|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА (ИУ12)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

**Отчет**

**Название**: Работа в Pandas на примере задачи «Titanic». Задача 1.

**Дисциплина:** Введение в искусственный интеллект

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ12-11М |  |  | Д.В. Кузнецов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д.Ю. Евсюков |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цели практики:**

• работа с данными, используя язык Python и пакет Pandas

• сделать предобработку данных

• нахождение простых закономерностей в данных

**Решение задач**

Импорт необходимых библиотек и подгрузка csv файла

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import re

data = pd.read\_csv('titanic.csv')

1. Какое количество мужчин и женщин ехало на корабле?

**Решение:**

sex\_counts = data['Sex'].value\_counts()

male\_count = sex\_counts['male']

female\_count = sex\_counts['female']

print(f'Количество мужчин: {male\_count}, Количество женщин: {female\_count}')

**Ответ:**

*Количество мужчин: 577, Количество женщин: 314*

1. Какой части пассажиров удалось выжить? Посчитайте долю выживших пассажиров.

**Решение:**

survived\_percentage = (data['Survived'].sum() / len(data)) \* 100

print(f'Доля выживших пассажиров: {survived\_percentage:.2f}%')

**Ответ:**

*Доля выживших пассажиров: 38.38%*

1. Какую долю пассажиры первого класса составляли среди всех пассажиров?

**Решение:**

first\_class\_percentage = (data[data['Pclass'] == 1].shape[0] / len(data)) \* 100

print(f'Доля пассажиров первого класса: {first\_class\_percentage:.2f}%')

**Ответ:**

*Доля пассажиров первого класса: 24.24%*

1. Какого возраста были пассажиры? Посчитайте среднее и медиану возраста пассажиров.

**Решение:**

mean\_age = data['Age'].mean()

median\_age = data['Age'].median()

print(f'Средний возраст: {mean\_age:.2f}, Медианный возраст: {median\_age:.2f}')

**Ответ:**

*Средний возраст: 29.70, Медианный возраст: 28.00*

1. Коррелируют ли число братьев/сестер с числом родителей/детей? Посчитайте корреляцию Пирсона между признаками SibSp и Parch.

**Решение:**

result = data[['SibSp', 'Parch']].corr()

sub\_data = pd.DataFrame.copy(data[['Pclass', 'Fare', 'Age', 'Survived', 'SibSp', 'Parch', 'Sex']])

sub\_data['Sex'] = sub\_data.Sex.map({'male': 0, 'female': 1})

sub\_data\_corr = sub\_data[['Pclass', 'Fare', 'Age', 'Survived', 'SibSp', 'Parch', 'Sex']].corr()

sns.heatmap(sub\_data\_corr, annot=True, xticklabels=sub\_data\_corr.columns.values, yticklabels=sub\_data\_corr.columns.values)

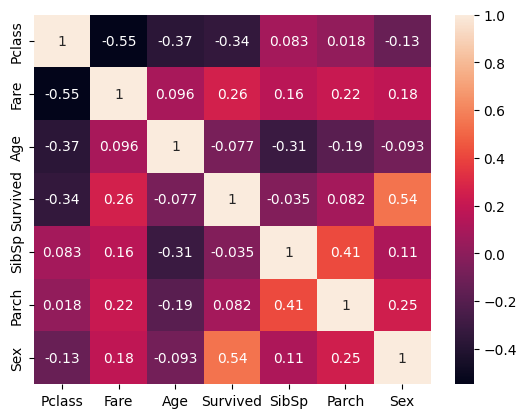
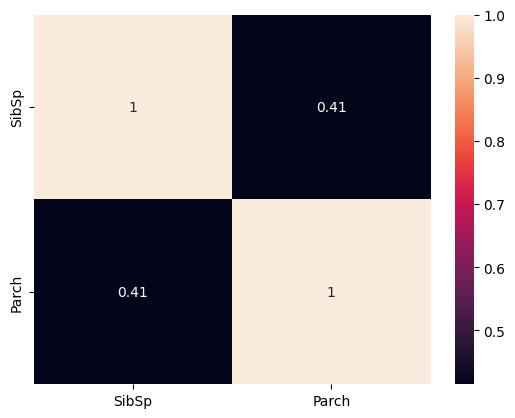
plt.show()

sub\_data\_corr = sub\_data[['SibSp', 'Parch']].corr()

sns.heatmap(sub\_data\_corr, annot=True, xticklabels=sub\_data\_corr.columns.values, yticklabels=sub\_data\_corr.columns.values)

plt.show()

print(f'Корреляция между SibSp и Parch:\n{result}')

**Ответ:**

*Корреляция между SibSp и Parch:*

*SibSp Parch*

*SibSp 1.000000 0.414838*

*Parch 0.414838 1.000000*

1. Какое самое популярное женское имя на корабле?

**Решение:**

filtered\_data = data.loc[data['Sex'] == 'female'].Name

prefixes = ['Mrs. ', 'Miss. ', 'Ms. ']

first\_names = []

def extract\_first\_name(name):

    if '(' in name:

        if '("' in name:

            return name.split(prefix)[1].split(' ')[0]

        short\_name = re.sub(r'\W+', '', name[name.find('(')+1:name.find(')')].split(' ', 1)[0])

    else:

        short\_name = name.split(prefix)[1].split(' ')[0]

    return short\_name

for name in filtered\_data:

    short\_name = None

    for prefix in prefixes:

        if prefix in name:

            short\_name = extract\_first\_name(name)

            first\_names.append(short\_name)

            break

name\_counts = pd.Series(first\_names).value\_counts()

print(name\_counts)

most\_popular\_name = name\_counts.idxmax()

count\_of\_most\_popular\_name = name\_counts.max()

print(f'Самое популярное имя: {most\_popular\_name}, Количество: {count\_of\_most\_popular\_name}')

**Ответ:**

*Самое популярное имя: Anna, Количество: 15*

**Вывод**

В ходе лабораторной работы успешно достигнуты поставленные цели, включая освоение работы с данными в Python с использованием библиотеки Pandas. Предварительная обработка данных позволила выявить простые закономерности, такие как распределение мужчин и женщин на корабле, процент выживших, и доля пассажиров первого класса. Также проведены расчеты среднего и медианного возраста пассажиров, анализ корреляции между числом братьев/сестер и родителей/детей, и выделено самое популярное женское имя.