Import wymaganych pakietów

In [1]:
```
import numpy as np
import pandas as pd
```

Wczytanie pliku

In [2]:
```
df = pd.read_csv("Zbiór danych Titanic.arff.txt", header = 0, na_values = "?")
df.head(20)
```

Out[2]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
5	1	1	Anderson, Mr. Harry	male	48.0000	0	0	19952	26.5500	E12	S	3	NaN	New York, NY
6	1	1	Andrews, Miss. Kornelia Theodosia	female	63.0000	1	0	13502	77.9583	D7	S	10	NaN	Hudson, NY
7	1	0	Andrews, Mr. Thomas Jr	male	39.0000	0	0	112050	0.0000	A36	S	NaN	NaN	Belfast, NI
8	1	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53.0000	2	0	11769	51.4792	C101	S	D	NaN	Bayside, Queens, NY
9	1	0	Artagaveytia, Mr. Ramon	male	71.0000	0	0	PC 17609	49.5042	NaN	C	NaN	22.0	Montevideo, Uruguay
10	1	0	Astor, Col. John Jacob	male	47.0000	1	0	PC 17757	227.5250	C62 C64	C	NaN	124.0	New York, NY
11	1	1	Astor, Mrs. John Jacob (Madeleine Talmadge Force)	female	18.0000	1	0	PC 17757	227.5250	C62 C64	C	4	NaN	New York, NY
12	1	1	Aubart, Mme. Leontine Pauline	female	24.0000	0	0	PC 17477	69.3000	B35	C	9	NaN	Paris, France
13	1	1	Barber, Miss. Ellen 'Nellie'	female	26.0000	0	0	19877	78.8500	NaN	S	6	NaN	NaN
14	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0000	0	0	27042	30.0000	A23	S	B	NaN	Hessle, Yorks
15	1	0	Baumann, Mr. John D	male	NaN	0	0	PC 17318	25.9250	NaN	S	NaN	NaN	New York, NY
16	1	0	Baxter, Mr. Quigg Edmond	male	24.0000	0	1	PC 17558	247.5208	B58 B60	C	NaN	NaN	Montreal, PQ
17	1	1	Baxter, Mrs. James (Helene DeLaudeniere Chaput)	female	50.0000	0	1	PC 17558	247.5208	B58 B60	C	6	NaN	Montreal, PQ
18	1	1	Bazzani, Miss. Albina	female	32.0000	0	0	11813	76.2917	D15	C	8	NaN	NaN
19	1	0	Beattie, Mr. Thomson	male	36.0000	0	0	13050	75.2417	C6	C	A	NaN	Winnipeg, MN

In [28]:
```
names = ['name','sex', 'cabin', 'embarked', 'home.dest']
```

In [27]:
```
for name in names:
    print(f'Liczba etykiet zmiennej {name}: {len(df[name].unique())}')
```

```
Liczba etykiet zmiennej name: 1307
Liczba etykiet zmiennej sex: 2
Liczba etykiet zmiennej cabin: 187
Liczba etykiet zmiennej embarked: 4
Liczba etykiet zmiennej home.dest: 370
```

In [29]:
```
print(f"Ilość rekordów w tabeli: {len(df['name'])}")
```

```
Ilość rekordów w tabeli: 1309
```

Ilość unikalnych imion jest różna od ilości rekordów w tabeli. Sprawdzam czy nie występują duplikaty:

In [24]:
```
dup_names = df[df[['name']].duplicated()].name.values
```

In [25]:
```
df[df['name'] == dup_names[0]]
```

Out[25]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest	CabinRedu
725	3	1	Connolly, Miss. Kate	female	22.0	0	0	370373	7.7500	NaN	Q	13	NaN	Ireland	
726	3	0	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292	NaN	Q	NaN	NaN	Ireland	

In [23]:
```
df[df['name'] == dup_names[1]]
```

Out[23]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest	CabinReduced
924	3	0	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q	NaN	70.0	NaN	n
925	3	0	Kelly, Mr. James	male	44.0	0	0	363592	8.0500	NaN	S	NaN	NaN	NaN	n

Nie występują duplikaty - zduplikowane rekordy to osoby przypadkowo nazywające się tak samo. **Ilość pasażerów wynosi zatem 1309**.

Najciekawsze wyniki dały informacje o kardynalności zmiennych **cabin i home.dest**, w których możemy zauważyć dużą ilość kategorii (wysoką kardynalność), ale o wiele mniejszą od liczby pasażerów. Oznacza to, że możemy spróbować dokonać redukcji kardynalności tych cech i dzięki temu uogólnić rozważania (stworzyć model wartości przeciętnej i uniknąć overfittingu).

Cechy o małej kardynalności:

- sex
- embarked

Cechy o dużej kardynalności:

- cabin
- home.dest

Redukcja kardynalności zmiennej Cabin

In [7]:
```
df['cabin'].value_counts()
```

Out[7]:
```
cabin
C23 C25 C27        6
B57 B59 B63 B66    5
G6                 5
D                  4
F4                 4
                  ..
F E46              1
F E57              1
F E69              1
E10                1
F38                1
Name: count, Length: 186, dtype: int64
```

In [8]:
```
df['CabinReduced'] = df['cabin'].astype(str).str[0]
df[['cabin', 'CabinReduced']].head(20)
```

Out[8]:

	cabin	CabinReduced
0	B5	B
1	C22 C26	C
2	C22 C26	C
3	C22 C26	C
4	C22 C26	C
5	E12	E
6	D7	D
7	A36	A
8	C101	C
9	NaN	n
10	C62 C64	C
11	C62 C64	C
12	B35	B
13	NaN	n
14	A23	A
15	NaN	n
16	B58 B60	B
17	B58 B60	B
18	D15	D
19	C6	C

In [44]:
```
#unikalne wartości zmiennej cabin
df['cabin'].unique()
```

Out[44]:
```
array(['B5', 'C22 C26', 'E12', 'D7', 'A36', 'C101', nan, 'C62 C64', 'B35',
       'A23', 'B58 B60', 'D15', 'C6', 'D35', 'C148', 'C97', 'B49', 'C99',
       'C52', 'T', 'A31', 'C7', 'C103', 'D22', 'E33', 'A21', 'B10', 'B4',
       'E40', 'B38', 'E24', 'B51 B53 B55', 'B96 B98', 'C46', 'E31', 'E8',
       'B61', 'B77', 'A9', 'C89', 'A14', 'E58', 'E49', 'E52', 'E45',
       'B22', 'B26', 'C85', 'E17', 'B71', 'B20', 'A34', 'C86', 'A16',
       'A20', 'A18', 'C54', 'C45', 'D20', 'A29', 'C95', 'E25', 'C111',
       'C23 C25 C27', 'E36', 'D34', 'D40', 'B39', 'B41', 'B102', 'C123',
       'E63', 'C130', 'B86', 'C92', 'A5', 'C51', 'B42', 'C91', 'C125',
       'D10 D12', 'B82 B84', 'E50', 'D33', 'C83', 'B94', 'D49', 'D45',
       'B69', 'B11', 'E46', 'C39', 'B18', 'D11', 'C93', 'B28', 'C49',
       'B52 B54 B56', 'E60', 'C132', 'B37', 'D21', 'D19', 'C124', 'D17',
       'B101', 'D28', 'D6', 'D9', 'B80', 'C106', 'B79', 'C47', 'D30',
       'C90', 'E38', 'C78', 'C30', 'C118', 'D36', 'D48', 'D47', 'C105',
       'B36', 'B30', 'D43', 'B24', 'C2', 'C65', 'B73', 'C104', 'C110',
       'C50', 'B3', 'A24', 'A32', 'A11', 'A10', 'B57 B59 B63 B66', 'C28',
       'E44', 'A26', 'A6', 'A7', 'C31', 'A19', 'B45', 'E34', 'B78', 'B50',
       'C87', 'C116', 'C55 C57', 'D50', 'E68', 'E67', 'C126', 'C68',
       'C70', 'C53', 'B19', 'D46', 'D37', 'D26', 'C32', 'C80', 'C82',
       'C128', 'E39 E41', 'D', 'F4', 'D56', 'F33', 'E101', 'E77', 'F2',
       'D38', 'F', 'F G63', 'F E57', 'F E46', 'F G73', 'E121', 'F E69',
       'E10', 'G6', 'F38'], dtype=object)
```

In [10]:
```
print(f"Ilość etykiet zmiennej CabinReduced: {len(df['CabinReduced'].unique())}")
```

```
Ilość etykiet zmiennej CabinReduced: 9
```

In [11]:
```
print(f"Ilość etykiet względem zmiennej cabin zredukowano o {(187-9)/187*100:.2f}%")
```

```
Ilość etykiet względem zmiennej cabin zredukowano o 95.19%
```

In [12]:
```
df.groupby(["pclass", 'CabinReduced'])[['cabin']].count()
```

Out[12]:

pclass	CabinReduced	cabin
1	A	22
	B	65
	C	94
	D	40
	E	34
	T	1
	n	0
2	D	6
	E	4
	F	13
	n	0
3	E	3
	F	8
	G	5
	n	0

In [49]:
```
#wartości brakujące dla danych kabin:
df.groupby(['pclass', 'CabinReduced'])[['cabin']].apply(lambda x: x.isnull().sum())
```

Out[49]:

pclass	CabinReduced	cabin
1	A	0
	B	0
	C	0
	D	0
	E	0
	T	0
	n	67
2	D	0
	E	0
	F	0
	n	254
3	E	0
	F	0
	G	0
	n	693

Można zauważyć, że kabiny **A, B, C, D, E, T** są powiązane z przynależnością do **klasy pierwszej**

Kabiny **D, E, F** - do **klasy drugiej**

Kabiny **E, F, G** - do **klasy trzeciej**

Kabiny E występują w każdej klasie (najwięcej w pierwszej)

Kabiny D - w pierwszej i drugiej

Kabiny F - w druiej i trzeciej

Zauważamy też, że w klasie drugiej i trzeciej tylko niewiele osób miało przydzieloną kabinę.

Możemy dokonać redukcji akurat tej zmiennej, ponieważ literowe oznaczenie kabiny może nieść wystarczającą informację potrzebną dla utworzenia modelu i analizy danych (np. jest to typ kabiny i świadczy o statusie społecznym danego pasażera lub o jego rodzinie).

Zmiejszenie kardynalności danej cechy pozwoli nam też na oszczędzenie zasobów obliczeniowych i pamięciowych komputera (i np. przyspieszenie wykonywania zapytań).

Negatywne skutki jakie mogą wystąpić są związane z tym, że obcinając numery kabin być może jednak tracimy jakąś ważną informację (np. o ich położeniu).

In [ ]: