

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

Поляк М.Д.

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Знакомство с Jupyter notebook и Google Colab

по дисциплине: Основы машинного обучения

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

4134к

подпись, дата

Я.Б. Салий

инициалы, фамилия

Санкт-Петербург
2024

Цель работы

Знакомство со средами Jupyter Notebook и Google Colaboratory, а также библиотеками Pandas и matplotlib

Исходный код jupyter_assignment.ipynb:

```
### BEGIN YOUR CODE
#I have read through the Introduction and Overview notebook
READ_INTRODUCTION = True

#I understand (at a high level) what Jupyter notebooks are and how to read and
#interact with them (or I have been in touch with the course instructor to ask for help)
LEARNED_ABOUT_JUPYTER = True

#I've created (or already have) a Google account and can access Google
#Colaboratory under my own account
ACCESS_COLABORATORY = True

#I've created a GitHub account
CREATED_GITHUB_ACCOUNT = True
github_username = 'Ichika51'

#My info
my_name = 'Ярослав'
### END YOUR CODE

def greet(name):
    if isinstance(name, list):
        # Проверка, что список состоит из строк
        if all(isinstance(n, str) for n in name):
            # Форматирование для двух или более имен
            if len(name) > 2:
                return f'Hello, {', '.join(name[:-1])}, and {name[-1]}!'
            elif len(name) == 2:
                return f'Hello, {name[0]} and {name[1]}!'
            else:
                return f'Hello, {name[0]}!'
        else:
            return "Error: Все элементы должны быть строками!"
    elif isinstance(name, str):
        return f'Hello, {name}!'
    else:
        # Обработка других типов данных
        return f'Hello, {str(name)}!'

print(greet(["Ярослав", "Петя"]));
```

Исходный код pandas_exercise.ipynb:

```
Student_ID = 13

!wget https://raw.githubusercontent.com/markpolyak/datasets/refs/heads/main/data/
aircarrier_market_us_2019.zip

!unzip aircarrier_market_us_2019.zip

import pandas as pd
```

```

df = pd.read_csv('aircarrier_market_us_2019.csv')
columns = df.columns.tolist()
print(columns)

answer1 = df['CARRIER'].nunique()
print(answer1)

uk_to_us_data = df[(df['ORIGIN_COUNTRY_NAME'] == 'United Kingdom') &
                    (df['DEST_COUNTRY_NAME'] == 'United States')]

# Подсчет общих сумм
total_passengers = uk_to_us_data['PASSENGERS'].sum()
total_freight = uk_to_us_data['FREIGHT'].sum()
total_mail = uk_to_us_data['MAIL'].sum()
answer2 = max(total_passengers, total_freight, total_mail)
print(answer2)

mail_sums = df.groupby('UNIQUE_CARRIER_NAME')['MAIL'].sum()
freight_sums = df.groupby('UNIQUE_CARRIER_NAME')['FREIGHT'].sum()
total_cargo = mail_sums + freight_sums
answer3 = total_cargo.idxmax()
print(answer3)

us_to_other_countries = df[df['ORIGIN_COUNTRY'] == 'US']
total_passengers = us_to_other_countries.groupby('UNIQUE_CARRIER_NAME')['PASSENGERS'].sum()
answer4 = total_passengers.max()
print(answer4)

passengers_between_cities = df.groupby(['ORIGIN_CITY_NAME', 'DEST_CITY_NAME'])
['PASSENGERS'].sum().reset_index()
max_passengers_row = passengers_between_cities.loc[passengers_between_cities['PASSENGERS'].idxmax()]
origin_city = max_passengers_row['ORIGIN_CITY_NAME']
dest_city = max_passengers_row['DEST_CITY_NAME']
answer5 = [origin_city, dest_city]
print(answer5)

origin_city = answer5[0]
dest_city = answer5[1]
filtered_data = df[(df['ORIGIN_CITY_NAME'] == origin_city) & (df['DEST_CITY_NAME'] == dest_city)]
passengers_per_carrier = filtered_data.groupby('UNIQUE_CARRIER_NAME')
['PASSENGERS'].sum().reset_index()
top_carriers = passengers_per_carrier.nlargest(3, 'PASSENGERS')
total_passengers = filtered_data['PASSENGERS'].sum()
# Рассчитываем процент перевозок
percentages = (top_carriers['PASSENGERS'] / total_passengers) * 100
answer6 = percentages.round(2).tolist()
print(answer6)

f_class_data = df[df['CLASS'] == 'F']
passengers_per_country = f_class_data.groupby('ORIGIN_COUNTRY_NAME')
['PASSENGERS'].sum().reset_index()
total_passengers_f_class = passengers_per_country['PASSENGERS'].sum()
passengers_per_country['PERCENTAGE'] = (passengers_per_country['PASSENGERS'] /
total_passengers_f_class) * 100
# Сортируем страны по количеству пассажиров и выбираем третью по величине
third_country = passengers_per_country.nlargest(3, 'PASSENGERS')['ORIGIN_COUNTRY_NAME'].iloc[-1]
answer7 = third_country
print(answer7)

origin_city = answer5[0]
dest_city = answer5[1]

```

```

filtered_data = df[(df['ORIGIN_CITY_NAME'] == origin_city) & (df['DEST_CITY_NAME'] == dest_city)] |
((df['ORIGIN_CITY_NAME'] == dest_city) & (df['DEST_CITY_NAME'] == origin_city))]
monthly_passengers = filtered_data.groupby('MONTH')['PASSENGERS'].sum().reset_index()
max_passenger_month = monthly_passengers.loc[monthly_passengers['PASSENGERS'].idxmax()]
answer8 = max_passenger_month['MONTH']
print(answer8)

```

```

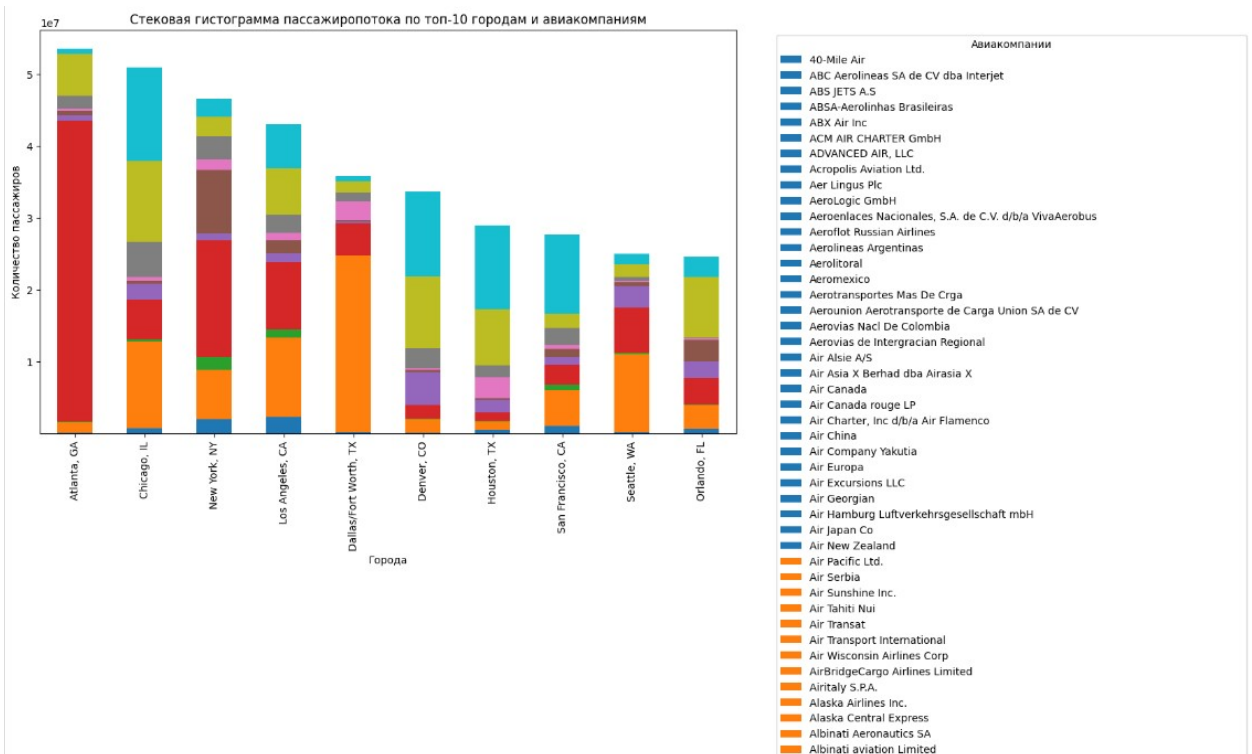
import matplotlib.pyplot as plt
import numpy as np

```

```

passenger_flow = df.groupby(['ORIGIN_CITY_NAME', 'UNIQUE_CARRIER_NAME'])
['PASSENGERS'].sum().unstack(fill_value=0)
top_cities = passenger_flow.sum(axis=1).nlargest(10).index
passenger_flow_top = passenger_flow.loc[top_cities]
ax = passenger_flow_top.plot(kind='bar', stacked=True, colormap='tab10', figsize=(12, 7))
plt.title('Стековая гистограмма пассажиропотока по топ-10 городам и авиакомпаниям')
plt.xlabel('Города')
plt.ylabel('Количество пассажиров')
plt.legend(title='Авиакомпании', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()

```



Исходный код matplotlib_assignment.ipynb:

```

import numpy as np
from matplotlib import pyplot as plt

```

```
Student_ID = 13
```

```
def my_function(x,a,b,c,d):
```

```

    return a * np.sin(x) ** 2 + b * np.log(x) / np.log(c)

def my_function(x, a, b, c):
    return np.sqrt((c**2 - a * x**2) / b)

a = 1
b = 1
c = 2
x = np.linspace(0, 1.4, 100)

figure, axes = plt.subplots(figsize=(12, 8))

# Наборы параметров для графиков
params = [
    (1, 1, 2, 'blue', 'График 1: a=1, b=1, c=2', '-'),
    (1, 2, 2, 'red', 'График 2: a=1, b=2, c=2', '--'),
    (2, 1, 2, 'green', 'График 3: a=2, b=1, c=2', '.'),
    (2, 2, 2, 'orange', 'График 4: a=2, b=2, c=2', '-.')
]

# Все графики на одном графике
for a, b, c, color, title, linestyle in params:
    y = my_function(x, a, b, c)
    axes.plot(x, y, linestyle=linestyle, color=color, linewidth=2, label=title)

# Настройки графика
axes.set_title("Графики функции y(x) с разными параметрами", fontsize=16)
axes.set_xlabel("x", fontsize=14)
axes.set_ylabel("y(x)", fontsize=14)
axes.legend(fontsize=10)
axes.grid(True)

plt.show()

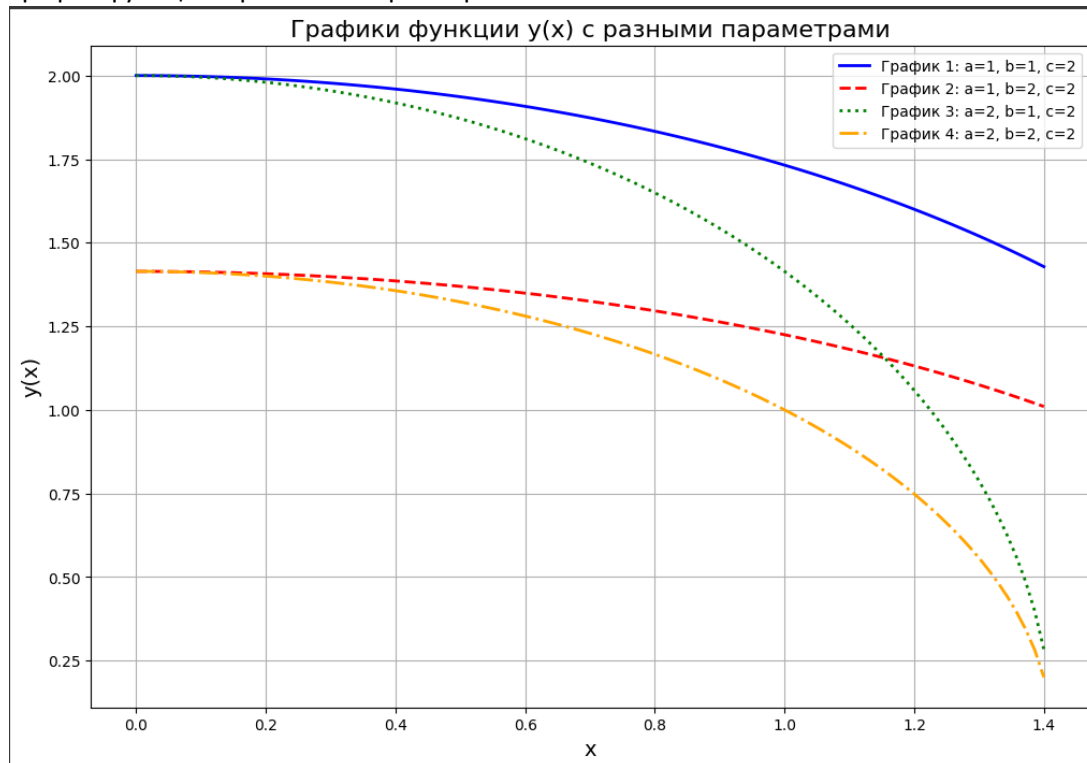
```

Функция в Latex:

```
$$ a \cdot x^2 + b \cdot y^2 = c^2 $$
```

$$a \cdot x^2 + b \cdot y^2 = c^2$$

График функции с разными параметрами:



Выводы

В ходе лабораторной работы были выполнены три задания, каждое из которых направлено на изучение различных инструментов и библиотек для анализа данных и визуализации в Python.

Задание 1: Знакомство с Jupyter и Google Colab:

В этом задании мы ознакомились с платформой Jupyter Notebook и Google Colaboratory. Это инструменты, которые позволяют выполнять код на Python в интерактивном режиме, легко визуализировать результаты и делиться ими с другими.

Задание 2: Знакомство с Matplotlib и LaTeX:

В данном задании мы изучили, как использовать библиотеку Matplotlib для визуализации данных и графиков. Мы научились строить графики и настраивать их внешний вид, а также использовать LaTeX для оформления математических формул. Это позволяет делать визуализации более информативными и эстетически привлекательными, что особенно важно при представлении научных результатов.

Задание 3: Знакомство с Pandas:

В этом задании мы работали с библиотекой Pandas, которая является мощным инструментом для анализа и манипуляции данными. Мы загрузили датасет, провели его разведочный анализ, формулировали гипотезы о значении полей и проверяли их. Это помогло нам понять структуру данных и выявить важные закономерности, что является неотъемлемой частью анализа данных.