

ML API

Реализация API и Telegram bot

Проект состоит из модуля “app”, в котором “mlmodels.py” реализует класс с CRUD, и “views.py”, в котором находятся классы, реализующие функционал API. А также файлов “bot.py”, “cli.py”, “config.py”, “helpers.py”, “log.py” и “requirements.txt”. В них находится логика работы telegram bot, запуска API, credentials, вспомогательные функции и логика работы логгера и зависимости соответственно. В папке data будут храниться данные.

Описание API

Модели, использованные в работе:

- {
- 1: Logistic Regression, (1)
- 2: Decision Tree
- }

В данной работе работа строится на 4 основных urls:

- http://0.0.0.0:5000/api/ml_models
- http://0.0.0.0:5000/api/ml_models/<int:id>
- http://0.0.0.0:5000/api/ml_models/<int:id>/service
- http://0.0.0.0:5000/api/ml_models/<int:id>/retrain

Рассмотрим методы, реализованные в каждой из ручек.

1) http://0.0.0.0:5000/api/ml_models

В этой ручке реализован метод GET: Отображает доступные для обучения модели и гиперпараметры.

2) http://0.0.0.0:5000/api/ml_models/<int:id>

В этой ручке реализован метод GET. Он отображает обученные модели в формате:

{1: {Обученные}, 2:{модели}}, где {1, 2} - ID моделей.

3) http://0.0.0.0:5000/api/ml_models/<int:id>/service

Это ручка предназначена для обучения моделей и получения предсказаний.

В ней реализованы методы:

- GET - для получения информации о доступных моделях
- POST - для получения предсказания модели
- PUT - для обучения модели
- DELETE - для удаления обученных моделей

4) http://0.0.0.0:5000/api/ml_models/<int:id>/retrain

Это ручка предназначена для переобучения моделей.

В ней реализованы:

- GET - для получения информации о доступных моделях
- PUT - для переобучения выбранной модели

Описание Telegram Bot

Имя пользователя: *@test_mlapi_bot*

Бот работает за счёт 5-ти основных команд:

1) /start - Отображение справки.

2) /models - Отображение, моделей, доступных для обучения и списка их гиперпараметров.

3) /train - Начало обучение и инструкция для обучения.

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
5.1	3.5	1.4	0.2	0
4.9	3.0	1.4	0.2	0
4.7	3.2	1.3	0.2	0
4.6	3.1	1.5	0.2	0
5.0	3.6	1.4	0.2	0
5.4	3.9	1.7	0.4	0

Чтобы обучить модель, нужно после команды `/train` загрузить данные в формате csv. Образец данных:

После загрузки данных, для обучения модели нужно ввести “train {ID} {HYPERPERS}”, где {ID} - ID модели из (1), {HYPERPERS} - гиперпараметры модели (Пример {“max_iter”:1000}). Последняя колонка должна называться “target”

4) `/predict` - Начало прогнозирования и инструкция для прогнозирования

Для получения предсказания, после команды `/predict` нужно ввести “predict {ID} {Номер модели} {data}”, где {data} - вектор признаков через пробел (Пример: predict 1 0 5.1 3.5 1.4 0.2)

5) `/retrain` - Начало переобучения и инструкция для переобучения

Для переобучения модели после команды `/retrain` нужно загрузить данные в формате как в обучении. После этого ввести retrain {ID} {Номер модели} {Hypers} (Пример: retrain 1 0 {“max_iter”:2000})

Алгоритм действий

Сначала запускаем API командой `python cli.py`. И API можно пользоваться.

Затем запускаем бота командой `python bot.py`. И ботом можно пользоваться.

Примечания

При работе с ботом желательно следовать инструкциям, так как там не отработаны исключительные ситуации.