

Сиотанов А.М.

(a.siotanov@mail.ru)

ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ ПО ПРОГРАММНОМУ
ПРОДУКТУ
ОСНОВАННАЯ НА МАШИННОМ ОБУЧЕНИИ
ПО ПРОГНОЗИРОВАНИЮ НОМИНАЦИИ КОМПАНИИ
ОСНОВЫВАЯСЬ
НА СТАТЬИ ИХ ХАБР, ПОД УПРАВЛЕНИЕМ
ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Санкт-Петербург

2023

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ.....	3
2. РЕАЛИЗАЦИЯ ПРОДУКТА.....	5
3. РАЗРАБОТКА WEB ИНТЕРФЕЙСА.....	7
3.1 Разработка home.....	7
3.2 Разработка result.....	8

1. ВВЕДЕНИЕ

Данная техническая документация относится к разработке программного продукта на основе машинного обучения. Целью продукта является предсказание номинации компании в конкурсе «Рунета» проводимое на территории Российской Федерации.

В этом руководстве рассматривается базовая реализация приложения Flask, т.е. создание веб-сервера и простого REST API. Так же здесь описаны способы достижения поставленной цели, благодаря модели, обученной на статьях в расположенных в интернет источниках.

Продукт полностью написан на языке программирования «Python»,частичным добавлением языка HTML, с использованием следующих программных сред разработки:

- 1) Visual Studio Code
- 2) PyCharm

Программный код соответствует стандарту PEP 8.

Обеспечение управление продуктом посредством графического интерфейса используется библиотека Flask - это легкое веб-приложение, которое легко использовать и масштабировать до сложных приложений.

структура проекта

Для обеспечения правильной работы продукта следует быть внимательным к структуре данных программных файлов. Пример которого приведён на рисунке 1.

```
- API
  - static

  - templates
    - css
      - style.css
    - images
      - fon.jpg
      - home.html
      - result.html
  - app.py
  - ZjataiaModel.model
```

Рисунок 1- Структура файлов

2. РЕАЛИЗАЦИЯ ПРОДУКТА

Реализация продукта и его запуск расположены в файле *app.py*. В нем находится весь исходный код, который можно увидеть на рисунке 2.

```
1  from flask import Flask, render_template,request
2  import pandas as pd
3  from sklearn.feature_extraction.text import CountVectorizer
4  from sklearn.naive_bayes import MultinomialNB
5  from sklearn.externals import joblib
6  import fasttext
7  fasttext.FastText.eprint = lambda x: None
8
9  app = Flask(__name__)
10
11 @app.route('/')
12 def home():
13     return render_template('home.html')
14
15 @app.route('/predict', methods = ['POST'])
16 def predict():
17
18     clf = fasttext.load_model("OptimizedModel.model")
19
20     if request.method == 'POST':
21         comment = request.form['comment']
22         data = [comment]
23         vect = cv.transform(data).toarray()
24         ml_prognoz = clf.predict(vect)
25
26     return render_template('result.html', prognos = ml_prognos)
27
28
29 if __name__ == '__main__':
30     app.run()
31
```

Рисунок 2- исходный код app.py

Первым этапом идёт импортированные библиотек, который позволит выполнить поставленную задачу.

Второй этап это объявление об использовании библиотеки FLASK, что позволит создать web-среду для работы.

Третий этап заключается в функции перехода на домашнюю страницу графического интерфейса с элементами управления.

Четвёртый этап несёт в себе импортированные модели, после чего написанный пользователем код в определённой графе примет в себя модели и произведёт прогнозирование.

Пятый этап основан на выводе предсказания на другую web-среду с выводом на экран результата работы.

Шестой и заключительный этап заключается в запуске продукта.

3. РАЗРАБОТКА WEB ИНТЕРФЕЙСА

3.1 Разработка home

Чтобы управлять программным продуктом необходим графический интерфейс, с помощью которого модель может принимать данные и прогнозировать их.

Для этого создан файл home.html с исходным кодом которого можно ознакомиться на рисунке 3.

```
<!DOCTYPE html>
<html>
<head>
  <title>SIOTANOV ALEXEY ML WEB API</title>
  <style>
    body {
      background-image: url(images/fon.jpg); /* Путь к фоновому изображению */
      background-position: center;
      background-size: 100%;
      backdrop-filter: blur(1px);
      color: #ffff;
    }
  </style>
  <link rel="stylesheet" type="text/css" href="css/style.css">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>

  <div align="center" class="bg-info">
<h1>Модель для прогнозирования</h1></div>
<div class="big" align="center">
  <form action="{{url_for('predict')}}" method="POST">
    <h3>Введите URL ссылки компании сюда</h3>

    <textarea name="comment" rows="4" cols="50"></textarea>
    <br>
    <input type="submit" name="" value="Прогноз" class="btn btn-info">
  </form>
</div>
<br>
<div align="center">
  <h3><strong>Справка по программе:</strong></h3>
  <p>Вам необходимо вставить URL ссылку статьи любой компании с Хабр.</p>
  <p>После чего необходимо нажать на кнопку прогноз.</p>
  <p>Далее подождать пока модель сделает прогноз и выведет его вам.</p>
  <a><strong>Спасибо за выбор данного программного продукта!</strong></a>
</div>

</body>
<footer>
  <div align="center">
    <p>Данный программный продукт создан исключительно для соревнования регионального этапа "Профессионалы 2023" в СПб</p>
    <p>© 2023 Siotanov Alexey</p>
  </div>
</footer>
</html>
```

Рисунок 3- Исходный код home.html

Состоит страница из нескольких блоков:

- 1) Заголовок первого порядка;
- 2) Окно для предоставления информации от пользователя;
- 3) Кнопка для запуска программного продукта;\
- 4) Справка по программе.

На рисунке 4 вы можете наблюдать графический интерфейс, а также пример его использования.

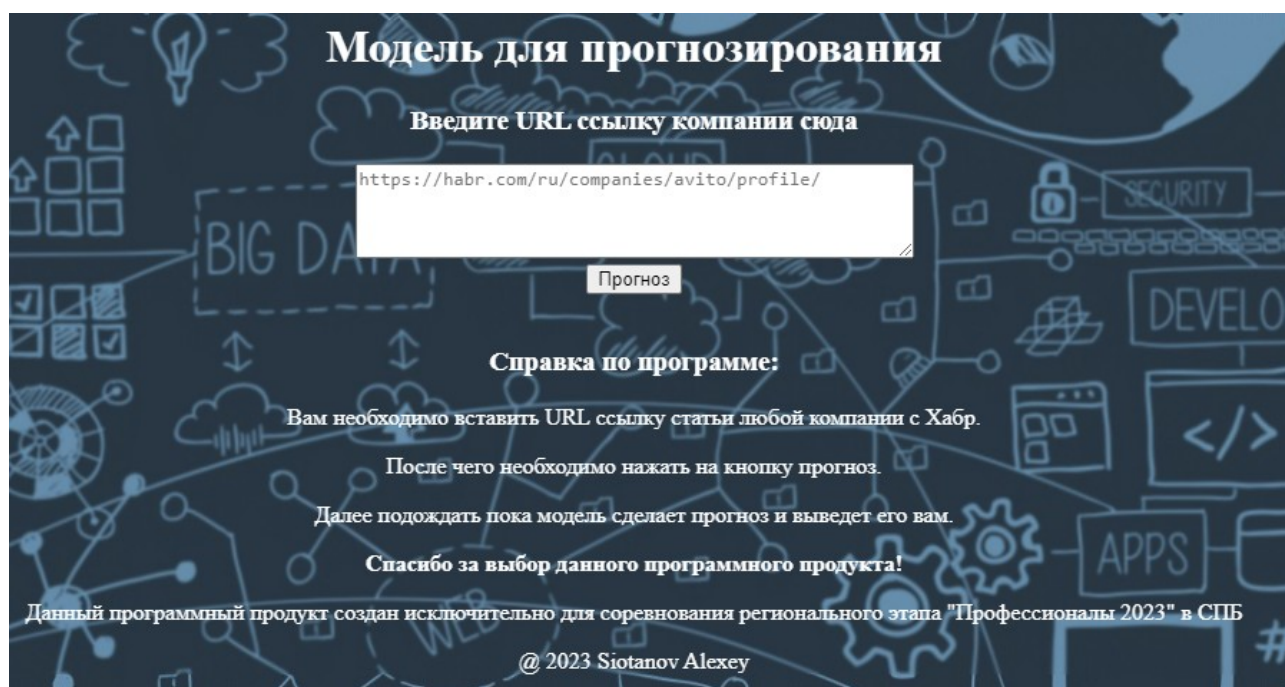


Рисунок 4- Графический интерфейс home.html

3.2 Разработка result

После выполнения программы, она выведет информацию на новую страницу, для этого реализована отдельная web-страница с графическим интерфейсом. Исходный код можно наблюдать на рисунке 5.


```

<!DOCTYPE html>
<html>
<head>
  <title>Result</title>
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename= 'css/style.css')}}">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
  <div class="bg-info" align="center">
    <h2>Результат</h2>
  </div>
  <div align="center">
    <h2 class="text-danger">Прогноз следующий:</h2>
    {% prognos %}
  </div>
</body>
</html>

```

Рисунок 5- Исходный код result.html

Структура web-страницы состоит из:

- 1) Заголовка второго порядка;
- 2) Вывода предсказания программы.

На рисунке 6 вы можете наблюдать графический интерфейс результативной web-старницы, а также пример его использования.

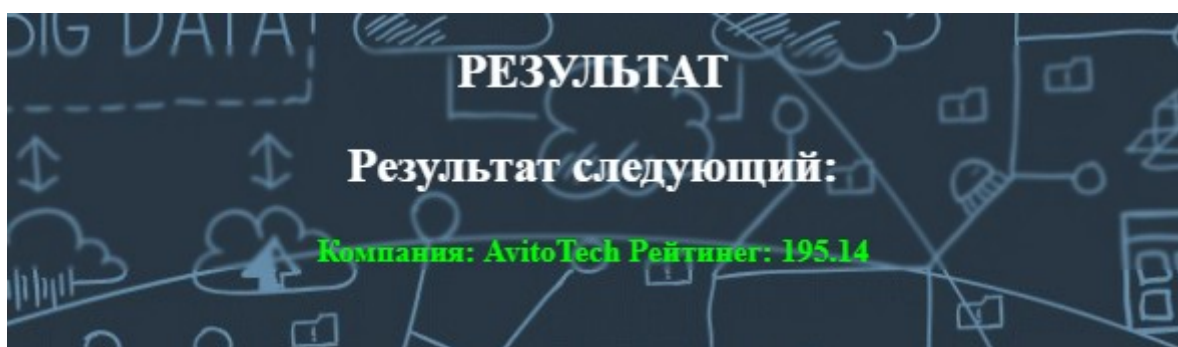


Рисунок 6- Прмиер result.html