Даны два датасета с положительными и отрицательными постами Twitter, сделанных на русском языке. Требуется создать модель для определения тональности постов. Исходя из этого, необходимо:

1. **Определить необходимые поля и создать структуру набора данных.** Определить, какие атрибуты имеют наибольшее влияние на определение тональности текста, и оставить только их для последующего обучения. Также необходимо обосновать выбор дополнительных атрибутов и причину исключения каких-либо данных из исходного набора документов.
2. **Провести предварительную обработку текстов размеченного датасета**: очистка данных, удаление стоп-слов, специальных символов, лишних пробелов; перевод в нижний регистр; лемматизация/стемминг.
3. **Провести тематическое моделирование** положительных и отрицательных постов: определите слова, специфичные для каждого класса: облако слов, модели тематического моделирования.
4. **Векторизация**. Получить векторное представление текстов.
5. **Классификация**. Проведите обучение классификатора на полученных векторах. Рассмотреть не менее 3 моделей классификации. Проведите оценку качества моделей.
6. **Парсинг данных.** Для улучшения классификатора требуется обогатить набор данных. Для этого необходимо провести парсинг текстовых данных с сайта: собрать не менее 500 комментариев пользователей. Далее проведите классификацию полученных комментариев с помощью обученной модели. Проанализируйте корректность работы модели классификации, выведите вероятности прогноза. Необходимо забрать из выборки только те комментарии, в которых классификатор имеет оценку больше 0.8 или меньше 0.2 и добавить их в корпус размеченных текстов. Далее по новой выборке снова следует провести векторизацию, обучить модель. Из спарсенных данных убираем текст, добавленный в корпус размеченных текстов и проводим прогноз по оставшейся выборке. Проделайте этот процесс несколько раз.
7. **Разработка API.** Разработайте программный интерфейс для итоговой модели машинного обучения. API должен позволить приложению пользователя получать доступ к моделям для определения тональности комментариев пользователей.
8. **Разработка приложения.** Разработайте приложение с графическим интерфейсом, которое должно с помощью разработанного API показывать результат эмоциональной окраски введенных комментариев. Добавить вкладку со статистическими данными датасета (общее количество комментариев, средняя длина постов, доля положительных и отрицательных постов и т.д.) и визуализацией необходимых данных. Приложение также должно предоставлять справку по имеющимся командам и их параметрам.
9. **Программная документация.** Для разработанного приложения и API составьте программную документацию и руководство пользователя
10. **Оформить отчет.** Отчет должен охватывать результаты всей сессии, быть содержательный и выполнен профессионально.

Дан датасет с новостями на русском языке (https://github.com/yutkin/Lenta.Ru-News-Dataset/releases). Требуется создать модель для классификации новостей по категориям. Исходя из этого, необходимо:

1. **Определить необходимые поля и создать структуру набора данных.** Определить, какие атрибуты имеют наибольшее влияние на классификацию новостей и оставить только их для последующего обучения. Также необходимо обосновать выбор дополнительных атрибутов и причину исключения каких-либо данных из исходного набора документов.
2. **Провести предварительную обработку текстов размеченного датасета**: очистка данных, удаление стоп-слов, специальных символов, лишних пробелов; перевод в нижний регистр; лемматизация/стемминг.
3. **Провести тематическое моделирование** для каждой категории новостей: определите слова, специфичные для каждого класса: облако слов, модели тематического моделирования.
4. **Векторизация**. Получить векторное представление текстов.
5. **Классификация**. Проведите обучение классификатора на полученных векторах. Рассмотреть не менее 3 моделей классификации. Проведите оценку качества моделей.
6. **Парсинг данных.** Для улучшения классификатора требуется обогатить набор данных. Для этого необходимо провести парсинг текстовых данных с сайта: собрать не менее 500 новостей. Далее проведите классификацию полученных текстов новостей с помощью обученной модели. Проанализируйте корректность работы модели классификации, выведите вероятности прогноза. Необходимо забрать из выборки только те новости, в которых классификатор имеет оценку больше 0.8 или меньше 0.2 и добавить их в корпус размеченных текстов. Далее по новой выборке снова следует провести векторизацию, обучить модель. Из спарсенных данных убираем текст, добавленный в корпус размеченных текстов и проводим прогноз по оставшейся выборке. Проделайте этот процесс несколько раз.
7. **Разработка API.** Разработайте программный интерфейс для итоговой модели машинного обучения. API должен позволить приложению пользователя получать доступ к модели для определения категории новостей.
8. **Разработка приложения.** Разработайте приложение с графическим интерфейсом, которое должно с помощью разработанного API показывать спрогнозированную категорию новостей. Добавить вкладку со статистическими данными датасета (общее количество новостей, средняя длина постов, доля каждой категории новостей и т.д.) и визуализацией необходимых данных. Приложение также должно предоставлять справку по имеющимся командам и их параметрам.
9. **Программная документация.** Для разработанного приложения и API составьте программную документацию и руководство пользователя
10. **Оформить отчет.** Отчет должен охватывать результаты всей сессии, быть содержательный и выполнен профессионально.

Архив содержит файлы с информацией, полученной с сайта регистрации жалоб граждан. Один файл соответствует одной жалобе. Необходимо автоматизировать маршрутизацию обращений и жалоб граждан, поступающих с сайта «умного города». С точки зрения постановки задачи в терминах машинного обучения, – решить задачу классификации текстов.

В этом модуле необходимо сформировать набор данных, содержащий данные по всем жалобам.

1. **Определить необходимые поля и создать структуру набора данных.** Определить, какие атрибуты имеют наибольшее влияние на классификацию текстов и оставить только их для последующего обучения. Также необходимо обосновать выбор дополнительных атрибутов и причину исключения каких-либо данных из исходного набора документов.
2. **Провести предварительную обработку текстов размеченного датасета**: очистка данных, удаление стоп-слов, специальных символов, лишних пробелов; перевод в нижний регистр; лемматизация/стемминг.
3. **Провести тематическое моделирование** для каждой категории жалоб: определите слова, специфичные для каждого класса: облако слов, модели тематического моделирования.
4. **Векторизация**. Получить векторное представление текстов.
5. **Классификация**. Проведите обучение классификатора на полученных векторах. Рассмотреть не менее 3 моделей классификации. Проведите оценку качества моделей.
6. **Парсинг данных.** Для улучшения классификатора требуется обогатить набор данных. Для этого необходимо провести парсинг текстовых данных с сайта: собрать не менее 500 обращений пользователей. Далее проведите классификацию полученных текстов обращений с помощью обученной модели. Проанализируйте корректность работы модели классификации, выведите вероятности прогноза. Необходимо забрать из выборки только те общения, в которых классификатор имеет оценку больше 0.8 или меньше 0.2 и добавить их в корпус размеченных текстов. Далее по новой выборке снова следует провести векторизацию, обучить модель. Из спарсенных данных убираем текст, добавленный в корпус размеченных текстов и проводим прогноз по оставшейся выборке. Проделайте этот процесс несколько раз.
7. **Разработка API.** Разработайте программный интерфейс для итоговой модели машинного обучения. API должен позволить приложению пользователя получать доступ к модели для определения категории обращений.
8. **Разработка приложения.** Разработайте приложение с графическим интерфейсом, которое должно с помощью разработанного API показывать спрогнозированную категорию обращений. Добавить вкладку со статистическими данными датасета (общее количество обращений, средняя длина постов, доля каждой категории обращений и т.д.) и визуализацией необходимых данных. Приложение также должно предоставлять справку по имеющимся командам и их параметрам.
9. **Программная документация.** Для разработанного приложения и API составьте программную документацию и руководство пользователя
10. **Оформить отчет.** Отчет должен охватывать результаты всей сессии, быть содержательный и выполнен профессионально.

Дан набор данных, содержащий текст на разных языках. Необходимо разработать модель для определения языка.

1. **Определить необходимые поля и создать структуру набора данных.** Определить, какие атрибуты имеют наибольшее влияние на распознавание языка и оставить только их для последующего обучения. Также необходимо обосновать выбор дополнительных атрибутов и причину исключения каких-либо данных из исходного набора документов.
2. **Провести предварительную обработку текстов размеченного датасета**: очистка данных, удаление стоп-слов, специальных символов, лишних пробелов; перевод в нижний регистр; лемматизация/стемминг.
3. **Провести тематическое моделирование** для текста на каждом языке: определите слова, специфичные для каждого класса: облако слов, модели тематического моделирования.
4. **Векторизация**. Получить векторное представление текстов.
5. **Классификация**. Проведите обучение классификатора на полученных векторах. Рассмотреть не менее 3 моделей классификации. Проведите оценку качества моделей.
6. **Парсинг данных.** Для улучшения классификатора требуется обогатить набор данных. Для этого необходимо провести парсинг текстовых данных с сайта: собрать не менее 500 текстов на разных языках. Далее проведите классификацию полученных текстов обращений с помощью обученной модели. Проанализируйте корректность работы модели классификации, выведите вероятности прогноза. Необходимо забрать из выборки только тот текст, в которых классификатор имеет оценку больше 0.8 или меньше 0.2 и добавить их в корпус размеченных текстов. Далее по новой выборке снова следует провести векторизацию, обучить модель. Из спарсенных данных убираем текст, добавленный в корпус размеченных текстов и проводим прогноз по оставшейся выборке. Проделайте этот процесс несколько раз.
7. **Разработка API.** Разработайте программный интерфейс для итоговой модели машинного обучения. API должен позволить приложению пользователя получать доступ к модели для определения языка текста.
8. **Разработка приложения.** Разработайте приложение с графическим интерфейсом, которое должно с помощью разработанного API показывать спрогнозированный язык текста. Добавить вкладку со статистическими данными датасета (общее количество текста, средняя длина текстов, доля каждого языка текстов и т.д.) и визуализацией необходимых данных. Приложение также должно предоставлять справку по имеющимся командам и их параметрам.
9. **Программная документация.** Для разработанного приложения и API составьте программную документацию и руководство пользователя
10. **Оформить отчет.** Отчет должен охватывать результаты всей сессии, быть содержательный и выполнен профессионально.