Vehicles' accident rate prediction

based on simulated data and real data with weighted function

팀 프리라이더

강수미(팀장), 강혜진, 박지훈, 이관호





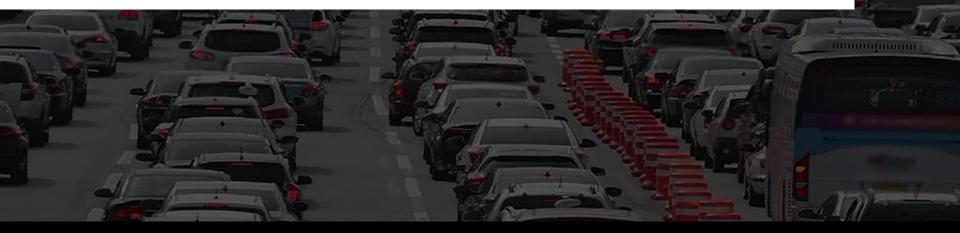


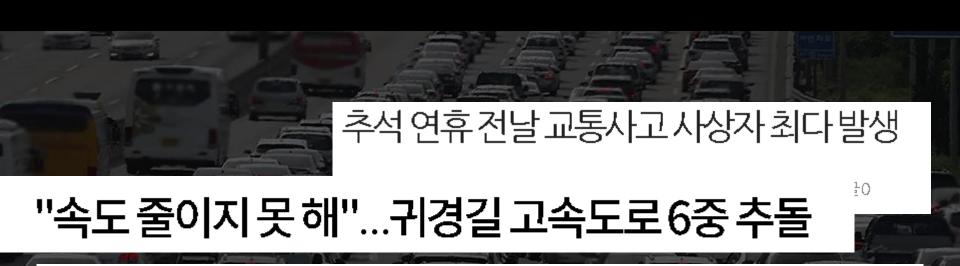




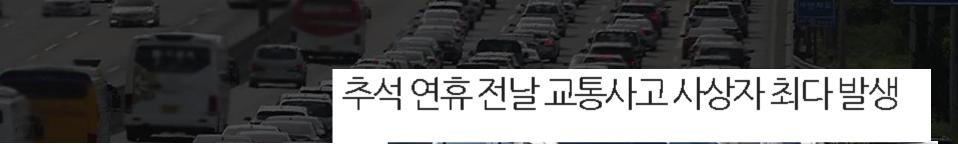


"속도줄이지못해"...귀경길고속도로6중추돌





하루 807건 '쾅'…추석 연휴 전날 교통사고 발생 가장 많다

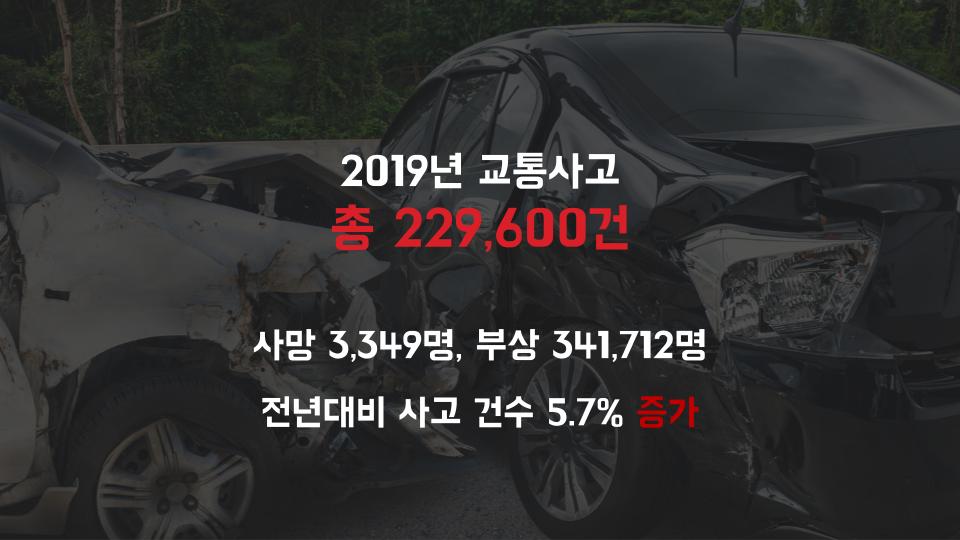


''속도줄이지 못해''...

하루 807건 '쾅'…추석







목차

- ✓ 프로젝트 주제 선정 배경
- ✓ 주제 소개
- ✓ 데이터 수집
- ✓ 데이터 전처리
- ✓ 목적 함수 설계
- ✓ 기대효과







※ 보유율 = 운전면허소지자 / 총인구 * 100

주제선정 배경

교통사고 발생율 증가

늘어나는 자동차

운전면허 소지자 증가

빅데이터 시대

데이터 분석을 통한

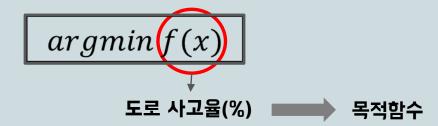
사고 요인 상관 분석

그리고

교통사고 발생 예측률 도출

주제(문제 정의)

교통사고 발생률을 나타내는 함수 도출



목적함수 도출을 위한 프로세스

Accident Rate =
$$f(x_i, w_j), i = 1 \cdots n, j = 1 \cdots n$$



기존 오픈데이터의 한계

[도로교통공단 통합 DB]

구분	합계	01월	02월
사고건수	1,292,864	100,132	87,308
사망자수	3,349	296	203
부상자수	2,053,971	155,811	144,198

- 모든 DB가 사고 관련 통계데이터로 구성. 하지만, 실제 사고 데이터 존재 x

[도로교통공단 위험도로 예측 시스템]



-도로 유형별 가중치로 위험도 예측 가능. 하지만, 각 요소 별 가중치가 공개되어있지 않음.

[교통데이터 거래소 오픈 데이터]



-한국 도로공사 제공 데이터는 고속도로 이동 관련 데이터 위주.

데이터 수집 방향

- 집계 위주의 데이터
- 교통사고 요인 확인 불가로 상관 분석에 어려움



한국 도로공사에 실제 사고 데이터 제공 요청

시카고 사고 데이터, 가상데이터 활용

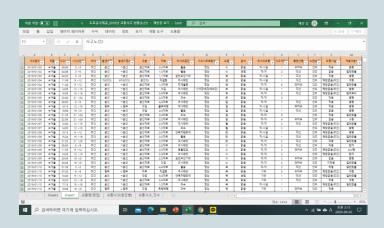
실제 데이터



대한민국 2019년 고속도로 사고 데이터

(출처 : 한국도로공사)

총 27개의 칼럼



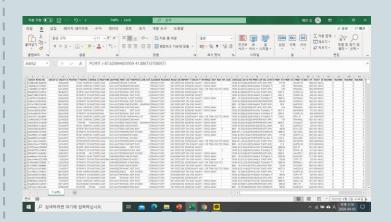


시카고 사고 데이터

(출처 : 구글 오픈데이터)

https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3if

총 49개의 칼럼



가상 데이터

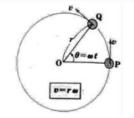
실제 데이터의 한계 (교통사고 발생 당시의 속도, 회전각에 대한 정보 부족)

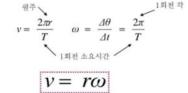


난수와 구심력 공식을 활용한 시뮬레이션 사고 데이터 생성









$$F_c = \frac{mv^2}{r} = mr\omega^2 = mr\frac{4\pi^2}{T^2}$$
 ਦੇੀ:[N]

F_e: 구심력 [N]
m : 추의 질량 [Kg]
v : 선속도 [m/sec]
v : 각속도 [rad/sec]
r : 회전 궤도 반경 [m]
T : 주기 [sec]

가상 데이터

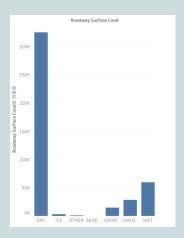
실제 데이터의 한계 (교통사고 발생 당시의 속도, 회전각에 대한 정보 부족)



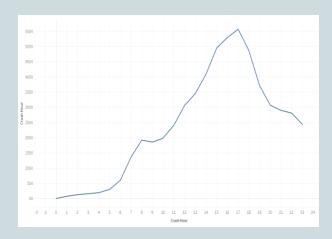
난수와 구심력 공식을 활용한 시뮬레이션 사고 데이터 생성

	Α	В	С	D
1	start	speed	angle	end
2	(0, 0)	39	0.750492	(-36.568, 34.1)
3	(0, 0)	54	0.663225	(-39.401, 30.783)
4	(0, 0)	56	0.698132	(-38.302, 32.139)
5	(0, 0)	37	0.750492	(-36.568, 34.1)
6	(0, 0)	31	0.680678	(-38.857, 31.466)
7	(0, 0)	40	0.663225	(-39.401, 30.783)
8	(0, 0)	39	0.698132	(-38.302, 32.139)
9	(0, 0)	46	0.663225	(-39.401, 30.783)
10	(0, 0)	64	0.680678	(-38.857, 31.466)
11	(0, 0)	44	0.715585	(-37.735, 32.803)
12	(0, 0)	43	0.645772	(-39.932, 30.091)
13	(0, 0)	79	0.767945	(-35.967, 34.733)
14	(0, 0)	71	0.750492	(-36.568, 34.1)
15	(0, 0)	63	0.767945	(-35.967, 34.733)
16	(0, 0)	43	0.767945	(-35.967, 34.733)
17	(0, 0)	55	0.663225	(-39.401, 30.783)
18	(0, 0)	56	0.698132	(-38.302, 32.139)
19	(0, 0)	31	0.680678	(-38.857, 31.466)
4)	si	mulationData	(+)

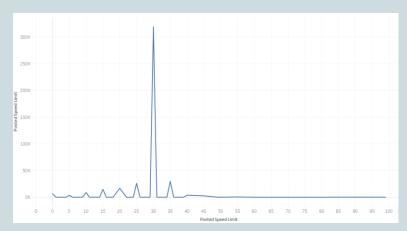
Key Xs 도출을 위한 EDA



Roadway Surface Condition



Crash Hour



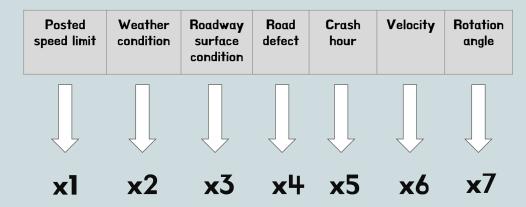
Post Speed Limit

Accident Rate = $f(x_i, w_j), i = 1 \cdots n, j = 1 \cdots n$

수집 데이터



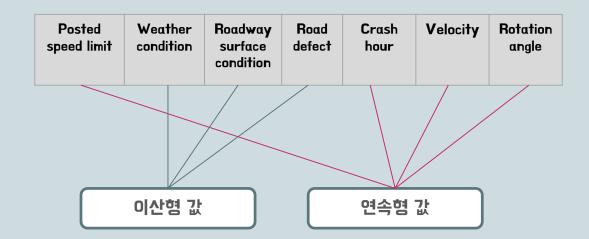
Accident Rate에 유의미하게 관여하는 attributes



Data Transformation

	CRASH_RECORD_ID	RD_NO	CRASH_DATE_EST_I	CRASH_DATE	POSTED_SPEED_LIMIT	TRAFFIC_CONTROL_DE
0	073682ef84ff827659552d4254ad1b98bfec24935cc9cc	JB460108	NaN	10/02/2018 06:30:00 PM	10	NO CONTR
1	1560fb8a1e32b528fef8bfd677d2b3fc5ab37278b157fa	JC325941	NaN	06/27/2019 04:00:00 PM	45	NO CONTR
2	009e9e67203442370272e1a13d6ee51a4155dac65e583d	JA329216	NaN	06/30/2017 04:00:00 PM	35	STOP SIGN/FLAS
3	00e47f189660cd8ba1e85fc63061bf1d8465184393f134	JC194776	NaN	03/21/2019 10:50:00 PM	30	TRAFFIC SIG
4	0126747fc9ffc0edc9a38abb83d80034f897db0f739eef	JB200478	NaN	03/26/2018 02:23:00 PM	35	NO CONTR
433054	e481dd27bc76e18b9a627f605d6d62c5750bca3311222a	JD313387	NaN	07/28/2020 04:00:00 PM	30	NO CONTR
433055	df74080be8ceeb327edd744a7f34507bc32c3f76a3f00c	JD314102	NaN	07/29/2020 02:00:00 PM	30	NO CONTR
433056	f2ad41aeabd38b5c073f887c66ba64118d1c8843395d5c	JD313908	NaN	07/29/2020 10:50:00 AM	30	TRAFFIC SIG
433057	fcc0426860ee85a4e6db60be5572d317452eb63506d0df	JD314407	NaN	07/29/2020 06:30:00 PM	30	NO CONTR
433058	f829a8bd9a1afc120ed6720cdb20a22b1a90b81278002b	JD105771	NaN	01/06/2020 08:30:00 AM	15	NO CONTR
433059	rows × 49 columns					

	POSTED_SPEED_LIMIT	TRAFFIC_CONTROL_DEVICE	DEVICE_CONDITION	WEATHER_CONDITION	LIGHTING_CONDITION	LANE_CNT	ROADWAY_SU
0	10	1	1	1	3	NaN	
1	45	1	1	1	1	NaN	
2	35	3	2	1	1	4.0	
3	30	2	2	1	2	4.0	
4	35	1	1	1	1	NaN	
433054	30	1	1	1	1	NaN	
433055	30	1	1	1	1	NaN	
433056	30	2	2	1	1	NaN	
433057	30	1	1	1	1	NaN	
433058	15	1	1	1	1	NaN	
433059	rows × 9 columns						



이산형 데이터 전처리

	WEATHER_CONDITION	weight
0	CLEAR	0.83
1	RAIN	0.09
2	SNOW	0.04
3	CLOUDY/OVERCAST	0.03
4	OTHER	0.01

문자를 숫자(랭크)로 치환 랭크에 대한 빈도수를 가중치로 부여

연속형 데이터 전처리

	SPEED	weight
0	30~40	0.1950
1	40~50	0.2054
2	50~60	0.1850
3	60~70	0.2046
4	70~	0.2100

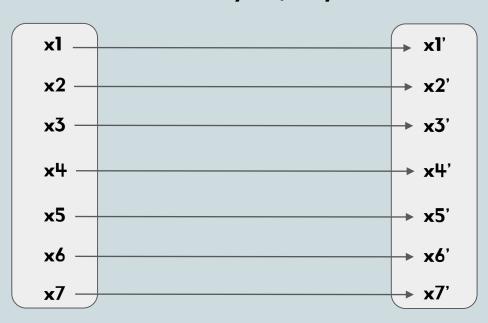
연속형 숫자를 그룹화 그룹화된 랭크에 대한 빈도수를 가중치로 부여

Key Xs 전처리 프로세스 - 가중치 부여

	POSTED_SPEED_LIMIT	TRAFFIC_CONTROL_DEVICE	DEVICE_CONDITION	WEATHER_CONDITION	LIGHTING_CONDITION	LANE_CNT	ROADWAY_SU
0	10	1	1	1	3	NaN	
1	45	1	1	1	1	NaN	
2	35	3	2	1	1	4.0	
3	30	2	2	1	2	4.0	
4	35	1	1	1	1	NaN	
433054	30	1	1	1	1	NaN	
433055	30	1	1	1	1	NaN	
433056	30	2	2	1	1	NaN	
433057	30	1	1	1	1	NaN	
433058	15	1	1	1	1	NaN	
433059	rows × 9 columns						

	POSTED_SPEED_LIMIT	WEATHER_CONDITION	ROADWAY_SURFACE_COND	ROAD_DEFECT	CRASH_HOUR	INJURIES_TOTAL
0	0.03	0.83	0.78	0.97	0.24	0.85
1	0.02	0.83	0.78	0.97	0.42	0.85
2	0.08	0.83	0.78	0.97	0.42	0.85
3	0.80	0.83	0.78	0.97	0.24	0.85
4	0.08	0.83	0.78	0.97	0.42	0.85
433054	0.80	0.83	0.78	0.97	0.42	0.13
433055	0.80	0.83	0.78	0.97	0.42	0.85
433056	0.80	0.83	0.78	0.97	0.26	0.13
433057	0.80	0.83	0.78	0.97	0.24	0.85
433058	0.07	0.83	0.78	0.97	0.26	0.85
357558	rows × 6 columns					

Transform (By frequency weight)



Accident Rate = $f(x_i, w_j)$, $i = 1 \cdots n$, $j = 1 \cdots n$



델파이 기법으로 weight 도출

최종 사고율 (Accident Rate)

1. 각 사고 요인별 Weight값은 델파이기법을 이용해 설정한다.

$$w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 = 1$$

2. 총 1000가지의 임의의 요인별 조합을 샘섬 후 앞에서 찾은 각 요인별 빈도수 가중치로 치환

(ex)
$$(x_1, x_2, x_3, x_4, x_5, x_6, x_7) \xrightarrow{\text{transform}} (x'_1, x'_2, x'_3, x'_4, x'_5, x'_6, x'_7)$$

Accident Rate =
$$w_1 * x'_1 + w_2 * x'_2 + w_3 * x'_3 + w_4 * x'_4 + w_5 * x'_5 + w_6 * x'_6 + w_7 * x'_7$$

최종 사고율 (Accident Rate)

	POSTED_SPEED_LIMIT	WEATHER_CONDITION	ROADWAY_SURFACE_COND	ROAD_DEFEC1
0	0.03	0.04	0.01	0.97
1	0.03	0.03	0.78	0.97
2	0.07	0.09	0.15	0.97
3	0.03	0.03	0.01	0.97
4	0.80	0.04	0.01	0.03
95	0.02	0.01	0.02	0.03
96	0.03	0.01	0.15	0.97
97	0.80	0.04	0.04	0.97
98	0.07	0.03	0.78	0.97
99	0.02	0.03	0.02	0.03
00	rows × 7 columns			

fx1 0 0.171779 1 0.250829 2 0.286796 3 0.192904 4 0.279012 995 0.274546 996 0.492837 997 0.123471 998 0.325671 999 0.259521

Input data

result

그렇다면..?

기대효과

추후 사고 발생률 함수를 발전시켜서,

자율주행 시스템의 의사결정을 돕고 <u>완성도</u>에 기여할 수 있는 가능성 기대



기대효과

추후 사고 발생률 함수를 발전시켜서.

사고를 발생시키는 원인자들의 특정 조합을 도출해내 획기적으로 사고 발생 상황을 피하는 알고리즘 설계에 도움



THANK YOU

Q&A