BSCI 5110 – PROGRAMMING

PREDICTION MODEL USING ARCHIVED TEXAS DEPARTMENT OF TRANSPORTATION (TXDOT) DATA

GROUP 6

MODOU BITEYE

ROBERT FAJARDO

ADEELA GULZARI

TEVI LAWSON

KATE NGUYEN

Table of Contents

I. Executive Summary

Crash Database, that we used for our project, is one of the primary data sources for road safety research. Therefore, their quality is fundamental for the accuracy of crash analyses and, consequently the design of effective countermeasures. This report will aim to answer the five "W" questions (i.e. When?, Where?, What?, and Why?) of each crash by including a range of attributes. Hence, this paper will review current literature in two major cities of Texas – Dallas and Houston – of crash data quality for each of these questions separately. The Dataset is about Texas Department of Transportation Website. The data is based on traffic collision and road accidents that have occurred in the two major cities of Texas, Dallas and Houston, which comes under Texas Department of Transportation Authority between the years 2017 and 2019. Our purpose is to gain insights about the factors that contribute to road accidents in Texas using different mining techniques such as linear regression and logistic regression.

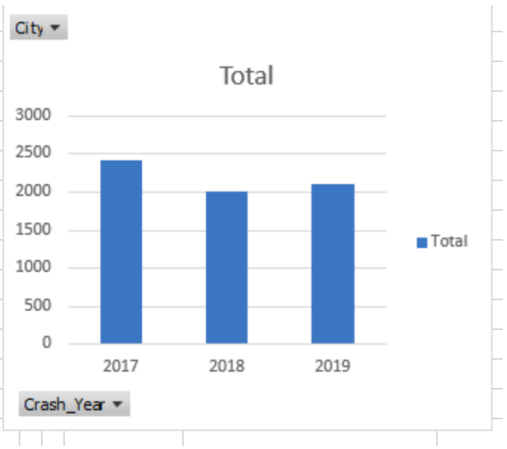
II. Introduction

Texas, the second largest state in the US, with a relatively high population density in two of its big cities, Dallas and Houston experience many road crashes every year. We worked on Texas Department of Transportation data to predict the number of deaths per unit of the two big cities (Houston and Dallas) in Texas and using two datasets defined by their respective cities. The report presents the findings from datasets, made available by the Crash Records Information System managed by the Texas Department of Transportation (TxDOT), whose goal is to provide a system framework to identify elements for roadway departure crashes. The variables we chose are based on the independent variables including weather condition, driver age, gender, ethnicity, speed limit, crash year, number of lanes, blood alcohol content, days of the week. This evaluation of the data will help to develop countermeasures that will reach those most at risk of death and injury in motor vehicle traffic crashes.

Crash Summary:

Texas is approximately 12 months behind in crash data assimilation and analysis, so the most recent year for which complete data is available is calendar year 2019. The number of fatalities per hundred million vehicle miles driven has decreased despite the rapid increases in population and miles driven from 2017 to 2019.

City	(All)
Row Labels	Sum of Unit_Death_Count
2017	2420
2018	2008
2019	2108
Grand Total	6536



III. Background and Study Objectives

Our study uses fatal crash data and geospatial analysis to examine the temporal and spatial distribution of relative fatal crashes in Dallas and Houston from 2017 to 2019. The purpose of this project is to explore the factors that contribute to road accidents in the two big cities, Dallas and Houston. There are numerous traffic collisions and accidents that have occurred in these two major cities. We will be running a multiple linear regression model for each city to predict which independent variables best explains and predicts the number of deaths per crash observed using the "number of deaths" as our target variable. Then, we will be performing a logistic regression, using a coded dummy variable "Death_YorN" to predict the probability of whether there will be a death or no death of an accident. To summarize, our objectives include:

- Propose a multiple linear regression model for each city, Dallas and Houston, to predict the unit death count when an accident happens
- Propose a logistic regression model for each city, Dallas and Houston, to predict the probability of whether there will be a death or not in case of an accident

IV. Meta Data

The dataset began with 19 variables and 26,131 observations for all of Texas. We are considering 11 variables and 11,476 data observations as sample size and splitting the data into two sets defined as TxDOT_Dallas and TxDot_Houston for the sake of statistical analysis. Among the 11 independent variables, we are interested in predicting the number of deaths using city, crash year,

speed limit, weather condition, time of day, vehicle color, age, ethnicity, gender, and blood alcohol content.

We will check the relationship between the independent variables with the dependent variable, Unit_Death_Count to predict the number of deaths per unit in those two cities, and then perform the machine learning models to generate a theory on the causes of death.

V. Data Preparation

1. Data Cleaning

Originally, there were 19 independent variables. We dropped some of the redundant variables because there were not providing enough information for our objectives or those that were not interesting, using drop (). We kept the following variables:

a. Missing value
b. Data type
c. Substring deletion

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11476 entries, 0 to 11475
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   City                            11476 non-null  object
 1   Crash_Year                      11476 non-null  int64
 2   Speed_Limit                     11420 non-null  float64
 3   Weather                         11476 non-null  int64
 4   Timeofday                       11476 non-null  int64
 5   Unit_Death_Count                11476 non-null  int64
 6   Vehicle_Color                   9692 non-null   object
 7   Person_Age                      11476 non-null  int64
 8   Person_Ethnicity                11364 non-null  object
 9   Person_Gender                   11476 non-null  object
 10  Person_Blood_Alcohol_ContentTest 11476 non-null  float64
 11  Death_YorN                      11476 non-null  int64
```

Missing values appear as empty strings in the output. The yellow highlight on the table shows Speed_Limit, Vehicle_Color and Person_Ethnicity containing the missing values. The green highlight shows the data type of each variable.

City	Crash_Year	Speed_Limit	Weather	Timeofday	Unit_Death_Count	Vehicle_Color	Person_Age	Person_Ethnicity	Person_Gender	F
DALLAS	2017	45.0	1	0	1	BLACK	21	B - BLACK	1 - MALE	
DALLAS	2017	65.0	1	1	0	BROWN	28	B - BLACK	2 - FEMALE	
DALLAS	2017	65.0	1	1	1	NaN	23	B - BLACK	1 - MALE	
DALLAS	2017	65.0	1	1	0	BLUE	25	W - WHITE	1 - MALE	
DALLAS	2017	40.0	0	0	0	WHITE	21	B - BLACK	2 - FEMALE	
...	...	...	...	...	...	...	...	...	...	
HOUSTON	2019	40.0	0	1	0	GRAY	39	B - BLACK	1 - MALE	
HOUSTON	2019	40.0	0	1	1	NaN	56	B - BLACK	1 - MALE	
HOUSTON	2019	30.0	0	1	1	BLUE	53	W - WHITE	1 - MALE	
HOUSTON	2019	35.0	0	1	1	SILVER	36	H - HISPANIC	1 - MALE	
HOUSTON	2019	35.0	0	1	1	SILVER	40	H - HISPANIC	2 - FEMALE	

(Figure 2 - Remove substring from string)

We need to manipulate our string to remove extra information from string for better understanding and faster processing. Giving a task in which substring needs to be removed from the beginning of the string. Under ‘Person_Ethnicity’ and ‘Person_Gender’, we removed the letter character, numerical value and hyphen.

2. Use of Array and Data Frame

We currently investigate the relationship between independent variables that cause car accidents and the number of deaths by using the datasets of road accidents from TxDOT. As we have many rows of accident data, rather than manually look up the data for each row, we will automate the data retrieval. Within the script, we will use a DataFrame to load and manipulate the data. This is a way to easily load and process the number of deaths and combine it with the historical data.

3. Inputting Missing Value

- The missing values processing is multiple folds
- A mean and mode substitution for missing variables has been added to our dataset
- Drop variables not impacting the outcome of our regression models
- Elimination of duplication in the dataset
- Suppression of latency caused by null values

4. Feature Selection

Dependent Variable: The dependent variables are being tested and measured in an experiment and are dependent on the independent variables.

- Unit_Death_Count - This is an interval type of variable with ranging from 0 to 3
- Death_YorNo - This is a binary target variable for the model comparison of our data. We want to know what variables attributed to the cause death in both cities to find correlation.

Independent Variables: The independent variables are used to determine the dependent variable.

- Person_Gender - Binary variable that displays 1 for male and 0 for female.
- Person_Age - An interval variable that has range from 16-90 year.
- Person_Ethnicity – An interval variable that evaluate vehicle traffic fatalities by race.
- Driver_Alcohol_Result - This is a binary variable that denotes 1 for alcohol in the system and 0 for negative alcohol results in the driver's system.
- Time_of_day – Binary variable that displays 0 for daytime and 1 for nighttime.
- Speed_Limit - An interval type of variable with speed ranging from 5 to 85.
- Vehicle_Color – An interval variable that details the color of vehicle crashed.
- Crash_Year – This is an interval variable that details the year that the crash occurred.
- Weather_Condition – Binary variable with a measurement of 0 for clear weather and 1 for unfavorable weather.
- Person_Blood_Alcohol_ContentTest – This is an interval variable that measures the amount of alcohol in the driver's system at the time of the accident. Any amount over 0.08 is legally booked as intoxicated by Texas Law.

5. Steps to Fix the Problem

- Step 1: We used strip() and Istrip() functions to fix the format of the text of 'Person_Ethnicity' and 'Person_Gender', as shown in the output below:

```
df2['Person_Ethnicity'].value_counts()
```

```
HISPANIC    4212
BLACK       3620
WHITE       2848
OTHER        372
ASIAN        296
AMER. IN      16
Name: Person_Ethnicity, dtype: int64
```

```
df2['Person_Gender'].value_counts()
```

```
MALE      8064
FEMALE    3412
Name: Person_Gender, dtype: int64
```

- Step 2: We changed the data type of categorical variables for Data Exploration
- Step 3: We imputed the missing values by using 'mean' for 'Speed_Limit' and 'value_counts' function for the others, as shown in the output below:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11476 entries, 0 to 11475
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   City                            11476 non-null  category
 1   Crash_Year                      11476 non-null  int64
 2   Speed_Limit                     11476 non-null  float64
 3   Weather                         11476 non-null  category
 4   Timeofday                       11476 non-null  category
 5   Unit_Death_Count                11476 non-null  int64
 6   Vehicle_Color                   11476 non-null  category
 7   Person_Age                      11476 non-null  int64
 8   Person_Ethnicity                11476 non-null  category
 9   Person_Gender                   11476 non-null  category
 10  Person_Blood_Alcohol_ContentTest  11476 non-null  float64
 11  Death_YorN                      11476 non-null  category
dtypes: category(7), float64(2), int64(3)
memory usage: 528.3 KB
```

(Figure 3 - Changing the data type of categorical variables)

6. Splitting the Dataset between Cities: Dallas and Houston

We split the dataset between the cities Dallas and Houston and perform our analysis separately on each city. The following output shows the result of this section of code:

	City	Crash_Year	Speed_Limit	Weather	Timeofday	Unit_Death_Count	Vehicle_Color	Person_Age	Person_Ethnicity	Person_Gender	Person_Blood_Alco
0	DALLAS	2017	45.0	1	0	1	BLACK	21	BLACK	MALE	
1	DALLAS	2017	65.0	1	1	0	BROWN	28	BLACK	FEMALE	
2	DALLAS	2017	65.0	1	1	1	WHITE	23	BLACK	MALE	
3	DALLAS	2017	65.0	1	1	0	BLUE	25	WHITE	MALE	
4	DALLAS	2017	40.0	0	0	0	WHITE	21	BLACK	FEMALE	

	City	Crash_Year	Speed_Limit	Weather	Timeofday	Unit_Death_Count	Vehicle_Color	Person_Age	Person_Ethnicity	Person_Gender	Person_Blood
5392	HOUSTON	2017	30.0	0	1	1	WHITE	24	HISPANIC	MALE	
5393	HOUSTON	2017	40.0	0	0	1	WHITE	22	BLACK	MALE	
5394	HOUSTON	2017	60.0	1	1	1	WHITE	30	BLACK	MALE	
5395	HOUSTON	2017	60.0	1	1	1	YELLOW	32	WHITE	MALE	
5396	HOUSTON	2017	60.0	1	1	0	WHITE	45	HISPANIC	MALE	

VI. Data Exploration and Visualization

1. Descriptive Statistics

```
Dallas.describe()
```

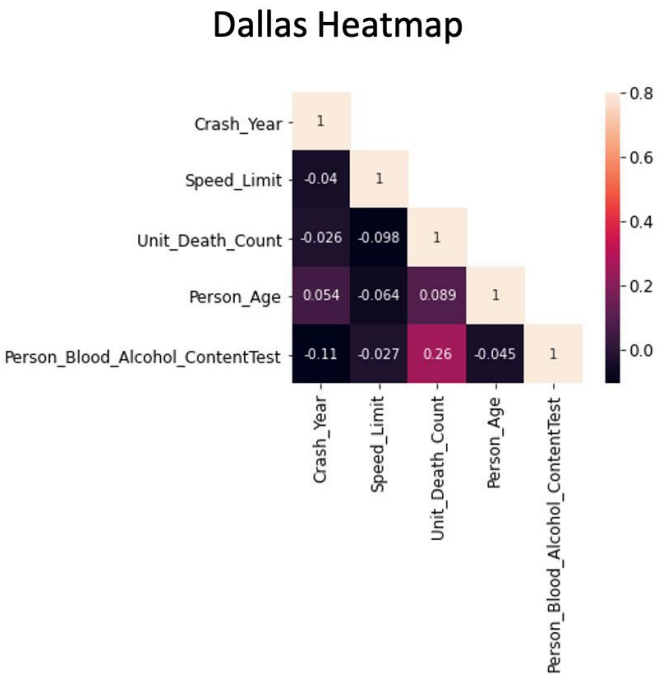
	Crash_Year	Speed_Limit	Unit_Death_Count	Person_Age	Person_Blood_Alcohol_ContentTest
count	5392.000000	5392.000000	5392.000000	5392.000000	5392.000000
mean	2017.979970	49.763662	0.553412	38.251484	0.023069
std	0.794369	14.404489	0.577289	16.051525	0.063958
min	2017.000000	15.000000	0.000000	16.000000	0.000000
25%	2017.000000	35.000000	0.000000	25.000000	0.000000
50%	2018.000000	45.000000	1.000000	34.000000	0.000000
75%	2019.000000	65.000000	1.000000	50.000000	0.000000
max	2019.000000	75.000000	3.000000	90.000000	0.380000

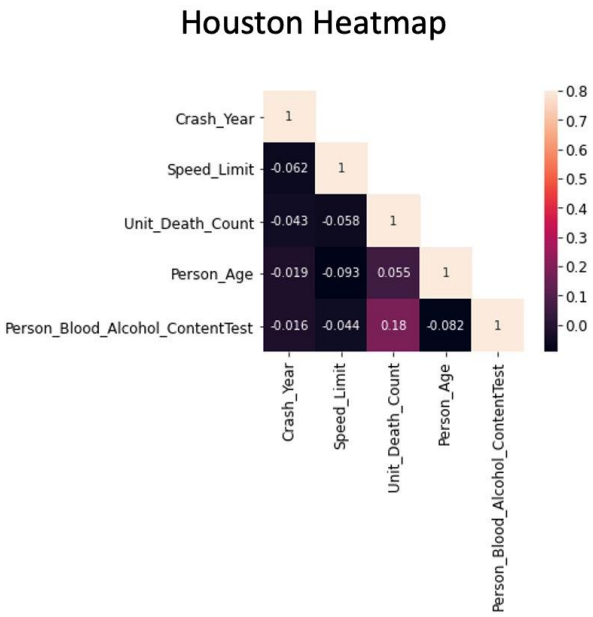
```
Houston.describe()
```

	Crash_Year	Speed_Limit	Unit_Death_Count	Person_Age	Person_Blood_Alcohol_ContentTest
count	6084.000000	6084.000000	6084.000000	6084.000000	6084.000000
mean	2017.982906	44.930574	0.583826	39.882314	0.026007
std	0.843326	11.627468	0.582253	16.875402	0.068429
min	2017.000000	20.000000	0.000000	16.000000	0.000000
25%	2017.000000	35.000000	0.000000	26.000000	0.000000
50%	2018.000000	40.000000	1.000000	36.000000	0.000000
75%	2019.000000	60.000000	1.000000	53.000000	0.000000
max	2019.000000	65.000000	3.000000	117.000000	0.406000

The Descriptive Statistics process describes the variables of data in this study. It delivers summaries on the measures. The measurement under this include mean, variance/standard deviation, and the percentage of limited prediction. We can see the descriptive statistics of both the data subsets, Dallas and Houston as shown in the output above.

2. Heatmap

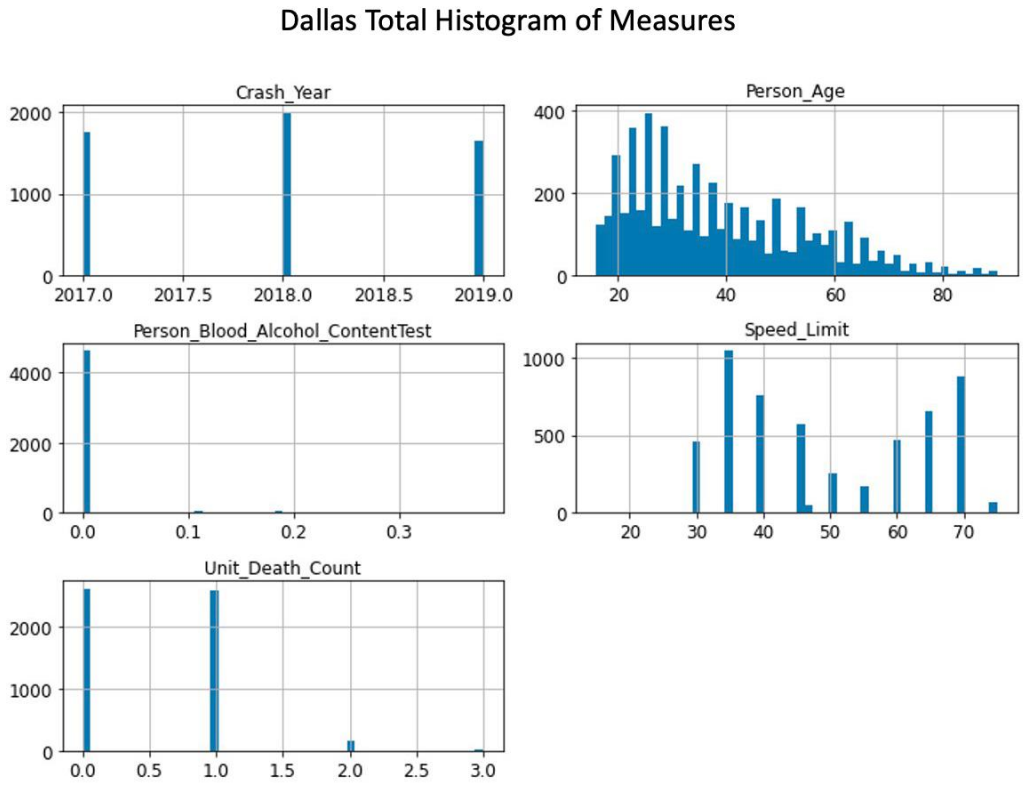




The heatmaps above show the correlation between the variables of the two datasets. The Dallas heatmap shows there is positive correlation between years and Person_Age. Moreover, not only Unit_Death_Count and Person_Age, also
Unit_Death_Count and Person_Blood_Alcohol_ContentTest are having positive correlation. Similar to the Dallas Heatmap, Houston Heatmap is having positive correlation between Unit_Death_Count and Person_Age and Person_Blood_Alcohol_ContentTest.

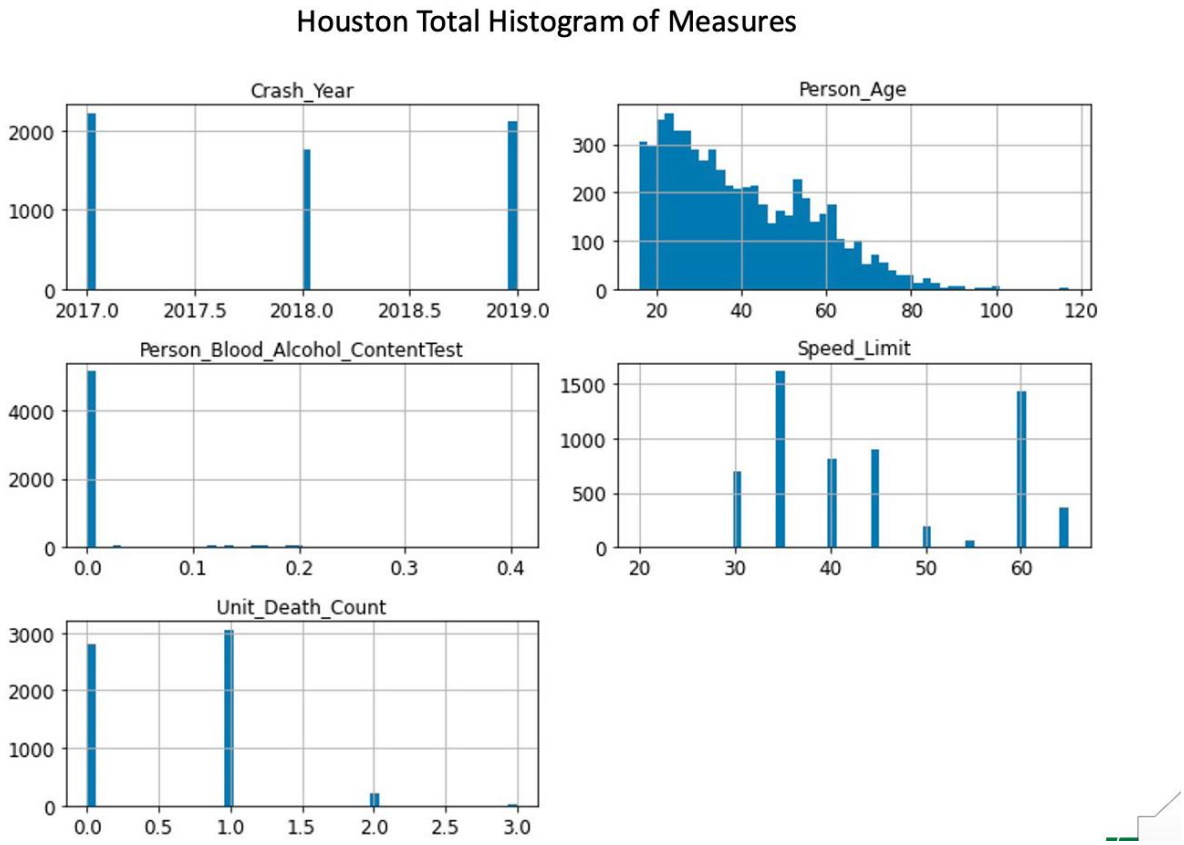
3. Histogram

The purpose of using histogram function here is to explore the distribution of the numeric variables such as Crash_Year, Person_Age, Person_Blood_Alcohol_ContentTest and Speed_Limit.



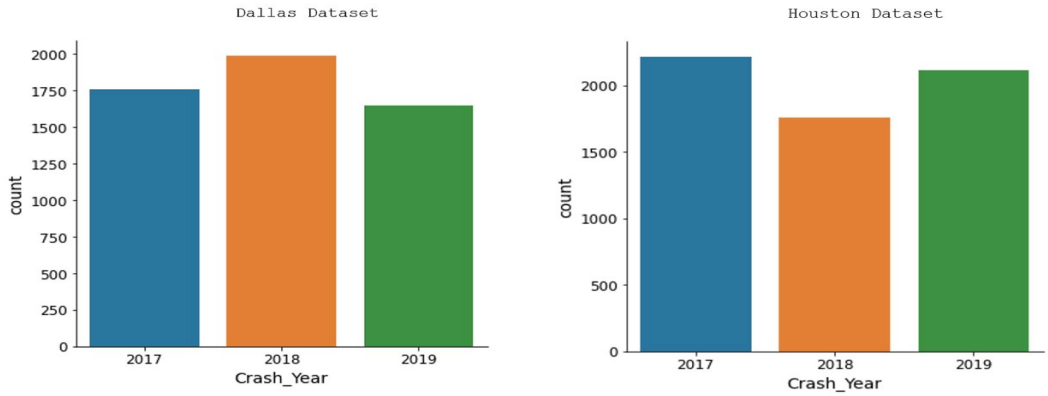
According to the 'Dallas Total Histogram of Measures', the high volume of road crash occurred in year 2018. The people group with age range between 20s to 40s caused the most of accidents in three years from 2017 to 2019. And at the Speed_Limit, a lot of fatalities were in 35 as well as

70 speed limits. There was low level of Person_Blood_Alcohol_ContentTest. That means, not a lot people was using alcohol while driving.

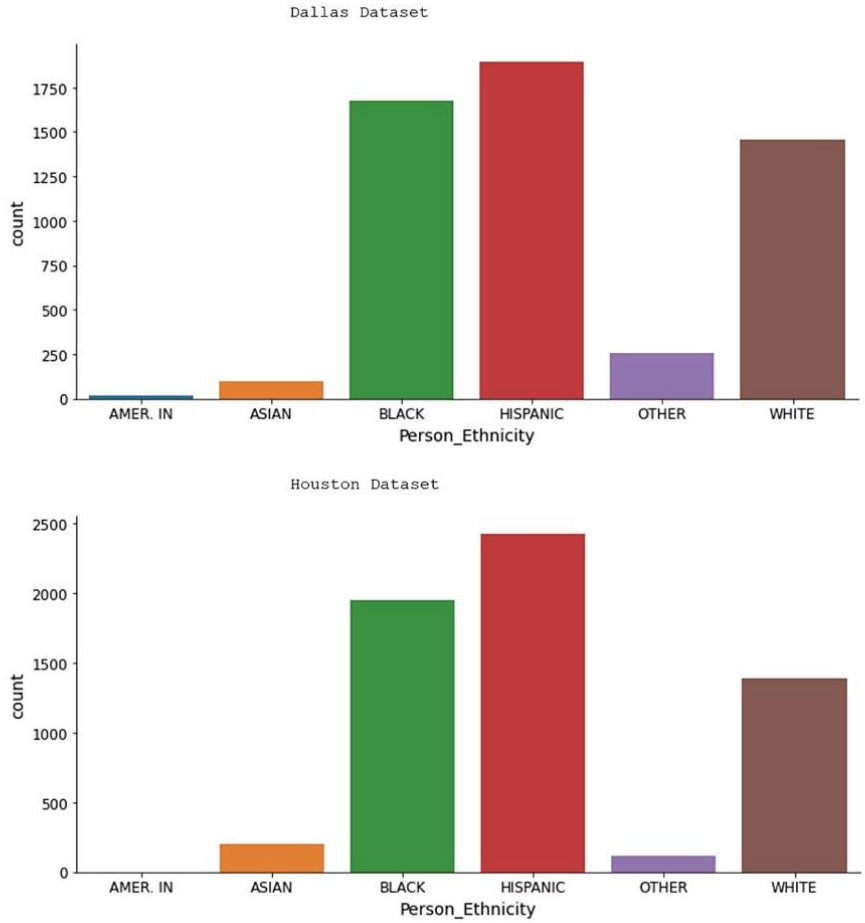


Similar to the Dallas Total Histogram of Measures, the most causing road crash were ranging between age of 20s to 30s and at the 35 and 60 speed limits showing on Houston Total Histogram of Measures. However, in 2017 and 2019, these years were occurring many fatalities by car accidents compared to the Dallas table.

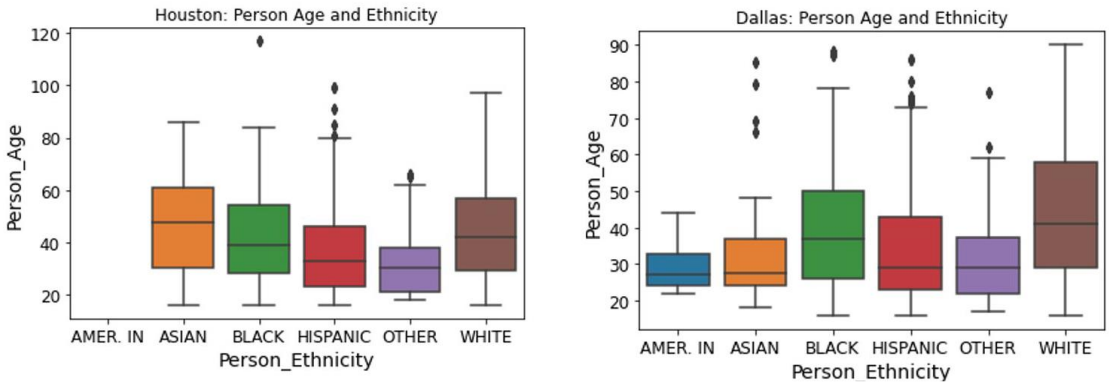
4. Frequency of Fatal Crashes Histogram per Year



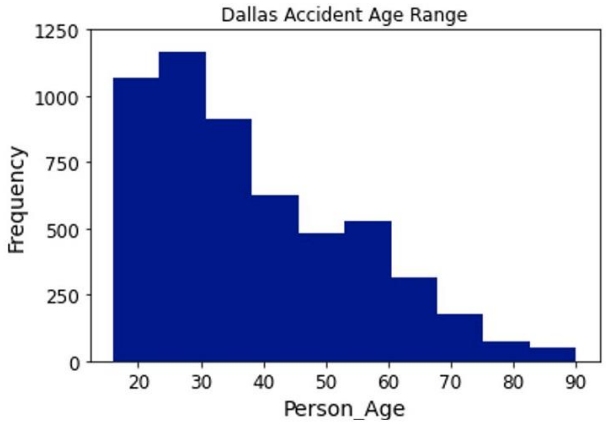
5. Aggregate Death by Ethnicity and Age Range

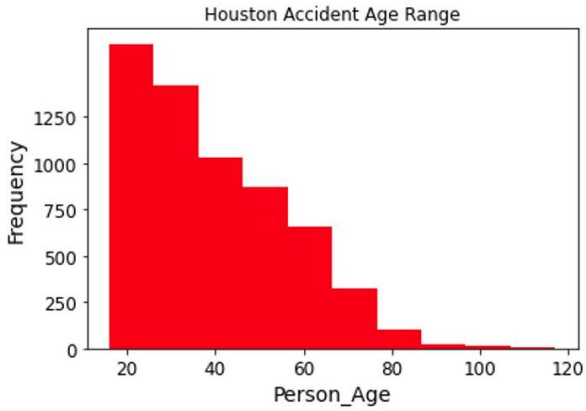


According to the two bar charts of Dallas and Houston showing above, the 'Black' and 'Hispanic' were two ethnicities involved in crashes, followed by 'White'. These groups were having the highest number of deaths. Nevertheless, there were a lot of people in the White group were killed in accidents. The two box plots below indicate the relationship between the number of deaths and ethnicities per age range.



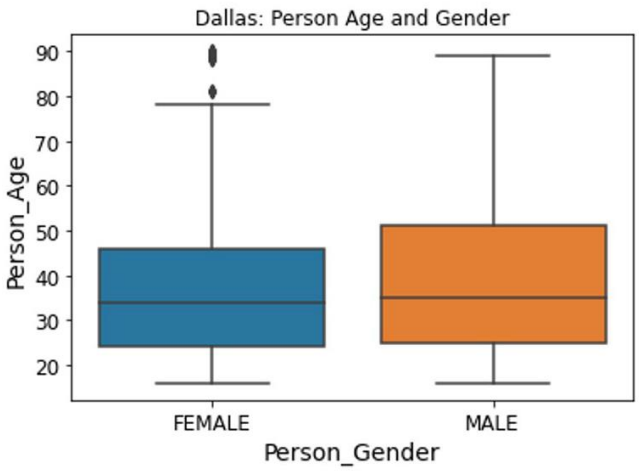
6. Frequency of Accidents by Age Range and Gender

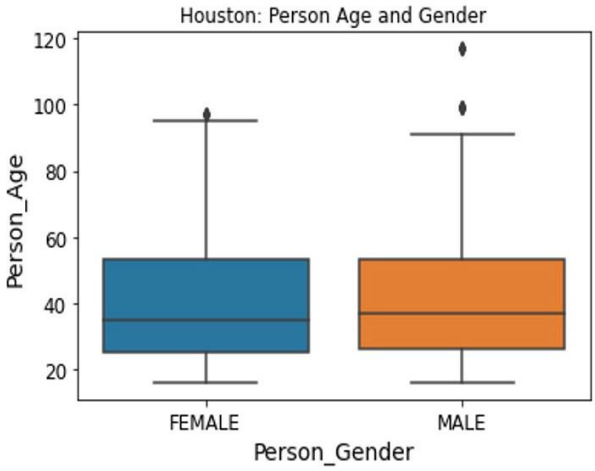




The two histograms above were showing the number of deaths per age range. In the Dallas and Houston Accident Age Range table, the age involved road crashes ranging in 20s to 35s. It seemed to be the most accidents and death were caused by the younger people.

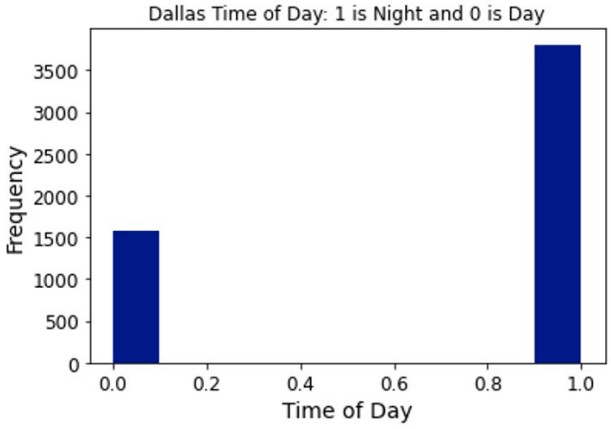
7. Accidents by Person_Age and Person_Gender

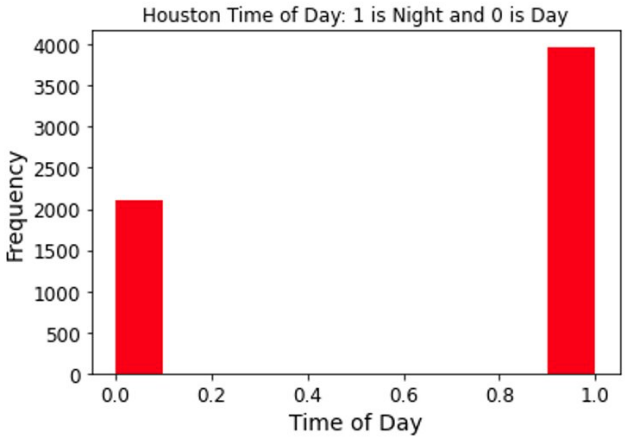




From the two datasets, the median age range for both male and female was about 35s. At the Dallas dataset, there was more male involving crashes more than man. However, the accidents showing on Houston dataset result were caused by male and female were almost equally. This is mean that there were not a lot of Male driver could drive more safe than Female driver.

8. Frequency of Accidents in a Time of Day





Through the two datasets above, there were more accidents happened at the nighttime than daytime. There were over 3,500 deaths in car accidents through the year 2017 and 2019.

VII. Feature Engineering

Based on the missing values of Vehicle_Color, Person_Ethnicity and Person_Gender, a dummy code was created which consisted of zeros and ones in order to take on a specific value. The feature, City was dropped from each dataset as it was no longer needed.

- Dummy codding - Dallas Dataset

```
# Transform the dummy variables for Dallas; dropping one to get n-1 dummy variables for model fitting
dummy_cols =['Vehicle_Color', 'Person_Ethnicity', 'Person_Gender']
dallas = pd.get_dummies(Dallas, columns=dummy_cols, drop_first = True)
dallas.head()
```

	City	Crash_Year	Speed_Limit	Weather	Timeofday	Unit_Death_Count	Person_Age	Person_Blood_Alcohol_ContentTest	Death_YorN	Vehicle_Color_BLAC
0	DALLAS	2017	45.0	1	0	1	21	0.000	1	
1	DALLAS	2017	65.0	1	1	0	28	0.000	0	
2	DALLAS	2017	65.0	1	1	1	23	0.144	1	
3	DALLAS	2017	65.0	1	1	0	25	0.000	0	
4	DALLAS	2017	40.0	0	0	0	21	0.000	0	

5 rows × 27 columns

```
#We don't need the city variable anymore
dallas_data = dallas.drop('City', axis=1)
dallas_data.head()
len(dallas_data)
```

	Crash_Year	Speed_Limit	Weather	Timeofday	Unit_Death_Count	Person_Age	Person_Blood_Alcohol_ContentTest	Death_YorN	Vehicle_Color_BLACK	Vehic
0	2017	45.0	1	0	1	21	0.000	1	1	
1	2017	65.0	1	1	0	28	0.000	0	0	
2	2017	65.0	1	1	1	23	0.144	1	0	
3	2017	65.0	1	1	0	25	0.000	0	0	
4	2017	40.0	0	0	0	21	0.000	0	0	

(Figure 4 - Data Exploration)

- Dummy codding – Houston Dataset

```
# Transform the dummy variables for Houston; dropping one to get n-1 dummy variables for model fitting
dummy_cols =['Vehicle_Color', 'Person_Ethnicity', 'Person_Gender']
houston = pd.get_dummies(Houston, columns=dummy_cols, drop_first = True)
houston.head()
```

	City	Crash_Year	Speed_Limit	Weather	Timeofday	Unit_Death_Count	Person_Age	Person_Blood_Alcohol_ContentTest	Death_YorN	Vehicle_Color
5392	HOUSTON	2017	30.0	0	1	1	24	0.278	1	
5393	HOUSTON	2017	40.0	0	0	1	22	0.000	1	
5394	HOUSTON	2017	60.0	1	1	1	30	0.000	1	
5395	HOUSTON	2017	60.0	1	1	1	32	0.090	1	
5396	HOUSTON	2017	60.0	1	1	0	45	0.000	0	

5 rows × 27 columns

```
houston_data = houston.drop('City', axis=1)
houston_data.head()
len(houston_data)
```

	Crash_Year	Speed_Limit	Weather	Timeofday	Unit_Death_Count	Person_Age	Person_Blood_Alcohol_ContentTest	Death_YorN	Vehicle_Color_BLACK	Ve
5392	2017	30.0	0	1	1	24	0.278	1	0	
5393	2017	40.0	0	0	1	22	0.000	1	0	
5394	2017	60.0	1	1	1	30	0.000	1	0	
5395	2017	60.0	1	1	1	32	0.090	1	0	
5396	2017	60.0	1	1	0	45	0.000	0	0	

5 rows × 26 columns

(Figure 4 - Data Exploration)

- Convert categorical variables to numeric for Regression Analysis

```
## DALLAS DATASET
## Converting categorical variables to numeric for regression
dallas_data['Weather'] = dallas_data.Weather.astype('uint8')
dallas_data['Timeofday'] = dallas_data.Timeofday.astype('uint8')
dallas_data['Death_YorN'] = dallas_data.Death_YorN.astype('uint8')
```

```
## HOUSTON DATASET
## Converting categorical variables to numeric for regression
houston_data['Weather'] = houston_data.Weather.astype('uint8')
houston_data['Timeofday'] = houston_data.Timeofday.astype('uint8')
houston_data['Death_YorN'] = houston_data.Death_YorN.astype('uint8')
```

VIII. Models and Analysis

1. Predictive Models

We have the following models:

- Multiple Linear Regression for Dallas dataset

- Multiple Linear Regression for Houston dataset

(The models are evaluated by using RMSE and K-fold Cross-Validation)

- Logistic Regression for Dallas dataset
- Logistic Regression for Houston dataset

(The models are evaluated by using Accuracy and Confusion Matrix)

Train-Test Split

In [83]:
```
# DALLAS DATASET
## generate the independent variable and dependent variable
X_dallas = dallas_data.drop(['Unit_Death_Count','Death_YorN'], axis = 1)
y_dallas = dallas_data['Unit_Death_Count']
## generate train and test datasets
from sklearn.model_selection import train_test_split
X_train_d, X_test_d, y_train_d, y_test_d=train_test_split(X_dallas, y_dallas, test_size=0.2,random_state=42)
```

In [84]:
```
# HOUSTON DATASET
## generate the independent variable and dependent variable
X_houston = houston_data.drop(['Unit_Death_Count','Death_YorN'], axis = 1)
y_houston = houston_data['Unit_Death_Count']
## generate train and test datasets
from sklearn.model_selection import train_test_split
X_train_h, X_test_h, y_train_h, y_test_h=train_test_split(X_houston, y_houston, test_size=0.2,random_state=42)
```

2. Linear Regression

Dallas

The below figure shows the output of the Linear Regression for Dallas dataset. This output provides the cofficients, p-value(significance) and standard error of each feature. We can observe the three highly significant variables are Speed_Limit, Person_Age and Person_Blood_Alcohol_ContentTest. With one unit increase in the Person_Blood_Alcohol_ContentTest, the Unit_Death_Count will increase by 2.3045 units. And for the Speed_Limit, it has a negative coefficient significant on Unit_Death_Count. Then, its descriptive statistics did not always mean that we were going to have a higher death rate. The value of the **R-Square and Adjusted R-square** are 0.093 and 0.088 respectively. Even though R-square of the model is not too high, but the value of adjusted R-square indicates that, the variables that we are using in the model add value in the explanation of variation of the dependent variable 'Unit_Death_Count', as it is not much different from R-square.

The below figure also presents the skewness, kurtosis and other statistics which are within the acceptable ranges.

OLS Regression Results

Dep. Variable:	Unit_Death_Count	R-squared:	0.093
Model:	OLS	Adj. R-squared:	0.088
Method:	Least Squares	F-statistic:	18.26
Date:	Tue, 24 Nov 2020	Prob (F-statistic):	1.30e-73
Time:	14:53:16	Log-Likelihood:	-3539.6
No. Observations:	4313	AIC:	7129.
Df Residuals:	4288	BIC:	7289.
Df Model:	24		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	4.5293	21.878	0.207	0.836	-38.363	47.422
Crash_Year	-0.0021	0.011	-0.194	0.846	-0.023	0.019
Speed_Limit	-0.0038	0.001	-6.416	0.000	-0.005	-0.003
Weather	0.0076	0.019	0.404	0.686	-0.029	0.044
Timeofday	0.0232	0.020	1.182	0.237	-0.015	0.062
Person_Age	0.0028	0.001	4.974	0.000	0.002	0.004
Person_Blood_Alcohol_ContentTest	2.3045	0.137	16.771	0.000	2.035	2.574
Vehicle_Color_BLACK	-0.0145	0.080	-0.182	0.856	-0.171	0.142
Vehicle_Color_BLUE	0.0019	0.082	0.023	0.982	-0.159	0.163
Vehicle_Color_BROWN	-0.2239	0.110	-2.029	0.043	-0.440	-0.008
Vehicle_Color_GOLD	-0.0920	0.093	-0.995	0.320	-0.273	0.089
Vehicle_Color_GRAY	-0.0517	0.081	-0.635	0.525	-0.211	0.108
Vehicle_Color_GREEN	0.0005	0.092	0.005	0.996	-0.180	0.181
Vehicle_Color_MAROON	-0.0895	0.098	-0.915	0.360	-0.281	0.102
Vehicle_Color_OTHER	-0.0128	0.088	-0.145	0.885	-0.186	0.160
Vehicle_Color_RED	-0.1246	0.083	-1.495	0.135	-0.288	0.039
Vehicle_Color_SILVER	-0.0457	0.083	-0.548	0.584	-0.209	0.118
Vehicle_Color_WHITE	-0.0213	0.078	-0.272	0.786	-0.174	0.132
Vehicle_Color_YELLOW	-0.2402	0.125	-1.926	0.054	-0.485	0.004
Person_Ethnicity_ASIAN	0.2500	0.160	1.561	0.119	-0.064	0.564
Person_Ethnicity_BLACK	0.3541	0.149	2.376	0.018	0.062	0.646
Person_Ethnicity_HISPANIC	0.2733	0.149	1.835	0.067	-0.019	0.565
Person_Ethnicity_OTHER	0.2931	0.153	1.911	0.056	-0.008	0.594
Person_Ethnicity_WHITE	0.3719	0.149	2.492	0.013	0.079	0.664
Person_Gender_MALE	-0.0001	0.019	-0.007	0.994	-0.037	0.037

Omnibus:	391.607	Durbin-Watson:	2.008
Prob(Omnibus):	0.000	Jarque-Bera (JB):	525.143
Skew:	0.756	Prob(JB):	9.26e-115
Kurtosis:	3.796	Cond. No.	5.26e+06

(Figure 5 - Linear Regression)

Houston

We ran regression on the Houston dataset and both the techniques of regression. The output of the regression result is provided below. The variables Crash_Year, Speed_Limit, and Person_ Ethnicity are significant at alpha 5%. The Person_Ethnicity had extremely high coefficient number. The value of the **R-Square and Adjusted R-square** are 0.074 and 0.070 respectively. Even though R-square of the model is not too high, but the value of adjusted R-square indicates that, the variables that we are using in the model add value in the explanation of variation of the dependent variable 'Unit_Death_Count', as it is not much different from R-square. However, the Speed_Limit was very similar to Dallas. Speed_Limit had a negative correlation with Unit_Death_Count.

OLS Regression Results

Dep. Variable:	Unit_Death_Count	R-squared:	0.074
Model:	OLS	Adj. R-squared:	0.070
Method:	Least Squares	F-statistic:	16.91
Date:	Tue, 24 Nov 2020	Prob (F-statistic):	2.64e-65
Time:	14:53:21	Log-Likelihood:	-4082.4
No. Observations:	4867	AIC:	8213.
Df Residuals:	4843	BIC:	8369.
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	55.3066	16.233	3.407	0.001	23.483	87.130
Crash_Year	-0.0327	0.010	-3.390	0.001	-0.052	-0.014
Speed_Limit	-0.0023	0.001	-3.284	0.001	-0.004	-0.001
Weather	-0.0041	0.018	-0.228	0.820	-0.040	0.032
Timeofday	0.0133	0.018	0.758	0.449	-0.021	0.048
Person_Age	0.0019	0.001	3.806	0.000	0.001	0.003
Person_Blood_Alcohol_ContentTest	1.4463	0.126	11.503	0.000	1.200	1.693
Vehicle_Color_BLACK	0.2868	0.077	3.720	0.000	0.136	0.438
Vehicle_Color_BLUE	0.2151	0.081	2.665	0.008	0.057	0.373
Vehicle_Color_BROWN	0.1385	0.117	1.183	0.237	-0.091	0.368
Vehicle_Color_GOLD	0.3118	0.097	3.229	0.001	0.122	0.501
Vehicle_Color_GRAY	0.1129	0.079	1.428	0.153	-0.042	0.268
Vehicle_Color_GREEN	0.3419	0.099	3.445	0.001	0.147	0.536
Vehicle_Color_MAROON	0.1749	0.091	1.932	0.053	-0.003	0.352
Vehicle_Color_OTHER	0.1103	0.090	1.222	0.222	-0.067	0.287
Vehicle_Color_RED	0.2755	0.080	3.427	0.001	0.118	0.433
Vehicle_Color_SILVER	0.2087	0.079	2.651	0.008	0.054	0.363
Vehicle_Color_WHITE	0.3457	0.076	4.533	0.000	0.196	0.495
Vehicle_Color_YELLOW	0.4339	0.106	4.110	0.000	0.227	0.641
Person_Ethnicity_ASIAN	11.1078	3.248	3.420	0.001	4.740	17.475
Person_Ethnicity_BLACK	10.9520	3.246	3.374	0.001	4.588	17.316
Person_Ethnicity_HISPANIC	10.9924	3.246	3.386	0.001	4.628	17.357
Person_Ethnicity_OTHER	11.2963	3.247	3.479	0.001	4.932	17.661
Person_Ethnicity_WHITE	10.9581	3.246	3.376	0.001	4.594	17.322
Person_Gender_MALE	0.0354	0.018	2.005	0.045	0.001	0.070

Omnibus:	348.053	Durbin-Watson:	2.047
Prob(Omnibus):	0.000	Jarque-Bera (JB):	436.405
Skew:	0.667	Prob(JB):	1.72e-95
Kurtosis:	3.612	Cond. No.	1.54e+19

(Figure 5 - Linear Regression)

3. Logistic Regression

Dallas

Logit Regression Results

Dep. Variable:	Death_YorN	No. Observations:	4313
Model:	Logit	Df Residuals:	4289
Method:	MLE	Df Model:	23
Date:	Tue, 24 Nov 2020	Pseudo R-squ.:	inf
Time:	18:07:31	Log-Likelihood:	-1.7636e+05
converged:	True	LL-Null:	0.0000
Covariance Type:	nonrobust	LLR p-value:	1.000

	coef	std err	z	P>|z|	[0.025	0.975]
Crash_Year	-0.0005	0.000	-1.214	0.225	-0.001	0.000
Speed_Limit	-0.0142	0.002	-6.154	0.000	-0.019	-0.010
Weather	0.0214	0.073	0.292	0.770	-0.122	0.165
Timeofday	-0.0774	0.074	-1.041	0.298	-0.223	0.068
Person_Age	0.0152	0.002	7.101	0.000	0.011	0.019
Person_Blood_Alcohol_ContentTest	15.5498	1.047	14.851	0.000	13.498	17.602
Vehicle_Color_BLACK	-0.4186	0.319	-1.312	0.190	-1.044	0.207
Vehicle_Color_BLUE	-0.0905	0.327	-0.276	0.782	-0.732	0.551
Vehicle_Color_BROWN	-1.0322	0.457	-2.261	0.024	-1.927	-0.137
Vehicle_Color_GOLD	-0.3554	0.369	-0.962	0.336	-1.079	0.368
Vehicle_Color_GRAY	-0.3726	0.325	-1.148	0.251	-1.009	0.264
Vehicle_Color_GREEN	0.0026	0.365	0.007	0.994	-0.713	0.718
Vehicle_Color_MAROON	-0.3826	0.385	-0.994	0.320	-1.137	0.372
Vehicle_Color_OTHER	-0.1711	0.351	-0.488	0.626	-0.858	0.516
Vehicle_Color_RED	-0.6370	0.334	-1.909	0.056	-1.291	0.017
Vehicle_Color_SILVER	-0.4543	0.333	-1.364	0.173	-1.107	0.199
Vehicle_Color_WHITE	-0.1231	0.314	-0.392	0.695	-0.738	0.492
Vehicle_Color_YELLOW	-1.0218	0.488	-2.095	0.036	-1.978	-0.066
Person_Ethnicity_ASIAN	1.0780	0.800	1.348	0.178	-0.490	2.646
Person_Ethnicity_BLACK	1.4288	0.767	1.863	0.062	-0.074	2.932
Person_Ethnicity_HISPANIC	1.1033	0.767	1.439	0.150	-0.400	2.606
Person_Ethnicity_OTHER	0.9316	0.781	1.192	0.233	-0.600	2.463
Person_Ethnicity_WHITE	1.4584	0.768	1.900	0.057	-0.046	2.963
Person_Gender_MALE	0.0467	0.073	0.643	0.520	-0.096	0.189

(Figure 6 - Logistic Regression)

The output for logistic regression on Dallas dataset is shown above. The model that we employed was Logit, using Maximum Likelihood Estimation method to estimate the parameters of the model. The output shows that the features that were highly significant include Speed_Limit, Person_Age and Person_Blood_Alcohol_ContentTest, which is consistent with the output of the linear regression for the Dallas dataset. The coefficient of Person_Blood_Alcohol_ContentTest is 15.5498, which is high. We also produced the odds ratio estimates for the Dallas dataset as shown below:

Odds Ratio Estimates – Dallas

```
estimates = log_reg_d.params
intervals = log_reg_d.conf_int()
intervals['Odds Ratio'] = estimates
intervals.columns = ['5%', '95%', 'Odds Ratio']
print(np.exp(intervals))
```

	5%	95%	Odds Ratio
Crash_Year	0.998665	1.000314e+00	9.994893e-01
Speed_Limit	0.981399	9.903417e-01	9.858603e-01
Weather	0.885282	1.178899e+00	1.021596e+00
Timeofday	0.800040	1.070743e+00	9.255474e-01
Person_Age	1.011082	1.019615e+00	1.015340e+00
Person_Blood_Alcohol_ContentTest	727642.230612	4.410245e+07	5.664874e+06
Vehicle_Color_BLACK	0.351987	1.229846e+00	6.579434e-01
Vehicle_Color_BLUE	0.480887	1.735236e+00	9.134838e-01
Vehicle_Color_BROWN	0.145590	8.715789e-01	3.562208e-01
Vehicle_Color_GOLD	0.339859	1.445399e+00	7.008791e-01
Vehicle_Color_GRAY	0.364643	1.301531e+00	6.889078e-01
Vehicle_Color_GREEN	0.490311	2.050029e+00	1.002573e+00
Vehicle_Color_MAROON	0.320718	1.450486e+00	6.820530e-01
Vehicle_Color_OTHER	0.423819	1.675749e+00	8.427419e-01
Vehicle_Color_RED	0.275026	1.017017e+00	5.288725e-01
Vehicle_Color_SILVER	0.330461	1.219664e+00	6.348629e-01
Vehicle_Color_WHITE	0.478014	1.635523e+00	8.841962e-01
Vehicle_Color_YELLOW	0.138412	9.361116e-01	3.599567e-01
Person_Ethnicity_ASIAN	0.612750	1.409524e+01	2.938853e+00
Person_Ethnicity_BLACK	0.928325	1.876591e+01	4.173830e+00
Person_Ethnicity_HISPANIC	0.670425	1.355013e+01	3.014025e+00
Person_Ethnicity_OTHER	0.548878	1.174088e+01	2.538565e+00
Person_Ethnicity_WHITE	0.955037	1.935116e+01	4.298962e+00
Person Gender MALE	0.908783	1.208185e+00	1.047845e+00

For the Person_Blood_Alcohol_ContentTest variable increased by 15.5498. It explained that there were many fatal accidents caused by people with positive blood alcohol content percentages. So, the probability of dying increased blood alcohol content in the drivers' blood steam.

Houston

We also ran logistic regression on Houston dataset and have provided the output below. The features that were highly significant include Speed_Limit, Person_Age, Person_Blood_Alcohol_ContentTest, Vehicle_Color_Black, Vehicle_Color_Gold, Vehicle_Color_Green, Vehicle_Color_Red, Vehicle_Color_White, Vehicle_Color_Yellow and Person_Ethinicties. It is interesting to see that these Vehicle Color features have a positive relationship with the dependent variable, Death_YorN. In other words, we can say that if the color of the vehicle is one of these colors listed here, the probability of a fatal crash increases.

Logit Regression Results

Dep. Variable:	Death_YorN	No. Observations:	4867
Model:	Logit	Df Residuals:	4843
Method:	MLE	Df Model:	23
Date:	Tue, 24 Nov 2020	Pseudo R-squ.:	inf
Time:	18:07:32	Log-Likelihood:	-1.6000e+05
converged:	True	LL-Null:	0.0000
Covariance Type:	nonrobust	LLR p-value:	1.000

	coef	std err	z	P>|z|	[0.025	0.975]
Crash_Year	-0.0681	0.036	-1.872	0.061	-0.139	0.003
Speed_Limit	-0.0078	0.003	-2.906	0.004	-0.013	-0.003
Weather	0.0212	0.069	0.308	0.758	-0.114	0.156
Timeofday	0.0092	0.065	0.141	0.888	-0.119	0.137
Person_Age	0.0142	0.002	7.484	0.000	0.011	0.018
Person_Blood_Alcohol_ContentTest	8.4045	0.649	12.959	0.000	7.133	9.676
Vehicle_Color_BLACK	1.1643	0.317	3.678	0.000	0.544	1.785
Vehicle_Color_BLUE	0.6847	0.329	2.083	0.037	0.040	1.329
Vehicle_Color_BROWN	0.6754	0.463	1.458	0.145	-0.233	1.583
Vehicle_Color_GOLD	1.0879	0.381	2.852	0.004	0.340	1.836
Vehicle_Color_GRAY	0.5488	0.324	1.696	0.090	-0.085	1.183
Vehicle_Color_GREEN	1.3625	0.394	3.456	0.001	0.590	2.135
Vehicle_Color_MAROON	0.7988	0.363	2.202	0.028	0.088	1.510
Vehicle_Color_OTHER	0.2006	0.367	0.546	0.585	-0.520	0.921
Vehicle_Color_RED	0.9857	0.328	3.008	0.003	0.343	1.628
Vehicle_Color_SILVER	0.6339	0.322	1.967	0.049	0.002	1.266
Vehicle_Color_WHITE	1.4958	0.314	4.765	0.000	0.881	2.111
Vehicle_Color_YELLOW	0.7750	0.415	1.870	0.062	-0.037	1.587
Person_Ethnicity_ASIAN	136.3253	73.444	1.856	0.063	-7.622	280.272
Person_Ethnicity_BLACK	135.9538	73.437	1.851	0.064	-7.979	279.887
Person_Ethnicity_HISPANIC	135.9791	73.438	1.852	0.064	-7.956	279.914
Person_Ethnicity_OTHER	136.8466	73.437	1.863	0.062	-7.087	280.780
Person_Ethnicity_WHITE	135.8860	73.437	1.850	0.064	-8.049	279.821
Person_Gender_MALE	0.1789	0.066	2.711	0.007	0.050	0.308

(Figure 6 - Logistic Regression)

Odds Ratio Estimate – Houston

```
estimates = log_reg_h.params
intervals = log_reg_h.conf_int()
intervals['Odds Ratio'] = estimates
intervals.columns = ['5%', '95%', 'Odds Ratio']
print(np.exp(intervals))
```

	5%	95%	Odds Ratio
Crash_Year	0.869878	1.003222e+00	9.341737e-01
Speed_Limit	0.986990	9.974581e-01	9.922103e-01
Weather	0.892467	1.168989e+00	1.021413e+00
Timeofday	0.887773	1.147335e+00	1.009244e+00
Person_Age	1.010568	1.018136e+00	1.014345e+00
Person_Blood_Alcohol_ContentTest	1253.101734	1.592338e+04	4.466947e+03
Vehicle_Color_BLACK	1.722677	5.958329e+00	3.203791e+00
Vehicle_Color_BLUE	1.041295	3.777256e+00	1.983239e+00
Vehicle_Color_BROWN	0.792463	4.871651e+00	1.964841e+00
Vehicle_Color_GOLD	1.405355	6.268371e+00	2.968044e+00
Vehicle_Color_GRAY	0.918177	3.263992e+00	1.731162e+00
Vehicle_Color_GREEN	1.803445	8.459366e+00	3.905893e+00
Vehicle_Color_MAROON	1.091642	4.525909e+00	2.222763e+00
Vehicle_Color_OTHER	0.594767	2.511350e+00	1.222157e+00
Vehicle_Color_RED	1.409806	5.093455e+00	2.679699e+00
Vehicle_Color_SILVER	1.002270	3.545154e+00	1.884994e+00
Vehicle_Color_WHITE	2.412265	8.257030e+00	4.462975e+00
Vehicle_Color_YELLOW	0.963211	4.891128e+00	2.170527e+00
Person_Ethnicity_ASIAN	0.000490	5.257062e+121	1.604441e+59
Person_Ethnicity_BLACK	0.000343	3.575173e+121	1.106581e+59
Person_Ethnicity_HISPANIC	0.000351	3.674470e+121	1.134952e+59
Person_Ethnicity_OTHER	0.000836	8.736159e+121	2.702134e+59
Person_Ethnicity_WHITE	0.000320	3.346326e+121	1.034057e+59
Person_Gender_MALE	1.050798	1.361156e+00	1.195951e+00

The coefficient of Person_Ethnicity variable is a high number. The probability of dying explained many deaths was caused by depending on the ethnicity of each group.

IX. Model Evaluation

Linear Regression: RMSE - Dallas vs Houston

```
### Calculate MSE
#DALLAS
from math import sqrt
from sklearn.metrics import mean_squared_error
RMSElinear_d=sqrt(mean_squared_error(y_true=y_test_d,y_pred=y_prediction_d))
print("Dallas RMSE:", RMSElinear_d)
```

```
Dallas RMSE: 0.544959043405608
```

```
RMSElinear_h=sqrt(mean_squared_error(y_true=y_test_h,y_pred=y_prediction_h))
print("Houston RMSE:", RMSElinear_h)
```

```
Houston RMSE: 0.5836733260075451
```

Lower values of RMSE indicate better fit. It was very consistent with the R-square result in the two Linear Regression tables of Dallas and Houston, as the RMSE of the Dallas dataset is lower than the RMSE of the Houston dataset.

Linear Regression: K-fold Cross-Validation

```
from sklearn.model_selection import cross_val_score
scores_d = cross_val_score(regressor, X_train_d, y_train_d, scoring="neg_mean_squared_error", cv=5)
d_rmse_scores = np.sqrt(-scores_d)
```

Output:

```
********DALLAS********
Scores: [0.57234609 0.55854014 0.54021685 0.54104613 0.55503445]
Mean: 0.5534367302317309
Standard deviation: 0.011953688545891846
```

```
*********HOUSTON********
Scores: [0.56383655 0.56936412 0.54915874 0.56914135 0.5648962 ]
Mean: 0.5632793926641184
Standard deviation: 0.007398698349365959
```

We used the following process to employ k-fold cross-validation:

- Randomly split the training set into 5 distinct subsets called folds.
- Train and evaluate the regression model 5 times, picking a different fold for evaluation every time and training on the other 4 folds.
- The result is consistent with the total RMSE computed previously

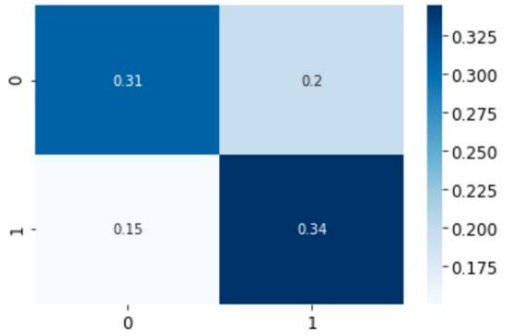
Logistic Regression: Accuracy and Confusion Matrix

DALLAS: Accuracy & Confusion Matrix

```
Confusion Matrix :

array([[372, 162],
       [212, 333]], dtype=int64)

Test accuracy =  0.6311399443929564
```

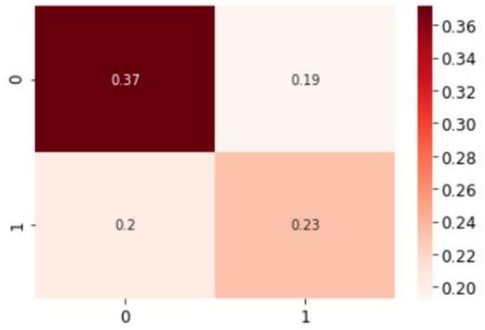


HOUSTON: Accuracy & Confusion Matrix

```
Confusion Matrix :

array([[284, 246],
       [235, 452]], dtype=int64)

Test accuracy =  0.5883319638455218
```



(Figure 7 - Accuracy & Confusion Matrix)

The two tables above showed the model evaluation for logistic regression and observation the accuracy and confusion matrix for Dallas and Houston. The test accuracy for the model in Dallas was 0.63 overall much better the result for over Houston, which is accuracy and confusion matrix is 0.58.

1. Summary Analysis

Our initial regression on the Dallas and Houston datasets produced similar results that the variables Person_Blood_Alcohol_ContenTest, Speed_Limit, Vehicle_Color, and Person_Age are significant (at alpha 5%) in explaining the reason behind fatal accidents in Texas. The prediction and evaluation model provide a snapshot of alcohol-involved deaths and drunk driving. The information can help local public health decisionmakers and community partners see gaps and identify relevant strategies to address the problem of drunk driving. Crashes, injuries and fatalities caused by drunk drivers continue to be the major traffic safety problem in Dallas and Houston. Therefore, fatalities related to alcohol are increasing and Texas should again experience a decrease in alcohol-related fatalities this year. We used the linear regression prediction model, and modified them for our machine learning and regression exploration targeting the Binary variable Death_YorNo (coded as 1 or 0) and dependent variable Unit_Death_Count and applied logistic

regression, Odds Ratio Estimate, RMSE and K-fold Cross-Validation on the data, and harvested more insights from the data. The analysis and model results showed that both the linear regression and logistic regression models for the Dallas dataset is better than the models for Houston dataset in terms of R-Square, RMSE, Accuracy and Confusion Matrix.

X. Conclusion

Traffic crashes are widely considered the leading cause of human in. There is no guaranteed way to avoid car crashes. However, more accidents happen at certain times, and in certain places. Starting with collecting all Houston and Dallas crash data from the Crash Records Information System managed by the Texas Department of Transportation (TxDOT), we identified often used by previous research on fatal accident attributes. We can explain factors that contribute to the probability of fatal accidents. Primary factors that affect the probability of fatal accidents are positive Blood Alcohol Content, male drivers, color of cars and ethnicity. Common causes of car accidents include drivers driving under the blood alcohol, speeding, weather, gender, age and race. Driving under the influence of alcohol accounts for about one-third of car accident fatalities. Drivers who are speeding have less control over their cars and less time to respond to hazards.

XI. Appendix

Figure 1: Import data

```
#import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import math

import os

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

from scipy import stats
from scipy.stats import kurtosis, skew

%matplotlib inline
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt
import matplotlib as mpl


from statsmodels.stats import diagnostic as diag
from statsmodels.stats.outliers_influence import variance_inflation_factor
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)
PROJECT_ROOT_DIR = "."
CHAPTER_ID = "end_to_end_project"
PROJECT_ROOT_DIR = "."
IMAGES_PATH = os.path.join(PROJECT_ROOT_DIR, "images", CHAPTER_ID)
os.makedirs(IMAGES_PATH, exist_ok=True)

def save_fig(fig_id, tight_layout=True, fig_extension="png", resolution=300):
    path = os.path.join(IMAGES_PATH, fig_id + "." + fig_extension)
    print("Saving figure", fig_id)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format=fig_extension, dpi=resolution)
```

Figure 2 - Remove substring from string

```
In [65]: #Deleting "Initial - " before the ethnicity
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.lstrip('B')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.lstrip('H')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.lstrip('W')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.lstrip('I')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.lstrip('A')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.lstrip('98')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.strip(' ')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.lstrip('-')
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].str.strip(' ')

In [66]: #Deleting "1 -" and "2 -" before Gender
         df2['Person_Gender'] = df2['Person_Gender'].str.lstrip('1')
         df2['Person_Gender'] = df2['Person_Gender'].str.lstrip('2')
         df2['Person_Gender'] = df2['Person_Gender'].str.strip(' ')
         df2['Person_Gender'] = df2['Person_Gender'].str.lstrip('-')
         df2['Person_Gender'] = df2['Person_Gender'].str.strip(' ')
```

Figure 3 - Changing the data type of categorical variables

```
In [69]: #Changing the data type of categorical variables
         df2['City'] = df2.City.astype('category')
         df2['Vehicle_Color'] = df2.Vehicle_Color.astype('category')
         df2['Person_Ethnicity'] = df2.Person_Ethnicity.astype('category')
         df2['Person_Gender'] = df2.Person_Gender.astype('category')
         df2['Weather'] = df2.Weather.astype('category')
         df2['Timeofday'] = df2.Timeofday.astype('category')
         df2['Death_YorN'] = df2.Death_YorN.astype('category')

In [70]: df2['Speed_Limit'] = df2['Speed_Limit'].fillna(df2['Speed_Limit'].mean())
         df2['Vehicle_Color'] = df2['Vehicle_Color'].fillna(df2['Vehicle_Color'].value_counts().index[0])
         df2['Person_Ethnicity'] = df2['Person_Ethnicity'].fillna(df2['Person_Ethnicity'].value_counts().index[0])
```

Figure 4 - Data Exploration

```
import seaborn as sns
sns.factorplot(x='Crash_Year',data=Dallas,kind='count',size=5,aspect=1.1)
print("                        Dallas Dataset")
```

```
sns.factorplot(x='Crash_Year',data=Houston,kind='count',size=5,aspect=1.1)
print("                        Houston Dataset")
```

```
sns.factorplot(x='Person_Ethnicity',data=Dallas,kind='count',size=5,aspect=2)
print("                            Dallas Dataset")
```

```
sns.factorplot(x='Person_Ethnicity',data=Houston,kind='count',size=5,aspect=2)
print("                            Houston Dataset")
```

Figure 5 - Linear Regression

DALLAS DATASET

In [85]:
```
# DALLAS DATASET
## fit linear model on training dataset
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(X_train_d, y_train_d)

## predicting with the test dataset
y_prediction_d=regressor.predict(X_test_d)
y_prediction_d
y_prediction_d.shape
```

Out[85]:
```
LinearRegression()
```

Out[85]:
```
array([0.52254191, 0.38669586, 0.39413357, ..., 0.73090439, 0.77014601,
       0.47801741])
```

Out[85]:
```
(1079,)
```

In [86]:
```
# DALLAS REGRESSION SUMMARY
# For better interpretation of linear regression model, consider statsmodels package
import statsmodels.api as sm

# Add a constant
X_train_d = sm.add_constant(X_train_d)

# Fit and summarize OLS model
mod = sm.OLS(y_train_d, X_train_d)
res = mod.fit()
print(res.summary())
```

HOUSTON DATASET

In [87]:
```
## predicting with the test dataset
y_prediction_h=regressor.predict(X_test_h)
y_prediction_h
y_prediction_h.shape
```

Out[87]:
```
array([0.6470531 , 1.1594512 , 0.48702579, ..., 0.59118217, 0.61173826,
       0.6620674 ])
```

Out[87]:
```
(1217,)
```

In [88]:
```
# HOUSTON REGRESSION SUMMARY
# For better interpretation of linear regression model, consider statsmodels package
import statsmodels.api as sm

# Add a constant
X_train_h = sm.add_constant(X_train_h)

# Fit and summarize OLS model
mod = sm.OLS(y_train_h, X_train_h)
res = mod.fit()
print(res.summary())
```

Figure 6 - Logistic Regression

DALLAS DATASET

In [97]:
```
log_reg_d = sm.Logit(y_train_dl, X_train_dl).fit()
```

```
Optimization terminated successfully.
         Current function value: 40.889929
         Iterations 7
```

In [98]:
```
print(log_reg_d.summary())
```

HOUSTON DATASET

In [100]:
```
log_reg_h = sm.Logit(y_train_hl, X_train_hl).fit()
```

```
Optimization terminated successfully.
         Current function value: 32.874581
         Iterations 7
```

In [101]:
```
print(log_reg_h.summary())
```

Figure 7 - Accuracy & Confusion Matrix

DALLAS: Accuracy & Confusion Matrix

In [103]:
```
# Testing the accuracy of the model
from sklearn.metrics import (confusion_matrix, accuracy_score)

# confusion matrix
print ("Confusion Matrix :")
confusion_matrix(y_true=y_test_dl, y_pred=prediction_d)

# accuracy score of the model
print('Test accuracy = ', accuracy_score(y_test_d, prediction_d))
```

```
Confusion Matrix :
```

Out[103]:
```
array([[372, 162],
       [212, 333]], dtype=int64)
```

```
Test accuracy =  0.6311399443929564
```

In [104]:
```
# label the confusion matrix
labels = [1,0]
cm_d = confusion_matrix(y_true=y_test_dl, y_pred=prediction_d, labels=labels)
```

In [105]:
```
# To plot it using seaborn package
sns.heatmap(cm_d/np.sum(cm_d), annot=True, cmap='Blues')
```

HOUSTON: Accuracy & Confusion Matrix

In [106]:
```
# confusion matrix
print ("Confusion Matrix :")
confusion_matrix(y_true=y_test_hl, y_pred=prediction_h)

# accuracy score of the model
print('Test accuracy = ', accuracy_score(y_test_h, prediction_h))
```

```
Confusion Matrix :
```

Out[106]:
```
array([[284, 246],
       [235, 452]], dtype=int64)
```

```
Test accuracy =  0.5883319638455218
```

In [107]:
```
# label the confusion matrix
labels = [1,0]
cm_h = confusion_matrix(y_true=y_test_hl, y_pred=prediction_h, labels=labels)
```

In [108]:
```
# To plot it using seaborn package
sns.heatmap(cm_h/np.sum(cm_h), annot=True, cmap='Reds')
```

Out[108]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2425d117280>
```

XII. References

National highway traffic safety administration (NHTSA) – URL: https://www.nhtsa.gov/risky-driving/drunk-driving

TxDOT Crash Query Tool – URL: https://cris.dot.state.tx.us/public/Query/app/welcome

Texas Department of Transportation Website - URL: https://www.txdot.gov/