
22. NumPy & Pandas

(and embeddings)

9 януари 2025

Контролно на 16.01.2025 (Четвъртък)

Що е то data science, ML и AI?

- **Data Science** - Фокусът е извличане на ценна информация от наличните данни с цел вземане на информирани решения. (Финансови анализи, Класификация и таргетиране на клиенти, тенденции при клиенти)
- **Machine Learning** - Предоставя начин на самата машина да синтезира данни, да се учи от тях и да използва ключовата информация, за да се подобрява. (Рекомендър системи, търсачки, финансови модели)
- **Artificial Intelligence** - Фокуса е да разрешим на една машина да извършва комплексни интелектуални задачи, както би правил един човек (Чатботове, GO, Гласови асистенти, Co-pilot)

Примери

- <https://chat.openai.com/chat>
- <https://github.com/features/copilot>
- <https://replit.com/site/ghostwriter>
- <https://v0.dev/>
- <https://claude.ai/>
- <https://bolt.new/>
- <https://lovable.dev/>
- <https://notebooklm.google.com/> <- Брутално за учење
- <https://www.cursor.com/> -> VS Code Fork that is >>>> than copilot

Защо точно Питоня

- Изключително популярен с много голямо общество
- Разполага с множество интеграции
- Голямо разнообразие от библиотеки, които разрешават един ДС, МЛ, АИ проект да бъде завършен от начало до край
- Ключови библиотеки: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, PyTorch, Tensorflow, etc.

Pandas 🐼 & NumPy

- Вашите най-добри приятели, когато става въпрос за боравене с данни
- **Pandas** - не е кръстено на панди за съжаление, а на Panel Data
- **NumPy** - далеч по-скучно - Numerical Python

NumPy

- NumPy е основната библиотека, когато говорим за всякакъв тип сметки в Python
- Дава възможност за супер ефективна работа с масиви от всякакви измерения
- Масив - таблица от елементи от един и същ тип, като елементите са индексирани от tuples

NumPy array

- Като списък, ама с екстри
- Защо NumPy array вместо списък?
 - Всяка стойност в NumPy списъка има фиксиран тип
 - Използват contiguous memory -> непрекъснатата памет
 - Позволяват векторни операции
 - Cache locality
- Къде се използва NumPy?
 - Като MATLAB, но в Python.
 - Pandas
 - Изображения (Matplotlib)
 - ML

NumPy array операции

- `np.zeros()`, `np.ones()`, `np.identity()`, `np.random.randint()`
- `np.char.add()`, `np.char.multiply()`, `np.char.center()`
- `np.char.split()`, `np.char.splitlines()`, `np.char.strip()`
- `np.char.join()`, `np.char.replace()`
- `reshape()`, `transpose()`

NumPy array операции

- `reshape()`, `transpose()`
- `np.add()`, `np.subtract()`, `np.multiply()`, `np.divide()`
- `np.nditer`

Въпрос?

```
array([[1., 1., 1., 1., 1.],  
       [1., 0., 0., 0., 1.],  
       [1., 0., 9., 0., 1.],  
       [1., 0., 0., 0., 1.],  
       [1., 1., 1., 1., 1.]])
```

Pandas

- Основата му е NumPy
- Разрешава ни да извършваме всякакъв вид манипулации на данни, анализиране, почистване, като цяло до свеждане във вид за използване на някакъв тип модел или алгоритъм

Основни структури в Pandas

- **Pandas Series** - едноизмерен масив, като всеки обект вътре има име (label), обикновено това е индекс, но може да бъде променян. Този тип обект може да се състои от всякакъв тип данни (за разлика от numpy array)
- **Pandas Dataframe** - Двумерен масив (практически просто таблица)

Series

- Създаване - от списъци, от NumPy масиви, от речници
- Можем да променяме индекса
- Манипулации

DataFrame

- Dataframe - практически колкото искате Series обекта в един. Създаваме таблица от такива
- Начини за създаване
- Манипулации

DataFrame ключови операции

- `columns`, `index`
- `head()`, `tail()`
- `describe()`
- `iloc`, `loc`
- `groupby()`, `sort_values()`
- `fillna()`, `dropna()`, `isna()`

Видове файлови формати (основните)

- .xlsx, .csv
- Pickle
- Parquet
- Feather
- hdf/json
- msgpack

Малко за НЛП-то

- Клон в AI, който се фокусира в това да разреши на машините да отговарят на “човешкия” език по ценен начин.
- Защо това е важно?
- Примери: Преводачи, Виртуални асистенти, Сърч Енджини, Сентимент Анализ, Филтриране на имейли, Чатботове и тн.

Embeddings

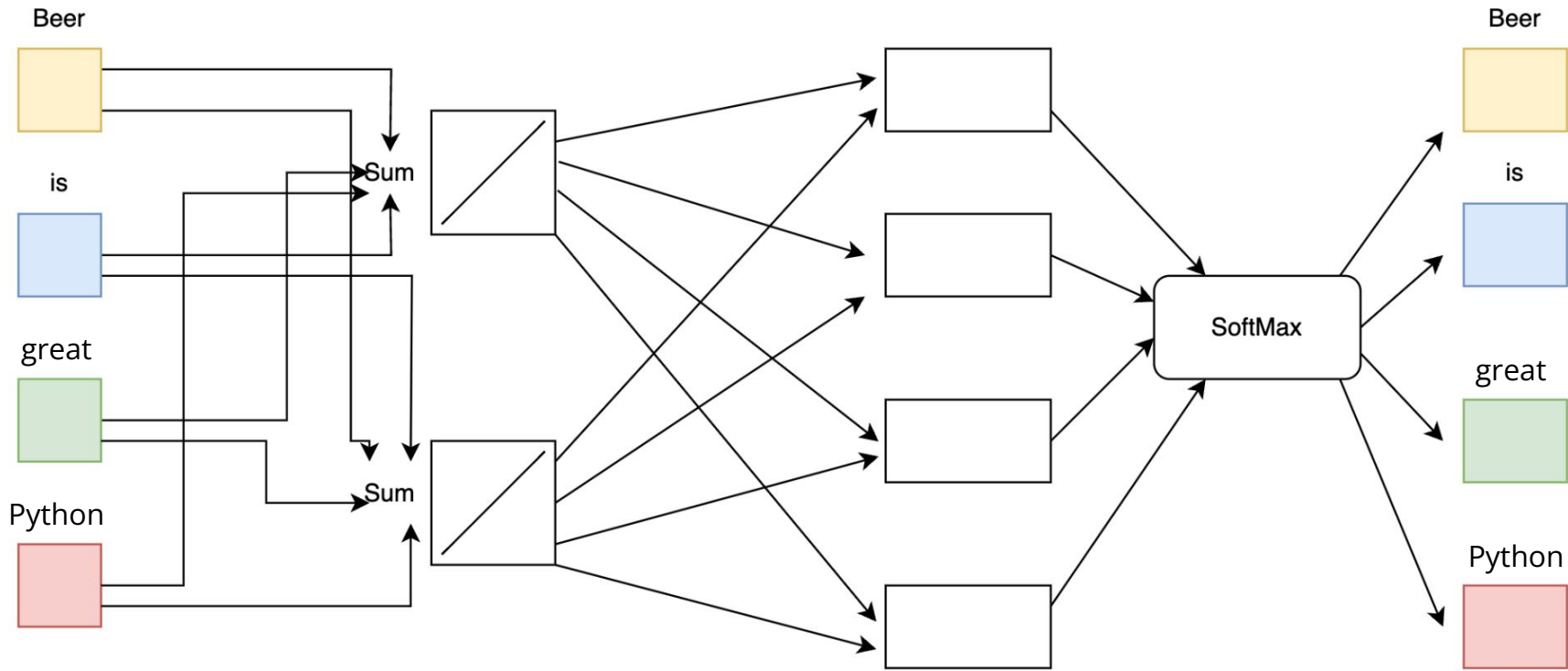
- Числено представяне на думи, където всяка дума е репрезентирана от вектор в непрекъснато векторно пространство. Обикновено се състоят от десетки или стотици измерения.
- Улавянето на семантична връзка между думите.
- Улавяне на контекстуална информация в изречения.
- Позволяват ни да намалим нужните пространства, за да работим с думи.
- Използват Невронни мрежи.

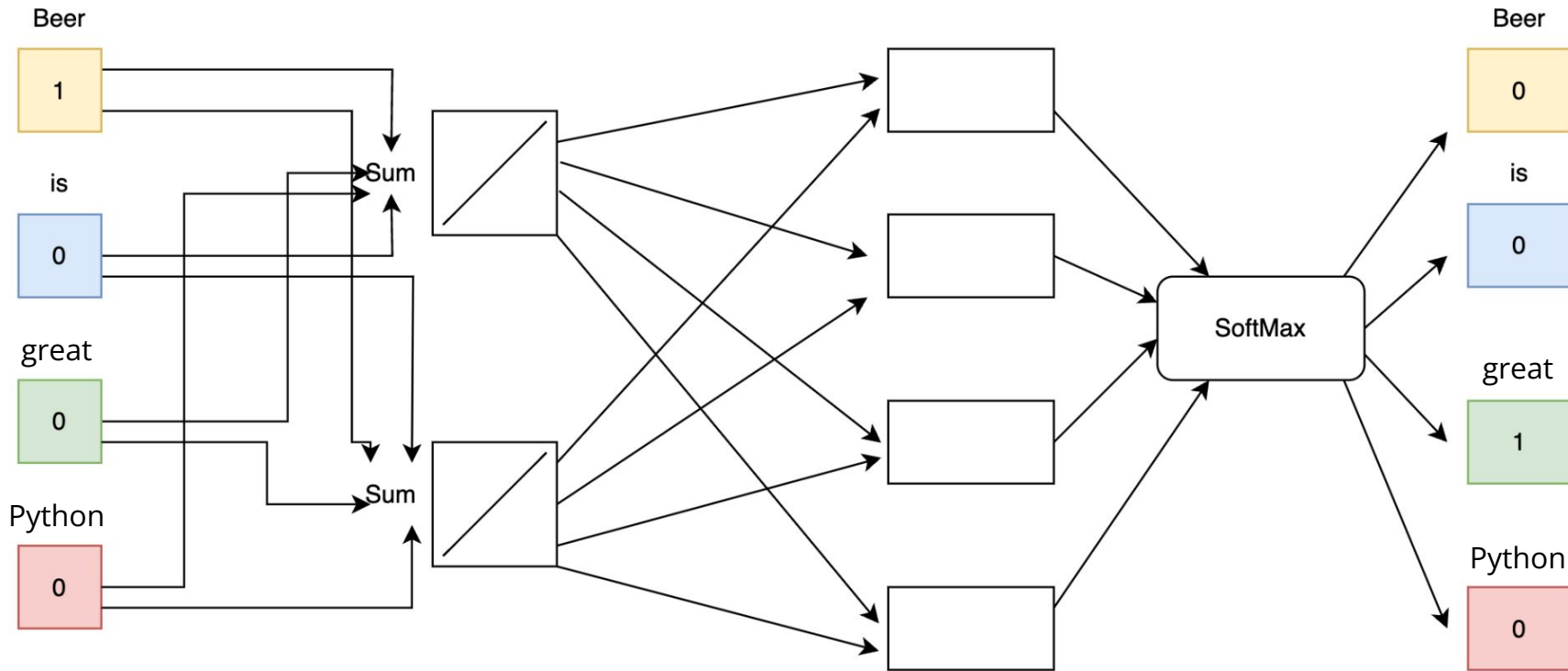
Word2Vec

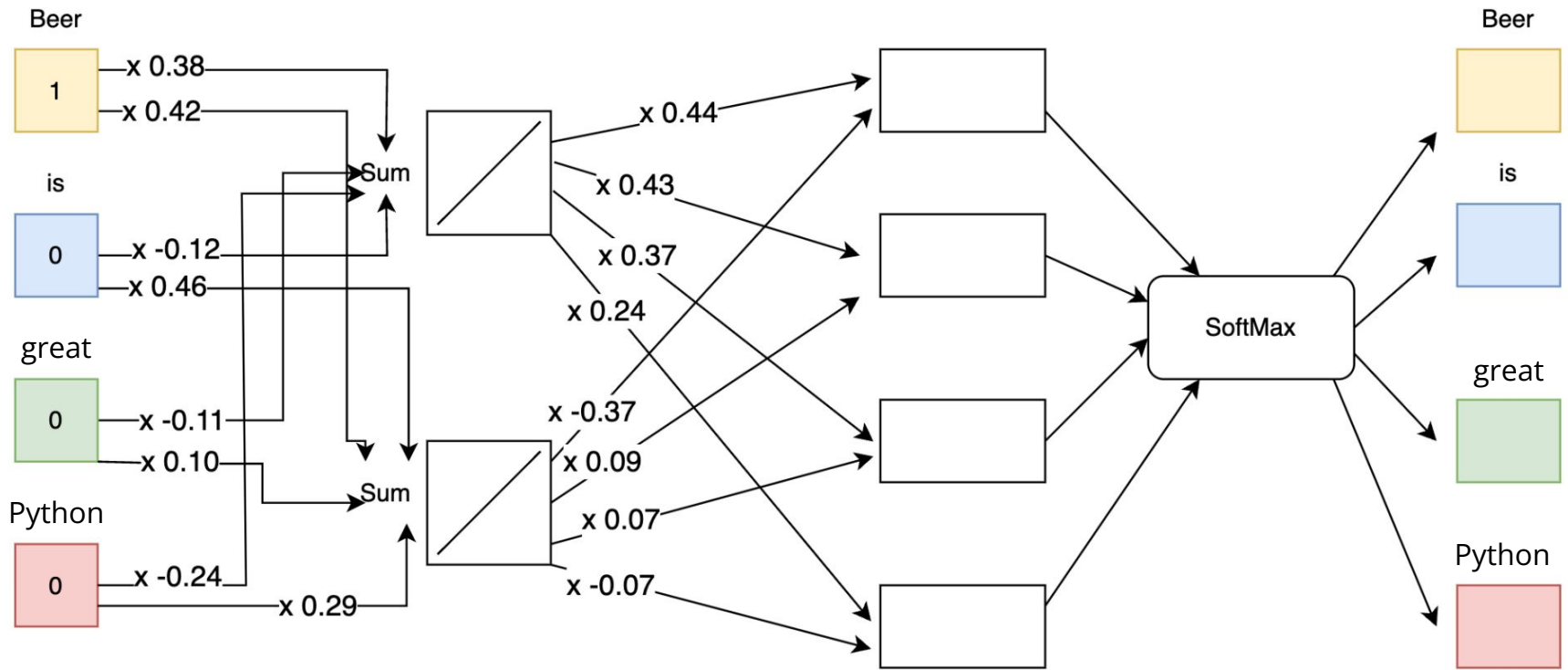
- Появява се през 2013- Естествено от Гугъл (Томас Миколов и неговия екип)
- Continuous bag of words:
 - Дадено ни е изречението: Пайтън е много __ език.
 - Искане да попълним празното място с подходяща дума, например- “як” или “готин” (може и “труден”).
- Skip-Gram:
 - Като CBOW, ама наобратно. Дадена ни е думата “бира” .
 - Искане да предскажем възможните думи наоколо - “крафт”, “лагер”, “пия”.
- TOKENS -> <https://tokens-lpj6s2duga-ew.a.run.app/>

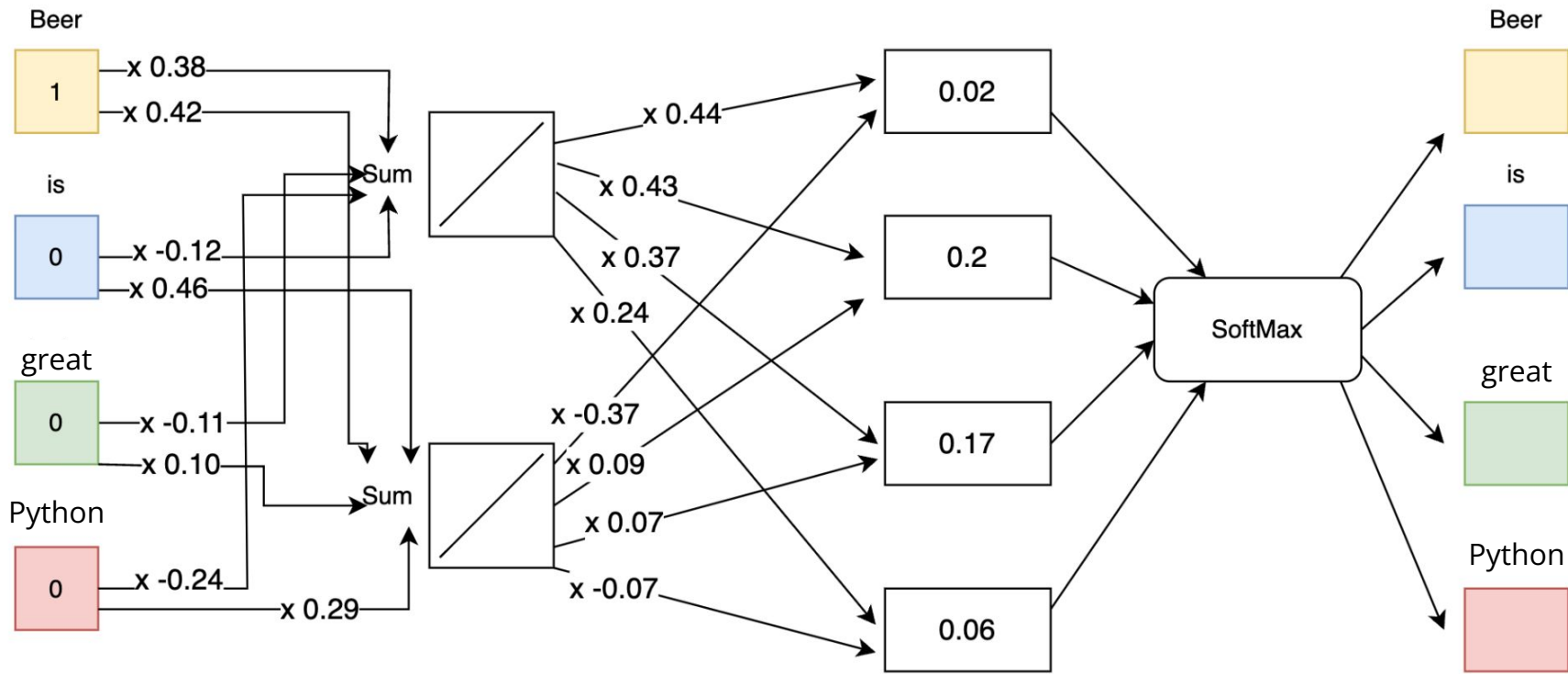
Нека направим един много прост Embedding

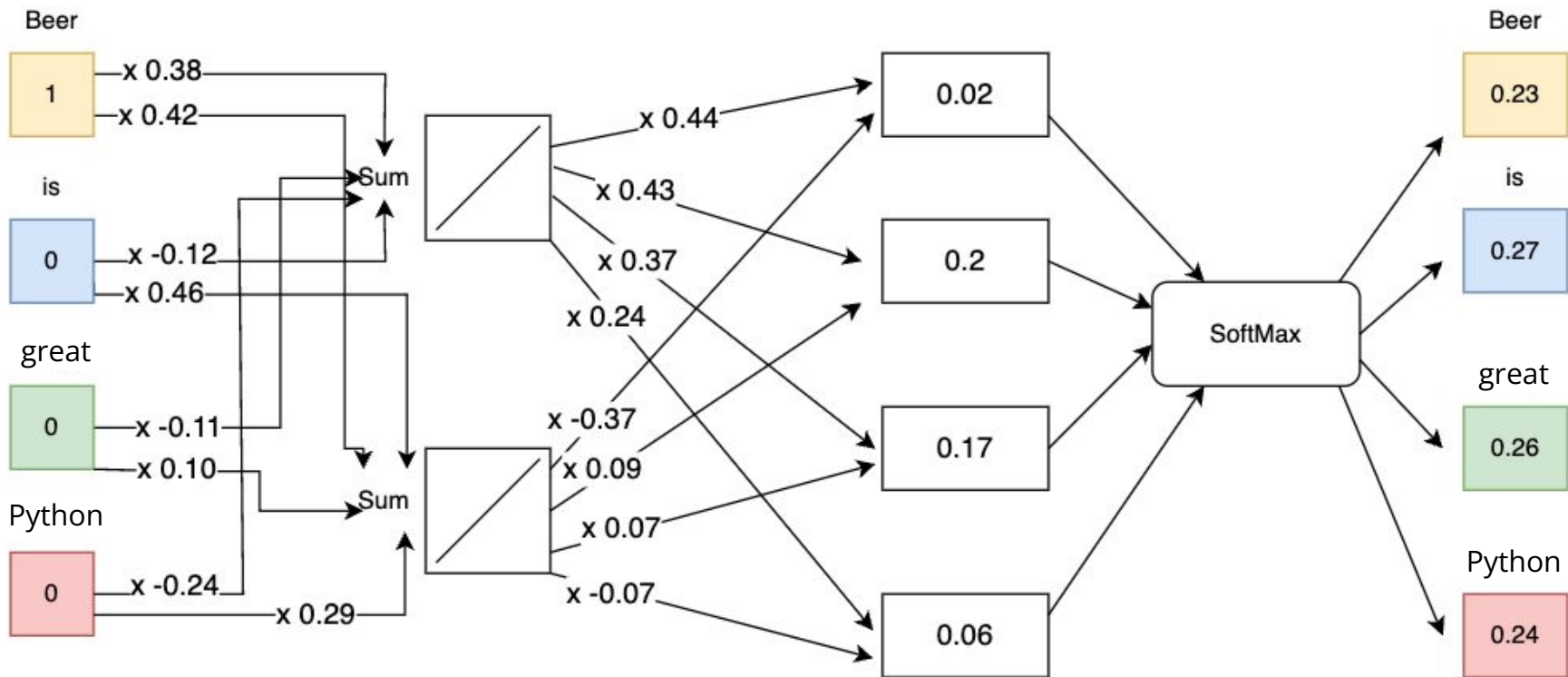
- Взимаме следните 2 изречения:
 - Beer is great
 - Python is great
- Tokenization (Много важна стъпка на обработване):
 - ["Beer", "is", "great"]
 - ["Python", "is", "great"]
- Нека сега направим един наш Embeddings Модел
- Извинявам се за следващите слайдове

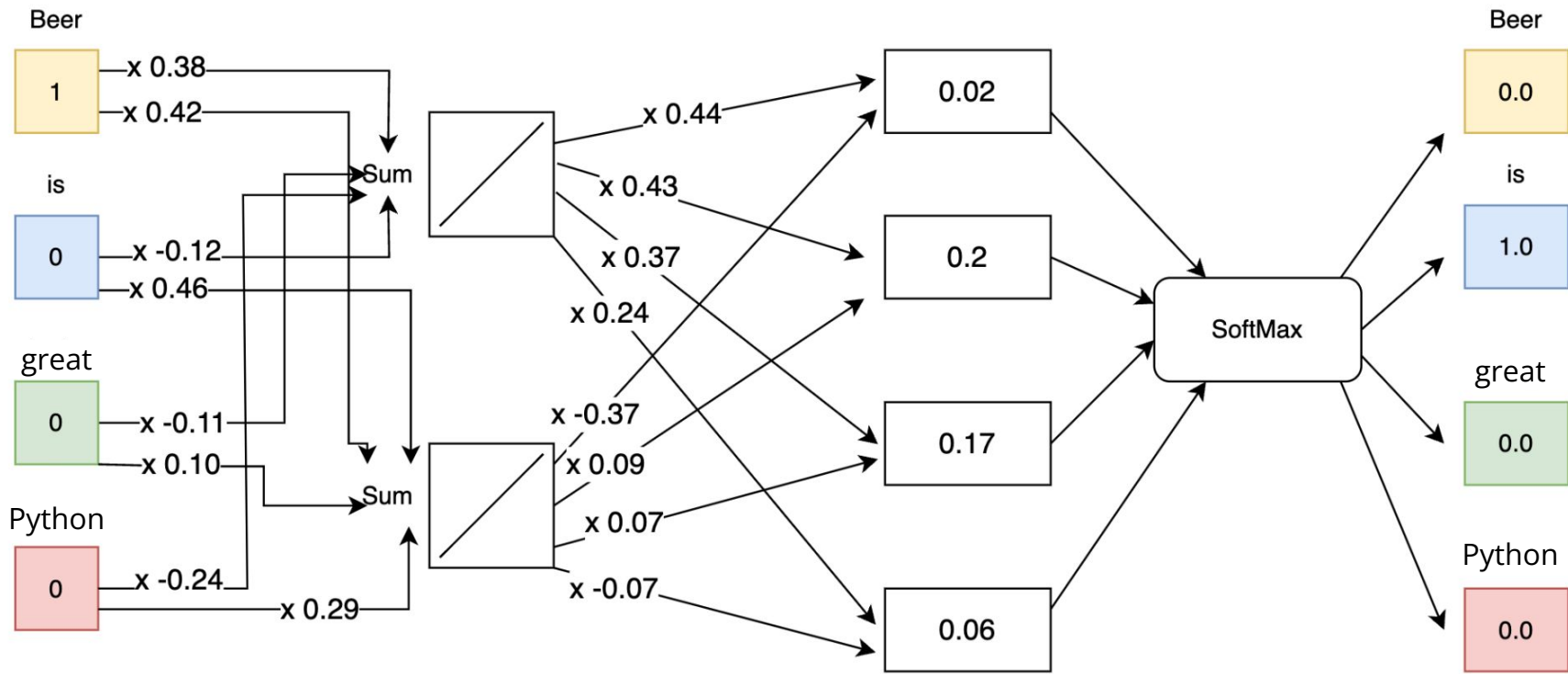












Накратко

Harvard
Business
Review



Harvard
Business
Review

DATA SCIENTIST

*The Sexiest Job of the 21st
Century*

WWW.FACEBOOK.COM/EMCACADEMICALLIANCE

EMC[®]

Още ресурси

- [Ноутбуци за NumPy & Pandas](#) - Може да си направите копие и да екзекютвате в DeepNote
- <https://www.kaggle.com/>
- <https://huggingface.co/>
- [Neural Network Playground](#)
- <https://bbycroft.net/llm>
- <https://tokens-lpj6s2duga-ew.a.run.app/>
- <https://drive.google.com/drive/folders/15Rt6yKom94WjaWIM0K7hBvK6hCLzBTyM?usp=sharing>
- Link to notebooks and data

Въпроси?