**ЧАСТЬ 1**

***Исходные данные:*** данные о стоимости домов и параметры каждого дома (файл train.csv). Описание параметров представлено в файле data\_description.txt.

**Задача:** построитьлинейную модель задачи, обучить ее на тренировочном наборе данных.

**Рекомендации к выполнению:**

Разбить исходный датасет случайным образом на три (train, cross-validation и test) или два набора (train и test).

Обучение модели провести на тренировочном наборе, посчитать ошибку на тестовом.

Рекомендовано использовать RMLSE (Root Mean Squared Logarithmic Error), L\_1 регуляризацию (lasso regression), избавление от категориальных признаков (в случае Python - get\_dummies). Рекомендовано выполнять на языке Python (numpy, pandas, sklearn...).

**ЧАСТЬ 2**

***Исходные данные:*** данные о стоимости домов и параметры каждого дома (файл train.csv). Описание параметров представлено в файле data\_description.txt.

**Задача:** построитьXGBoost модель задачи, обучить ее на тренировочном наборе данных, провести процедуру гипероптимизации (оптимальный подбор параметров модели) с помощью встроенных библиотек.

**Рекомендации к выполнению:**

Разбить исходный датасет случайным образом на три (train, cross-validation и test) или два набора (train и test).

Обучение модели провести на тренировочном наборе, посчитать ошибку на тестовом.

Рекомендовано использовать RMLSE (Root Mean Squared Logarithmic Error), избавление от категориальных признаков (в случае Python - get\_dummies). Рекомендовано выполнять на языке Python (numpy, pandas, sklearn, xgboost, hyperopt (fmin)...).

**Примечание:** использовать реализованные модели, не нужно писать с нуля.

**Контрольные вопросы:**

1) Линейная регрессия.

2) Полиномиальная регрессия.

3) критерий качества, функция ошибки в линейных моделях.

4) категориальные признаки

5) гипероптимизация параметров

6) метод наименьших квадратов