

Проект курса ML1

Анализ веб-документов

- Основная идея:

Выделяем признаки на основе текста, содержащегося на странице (не используем url, метаданные и т. п.)


Поэтому была проведена небольшая работа с контентом — стемминг (+ регулярки, стоп слова)

Основная трудность — моя машина плохо справляется с большим объемом данных, так что в итоге пришлось работать только с заголовками
=((

- Идея 1:

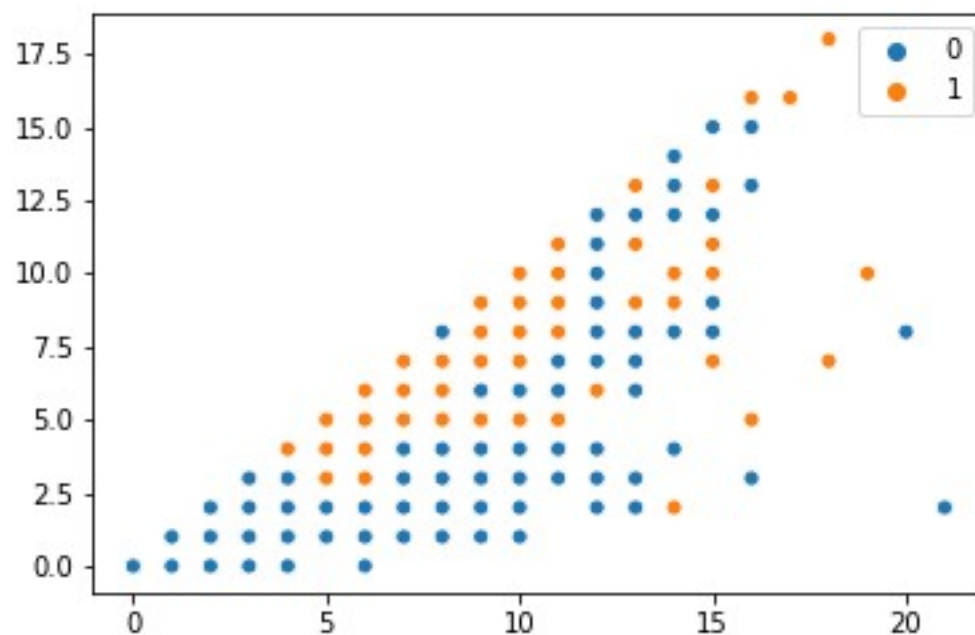
Обрабатываем каждую страницу:


- удаляем мета-данные, ссылки, стили и т. д.
- оставляем в тексте только русские и английские слова (регулярочки)
- удаляем стоп-слова
- стемминг
- получаем итоговый текст для каждой страницы, который нужно преобразовать в вектор

- 
- Для получения векторов использовался TfidfVectorizer
 - Проблема — все это очень долго, а времени оставалось мало
 - Классификаторы — LinearRegression, LogisticRegression
 - Итог: score на трейне 0.43
 - Была еще идея с n-граммами, но я решила поработать с заголовками, так как это быстрее

- Идея 2 (ну как идея, продолжение домашки №2):

Использовалась идея, предложенная преподавателем — использовать признаки, основанные на максимальном числе общих слов в заголовках. Добавила еще к этому признак — номер группы.



- 
- Из обработки делала только стемминг — это дало небольшой прирост score
 - Еще была идея к словам заголовков добавлять самые встречающиеся слова из текстов страниц, но тоже не хватило времени
 - Матрица с числом общих слов в заголовках была преобразована с помощью StandardScaler (для нормализации и стандартизации). Это было нужно для улучшения результатов SGDClassifier (линейные модели со стохастическим градиентным спуском)

- Параметры классификатора подбирала по roc_auc_score



- roc_auc_score для лучшей модели 0.72

- Порог подбирала по f1-мере для найденного классификатора



- f1_score для найденного порога 0.6

- Итог:

предсказываем, используя найденный классификатор и порог.

f1_score на лидерборде 0.63

