# Предсказание авторства книг в жанре фэнтези

Иванов И. Кузьмин П. Шлюпенков С.

# В чем идея?

В данном исследовании представлена модель, которая классифицирует данные по авторам в жанре фэнтези.

Для нашего исследования мы взяли 5 из самых популярных произведений в жанре фэнтези: Дж. Р. Р. Толкин "Властелин колец", А. Сапковский "Последнее желание", Дж. Мартин "Игра престолов", Р. Желязны "Хроники Амбера" и Н. Гейман "Американские боги".

Мы создадим датасет, в котором у нас будут содержаться данные для решения задачи классификации предложений по авторству из исходных текстов.

# Цели и задачи

**Цель** работы заключается в создании модели-классификатора указания авторства в книгах жанра фэнтези.

**Задачи:**

1. Собрать необходимые данные
2. Провести препроцессинг данных и подготовить их к анализу
3. Создать модель-классификатор
4. Провести анализ полученных результатов

# Этапы работы

Исследование поделено на две части:

В первой части мы собираем и подготавливаем данные для последующей загрузки в модель.

Во второй части мы проводим необходимый анализ текста для выполнения задачи классификации. Вторая часть состоит из трех независимых друг от друга частей. В каждой используются свои решения задачи классификации.

```
out_data = pd.DataFrame()
out_data['text'] = combined
out_data['author'] = labels
print(out_data.head())
print(out_data.tail())
```

```
                                                   text   author
0  Тысячи лет пираты с Железных Островов – железн...   Martin
1  И на сей раз на него нахлынул приступ ярости, ...   Gaiman
2                                  И ответа не было.   Gaiman
3  Любопытство мое было подогрето до такой степен...  Tolkien
4       Пламя лизало камень злыми красными язычками.   Martin
                                                  text   author
49995  Потом я разглядел, что у нее в левой руке, и б...  Zelazny
49996                      Знал, наверное, еще как знал!  Tolkien
49997  Он вытер руки о джинсы и протянул Тени могучую...   Gaiman
49998            А потом сказала с неуверенной улыбкой:.   Gaiman
49999  Кто-то заметил на дереве краснохвостого ястреб...   Gaiman
```

+ Code + Markdown

Получившийся необработанный датасет

```
data['text'] = normed_text
data.to_csv('preprocessed_data.csv', index=False)
print(data.head())
print(data.tail())
```

```
                                                text   author
0  тысячи лет пираты с железных островов  железны...   Martin
1  и на сей раз на него нахлынул приступ ярости г...   Gaiman
2                                 и ответа не было   Gaiman
3  любопытство мое было подогрето до такой степен...  Tolkien
4        пламя лизало камень злыми красными язычками   Martin
                                                text   author
49995  потом я разглядел что у нее в левой руке и быс...  Zelazny
49996                          знал наверное еще как знал  Tolkien
49997  он вытер руки о джинсы и протянул тени могучую...   Gaiman
49998              а потом сказала с неуверенной улыбкой   Gaiman
49999  ктото заметил на дереве краснохвостого ястреба...   Gaiman
```

+ Code + Markdown

Получившийся обработанный датасет

# БЛОК 1
# Сбор и подготовка данных

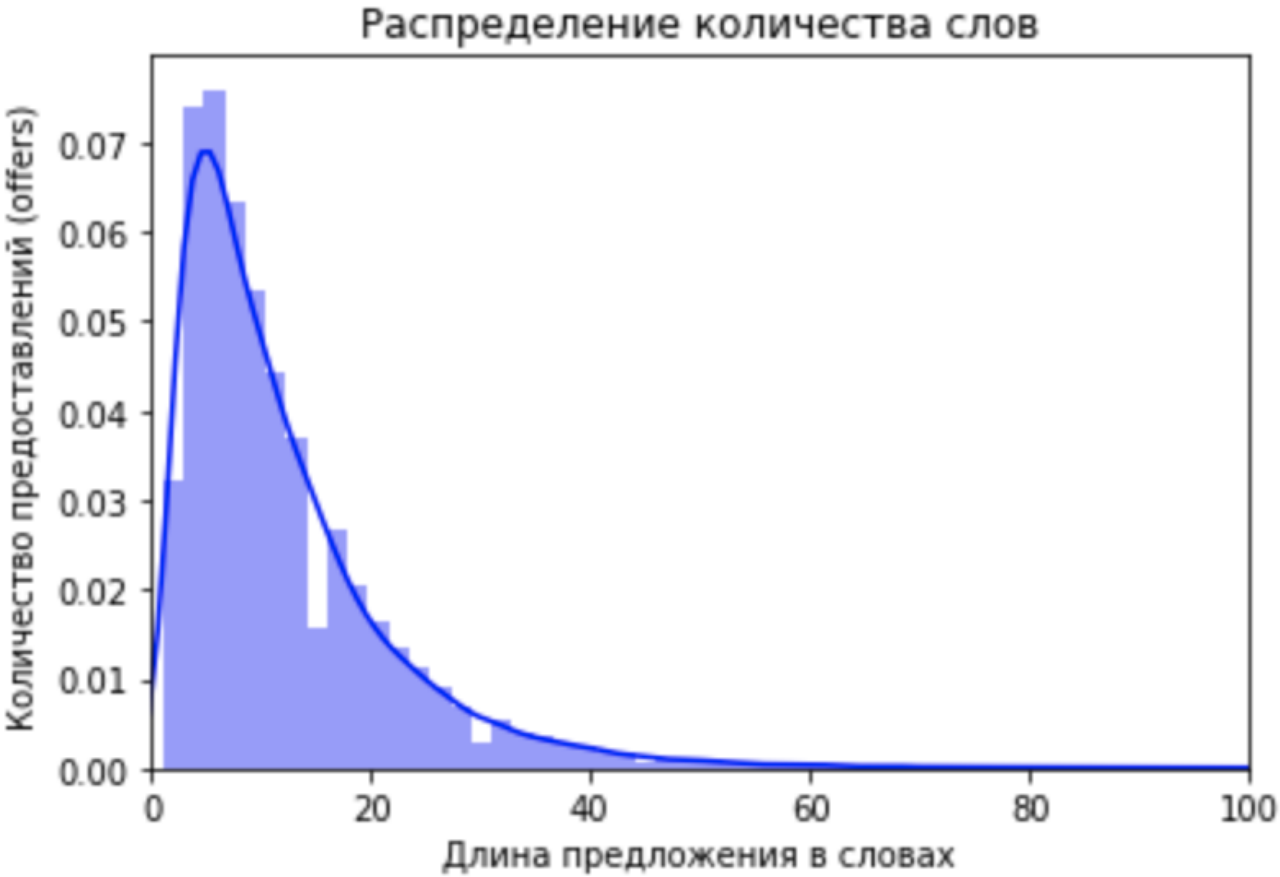
```
authors = Counter(author)
authors
```

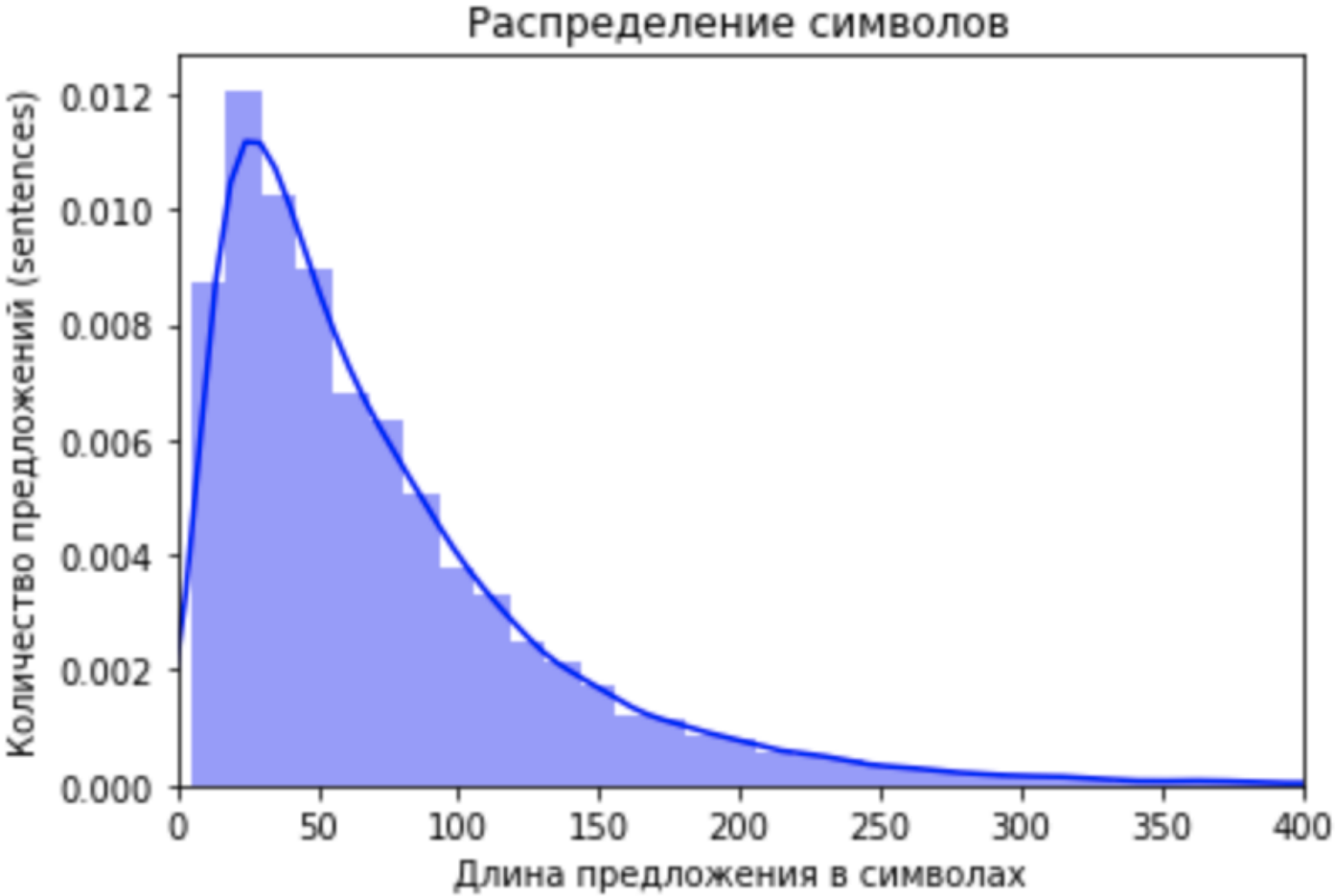
```
Counter({'Martin': 10000,
         'Gaiman': 10000,
         'Tolkien': 10000,
         'Zelazny': 10000,
         'Sapkowski': 10000})
```

В нашем датасете мы равномерно распределили количество предложений по авторам

```
word_count = np.array([len(sent.split()) for sent in text])
char_count = np.array([len(sent) for sent in text])
ave_length = char_count / word_count
```

Пока мы не обработали данные для работы с ними в способах 1 и 2 посчитаем базовую статистику


Распределение количества слов
Количество предоставлений (offers)
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0
20
40
60
80
100
Длина предложения в словах


Распределение символов
Количество предложений (sentences)
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
50
100
150
200
250
300
350
400
Длина предложения в символах

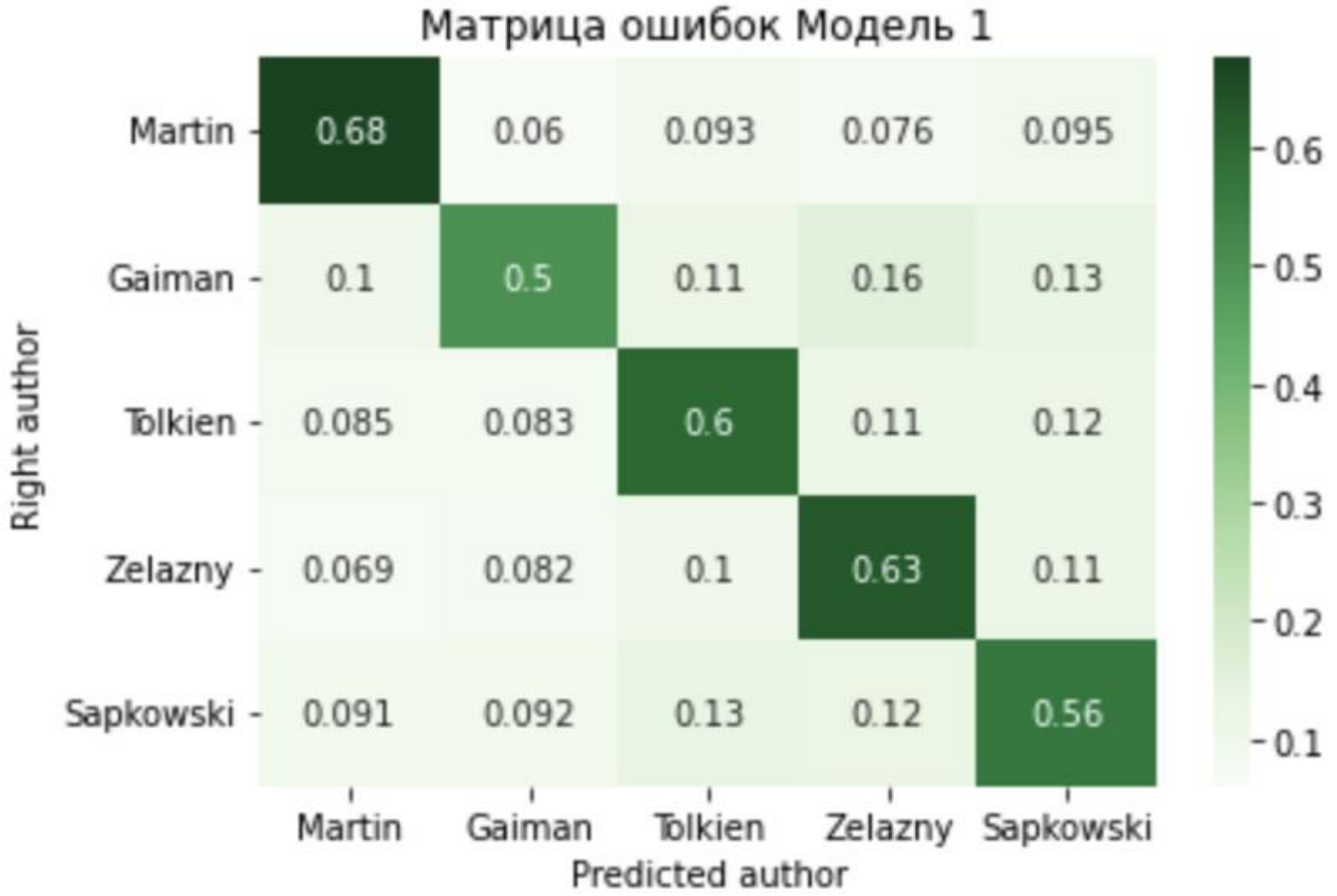

Распределение средней длины слова
Количество предложений (sentences)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0
2
4
6
8
10
Средняя длина слова в символах

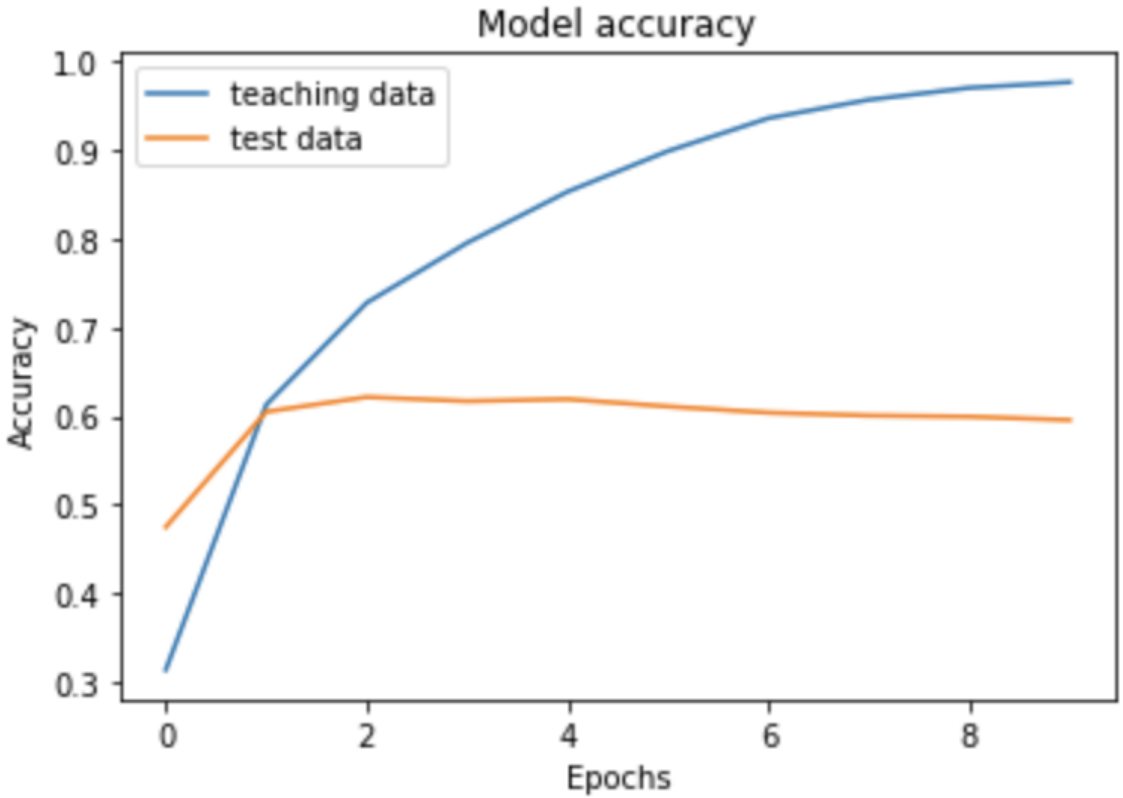
# БЛОК 2
# Анализ текста и выполнение задачи классификации

# Способ 1
# N-граммы: сложный и долгий метод

```
Accuracy: 0.5949
Average Precision: 0.5960417779145594
Average Recall: 0.5949
Average F1 Score: 0.5939566255094416
Learning time: 577.1681146621704 секунд
Prediction time: 272.6482763290405 секунд
Confusion matrix:
 [[1359  121  187  153  191]
 [ 200 1004  230  312  260]
 [ 168  164 1186  224  229]
 [ 138  163  208 1267  224]
 [ 184  186  259  250 1133]]
```

По итогу рассчета наших n-грамм (1-3) мы получаем первую модель и следующие метрики

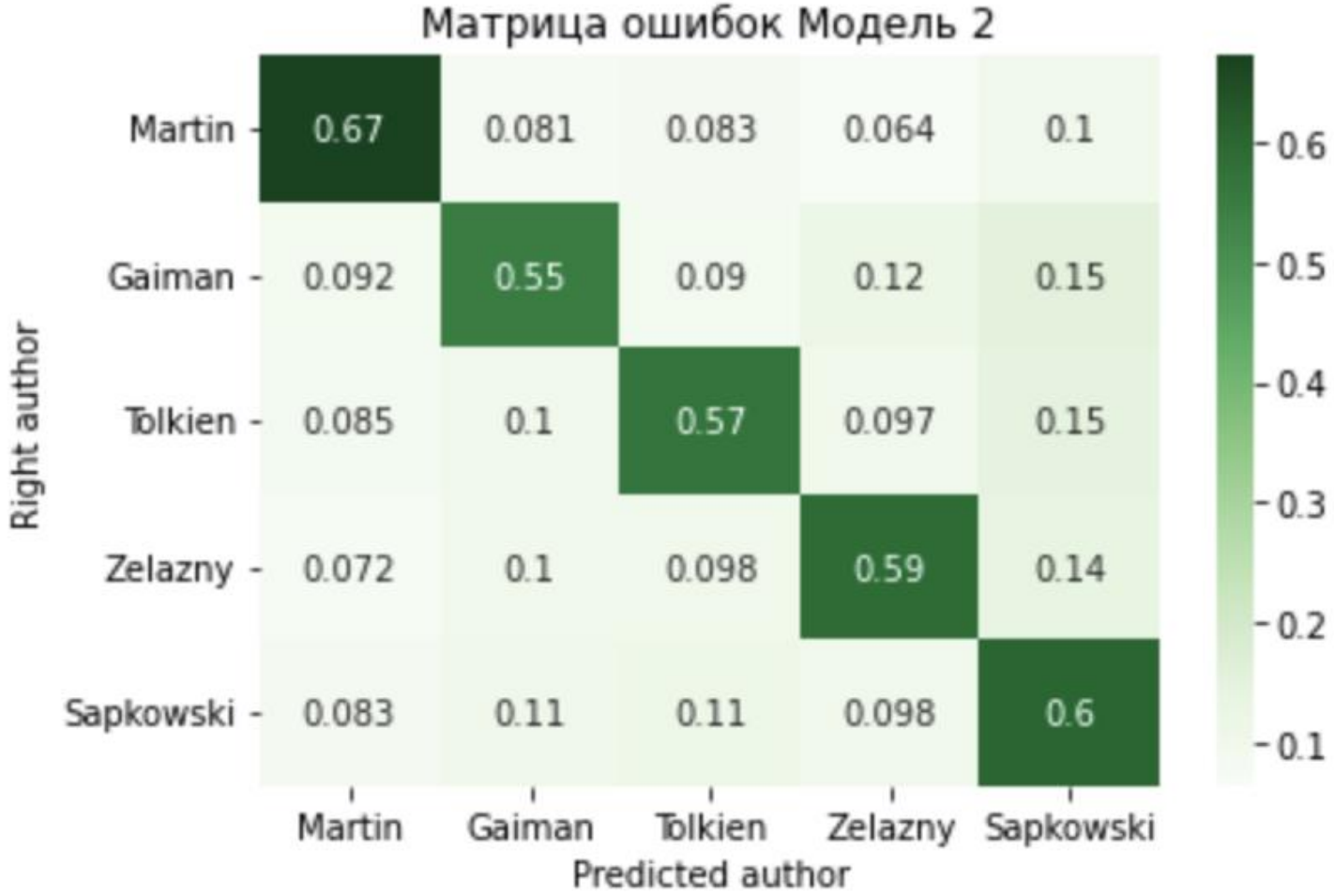

Матрица ошибок Модель 1
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Right author
0.68 0.06 0.093 0.076 0.095
0.1 0.5 0.11 0.16 0.13
0.085 0.083 0.6 0.11 0.12
0.069 0.082 0.1 0.63 0.11
0.091 0.092 0.13 0.12 0.56
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Predicted author
0.6
0.5
0.4
0.3
0.2
0.1

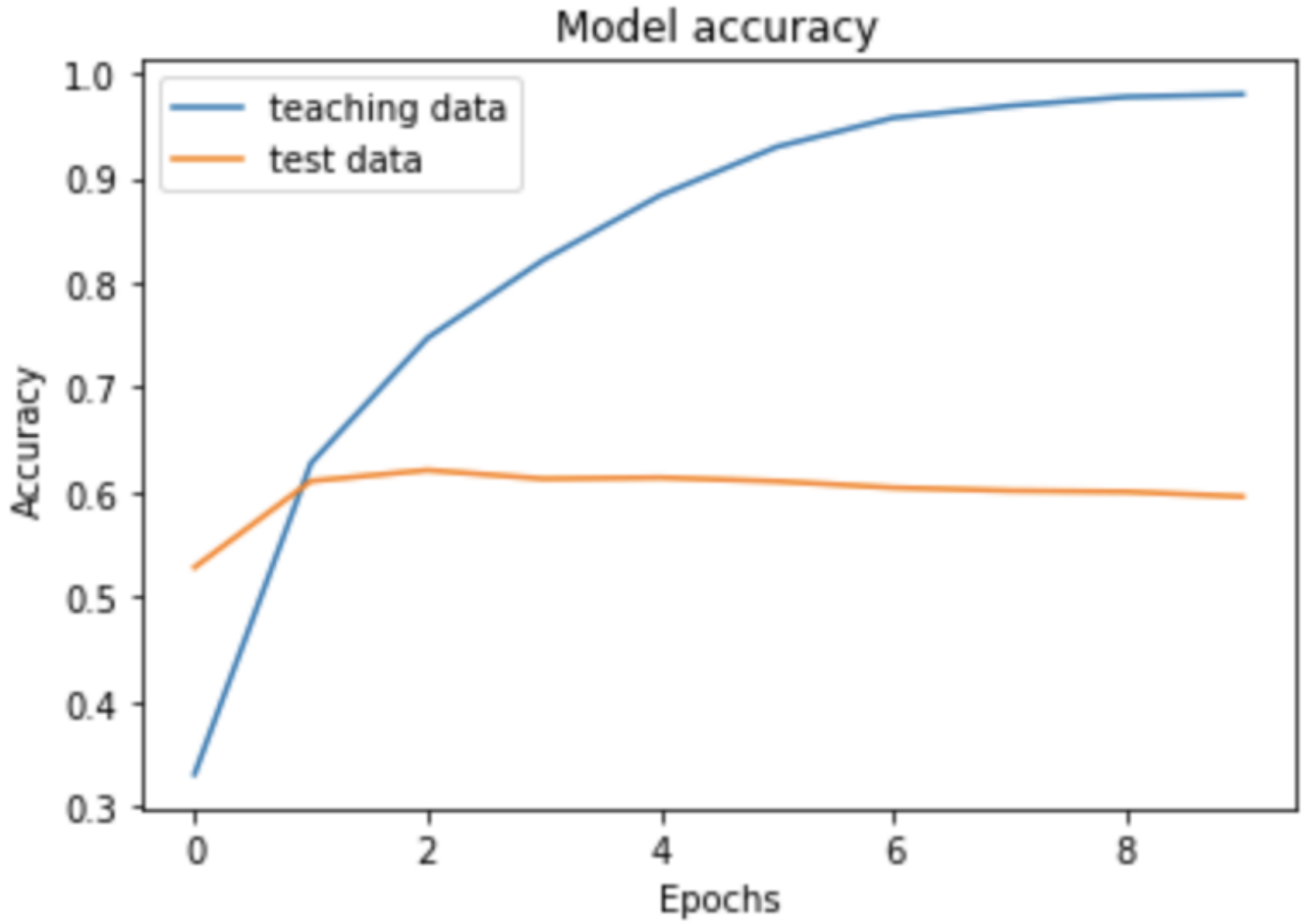

Model accuracy
teaching data
test data
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8


Model error
teaching data
test data
1.75
1.50
1.25
1.00
0.75
0.50
0.25
Error
0
2
4
6
8
Epochs

```
Accuracy: 0.5975
Average Precision: 0.5992085171792201
Average Recall: 0.5974999999999999
Average F1 Score: 0.5976984427392338
Learning time: 806.9759614467621 секунд
Predict time: 376.9495553970337 секунд
Confusion matrix:
 [[1350  162  167  129  203]
 [ 185 1103  180  233  305]
 [ 168  199 1126  192  286]
 [ 144  202  196 1183  275]
 [ 166  212  224  197 1213]]
```

После апгрейда модели мы получаем вторую модель и следующие метрики


Матрица ошибок Модель 2
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Right author
0.67 0.081 0.083 0.064 0.1
0.092 0.55 0.09 0.12 0.15
0.085 0.1 0.57 0.097 0.15
0.072 0.1 0.098 0.59 0.14
0.083 0.11 0.11 0.098 0.6
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Predicted author
0.6
0.5
0.4
0.3
0.2
0.1

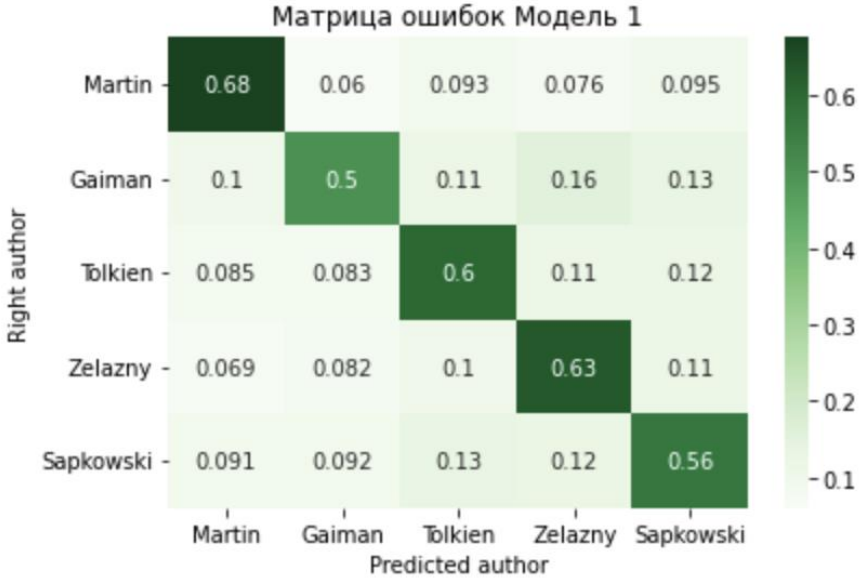

Model accuracy
teaching data
test data
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8

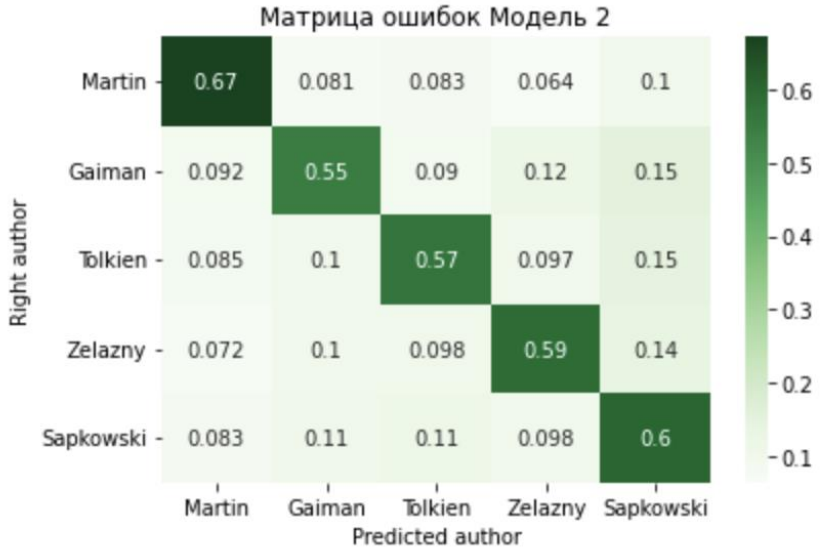

Model error
teaching data
test data
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Error
0
2
4
6
8
Epochs

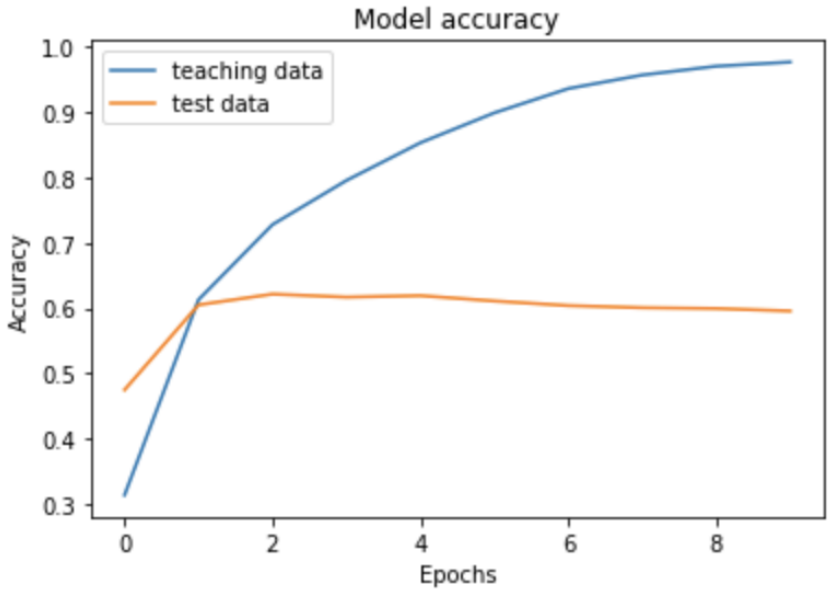
```
Accuracy: 0.5949
Average Precision: 0.5960417779145594
Average Recall: 0.5949
Average F1 Score: 0.5939566255094416
Learning time: 577.1681146621704 секунд
Prediction time: 272.6482763290405 секунд
Confusion matrix:
 [[1359  121  187  153  191]
 [ 200 1004  230  312  260]
 [ 168  164 1186  224  229]
 [ 138  163  208 1267  224]
 [ 184  186  259  250 1133]]
```

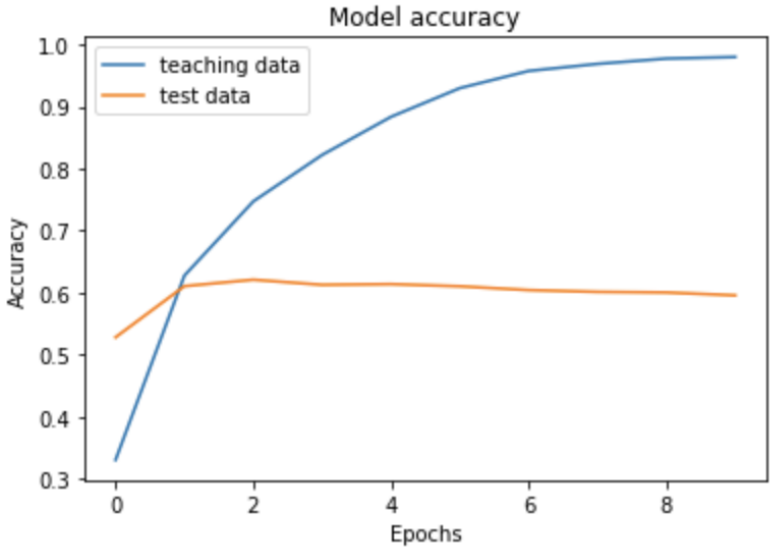
```
Accuracy: 0.5975
Average Precision: 0.5992085171792201
Average Recall: 0.5974999999999999
Average F1 Score: 0.5976984427392338
Learning time: 806.9759614467621 секунд
Predict time: 376.9495553970337 секунд
Confusion matrix:
 [[1350  162  167  129  203]
 [ 185 1103  180  233  305]
 [ 168  199 1126  192  286]
 [ 144  202  196 1183  275]
 [ 166  212  224  197 1213]]
```

Наглядно


Матрица ошибок Модель 1
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Right author
0.68 0.06 0.093 0.076 0.095
0.1 0.5 0.11 0.16 0.13
0.085 0.083 0.6 0.11 0.12
0.069 0.082 0.1 0.63 0.11
0.091 0.092 0.13 0.12 0.56
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Predicted author
0.6
0.5
0.4
0.3
0.2
0.1


Матрица ошибок Модель 2
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Right author
0.67 0.081 0.083 0.064 0.1
0.092 0.55 0.09 0.12 0.15
0.085 0.1 0.57 0.097 0.15
0.072 0.1 0.098 0.59 0.14
0.083 0.11 0.11 0.098 0.6
Martin
Gaiman
Tolkien
Zelazny
Sapkowski
Predicted author
0.6
0.5
0.4
0.3
0.2
0.1


Model accuracy
teaching data
test data
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8


Model accuracy
teaching data
test data
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8


Model error
teaching data
test data
Error
Epochs
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0
2
4
6
8


Model error
teaching data
test data
Error
Epochs
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
2
4
6
8

```python
for i in range(20):
    print('Предложение', i, '- Верный ответ =', author_test[i],  'Предсказание Модели 1 =', author_pred1[i], 'Предсказание Модели 2 =', author_pred2[i])
    print(text_test[i], '\n')
```

```
Предложение 0 - Верный ответ = Gaiman Предсказание Модели 1 = Sapkowski Предсказание Модели 2 = Sapkowski
геродота

Предложение 1 - Верный ответ = Sapkowski Предсказание Модели 1 = Sapkowski Предсказание Модели 2 = Gaiman
держи рот на замке и не встревай

Предложение 2 - Верный ответ = Martin Предсказание Модели 1 = Martin Предсказание Модели 2 = Martin
с ним был сир марк пайпер они привели сына сира реймена дарри парнишку не старше брана

Предложение 3 - Верный ответ = Gaiman Предсказание Модели 1 = Tolkien Предсказание Модели 2 = Sapkowski
я слышал чейто голос

Предложение 4 - Верный ответ = Sapkowski Предсказание Модели 1 = Gaiman Предсказание Модели 2 = Gaiman
и не крутили задом у мужчин перед глазами

Предложение 5 - Верный ответ = Sapkowski Предсказание Модели 1 = Sapkowski Предсказание Модели 2 = Sapkowski
 тебе не уехать геральт

Предложение 6 - Верный ответ = Tolkien Предсказание Модели 1 = Tolkien Предсказание Модели 2 = Tolkien
я сразу же покинул дэнетора но еще по дороге на север получил весть из лориэна арагорнде побывал там и просил передать что он отыскал существо именуемое голлумом

Предложение 7 - Верный ответ = Sapkowski Предсказание Модели 1 = Sapkowski Предсказание Модели 2 = Sapkowski
ни одного родственника или жениха дочки

Предложение 8 - Верный ответ = Zelazny Предсказание Модели 1 = Gaiman Предсказание Модели 2 = Gaiman
в каком смысле

Предложение 9 - Верный ответ = Zelazny Предсказание Модели 1 = Sapkowski Предсказание Модели 2 = Sapkowski
для этого мне пришлось вызвать логрус который упал между нами словно нож гильотины и дернул меня так словно я прикоснулся к оголенному проводу под напряжением

Предложение 10 - Верный ответ = Gaiman Предсказание Модели 1 = Martin Предсказание Модели 2 = Martin
я хочу с ней попрощаться
```

# Способ 2
# Стемминг и векторные представления: логистическая регрессия

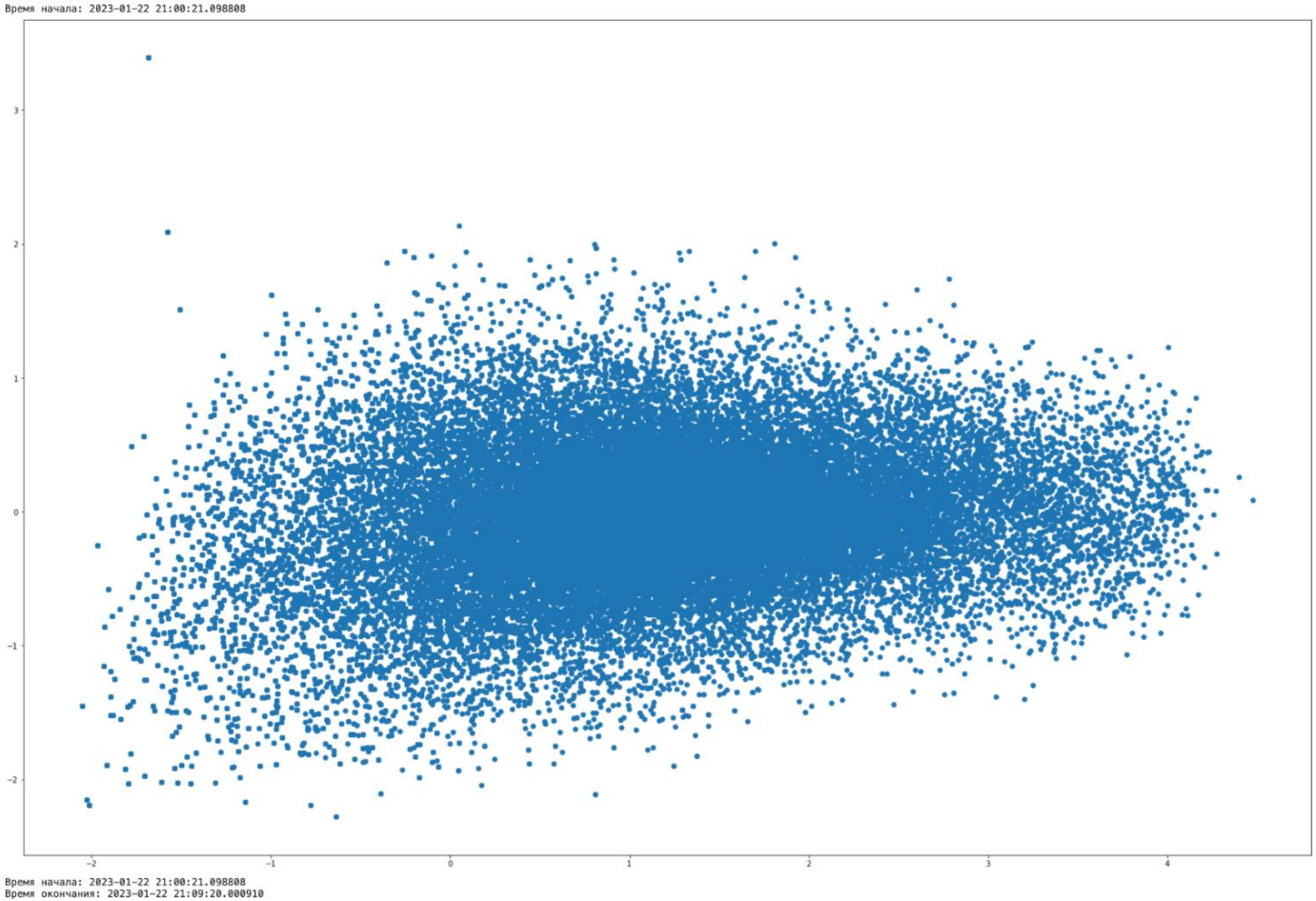
| | text | author |
|---|---|---|
| 0 | тысяч лет пират железн остров железн людьм наз... | Martin |
| 1 | се нахлынул приступ ярост глухо темны собра ку... | Gaiman |
| 2 | ответ | Gaiman |
| 3 | любопытств мо подогрет тако степен последова н... | Tolkien |
| 4 | плам лиза камен злым красн язычк | Martin |

| | text | author |
|---|---|---|
| 49995 | разглядел лево рук быстр посмотрел джул | Zelazny |
| 49996 | знал наверн знал | Tolkien |
| 49997 | вытер рук джинс протянул тен могуч лапищ слыша... | Gaiman |
| 49998 | сказа неуверенно улыбко | Gaiman |
| 49999 | замет дерев краснохв ястреб сказа эт скор соко... | Gaiman |

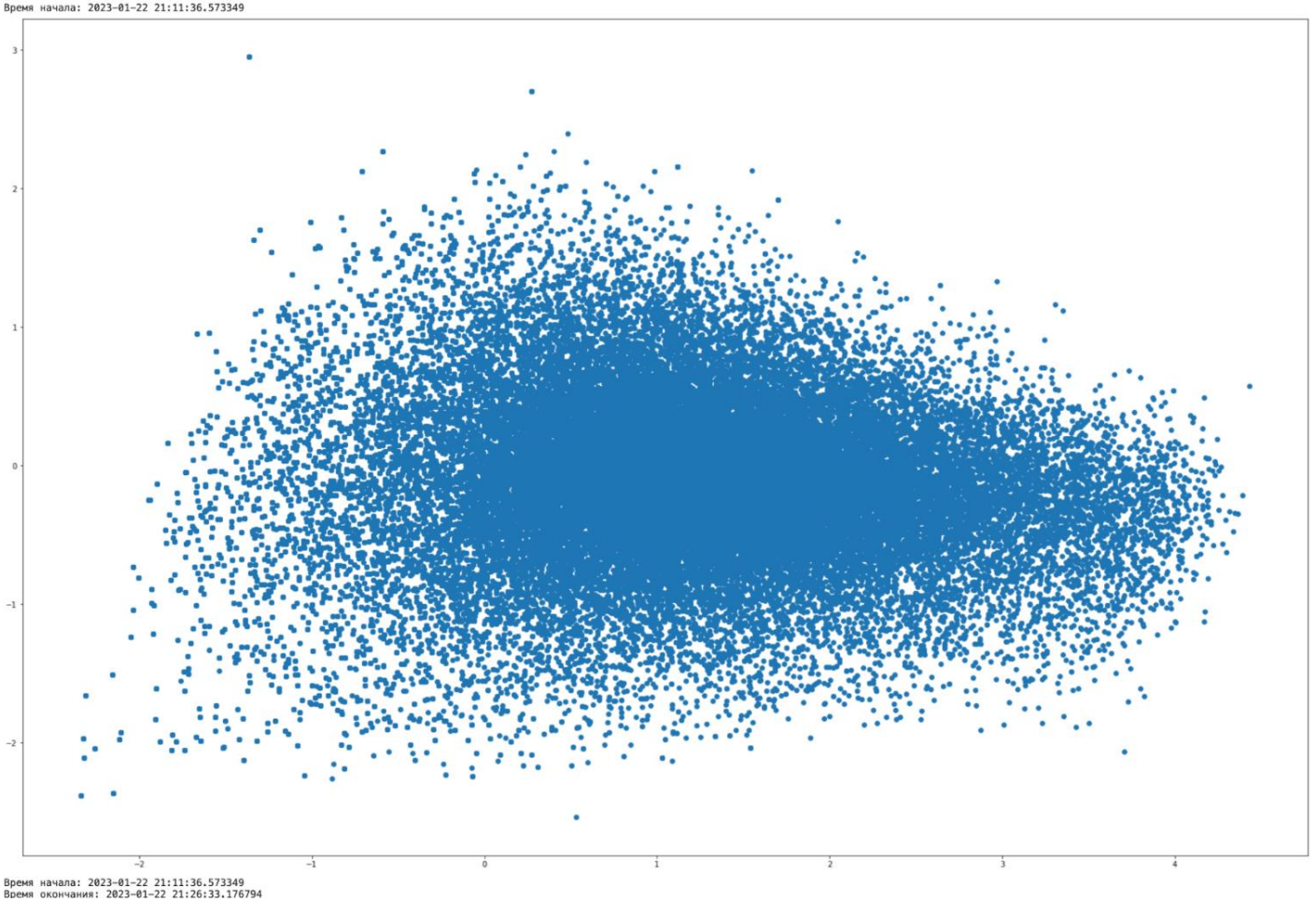
```
(42500, 246287) (7500, 246287)
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
0.6726666666666666
```

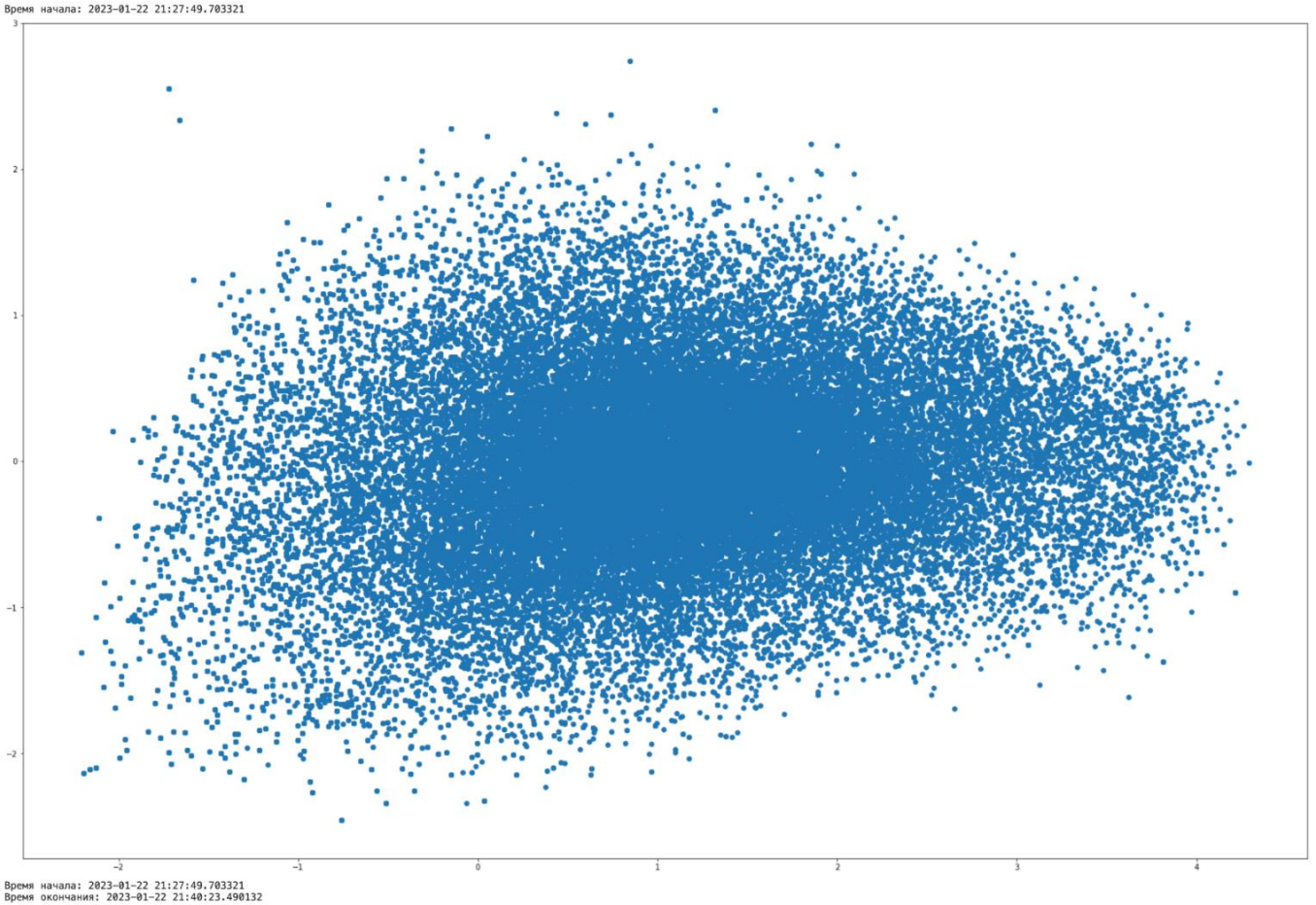
# Способ 3
# Визуализация эмбеддингов для каждой книги



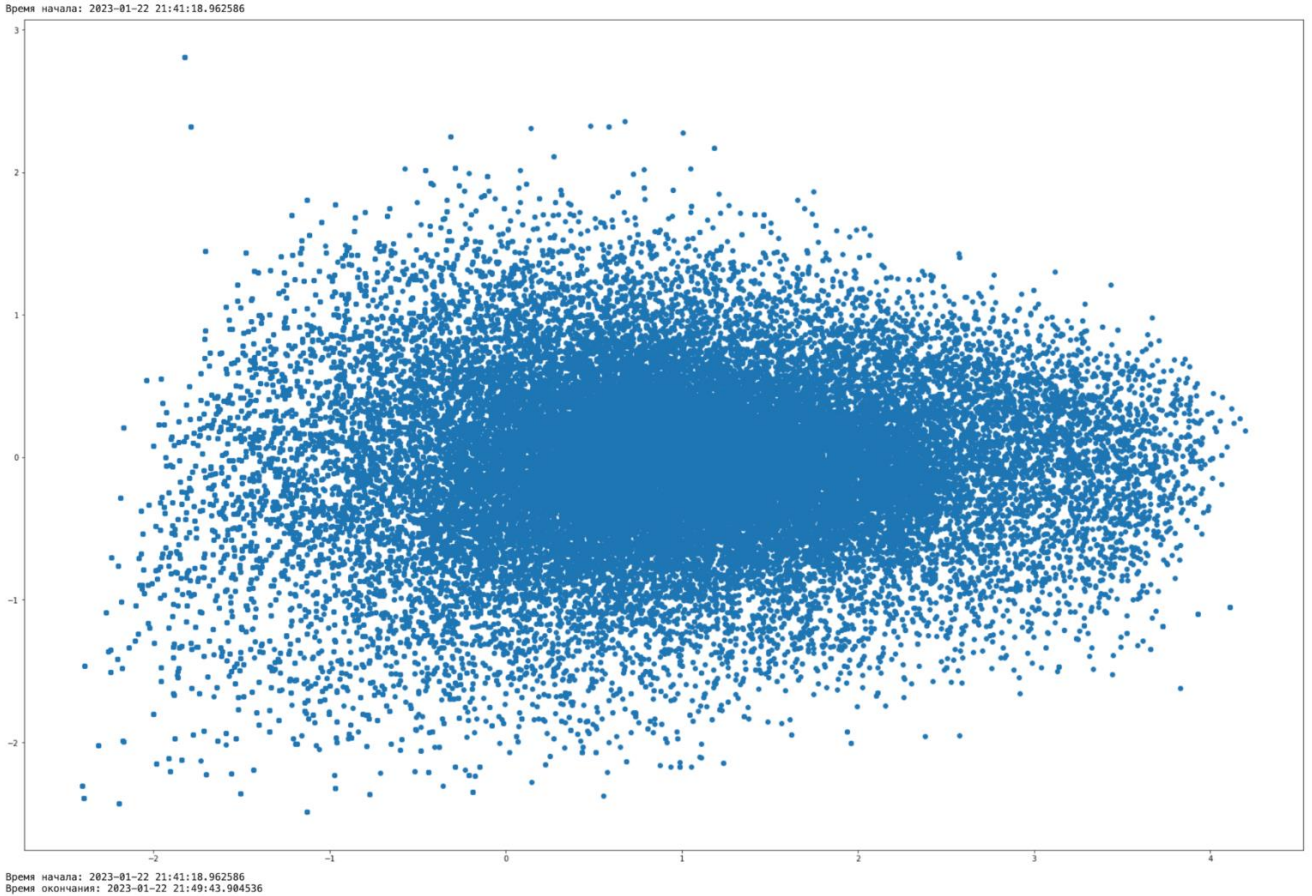

Считаем эмбеддинги для "Американские боги" Н. Геймана.



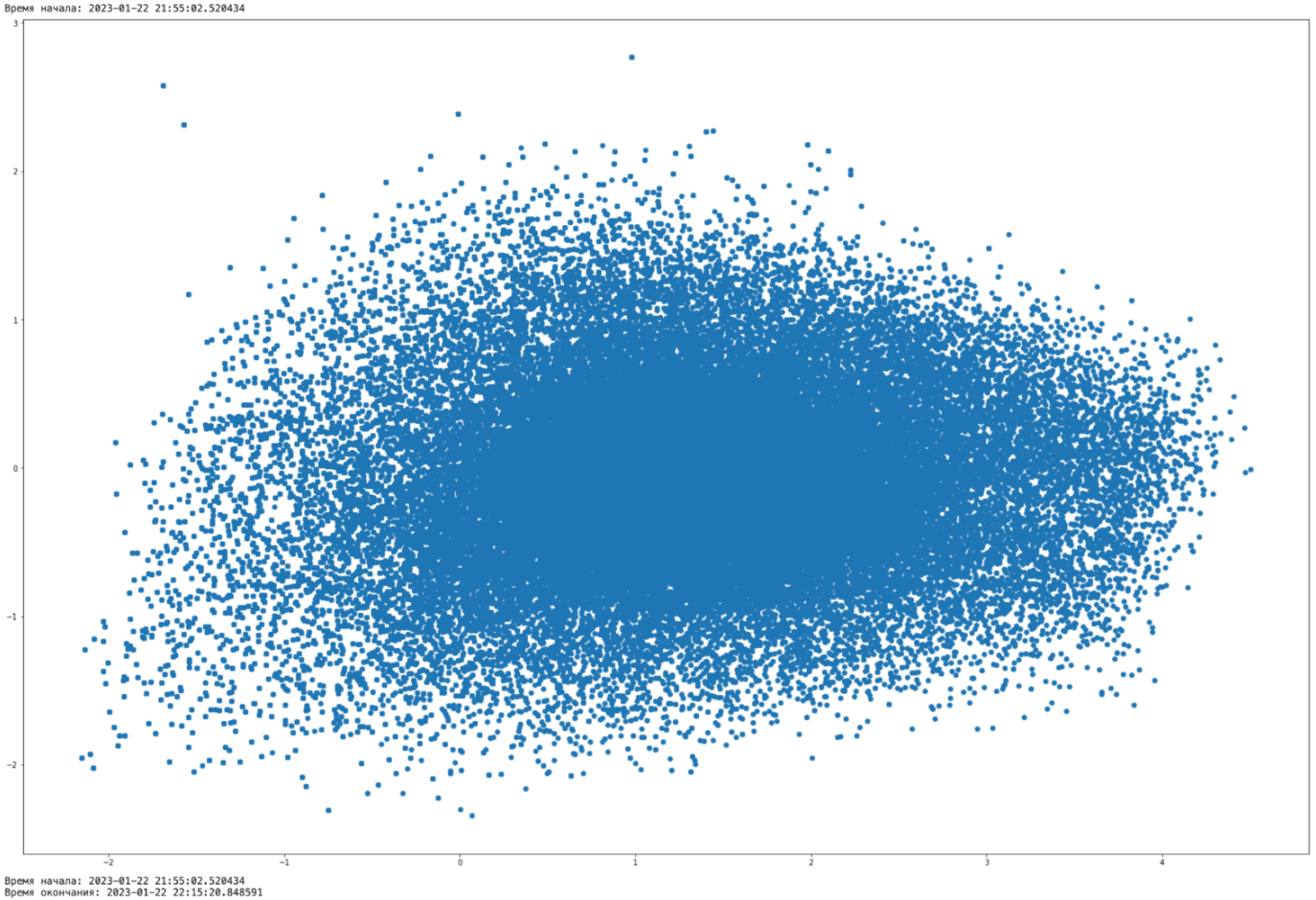

Считаем эмбеддинги для "Хроники Амбера" Р. Желязны.




Считаем эмбеддинги для "Игра престолов" Дж. Мартина.




Считаем эмбеддинги для "Последнее желание. Меч предназначения" А. Сапковского.




Считаем эмбеддинги для "Властелин колец" Дж. Р. Р. Толкина.