RIGHT-OF-WAY           3125
FOLLOWING TOO CLOSELY                   2388
NOT APPLICABLE                          1415
IMPROPER OVERTAKING/PASSING             1364
FAILING TO REDUCE SPEED TO AVOID CRASH  1232
IMPROPER BACKING                        1026
DRIVING SKILLS/KNOWLEDGE/EXPERIENCE     1024
IMPROPER LANE USAGE                      990
WEATHER                                  971
Name: PRIM_CONTRIBUTORY_CAUSE, dtype: int64
```

```
#for SEC_CONTRIBUTORY_CAUSE with 38 unique values
transformed_column2,new_category_list=cumulatively_categorise(crash22['SEC_CONTRIBUTORY_CAUSE'],return_categories_list=

#Check the unique values
transformed_column2.value_counts()
```

```
NOT APPLICABLE                         11797
UNABLE TO DETERMINE                    11200
Other                                   3783
FAILING TO REDUCE SPEED TO AVOID CRASH   967
DRIVING SKILLS/KNOWLEDGE/EXPERIENCE      952
FAILING TO YIELD RIGHT-OF-WAY            906
Name: SEC_CONTRIBUTORY_CAUSE, dtype: int64
```

The **initial** dataset consists of

- **605,121** records, and **49** features

Our **focus** on **2022**

- **29,605 records, and 26 features**
- 19 categorical, 7 numerical variables

Feature Engineering: reduce the cardinality

- from **36** to **11** in the PRIM_CONTRIBUTORY_CAUSE
- from **38** to **6** in the SEC_CONTRIBUTORY_CAUSE

Crash Type is a target variable

- 0 - no injury / drive away
- 1 - injury and / or tow due to crash

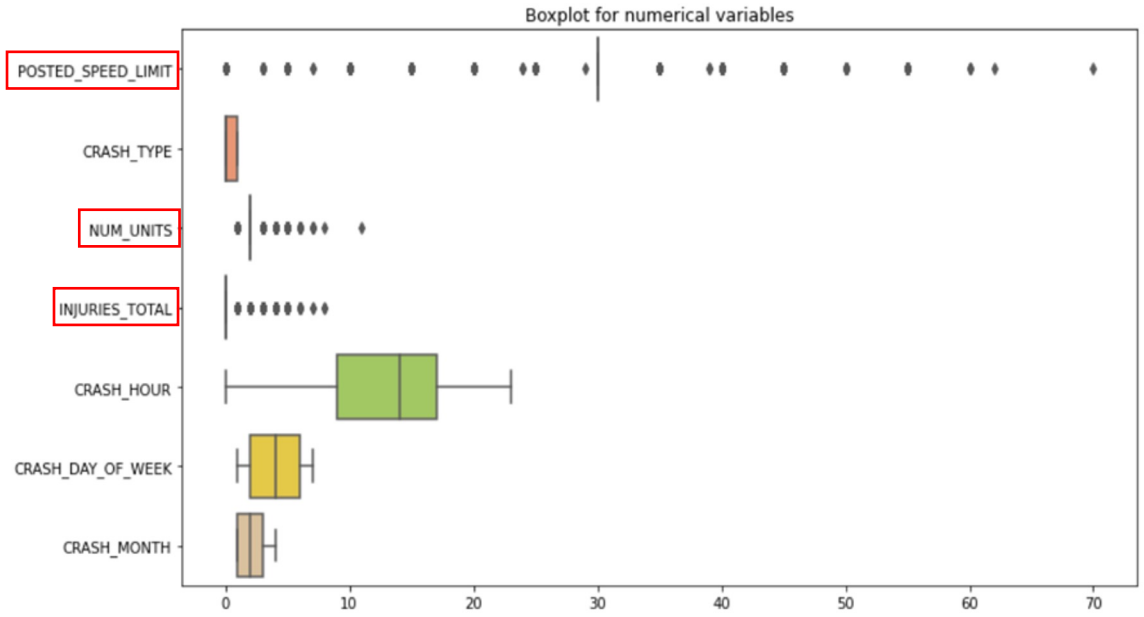
EDA: Data Cleaning

Checking for missing values

```
In [53]: # Check percentage of missing values
         percent_missing = crash22.isnull(). sum() * 100 / len(crash22)
         percent_missing
```

Out[53]:

CRASH_DATE	0.000000
POSTED_SPEED_LIMIT	0.000000
TRAFFIC_CONTROL_DEVICE	0.000000
DEVICE_CONDITION	0.000000
WEATHER_CONDITION	0.000000
LIGHTING_CONDITION	0.000000
FIRST_CRASH_TYPE	0.000000
TRAFFICWAY_TYPE	0.000000
LANE_CNT	99.989867
ROADWAY_SURFACE_COND	0.000000
ROAD_DEFECT	0.000000
REPORT_TYPE	2.979226
CRASH_TYPE	0.000000
INTERSECTION_RELATED_I	75.865563
NOT_RIGHT_OF_WAY_I	95.287958
HIT_AND_RUN_I	65.789563
DAMAGE	0.000000
PRIM_CONTRIBUTORY_CAUSE	0.000000
SEC_CONTRIBUTORY_CAUSE	0.000000
WORK_ZONE_TYPE	99.804087
NUM_UNITS	0.000000
MOST_SEVERE_INJURY	0.222935
INJURIES_TOTAL	0.000000
CRASH_HOUR	0.000000
CRASH_DAY_OF_WEEK	0.000000
CRASH_MONTH	0.000000



Missing values

- Deleting 5 columns – with more than 50%
- Drop rows – with less than 3%

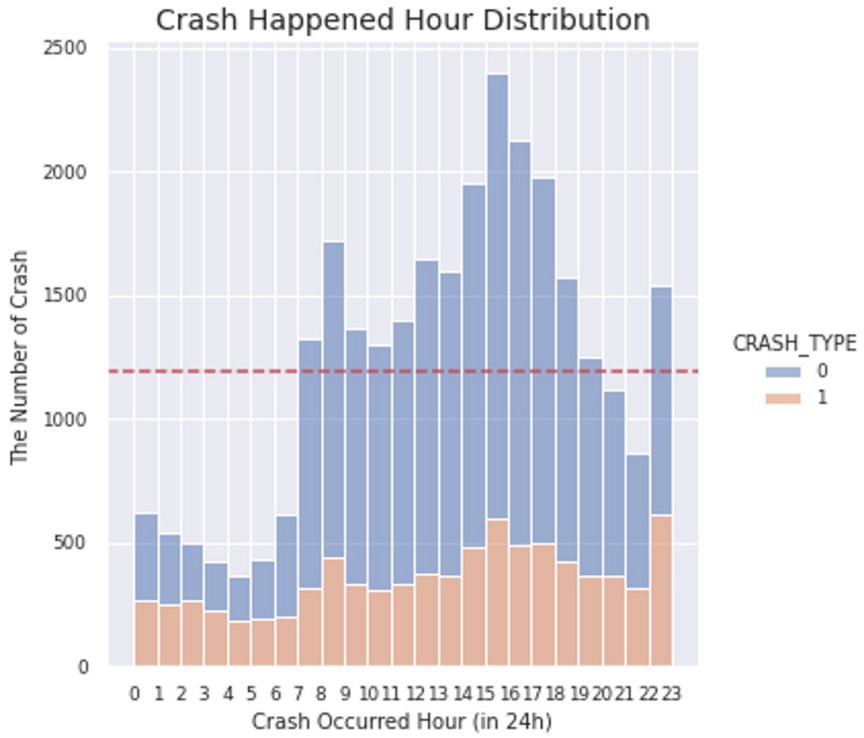
Outliers

- 3 columns with outliers
- Removed only single outlier values
- Keep the rest outliers

The dataset shape now is **28,653 rows and 21 columns**

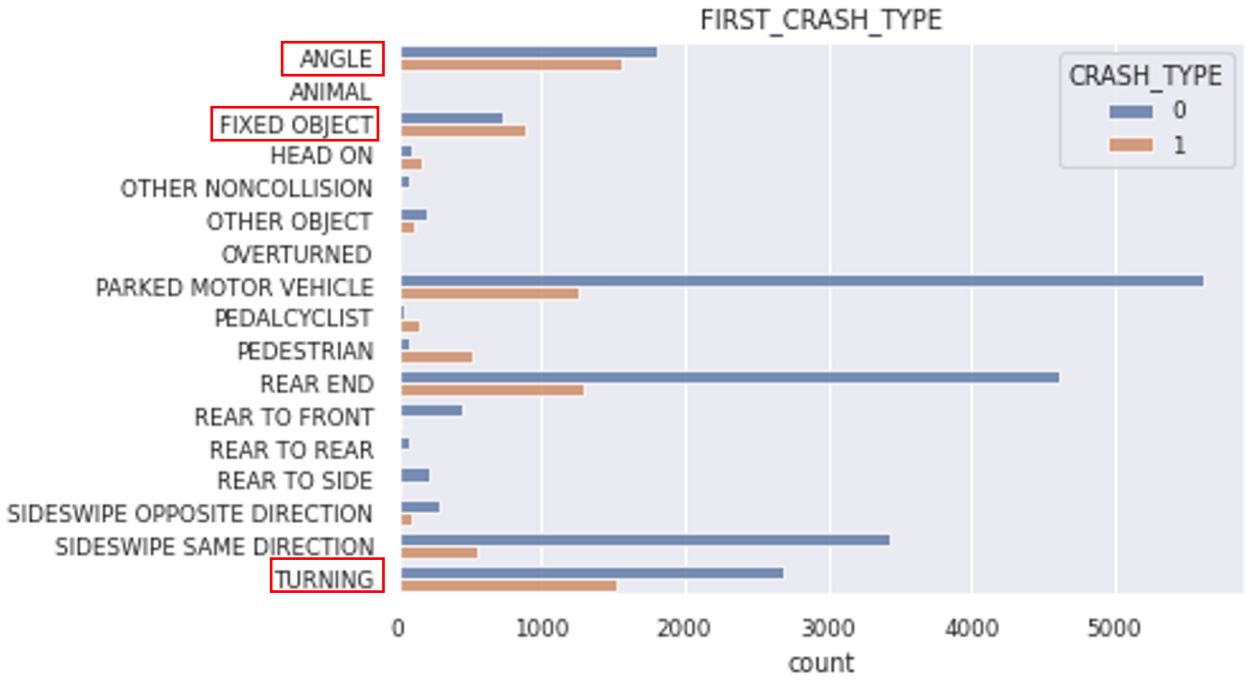
EDA: Data Visualization

Q: What time of the day has the most traffic crashes?



- Most crashes: 2 - 3 pm
- Serious crashes: 3 – 4 pm & 10 -11 pm

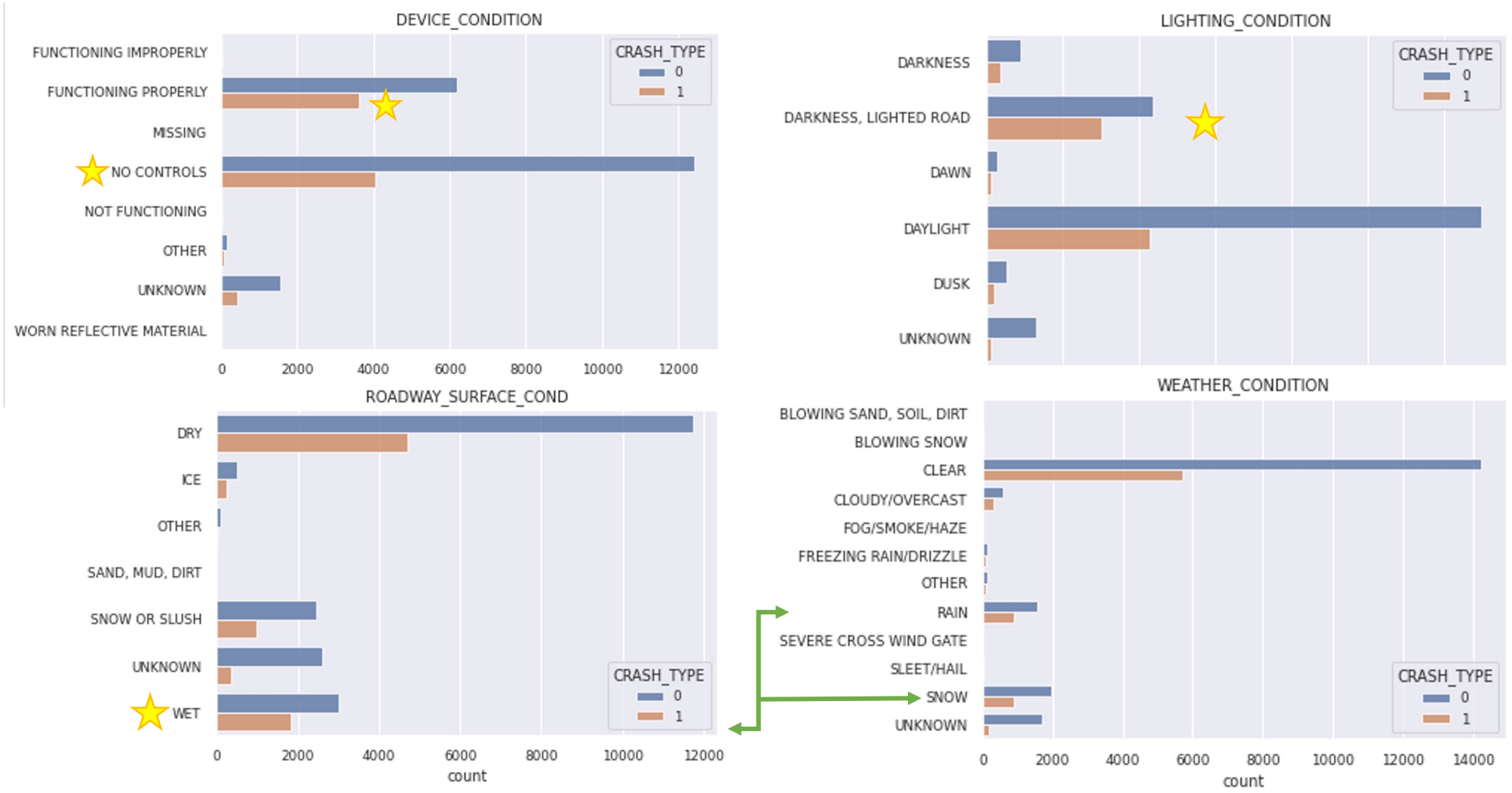
Q: What are the common types of traffic crashes and their causes?



- Most of injury crashes: Angle & Turning
- Injury > No Injury: Pedestrian, Fixed Object

EDA: Data Visualization

Q: How environmental factors affect crashes?



Modeling – Data Preparation

```
#Show most highly correlated variables
def corrFilter(x: pd.DataFrame, bound: float):
    xCorr = x.corr()
    xFiltered = xCorr[((xCorr >= bound) | (xCorr <= -bound)) & (xCorr !=1.000)]
    xFlattened = xFiltered.unstack().sort_values().drop_duplicates()
    return xFlattened

corrFilter(X, .7)
```

```
REPORT_TYPE_ON SCENE                       REPORT_TYPE_NOT ON SCENE (DESK REPORT)      -1.000000
ROAD_DEFECT_UNKNOWN                        ROAD_DEFECT_NO DEFECTS                      -0.944185
INJURIES_TOTAL                             MOST_SEVERE_INJURY_NO INDICATION OF INJURY  -0.857666
DEVICE_CONDITION_FUNCTIONING PROPERLY      DEVICE_CONDITION_NO CONTROLS                -0.839448
DAMAGE_$501 - $1,500                       DAMAGE_OVER $1,500                          -0.807635
DEVICE_CONDITION_FUNCTIONING PROPERLY      TRAFFIC_CONTROL_DEVICE_NO CONTROLS          -0.806895
LIGHTING_CONDITION_DARKNESS, LIGHTED ROAD  LIGHTING_CONDITION_DAYLIGHT                 -0.728432
MOST_SEVERE_INJURY_NO INDICATION OF INJURY MOST_SEVERE_INJURY_NONINCAPACITATING INJURY -0.723675
TRAFFIC_CONTROL_DEVICE_TRAFFIC SIGNAL      TRAFFIC_CONTROL_DEVICE_NO CONTROLS          -0.710867
                                           DEVICE_CONDITION_FUNCTIONING PROPERLY        0.736137
DEVICE_CONDITION_NO CONTROLS               TRAFFIC_CONTROL_DEVICE_NO CONTROLS           0.923540
POSTED_SPEED_LIMIT                         POSTED_SPEED_LIMIT                                NaN
dtype: float64
```

	feature	VIF
0	REPORT_TYPE_ON SCENE	71.154025
1	REPORT_TYPE_NOT ON SCENE (DESK REPORT)	76.264787
2	ROAD_DEFECT_UNKNOWN	9.336687
3	ROAD_DEFECT_NO DEFECTS	9.259547
4	INJURIES_TOTAL	3.801230
5	MOST_SEVERE_INJURY_NO INDICATION OF INJURY	4.930353
6	DEVICE_CONDITION_FUNCTIONING PROPERLY	4.515433
7	DEVICE_CONDITION_NO CONTROLS	8.660927
8	DAMAGE_OVER $1,500	2.909204
9	DAMAGE $501 - $1,500	2.930201
10	TRAFFIC_CONTROL_DEVICE_NO CONTROLS	7.260381
11	LIGHTING_CONDITION_DARKNESS, LIGHTED ROAD	2.255270
12	LIGHTING_CONDITION_DAYLIGHT	2.247147
13	MOST_SEVERE_INJURY_NONINCAPACITATING INJURY	2.102278
14	TRAFFIC_CONTROL_DEVICE_TRAFFIC SIGNAL	2.404926
15	POSTED_SPEED_LIMIT	1.051912

- Assigning features (X) and target variable (y)
- Encoding Categorical Variables (Dummy)

Checking for multicollinearity

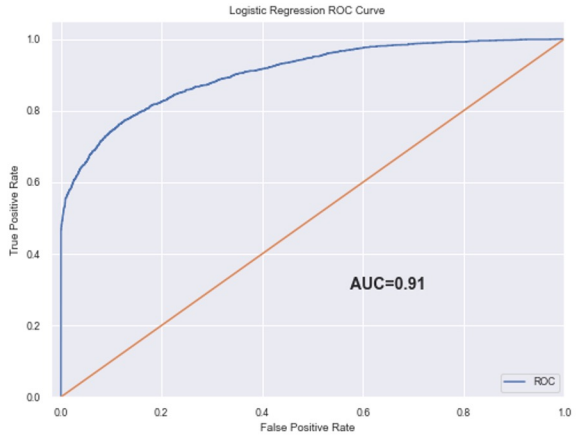
- **Correlation matrix ≥ +- 0.7**
- **VIF score ≥ 7**
- 6 columns with high correlation

Split Data

- Training and Testing (**70/30**)
- Train set - 20,057 rows, and 121 columns
- Test set has 8,596 rows and 121 columns

Modeling – Logistic Regression

```
#Building the model and fitting the data
from sklearn.linear_model import LogisticRegression

LR = LogisticRegression(penalty='l1', solver='liblinear')
model= LR.fit(X_train, y_train)
predictions = LR.predict(X_test)
```



Accuracy: 0.87

Dep. Variable:	CRASH_TYPE	No. Observations:	20057
Model:	Logit	Df Residuals:	19944
Method:	MLE	Df Model:	112
Date:	Sat, 14 May 2022	Pseudo R-squ.:	0.4848
Time:	03:00:04	Log-Likelihood:	-6188.0
converged:	False	LL-Null:	-12010.
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
POSTED_SPEED_LIMIT	0.0203	0.005	3.868	0.000	0.010	0.031
NUM_UNITS	1.1876	0.062	19.165	0.000	1.066	1.309
INJURIES_TOTAL	-2.7669	514.586	-0.005	0.996	-1011.337	1005.803
CRASH_HOUR	-0.0177	0.004	-4.574	0.000	-0.025	-0.010
CRASH_DAY_OF_WEEK	-0.0165	0.011	-1.443	0.149	-0.039	0.006
CRASH_MONTH	0.0194	0.024	0.799	0.424	-0.028	0.067
TRAFFIC_CONTROL_DEVICE_DELINEATORS	1.1095	0.715	1.551	0.121	-0.293	2.512
TRAFFIC_CONTROL_DEVICE_FLASHING CONTROL SIGNAL	0.1933	1.297	0.149	0.882	-2.350	2.736
TRAFFIC_CONTROL_DEVICE_NO PASSING	-16.3397	2.61e+04	-0.001	1.000	-5.12e+04	5.12e+04
TRAFFIC_CONTROL_DEVICE_OTHER	-0.2894	0.314	-0.922	0.357	-0.905	0.326
TRAFFIC_CONTROL_DEVICE_OTHER RAILROAD CROSSING	-9.4059	152.989	-0.061	0.951	-309.259	290.447
TRAFFIC_CONTROL_DEVICE_OTHER REG. SIGN	0.4529	0.715	0.634	0.526	-0.948	1.854
TRAFFIC_CONTROL_DEVICE_OTHER WARNING SIGN	0.9776	0.767	1.275	0.202	-0.525	2.481
TRAFFIC_CONTROL_DEVICE_PEDESTRIAN CROSSING SIGN	0.0769	0.856	0.090	0.928	-1.600	1.754
TRAFFIC_CONTROL_DEVICE_POLICE/FLAGMAN	0.9672	1.070	0.904	0.366	-1.130	3.064
TRAFFIC_CONTROL_DEVICE_RAILROAD CROSSING GATE	-0.7786	0.900	-0.865	0.387	-2.543	0.986
TRAFFIC_CONTROL_DEVICE_RR CROSSING SIGN	0.9473	1.217	0.779	0.436	-1.437	3.332
TRAFFIC_CONTROL_DEVICE_SCHOOL ZONE	1.4141	0.833	1.697	0.090	-0.219	3.047
TRAFFIC_CONTROL_DEVICE_STOP SIGN/FLASHER	-0.1866	0.137	-1.365	0.172	-0.455	0.081
TRAFFIC_CONTROL_DEVICE_TRAFFIC SIGNAL	-0.2193	0.139	-1.582	0.114	-0.491	0.052
TRAFFIC_CONTROL_DEVICE_UNKNOWN	-0.4716	0.217	-2.173	0.030	-0.897	-0.046

- **L1** regularization for optimization

ROC curve - the summary of classifier performance

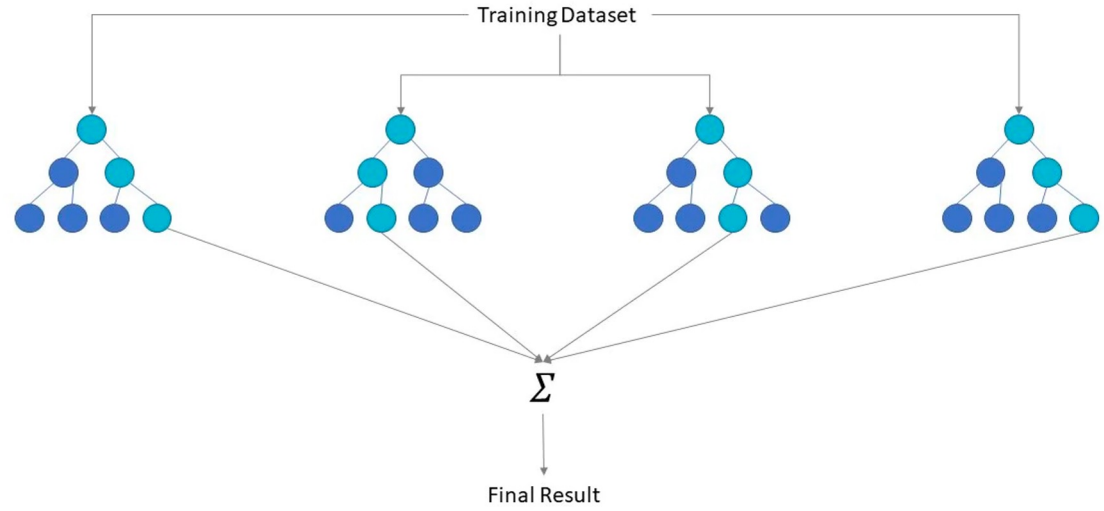
- AUC is **91%**

Summary

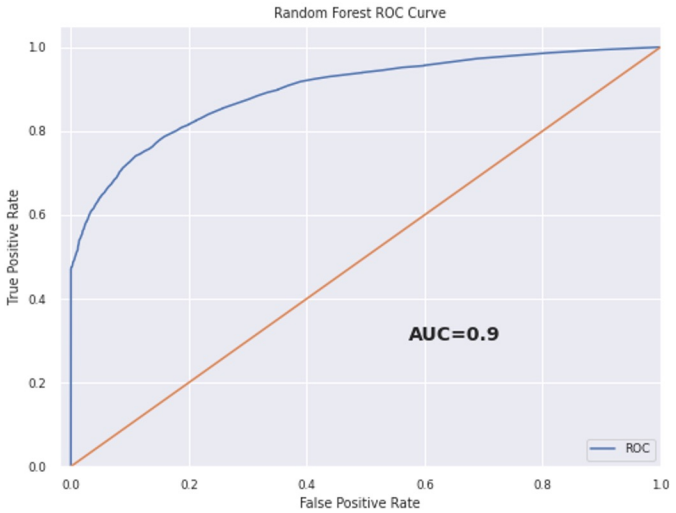
- Most **significant** variables with the p-value below the alpha level is **0.1**
- **5** significant variables with p-value **<**0.1

Modeling – Random Forest

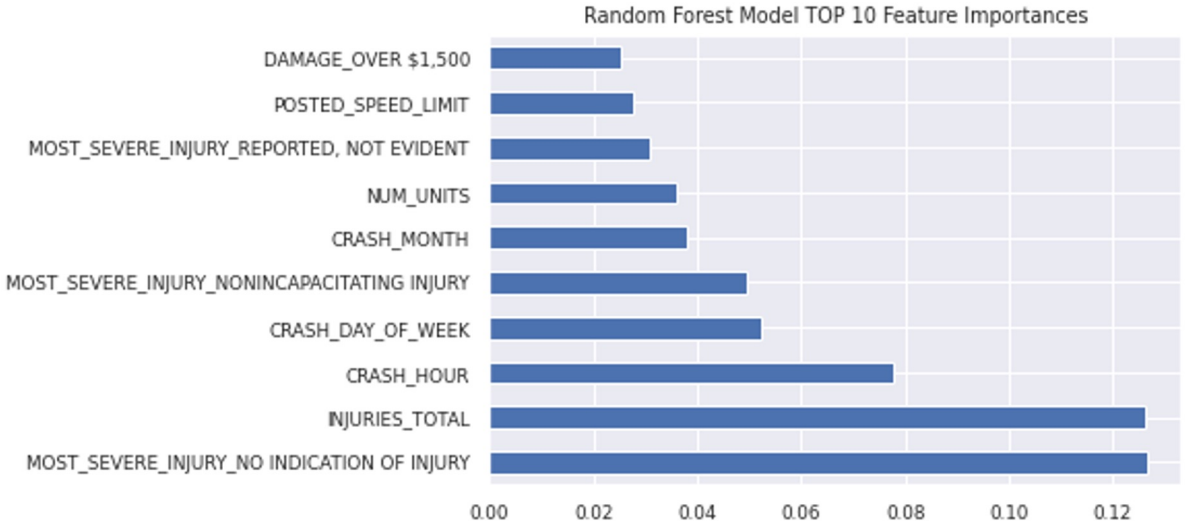
```
# import random forest model
from sklearn.ensemble import RandomForestClassifier
from numpy import mean,std
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import cross_val_score
from sklearn import metrics

# define the RamdomForest model
rfc = RandomForestClassifier().fit(X_train,y_train)

# define the model evaluation procedure
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

# evaluate the model and collect the scores
n_scores = cross_val_score(rfc, X_train, y_train, scoring='accuracy', cv=cv, n_jobs=-1)

# report the model performance
print('Mean Accuracy: %.3f (%.3f)' % (mean(n_scores), std(n_scores)))
```




```
Mean Accuracy: 0.859 (0.007)
```



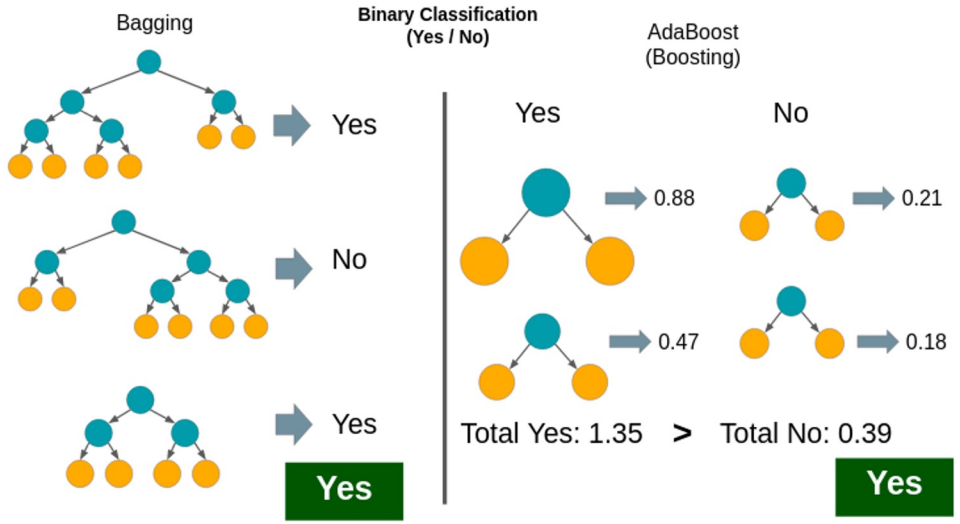
Accuracy: 0.86

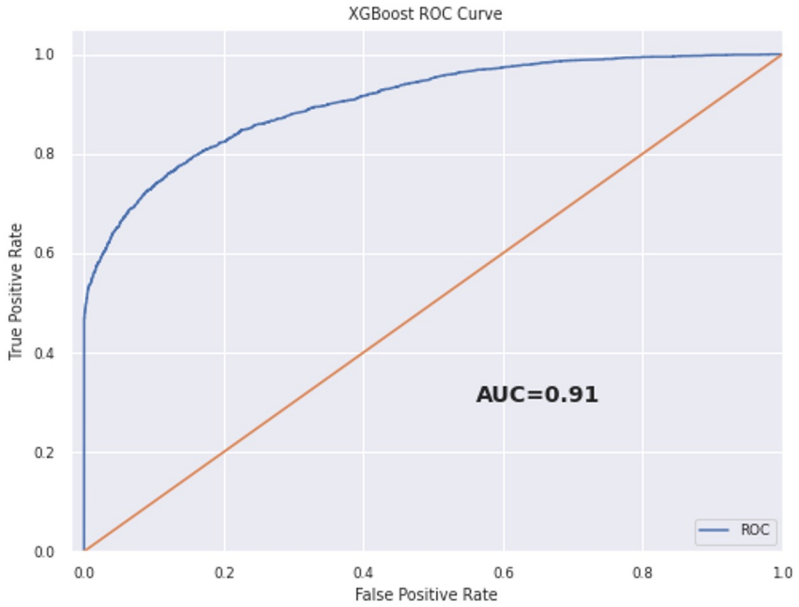


Modeling – XGBoost

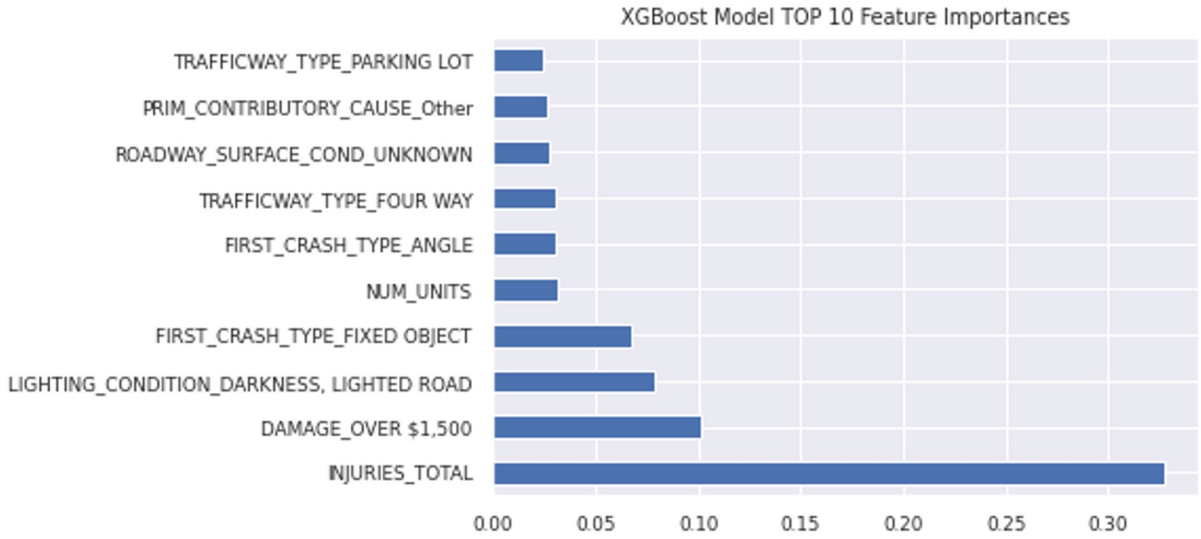
```
# build XGboost model
from xgboost import XGBClassifier
xgb = XGBClassifier(learning_rate =0.1, n_estimators=100 , random_state=42)
xgb.fit(X_train, y_train)

# define the model evaluation procedure
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

# evaluate the model and collect the scores
n_scores = cross_val_score(xgb, X_train, y_train, scoring='accuracy', cv=cv, n_jobs=-1)

# report the model performance
print('Mean Accuracy: %.3f (%.3f)' % (mean(n_scores), std(n_scores)))
```

```
Mean Accuracy: 0.864 (0.008)
```





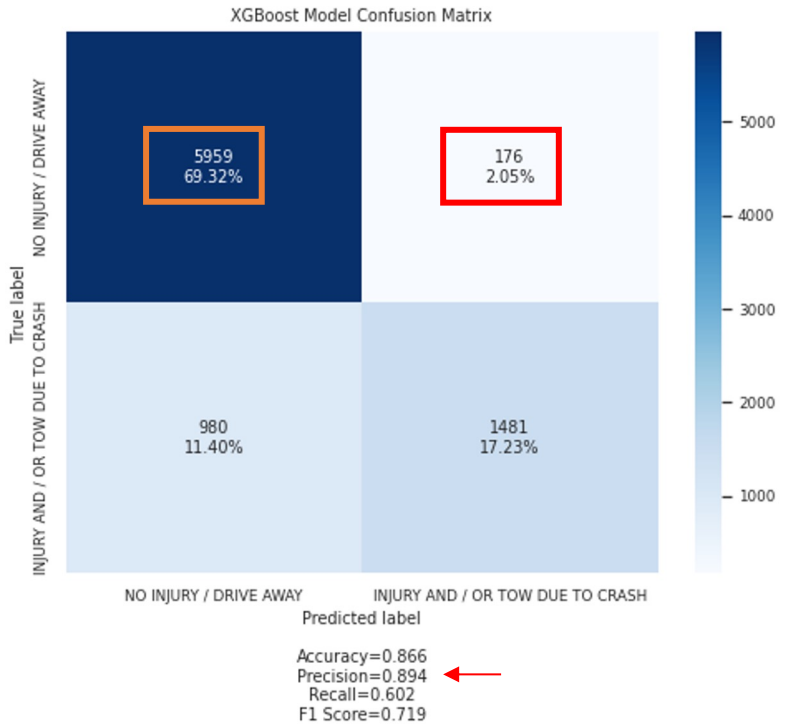
Accuracy: 0.87



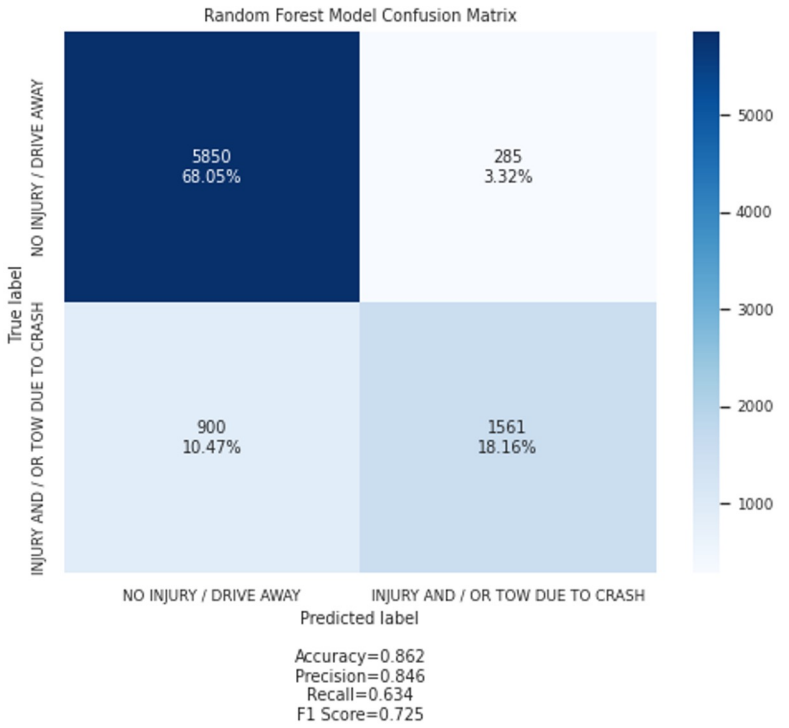
Model Comparison

FP & FN -

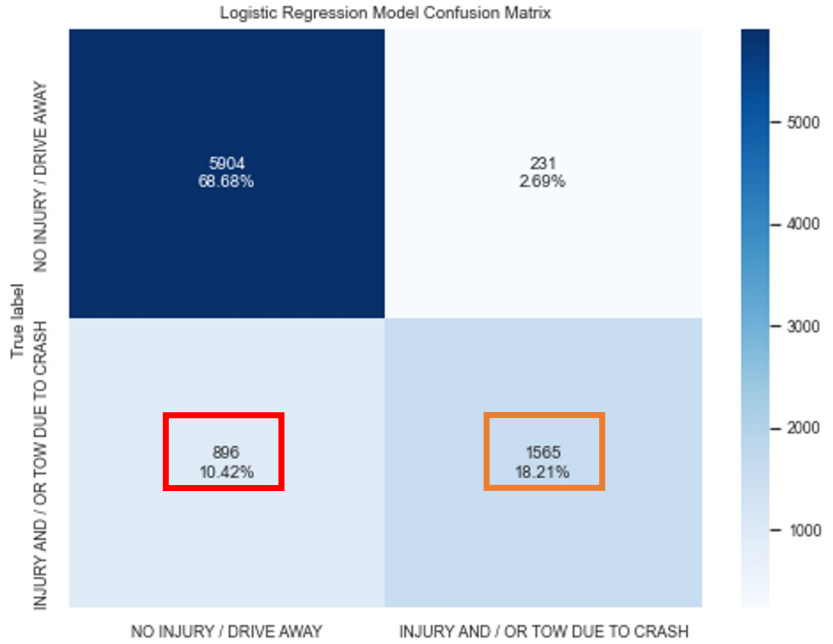
TP & TN -



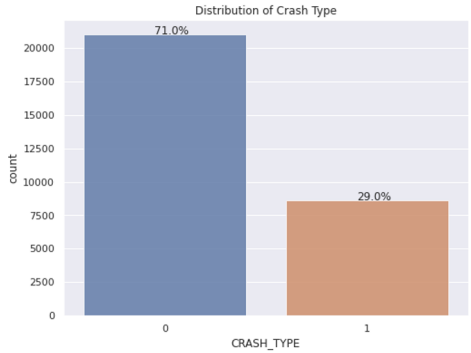
Accuracy=0.866
Precision=0.894
Recall=0.602
F1 Score=0.719



Accuracy=0.862
Precision=0.846
Recall=0.634
F1 Score=0.725



Accuracy=0.869
Precision=0.871
Recall=0.636
F1 Score=0.735



$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- **XGBoost** model - most True Negatives, and least False Positives
- **Logistic Regression** model **-** most True Positives, and least False Negatives

recall values -> imbalance between classes

Conclusion & Recommendations

Q: What factors affect the severity of the traffic crash?

- **CRASH_HOUR**

Most crashes that involves any injury or vehicle tow is caused during the high traffic peak from 7 to 9 am and 3 to 6 pm.

- **NUM_UNITS**

The probability of getting injury rises if the crash involves more than 1 unit whether it is a vehicle, pedestrian, or bicyclist.

- **POSTED_SPEED_LIMIT**

The rising posted speed limit also might affect the crash severity.

*We would recommend the city of Chicago to pay more attention and take the necessary precautions during the **peak hours** and try to **reduce** the traffic.*

Q & A
Thank You

Reference

- Chen, & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

- Levy , J. (2022, April 20). *Traffic crashes - crashes: City of chicago: Data Portal*. Chicago Data Portal. Retrieved April 21, 2022, from https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3if?category=Transportation&view_name=Traffic-Crashes-Crashes

- Sangani, R. (2021, August 11). *Dealing with features that have high cardinality*. Medium. Retrieved May 15, 2022, from https://towardsdatascience.com/dealing-with-features-that-have-high-cardinality-1c9212d7ff1b