

Milestone 4

Dataset: Crashes in Chicago (2015-2023)

Team 1:

Indira Aitkulova









Antonio Azevedo

Alex Chae

Ashley Nguyen



Table of Contents

		Motivation & Literature Review
		Data Description
		Visualizations
		Machine Learning Models



Motivation & Literature

Motivation & Literature

Stereotypes against female drivers:

- American family life = women at home, as wives and mothers
- Maintaining a car = masculinity
- Women drivers portrayed as dangerous in folklores, in order to “keep women in their place”
→ Negative stereotypes against women drivers (Berger, 1986)

But, there are evidence that this is **changing** :

- 51% women said women drive safer & 39% of men said men drive safer (Edgersen, 2011)
- Youngest males are most likely to die with speed-driving (National Highway Traffic Safety Administration)
- Men are 3.1 times more likely to be drunk driving (Edgersen, 2011)

→ So what is the relationship between driver's sex & car crashes?

Motivation & Literature

Conflicting results in literature...

Women drivers drive more dangerously:

- Older women (70+) are overrepresented in car crashes that occur in the “safest” conditions (Baker et al., 2003)
- Older women tend to cause more crashes when driving long distances (Chipman, 1993)
- Women had higher involvement in all police-reported crashes (Massie, 1995)
- Women generally have lower annual mileages, which causes more car crashes (Massie, 1997)

Motivation & Literature

Conflicting results in literature...

No evidence that women drivers are worse:

- Drunk driving & Alcohol abuse in driving is still primarily a male problem in northern Sweden (Ostrom et al, 2015)
- Young, male drivers are the most dangerous in all conditions (Chipman, 1993)
- Men have higher crash risk in all conditions of light conditions & crash severity (Massie, 1997)
- Men are at a higher risk in being in a fatal crash (Massie, 1995)
- Women drivers make more mistakes when they are stereotyped (Moe, 2015)

→ All show that driver's gender is not the only determinant of car crash!

Motivation & Literature

Ex.

A study by Pérez et al. published in the US National Library of Medicine and run on Traffic Crashes in Catalonia, Spain (2004-2008) has shown that there exists interaction between sex and age in road traffic injury risk, being higher among men in some age groups, and among women in other groups. However, these age groups vary depending on mode of transport and severity.

- There seems to be a complex relationship between gender and traffic injury rate, swaying in both directions depending on age.
- Is this a particular phenomenon of the Catalonia dataset? We suspect the relationship will also be complex in Chicago: there should be some differences, but dependent on other variables (so not necessarily a reflection of a gender's driving skills).

Motivation & Literature

In our project:

- Use *real-life data* on car crashes spanning 8 years in Chicago
 - Comprehensive dataset on *demographic information* of drivers & *driving conditions*
 - Statistical methods to find what *other factors* might affect car crash outcomes
 - *Machine learning* models to predict the influence of driver's gender on car crashes
- Examine the relationship between driver's gender & car crashes
- We predict that the presence of a relationship, but not the direction

Motivation & Literature

Hypothesis:

There **is a difference** in injury rate in crashes with female drivers compared to drivers of other genders.

H_0 : There is no difference in injury rate between crashes with female drivers and crashes with no female drivers

H_a : There is a difference in injury rate between crashes with female drivers and crashes with no female drivers

```
T statistic: -12.194224021177627 p-value: 3.34706898828716e-34  
We reject the null hypothesis
```

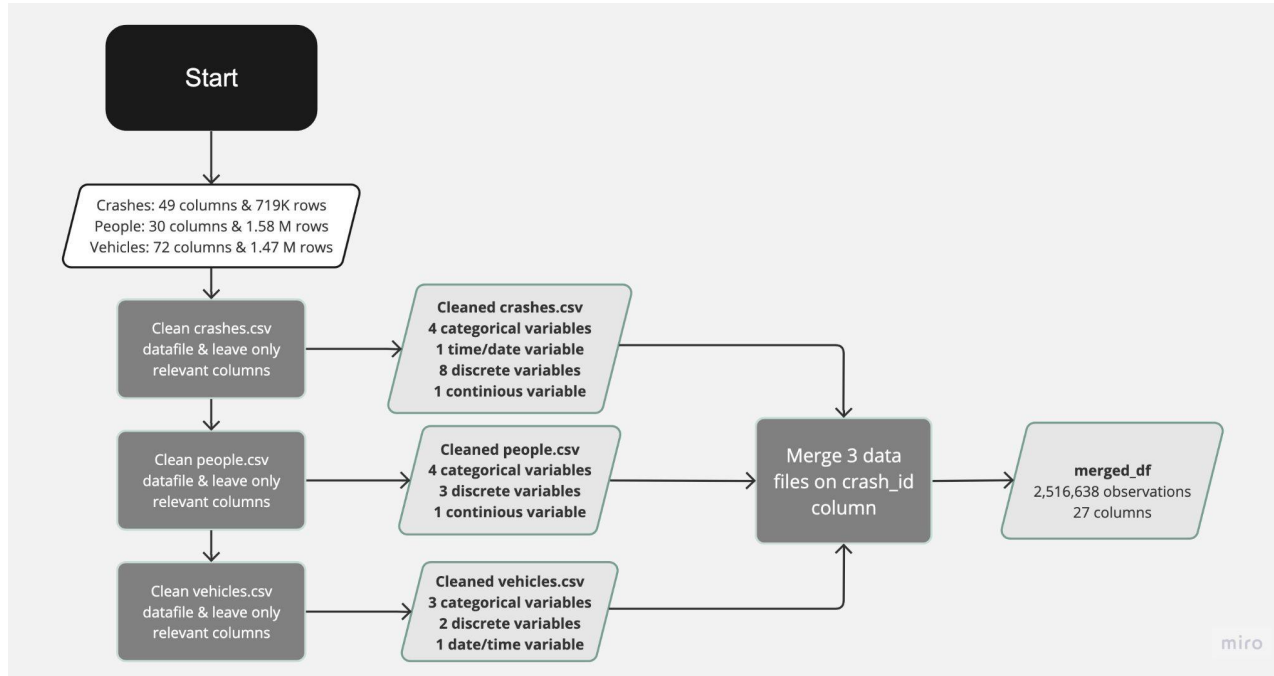
→ There is a difference between injury rate between genders

HOW?

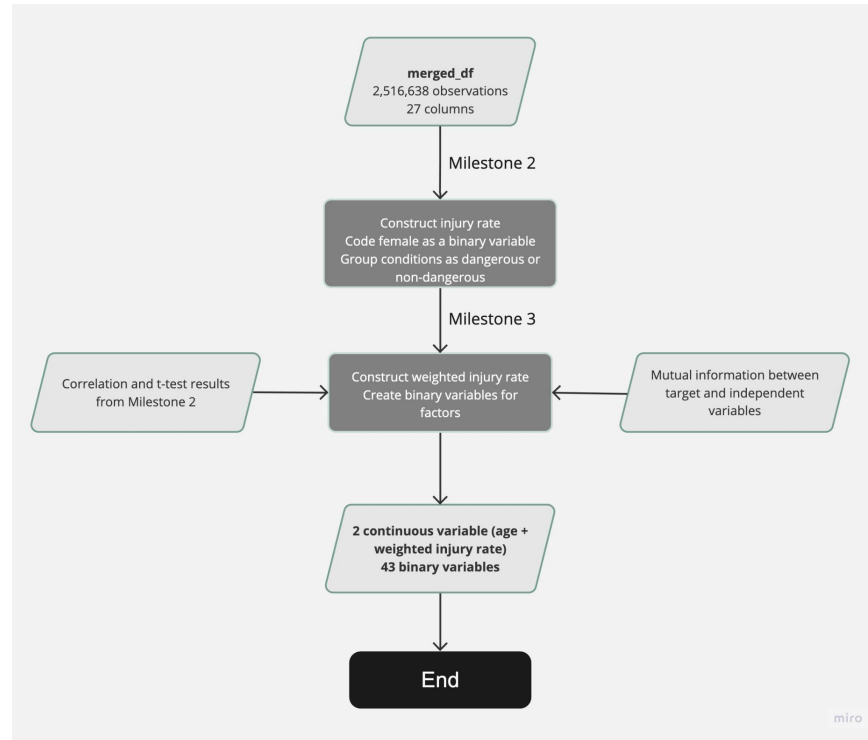


Data Description

Data journey



Data journey

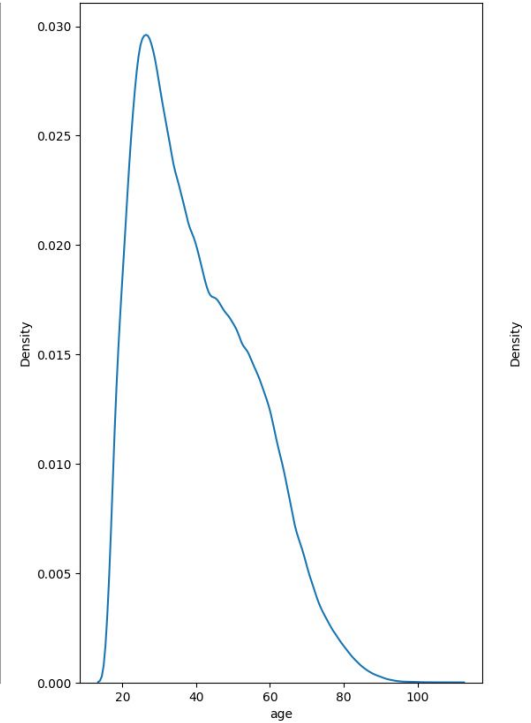
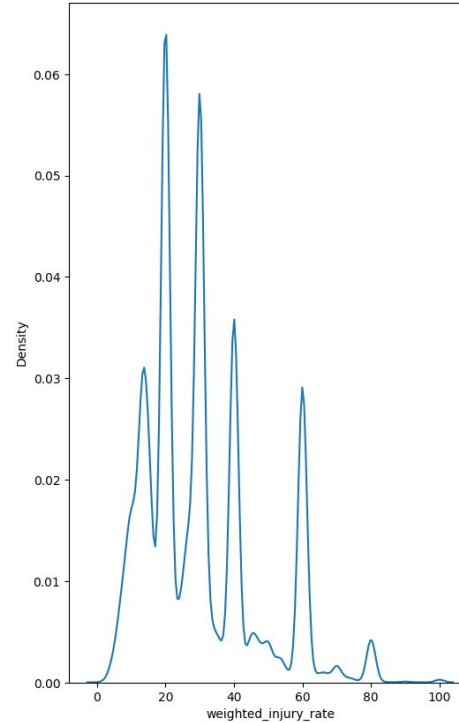


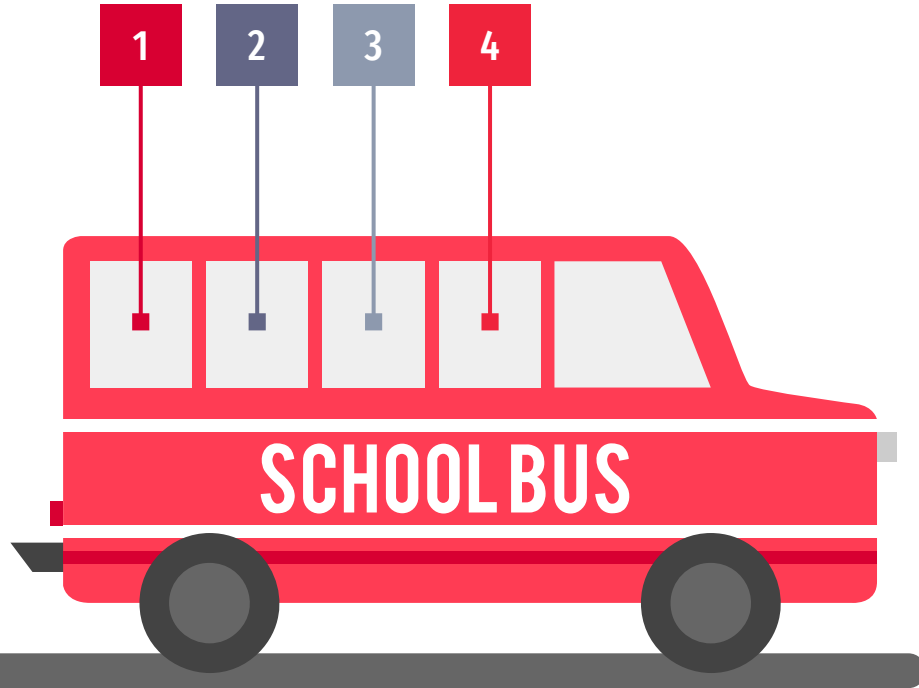
Data description

Variable	Number of 0s	Number of 1s	Mean	Total Count
device_condition_functioning properly	1025726	774949	0.430366	1800675
device_condition_no controls	905802	894873	0.496965	1800675
device_condition_not functioning	1794633	6042	0.003355	1800675
device_condition_functioning improperly	1789680	10995	0.006106	1800675
device_condition_other	1786055	14620	0.008119	1800675
device_condition_missing	1800430	245	0.000136	1800675
lighting_condition_daylight	570037	1230638	0.683431	1800675
lighting_condition_darkness, lighted road	1421475	379200	0.210588	1800675
lighting_condition_darkness	1727668	73007	0.040544	1800675
weather_condition_clear	361727	1438948	0.799116	1800675
weather_condition_cloudy/overcast	1741273	59402	0.032989	1800675
weather_condition_snow	1734922	65753	0.036516	1800675
weather_condition_rain	1630107	170568	0.094724	1800675
weather_condition_fog/smoke/haze	1798022	2653	0.001473	1800675
weather_condition_blowing snow	1799678	997	0.000554	1800675
weather_condition_blowing sand, soil, dirt	1800659	16	0.000009	1800675
vehicle_age	72112	110948	8.170103	1800675
vehicle_defect_none	713800	1086875	0.603593	1800675
vehicle_defect_unknown	1125386	675289	0.375020	1800675
vehicle_defect_brakes	1793290	7385	0.004101	1800675
vehicle_defect_tires	1799800	875	0.000486	1800675
vehicle_defect_suspension	1800398	277	0.000154	1800675
vehicle_defect_windows	1800584	91	0.000051	1800675
vehicle_defect_lights	1800573	102	0.000057	1800675
vehicle_defect_wheels	1800232	443	0.000246	1800675
vehicle_defect_steering	1799930	745	0.000414	1800675
vehicle_defect_fuel system	1800524	151	0.000084	1800675
vehicle_defect_trailer coupling	1800652	23	0.000013	1800675
vehicle_defect_exhaust	1800650	25	0.000014	1800675
exceed_speed_limit_i	1798660	2015	0.001119	1800675
sex	1074792	725883	0.403117	1800675
any_injuries	1496445	304230	0.168953	1800675

Data description

	age	weighted_injury_rate
count	1.800675e+06	1.800675e+06
mean	4.037600e+01	5.045469e+00
std	1.536730e+01	1.306084e+01
min	1.600000e+01	0.000000e+00
25%	2.800000e+01	0.000000e+00
50%	3.800000e+01	0.000000e+00
75%	5.100000e+01	0.000000e+00
max	1.100000e+02	1.000000e+02

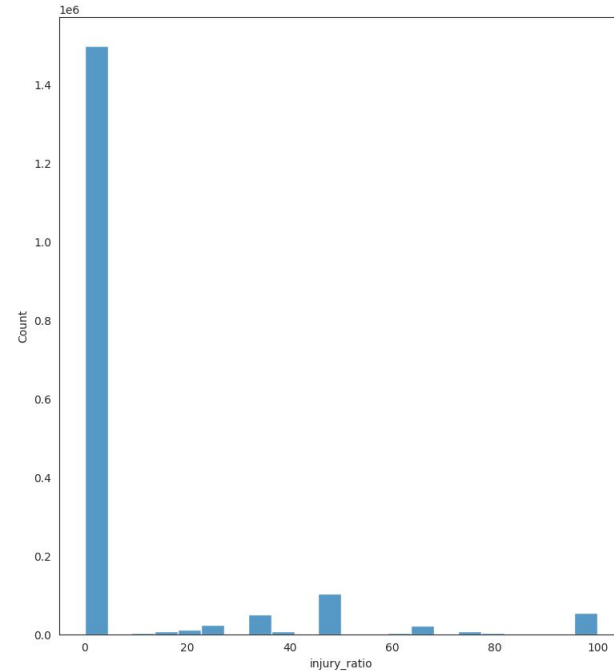
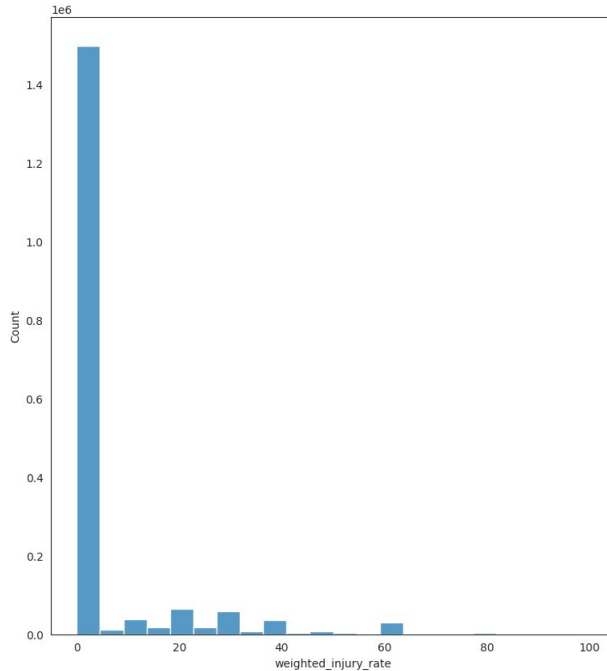




Visualizations

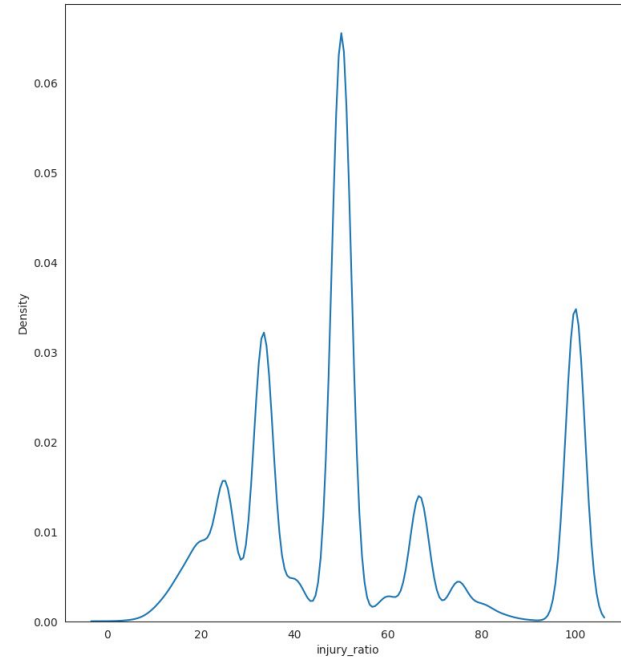
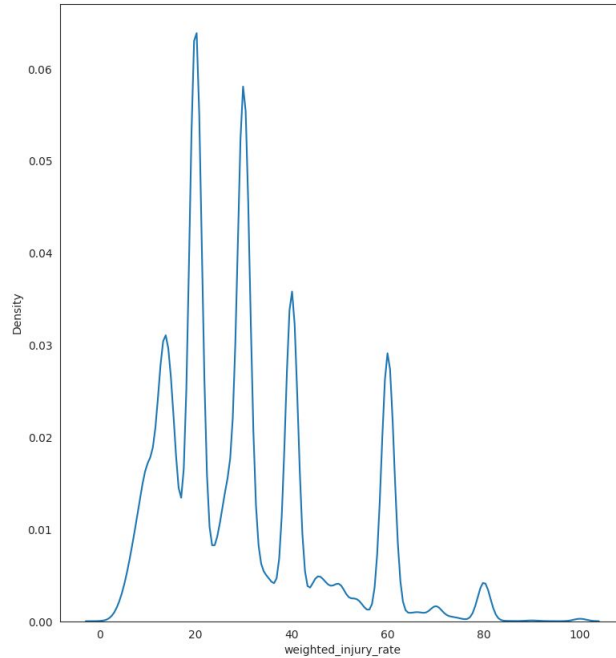
Exploratory Data Analysis - New Variable

- Similar number of 0's, but a different distribution of nonzero values

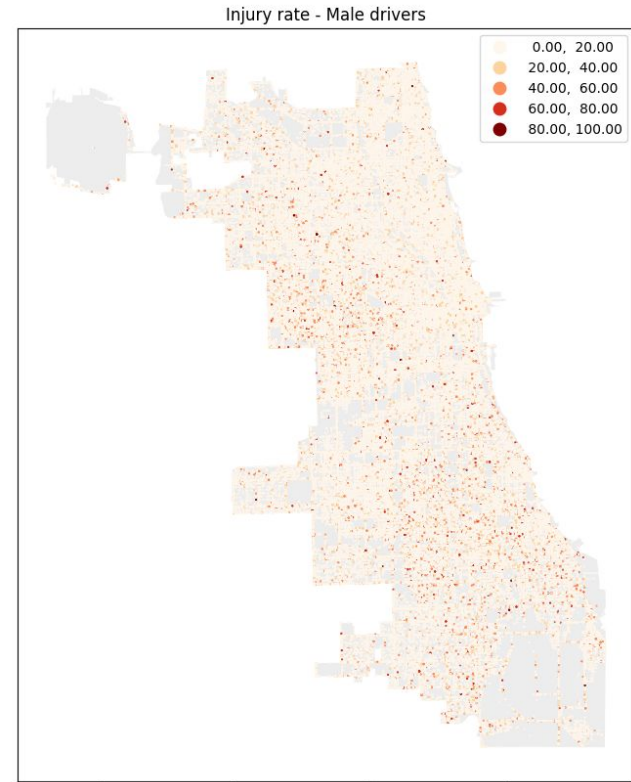
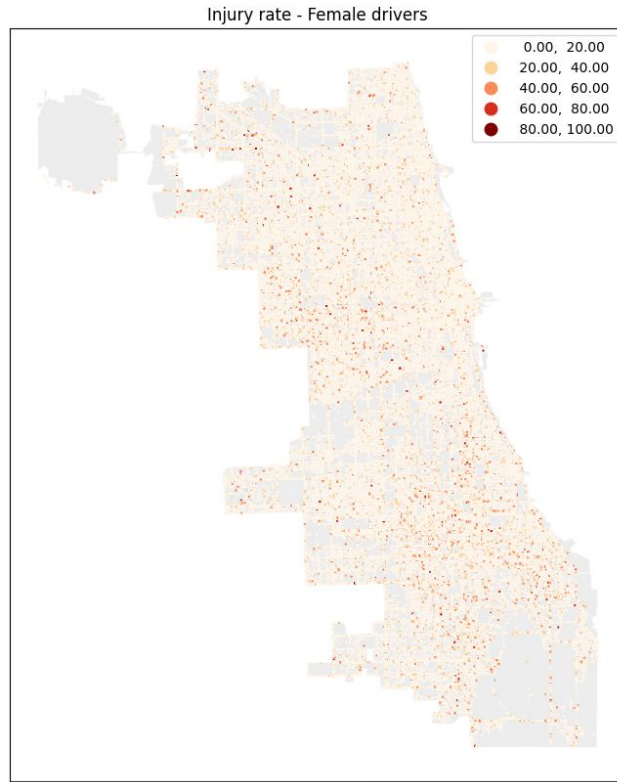


Exploratory Data Analysis - Injury Rate distribution

- What happens away from 0?
- More meaning in lower injury rates: a very high injury rate is now an outlier

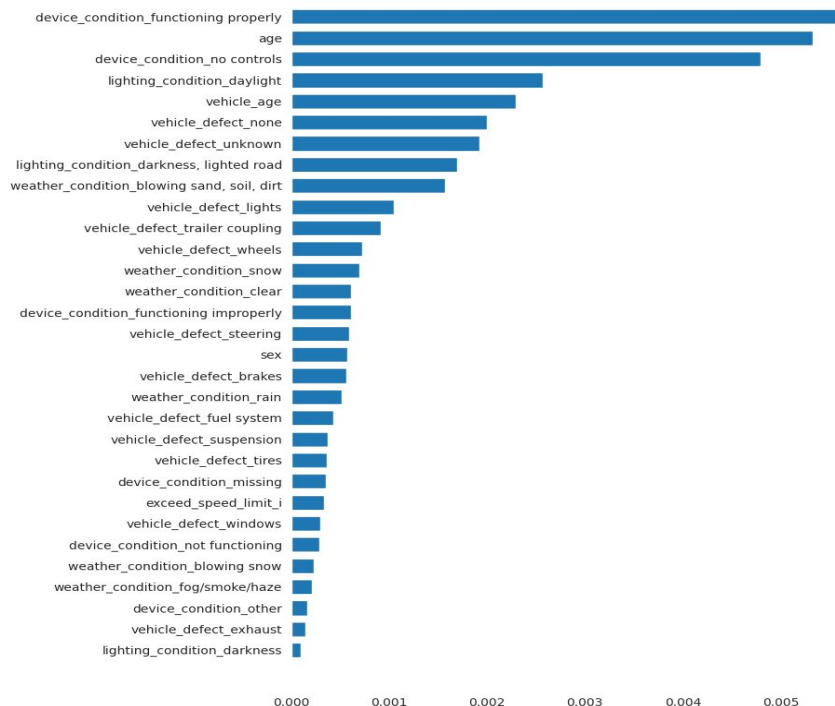


Exploratory Data Analysis - Geographical Analysis



Feature Selection

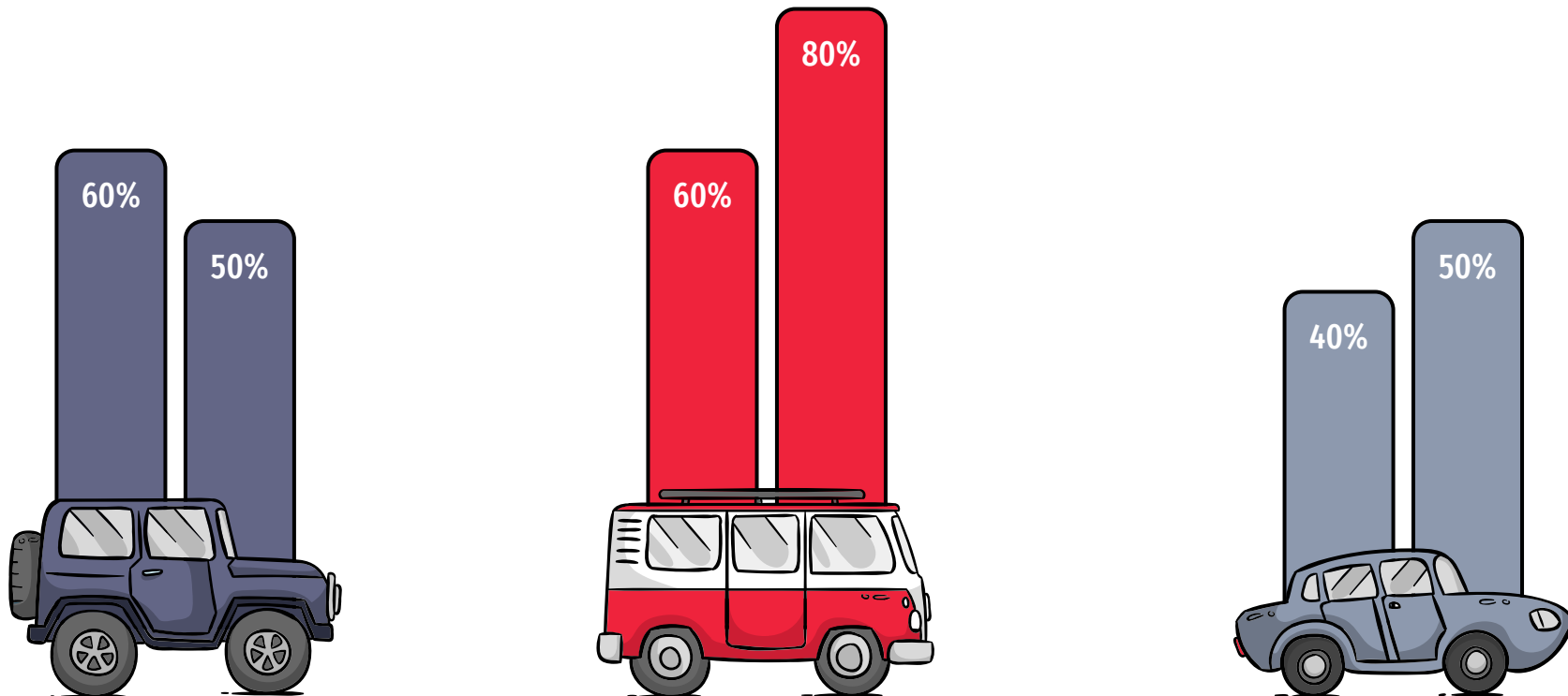
Mutual Information with Injury Rate



Features cut off:

	0
device_condition_worn reflective material	0.000000
weather_condition_cloudy/overcast	0.000022
weather_condition_freezing rain/drizzle	0.000000
weather_condition_other	0.000000
weather_condition_sleet/hail	0.000000
weather_condition_severe cross wind gate	0.000000
lighting_condition_dawn	0.000000
lighting_condition_dusk	0.000000
vehicle_defect_other	0.000000
vehicle_defect_engine/motor	0.000000
vehicle_defect_signals	0.000000
vehicle_defect_restraint system	0.000000
vehicle_defect_cargo	0.000000

Machine Learning Models



Regression Model #1: Linear Regression

```
[ ] ### Linear Regression
    LR = LinearRegression()

    param_grid = {
        'fit_intercept': [True, False]

    # Perform grid search to find the best hyperparameters
    grid_search = GridSearchCV(LR, param_grid, cv=5)
    grid_search.fit(x_train_scaled, y_train)

    # Get the best hyperparameters and fit the model
    best_params = grid_search.best_params_
    best_LR = LinearRegression(**best_params)
    best_LR.fit(x_train_scaled, y_train)
    y_pred = best_LR.predict(x_test_scaled)

    # Calculate metrics
    linreg_mae = mean_absolute_error(y_test, y_pred)
    linreg_mse = mean_squared_error(y_test, y_pred)
    linreg_r2 = r2_score(y_test, y_pred)

    print(f'Mean Absolute Error: {linreg_mae}.\n
          Mean Squared Error: {linreg_mse}.\n
          R-squared Score: {linreg_r2}.')
```

```
Mean Absolute Error: 8.031030849776158.
Mean Squared Error: 162.61305060301564.
R-squared Score: 0.01632448617401605.
```

- Baseline model to compare

Regression Model #2: Ridge Linear Regression

```
▶ ### Ridge Linear Regression
alphas = np.linspace(0.01,3,10)

# Initializing the instance of Ridge
ridge = Ridge()

# Setting parameter grid for grid search
param_grid = {'alpha': alphas}

# defining grid search with 5-fold cross validation
grid_search = GridSearchCV(ridge, param_grid, cv = 5)

# fitting the train
grid_search.fit(x_train_scaled, y_train)

# Printing the best set of parameters
print('Best parameters {}'.format(grid_search.best_params_))
```

```
Best parameters {'alpha': 3.0}
Mean Absolute Error: 8.031071148240251.
Mean Squared Error: 162.61285460529766.
R-squared Score: 0.01632567179937172.
```

- L2 Regularization
- Avoid overfitting
- Ridge regularization will shrink the coefficients for least important features, very close to zero

Regression Model #3: Lasso Linear Regression

```
▶ ### Lasso Linear Regression
alphas = np.linspace(0.01,1.5,5)

# Initializing the instance of Lasso
lasso = Lasso()

# Setting parameter grid for grid search
param_grid = {'alpha': alphas}

# defining grid search with 5-fold cross validation
grid_search = GridSearchCV(lasso, param_grid, cv = 5)

# fitting the train
grid_search.fit(x_train_scaled, y_train)

# Printing the best set of parameters
print('Best parameters {}'.format(grid_search.best_params_))
```

```
☞ Best parameters {'alpha': 0.01}
Mean Absolute Error: 8.05222366030545.
Mean Squared Error: 162.80963239506278.
R-squared Score: 0.015135327649631214.
```

- L1 Regularization
- Sparser model
- Lasso regularization will shrink the coefficients for least important features to zero

Regression Model #4: Linear SVM Regression

```
▶ ### Linear SVM Regression
from sklearn.svm import LinearSVR

# Set the hyperparameters to tune
param_grid = {'C': [0.1, 1, 10]}

# Create an instance of the SVR model
svr = LinearSVR()

# Perform grid search to find the best hyperparameters
grid_search = GridSearchCV(svr, param_grid, cv=5)
grid_search.fit(x_train_scaled, y_train)

# Printing the best set of parameters
print('Best parameters {}'.format(grid_search.best_params_))
```

```
☞ Best parameters {'C': 10}
Mean Absolute Error: 4.901510555749394.
Mean Squared Error: 189.2921588550866.
R-squared Score: -0.14506222553796655.
```

- Finds the line that has the maximum distance from the closest data points
- More robust to outliers.
- While the R square is worse than linear regression, the MAE is significantly lower.

Regression Model #5: Decision Tree Regression

```
▶ # Decision Tree Regression
from sklearn.tree import DecisionTreeRegressor

# Define the parameter grid to search over
param_grid = {
    'max_depth': [5, 10, 15],
    'min_samples_split': [5, 10]
}

# Create an instance of the decision tree regressor model
dt = DecisionTreeRegressor()

# Perform grid search to find the best hyperparameters
grid_search = GridSearchCV(dt, param_grid, cv=5)
grid_search.fit(x_train_scaled, y_train)

# Printing the best set of parameters
print('Best parameters {}'.format(grid_search.best_params_))
```

Mean Absolute Error: 8.04050854406083.
Mean Squared Error: 162.67717661329857.
R-squared Score: 0.015936576434417926.

- Model nonlinear relationships
- R-square is not as high as linear regression
- MAE and MSE are worse than linear regression.

Regression Models Performance Summary

	Baseline Model				
Model	Linear Regression	Ridge Regression	Lasso Regression	Linear SVM Regression	Decision Tree Regression
MAE	8.031031	8.031071	8.052224	4.901446	8.040509
MSE	162.613051	162.612855	162.809632	189.328052	162.677177
R2	0.016324	0.016326	0.015135	-0.145279	0.015937

Importance of Driver's Gender

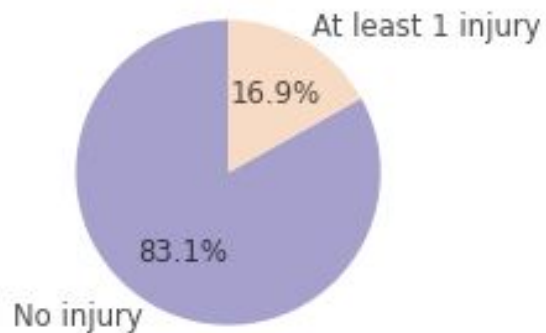
	(1) Without CVs	(2) External	(3) External, Vehicles	(4) External, Vehicles, Drivers
sex	0.1323*** (0.0206)	0.2063*** (0.0206)	0.2389*** (0.0205)	0.2193*** (0.0206)
device_condition_functioning properly		1.7245*** (0.0475)	9.9541*** (0.2118)	2.3203*** (0.0483)
device_condition_no controls		-0.0502 (0.0472)	-0.9190*** (0.1582)	0.5542*** (0.0480)
device_condition_not functioning		1.8888*** (0.1789)	0.4934*** (0.1585)	2.5058*** (0.1787)
device_condition_functioning improperly		2.1762*** (0.1368)	0.8178*** (0.2191)	2.7821*** (0.1367)
device_condition_other		1.0187*** (0.1203)	0.4689 (0.4671)	1.4311*** (0.1201)
device_condition_missing		1.7228** (0.8524)	0.1539 (0.7956)	2.3198*** (0.8500)
weather_condition_clear		0.7466*** (0.0421)	-0.2150 (1.4211)	0.9164*** (0.0420)
weather_condition_snow		-0.2512*** (0.0663)	1.3424 (1.3573)	-0.1217* (0.0662)
weather_condition_rain		0.9376*** (0.0519)	-0.4452 (0.6430)	1.0707*** (0.0518)
weather_condition_fog/smoke/haze		1.5085*** (0.2666)	2.4596*** (0.5002)	1.6776*** (0.2658)
weather_condition_blowing snow		0.9057** (0.4245)	0.0713 (1.1878)	0.7973* (0.4233)
weather_condition_blowing sand, soil, dirt		-3.4006 (3.2066)	-2.8539 (3.0185)	-3.9321 (3.1973)
lighting_condition_daylight		-0.1248*** (0.0430)	6.5520** (2.6713)	-0.0033 (0.0429)
lighting_condition_darkness, lighted road		2.1423*** (0.0467)	14.4873*** (0.3170)	2.1031*** (0.0466)
age				-0.8872*** (0.0619)
const	4.8680*** (0.0131)	3.0295*** (0.0599)	1.5061*** (0.1699)	1.7470*** (0.1707)
R-squared	0.0000	0.0108	0.0164	0.0165
R-squared Adj.	0.0000	0.0108	0.0164	0.0165

Standard errors in parentheses.

* p<.1, ** p<.05, ***p<.01

Classification

Proportion of Crashes with Injuries/ No Injuries



```
[ ] # make any_injuries column  
df['any_injuries'] = np.where(df['weighted_injury_rate']>0, 1, 0)
```

Feedback incorporation

```
from imblearn.under_sampling import RandomUnderSampler
import pandas as pd

# Extract the target variable and feature variables
y = df_class['any_injuries']
X = df_class.drop(['any_injuries'], axis=1)

# Combine X and y into a single dataframe
df_combined = pd.concat([X, y], axis=1)

# Perform random undersampling to achieve equal class distribution
rus = RandomUnderSampler(sampling_strategy='auto', random_state=42)
X_resampled, y_resampled = rus.fit_resample(X, y)

# Check the class distribution
print("Class distribution:")
print("Class 0:", sum(y_resampled == 0))
print("Class 1:", sum(y_resampled == 1))
```

```
Class distribution:
Class 0: 304230
Class 1: 304230
```

Classification Model # 1: Linear SVM Classification

```
c_params = {'C': [0.1, 1, 5]}

svm= LinearSVC()
svm_grid = GridSearchCV(estimator= svm, param_grid = c_params, cv=3)
>svm_grid.fit(X_train, y_train)

print('Best parameters {}'.format(svm_grid.best_params_))
print('Best score {}'.format(svm_grid.best_score_))

pd.DataFrame(svm_grid.cv_results_)
```

```
Best parameters {'C': 0.1}
Best score 0.8326162411317978
```

Classification Model # 1: Linear SVM Classification

```
svm_prec = precision_score(y_pred, y_test)
svm_recall = recall_score(y_pred, y_test)
svm_auc = roc_auc_score(y_pred, y_test)
svm_acc= svmclas.score(X_test, y_test)

print('Test Precision {}'.format(svm_prec))
print('Test Recall {}'.format(svm_recall))
print('Test AUC-ROC {}'.format(svm_auc))
print('Test Accuracy {}'.format(svm_acc))
```

```
Test Precision 0.506483913328956
Test Recall 0.5939021808172772
Test AUC-ROC 0.581397383013924
Test Accuracy 0.579569733425369
```

Classification Model # 2: Logistic Regression

```
from sklearn.metrics import accuracy_score, r2_score

def model_logit(x_train, y_train, x_test, y_test):
    C_range = [0.01, 0.1, 1, 10] # range of C parameters to try in GridSearch
    param_grid = {"C": C_range}

    logreg = LogisticRegression(penalty="l2", max_iter=1000) #we use l2 because best features are already selected
    logreg_cv = GridSearchCV(param_grid=param_grid, estimator=logreg, cv=3, scoring="accuracy")
    logreg_cv.fit(x_train, y_train)

    y_pred=logreg_cv.predict(x_test)

    print(logreg_cv.best_params_)

    accuracy = accuracy_score(y_test, y_pred)
    return accuracy

print(model_logit(x_train_scaled, y_train, x_test_scaled,y_test))
```

The GridSearch results in the following when run locally in another computer:

```
{'C': 0.01}
0.8325544587446374
```


Classification Model # 2: Logistic Regression

```
logreg_prec = precision_score(y_pred, y_test)
logreg_recall = recall_score(y_pred, y_test)
logreg_auc = roc_auc_score(y_pred, y_test)
logreg_acc= logreg.score(X_test, y_test)

print('Test Precision {}'.format(logreg_prec))
print('Test Recall {}'.format(logreg_recall))
print('Test AUC-ROC {}'.format(logreg_auc))
print('Test Accuracy {}'.format(logreg_acc))
```

```
Test Precision 0.5650689428759028
Test Recall 0.5842894969108562
Test AUC-ROC 0.5811086837173116
Test Accuracy 0.5810077901587615
```

Classification Model # 2: Logistic Regression

12	vehicle_defect_brakes	-1.057977
13	vehicle_defect_tires	-0.428379
14	vehicle_defect_suspension	-0.094520
15	vehicle_defect_windows	-0.052462
16	vehicle_defect_lights	-0.068538
17	vehicle_defect_wheels	-0.238672
18	vehicle_defect_steering	-0.252961
19	vehicle_defect_fuel system	-0.070625
20	vehicle_defect_trailer coupling	-0.020167
21	vehicle_defect_exhaust	0.008062
22	exceed_speed_limit_i	0.851537
23	weather_condition_clear	0.520650
24	weather_condition_cloudy/overcast	0.620550
25	weather_condition_snow	0.349579
26	weather_condition_rain	0.609880
27	weather_condition_fog/smoke/haze	0.422003
28	weather_condition_blowing snow	0.209148
29	weather_condition_blowing sand, soil, dirt	-0.009706
30	age	-0.002697
31	sex	0.059900

We see that

- A lot of coefficients are intuitive, e.g.
- No control devices, snow, darkness, and vehicle age are associated with higher probability of injuries crashes
- Or higher age is associated with lower probability of injuries crashes => older drivers = more conscious drivers?
- However, some of the coefficients do not make a lot of sense: device control functioning properly, clear weather conditions, breaks defects
- Also, we see that **being female** is associated with **increased probability** of a crash to involve any injury
- Can not jump to causation conclusions
- Omitted variable bias is possible



Classification Model # 3: Decision Tree Classification

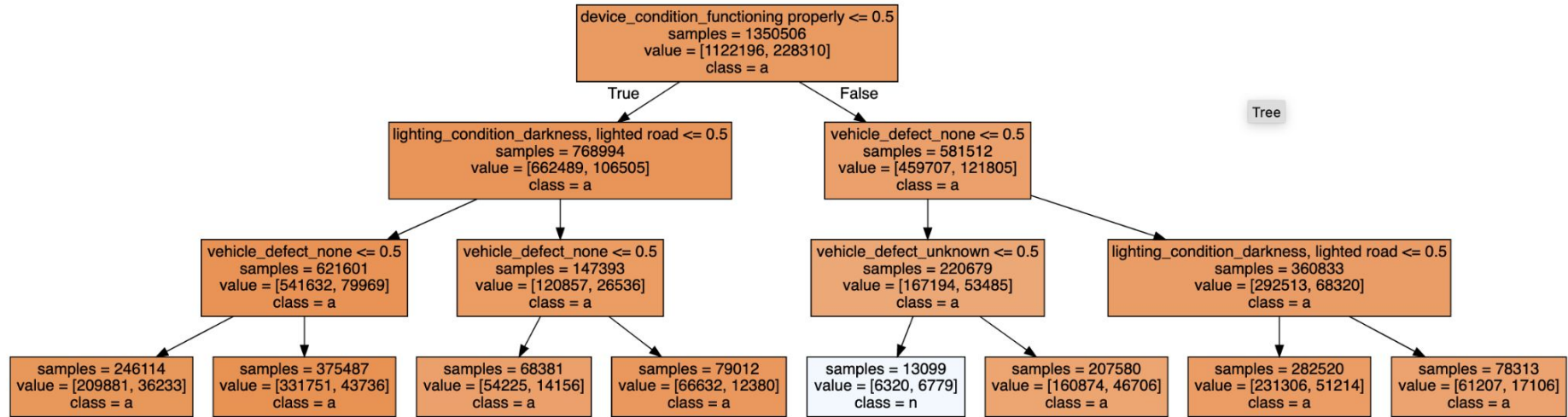
```
#Finding best parameters for the model
from sklearn.tree import DecisionTreeClassifier
param_grid = {'max_depth': [2, 3, 4, 5, 6, 7, 8, 9, 10]}

clf = DecisionTreeClassifier(random_state=0)
grid_search = GridSearchCV(clf, param_grid=param_grid, cv=3, scoring='accuracy')
grid_search.fit(X_train, y_train)

# print the best value for max_depth and the corresponding score
print('Best max_depth:', grid_search.best_params_['max_depth'])
print('Best score:', grid_search.best_score_)
```

```
Best max_depth: 7
Best score: 0.8346893682360812
```

Classification Model # 3: Decision Tree Classification



For illustrative purposes only, max_depth is 3

Classification Model # 3: Decision Tree Classification

```
dt_prec = precision_score(y_pred, y_test)
dt_recall = recall_score(y_pred, y_test)
dt_auc = roc_auc_score(y_pred, y_test)
dt_acc= dt.score(X_test, y_test)

print('Test Precision {}'.format(dt_prec))
print('Test Recall {}'.format(dt_recall))
print('Test AUC-ROC {}'.format(dt_auc))
print('Test Accuracy {}'.format(dt_acc))
```

```
Test Precision 0.5402987524622456
Test Recall 0.5892409595417114
Test AUC-ROC 0.5819199693807718
Test Accuracy 0.5813200539065838
```

Classification Model # 4: Random Forest Classification

```
clf_prec = precision_score(y_pred, y_test)
clf_recall = recall_score(y_pred, y_test)
clf_auc = roc_auc_score(y_pred, y_test)
clf_acc= clf.score(X_test, y_test)

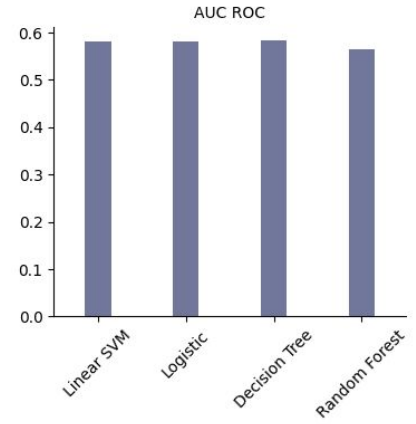
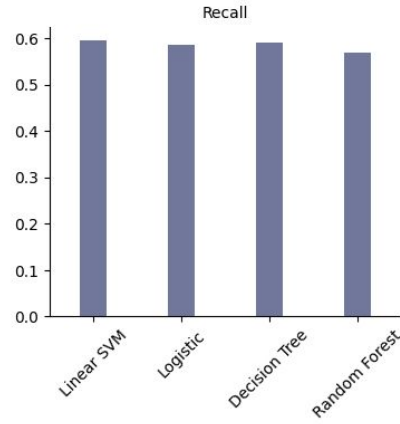
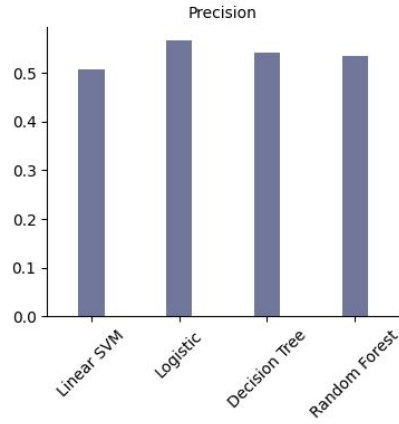
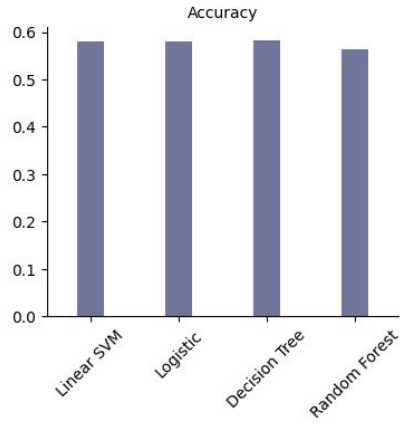
print('Test Precision {}'.format(clf_prec))
print('Test Recall {}'.format(clf_recall))
print('Test AUC-ROC {}'.format(clf_auc))
print('Test Accuracy {}'.format(clf_acc))
```

```
Test Precision 0.5343893630991464
Test Recall 0.5684377782822022
Test AUC-ROC 0.56406552300273
Test Accuracy 0.5638086316273871
```

Classification Summary:

	Classifier Model	Accuracy	Precision	Recall	AUC ROC
0	Linear SVM	0.832488	0.042188	0.559112	0.697566
1	Logistic	0.832493	0.031066	0.584183	0.709464
2	Decision Tree	0.834723	0.048990	0.646014	0.741594
3	Random Forest	0.822975	0.082701	0.388696	0.613935

	Classifier Model	Accuracy	Precision	Recall	AUC ROC
0	Linear SVM	0.579570	0.506484	0.593902	0.581397
1	Logistic	0.581008	0.565069	0.584289	0.581109
2	Decision Tree	0.581320	0.540299	0.589241	0.581920
3	Random Forest	0.563809	0.534389	0.568438	0.564066



Importance of the Driver's Gender:

results_df

Baseline

	Features	Accuracy	Precision	Recall	AUC ROC
0	External, Weather, Vehicle Factors, and Age	0.579437	0.582694	0.566213	0.579466
1	Female, External, Weather, Vehicle Factors, an...	0.579660	0.590204	0.527222	0.579775
2	Female	0.504428	0.506776	0.412114	0.504630
3	External Factors	0.566742	0.556666	0.664948	0.566527
4	Female and External Factors	0.566782	0.556865	0.663243	0.566570
5	Weather Factors	0.511074	0.506742	0.912613	0.510193
6	Female and Weather Factors	0.512132	0.515452	0.440268	0.512290
7	Vehicle Factors	0.546264	0.550913	0.511335	0.546341
8	Female and Vehicle Factors	0.547270	0.553432	0.499856	0.547374
9	Age	0.511376	0.515779	0.406762	0.511606
10	Female and Age	0.513138	0.516078	0.455854	0.513264

Recall Milestone 2

	(1)	(2)	(3)	(4)
	Without CVs	External	External, Vehicles	External, Vehicles, Drivers
sex_coded	0.4421*** (0.0363)	0.5688*** (0.0363)	-0.6212 (1.7534)	-0.6242 (1.7561)
device_coded		1.5813*** (0.1301)	-1.9798 (5.1137)	-1.9723 (4.8122)
weather_coded		0.2022*** (0.0468)	-3.7738** (1.9209)	-3.7941** (1.8831)
lighting_coded		3.0358*** (0.0390)	1.9771 (1.6370)	2.0444 (1.6424)
vehicle_age			0.0821 (0.1208)	0.0812 (0.1320)
vehicle_defect_coded			5.9809*** (1.6324)	5.9917*** (1.6402)
speed_limit_coded			8.8640*** (1.7752)	8.8937*** (1.6975)
age				0.0201 (0.0565)
const	9.1056*** (0.0231)	8.0800*** (0.0272)	20.9924*** (2.2964)	20.2139*** (3.0590)
R-squared	0.0001	0.0039	0.0216	0.0216
R-squared Adj.	0.0001	0.0039	0.0184	0.0180

Standard errors in parentheses.
 * p<.1, ** p<.05, ***p<.01

Importance of the Driver's Gender in Logit regression:

	Feature	Coefficient	Accuracy	Precision	Recall	AUC	ROC
0	Female	0.043745	0.504428	0.506776	0.412114	0.504630	
1	Female and External Factors	0.537048	0.564514	0.562469	0.589447	0.564459	
2	Female, External and Weather Factors	-0.011889	0.565664	0.558588	0.635102	0.565512	
3	Female, External, Weather and Vehicle Factors	0.016612	0.578470	0.582413	0.561057	0.578508	
4	Female, External, Weather, Vehicle Factors and...	-0.002628	0.578516	0.582603	0.560270	0.578556	

Limitations

- The dataset is skewed with the majority of crashes not involving injuries
 - It's possible, as a result, that the dataset is not suited to predict severity of injury
- Weighted injury rate is a self-constructed measure
 - The fact that we could not obtain very significant results for regression suggests the variable may not be suitable for making predictions based on the features we have
- Possibly, important features were overlooked in the analysis stage, since the dataset is skewed
- Dataset is only in Chicago, so the results might not be generalizable to all car crashes

Conclusion

- Regression models did not perform well in general
 - Dataset is not suited to predict severity of injury
 - Weighted injury rate is not an ideal dependent variable
- Classification models performed well
 - Dataset is more suited to predict injury/ no injury
- Driver's **gender helps predict injury/ no injury**
- **Female drivers tend to be associated with higher number of crashes involving injuries, but less severe injuries** than non-female drivers

Conclusion

- **Confirm** our hypothesis (Driver's gender influences injury rate)
- **Coherent** with literature
 - Driver's gender matters in predicting car crash
 - Women are more likely to be involved in any car crash
 - But men are more likely to cause fatal car crashes
 - Other factors matter!

References

- Baker, T.K., Falb, T., Voas, R., & Lacey, J. (2003). Older women drivers: Fatal crashes in good conditions. *Journal of Safety Research*, 34(4), pp.399-405.
<https://doi.org/10.1016/j.jsr.2003.09.012/>
- Berger, M.L. (1986). Women drivers: The emergence of folklore and stereotypic opinions concerning feminine automotive behavior. *Women Studies International Forum*, 9(3), pp.257-263.
- Edgerton, B. (2011, October 11). *Men vs. women: Who are safer drivers?* CBS News. <https://www.cbsnews.com/news/men-vs-women-who-are-safer-drivers/>.
- Massie, D.L., Campbell, K.L., & Williams, A.F. (1995). Traffic accident involvement rates by driver age and gender. *Accident Analysis & Research*, 27(1), pp.73-87.
[https://doi.org/10.1016/0001-4575\(94\)00050-V](https://doi.org/10.1016/0001-4575(94)00050-V).
- Massie, D.L., Green, P.E. & Campbell, K.L. (1997). Crash involvement rates by driver gender and role of average annual mileage. *Accident Analysis & Prevention*, 29(5), pp.675-685. [https://doi.org/10.1016/S0001-4575\(97\)00037-7](https://doi.org/10.1016/S0001-4575(97)00037-7)
- Moe, A., Cadinu, M., & Maass, A. (2015). Women drive better if not stereotyped. *Accident Analysis & Prevention*, 85, pp.199-206.
<https://doi.org/10.1016/j.aap.2015.09.021>.
- National Highway Traffic Safety Administration. *The Problem*. https://one.nhtsa.gov/nhtsa/Safety1nNum3ers/august2015/S1N_Aug15_Speeding_2.html.
- Ostrom, M., Sjogren, H., & Eriksson, A. (1995). Role of alcohol in traffic crashes involving women: Passenger car fatalities in northern Sweden. *Journal of Studies on Alcohol*, 56(5), pp.506-512. <https://doi.org/10.15288/jsa.1995.56.506>
- Santamariña-Rubio E, Pérez K, Olabarria M, Novoa AM. (2014). Gender differences in road traffic injury rate using time travelled as a measure of exposure. *Accid Anal Prev*, 65, pp.1-7. doi: 10.1016/j.aap.2013.11.015. Epub 2013 Dec 16. PMID: 24384384.

Thank you :)

