

Санкт-Петербургский государственный университет  
Прикладная математика, программирование и искусственный интеллект

Отчет по учебной практике 1 (научно-исследовательской работе) (семестр 2)  
Классификация текстов

Выполнил:  
Миронцев Валентин Олегович



Научный руководитель:  
Кандидат физико-математических наук, доцент  
Голяндина Нина Эдуардовна  
Кафедра статического моделирования

Работа выполнена на отличном уровне и может быть зачтена с  
оценкой А.

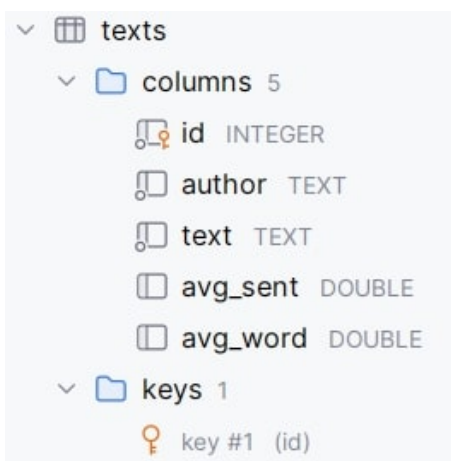


## 1 Введение

Мне была поставлена следующая задача: "Найдите в открытом доступе рассказы двух авторов, которые, на ваш взгляд, должны довольно сильно отличаться по стилю. Нужно придумать алгоритм, который будет распознавать, чей рассказ. Предлагается посчитать среднюю длину слова, среднее число слов в предложении, частоту слов. Если взять две характеристики, то можно нарисовать двумерный график, где по одной оси откладывается, скажем, средняя длина слов, а по другой - еще что-то. Каждый рассказ - одна точка. Точки можно раскрасить согласно автору. Проверьте, что выбраны авторы, для которых линейная классификация имеет смысл, но в то же время видно, что безошибочной классификации не получится. Произведений авторов должно быть несколько сотен, поэтому нужно либо брать рассказы, либо делить большие произведения на части. Далее нужно провести границу в виде прямой линии, которая разделяет точки с минимальной ошибкой - ее и используйте для классификации. Эту границу можно провести просто вручную и получить ее формулу. Второй вариант - написать программу для численного поиска разделяющей прямой, дающей лучшую точность. Сравните эти два способа по точности. Перед исследованием все рассказы надо разделить на две части случайным образом - те, на которых будет строиться классифицирующее правило, и те, на которых потом будет проверяться точность классификации. Результат представляйте в виде таблиц классификации и точности классификации на обучающем и тестовом множествах. Результат также визуализируйте на скаттерплотах."

## 2 Подготовка

Для классификации были выбраны писатели Говард Филлипс Лавкрафт и Эрнест Хемингуэй. Классифицировать мы будем по двум следующим параметрам: среднее количество слов в предложении и средняя длина слов. Для каждого автора было собрано по 100 текстов (рассказы или отрывки романов) на языке оригинала. Все тексты были занесены в базу данных "TestDB.db" в таблицу "texts" со следующей структурой (Рис. 1):



texts
columns 5
id INTEGER
author TEXT
text TEXT
avg_sent DOUBLE
avg_word DOUBLE
keys 1
key #1 (id)

*author*: автор.  
*text*: рассказ или отрывок.  
*avg\_sent*: средняя длина предложения.  
*avg\_word*: средняя длина слова.

Далее для каждого отрывка вычислим среднюю длину предложения и среднюю длину слова с помощью следующего кода на языке Python:

```
1 import sqlite3 as sl
2 import re
3
4 con = sl.connect('TestDB.db')
5 with con:
6     data = con.execute('SELECT * FROM texts')
7     for row in data:
8         s = str(row[2])
9         s = s.translate({ord(i): None for i in ',;\\":---FF[](){}""'})
10        s = s.replace('\\n', ' ').replace('\\r', ' ').replace('NBSP', ' ')
11        sent_len = 0 # summary len of sentences OR number of words
12        word_len = 0
13        a = re.split("[.!?]", s) # a is text divided into sentences
14        for i in range(len(a)):
15            if len(a[i]) == 0:
16                continue
17            a[i] = a[i].split() # divide each sentence into words
18            sent_len += len(a[i])
19            for word in a[i]:
20                word_len += len(word)
21        sql = """UPDATE texts SET avg_sent = ?, avg_word = ? WHERE id = ?"""
22        val = [(sent_len / len(a), word_len / sent_len, row[0])]
23        with con:
24            con.executemany(sql, val)
```

### 3 Классификация

Для классификации разобьём тексты каждого из авторов на 4 группы по 25 текстов. Будем строить прямую для классификации по 75 отрывкам, а на оставшихся 25 — тестировать точность. И так сделаем четыре раза для различных комбинаций этих групп.

Запишем необходимые данные в массивы, по которым будем строить двумерный график и искать прямую (пример кода, для одной из комбинаций):

```
8 sentences_love_train = []
9 words_love_train = []
10 sentences_hem_train = []
11 words_hem_train = []
12 data = con.execute("SELECT * FROM texts WHERE id BETWEEN 1 and 75")
13 for row in data:
14     sentences_love_train.append(row[3])
15     words_love_train.append(row[4])
16 data = con.execute("SELECT * FROM texts WHERE id BETWEEN 101 and 175")
17 for row in data:
18     sentences_hem_train.append(row[3])
19     words_hem_train.append(row[4])
```

Пусть мы имеем целевую функцию  $f(k, b, A)$ , где  $k$  и  $b$  - коэффициенты прямой классификации,  $A$  - множество точек по которым мы строим, прямую, а значение функции - это количество неверно классифицированных точек для данной прямой. Данную функцию можно задать следующим образом:

$$f(k, b, A) = \sum_{(x,y,z) \in A} \mathbf{1}_{(y-kx-b)z < 0}((x, y, z)) = \sum_{(x,y,z) \in A} [(y - kx - b)z < 0]$$

Наша цель: найти такие  $k$  и  $b$ , при фиксированном  $A$ , для которых значение функции будет минимальным.

Найдём прямую перебором коэффициента наклона и свободного члена. Ищем мы её, соответственно, по минимальному суммарному числу текстов первого автора, оказавшихся над линией, и текстов второго автора, оказавшихся над линией. Определив среднее значение каждой характеристики для каждого из авторов (с помощью нескольких SQL-запросов), мы получим две точки: (22.74, 4.56), (10.38, 4.00) — примерные "центры скопления" точек по каждому автору (ниже можно наблюдать график, по которому станет понятно, о чём речь). Из этих данных мы можем сделать вывод, что прямая должна иметь угол наклона в интервале  $(\frac{\pi}{2}, \pi)$  и не очень большой положительный свободный коэффициент (в пределах 100). Найдём прямую с помощью следующего кода:

```

21 k = 0
22 b = 0
23 mn = 1000
24 c = 0
25 alp = pi / 2 + 0.0001
26 while c < 100: # searching line equation
27     alp = pi / 2 + 0.0001
28     while alp <= pi - 0.001:
29         count = 0
30         for i in range(75):
31             if words_love_train[i] < math.tan(alp) * sentences_love_train[i] + c:
32                 count += 1
33             if words_hem_train[i] > math.tan(alp) * sentences_hem_train[i] + c:
34                 count += 1
35         if count < mn:
36             mn = count
37             k = math.tan(alp)
38             b = c
39         alp += 0.01
40     c += 0.01

```

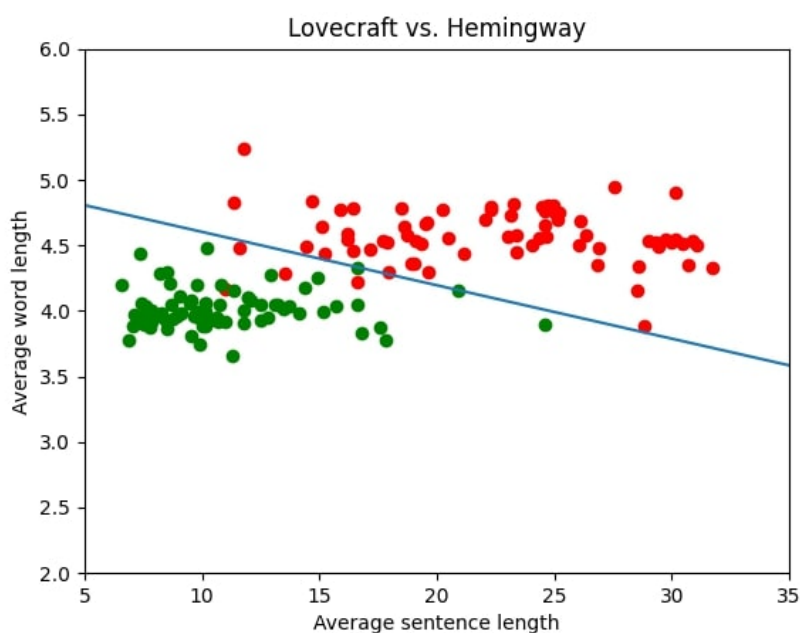
И построим двумерный график, с нанесённой на него прямой (для данного набора тестов — 4 точки находятся по неверную сторону прямой):

```

43 plt.axis([5, 35, 2, 6])
44 plt.title('Lovecraft vs. Hemingway')
45 plt.ylabel('Average word length')
46 plt.xlabel('Average sentence length')
47 plt.plot(sentences_love_train, words_love_train, 'ro')
48 plt.plot(sentences_hem_train, words_hem_train, 'go')
49 plt.axline([0, b], [-b / k, 0])
50 plt.show()

```

Получим вот такой график (красная точка — текст Лавкрафта, зелёная — Хемингуэя):



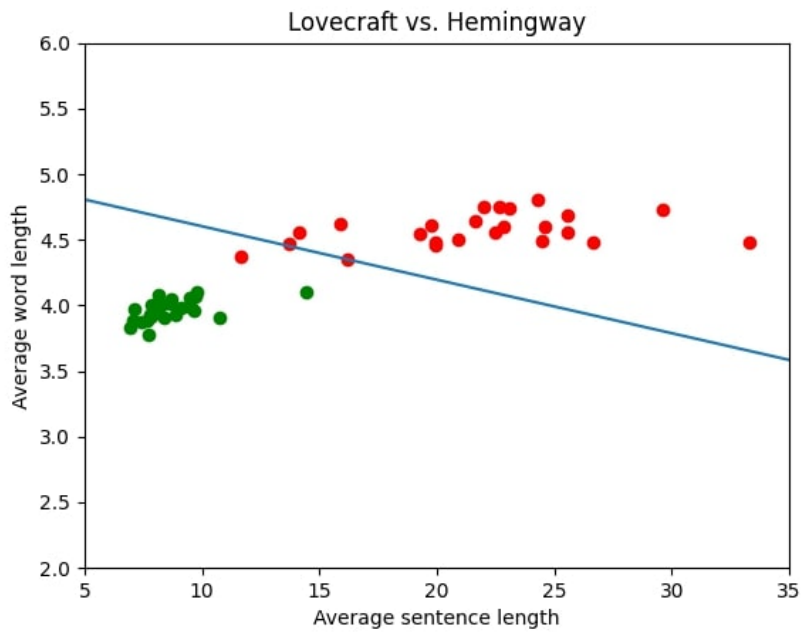
Далее, аналогичным методом, запишем в массивы данные для тестирования, а затем посчитаем количество точек, оказавшихся на неверной стороне (для этих данных их оказалось 2):

```

65 wrong = 0
66 for i in range(25):
67     if words_love_test[i] < k * sentences_love_test[i] + b:
68         wrong += 1
69     if words_hem_test[i] > k * sentences_hem_test[i] + b:
70         wrong += 1
71 print(wrong)

```

И построим график:



## 4 Заключение

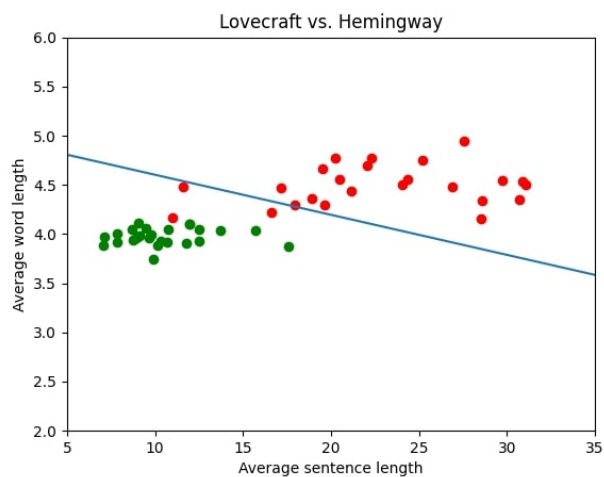
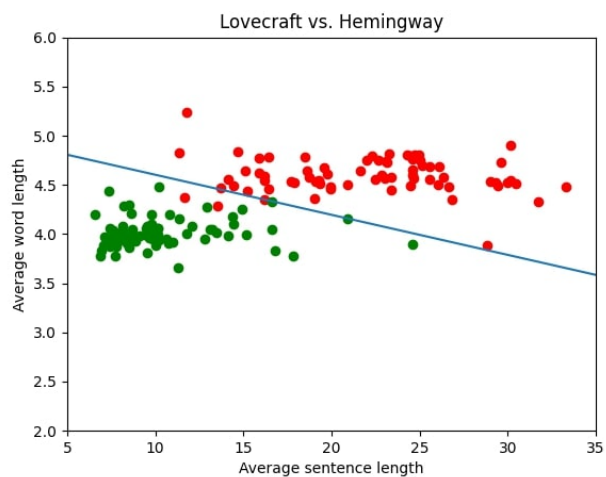
Проделав 4 теста, для различных комбинаций данных, мы получили следующие результаты:

	Кол-во неверных точек		Точность	
	train	test	train	test
1-й тест	4	2	0.053	0.08
2-й тест	3	3	0.04	0.12
3-й тест	3	4	0.04	0.16
4-й тест	3	2	0.04	0.08
Среднее значение	3	2	0.04325	0.11

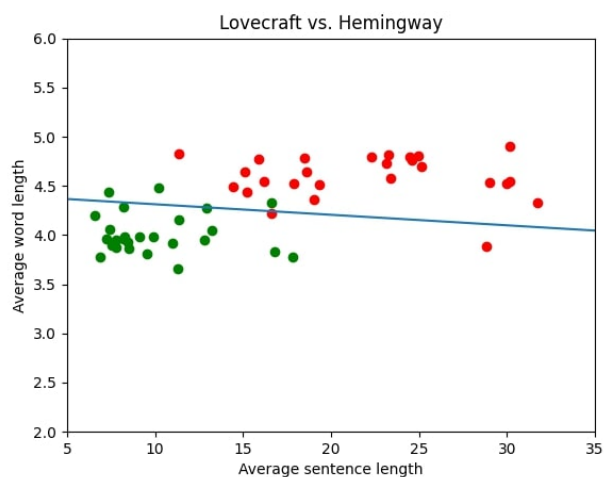
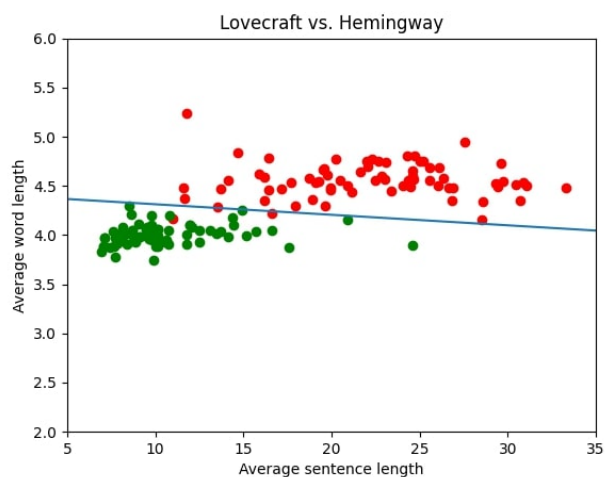
Решая, эту задачу, мы попрактиковали метод линейной классификации, а также затронули использование библиотек SQLite3 и PyPlot для Python.

Ниже изображены графики для остальных тестов:

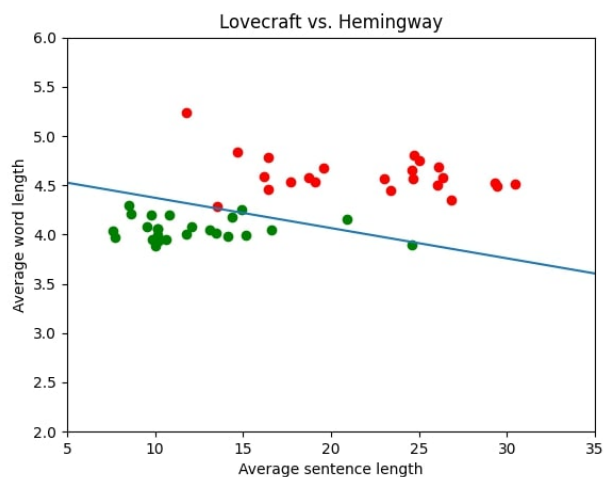
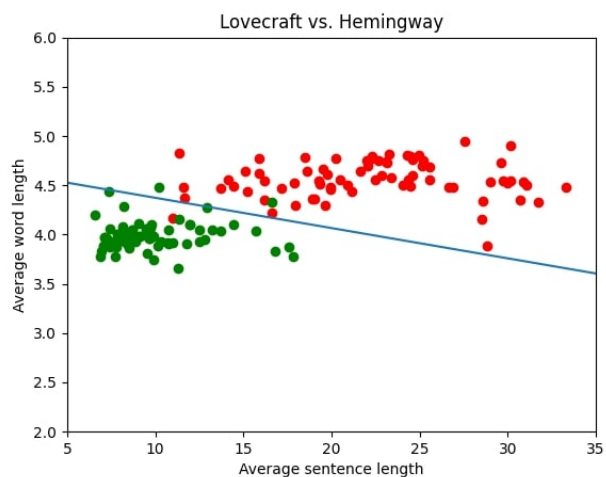
Тест №2:



Тест №3:



Тест №4:



## 5 Источники

- Hemingway E. The Complete Short Stories of Ernest Hemingway: The Finca Vigía Edition. - New York: Simon Schuster Inc., 1987. - 535 с.
- Electronic Texts of H.P. Lovecraft's Works // The H.P. Lovecraft Archive URL: <https://www.hplovecraft.com/writings/texts/> (дата обращения: 06.05.2023).
- A Farewell to Arms // Online Read Free Novel URL: [https://onlinereadfreenovel.com/ernest-hemingway/31949-a\\_farewell\\_to\\_arms\\_read.html](https://onlinereadfreenovel.com/ernest-hemingway/31949-a_farewell_to_arms_read.html) (дата обращения: 08.05.2023).
- The Old Man and the Sea // Online Read Free Novel URL: [https://onlinereadfreenovel.com/ernest-hemingway/31948-the\\_old\\_man\\_and\\_the\\_sea\\_read.html](https://onlinereadfreenovel.com/ernest-hemingway/31948-the_old_man_and_the_sea_read.html) (дата обращения: 08.05.2023).