titanic_cardinality

April 13, 2025

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
with open("./Zbiór danych Titanic.arff", "r") as f:
    headers = f.read()

columns = [header.split()[1].strip("'") for header in headers.split("\n") if␣
 ↪header.lower().startswith("@attribute")]
titanic_df = pd.read_csv('./Zbiór danych Titanic.arff', skiprows=17)
titanic_df.columns = columns
titanic_df = titanic_df.replace("?", np.nan)
```

0.1 1.

[3]:
```
cats = ['pclass', 'survived', 'name', 'sex', 'ticket', 'cabin', 'embarked',␣
 ↪'boat', 'home.dest']
cardinality = pd.DataFrame(columns=['len'])

for col in cats:
    cardinality.loc[col] = len(titanic_df[col].unique())

for col in cardinality.sort_values(by='len', ascending=False).index:
    print('Liczba etykiet zmiennej {}: {}'.format(col, cardinality.loc[col,␣
 ↪'len']))
```

```
Liczba etykiet zmiennej name: 1306
Liczba etykiet zmiennej ticket: 929
Liczba etykiet zmiennej home.dest: 370
Liczba etykiet zmiennej cabin: 187
Liczba etykiet zmiennej boat: 28
Liczba etykiet zmiennej embarked: 4
Liczba etykiet zmiennej pclass: 3
Liczba etykiet zmiennej sex: 2
Liczba etykiet zmiennej survived: 2
```

0.2 2.

[4]:
```
print("Liczba wszystkich pasażerów {}".format(len(titanic_df)))
```

```
Liczba wszystkich pasażerów 1308
```

0.3 3.

0.3.1 Mała moc zbioru

Zmienne `survived`, `sex` mają dwie różne wartości (takie dane reprezentują). Zmienna `embarked` ma 3 wartości, ponieważ Titanic odbijał od tylko trzech portów. Zmienna `boat` ma tylko 28 wartości, ponieważ uznano taką ilość za odpowiednią (nie było). Zmienna `pclass` ma 3 różne wartości, ponieważ przedstawia 3 klasy.

0.3.2 Duża moc zbioru

Zmienne `name`, `ticket`, `cabin`, `home.dest` mają dużą moc, gdyż są one różne dla prawie każdego pasażera.

0.4 4.

[5]:
```
len(titanic_df['cabin'].dropna().unique())
```

[5]:
```
186
```

0.5 5.

[6]:
```
titanic_df['CabinReduced'] = titanic_df['cabin'].dropna().astype(str).str[0]
titanic_df[['cabin', 'CabinReduced']].head(20)
```

[6]:

	cabin	CabinReduced
0	C22 C26	C
1	C22 C26	C
2	C22 C26	C
3	C22 C26	C
4	E12	E
5	D7	D
6	A36	A
7	C101	C
8	NaN	NaN
9	C62 C64	C
10	C62 C64	C
11	B35	B
12	NaN	NaN
13	A23	A
14	NaN	NaN
15	B58 B60	B
16	B58 B60	B
17	D15	D

```
18       C6            C
19      D35            D
```

0.6 6.

[7]:
```
reduced = []
for cat in ['cabin', 'CabinReduced']:
    reduced.append(len(titanic_df[cat].unique()))
    print(f'Liczba etykiet zmiennej {cat}: {len(titanic_df[cat].unique())}')

print('Kardynalność zredukowano o {}%'.format(np.round((reduced[0] -␣
 ↪reduced[1]) / reduced[0] * 100, 2)))
```

```
Liczba etykiet zmiennej cabin: 187
Liczba etykiet zmiennej CabinReduced: 9
Kardynalność zredukowano o 95.19%
```

0.7 7.

0.7.1 Dlaczego dokonuję redukcji akurat zmiennej `cabin`?

1) Zwiększa sie przejrzystość danych, gdyż `cabin` ma wysoką kardynalność (zamiast 187 etykiet zostaje nam 9)
2) Takie było polecenie nr 5, czyli mam obowiązek zrealizować to polecenie

Wpływ

- Zamiast kilkuset kabin mamy kilka kategorii, co zwiększa przejrzystość danych, dzięki czemu można się skupić an istotnych cechach

Negatywne skutki:

- Utrata szczegółowych informacji o kabinach, co może zmniejszyć dokładność, co może doprowadzić do zniknięcia drobnych wzorców (np. odległość kabiny od łodzi ratunkowej)