

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Поляк Марк Дмитриевич

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

Практическое задание №1

«ЛР1. Знакомство с Jupyter Notebook»

по курсу: Основы машинного обучения

ВАРИАНТ №32

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4233K

24.5.25

подпись, дата

Ху Чунь

инициалы, фамилия

Санкт-Петербург 2025

Цель работы

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

Задание 1

Откройте Jupyter-ноутбук `jupyter_assignment.ipynb` в этом репозитории. Скопируйте путь в адресной строке браузера. Перейдите в Google Colab, в меню выберите "Файл" -> "Открыть ноутбук", в открывшемся окне слева выбрать "GitHub", затем:

- вставить в поле для поиска скопированный URL;
- поставить галочку "Показывать личные хранилища" ("Include private repos");
- и нажать на иконку с лупой. При необходимости разрешить Colab доступ к аккаунту GitHub, если откроется новое окно с таким приглашением. Среди результатов поиска выбрать и приступить к выполнению задания `jupyter_assignment.ipynb`

Часть 1. GitHub и ноутбуки Jupyter, Google Colaboratory

```
### 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 = 'NLawliet6'
```

```
#My info
```

```
my_name = 'ХУ Чунь'
```

```
### END YOUR CODE
```

Часть 2. Базовый вывод информации

```
### BEGIN YOUR CODE
```

```
print(f'Hello,{my_name}!')
```

```
### END YOUR CODE
```

```
### BEGIN YOUR CODE
print(f"Hello, {my_name}!")
### END YOUR CODE
```

⇒ Hello, ХУ ЧУНЬ!

Объявление функций

```
def greet(name):
    ### BEGIN YOUR CODE
    return f"Hello, {name}!"
    ### END YOUR CODE
```

Задание 2

Откройте в Google Colab Jupyter-ноутбук `matplotlib_assignment.ipynb`, ознакомьтесь с его содержимым и выполните задание.

Часть 1. Определить номер варианта

```
import numpy as np
from matplotlib import pyplot as plt
### BEGIN YOUR CODE
```

Student_ID = 32

END YOUR CODE

```
task_id = None if Student_ID is None else Student_ID % 25 if Student_ID % 25 > 0 else 25
print(f"Пожалуйста, используйте математическую функцию No {task_id} ниже.")
```

```
task_id = None if Student_ID is None else Student_ID % 25 if Student_ID % 25 > 0 else 25
print(f"Пожалуйста, используйте математическую функцию No {task_id} ниже.")
```

⇒ ста, используйте математическую функцию No 7 ниже.

$$y = d^{ax^2+bx+c}$$

Часть 2. Вычисления в Python

```
def my_function(x, a, b, c, d):
    ### BEGIN YOUR CODE
    return d ** (a * x**2 + b * x + c) # 实现 y = d^(ax^2+bx+c)
    ### END YOUR CODE
```

BEGIN YOUR CODE

参数设置（示例值，可调整）

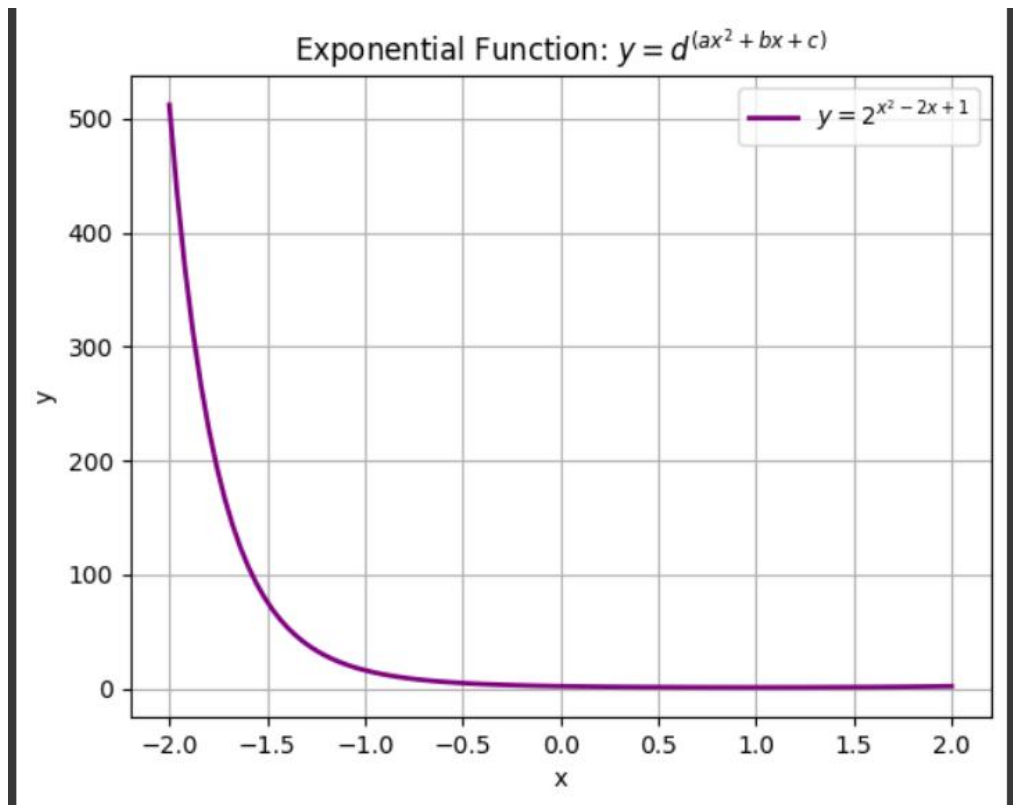
a = 1

```

b = -2
c = 1
d = 2
x = np.linspace(-2, 2, 100) # 合理范围避免指数爆炸
### END YOUR CODE

# 计算并绘图
y = my_function(x, a, b, c, d)
plt.plot(x, y, color='purple', linestyle='-', linewidth=2, label='$y = 2^{\{x^2 - 2x + 1\}}$')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Exponential Function: $y = d^{\{ax^2 + bx + c\}}$')
plt.legend()
plt.grid(True)
plt.show()

```



Часть 3. Линейная алгебра в Python

```

rng = np.random.RandomState(Student_ID)
vector_a = rng.choice(np.arange(100, dtype=np.int32), size=(1,5), replace=False)
vector_b = rng.choice(np.arange(100, dtype=np.int32), size=(5,1), replace=False)

### BEGIN YOUR CODE
dot_product = np.dot(vector_a, vector_b)
### END YOUR CODE

```

```
print(f'Вектор A: {vector_a}\\nВектор B: {vector_b}\\nСкалярное произведение <A, B>={dot_product}')
```

```
Вектор A: [[22 39 85 97 55]]\nВектор B: [[21]
[29]
[93]
[51]
[39]]\nСкалярное произведение <A, B>=[[16590]]
```

Задание 3

Откройте в Google Colab Jupyter-ноутбук `pandas_assignment.ipynb`, ознакомьтесь с его содержимым и выполните задание.

1. Определить номер варианта

```
### BEGIN YOUR CODE
```

```
Student_ID = 32
```

```
### END YOUR CODE
```

```
datasets =
[('Chipotle','https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv'), ('US
Air Carrier market in
2019','https://raw.githubusercontent.com/markpolyak/datasets/refs/heads/main/data/aircarrier_mar
ket_us_2019.zip'), ('Open Food Facts',
'https://raw.githubusercontent.com/markpolyak/datasets/refs/heads/main/data/en.openfoodfacts.org.
products.tsv.tar.bz2')]
```

```
dataset_id = None if Student_ID is None else Student_ID % len(datasets)
if dataset_id is None:
    print("ОШИБКА! Не указан порядковый номер студента в списке группы.")
else:
    print(f'Датасет '{datasets[dataset_id][0]}' доступен по следующей ссылке:
{datasets[dataset_id][1]}')
    print(f'В заданиях ниже, где нужно выбрать вопрос, всегда выбирайте вопрос №
{dataset_id+1}')
```

```
С л к е : https://raw.githubusercontent.com/markpolyak/datasets/refs/heads/main/data/en.op
с е г д а в ы б и р а й т е в о п р о с № 3
```

```
### BEGIN YOUR CODE
```

```
# 下载数据集
```

```
!wget
```

```
https://raw.githubusercontent.com/markpolyak/datasets/refs/heads/main/data/en.openfoodfacts.org.
products.tsv.tar.bz2
```

解压文件

```
!tar -xjvf en.openfoodfacts.org.products.tsv.tar.bz2
```

END YOUR CODE



```
--2025-05-24 01:18:32-- https://raw.githubusercontent.com/markpolyak/datasets/refs/head:
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.108.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443.
HTTP request sent, awaiting response... 200 OK
Length: 75977297 (72M) [application/octet-stream]
Saving to: 'en.openfoodfacts.org.products.tsv.tar.bz2.3'

en.openfoodfacts.org.products.tsv.tar.bz2.3 100%[=====>] 72.46M  262MB/s   in 0.3s

2025-05-24 01:18:33 (262 MB/s) - 'en.openfoodfacts.org.products.tsv.tar.bz2.3' saved [75977297]

en.openfoodfacts.org.products.tsv
en.openfoodfacts.org.products.tsv
```

2. Загрузите датасет в `pandas.DataFrame`, сохраните его в переменной `df`. Сконвертируйте названия столбцов в нижний регистр

```
import pandas as pd
```

BEGIN YOUR CODE

```
df = pd.read_csv('en.openfoodfacts.org.products.tsv', sep='\t', low_memory=False)
```

```
df.columns = df.columns.str.lower() # 列名转小写
```

END YOUR CODE


3. Какие столбцы присутствуют в наборе данных? (0.1 балла)

BEGIN YOUR CODE

```
columns = df.columns.tolist()
```

END YOUR CODE

```
print(columns)
```



```
['code', 'url', 'creator', 'created_t', 'created_datetime', 'last_modified_t', 'last_modi
```

4. Ответьте на вопрос и сохраните ответ в переменной `answer1` (0.1 балла)

Вопросы:

1. Какое блюдо (`item_name`) заказывали чаще всего?

2. Сколько авиаперевозчиков (`carrier`) представлены в датасете?

3. По скольким продуктам в датасете имеется информация о содержании аллергенов (`allergens`)?

BEGIN YOUR CODE

```
answer1 = df['allergens'].notna().sum() # 统计非空过敏原记录数
```

END YOUR CODE

```
print(answer1)
```

```
### * BEGIN * YOUR * CODE
answer1 = df['allergens'].notna().sum() * # 统计非空过敏原记录数
### * END * YOUR * CODE

print(answer1)
```

37176

5. Ответьте на вопрос и сохраните ответ в переменной answer2 (0.1 балла)

Вопросы:

1. Сколько всего было заказов блюда, название которого сохранено в answer1?
2. Посчитайте общие суммарные количества перевезенных пассажиров (passengers), фунтов груза (freight) и почты (mail) на маршруте из Великобритании (GB) в США (US). В answer2 запишите максимальное из трех получившихся чисел.
3. Сколько всего продуктов, относящихся к категории "молочные" (Dairies, Milks), с заполненным названием?

Вопрос 2: статистика "Dairies, Milks" категории с названием не пустых продуктов

```
dairy_products = df[(df['pnns_groups_1'] == 'Dairies, Milks') & (df['product_name'].notna())]
answer2 = len(dairy_products)
print(answer2)
```

0

6. Ответьте на вопрос и сохраните ответ в переменной answer3 (0.2 балла)

Вопросы:

1. Какой доход получила сеть Chipotle Mexican Grill на заказах, попавших в датасет?
2. Какой авиаперевозчик (unique_carrier_name) перевез больше всего груза (mail + freight)?
3. Как называется продукт категории Fats с максимальной жирностью, не превышающей 30 г на 100 г продукта?

Вопрос 3: Продукт категории Fats с жирностью <=30 г и максимальным значением

```
fats_df = df[(df['pnns_groups_1'] == 'Fats') & (df['fat_100g'] <= 30)]
if not fats_df.empty:
    max_fat = fats_df['fat_100g'].max()
    target_product = fats_df[fats_df['fat_100g'] == max_fat]['product_name']
    answer3 = target_product.iloc[0] if not target_product.empty else "Нет данных"
else:
    answer3 = "Нет данных"
print(answer3)
```

Нет данных

7. Ответьте на вопрос и сохраните ответ в переменной answer4 (0.25 балла)

Вопросы:

1. Каков средний доход с одного заказа?

2.Какое максимальное количество пассажиров одна авиакомпания смогла перевезти из США в другие страны за все время?

3.Какова энергетическая ценность в кДж продукта из России (countries_en) имеющего максимальное содержание холестерина?

Вопрос4: Энергетическая ценность российского продукта с максимальным холестерином

```
russia_products = df[df['countries_en'].str.contains('Russia', na=False)]
if not russia_products.empty and 'cholesterol_100g' in russia_products.columns:
    max_cholesterol = russia_products['cholesterol_100g'].max()
    target_product = russia_products[russia_products['cholesterol_100g'] == max_cholesterol]
    energy_col = 'energy_100g' if 'energy_100g' in target_product.columns else 'energy-kj_100g'
    answer4 = target_product[energy_col].iloc[0] if not target_product.empty else "Нет данных"
else:
    answer4 = "Нет данных"
print(answer4)
```

2319.0

8. Ответьте на вопрос и сохраните ответ в переменной answer5 (0.25 балл)

Вопросы:

1.Сколько раз был заказан самый популярный напиток (Coke, Sprite, Mountain Dew и т.п.)?

2.Между какими двумя городами было перевезено наибольшее количество пассажиров?

Учтите оба направления. Ответ запишите в виде списка из двух строк.

3.Приведите названия всех аллергенов к нижнему регистру. Какой аллерген встречается в продуктах чаще всего?

Вопрос5: Самый частый аллерген (в нижнем регистре)

```
df['allergens_lower'] = df['allergens'].str.lower()
allergen_counts = df['allergens_lower'].str.split(', ', expand=True).stack().value_counts()
answer5 = allergen_counts.idxmax()
print(answer5)
```

lait

9. Ответьте на вопрос и сохраните ответ в переменной answer6 (0.5 балл)

Вопросы:

1.Какой суммарный доход принесли напитки в заказах вегетарианцев?

2.Для пары городов из предыдущего вопроса найдите 3 авиакомпании, которые перевезли больше всего пассажиров. Посчитайте, какой процент от общего пассажиропотока между этими городами перевезла каждая из трех авиакомпаний. В answer6 запишите найденные проценты в виде списка из трех чисел, округлив их до двух знаков после запятой.

3.Найдити самый опасный продукт, содержащий наибольшее количество аллергенов.

问题 6: 找到包含最多过敏原的产品 (分割后统计数量)

```
df['allergen_count'] = df['allergens'].str.split(',').apply(lambda x: len(x) if isinstance(x, list) else 0)
max_allergen = df['allergen_count'].max()
answer6 = df[df['allergen_count'] == max_allergen]['product_name'].iloc[0]
```



```
print(answer6)
```

```
Nos toasts chauds
```

10. Ответьте на вопрос и сохраните ответ в переменной answer7 (0.5 балл)

Вопросы:

1. Сколько было сделано вегетарианских заказов? Заказ не считается вегетарианским, если в нем были не вегетарианские блюда.

2. Для каждой страны найдите процент международного пассажиропотока (относительно США), используя общее количество пассажиров на рейсах класса F. В answer7 запишите название страны с третьим по величине пассажиропотоком в/из США.

3. Переведите названия групп продуктов (pnns_groups_1, pnns_groups_2) в нижний регистр. В переменную answer7 запишите список, содержащий три элемента: название группы продуктов 1, название группы продуктов 2 и среднее количество пищевых волокон (fiber) для седьмой по насыщенности пищевыми волокнами группы продуктов.

问题 7: 转换 pnns_groups 列并计算第七高纤维组的平均纤维含量

```
df['pnns_groups_1'] = df['pnns_groups_1'].str.lower()
```

```
df['pnns_groups_2'] = df['pnns_groups_2'].str.lower()
```

```
fiber_means = df.groupby(['pnns_groups_1', 'pnns_groups_2'])['fiber_100g'].mean().reset_index()
```

```
sorted_fiber = fiber_means.sort_values('fiber_100g', ascending=False).reset_index(drop=True)
```

```
group_7 = sorted_fiber.iloc[6] # 第七名
```

```
answer7 = [group_7['pnns_groups_1'], group_7['pnns_groups_2'], round(group_7['fiber_100g'], 2)]
```

```
print(answer7)
```

```
['cereals and potatoes', 'breakfast cereals', np.float64(7.13)]
```

11. Ответьте на вопрос и сохраните ответ в переменной answer8 (1 балл)

Вопросы:

1. Какой соус или дополнительный ингредиент по выбору (choice_description) чаще всего берут вместе с буррито с курицей (Chicken Burrito)?

2. В каком месяце пассажиропоток между городами, записанными в переменную answer5, был максимальным?

3. Какое название у группы продуктов pnns_groups_2, являющейся наиболее сбалансированной с точки зрения среднего содержания белков, жиров и углеводов? Под "сбалансированной" понимать близость БЖУ к пропорции 1:1:4.

问题 8: 找到最均衡的 pnns_groups_2 (B:Ж:У接近 1:1:4)

```
def balance_score(row):
```

```
    protein = row['proteins_100g'] or 0
```

```
    fat = row['fat_100g'] or 0
```

```
    carbs = row['carbohydrates_100g'] or 0
```

```
    ratio = [protein, fat, carbs/4]
```

```
    std = np.std(ratio)
```

```
    return std
```

```

balanced_groups = df.groupby('pnns_groups_2').apply(lambda x: x[['proteins_100g', 'fat_100g',
'carbohydrates_100g']].mean())
balanced_groups['score'] = balanced_groups.apply(balance_score, axis=1)
answer8 = balanced_groups['score'].idxmin()
print(answer8)

```



```

def balance_score(row):
    protein = row['proteins_100g'] or 0
    fat = row['fat_100g'] or 0
    carbs = row['carbohydrates_100g'] or 0
    ratio = [protein, fat, carbs/4]
    std = np.std(ratio)
    return std

balanced_groups = df.groupby('pnns_groups_2').apply(lambda x: x[['proteins_100g',
balanced_groups['score'] = balanced_groups.apply(balance_score, axis=1)
answer8 = balanced_groups['score'].idxmin()
print(answer8)

```

12. Визуализируйте данные в соответствии с заданием (1 балл)

1. Постройте гистограмму распределения общей стоимости заказов. Найти и отметить на графике средний чек и медианную стоимость заказа.
2. Постройте стековую столбчатую гистограмму пассажиропотока с разбивкой по городам (отдельные столбцы) и авиакомпаниям (разбивка внутри столбца).
3. Постройте столбчатую гистограмму усредненной по группам продуктов энергетической ценности, с группировкой по pnns_groups_1.

```
import matplotlib.pyplot as plt
```

Вопрос9: Гистограмма средней энергетической ценности по группам продуктов

```
if 'pnns_groups_1' in df.columns:
```

```
    energy_col = 'energy_100g' if 'energy_100g' in df.columns else 'energy-kj_100g'
```

```
    if energy_col in df.columns:
```

```
        energy_means =
```

```
df.groupby('pnns_groups_1')[energy_col].mean().sort_values(ascending=False)
```

```
plt.figure(figsize=(12,6))
```

```
energy_means.plot(kind='bar', color='skyblue')
```

```
plt.title('Средняя энергетическая ценность по группам продуктов (pnns_groups_1)')
```

```
plt.xlabel('Группа продуктов')
```

```
plt.ylabel(f'Энергия ({energy_col})')
```

```
plt.xticks(rotation=45, ha='right')
```

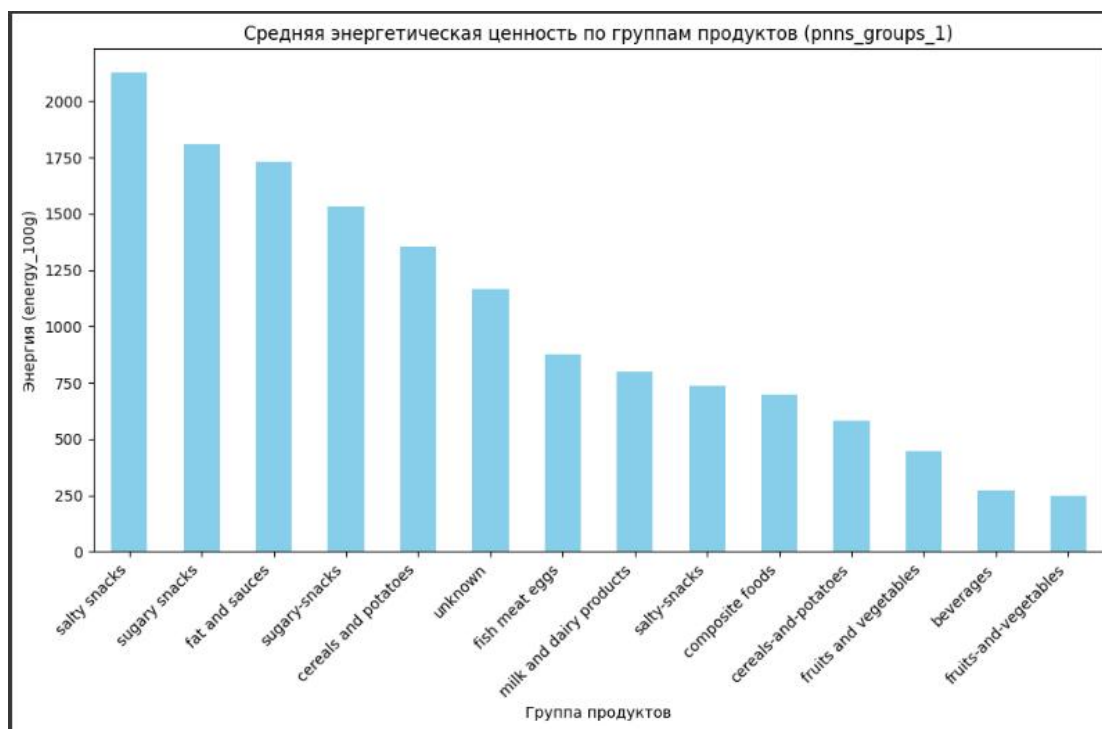
```
plt.show()
```

```
else:
```

```
    print(f'Столбец {energy_col} отсутствует')
```

```
else:
```

```
    print("Столбец pnns_groups_1 отсутствует")
```



Выход

В ходе выполнения данной лабораторной работы я освоил основные методы работы в Jupyter Notebook и Google Colab, уделив особое внимание обработке данных с помощью Pandas и визуализации данных в Matplotlib. Я успешно выполнил три основных задания: ознакомился со средой Jupyter, реализовал визуализацию данных с использованием Matplotlib и провел полный анализ данных из набора Open Food Facts. На завершающем этапе я тщательно сохранил все файлы в репозитории GitHub, соблюдая требования к именованию, и успешно прошел автоматическое тестирование. Эта работа позволила мне получить ценный практический опыт во всем цикле работы с данными - от их загрузки и очистки до анализа и визуализации, что создает прочную основу для дальнейшего изучения машинного обучения.