5. Домашнее задание по теме Pandas – продвинутый

Формулировка задания:

Необходимо загрузить и обработать предложенный датасет (Титаник), провести аналитику данных с помощью фреймворка Python Pandas.

Планируемый результат:

1. Датасет загружен в Colab
2. В датасете отсутствуют пустые ячейки
3. Представлена простая аналитика по датасету

Описания плана работы:

1. Создать новый ноутбук в Colab
2. Сохранить датасет в Google диске и загрузить в ноутбук Colab https://www.kaggle.com/datasets/yasserh/titanic-dataset/code

*Загрузить датасет по ссылке из интернета (репозитория Kaggle)

3. Перечень и описание столбцов:

Поле	Описание
Survived	выжил (1-да, 0- нет)
Pclass	Класс круиза
Name	ФИО
Sex	Пол
Age	Возраст
SibSp	Число братьев, сестер или супругов на борту у человека
Parch	Количество родителей или детей, с которыми путешествовал каждый пассажир
Ticket	Номер билета
Fare	Цена билета

```
Cabin    | Номер каюты
Embarked | Порт посадки
```

4. Определить количество пустых ячеек
5. *Заполнить пустые ячейки используя любую логику замещения данных
6. Если пункт 5 не выполнен, то удалить строки имеющие пустые ячейки.
7. По каждому признаку произвести аналитику:
    1. Количество уникальных значений
    2. Минимальное и максимальное значение
8. *С помощью аналитики определить влияние всех признаков на признак Survived (выживание).
   Пример: Parch и SibSP отрицательно влияли на выживание при крушении , так как чем больше эти параметры тем ниже процент выживания (одиночке выжить проще).

Результатом домашнего задания будет Таблица в Colab с комментариями в качестве ответов на пункты ДЗ.

Перечень инструментов, необходимых для реализации деятельности:
1) Google Colab https://colab.research.google.com/
2) *PyCharm

```
# подключение библиотек
import pandas as pd
import seaborn as sns


#чтоб не подключать диск с датасетом, загружаю по ссылке
import gdown
shareUrl = 'https://drive.google.com/file/d/1HtY8F6wKSBVn8CKSOOGmVDsynlAwhI7f/view?usp=sharing'
token = shareUrl[32:shareUrl.find('/view?usp=sharing')]
url = f'https://drive.google.com/uc?export=download&id={token}'
gdown.download(url, 'Titanic-Dataset.csv', quiet=False)
```

```
Downloading...
```

```
# загрузка данных
df = pd.read_csv('/content/Titanic-Dataset.csv')
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN

```
# 4. Определить количество пустых ячеек
print(df.isna().sum())
print(f'Total: {df.isna().sum().sum()}')
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
Total: 866
```

```
# посмотрим информацию по таблице
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
```

```
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
None


# 5. *Заполнить пустые ячейки используя любую логику замещения данных
# 6. Если пункт 5 не выполнен, то удалить строки имеющие пустые ячейки.
# для возраста возьмем среднее значение
# для порта первое значение
# каюту удалим т.к. большая часть значений пустая
df = df.fillna({'Age' : df.Age.mean()})
df = df.drop(labels='Cabin', axis=1)
df = df.fillna(method='ffill')
print(df.isna().sum())

PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64


# добавлю новое поле для удобства поиска корреляции
```

```
sex_map = {'male' : 1,
           'female' : 0}
df['is_male'] = df['Sex'].map(sex_map)
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500

891 rows × 12 columns

```
#7. По каждому признаку произвести аналитику:
#  1. Количество уникальных значений
#  2. Минимальное и максимальное значение
category_cols = df.select_dtypes(include=['object'])
category_cols_names = category_cols.columns.tolist()
print(f'категориальные столбцы: {category_cols_names}\n')
for col in category_cols_names:
    print(f'информация о столбце {col}:')
    print(f'количество уникальных значений: {category_cols[col].nunique()}')
    print(f'масимальное значение: {category_cols[col].max()}')
    print(f'минимальное значение: {category_cols[col].min()}')

# небольшая неоднозначность по заданию: "По каждому признаку" если признак это категориальный параметр то max и min для этого малоинформативен
# поэтому дополнительно выведу информацию по числовым значениям
```

```
category_int = df.select_dtypes(include=['int64','float64']) #выбрали столбцы с интервальные переменными
category_int_names = category_int.columns.tolist()
print(f'\n\nинтервальные столбцы: {category_int_names}\n')

for col in category_int_names:
    print(f'информация о столбце {col}:')
    print(f'количество уникальных значений: {category_int[col].nunique()}')
    print(f'максимум: {category_int[col].max()}')
    print(f'минимум: {category_int[col].min()}')
```

```
 категориальные столбцы: ['Name', 'Sex', 'Ticket', 'Embarked']

 информация о столбце Name:
 количество уникальных значений: 891
 масимальное значение: van Melkebeke, Mr. Philemon
 минимальное значение: Abbing, Mr. Anthony
 информация о столбце Sex:
 количество уникальных значений: 2
 масимальное значение: male
 минимальное значение: female
 информация о столбце Ticket:
 количество уникальных значений: 681
 масимальное значение: WE/P 5735
 минимальное значение: 110152
 информация о столбце Embarked:
 количество уникальных значений: 3
 масимальное значение: S
 минимальное значение: C


 интервальные столбцы: ['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'is_male']

 информация о столбце PassengerId:
 количество уникальных значений: 891
 максимум: 891
 минимум: 1
 информация о столбце Survived:
 количество уникальных значений: 2
 максимум: 1
 минимум: 0
 информация о столбце Pclass:
 количество уникальных значений: 3
 максимум: 3
 минимум: 1
```

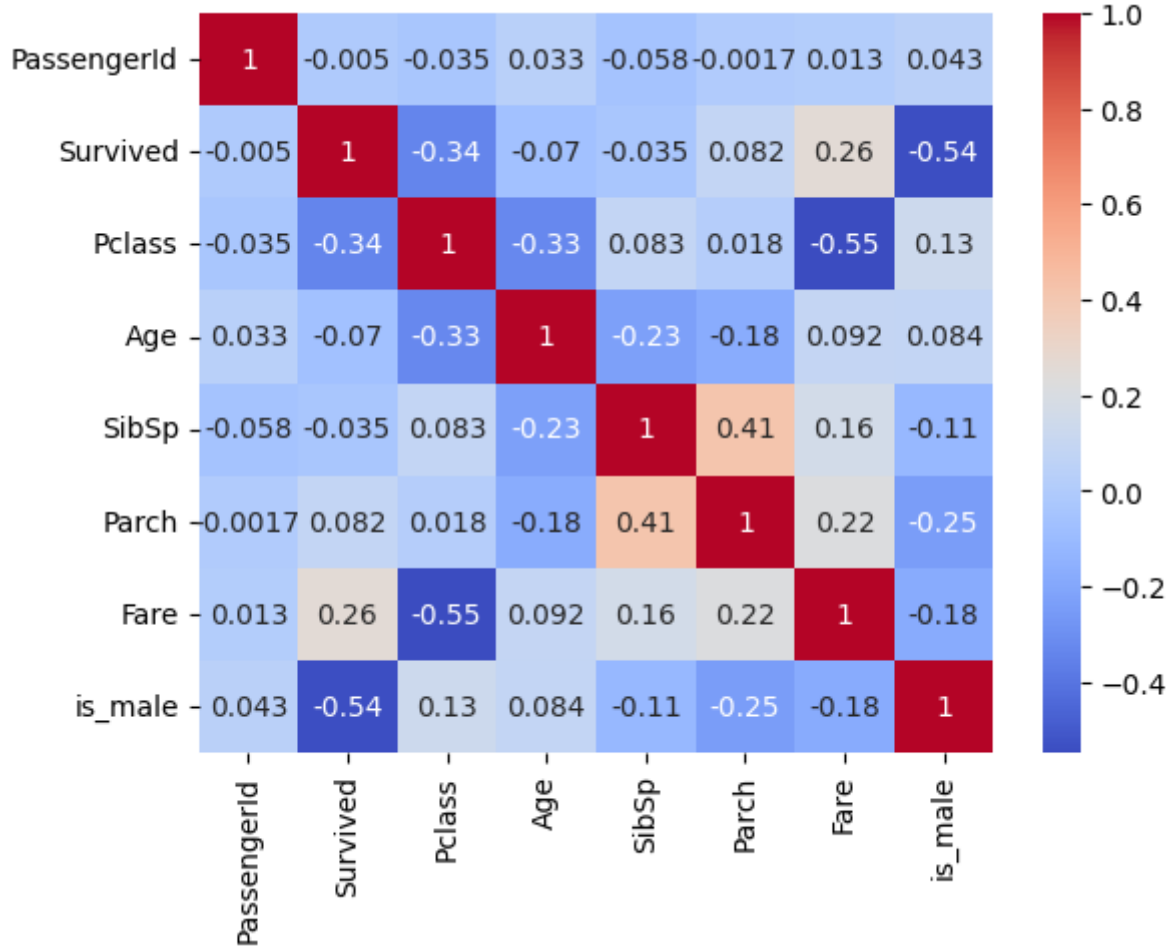
```
информация о столбце Age:
количество уникальных значений: 89
максимум: 80.0
минимум: 0.42
информация о столбце SibSp:
количество уникальных значений: 7
максимум: 8
минимум: 0
информация о столбце Parch:
количество уникальных значений: 7
максимум: 6
минимум: 0
информация о столбце Fare:
количество уникальных значений: 248
максимум: 512.3292
минимум: 0.0
информация о столбце is_male:
количество уникальных значений: 2
максимум: 1
минимум: 0
```

```
df.describe(include = 'all')
```

```
# 8. *С помощью аналитики определить влияние всех признаков на признак Survived (выживание).
# потстроим матрицу корреляций
matrix_corr = df.corr();

# вывод корреляционной матрицы
sns.heatmap(matrix_corr, annot=True, cmap='coolwarm');
```

```
<ipython-input-101-4366054f6d3c>:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future versior
  matrix_corr = df.corr();
```




```
print('При общем отношении ж-м')
print(df.groupby(['Sex']).size().transform(lambda x: x/x.sum()))
print('Среди выживших явно преобладает количество женщин')
```

```
print(df.groupby(['Survived', 'Sex']).size().transform(lambda x: x/x.sum()))
```

```
При общем отношении ж-м
Sex
female    0.352413
male      0.647587
dtype: float64
Среди выживших явно преобладает количество женщин
Survived  Sex
0         female    0.090909
          male      0.525253
1         female    0.261504
          male      0.122334
dtype: float64
```

```
print('Схожая ситуация наблюдается по соотношениям выживших пассажиров более дорогого класса')
print(df.groupby(['Pclass']).size().transform(lambda x: x/x.sum()))
print(df.groupby(['Survived', 'Pclass']).size().transform(lambda x: x/x.sum()))
```

```
Схожая ситуация наблюдается по соотношениям выживших пассажиров более дорогого класса
Pclass
1    0.242424
2    0.206510
3    0.551066
dtype: float64
Survived  Pclass
0         1         0.089787
          2         0.108866
          3         0.417508
1         1         0.152637
          2         0.097643
          3         0.133558
dtype: float64
```

```
print(df.groupby(['Parch']).size())
print(df.groupby(['Survived', 'Parch']).size())
print(df.groupby(['Survived', 'Parch']).size() / df.groupby(['Parch']).size())
print('Количество выживших одиночек - по отношению к изначальному их числу, наоборт показывает меньший процент выживания по сравнению с людей
```

```
Parch
0    678
1    118
```

```
2     80
3      5
4      4
5      5
6      1
dtype: int64
Survived  Parch
0         0        445
          1         53
          2         40
          3          2
          4          4
          5          4
          6          1
1         0        233
          1         65
          2         40
          3          3
          5          1
dtype: int64
Survived  Parch
0         0        0.656342
          1        0.449153
          2        0.500000
          3        0.400000
          4        1.000000
          5        0.800000
          6        1.000000
1         0        0.343658
          1        0.550847
          2        0.500000
          3        0.600000
          5        0.200000
dtype: float64
Количество выживших одиночек - по отношению к изначальному их числу, наоборт показывает меньший процент выживания по сравнению с людей с
```

```
print(df.groupby(['SibSp']).size())
print(df.groupby(['Survived', 'SibSp']).size())
print(df.groupby(['Survived', 'SibSp']).size() / df.groupby(['SibSp']).size())
print('Аналогичная ситуация и по соотношениям одиночек и семейных')
```

```
SibSp
```

```
0    608
1    209
2     28
3     16
4     18
5      5
8      7
dtype: int64
Survived  SibSp
0         0        398
          1         97
          2         15
          3         12
          4         15
          5          5
          8          7
1         0        210
          1        112
          2         13
          3          4
          4          3
dtype: int64
Survived  SibSp
0         0       0.654605
          1       0.464115
          2       0.535714
          3       0.750000
          4       0.833333
          5       1.000000
          8       1.000000
1         0       0.345395
          1       0.535885
          2       0.464286
          3       0.250000
          4       0.166667
dtype: float64
Аналогичная ситуация и по соотношениям одиночек и семейных


#для удобства анализа возраста добавим возрастную группу
df['age_group'] = df['Age'].apply(lambda x: 0 if x < 7 else 1 if x <= 18 else 2 if x < 55 else 3 )
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Er
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	
...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	

```
print(df.groupby(['age_group']).size())
print(df.groupby(['Survived', 'age_group']).size())
print(df.groupby(['Survived', 'age_group']).size() / df.groupby(['age_group']).size())
print('По резултату видно что в первую очередь спасали детей и подростков')
```

```
age_group
0     47
1     92
2    710
3     42
dtype: int64
Survived  age_group
0         0            14
          1            55
          2           451
          3            29
1         0            33
          1            37
          2           259
          3            13
dtype: int64
Survived  age_group
```

```
0         0            0.297872
          1            0.597826
          2            0.635211
          3            0.690476
1         0            0.702128
          1            0.402174
          2            0.364789
          3            0.309524
dtype: float64
```

По резултату видно что в первую очередь спасали детей и подростков

Поле	Результат анализа на выживание
Pclass	Класс круиза влияет на резальтат - чем выше класс тем больше шансов выжить
Sex	Пол однозначно влияет на выживание - в первую очередь спасали женщин
Age	Возраст повлиял на результаты выживания - в первую очередь спасали детей и подростков
SibSp	По этому параметру верно обратное утверждение - у одиночки меньше шансов выжить
Parch	По этому параметру верно обратное утверждение - у одиночки меньше шансов выжить

✓ 0 сек. выполнено в 19:07