

## Лабораторная работа 1,2

Реализовать содержимое функций, декларированных в репле: <https://repl.it.com/@zhukov/ClassicIntelligentEvaluations>

Разместить ссылку на собственный repl.it с вашей реализацией.

Для просрочивших сдачу задание требуется выполнить задание указанное в файле `ADDITIONAL_TASK.md`.

## Лабораторная работа 3, 4

Выполнить [лабораторную работу 3, 4](#), опубликованную на repl.it (сделать форк репла) или скопировать код к себе локально. Вариант: сделать копию [Google Colab блокнота](#) документ и выполнить её там).

При выполнении в Colab не забудьте выгрузить файл train.csv из repl.it или курса Moodle и загрузить этот файл в папку рядом с каталогом **sample\_data** (см. [скриншот](#) ниже) в Google Colab.

Выполните все задания и сделайте тесты для каждой написанной функции как показано в примере.

Для вычисления заданий **3, 4, 5, 6, 7, 8** используйте тип данных float с точностью два знака в дробной части.

**Часть 2.** Для набора данных из лабораторной работы 1 посчитать средние значения, медианы, максимальные и минимальные значения для столбцов Offline Spend, Online Spend.

Результаты работы опубликовать в виде бордов на repl.it и/или Colab.

### Примечание

*На случай блокировок иметь резервные копии ВСЕХ заданий в трех местах: в repl.it, Google Colaboratory и локально на жестком диске.*

## Лабораторная работа 5

Описание задания и стартовый код расположены в борде:

<https://colab.research.google.com/drive/1D50x9OOclUKL3BepB4VL2vwOCirzLTT9?usp=sharing>

Решение задачи регрессии.

Есть файл web\_traffic.tsv, который представляет статистику посещений веб-сервера компании и нашей задачей является предсказание значения посещения, для того, чтобы выделить в нужный момент времени дополнительную мощность сервера и не допустить его перегрузки.

Слева находится индекс или номер часа по порядку, в который производился замер посещаемости, а справа - количество запросов сайта за этот час.

## Лабораторная работа 6

Выполнить с использованием Google Colab (<https://colab.research.google.com/>).

Создайте документ и разместите в нём код, находящийся внизу. Загрузите файл с данными о ценах на недвижимость в колаб.

На основе кода ниже и данных, размещенных выше, реализуйте:

1. Визуализацию данных.
2. Линейную модель, которая учитывала бы только размер жилья.
3. Полиномиальную модель (степени 2 и 3), учитывающую только размер жилья.
4. Линейную модель (с помощью scikit-learn), которая учитывала бы размер жилья и количество комнат (см. *Примечание 1*)
5. Предскажите значения для двух объектах недвижимости с использованием этих трех моделей: 1650,3; 2200,4.
6. В ответе к лабораторной работе **и в колабе представьте** предсказанные значения стоимости объектов недвижимости для всех построенных моделей.
7. Оцените ошибку для созданных моделей. Опишите какая ошибка больше, а какая меньше и укажите причину.

```
%%capture

!wget https://www.dropbox.com/s/213etxuwv5nfo07/ex1data2.txt?dl=1 # этот пункт можно не
выполнять,

# данные лежат в Moodle: Источник данных ИСР 1.3.

import numpy as np

import matplotlib.pyplot as plt


data = np.genfromtxt("web_traffic.tsv", delimiter="\t")
x = data[:,0]
y = data[:,1]


x = x[~np.isnan(y)]
y = y[~np.isnan(y)]
```

```
f1p, residuals, rank, sv, rcond = np.polyfit(x, y, 1, full=True)
f1 = np.poly1d(f1p)
fx = np.linspace(0, x[-1], 500)

plt.scatter(x, y, s=10)
plt.plot(fx, f1(fx), linewidth=1.0, color='r')
plt.title('Трафик веб-сайта за последний месяц')
plt.xlabel("время")
plt.ylabel("запросы/час")
plt.xticks([w*7*24 for w in range(10)],
            ["неделя %i" % w for w in range(10)])
plt.autoscale(tight=True)
plt.grid(True, linestyle="-", color='0.8')
plt.show()
```

### Примечание 1

Следует заметить, что метод `polyfit` подходит только для создания модели, предсказывающей значение целевого параметра по **одной** переменной (мы используем размер дома). Однако, в этом пункте требуется создать модель, которая будет учитывать **и размер дома, и количество комнат**. Поэтому `polyfit` нам не подходит.

Для выполнения этого пункта следует использовать пакет `scikit learn` и оттуда модуль [LinearRegression](#). Здесь может возникнуть путаница в понятиях "полиномиальная" модель находится в блоке `LinearRegression`? Да, "полиномиальность" здесь - просто характеристика целевой функции модели. При этом модель всё равно предполагает, что целевой параметр и фичи модели **линейно** связаны — поэтому «линейная».

По ссылкам ниже вы найдете примеры использования этой модели:

- [https://scikit-learn.org/stable/auto\\_examples/linear\\_model/plot\\_ols.html#sphx-glr-auto-examples-linear-model-plot-ols-py](https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html#sphx-glr-auto-examples-linear-model-plot-ols-py)
- <https://towardsdatascience.com/polynomial-regression-bbe8b9d97491>
- <http://espressocode.top/python-implementation-of-polynomial-regression/>