ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

отформатировано: русский

Журнал практики

Студента **Махмудова Орхана Сакитовича~~Смирнова Даниила Алексеевича~~** _____ (ф. и. о.)

Институт **№8 «Информационные технологии и прикладная математика»**

Кафедра **№805 «Математическая кибернетика»**

Учебная группа М**80-205~~3~~Б-1~~8~~8**

отформатировано: русский

Направление подготовки (специальность) **01.03.04**
(шифр)
Прикладная математика
_____
(название направления, специальности)

Вид практики ***вычислительная***
(учебной, производственной, преддипломной или другой вид практики)

отформатировано: русский

Руководитель практики от МАИ
Волкова Татьяна Борисовна _____
(фамилия, имя, отчество) *(подпись)*

_____/_____/ "~~12~~" 26 ~~февраля~~июня"- 2020 г.

отформатировано: русский

(подпись студента) *(дата)*

1.Место и сроки проведения практики

Сроки проведения практики:

-дата начала практики ***26.06.20***

-дата окончания практики ***12.07.20***

Наименование предприятия ***МАИ***

_____

Название структурного подразделения (отдел, лаборатория)

каф. 805

2. Инструктаж по технике безопасности

_____/_____/ *25.06.20*

(подпись проводившего) *(дата проведения)*

3. Индивидуальное задание студенту

Разработать программу статистического моделирования центрированных гауссовских случайных процессов по заданной ковариационной функции. Провести ее апробацию на примере процессов дробного броуновского движения.

4.План выполнения индивидуального задания

1. Изучить алгоритм моделирования гауссовских случайных процессов и их связь с дробным броуновским движением
2. Написать код программы
3. Показать скриншоты работы программы

~~3. Реализовать соответствующие модели с помощью скриптов Python~~
~~4. Выдать по построенным моделям топ-5 видео к уроку~~
~~5. Оценить полученный результат~~

Руководитель практики от МАИ: *Волкова Т.Б.* /_____/

Руководитель от предприятия *Рыбаков К.А.* /_____/

_____/_____/ “ *26* ” *июня* 20*20 г.*
(подпись студента) *(дата)*

отформатировано: Шрифт: Times New Roman, 14 пт
отформатировано: Шрифт: Times New Roman, 14 пт
отформатировано: Шрифт: Times New Roman, 14 пт
отформатировано: Шрифт: Times New Roman, 14 пт
Отформатировано: По левому краю, Отступ: Слева: 1,25 см
отформатировано: русский
отформатировано: русский

5.Отзыв руководителя практики от предприятия

_____

_____

_____

_____

_____

Материалы, изложенные в отчёте студента, полностью соответствуют индивидуальному заданию

Руководитель от предприятия: *Рыбаков К.А.* /_____/

(фамилия, имя, отчество) *(подпись)*

«12» июля 2020г

М.П. (печать)

отформатировано: русский

6.Отчет студента о практике

Цель: Разработать программу статистического моделирования центрированных гауссовских случайных процессов по заданной ковариационной функции. Провести ее апробацию на примере процессов дробного броуновского движения.

~~**Имеющиеся данные**: Одна .csv таблица: cms_step_main.csv содержит всю информацию о слайдах, и набор файлов с субтитрами к видеоматериалами~~

~~**Бизнес-описание:**~~

~~Есть уроки, которые ведет преподаватель. На них ученикам показываются слайды, видео, тесты и тп. Будем пытаться предугадывать, какое видео показать ученику для закрепления материала. Плюсами этого являются: уменьшение времени, которое учитель проводит на уроке, освобождение учителя от задачи подбора видео, улучшенное понимание материала учеником.~~

План работы:

1. Задать начальную ковариационную функцию для отладки работы программы
2. Построить ковариационную матрицу R, исходя из заданной функции
3. Следуя алгоритму смоделировать нижнетреугольную матрицу A
4. Задать вектор стандартных нормальных величин и найти гауссовский случайный вектор значений
5. Сделать проверку полученных результатов, путём обратных действий
6. Задать ковариационную функцию дробного броуновского движения и найти все выше описанные элементы для этой функции

Описание работы:

В данной работе использовалась среда разработки Python 3-ей версии.

Отформатировано: Отступ: Слева: 0 см, Запретить размещение знаков препинания на полях, Не изменять интервал между восточноазиатскими и латинскими буквами, Выравнивание шрифтов: По опорной линии

отформатировано: русский

отформатировано: русский

отформатировано: Шрифт: не полужирный

отформатировано: русский

Источники: Михайлов Г. А. Войтишек А. В. «Методы Монте-Карло»

Статья про дробное броуновское движение на Wikipedia

https://en.wikipedia.org/wiki/Fractional_Brownian_motion

На начальном этапе нужно было задать ковариационную функцию, для дальнейшего моделирования ковариационной матрицы R, руководителем практики была предложена функция $K_t = \min(t,s)$, из которой получается положительно-определенная симметрическая матрица R, что удовлетворяет условиям заданным алгоритмом в учебник

Строится ковариационная матрица R по следующей формуле: $R_{ij} = \mathbf{E}\left((\eta_i - m_i)(\eta_j - m_j)\right).$

Далее по алгоритму нам нужно построить нижнетреугольную матрицу A, которая строится по рекурентной формуле приведенной ниже:

$$a_{ij} = \frac{R_{ij} - \sum_{k=1}^{j-1} a_{ik}\, a_{jk}}{\sqrt{R_{jj} - \sum_{k=1}^{j-1} a_{jk}^2}},$$

~~1) Распарсить файлы с текстовым контентом~~

~~2) Лемматизировать слова в данных файлах~~

~~3) Выполнить векторизацию слов различными методами~~

~~4) Выбрать метрику для вычисления схожести текстов и выполнить подсчёт~~

~~5) Сделать выводы о проделанной работе~~

~~**Виды задач:**~~

~~Вариации решения проблемы возникают на этапе векторизации слов контента и выбора метрики вычисления схожести документов. На первом этапе нужно распарсить .csv файл и файлы с субтитрами для дальнейшей работы. Эта часть выполнена с применением регулярных выражений библиотеки re.~~

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: Шрифт: 13 пт

причем $\sum_{k=1}^{0} a_{ik}\, a_{jk} = 0,\ 1 \le j \le i \le l.$

Отформатировано: По левому краю

Следующим шагом мы задаем вектор стандартных нормальных величин используя формулы:

$$\xi_1 = \sqrt{-2\ln\alpha_1}\sin 2\pi\alpha_2, \quad \xi_2 = \sqrt{-2\ln\alpha_1}\cos 2\pi\alpha_2,$$

~~Парсинг слайдов~~

In [1]:
```
from pprint import pprint
import re
import pandas as pd
from tqdm import tqdm
```

In [3]:
```
with open('/home/daniil/Рабочий стол/Научная работа/cms_step.csv/cms_step_main.csv') as f:
    content = f.read()
content = re.sub(r'(?m)^\\.*\n?', ' ', content)
chunks = re.split(r'(\n"201\d-\d\d-\d\d \d\d:\d\d:\d\d\.??\d{0,3})', content)
columns = re.sub(r'"', '', chunks.pop(0)).split(',')
columns.remove('type_desc')
str_slides = list(map(lambda x, y: x + y, chunks[0::2], chunks[1::2]))
slides = [i for i in str_slides if "<vim-content-section>" in i]
```

In [5]:
```
for i in range(len(slides)):
    if 'abou ' in slides[i]:
        print(i)
```

```
14096
14205
22255
36949
41478
59235
```

In [11]:
```
slides_content[0]
```

Out[11]:
```
" Read the text again and complete the answers to the questions My family Hi! My name's Ian Haig and I'm from Glasg
ow in Scotland. I'm not English &ndash; I'm Scottish! I have a big family. My father's name is Gordon and he's 50.
My mother is 45. Her name's Anna and she's very nice. My mother's Scottish but her mother and father are Italian. T
hey're from Milan. I have three sisters. Their names are Rosie , Jenny , and Valeria. Valeria is an Italian name. R
osie is 26. She's tall and beautiful , and she's married. Her husband's name is Tom and he's very tall. Jenny and V
aleria are 24 , but Jenny is tall with short hair and Valeria is short with long hair. We have a big , new house an
d an old car. It's slow and cheap , and very small! But it's OK! My mother and father are very short! Me? Well ,
I'm 21 , I'm not tall , and I'm slim with short hair. I have a girlfriend but I'm not married. Her name is Lucy and
she's very beautiful. She's tall with short , dark hair. She's English but she's nice! What is Ian's surname? It's
. Where is Ian from?  He's from Glasgow Glasgow in Scotland . What is Ian's father's name?  It's Gordon . How old i
s Ian's mother?  She's 45 forty-five . Where is Anna's mother from? She's from Italy Milan Milan , Italy . What are
Jenny's sisters' names?  Their names are Rosie , Valeria Rosie and Valeria Valeria , Rosie Valeria and Rosie . What
is Rosie's husband's name?  His name is Tom . Is their car fast and expensive?  No , it's slow and cheap slow , che
ap slow and cheap , and very small slow , cheap , small . Where is Ian's girlfriend from?  She's from England . "
```

In [12]:
```
print(slides_descr[2])
```

```
['2018-04-19 16:57:08.074', '2018-04-19 16:31:55.489', '1524155515362', '1493376238359', '0', '2018-01-19 14:04:1
4', 'f', '153525.000000', '1', '52608.000000', 't', '', '5912.000000', '12', '0', '2', 'Thanking via e-mail', '2018
-04-19 16:30:37', 'lesson', 'f', '']
```

In [13]:
```
drop_columns = ['emphasis_distribution', 'content_backup', 'is_active', 'cms_step_receptacle_id', 'adaptive_level']
for i in drop_columns:
    columns.remove(i)
```

In [14]:
```
df = pd.DataFrame(slides_descr, columns=columns)
```

In [15]:
```
drop_columns = ['is_deleted_forever','content', 'lesson_hw']
df.drop(drop_columns, axis=1, inplace=True)
```

In [16]:
```
df['slide_content'] = slides_content
```

In [17]:
```
df.head()
```

Out[17]:

	_sdc_batched_at	_sdc_received_at	_sdc_sequence	_sdc_table_version	comments	created_at	deleted	edited_by	emphasis	id	is_inte
0	2018-04-11 21:44:28.867	2018-04-11 21:26:47.199	1523482007087	1493376238359	0	2016-12-17 15:39:40	f	57060.000000	1	17613.000000	
1	2018-04-11 21:44:50.447	2018-04-11 21:27:22.828	1523482042683	1493376238359	0	2017-10-11 13:30:31	f	152948.000000	5	43666.000000	
2	2018-04-19 16:57:08.074	2018-04-19 16:31:55.489	1524155515362	1493376238359	0	2018-01-19 14:04:14	f	153525.000000	1	52608.000000	
3	2018-04-11 21:44:59.352	2018-04-11 21:27:44.10	1523482063970	1493376238359	0	2018-03-28 14:19:06	f	228426.000000	6	58017.000000	

In [18]:
```
df[df.index==86]
```

Out[18]:

	_sdc_batched_at	_sdc_received_at	_sdc_sequence	_sdc_table_version	comments	created_at	deleted	edited_by	emphasis	id	is_inter
86	2018-04-11 21:44:58.819	2018-04-11 21:27:42.423	1523482062338	1493376238359	0	2018-03-19 15:44:54	f	643630.000000	6	57029.000000	

In [19]:
```
df.to_csv('prepared_data.csv', index=False)
```

In [20]:
```
df1 = pd.read_csv('prepared_data.csv')
```

In [21]:
```
df1.head()
```

Out[21]:

	_sdc_batched_at	_sdc_received_at	_sdc_sequence	_sdc_table_version	comments	created_at	deleted	edited_by	emphasis	id	is_interactive	less
0	2018-04-11 21:44:28.867	2018-04-11 21:26:47.199	1523482007087	1493376238359	0	2016-12-17 15:39:40	f	57060.0	1.0	17613.0	t	1
1	2018-04-11 21:44:50.447	2018-04-11 21:27:22.828	1523482042683	1493376238359	0	2017-10-11 13:30:31	f	152948.0	5.0	43666.0	t	5
2	2018-04-19 16:57:08.074	2018-04-19 16:31:55.489	1524155515362	1493376238359	0	2018-01-19 14:04:14	f	153525.0	1.0	52608.0	t	5
3	2018-04-11 21:44:59.352	2018-04-11 21:27:44.10	1523482063970	1493376238359	0	2018-03-28 14:19:06	f	228426.0	6.0	58017.0	t	6
4	2018-04-11 21:44:31.686	2018-04-11 21:26:51.932	1523482011828	1493376238359	0	2017-03-06 13:57:07	t	38908.0	1.0	22804.0	t	

~~Парсинг субтитров~~

где (α1, α2) – пара независимых случайных чисел, нормально распределенных на отрезке (0,1)

Отсюда, зная матрицу A и вектор стандартных нормальных величин $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_l)$ можно найти гауссовский случайный вектор по формуле $\boldsymbol{\eta} = \mathbf{A}\boldsymbol{\xi} + \mathbf{m}$

где $m = (m_1 \ldots m_l)$ – вектор математических ожиданий, в рамках конкретной задачи он был нулевым

После построения всех матриц и векторов, была проделана проверка вектора математических ожиданий и матрицы R, которые оказались примерно равными с погрешностью ~ 0.01

Далее вместо ковариационной функции заданной в начале, для отладки работы программы, была введена функция дробного броуновского движения: $E[B_H(t)B_H(s)] = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H})$ где параметр H был взят равным ¼ и ¾.

Код программы:

```
import math
import numpy as np

def Gauss ():

    #создание массива времени t
    step = 1
    tstart = step
    tfinish = 6
    t = np.arange(tstart, tfinish, step)
    N = t.size

    #H = 0.25
    H = 0.25

        print ()
        print('параметр H =', H)
        # моделирование ковариционной матрицы R
        R = np.zeros((N,N))
        i = 0
        j = 0
```

Отформатировано: По левому краю

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: русский

отформатировано: Шрифт: 12 пт, не полужирный

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: Шрифт: не полужирный, английский (США)

Отформатировано: По центру

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

```
        while (i < N):
            j = 0
            while (j < N):
                R[i,j] = (pow(abs(t[i]),(2*H)) + pow(abs(t[j]),(2*H)) - pow(abs(t[i]-t[j]),2*H))/2
                j = j + 1
            i = i + 1

        # моделирование нижнетреугольной матрицы A
        A = np.zeros((N,N))
        i = 0
        for i in range(N):
            j = 0
            for j in range(i+1):
                sum1 = 0
                if (i == j):
                    k = 0
                    for k in range(j):
                        sum1 = sum1 + pow(A[j][k], 2)
                    A[i][j] = math.sqrt(R[j][j] - sum1)
                else:
                    k = 0
                    for k in range(j):
                        sum1 = sum1 + (A[i][k] * A[j][k])
                    A[i][j] = (R[i][j] - sum1) / A[j][j]
                j = j + 1
            i = i + 1

        print(R, 'Ковариционная матрица R')
        #print (np.linalg.cholesky(R))
        print()
        print(A, 'Нижнетреугольная матрица A')

        # генерация вектора стандартных нормальных величин и вектора значений
        b = 0
        if (math.fmod(N,2) != 0):
            N = N + 1
            b = 1
        alfa = np.random.uniform(0,1,N)
        ksi = np.zeros(N)
        i = 0
        while (i < alfa.size):
             ksi1 = math.sqrt(-2*math.log(alfa[i]))*math.sin(2*math.pi*alfa[i+1])
             ksi[i] = ksi1
             ksi2 = math.sqrt(-2*math.log(alfa[i]))*math.cos(2*math.pi*alfa[i+1])
             ksi[i+1] = ksi2
             i = i + 2
        if (b == 1):
            ksi = np.delete(ksi,-1)
            N = N - 1
        teta = A.dot(ksi)
        print()
        print (alfa.round(3) , 'Вектор альфа')
        print (ksi.round(3), 'Вектор стандартных нормальных величин')
        print(teta.round(3), 'Вектор значений')

        # проверка
        M = np.zeros(N)
         R = np.zeros((N,N))

        b = 0
        if (math.fmod(N,2) != 0):
            N = N + 1
            b = 1
        for i in range(10000):

            alfa = np.random.uniform(0,1,N)
```

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

```
            ksi = np.zeros(N)
            i = 0
            while (i < alfa.size):
                ksi1 = math.sqrt(-2*math.log(alfa[i]))*math.sin(2*math.pi*alfa[i+1])
                ksi[i] = ksi1
                ksi2 = math.sqrt(-2*math.log(alfa[i]))*math.cos(2*math.pi*alfa[i+1])
                ksi[i+1] = ksi2
                i = i + 2
            if (b == 1):
                ksi = np.delete(ksi,-1)
            teta = A.dot(ksi)
            teta = teta.reshape(1,-1)
            teta_t = teta.reshape(-1,1)
            A_t = teta_t.dot(teta)
             M =  M + (teta/10000)
             R =  R + (A_t/10000)
        print()
        print( M.round(3), 'Проверка вектора мат. ожиданий')
        print()
        print( R.round(3), 'Проверка матрицы R')
    return 0

print(Gauss())
```

Результат работы программы:

При H = 0.25

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: русский

```
параметр H = 0.25
[[1.         0.70710678 0.65891862 0.6339746  0.61803399]
 [0.70710678 1.41421356 1.07313218 1.         0.95911537]
 [0.65891862 1.07313218 1.73205081 1.3660254  1.27695261]
 [0.6339746  1.         1.3660254  2.         1.61803399]
 [0.61803399 0.95911537 1.27695261 1.61803399 2.23606798]] Ковариционная матрица R

[[1.         0.         0.         0.         0.        ]
 [0.70710678 0.95614516 0.         0.         0.        ]
 [0.65891862 0.63505667 0.94582244 0.         0.        ]
 [0.6339746  0.57701727 0.61517791 0.94163869 0.        ]
 [0.61803399 0.54604611 0.55290323 0.60639421 0.93942519]] Нижнетреугольная матрица A

[0.466 0.694 0.1   0.063 0.191 0.713] Вектор альфа
[-1.161 -0.423  0.832  1.977 -1.772] Вектор стандартных нормальных величин
[-1.161 -1.226 -0.247  1.393 -0.955] Вектор значений

[[ 0.008 -0.002 -0.009 -0.006  0.006]] Проверка вектора мат. ожиданий

[[1.007 0.712 0.673 0.645 0.621]
 [0.712 1.414 1.097 1.012 0.979]
 [0.673 1.097 1.753 1.372 1.3  ]
 [0.645 1.012 1.372 1.982 1.619]
 [0.621 0.979 1.3   1.619 2.249]] Проверка матрицы R
0
Press any key to continue . . .
```

При H = 0.75

```
параметр H = 0.75
[[ 1.          1.41421356  1.68386265  1.90192379  2.09016994]
 [ 1.41421356  2.82842712  3.51228977  4.          4.40630729]
 [ 1.68386265  3.51228977  5.19615242  6.09807621  6.77403259]
 [ 1.90192379  4.          6.09807621  8.          9.09016994]
 [ 2.09016994  4.40630729  6.77403259  9.09016994 11.18033989]] Ковариционная матрица R

[[1.         0.         0.         0.         0.        ]
 [1.41421356 0.91017972 0.         0.         0.        ]
 [1.68386265 1.24255502 0.90377875 0.         0.        ]
 [1.90192379 1.43957677 1.22457453 0.90040093 0.        ]
 [2.09016994 1.59348816 1.41016458 1.21504298 0.89857186]] Нижнетреугольная матрица A

[0.519 0.58  0.492 0.164 0.745 0.886] Вектор альфа
[-0.553 -1.003  1.02   0.616 -0.505] Вектор стандартных нормальных величин
[-0.553 -1.695 -1.255 -0.691 -1.02 ] Вектор значений

[[-0.024 -0.03  -0.031 -0.036 -0.026]] Проверка вектора мат. ожиданий

[[ 1.009  1.43   1.721  1.942  2.134]
 [ 1.43   2.85   3.571  4.065  4.472]
 [ 1.721  3.571  5.311  6.229  6.908]
 [ 1.942  4.065  6.229  8.143  9.236]
 [ 2.134  4.472  6.908  9.236 11.322]] Проверка матрицы R
0
Press any key to continue . . .
```

отформатировано: английский (США)

Выводы

В ходе выполнения работы, были получен необходимый опыт работы в среде Python. Был запрограммирован алгоритм численного моделирования гауссовского процесса, необходимого для решения задач связанных с дробным броуновским движением. Данный алгоритм применяется в прогнозировании финансовых активов, именно броуновское движение очень хорошо описывает данную модель, так как непрерывность цен и последовательное их изменение и есть независимые гауссовские случайные величины. Наиболее подробное описание данной модели основывается на показателе Хёрста - H, при $H = ½$ у нас броуновское движение является обычным и график цен будет соответствовать нормальному распределению и являться случайным. При H от 0 до 0,5 появляется зависимость между ценами, но она обратная и при H от 0,5 до 1, , если мы наблюдаем восходящую тенденцию, то в будущем она продолжится. (http://fxtreder.ru/fondovyj-rynok/stati/496-praktikum-trejdera-brounovskoe-dvizhenie-kak-model-dlya-prognozirovaniya-finansovykh-aktivov)

~~После парсинга стоит задача лемматизировать и векторизовать каждый файл с текстовым контентом.~~

~~**Лемматизация .**~~

отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: русский
отформатировано: Шрифт: не полужирный
Отформатировано: По левому краю

In [5]:
```
for i in range(len(df1.content)):
    l = lemmatize_sentence(df1.content[i])
    for j in l:
        if j == "." or j == "!" or j == "?" or j == "," or j == "'s" or j == "'m" or j == "'re" or j == "n't" :
            l.remove(j)
    df_lem_subs = df_lem_subs.append({'file': df1.file[i], 'content': ' '.join(l)}, ignore_index=True)
```

In [7]:
```
df_lem_subs
```

Out[7]:

	file	content
0	031e81a6d7638d05a9c04aecd5654ce6.vtt	i have five speak tip for you whenever you in ...
1	bb1032d99f8bd74fa6b0b9e4925627b4.vtt	it kill an estimated 540,000 people around the...
2	e6ee8f4af1a18180a54e163fbc538af5.vtt	we wish you a merry christmas we wish you a me...
3	d7366f0e0c4835412d938b866b4442b4.vtt	for many people around the world january 1 off...
4	d1b05a3762b133cf2242c69a15494f73.vtt	lydia lydia lydia um yeah i 've hear you… ⊢— n...
...	...	...
2106	b15b16d7e362ba29cf7e67d3482329cf.vtt	pull pull pull you can win this tug of war oh ...
2107	5aef88b9f3985947c05b058d4f434f8f.vtt	i 've get the french revolution here ... ameri...
2108	d14fb54198275298b6eb2059141dee31.vtt	oil company drill well deep into the ground th...
2109	8e77528700cc499bf23732f69aae59e6.vtt	let start with one two and three what's—where ...
2110	b0a6e921425af988dde3e920d24afff5.vtt	i be pam this be my duck puzzle duck puzzle /ʌ...

2111 rows × 2 columns

~~На этапе векторизации было опробовано несколько вариантов:~~

~~1) word2vec векторизация~~

~~2) tf-idf векторизация~~

~~TF-IDF (~~

~~term frequency - inverse document frequency) - статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.~~

$$\mathrm{tf}(t,d) = \frac{n_i}{\sum_k n_k}$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|(d_i \supset t_i)|}$$

$$\text{tf-idf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

отформатировано: английский (США)

отформатировано: английский (США)

отформатировано: английский (США)

Отформатировано: По левому краю

In [1]:
```
from pprint import pprint
import re
import pandas as pd
from tqdm import tqdm
```

In [2]:
```
df1 = pd.read_csv('prepared_subtitles.csv')
```

In [3]:
```
df1
```

Out[3]:

	file	content
0	031e81a6d7638d05a9c04aecd5654ce6.vtt	I have five speaking tips for you whenever yo...
1	bb1032d99f8bd74fa6b0b9e4925627b4.vtt	It kills an estimated 540,000 people around t...
2	e6ee8f4af1a18180a54e163fbc538af5.vtt	We wish you a Merry Christmas. We wish you a ...
3	d7366f0e0c4835412d938b866b4442b4.vtt	For many people around the world, January, 1 ...
4	d1b05a3762b133cf2242c69a15494f73.vtt	Lydia? Lydia? LYDIA?!? Um, yeah, I've heard y...
...	...	...
2106	b15b16d7e362ba29cf7e67d3482329cf.vtt	Pull! Pull! Pull! You can win this tug of war...
2107	5aef88b9f3985947c05b058d4f434f8f.vtt	I've got the French revolution here ... Ameri...
2108	d14fb54198275298b6eb2059141dee31.vtt	Oil companies drill wells deep into the groun...
2109	8e77528700cc499bf23732f69aae59e6.vtt	Let's start with one, two and three. What's—w...
2110	b0a6e921425af988dde3e920d24afff5.vtt	I am Pam. This is my duck puzzle. Duck puzzle...

2111 rows × 2 columns

In [4]:
```
import collections

def compute_tf(text):
#На вход берем текст в виде списка (list) слов
    #Считаем частотность всех терминов во входном массиве с помощью
    #метода Counter библиотеки collections
    tf_text = collections.Counter(text)
    for i in tf_text:
        #для каждого слова в tf_text считаем TF путём деления
        #встречаемости слова на общее количество слов в тексте
        tf_text[i] = tf_text[i]/float(len(text))
    #возвращаем объект типа Counter с TF всех слов текста
    return tf_text
```

In [5]:
```
compute_tf(df1.content[0].split())
```

Out[5]:
```
Counter({'I': 0.02356902356902357,
         'have': 0.016835016835016835,
         'five': 0.010101010101010102,
         'speaking': 0.006734006734006734,
         'tips': 0.010101010101010102,
         'for': 0.003367003367003367,
         'you': 0.03367003367003367,
         'whenever': 0.003367003367003367,
         "you're": 0.003367003367003367,
         'in': 0.006734006734006734,
         'public.': 0.003367003367003367,
         'The': 0.013468013468013467,
         'first': 0.003367003367003367,
         'is': 0.016835016835016835,
         'to': 0.020202020202020204,
         'know': 0.013468013468013467,
         'your': 0.010101010101010102,
         'content.': 0.003367003367003367,
         'Winging': 0.003367003367003367,
         'it': 0.010101010101010102,
```

Отформатировано: По левому краю

In [10]:
```
df = pd.DataFrame(columns=['word', 'value'])
```

In [11]:
```
df
```

Out[11]:

word	value

In [12]:
```
for dcs in vecs:
    for word, value in dcs.items():
        df = df.append({'word':word, 'value':value}, ignore_index=True)
```

In [14]:
```
df.head()
```

Out[14]:

	word	value
0	I	0.033736
1	have	0.029862
2	five	0.020158
3	speaking	0.014624
4	tips	0.020158

In [23]:
```
df.to_csv('subtitles_vectorized.csv', index=False)
```

~~3) CountVectorizer векторизация из sklearn.feature_extraction.text библиотеки~~

~~После оценки всех способов векторизация было принято решение использовать CountVectorizer векторизацию из sklearn.feature_extraction.text библиотеки для дальнейшего вычисления метрики.~~

~~Векторизация:~~

In [13]:
```
from collections import Counter
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

def get_vectors(*strs):
    text = [t for t in strs]
    vectorizer = CountVectorizer(text)
    vectorizer.fit(text)
    return vectorizer.transform(text).toarray()
```

~~рпрпрпрпавав~~

~~Последним этапом было использование метрики cosine similarity для подготовленных текстов.~~

~~Ручная реализация~~

In [101]:
```
cos_sim = 0
for i in range(len(df_slide)):
    word = df_slide.word[i]
    value = df_slide.value[i]
    if word in df_sub["word"].values:
        cos_sim += value * df_sub.loc[df_sub.loc[df_sub['word'] == word].index.tolist()[0]].value
cos_sim = cos_sim / (length_slide*length_sub)
cos_sim
```

~~Вариант 2:~~

Отформатировано: По левому краю

Отформатировано: По левому краю, междустрочный, одинарный

Отформатировано: междустрочный, одинарный

Отформатировано: По левому краю, междустрочный, одинарный

Отформатировано: По левому краю, Отступ: Слева: 0 см, междустрочный, одинарный

~~**Решение:**~~
~~**1 способ**~~

~~**1) прогноз количество уроков в определенные тайм-слоты с помощью методов регрессии или временных рядов.**~~
~~**2) с учетом распределения занятости преподавателей рассчитать их количество, чтобы покрыть все эти промежутки времени. Оценить доверительный интервал сколько нужно преподавателей.**~~

~~**2 способ**~~

~~**1) посчитать количество преподавателей, которые покрывали эти слоты и прогнозировать результаты напрямую.**~~
~~**2) оценить распределение количества уроков во времени, распределение начала уроков во времени, количество преподавателей и учеников.**~~

~~**Вариант 3:**~~
~~**прогнозировать количество преподавателей, учитывая приток и отток учителей и учеников.**~~

~~**Расчёт количества учеников, которые начнут, продолжат и перестанут учиться в определенные периоды времени, расчёт Customer Retention Rate и churn rate, прогноз статистик на будущее. Сезонность данных можно разложить на тренд и сезонную (недельную/месячную) составляющие.**~~

~~**Оценка качества прогноза: Разделить исходные данные на тестовые и проверочные, например взяв последний месяц в качестве проверочных данных. Далее при расчете качества оценить нужную метрику по какой-либо статистике, допустим среднее абсолютное отклонение.**~~
~~**Реализация прямого прогнозирования числа преподавателей по месяцам**~~

Отформатировано: Отступ: Слева: 0 см, междустрочный, одинарный

Отформатировано: междустрочный, одинарный

Отформатировано: Отступ: Слева: 0 см, междустрочный, одинарный

Отформатировано: По левому краю, Отступ: Слева: 0 см, междустрочный, одинарный

отформатировано: русский

~~Проверка модели через Sci-Kit~~

отформатировано: русский

~~Полученная модель полностью совпадает с прошлой~~

~~Реализация прогнозирования в зависимости от кол-ва уроков~~

Отформатировано: По левому краю

отформатировано: русский

Отформатировано: По левому краю, Отступ: Слева: 0 см, междустрочный, одинарный

Отформатировано: По левому краю, междустрочный, одинарный

Отформатировано: По левому краю

отформатировано: русский

Отформатировано: По левому краю, Отступ: Слева: 0 см, междустрочный, одинарный

Отформатировано: По левому краю, междустрочный, одинарный

Отформатировано: По левому краю

~~Реализация прогнозирования с учетом сезонности~~

отформатировано: русский

~~Реализация прогнозирования в зависимости от кол-ва учеников и сезонности~~

Отформатировано: Отступ: Слева: 0 см, междустрочный, одинарный

Отформатировано: По левому краю, Отступ: Слева: 0 см, междустрочный, одинарный

Отформатировано: По левому краю, междустрочный, одинарный

Реализация прогнозирования через коэффициенты оттока и притока

...

Вычисление косинусной схожести текстов с использованием sklearn для выбранного слайда

отформатировано: русский

отформатировано: русский

отформатировано: русский

```
In [57]: list_d = list(cos_sim.items())
         list_d.sort(key=lambda i: i[1], reverse=True)
         list_d[:5]

Out[57]: [('acff6091017453b6cf5ccd2e97c6effc.vtt', 0.5489933680931794),
          ('2429363d85604bdc84edcebfbd1f742b.vtt', 0.5382809704913228),
          ('3322f85a167450dddf6930f72e10c256.vtt', 0.5322670336146142),
          ('87607a8b89d8ca6a5b12ffb569541cca.vtt', 0.5288854508779627),
          ('6ef6416ce33bbeaeac7c73be4df3579e.vtt', 0.5282387126141093)]
```

~~Результатом работы является список кортежей имя файла – значение метрики.~~

Отформатировано: Отступ: Слева: 0 см, междустрочный, одинарный

Отформатировано: По левому краю, Отступ: Слева: 0 см, междустрочный, одинарный

~~Выводыд~~

Отформатировано: междустрочный, одинарный

~~д~~

Отформатировано: По левому краю, Отступ: Слева: 0 см, междустрочный, одинарный

~~Таким образом, оценивая качество прогнозов всех построенных моделей средней абсолютной ошибкой, получаем, что лучшей моделью является прогноз с использованием коэффициентов оттока и притока.~~
~~Худшим – прогноз на основе количества учеников.~~

Отформатировано: По левому краю, междустрочный, одинарный

~~В ходе исследований было выявлено, что использование векторизации из библиотеки sklearn и метрики cosine similarity будет давать наилучшие результаты. Тексты в топ-5 списке, полученные в результате работы, соответствуют тексту слайда на значение полученной метрики.~~

Отформатировано: Отступ: Слева: 0 см, междустрочный, одинарный

~~Примечание:~~

~~1. На титульном листе после подписи студента должна ставиться дата окончания практики;~~

~~2. В п.№3 записывается формулировка темы задания на практику;~~

~~3. План выполнения индивидуального задания (п.№4) оформляется на отдельном листе, и после подписи студента должна ставиться дата начала практики;~~

~~4. В п.№5 желательно руководителем практики от предприятия завершать отзыв фразой: «Материалы, изложенные в отчёте студента, полностью (или не полностью) соответствуют индивидуальному заданию».~~

~~5. Отзыв руководителя практики от предприятия пишется на отдельном листе;~~

~~6. Отчет студента по практике пишется на отдельных листах и его объем устанавливается руководителем практики от МАИ.~~

Отформатировано: Отступ: Слева: 0 см

Отформатировано: Отступ: Слева: 0 см, Первая строка: 0 см