БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
Ханты-Мансийского автономного округа – Югры  
«Сургутский государственный университет»

Политехнический институт

Кафедра прикладной математики

**ОТЧЁТ**

по теме: «Сравнение векторов»

по дисциплине: «Управление неструктурированной информацией»

Выполнил:

студент группы 601-31м  
Янченко Ксения Сергеевна

Проверил:

Профессор кафедры прикладной математики,

Чалей Иван Вацлавович

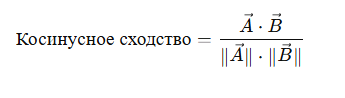
**г. Сургут**2024 г.

Косинусные сходства между вектором знаний Ксении Янченко и остальными:

1. С вектором [52,77,27,92,35]: **0.82**
2. С вектором [85,18,12,57,95]: **0.94**
3. С вектором [15,50,80,30,45]: **0.71**
4. С вектором [77,13,54,33,66]: **0.89**
5. С вектором [57,17,37,25,77]: **0.94**

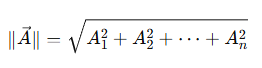
Наибольшая схожесть наблюдается с векторами **2** и **5** (значения 0.94). ​

Для расчета косинусного сходства использовалась следующая формула:



**Объяснение:**

1. **Скалярное произведение векторов** : Это сумма произведений соответствующих элементов двух векторов:
2. 
3. **Длины векторов (нормы)** Длина вектора вычисляется по формуле Евклидовой нормы:



Аналогично для 

1. **Результат**: Формула вычисляет угол между двумя векторами в многомерном пространстве. Чем ближе результат к 1, тем меньше угол между векторами, и тем более они схожи.

**Практическая интерпретация:**

Косинусное сходство используется для сравнения направлений векторов, игнорируя их длину. В данном контексте мы оценивали схожесть "векторов знаний", что позволяет понять, насколько близки они по направлениям, то есть, насколько похожи их структуры.

import numpy as np

# Вектор знаний Ксении Янченко

ksenia\_vector = np.array([56, 43, 14, 28, 65])

# Остальные векторы

other\_vectors = [

np.array([52, 77, 27, 92, 35]), #Иванчук Александра

np.array( [85, 18, 12, 57, 95]), #Островский Артем

np.array([15, 50, 80, 30, 45]), #Денисова Наталья

np.array([77, 13, 54, 33, 66]), #Дедяева Дарья

np.array([57, 17, 37, 25, 77]) #Усманов Ильнур

]

# Функция для вычисления косинусного сходства

def cosine\_similarity(vec1, vec2):

dot\_product = np.dot(vec1, vec2)

norm\_vec1 = np.linalg.norm(vec1)

norm\_vec2 = np.linalg.norm(vec2)

return dot\_product / (norm\_vec1 \* norm\_vec2)

# Вычисляем косинусное сходство для каждого из векторов

similarities = [cosine\_similarity(ksenia\_vector, vec) for vec in other\_vectors]

# Находим индекс вектора с максимальным сходством

max\_similarity\_index = np.argmax(similarities)

# Выбираем наиближайший вектор

closest\_vector = other\_vectors[max\_similarity\_index]

№Выводим на экран

print(similarities,max\_similarity\_index, closest\_vector)

Косинусное сходство всех векторов[0.8219309965947343, 0.9445409882173388, 0.7106732144174548, 0.8931268044151707, 0.9388506465986267]

Наиболее близкий вектор с Островким Артемом [85 18 12 57 95]