## Множественная линейная регрессия на чистых данных

### Цель работы

Познакомиться с основными приемами работы с множественными регрессионными моделями с использованием библиотеки sklearn.

### Задания для выполнения

1. Ваша задача - построить предиктивную модель предсказания целевой переменной в зависимости от значения нескольких факторов (признаков).
2. Загрузите датасет [отсюда](https://drive.google.com/drive/folders/1kWOOgTbu7MnfXg1hh5aAYGsBRePhkvGV?usp=sharing). Вам понадобится два файла - отдельно с признаками (помечен x) и значениями целевой переменной (помечен y). Файлы не содержат заголовков столбцов и строк.
3. Сделайте количественное описание датасета: количество точек данных, признаков, статистику по каждому признаку.
4. Разделите все наблюдения на обучающую и тестовую выборки в соотношении 80-20.
5. Для целевой и каждого из признаков постройте парную регрессионную модель. Визуализируйте линию регрессии и исходные данные на графике. При помощи коэффициентов оцените силу связи.
6. Постройте модель множественной линейной регрессии целевой переменной от всех показателей. Оцените качество регрессии, сделайте вывод.

### Методические указания

1. Отчет по лабораторной работе сдается в виде файла интерактивного ноутбука Jupyter notebook и должен включать весь программный код, реализующий задания лабораторной работы, подробные пояснения, обоснования и выводы там, где это требуется в виде текстовых ячеек.
2. Для сведения статистических данных можно воспользоваться средствами библиотек sklearn, numpy и pandas, либо табличным редактором Microsoft Excel. В любом случае, к отчету по лабораторной работе следует приложить как исходные данные без обработки, так и итоговый датасет, использующийся при обучении.
3. Для диагностики обучения регрессионных моделей рекомендуется строить кривые обучения. Используйте [пример](http://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html) из документации sklearn как образец.
4. Для оценки качества построенной регрессии можно использовать целый ряд показателей. Одним из самых простых и универсальных является коэффициент детерминации (R2-score). Как использовать его читайте в [официальной документации sklearn](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html#sklearn.metrics.r2_score).

Приведем пример создания модели парной линейной регрессии средствами библиотеки scikit learn:

import numpy as np

import pandas as pd

from sklearn import linear\_model

Импорт таблицы

X = pd.read\_csv('1\_x.csv')

Y = pd.read\_csv('1\_y.csv')

Создаем пустую модель:

reg = linear\_model.LinearRegression()

Обучение машины с помощью метода fit(X, Y) - расчет параметров для указанных данных:

reg.fit(X, Y)

Получаем коэффициенты bn:

reg.coef\_

Метод predict() рассчитывает теоретические значения признака

reg.predict([[data['x1'][1], data['x2'][3], data['x3'][0]]])

Методом score можно узнать значение метрики эффективности полученной модели:

reg.score(X, Y)

### Контрольные вопросы

1. Чем множественная регрессия отличается от парной?
2. Какую функцию ошибки использует модель линейной регрессии?
3. Какие метрики эффективности можно применять для оценки регрессионных моделей?
4. Какие еще регрессоры есть в библиотеке sklearn?

### Дополнительные задания

1. Для целевой переменной и каждого из признаков постройте полиномиальную регрессию второго, третьего и десятого порядков. Сделайте вывод о значимости регрессии.
2. Постройте сводную таблицу показателей точности всех построенных моделей по тестовой и обучающей выборке. Сделайте вывод, какие модели переобученнные, какие - недообученные.
3. Используйте другие метрики качества регрессионной модели в дополнение к коэффициенту детерминации.
4. (\*) Повторите те же действия на датасетах из [данной](https://drive.google.com/drive/folders/1zwBQtAhXO5u1J3e4XCo8N9_Az2YcTS0S?usp=sharing) папки. Замерьте время обучения на разных датасетах и количестве признаков. Сделайте вывод о зависимости скорости обучения от объема обучающей выборки.