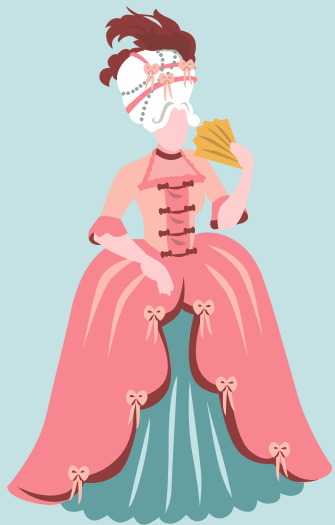


ТГ-Бот “Д’Артаньян”

Камская Милена и Харская Стефания



О чём проект



На определённых этапах изучения языка количество выучиваемых слов становится всё больше и больше. В какой-то момент можно даже прийти к тому, что это делать всё-таки лень. Наш бот предлагает не учить непосредственно перевод каких-то французских слов, а просто отмечать какую характеристику несёт это слово: положительную или отрицательную. Также подобный подход может пригодиться в условиях подготовки к экзамену, когда надо понимать не столько сам перевод слова, сколько эмоциональную окраску (за/против, да/нет, положительно/отрицательно).

Что умеет наш бот

01

Определять тональность
сообщения

02

Выводить слова, которые
повлияли на оценку

03

Игра – запоминание тональности слова. Пользователю предлагается слово, он должен выбрать его оценку (положительное / отрицательное)



Ход работы

Для создания нашего проекта мы воспользовались датасетом:

The Allociné dataset



С помощью этих данных мы обучили модель и сохранили её. Далее мы прописали функции бота, его реплики и реализовали “игру” по запоминанию слов. Итоговый проект выложен на гитхаб.



Технические детали и результаты

Функция `'preprocess_text()'` принимает текстовую строку в качестве входных данных и выполняет несколько этапов предварительной обработки текста. Вот краткое описание шагов:

1. Преобразование текста в нижний регистр
2. Удаление текста, заключенный в квадратные скобки: удаление метаданных или аннотаций, которые не имеют отношения к анализу текста.
3. Удаление знаков препинания для удаления всех знаков препинания из текста, за исключением апострофа (').
4. Удаление цифр

Функция возвращает предварительно обработанный текст в виде строки.





Технические детали и результаты

Logistic Regression:

Логистическая регрессия является простым и эффективным алгоритмом для задач классификации, включая задачу определения тональности.

Хорошо работает с линейно разделимыми данными.

Дает вероятностную интерпретацию результатов.

Multinomial Naive Bayes:

Мультиномиальный наивный Байесовский классификатор хорошо работает с текстовыми данными, что делает его хорошим выбором для анализа тональности текста.

Эффективен при работе с большими корпусами текста.

Хорошо справляется с множеством признаков.

Random Forest Classifier:

Случайный лес является мощным алгоритмом машинного обучения, который хорошо подходит для задач классификации, включая определение тональности.

Способен обрабатывать большое количество признаков и автоматически находить наиболее важные.

Устойчив к переобучению и хорошо работает на больших объемах данных.






Технические детали и результаты

`GridSearchCV` нами используется для настройки гиперпараметров модели логистической регрессии. Словарь `'param_grid'` определяет сетку гиперпараметров, по которым будет производиться поиск. В этом случае сетка включает в себя четыре гиперпараметра: `'C'`, `'solver'`, `'class_weight'` и `'max_iter'`. `'C'` - это величина, обратная коэффициенту регуляризации, `"solver"` - это алгоритм, используемый для оптимизации целевой функции логистической регрессии, `"class_weight"` используется для балансировки классов в данных, а `"max_iter"` - это максимальное количество итераций, за которые модель должна сходиться.

Затем создается объект `'GridSearchCV'` с использованием модели логистической регрессии, сетки параметров и других параметров, таких как количество повторных проверок (`'cv=3'`) и показатель оценки (`'scoring='roc_auc'`). Затем вызывается метод `'fit'` для подгонки модели к данным и выполнения поиска по сетке.






Технические детали и результаты

Наша модель: *LogisticRegression* с параметрами $C=10$, $class_weight='balanced'$, $solver='saga'$ и $max_iter=1500$

Окончательная версия модели была обучена на всех данных (*train*, *test*, *validation*), включая суммарно 200 000 записей для улучшения разнообразия данных и улучшения общей производительности.



Технические детали и результаты

Итоговая модель была сохранена с помощью модуля `Pickle`, который предоставляет возможность сериализовать и десериализовать объекты Python. Также в этом же формате мы сохранили векторизированные слова, чтобы бот не тратил на это время. В формате `.npy` были сохранены топ положительных и отрицательных слов по коэффициентам в модели. Отметим также, что коэффициенты значимых для тональности слов начинаются от 12 и от -12. Именно по этим числам мы и делили список для дальнейшей работы бота.



Что прячется за кодом бота



Наш бот работает следующим образом:

- Распаковываются все файлы (модель, векторизированные слова, топ слов по модели)
- По списку топ слов составляется словарь, который будет использоваться для “игры”
(от 12 — положительное, от -12 и вниз — отрицательное)
- Предобработка полученного сообщения для анализа происходит минимальная: к нижнему регистру, удаление чисел и знаков препинания (за исключением апострофа, который важен во французском языке)
- Оценка сообщения происходит с помощью `.predict_proba` (≥ 0.75 положительная, ≥ 0.5 скорее положительная, ≥ 0.25 скорее отрицательная, в других случаях — отрицательная)
- Выводится до трех слов, которые влияют на оценку сообщения, так как текст может быть очень уж маленьким

Плюсы и минусы проекта



Плюсы:

- Бот работает и может радовать учеников
- У нас высокая точность модели, которая оценивает текст
- Мы попробовали разные способы настройки параметров для обучения модели

Минусы:

- Пока это лексика уровня A2 - B1
- Боту необходимо время при запуске, чтобы распаковать файлы и подготовиться к общению с пользователем

Что надо улучшить

- Увеличить сложность лексики и набор слов
- Сделать игру по уровням языка
- Добавить новые функции в игру: перевод некоторых слов (если прямо хочется пользователю его узнать), ответы на время





Merci pour votre attention!



CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, infographics & images by **Freepik** and content by **Eliana Delacour**

Please keep this slide for attribution

