작성일자 : 2022/06/19

지능정보공학 과제
: 정규화/카테고리 변환

휴먼과 202210829 김진석

목차

1.1 제목

Pandas의 dataframe 재구성

1.2 해결 방법

데이터의 구성을 보았을 때 처음 생각난 아이디어는 인덱싱 이었다. 하지만 인덱싱으로는 주간 야간을 합친 발생건수를 출력할 수 없었다. 두 번째 생각해낸 방법은 그룹화였다. '법규위반'이 주간 야간에 따라 두 번 출력되기 때문에 '법규위반'으로 그룹화를 하고 sum 함수를 사용하여 주간과 야간의 합을 구할 수 있었다. 그리고 dataframe의 열을 '발생건수', '사망자수', '부상자수'로 재구성해 주간과 야간의 합을 구할 수 있었다.

1.3 결과

	발생건수	사망자수	부상자수
법규위반			
과속	377	107	682
교차로 통행방법 위반	14721	111	23759
기타	15461	197	23288
보행자 보호의무 위반	7106	174	7449
신호위반	25307	389	42120
안전거리 미확보	22275	97	39814
안전운전 의무 불이행	125391	3872	184018
중앙선 침범	13018	445	23435

1.4 결론

dataframe의 그룹화 하는 방법을 배웠고, 열을 재구성하는 방법으로 dataframe을 새로 구성할 수 있었다. 자기가 원하는 열을 기준으로 그룹화를 통해 다양한 결과를 도출 할 수 있었다.

2.1 제목

Simple Feature Scaling , Min-Max 방식으로 발생건수/사망자수/부상자수 정규화

2.2 해결 방법

교재에 첨부된 Scaling과 Min-Max 정규화 식을 '발생건수', '사망자수', '부상자수'에 식을 구했다. 그리고 위의 dataframe에 다시 재구성 하였다.

2.3 결과

	발생건수	사망자수	부상자수
법규위반			
과속	0.003007	0.027634	0.003706
교차로 통행방법 위반	0.117401	0.028667	0.129112
기타	0.123302	0.050878	0.126553
보행자 보호의무 위반	0.056671	0.044938	0.040480
신호위반	0.201825	0.100465	0.228891
안전거리 미확보	0.177644	0.025052	0.216359
안전운전 의무 불이행	1.000000	1.000000	1.000000
중앙선 침범	0.103819	0.114928	0.127352

<Scaling>

	발생건수	사망자수	부상자수
법규위반			
과속	0.000000	0.002649	0.000000
교차로 통행방법 위반	0.114739	0.003709	0.125873
기타	0.120658	0.026490	0.123304
보행자 보호의무 위반	0.053826	0.020397	0.036910
신호위반	0.199418	0.077351	0.226022
안전거리 미확보	0.175164	0.000000	0.213444
안전운전 의무 불이행	1.000000	1.000000	1.000000
중앙선 침범	0.101117	0.092185	0.124105

<Min-Max>

2.4 결론

일반적으로 지도학습 알고리즘을 적용하기 전에 데이터 전처리 과정이 꼭 필요하다는 것을 알았다. 또, 데이터 간의 값의 차이를 줄이기 위한 정규화 방법인 Simple Feature Scaling, Min-Max, Z-score에 대해 배웠다.

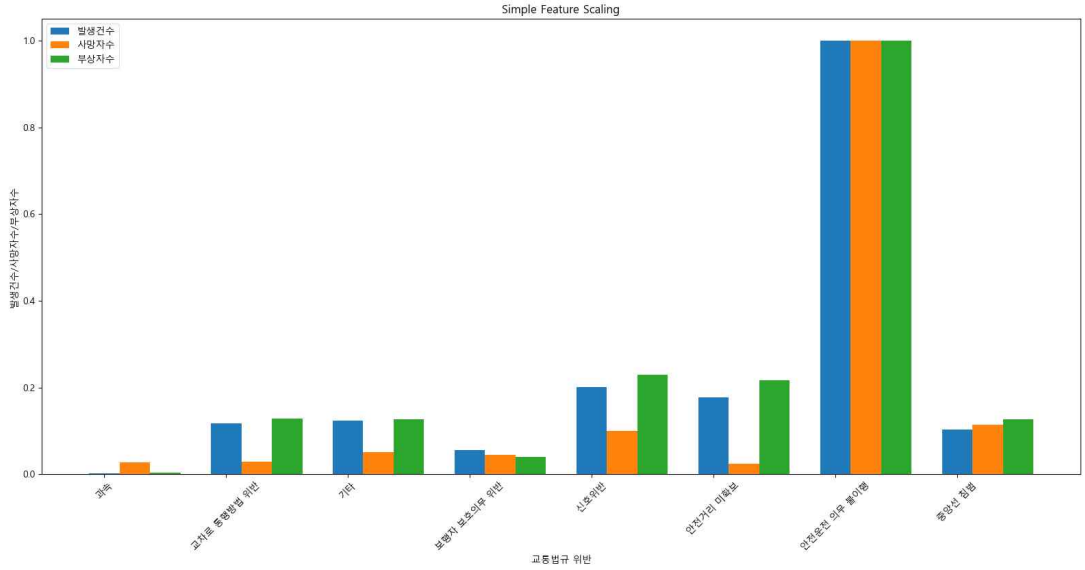
3.1 제목

matplotlib를 이용하여 정규화된 발생건수/사망자수/부상자수 bar 그래프로 시각화

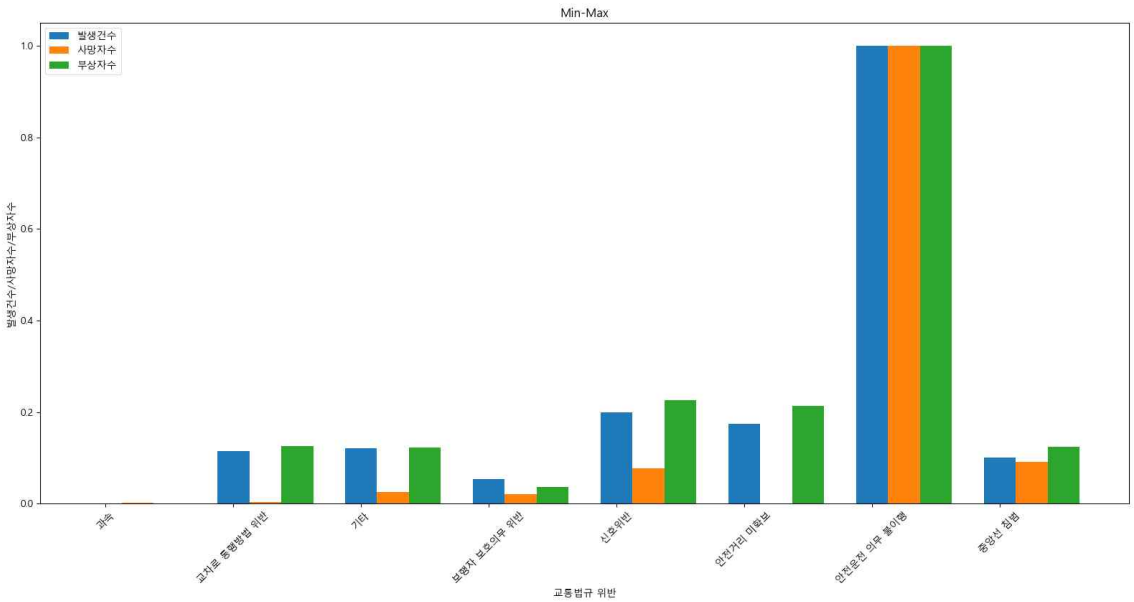
3.2 해결 방법

‘법규위반’의 값이 주간 야간으로 인해 두 번 나오기 때문에 unique() 메소드를 사용해 중복을 제거했다. 그리고 x축의 개수를 ‘법규위반’의 길이 만큼 구성했다. 그리고 3개의 데이터를 한 공간에 넣어야 하기 때문에 bar 함수를 3번 적어 ‘발생건수’, ‘사망자수’, ‘부상자수’ 별로 bar 그래프를 구성했다. 또한, 데이터들이 겹쳐지지 않게 범위를 각각 설정해 주었다. 그리고 범례를 왼쪽 위에 넣어 데이터와 겹쳐지지 않게 하였다.

3.3 결과



<Scaling>



<Min-Max>

3.4 결론

정규화 된 데이터를 시각화를 해보았다. 정규화하기 전에는 데이터의 값들의 차이가 커서 비교하기 힘들고 y축의 값을 설정하기 어려웠다. 하지만 정규화를 하고나서는 데이터의 비교가 더 쉬워진 거 같다. 하지만 아쉬운 점은 너무 작은 값은 데이터가 시각화 되지 않은 점이 아쉽다.

4.1 제목

발생건수/사망자수/부상자수에 대한 빈도를 [‘High’, ‘Mid’, ‘Low’]의 카테고리 값으로 변환한다.

4.2 해결 방법

교재에 있는 Data Formatting 방법을 참고 했다. cut() 함수는 새로운 열과 빈도수를 나누는 구간 값을 return 한다. pandas의 메소드인 cut()를 활용해 ‘발생건수’, ‘사망자수’, ‘부상자수’를 ‘Low’, ‘Mid’, ‘High’로 구간을 지정했다.

4.3 결과

```
                발생건수  사망자수    부상자수 발생건수_빈도수 사망자수_빈도수 부상자수_빈도수
법규위반
과속               377   107     682       Low       Low       Low
교차로 통행방법 위반   14721   111   23759       Low       Low       Low
기타              15461   197   23288       Low       Low       Low
보행자 보호의무 위반    7106   174    7449       Low       Low       Low
신호위반           25307   389   42120       Low       Low       Low
안전거리 미확보       22275    97   39814       Low       Low       Low
안전운전 의무 불이행  125391  3872  184018      High      High      High
중앙선 침범         13018   445   23435       Low       Low       Low
발생건수_빈도수 :  [   251.986      42048.33333333  83719.66666667 125391.       ]
사망자수_빈도수 :  [  93.225     1355.33333333 2613.66666667 3872.       ]
부상자수_빈도수 :  [   498.664  61794.    122906.    184018.   ]
```

4.4 결론

‘발생건수’, ‘사망자수’, ‘부상자수’의 빈도를 구하는 함수인 cut()에 대해 알았고 구간을 3개로 정했지만 데이터들의 값의 차이가 너무 커서 Low와 High로만 나뉜 거 같다. 데이터의 값의 차이가 더 적은 것으로 했으면 더 좋은 결과를 얻었을 거 같다.

5. 부록

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np



plt.rc('font', family='Malgun Gothic')

acci = pd.read_csv('acci1.csv', encoding='cp949')#euc-kr
```

<기본 설정>

```
# 1번 문제
x = pd.DataFrame(acci.groupby(acci['법규위반']).sum(), columns = ['발생건수','사망자수','부상자수'])
print(x)
```

<1번 문제>

```
# 2번 문제
#Simple Feature Scaling
x['발생건수'] =  x['발생건수'] / x['발생건수'].max()

x['사망자수'] =  x['사망자수'] / x['사망자수'].max()

x['부상자수'] =  x['부상자수'] / x['부상자수'].max()

print("================================================================================")
print(x)
```

<2-1번 문제>

```
#Min-Max

x['발생건수'] = (x['발생건수'] - x['발생건수'].min()) / (x['발생건수'].max() - x['발생건수'].min())
x['사망자수'] = (x['사망자수'] - x['사망자수'].min()) / (x['사망자수'].max() - x['사망자수'].min())
x['부상자수'] = (x['부상자수'] - x['부상자수'].min()) / (x['부상자수'].max() - x['부상자수'].min())
print("================================================================================")
print(x)
```

<2-2번 문제>

```
# 3-1 문제
nx  = acci['법규위반'].unique()
xs = np.arange(len(nx))
width = 0.25
ax=plt.axes()
plt.xticks(xs, nx, rotation = 45)
plt.xlabel('교통법규 위반')
plt.ylabel('발생건수/사망자수/부상자수')
plt.title('Simple Feature Scaling')
plt.bar(xs, x['발생건수'], width, label = '발생건수')
plt.bar(xs + width, x['사망자수'], width, label = '사망자수')
plt.bar(xs + 2*width, x['부상자수'], width, label = '부상자수')

plt.legend(loc="upper left")

plt.show()
```

<3-1번 문제>

```
nx  = acci['법규위반'].unique()
xs = np.arange(len(nx))
width = 0.25

plt.xticks(xs, nx, rotation = 45)
plt.xlabel('교통법규 위반')
plt.ylabel('발생건수/사망자수/부상자수')
plt.title('Min-Max')
plt.bar(xs, x['발생건수'], width, label = '발생건수')
plt.bar(xs + width, x['사망자수'], width, label = '사망자수')
plt.bar(xs + 2*width, x['부상자수'], width, label = '부상자수')

plt.legend(loc="upper left")

plt.show()
```

<3-2번 문제>

```
# 문제 4번

group_name = ['Low','Mid','High']
x['발생건수_빈도수'], mybin1 = pd.cut(x['발생건수'],3,labels = group_name, retbins = True)
x['사망자수_빈도수'], mybin2 = pd.cut(x['사망자수'],3,labels = group_name, retbins = True)
x['부상자수_빈도수'], mybin3 = pd.cut(x['부상자수'],3,labels = group_name, retbins = True)
print("=============================================================================")
print(x)
print("발생건수_빈도수 : ",mybin1)
print("사망자수_빈도수 : ", mybin2)
print("부상자수_빈도수 : ", mybin3)
```

<4번 문제>