

# РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА НА ОСНОВЕ УГЛУБЛЕННОГО НЕЙРОСЕТЕВОГО АНАЛИЗА

**Е. М. Марченко**

*Белорусский государственный университет, пр. Независимости, 4,  
220030, г. Минск, Беларусь, evgeniyamarchenko12@gmail.com  
Научный руководитель — В. А. Нифагин, кандидат физико-математических  
наук, доцент*

В работе проведен нейросетевой анализ данных, позволяющий статистически релевантно проанализировать отобранную информацию, составить рейтинговую таблицу, с помощью которой нейросеть генерирует алгоритм выбора информации для того или иного пользователя.

Нейросеть обучается принципам обработки информации на основе данных результатов опроса. Для этого формируется рейтинговая таблица, в которой выставлены экспертные оценки профессиональных и личностных качеств актеров по шкале от 1 до 5. Рекомендательная система должна предоставлять нанимателю (режиссеру, продюсеру) вероятностные оценки актеров, которые должны оптимальным образом соответствовать реализации проекта.

**Ключевые слова:** углубленный нейросетевой анализ; рекомендательная система; опрос; оценка профессиональных и личностных качеств.

Нейросети обычно используются для обработки данных большого объема и определения закономерностей между ними [1]. В данной работе нейросеть обучается принципам обработки информации, исходя из результатов опроса, для создания рекомендательной системы.

Работает нейросеть на основе имитации нейронных связей человеческого мозга. На базе полученных результатов, которые будут сгенерированы нейросетью, в дальнейшем будет разработан мультимедийный продукт.

Цель работы – теоретически обосновать и разработать рекомендательную систему при использовании нейросетевого анализа и в конечном итоге, создать UX-UI продукт в помощь режиссерам при выборе актеров для киносъемки.

Объект исследования – рейтинговая система для подбора актеров по заданным характеристикам в помощь режиссерам. Предмет исследования – процесс формирования рейтинговой системы и разработки UX-UI продукта в помощь режиссерам.

Стандартной методикой получения первичного массива данных является проведение опроса.

На основе уточняющих вопросов, формируется блок профессиональных и личностных качеств (skills), ниже представлены некоторые из них:

Skill1: наличие общего среднего образования  
Skill2: наличие высшего образования (бакалавриат)  
Skill...

Skill30: готов экспериментировать с видами съемки и условиями для съемки.

Информацию об оценках профессиональных и личностных качеств удобно представлять в виде таблицы (матрицы), где строки соответствуют актерам(музыкантам), а столбцы – их профессиональным и личностным качествам. На пересечении строки и столбца будет оценка экспертом соответствующего качества по пятибальной шкале.

Матрица выше имеет стандартную двумерную форму, где значение  $M$  (матрицы) не совпадает с  $N$  (количество строк и столбцов не совпадает). Ее называют user-item матрицей или UI матрицей. Именно с данными в такой форме нам предстоит работать.

Рассмотрим используемый рекомендательный алгоритм.

Существуют различные методы построения рекомендательных систем, когда по данным о клиентах или пользователях необходимо произвести подборку рекомендаций с целью оптимизации продажи товаров клиентам или предоставления услуг пользователям [2].

Для этого применяется традиционный кейс рекомендаций работодателю (режиссеру) по подбору актеров.

Рекомендательная система по данной информации должна предоставлять режиссеру (продюсеру) рекомендации подбора актеров, которые с высокой вероятностью должны оптимальным способом способствовать реализации проекта.

Сингулярное разложение матриц (SVD-разложение) – это представление прямоугольной матрицы в виде произведения нескольких матриц особого вида [3]. Цель этого разложения – это упростить некоторые вычисления, которые будут осуществляться над матрицей.

Мы представляем матрицу UI в виде:

$$UI_{N \times M} = U_{N \times N} \Sigma_{N \times M} V^T_{M \times M} \quad (1)$$

где матрица  $U$  будет представлять актеров, а матрица  $V$  – их качества, (при этом матрица  $V^T$  — это сопряженно-транспонированная матрица к  $V$ ),  $\Sigma$  – матрица размеров  $N \times M$ , на главной диагонали которой лежат неотрицательные числа (называемые сингулярными числами матрицы, откуда произошло название разложения).

Таким образом, у нас есть частично заполненная UI матрица оценками качеств актеров.

Наша задача заполнить все остальные пустые места UI матрицы.

### Дозаполнение профессиональных и личностных качеств актеров

Actor	Skill							
	Skill 1	Skill 2	Skill 3	Skill 4	Skill 5	Skill 6	Skill 7	Skill 8
Actor 1	5	4	5				md	md
Actor 2	5	md	5				md	md
Actor 3		3	5				4	md
Actor 4								
Actor 5	md	md	4				4	md
Actor 6								
Actor 7								
Actor 8								
Actor 9								
Actor 10	3	md	md				md	5

Для этого по частично заполненной UI матрице приближенно рассчитывается SVD разложение UI матрицы и далее находятся оценки исполнителей, перемножая матрицы полученного разложения. Для поиска приближенного SVD разложения необходимо минимизировать следующий функционал:

$$\text{Mat}_{N \times M} = U_{N \times N} \Sigma_{N \times M} V_{M \times M}^T * \min_{u_i, v_j} \Sigma_{i,j} (mat_{i,j} - u_i * v_j) \quad (2)$$

где  $u_i$ ,  $v_j$  строки и столбцы матриц  $U_{N \times N}$  и  $V_{M \times M}^T$  соответственно,  $mat_{i,j}$  – известные элементы  $UI_{N \times M}$  матрицы. Минимизация производится методом градиентного спуска [4].

```
python
import pandas as pd
import numpy as np
from sklearn.decomposition import NMF

# Загрузка данных из Excel
data = pd.read_excel('путь_к_вашему_файлу.xlsx', index_col=0)

# Заполнение пропущенных значений нулями
data.fillna(0, inplace=True)

# Преобразование DataFrame в матрицу numpy
X = data.to_numpy()

# Выполнение NMF
model = NMF(n_components=3, init='random', random_state=0)
W = model.fit_transform(X)
H = model.components_

# Создание DataFrame для рекомендац
recommended_actors = pd.DataFrame(H.T, index=data.columns, columns=[f'Feature (i
```

Алгоритм

Для выполнения матричного разложения данных, можно воспользоваться библиотеками Python, такими как pandas для загрузки данных и numpy для выполнения разложения [5].

Этот код загружает данные из вашего файла Excel, выполняет NMF алгоритм с 10 компонентами, и выводит оценки в виде списка актеров для каждой личностной и профессиональной характеристики с учетом их весовых коэффициентов. Вы можете изменить число компонентов и другие параметры NMF в соответствии с вашими потребностями.

Рекомендации для продюсера:

Feature 1:

Skill 1: 1.1510786050387942

Skill 26: 0.8482426535161857

Skill 25: 0.6959488793630834

Skill 28: 0.6791547396683382

Skill 30: 0.6351085541924353

Skill 27: 0.24099670331576542

Skill 22: 0.0

Skill 24: 0.0

Skill 29: 0.0

Feature 2:

Skill 29: 1.300444708210643

Skill 24: 1.2027290169203164

Skill 30: 0.32595839888143086

Skill 27: 0.29662887723543063

Skill 28: 0.2235272615405145

Skill 25: 0.21352545372596218

Skill 1: 0.0

Skill 22: 0.0

Skill 26: 0.0

Feature 10:

Skill 27: 2.8186365668042757

Skill 24: 2.528039692519172

Skill 1: 1.466844320753077

Skill 28: 1.0724495746459743

Skill 26: 1.065518177826533

Skill 25: 0.9724929026175573

Skill 22: 0.0

Skill 29: 0.0

### Выводы.

Стоит отметить, что в ходе первичной обработки данных была составлена рейтинговая таблица. Обработав информацию данного опроса, нейросеть генерировала алгоритм подбора актеров, на основе которого были составлены оптимальные характеристики актеров, что сократит время и затраты режиссеров и кастинг-директоров при выборе исполнителей, основанном на данных результатов анализа.

В статье описана реализация метода матричного разложения– NMF, так же приведен пример программной разработки его алгоритма. Оценка по метрике NMF демонстрирует высокую точность данного подхода, а внедрение в работу нейросетей сокращает время работы, что позволяет обработать большое количество информации, выявляет определенные закономерности и отфильтровывает информацию. Проверка найденных оценок на выборках малых и средних объемов выявила адекватность полученных результатов.

### Библиографические ссылки

1. *Козлов В. Н.* Системный анализ, оптимизация и принятие решений: учебное пособие // Санкт- Петербург: Издательство Политехнического университета, 2011.
2. *Круглов В.В., Борисов В.В.* Искусственные нейронные сети. Теория и практика. Москва, 2002 г., 382 с.
3. *Sultonov, A., Matveev, S.A., Budzinskiy, S.* Low-rank nonnegative tensor approximation via alternating projections and sketching, Computational and Applied Mathematics, 42(2), 68, 2023
4. *J. Hopcroft, R. Kannan* Foundations of Data Science: учебное пособие // Cambridge University Press, 2020.
5. *Пасхавер Б.* «Pandas в действии» // Б. Пасхавер. - Издательский дом «Питер»Python «Профессиональная литература», 2023 г., 645 с.