

Описание структуры проекта:

Первичные эксперименты:

- `scraping.ipynb`: несмотря на то, что из-за ограничений на скрепинг в Твиттере (можно просматривать не более 500 твитов в день) мы решили использовать уже собранные данные с Kaggle, наш датасет можно дополнять новыми данными, которые собираются в том же формате в данном ноутбуке
- `labeling.ipynb`: собранные данные мы размечаем с помощью предобученных моделей
- `eda_elon_musk.ipynb`: EDA и предварительная обработка данных. Также только предобработку отдельно от EDA можно найти в ноутбуке `preprocessing.ipynb`
- `checkpoint3_update.ipynb`: эксперименты с задачей классификации на наших данных

Клиент-серверное приложение

Функционал

- Работа с обработанным датасетом твитов Илона Маска.
- Загрузка датасета и выдача статистики по нему.
- Выбор из трёх моделей:
 - Логистическая регрессия
 - Дерево решений
 - Метод ближайших соседей
- Эксперименты с обучением моделей с настройкой гиперпараметров.
- Интерактивные кривые обучения для эксперимента.
- Инференс на новых данных.

Установка и запуск (`docker-compose.yml`)

```
sudo docker compose build
docker compose up -d
```

Сервер доступен на порту 8321. Клиент на 8501.

Описание API

Данный API предоставляет функционал для работы с моделями машинного обучения. Он включает в себя методы для установки активной модели, тренировки модели, получения списка доступных моделей и выполнения предсказаний. Вот описание каждого из доступных методов API:

1. Установка модели

- Метод: POST
- Эндпоинт: `/set/{model_id}`
- Параметры:
 - `model_id` (int): Идентификатор модели, которую необходимо сделать активной.

- Ответ: `ModelListItem`
 - `id` (int): Идентификатор установленной модели.
 - `model_type` (str): Тип модели.
 - `description` (str): Описание модели.
- Ошибки:
 - 422: Если модель с указанным `model_id` не найдена.

2. Тренировка модели

- Метод: POST
- Эндпоинт: `/fit`
- Тело запроса: `FitRequestItem`
 - `X_train` (`List[List[float]]`): Тренировочные данные.

- `y_train` (`List[float]`): Метки тренировочных данных.
- `hyperparameters` (`Dict[str, Any]`): Гиперпараметры для настройки модели.
- `val_dataset` (`ValDataset` | `None`): Валидационный набор данных.
- Ответ: `FitResponseItem`
 - `auc_roc` (`float`): Значение AUC-ROC на тренировочных данных.
 - `tpr_train` (`List[float]`): True Positive Rate на тренировочных данных.
 - `fpr_train` (`List[float]`): False Positive Rate на тренировочных данных.
 - `auc_roc_val` (`float` | `None`): Значение AUC-ROC на валидационных данных (если предоставлены).
 - `tpr_val` (`List[float]` | `None`): True Positive Rate на валидационных данных.
 - `fpr_val` (`List[float]` | `None`): False Positive Rate на валидационных данных.
- Ошибки:
 - 422: Если активная модель не установлена или обучение заняло слишком много времени.

4. Получение списка моделей

- Метод: GET
- Эндпоинт: `/models`
- Ответ: `List[ModelListResponseItem]` Список моделей, каждая из которых содержит:
 - `id` (`int`): Идентификатор модели.
 - `model_type` (`str`): Тип модели.
 - `description` (`str`): Описание модели.

5. Предсказание

- Метод: POST
- Эндпоинт: `/predict`
- Тело запроса: `PredictRequest`
 - `X` (`List[List[float]]`): Данные для предсказания.
- Ответ: `PredictionResponseItem`
 - `predict` (`List[int]`): Предсказанные значения.
 - `predict_prob` (`List[float]`): Вероятности предсказания для каждого класса.
- Ошибки:
 - 422: Если активная модель не установлена.

Дополнительно

- Обработчики ошибок: если модель не установлена или происходят ошибки во время тренировки или предсказания, API возвращает ошибку с кодом 422 и соответствующим сообщением.

Инструкция пользователя (клиент)

1. Откройте браузер по адресу: <http://localhost:8501/>

Deploy

Upload your dataset

Choose a file

Drag and drop file here

Limit 200MB per file • CSV

Browse files

Models

List models

Run Experiment

Valid fraction

0.10

Machine Learning Dashboard

2. Загрузите датсет (файл elon_musk_tweets_after_eda.csv).

Deploy

Upload your dataset

Choose a file

Drag and drop file here

Limit 200MB per file • CSV

Browse files

elon_musk_tweets_afte...

3.6MB

Models

List models

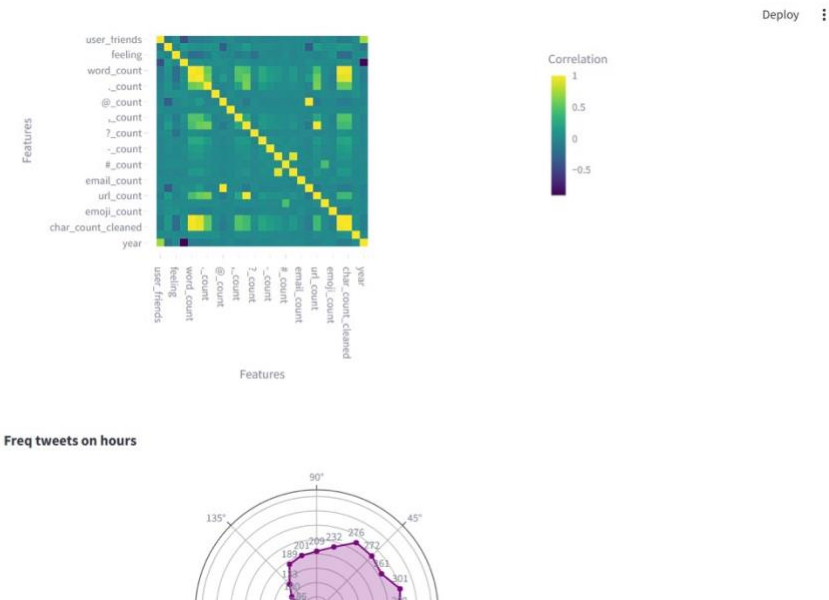
Run Experiment

Valid fraction

0.10

0.10

0.90





Upload your dataset

Choose a file

Drag and drop file here

Limit 200MB per file • CSV

Browse files

 elon_musk_tweets_afte... 

3.6MB


Models

List models

Run Experiment

Valid fraction

0.10



0.10 0.90

Run

Upload inference data



Наибольшее количество твитов Илон Маск пишет поздним днём и вечером. За июль 2022 - июнь 2023 гг. Маск написал

- 1) 419 постов в 17 часов;
- 2) 378 постов в 19 часов;
- 3) 350 постов в 18 и 20 часов.

В целом в 17:00-01:00 Илон Маск пишет около 300-400 твитов, т. е. он предпочитает писать твиты в вечернее-ночное время.

Меньше всего постов Илон Маск пишет утром и ранним днём: за аналогичный период в 10 часов Маск написал 100 твитов, в 11 часов - 55, а в 12 часов - 77.

3. Нажмите на кнопку "List models", чтобы получить список моделей на сервере. При выборе модели откроется список с гиперпараметрами доступными для модели.

The screenshot shows the Google Colab interface. At the top, there is a 'Browse files' button. Below it, a file named 'elon_musk_tweets_afte...' is shown with a size of 3.6MB. The 'Models' section is highlighted, featuring a 'List models' button. Below this, the 'Select model:' dropdown menu is open, showing 'DT' as the selected model. A 'Select' button is located below the dropdown. At the bottom, the 'Run Experiment' section is visible, showing a 'Valid fraction' of 0.10 with a corresponding progress bar.



Criterion

gini

Max Depth

16

1 50

Min Samples Split

4

2 10

Set Hyper

4. После настройки гиперпараметров установите размер валидационной выборки и запустите

тренировку кнопкой "Run".

Models

List models

Select model:

DT

Select

Run Experiment

Valid fraction

0.26

0.100.90

Run

Upload inference data

Choose an inference file

RUNNING...

Stop

Deploy

Min Samples Split

2

4

10

Set Hyper

In progress...

{

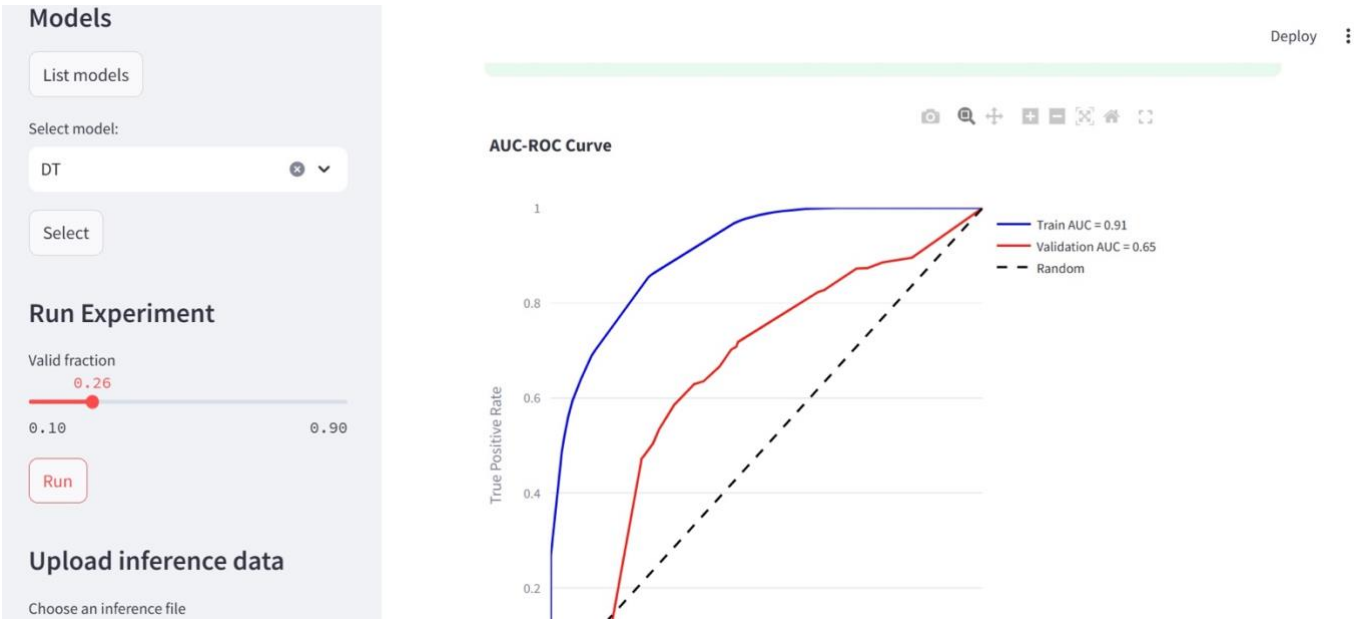
"criterion": "gini"

"max_depth": 16

"min_samples_split": 4

}

5. Если обучение прошло успешно отобразится интерактивная кривая ROC-AUC для обучающей и валидационной выборки.



6. Так же можно загрузить данные и провести инференс натренированной модели.

