Chicago Car Crashes

By: Yamuna Umapathy & Lotus Baumgarner

Business Problem:

Our Insurance Carrier wants us to build a model to predict the type of car crashes that lead to certain injuries (Incapacitating, Non-Incapacitating and No Injuries) in Chicago.
They want to use this information to help determine what type of injury they will be dealing with when they are presented with a new car accident.

Dataset:

Original Dataset comes from City of Chicago Data portal. It started with over 800K rows and 48 columns.

Feature Engineering

- Our Target Variable: MOST SEVERE INJURY

- Combined liked values within certain columns.
  - EX: Traffic Control Device had 19 different categorical values and was reduced down to 5 values.

- Dropped 32 of the original columns.
  - Deemed not useful (Photos Taken)
  - Missing too many values (Work Zone - missing > 800K)
  - Repetitive (Location = Longitude + Latitude)

Dataset: >800k rows, 17 columns

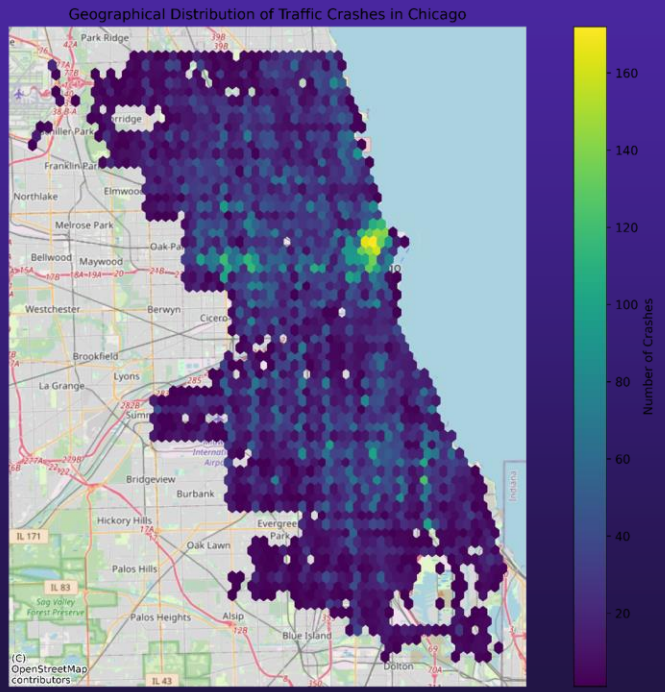
Down Sampling our Dataset

```
MOST_SEVERE_INJURY
NO_INJURY           700415
NON_INCAP_INJURY     97292
INCAP_INJURY         14676
Name: count, dtype: int64
```

1. Separated Dataframe by No Injury, Non-Incap, Incap
2. Set sample size to the length of Incap_Injury and used the sample() method to randomize the pulled rows.
3. Concatenated back into a single DataFrame.

Balanced Dataset: 44,000 rows and 17 columns

Visual: Crashes Vs Longitude, Latitude



Most crashes in a geographical area:

LAT: 41.90 & 41.95
LON: -87.65 & -87.60

Roughly the Near North Side area

Target and Columns

Most_Severe_Injury

X

16 Remaining Columns

Metric : Accuracy

After Column Transformation: X had 147 columns

Models with Logistic Regression and Random Forest

Logistic Regression:

	precision	recall	f1-score	support
INCAP_INJURY	0.59	0.58	0.59	2980
NON_INCAP_INJURY	0.56	0.63	0.59	2975
NO_INJURY	0.99	0.87	0.92	2867
accuracy			0.69	8822
macro avg	0.71	0.69	0.70	8822
weighted avg	0.71	0.69	0.70	8822

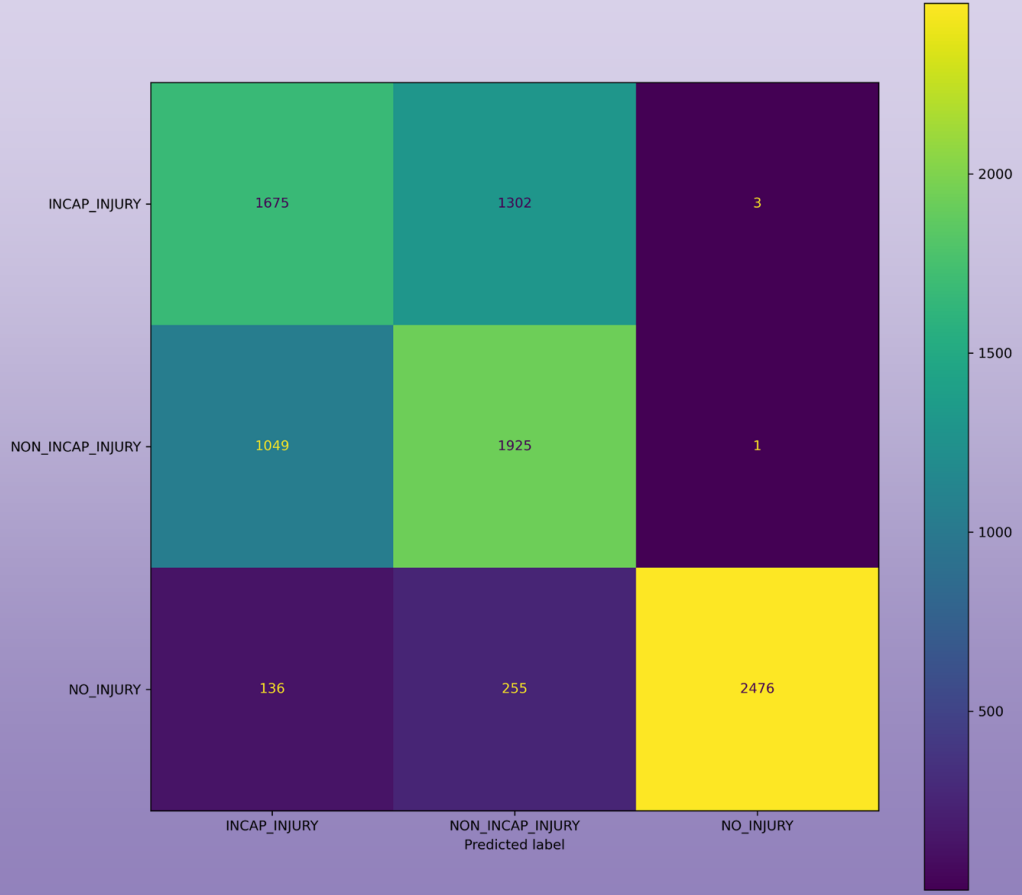
Random Forest Classifier:

	precision	recall	f1-score	support
INCAP_INJURY	0.55	0.69	0.61	2980
NON_INCAP_INJURY	0.57	0.50	0.53	2975
NO_INJURY	1.00	0.86	0.92	2867
accuracy			0.68	8822
macro avg	0.70	0.68	0.69	8822
weighted avg	0.70	0.68	0.68	8822

Random Forest had a lower accuracy score on the test than Logistic Regression.

Model Using XG Boost

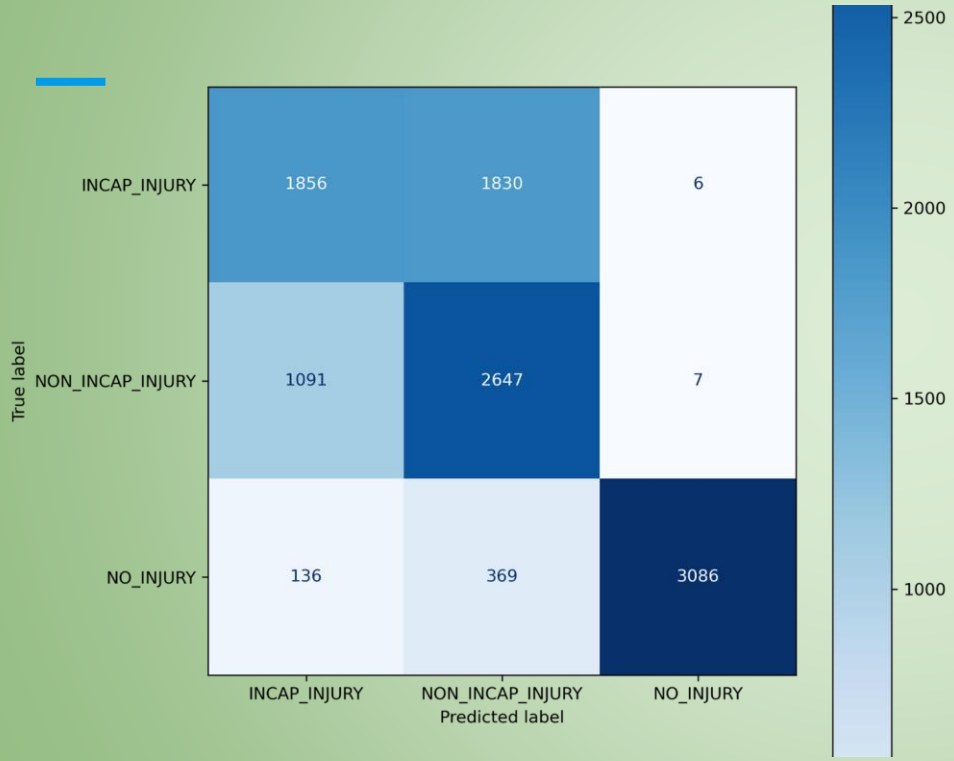
	precision	recall	f1-score	support
0	0.59	0.57	0.58	2980
1	0.55	0.65	0.60	2975
2	1.00	0.86	0.93	2867
accuracy			0.69	8822
macro avg	0.71	0.69	0.70	8822
weighted avg	0.71	0.69	0.70	8822

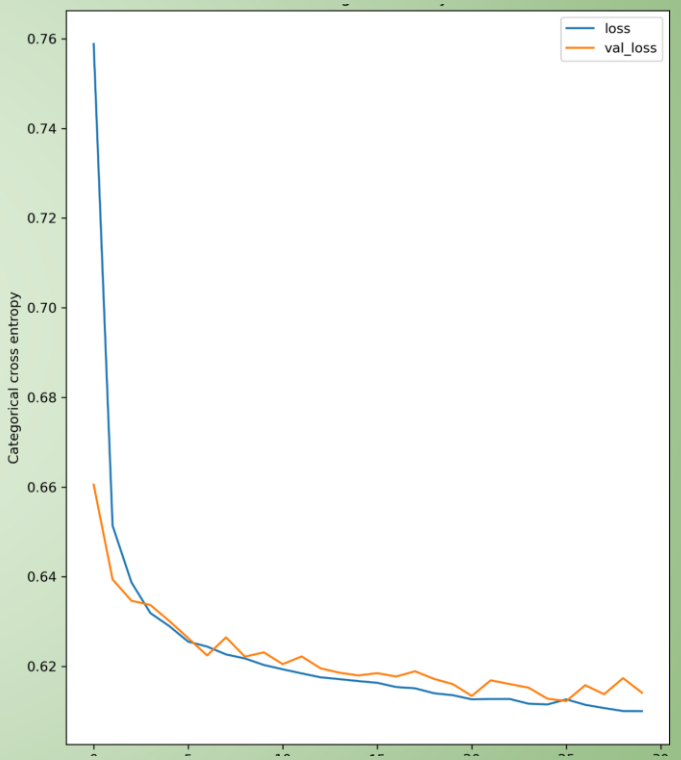


GridSearch with Hypertuned Parameters

MODEL	ACCURACY SCORE ON TEST
1. LOGISTIC REGRESSION	0.6903355850218983
2. RANDOM FOREST	0.6846572310961132
3. XGBOOST	0.6914144529630037

Tensorflow results





Conclusions:

Crashes and causes that tend to lead to Injuries:

- RearEnd
- Pedestrian/cyclist
- Head-on collision
- Driver's physical condition
- Weather
- Late Night

Next Steps:

Ours:

- Work more on our Hyperparameter Tuning and Feature Importance to achieve a higher accuracy score.

Insurance Carrier:

- Offer Drivers Insurance benefits such as lower premiums or deductibles for safer driving.
- Focus on drivers who live, work or visit the Near North Side area since they have the most crashes.

Questions ?

Yamuna Umapathy
u.yamuna@gmail.com
https://www.linkedin.com/in/yamuna-umapathy/

Lotus Baumgarner
LotusBaumgarner@gmail.com
https://www.linkedin.com/in/lotus-baumgarner/