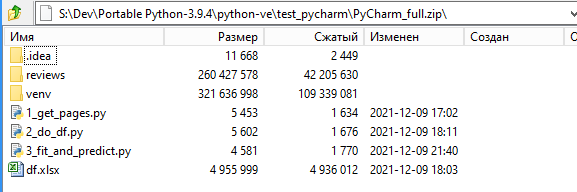
Задача

Выполнить парсинг отзывов с сайта [https://tabiturient.ru/sliv/n/?<page\_num>](https://tabiturient.ru/sliv/n/?%3cpage_num%3e)  в диапазоне с 12 по 4600 страницы. Сформировать таблицу с заданными полями по образцу. На основе полученных данных обучить алгоритм машинного обучения для прогнозирования тональности.

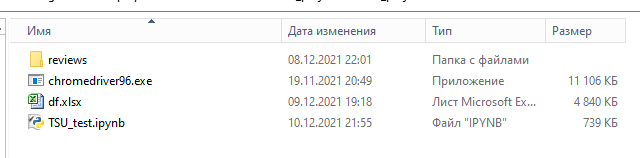
Итоги работы

Работа сдаётся в двух вариантах:

1. В PyCharm (т.к. код двух частей работы - сбор страниц и их парсинг - изначально писался здесь, затем, в соответствии с требованиями к сдаче проекта, был перенесён в jupyter notebook). Архив выхожен на гугл-диске: <https://drive.google.com/drive/folders/1ZlwpCl3SqEuMUG2Tu4ujomTzUwkmyAVL?usp=sharing>   
   По ссылке – два архива, полный, включая папку review со страницами отзывов и **папку виртуального окружения проекта venv** (содержимое полной версии на скрине ниже). В сокращённом архиве лежит только код (файлы .py в соответствии с этапами) и итоговая excel-таблица.



1. Jupyter notebook – основной формат сдачи, в соответствии с требованиями к тестовой работе. Архивы по той же ссылке. Такой же принцип деления – полная версия с папкой review и ChromeDriver’ом к Selenium и сокращённый вариант – только тетрадка и excel.



Краткое описание хода работы

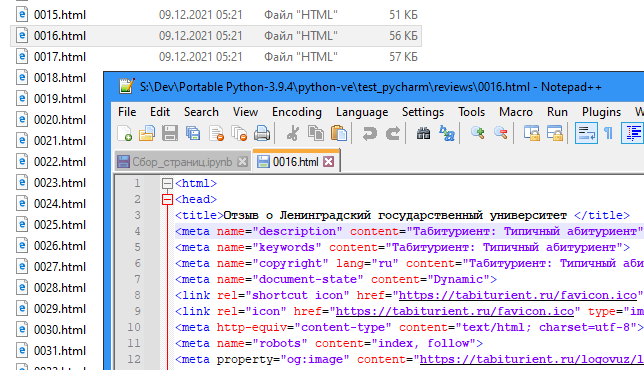
Работа разделена и выполнена в три этапа:

1. Сбор страниц с сайта-донора, сохранение их кода на локальный жёсткий диск.
2. Парсинг сохранённых страниц, извлечение требуемых данных.
3. Предобработка полученных данных, построение моделей машинного обучения.

Этап 1. Сбор страниц с отзывами

Сбор страниц реализован на двух функциях с помощью модуля requests. Первая функция пробегает все страницы в заданном диапазоне и сохраняет html-код на жёсткий диск. В случае ошибки (сбой соединения, отказ сайта и т.д.) номер страницы сохраняется в соответствующий список, и по завершении сканирования первично заданного диапазона управление передается второй функции, которая пробует повторно «достучаться» до страниц, из сохранённого списка, которые в первый раз загрузить не получилось.

Итог первого этапа – папка с файлами 0012-4600 c соответствующим html-кодом.

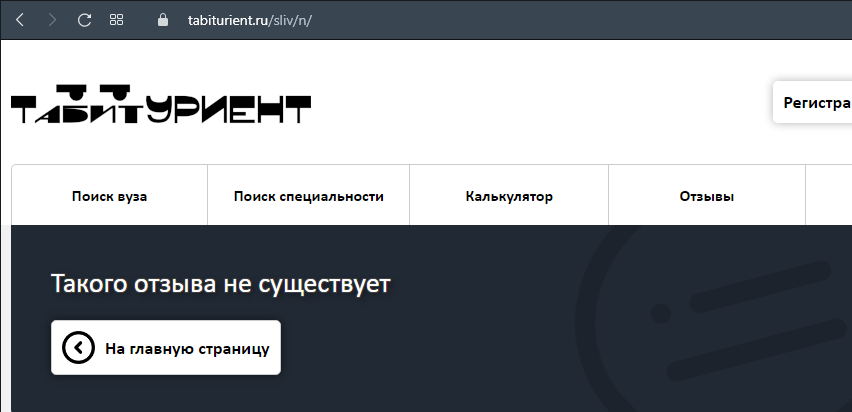


Этап 2. Парсинг файлов и формирование таблицы с необходимой информацией

Изначально хотелось работать с модулем Selenium. Причина – поддержка xpath, который позволяет максимально точно выбирать элементы страницы и который я неплохо знаю. Идея была реализована, но после запуска обнаружился чудовищный недостаток - очень медленная работа Selenium’а, чтобы обойти все страницы потребовалось несколько часов. Решено было переписать вторую часть на модуле bs4. Код для Selenium остался в ноутбуке, он рабочий, но закомментирован, чтобы не мешал в случае запуска всех ячеек.

Сложность, с которой пришлось столкнуться при реализации этого этапа – структура страницы оказалась не идентична для всего массива страниц, и пришлось выкручиваться с помощью if-ов.

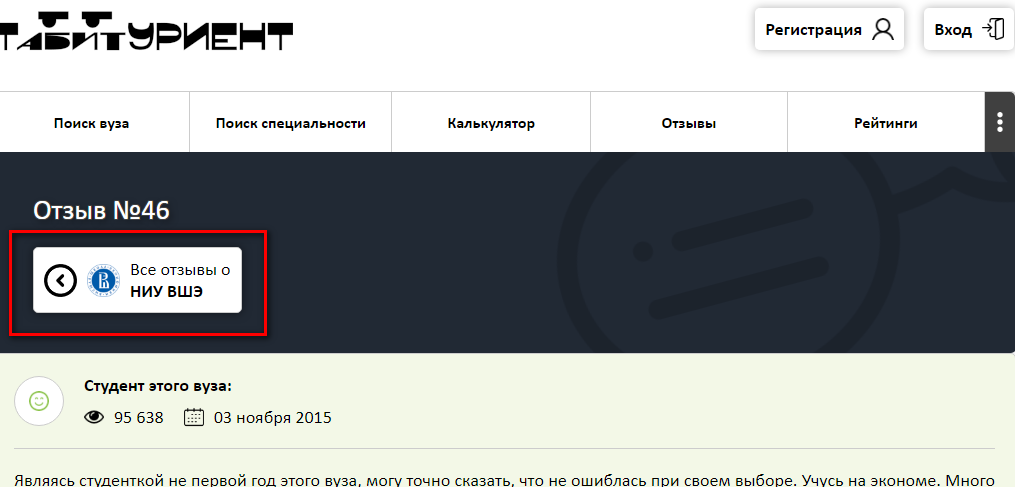
Во-первых, на большей части страниц отзывы отсутствовали.



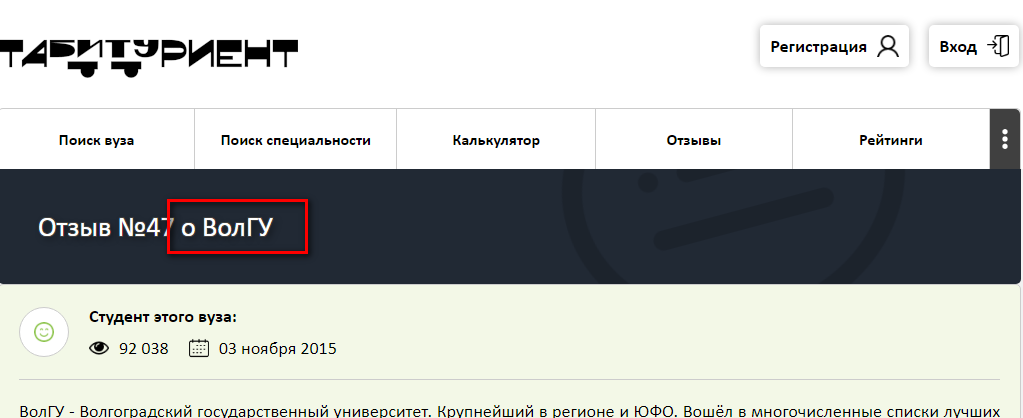
Скриншот страницы <https://tabiturient.ru/sliv/n/?1011>

Чтобы не терять информацию, в итоговой таблице я решил эти страницы сохранить, введя дополнительный признак «Статус» и присваивая для таких страниц статус «0». Страницы с наличиствующими отзывами получали статус «1».

Во-вторых, на части страниц, по какой-то причине наблюдались ошибки верстки и название вуза оказывалось не на «своём» месте. Под это пришлось подстраиваться.



«Нормальная» вёрстка: <https://tabiturient.ru/sliv/n/?46>



«Не нормальная» вёрстка: <https://tabiturient.ru/sliv/n/?47>

Некоторые отзывы написаны не на ВУЗ в целом, а на факультет или институт данного ВУЗа. Эту информацию я тоже решил оставить, т.к. она может быть использована как feature в некоторых моделях ML.

bs4 не поддерживает xpath, поэтому для «тонкой настройки» доступа к некоторым данным я использовал регулярные выражения.

Распарсивание страниц занимает некоторое время, поэтому, чтобы не потерять информацию в случае возможного сбоя, предусмотрено сохранение обработанных данных во временный файл через каждые 300 итераций.

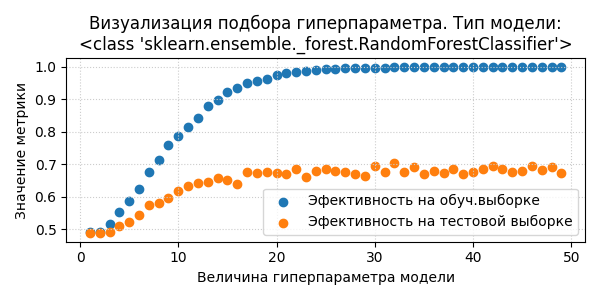
Результат второго этапа – excel-таблица c отзывами и их признаками.



Этап 3. Подготовка данных и ML

На третьем этапе сделан небольшой предварительный анализ данных, и косметические преобразования таблицы – числовые колонки приведены к формату int, колонка с датой – к формату datetime.date , отфильтрованы отзывы «в наличии», т.е. взяты только строки со статусом «1». Смысл большинства данных преобразований - больше в том, чтобы продемонстрировать навык обращения с инструментарием, т.к. в дальнейшем я не использовал результаты большинства этих преобразований.

В работе над третьим этапом я исходил из буквального понимания задачи – «обучить любой алгоритм машинного обучения для прогнозирования тональности» - т.е. был ориентирован на то, чтобы показать знание pipeline’a ML в целом, а не бороться за параметры модели с целью показать высокие результаты.



Я продемонстрировал знания основных этапов работы с языковыми моделями – тонизацию, очистку от стоп-слов и знаков препинания, стеммирование/ лемматизацию, векторизацию, обучение моделей классификации и кластеризации, а также модели определения тональности dostoevsky.

Каковы бы ни были результаты моего тестирования, задание было очень интересными и полезным. Спасибо.

