



**Кейс НТИ. «Развитие сервиса персональных траекторий»**

## **Команда MillingTeam**



Квашнин Денис  
DBA, DataScience



Ковалев Илья  
PM, Data Mining



Виктор Чекрыжов  
UI/UX, BackEnd

## Оглавление

1. Точность прогнозирования различных параметров.....	3
2. Описание используемых фреймворков и библиотек.....	5
3. Уровень реализации (макет, интерфейс, функциональность) .....	7
4. Уровень реализации (сложность используемых алгоритмов) .....	9
5. Наличие описания моделей прогнозирования .....	10
6. Вычислительная мощность, требуемая для работы модели.....	11
7. Сопроводительные материалы.....	12
8. Возможность развертывания модели в качестве сервиса на нашем сервере	12
9. Базовая документация по работе с ПО, реализующим модель.....	12
10.Полезность для пользователя .....	13
11.Наличие описанного user-сценария.....	14

## 1. Точность прогнозирования различных параметров

*\*Обученные модельки залиты в git*

Критерий достижимости цели (обученная моделька clfA.pkl)

	precision	recall	f1-score	support
0	0.89	0.68	0.77	4112
1	0.73	0.91	0.81	3943
accuracy			0.79	8055
macro avg	0.81	0.79	0.79	8055
weighted avg	0.81	0.79	0.79	8055

[[2792 1320]
[ 359 3584]]

Критерий конкретности цели (обученная моделька clfS.pkl)

	precision	recall	f1-score	support
0	0.76	0.88	0.81	4020
1	0.85	0.72	0.78	3958
accuracy			0.80	7978
macro avg	0.81	0.80	0.80	7978
weighted avg	0.81	0.80	0.80	7978

[[3520 500]
[1102 2856]]

Критерий ограниченности цели по времени (обученная моделька clfT.pkl)

	precision	recall	f1-score	support
0	0.83	0.88	0.86	5486
1	0.88	0.83	0.85	5643
accuracy			0.86	11129
macro avg	0.86	0.86	0.86	11129
weighted avg	0.86	0.86	0.86	11129

```
[[4836 650]
 [ 957 4686]]
```

Критерий образовательного направления цели (обученная моделька clfEducation.pkl)

	precision	recall	f1-score	support
0	0.86	0.72	0.78	3570
1	0.79	0.90	0.84	4214
accuracy			0.82	7784
macro avg	0.83	0.81	0.81	7784
weighted avg	0.82	0.82	0.81	7784

```
[[2560 1010]
 [ 414 3800]]
---
```

Критерий однозначности цели (обученная моделька clfUnambiguity)

```
12079 12079
```

	precision	recall	f1-score	support
0	0.93	0.75	0.83	6688
1	0.75	0.93	0.83	5391
accuracy			0.83	12079
macro avg	0.84	0.84	0.83	12079
weighted avg	0.85	0.83	0.83	12079

## 2. Описание используемых фреймворков и библиотек

Мы использовали следующие фреймворки и библиотеки:

### - **scikit-learn**

Самый распространенный выбор для решения задач классического машинного обучения. Из коробки предоставляет огромный выбор алгоритмов обучения с учителем и без учителя. Одно из основных преимуществ библиотеки состоит в том, что она работает на основе нескольких распространенных математических библиотек, и легко интегрирует их друг с другом. Также, имеет реализацию множества механизмов для скоринга полученного классификатора. Использовали для создания и обучения моделей.

### - **py morphology2**

Данный модуль представляет собой морфологический анализатор, позволяет выполнять токенизацию, леммирование, получать грамматическую информацию о слове и много чего крутого.

В нашем проекте мы используем его для получения лемм слов, на этапе фильтра шумов.

### - **nlTK**

Целый пакет разных модулей для символьной и статистической обработки естественного языка. Имеет возможность графического представления, а также готовые датасеты и фильтры (stopwords, к примеру). Обладает обширной документацией.

Применяли как фильтр для стоп-слов (с нашим расширением), в паре с \*py morphology2\*.

### - **natasha**

Специальный NLP фреймворк для работы с русским языком. Имеет огромное количество возможностей вышеназванных пакетов, а что-то умеет

делать даже лучше. Также имеет экстрактор фактов. Довольно интересный фреймворк. Мы используем как токенизатор и леммизатор, ещё один.

### **-numPy**

Все используемые библиотеки поддерживают numPy и структуры из него (например, ndarray), поэтому он и использовался.

### 3. Уровень реализации (макет, интерфейс, функциональность)

Сервер для backend был развернут на python и flask. На сервере происходит обучение моделей для критериев SMART по имеющемуся датасету (токенизация, lowercase, приводим в нормальную форму, убираем шум, пайп из TfidfVectorizer и SGDClassifier, производим обучение для каждого из критерия, получаем. pickle для каждого критерия)

Также для удобства взаимодействия с системой была разработана frontend часть, которая позволяет вводить цель и получать соответствие критериям SMART. Клиентская отправляет запрос серверу и получает обратно JSON с указанием соответствия по каждому из критериев)

В случае задания неконкретной цели (если ввели всего пару слов, вопросительная форма) будет выведено сообщение, что нужно ввести цель заново.

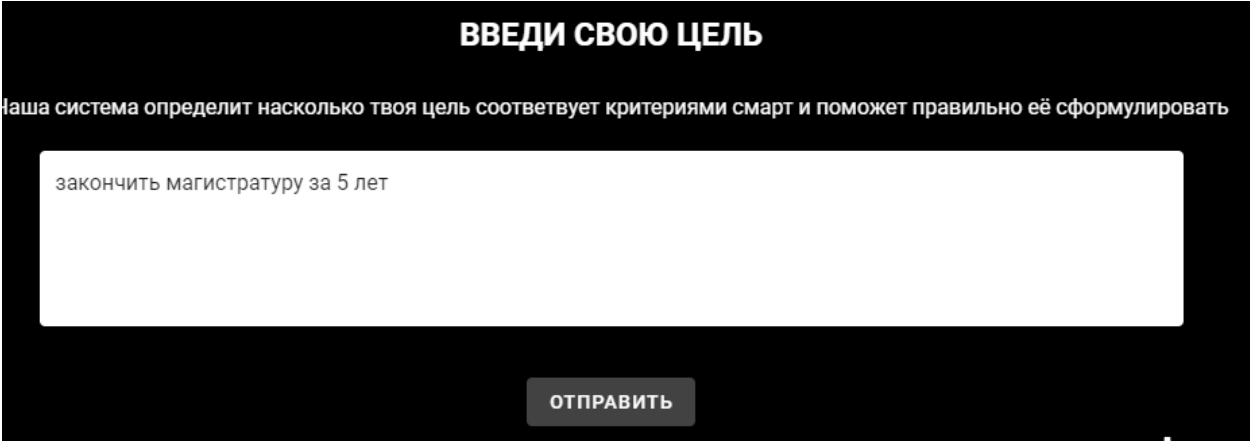


Рис. 1 Ввод цели

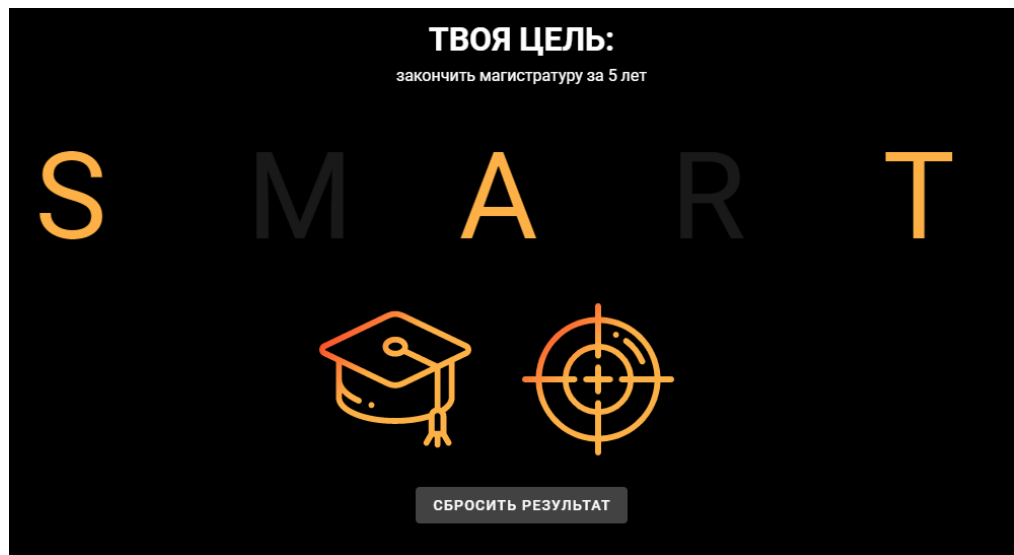


Рис. 2 Получение результата

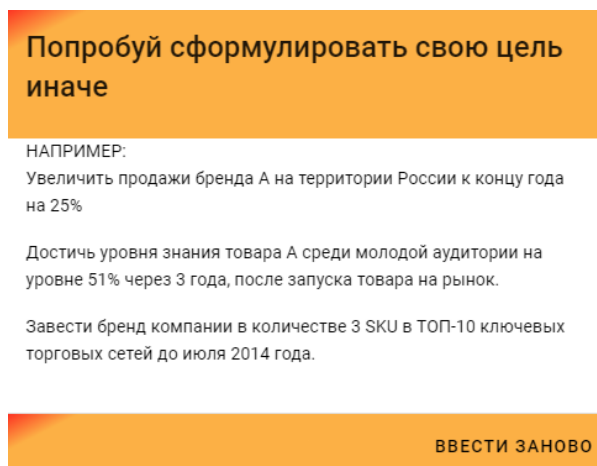


Рис. 3 Вывод сообщения при некорректно введённой цели



#### 4. Уровень реализации (сложность используемых алгоритмов)

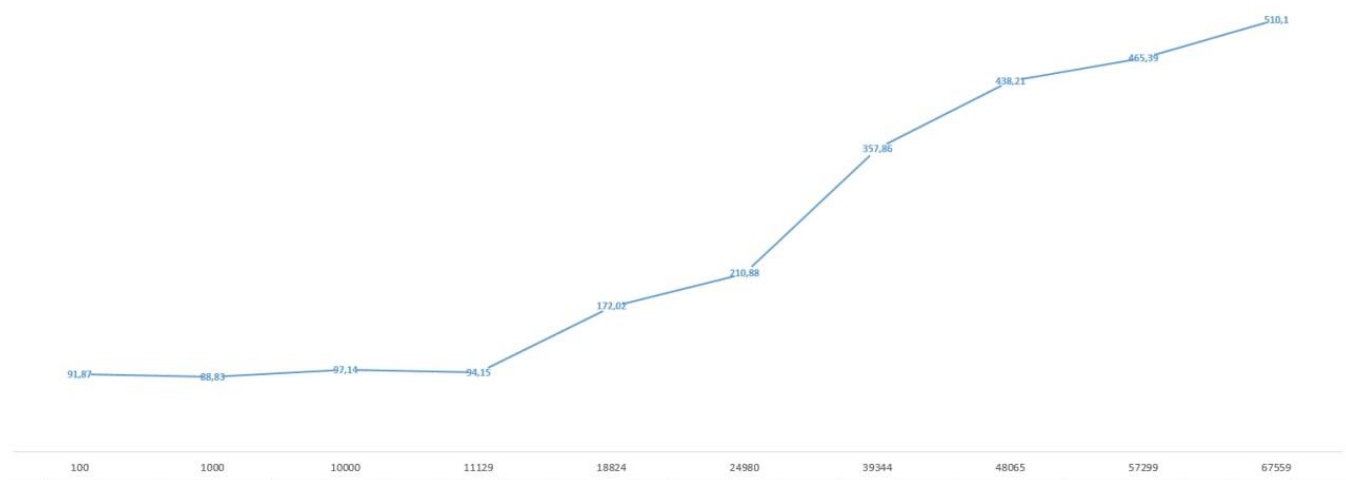


Рис. 4 Время на процесс в зависимости от количества данных

На этом графике отображается время (ось Y), затрачиваемое на подготовку разной величины входного массива данных (ось X), обучение модели и проверку метрик модели

## 5. Наличие описания моделей прогнозирования

\*Все обученные модельки (см. п.1), находятся в репозитории git

**CountVectorizer** в sklearn позволяет сконвертировать набор текстов в матрицу токенов, находящихся в тексте. Также имеется много полезных настроек, например, можно задать минимальное количество необходимое для появления токена в матрице и даже получить статистику по n-граммам. Следует учитывать, что **CountVectorizer** по умолчанию сам производит токенизацию и выкидывает слова с длиной меньшей чем два.

Далее это всё кидаем в **TfidfTransformer** трансформер и из него гоним в **SGDClassifier**.

Наша модель является экспериментальной, хотя и похоже на классический вариант с пайпом векторизатор->классификатор. Наибольший упор мы сделали на подготовку данных, а также пост-обработку результата самой модели. Также, разработанная система предоставляет интерфейс для применения вообще любой модели по принципу *hot reload*.

Методика системы такая, что информация проходит длинный путь обработки от банального леммирования и работы со словами, до обработки непосредственно самих целей как единицы, проверки возможности нести смысловую нагрузку (проверка на длину), тип предложения (вопрос или ещё что-то) даже банальная фильтрация по длине цели даёт приятный прирост в общем результате работы сервиса.

## 6. Вычислительная мощность, требуемая для работы модели

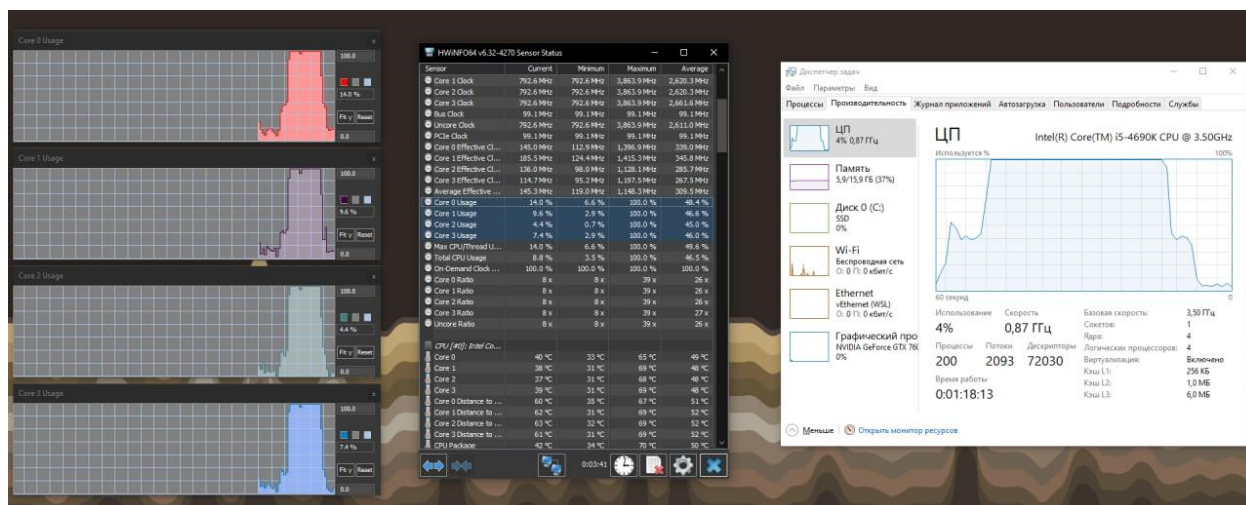


Рис. 5 Характеристики при обучении модельки

### Intel i5-8250U

Время полного обучения всех занимает около 5 мин, легкая моделька для облегчения работы с фронтом.

## 7. Сопроводительные материалы

Презентация находится в git репозитории.

Также на <https://smart.kovalev.team/> есть краткое описание этого сервиса

## 8. Возможность развертывания модели в качестве сервиса на нашем сервере

Система абсолютно модульная и может быть проведена интеграция с любой другой моделью обработки, не только с нашей

В readme frontend и backend есть описания, как установить и запустить эти части.

## 9. Базовая документация по работе с ПО, реализующим модель

В коде присутствуют комментарии, также к frontend и backend есть собственные readme, в которых описаны структуры проектов. Присутствуют файлы requirements, которые позволят установить необходимые зависимости.

## 10.Полезность для пользователя

Наша система определит насколько ваша цель соответствует критериями SMART и поможет правильно её сформулировать



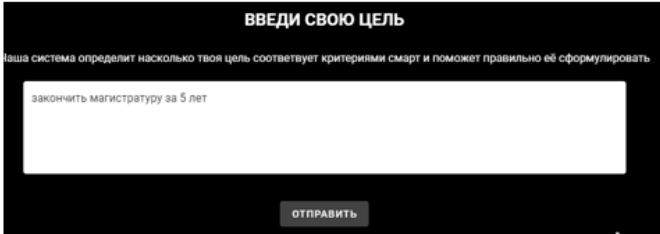
А также система определит соответствует ли цель образовательному направлению и однозначная ли она



## 11.Наличие описанного user-сценария

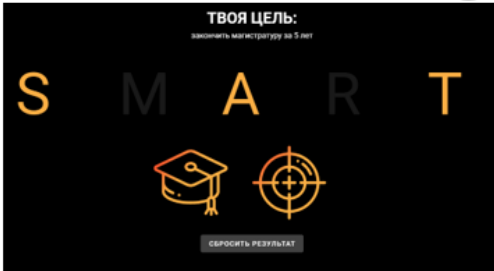
Как работать с нашим решением  
<https://smart.kovalev.team/>

1



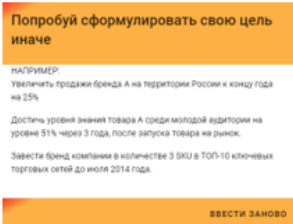
Ввести цель в специальное поле и нажать «Отправить»

2



Подсвечиваются критерии SMART соответствующие цели, а также достижима ли она и есть ли в ней образовательное направление

3



Если Вы ввели цель в вопросительной форме или, например, цель из пары слов, то будет предложено ввести цель повторно

Более подробно посмотреть, как работать можно по ссылке:

<https://drive.google.com/file/d/1274II2gK6pYmqBBzKtleuhKe9bv1IY55/view?usp=sharing>

