Машинное обучение и интеллектуальный анализ данных

Семинар 2

Г.А. Ососков*, О.И. Стрельцова*, Д.И. Пряхина*, Д.В. Подгайный*, А.В. Стадник*, Ю.А. Бутенко*

Государственный университет «Дубна»

***Лаборатория информационных технологий, ОИЯИ**

Дубна, Россия

Решение задачи

jupyter

python

Конверсия посетителей сайта
Предметная область: интернет-маркетинг

1. Подключение библиотек

NumPy

```
import numpy as np
import matplotlib.pyplot as plt
import math
```

2. Создание списков с исходными данными

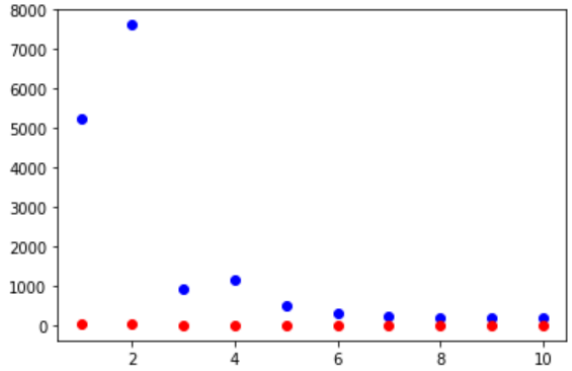
```
days = np.array(range(1, 11))
print(days)
views = np.array([5252, 7620, 941, 1159, 485, 299, 239, 195, 181, 180])
print(views)
downloads = np.array([21, 46, 9, 8, 3, 6, 4, 2, 2, 2])
print(downloads)
```

Решение задачи

Конверсия посетителей сайта
Предметная область: интернет-маркетинг

3. Визуализация исходных данных

```
plt.plot(days, views, 'bo', days, np.array(downloads), 'ro')
```

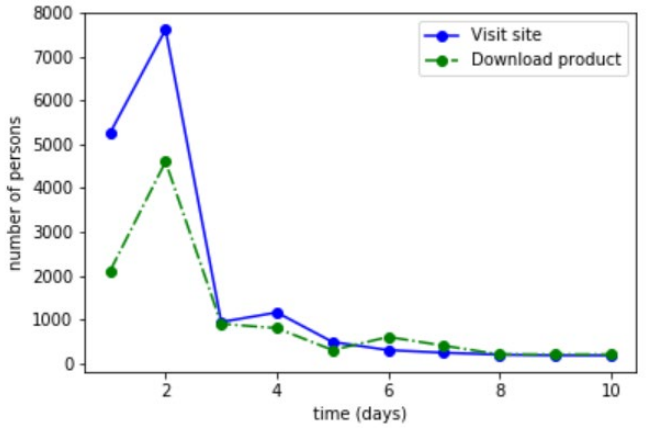


✓ **Необходима нормировка данных!**

→ **Нормировка –** это корректировка значений в соответствии с некоторыми функциями преобразования, с целью сделать их более удобными для сравнения.

4. Нормировка данных и повторная правильная (!) визуализация

```
plt.plot(days, views, 'bo-')
plt.plot(days, 100 * downloads, 'go-.')
plt.legend(('Visit site', 'Download product'),
           loc='upper right')
plt.xlabel('time (days)')
plt.ylabel('number of persons')
```



Решение задачи

Конверсия посетителей сайта
Предметная область: интернет-маркетинг

5. Поиск ежедневной и средней конверсии (%) посетителей сайта

```
everyDayConv = np.array(downloads / views)
print(everyDayConv)
averageConv = 100 * sum(downloads)/sum(views)
print(averageConv)
```

Как понять, что можно строить модель линейной регрессии?

1. **Построить диаграмму рассеяния**
2. **Найти коэффициент корреляции**

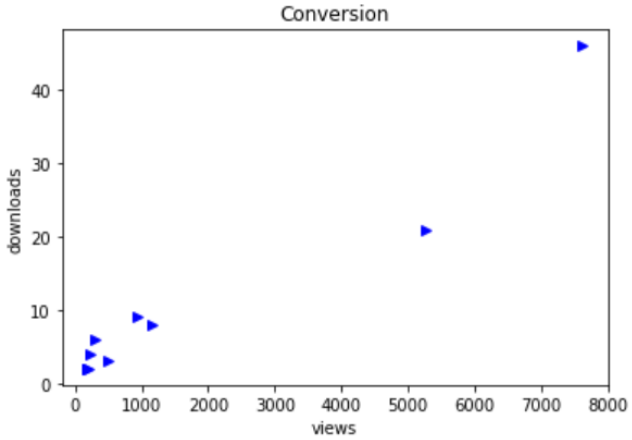
Диаграмма рассеяния — математическая диаграмма, изображающая значения двух переменных в виде точек на плоскости.

Решение задачи

Конверсия посетителей сайта
Предметная область: интернет-маркетинг

6. Построение диаграммы рассеяния

```
plt.plot(views, downloads, 'b>')
plt.xlabel('views')
plt.ylabel('downloads')
plt.title('Conversion')
```



7. Вычисление коэффициента корреляции

```
np.corrcoef(views, downloads)
```

```
array([[1.        , 0.9694434],
       [0.9694434, 1.       ]])
```

Можно строить модель линейной регрессии!

8. Вычисление статистических показателей исходных данных

```
# Среднее значение
print(np.average(views))
# Дисперсия
print(np.var(views))
# Стандартное отклонение
print(np.std(views))
# Минимальное и максимальное значение
print(np.min(views))
print(np.max(views))
```

Задание! Вычислите статистические показатели для downloads!

Решение задачи

Конверсия посетителей сайта
Предметная область: интернет-маркетинг

9. Построение модели линейной регрессии средствами *Scikit-Learn*

```
from sklearn import  linear_model
reg = linear_model.LinearRegression()

x_train = views[:, np.newaxis]
y_train = downloads[:, np.newaxis]

reg.fit(x_train, y_train)
```

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

$$w = (w_1, \ldots, w_p) \qquad w_0$$

y = a * x + b

10. Вывод коэффициентов построенной линии регрессии

```
print("a = ", reg.coef_)
print("b = ", reg.intercept_)
```

11. Проверка качества регрессионной модели R^2 **Коэффициент детерминации** [0, 1]

```
print(reg.score(x_train, y_train))
```

12. Предсказание значений по построенной модели

```
result = reg.predict([[300]])
print(result)
```

result – предсказание модели о том, сколько раз скачают программный продукт с сайта, если 300 человек посетят рассматриваемый сайт.

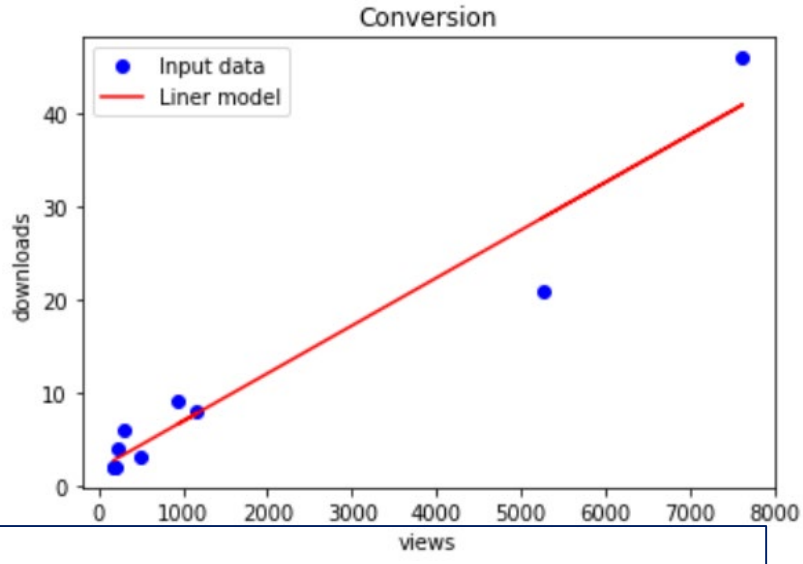
Решение задачи

Конверсия посетителей сайта
Предметная область: интернет-маркетинг

13. Визуализация модели линейной регрессии

```
wsite_y_pred = reg.predict(x_train)
plt.plot(x_train, wsite_y_pred)
```

Задание! Визуализировать линию регрессии на диаграмме рассеяния исходных данных!



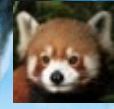
14. Ответить на следующие вопросы, сделав предсказание по модели линейной регрессии.

Задание!

1. При количестве посещений сайта в 8000 человек, сколько планируется получить зарегистрировавшихся (или скачавших) программный продукт?

2. Для обеспечения 500 скачиваний продукта, сколько человек должны зайти на сайт?

pandas – библиотека Python для обработки и анализа данных*



Формат данных CSV (comma-separated values)

```
import pandas as pd
```

Пример

Можно читать данные из CSV файла с помощью функции **read_csv** :

- По умолчанию предполагается, что поля разделены запятыми.

```
data = pd.read_csv('data/president_heights.csv')
data
heights = np.array(data['height(cm)'])
print(heights)
```

Задание 1:

- Вывести статистические показатели: средний рост и стандартное отклонение, минимальный и максимальный рост.
- Подсчитать количество президентов, чей рост превышает **170** см.
- Подсчитать количество президентов, чей рост превышает **170** см, но меньше **190 см.**

heights > 170

```
np.count_nonzero(heights > 170)
```

*Python Data Analysis Library: https://pandas.pydata.org/

Формат данных CSV (comma-separated values)

Задание 2:

Построить гистограмму роста президентов.

Написать собственную функцию, реализующую вычисление гистограммы (аналог функции `np.histogram`, с которой сравнить полученные результаты)

Воспользуйтесь:

- `np.linspace(…)`
- `np.zeros_like(…)`
- `np.searchsorted ()`
- `np.add.at ()`

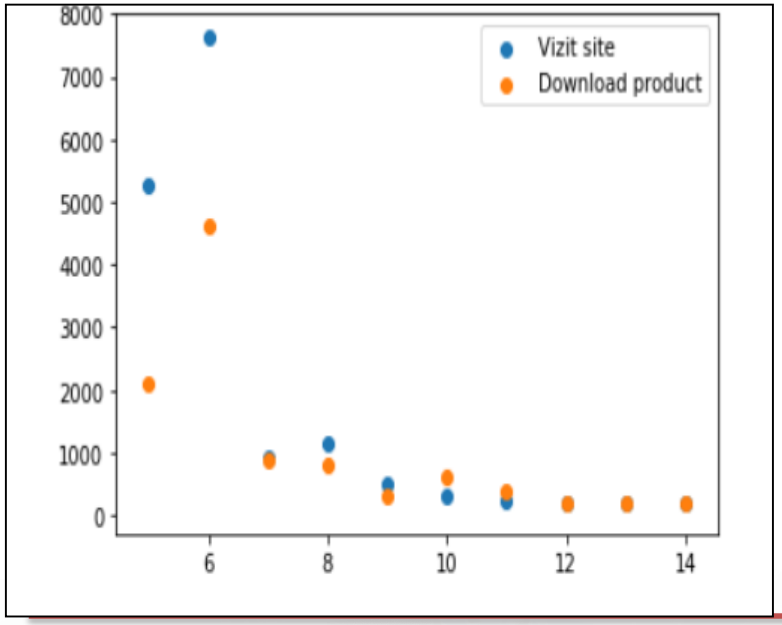
Задание 3:

Создать CSV файл данные по конверсии сайта, напечатать таблицу.

*Python Data Analysis Library: https://pandas.pydata.org/

Задачи машинного обучения. Корреляция и линейная регрессия.

Задача 1. Исследовать зависимость количества скачиваний программного продукта от количества посещений сайта.



К практическому занятию 1:

- Корреляция
- Метод наименьших квадратов (вывод)
- Линейная регрессия

Набор данных по продажам: взять из [1].

[1] Данные из Примера: конверсия посетителей сайта: https://habr.com/ru/company/nerepetitor/blog/250633/

Задачи машинного обучения. Корреляция и линейная регрессия.

Реализация средствами Python + NumPy

Реализация средствами Python + NumPy+Scikit-Learn

matplotlib – для визуализации данных, результатов и т.д.

Можно использовать **seaborn:** statistical data visualization

[1] Данные из Примера: конверсия посетителей сайта:
https://habr.com/ru/company/nerepetitor/blog/250633/

Задачи машинного обучения. Корреляция и линейная регрессия.

К практическому заданию:

- Построить зависимость зарегистрировавшихся на сайте от количества посещений

Для визуализации статистических данных можно использовать библиотеку **Seaborn [2,3]**

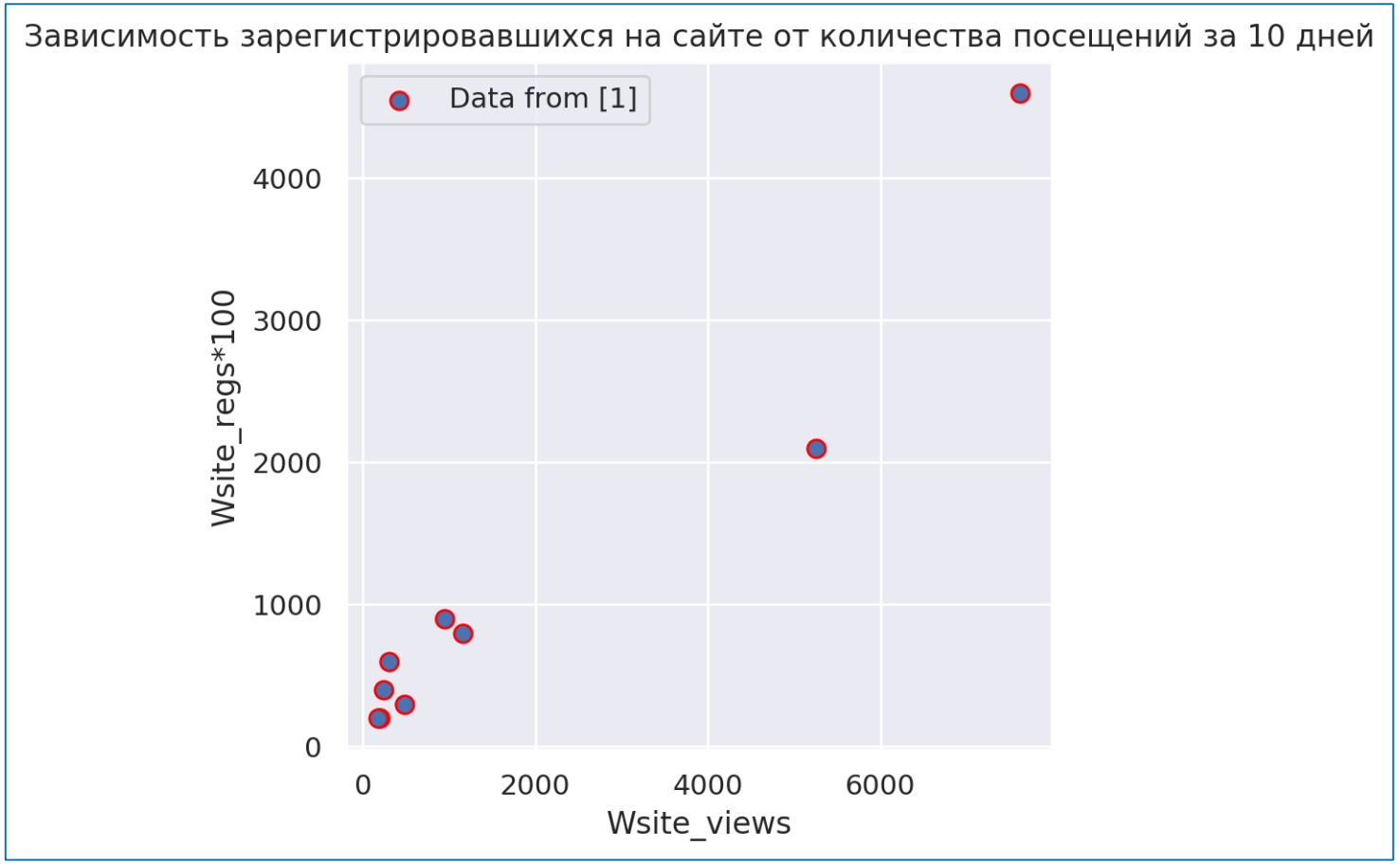
```
# use seaborn plotting defaults
import seaborn as sns; sns.set()
plt.figure(figsize=(5, 5), dpi=200)
plt.scatter(Wsite_views, Wsite_regs*100, edgecolor="red", s=50,
cmap='coolwarm', label="Data from [1]");
plt.ylabel("Wsite_regs*100")
plt.xlabel("Wsite_views")
plt.title("Зависимость зарегистрировавшихся на сайте от количества
посещений за 10 дней")
plt.legend()

plt.show()
```


[2] seaborn: statistical data visualization: https://seaborn.pydata.org/

[3] Python Seaborn Tutorial For Beginners

Задачи машинного обучения. Корреляция и линейная регрессия.



[2] **seaborn: statistical data visualization:** https://seaborn.pydata.org/

[3] **Python Seaborn Tutorial For Beginners**