Домашнее задание

Нужно реализовать rest api на базе flask (пример <https://github.com/fimochka-sudo/gb_flask_hw_api>)

По шагам:

1. выбрать себе датасет (который интересен или нравится больше всего), сделать pipeline (преобразования + модель), сохранить его на диск. Если не хочется пайплайн, то можно без него, но так вам же будет удобнее потом вызывать его из кода сервиса.
2. установить удобную для себя среду разработки (pycharm прекрасен - <https://www.jetbrains.com/pycharm/>)
3. научиться создавать там виртуальное окружение для python (в правом нижнем углу есть add interpreter)
4. для вашего проекта вам понадобится requirements.txt с пакетами. Можно за основу взять такой файл из проекта выше. Для его установки прям в pycharm можно открыть терминал и сделать pip install -r requirements.txt (находясь в корне проекта конечно же при этом)
5. завести себе аккаунт на guthub (если его еще нет). У самого github есть такой "hello world" по работе с ним - <https://guides.github.com/activities/hello-world/>
6. научиться запускать gunicorn (установить его конечно). Пример для проекта выше: gunicorn -w 5 -b 127.0.0.1:5000 run\_server:app
7. итоговый проект должен содержать: 1) каталог models/ (здесь модель-пайплайн предобученная) 2) файл run\_server.py (здесь основной код flask-приложения) 3) requirements.txt (список пакетов, которые у вас используются в проекте) 4) README.md (здесь какое-то описание, что вы делаете, что за данные и т.д)
8. (<b>Опционально</b>): front-end сервис какой-то, который умеет принимать от пользователя введеные данные и ходить в ваш api. На самом деле полезно больше вам, т.к если ваш проект будет далее развиваться (новые модели, интересные подходы), то это хороший пунктик к резюме и в принципе - строчка в портфолио)

Полезные ссылки:

1. датасеты (для полета мысли): <https://www.kaggle.com/datasets>
2. конкурс Сбербанка по недвижимости (можно этот набор данных также взять и обучить модель предсказывать стоимость жилья - неплохой такой сервис может получиться) - <https://www.kaggle.com/c/sberbank-russian-housing-market/data> Там же и ноутбуки с разными подходами есть.
3. презентация с занятия - <https://hackmd.io/@AndreyPhys/flask-ml-api#/>
4. минималистичный пример связки keras/flask <https://blog.keras.io/building-a-simple-keras-deep-learning-rest-api.html> для определения класса картинки
5. неплохой такой пример связки docker/flask - <https://cloud.croc.ru/blog/byt-v-teme/flask-prilozheniya-v-docker/>
6. <https://www.digitalocean.com/community/tutorials/how-to-build-and-deploy-a-flask-application-using-docker-on-ubuntu-18-04>

p.s. если проблемы с выбором датасета, то пишите пожалуйста - будем вместе думать)