АНАЛИТИЧЕСКИЙ ОТЧЕТ ПО ВАКАНСИЯМ BI-АНАЛИТИК

Аналитическая записка к модулю В

Чемпионат высоких технологий по компетенции «Специалист по анализу данных (BI-аналитик)»

|  |  |
| --- | --- |
|  | Студент гр. 21П-1  Д.И. Гизатуллин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

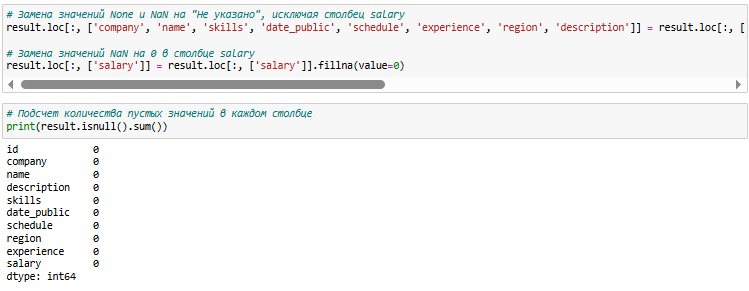
2024

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| 1 Постановка задачи | 3 |
| 2 Получение данных о вакансиях | 4 |
| 2.1 Парсинг данных | 4 |
| 2.1.1 Получение информации со страниц | 4 |
| 2.1.2 Получение вакансий из файлов страниц | 5 |
| 2.1.3 Удаление файлов, не прошедших проверку | 5 |
| 2.1.4 Формирование датафрейма | 6 |
| 2.2 Объединение датафрейма с первой сессии | 8 |
| 2.2.1 Объединение двух датафреймов в один | 8 |
| 2.2.2 Обработка пустых значений | 9 |
| 3. Работа с текстом | 9 |
| 3.1 Семантический анализ текста | 9 |
| 3.1.1 Предварительная обработка данных | 9 |
| 3.1.2 Облако слов | 12 |
| 3.1.3 Выявление эмоциональной окраски фраз | 17 |
| 3.1.4 Векторизация текстовых данных | 18 |
| 3.2 Тематическое моделирование | 21 |
| 3.3 Кластеризация | 25 |
| 3.4 Классификация | 27 |
| 3.5 Таблица с классами навыком, частотой и содержанием | 28 |
| 3.6 Проведение анализа классов навыков | 29 |
| 3.7 Файл в формате xlsx | 35 |
| Итоги | 35 |

|  |  |
| --- | --- |
| 1. Постановка задачи   Время на выполнения модуля: 4 часа  Краткое описание задания:  Необходимо выполнить парсинг данных для сбора информации с указанных веб-ресурсов. Сформировать структуру набора данных. Провести предварительную обработку данных. Выполнить построение и отбор признаков. Выполнить семантический анализ текста. Провести визуальный анализ статистики текстовых признаков. Сделать выводы, в том числе с использованием инструментов визуализации.  Полное описание задания:  Необходимо из описаний вакансий вычленить ключевые навыки. Навыки - конкретные умения и требования, которые предъявляет работодатель к соискателю.  С помощью алгоритмов анализа естественного языка необходимо провести предварительную обработку данных, провести кластеризацию и классификацию выделенных навыков в близкие по смыслу категории.  Провести анализ полученных классов навыков: определить наиболее востребованные и наименее востребованные навыки для выбранной группы вакансий; определить наиболее высокооплачиваемые навыки; определить региональную специфику востребованных навыков.  Результат модуля:  1. Таблица в формате xlsx с классом навыков, его содержанием и частотой встречаемости.  2. Аналитическая записка в формате docx с основными выводами  2. Получение данных о вакансиях  2.1 Парсинг данных  2.1.1 Получение информации со страниц  Возьмем существующий датасет с нашими вакансиями и спарсим дополнительные вакансии для нашего датасета.  Получим страницы в виде json файлов через API hh.ru. Можете ознакомиться с документацией API HH на их сайте.      Страницы собраны, теперь можно из этих страниц найти вакансии, которые соджержутся в json формате  2.1.2 Получение вакансий из файлов страниц    Вакансии собраны, теперь найдём ключевые поля для нашего датасета, предварительно удалив файлы, которые запрашивают captcha.  2.1.3 Удаление файлов, не прошедших проверку  Обычно такие файлы весят меньше 2 КБ, поэтому будем удалять именно их. (Можно убедиться что файлы у которых вес меньше 2 КБ нет нужных нам параметров таких как описание, название и т.д., вы можете открыть файл в блокноте и убедиться в этом сами). Кстати, таких файлов довольно много, вообще это можно обойти используя несколько прокси, но для этого нужны выделенные апи разных стран, так что спарсим сколько сможем, думаю из 5к вакансии останется где то 1-2к.    Теперь проверим сколько у нас получилось валидных вакансий после удаления лишних.    В итоге у нас получилось 1202 валидных вакансий.  2.1.4 Формирование датафрейма  Теперь из наших файлов можно сформировать датасет с нужной информацией по вакансиям. В нашем случае это будут следующие признаки:   * Идентификатор * Название * Название компании * Описание * Опыт работы * График работы * Дата публикации * Регион * Заработная плата     Сформируем датафрейм из наших получившихся списков.    Теперь нужно удалить HTML теги из описания, сделаем это через библиотеку BeautifulSoup4. |  |
| 2.2 Объединение датафрейма с первой сессии  2.2.1 Объединение двух датафреймов в один  Загрузим датафрейм из первой сессии. После чего объеденим его с полученным датафреймом при помощи парсинга и удалим пустые значения из объединённого датафрейма. |  |

2.2.2 Обработка пустых значений



Как можно заметить, пустых значений в датафрейме нет, теперь можно поработать с категориальными (текстовыми) признаками в датафрейме.

3. Работа с текстом

3.1 Семантический анализ текста

3.1.1 Предварительная обработка данных

Семантический анализ текста — это процесс, который помогает понять смысл текста с учётом контекста, логической структуры предложений и грамматических ролей.

1. Перевод всех букв в тексте в нижний регистр

2. Удаление знаков препинания

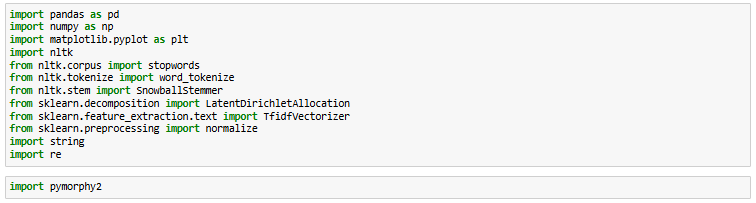
3. Токенизация по словам

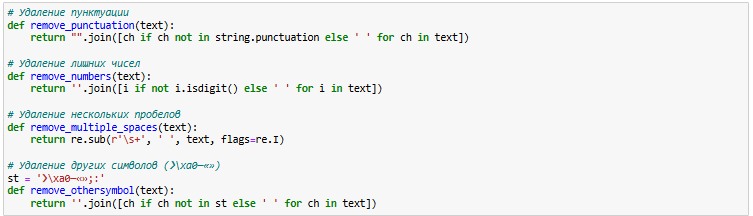
4. Удаление стоп слов

5. Стемминг (приведение слов к нормальной форме)

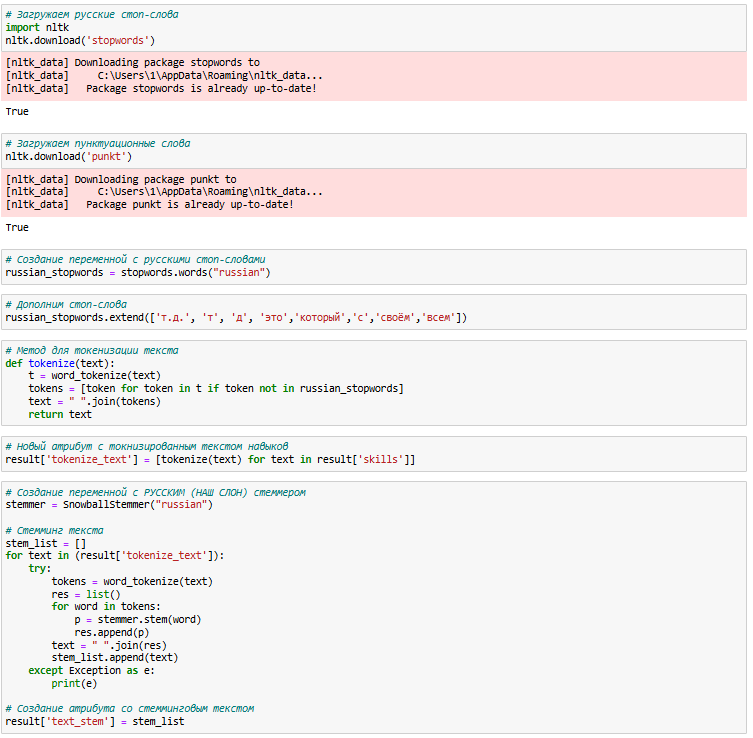
6. Лемматизация

Создадим методы для форматирования текстов (удаление пунктуации, лишних чисел, несколько пробелов подряд, удаление ненужных символов).





Теперь удалим стоп-слова и пунктуацию из нашего текста. И используем стемминг.



Теперь в датафрейме есть поле со стемминговым текстом, он поможет сделать леммитизацию текста для машинного вида текста.



Предварительная обработка текста прошла успешно. Теперь можно работать с леммитизированным текстом.

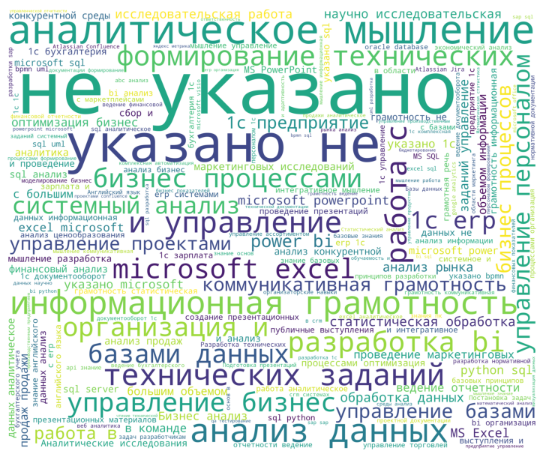
3.1.2 Облако слов

О́блако те́гов (облако слов, или взвешенный список, представленное(-ый) визуально) — это визуальное представление списка категорий (или тегов, также называемых метками, ярлыками, ключевыми словами и т. п.). Обычно используется для описания тегов (метаданных) на веб-сайтах или для представления неформатированного текста. Ключевые слова чаще всего представляют собой отдельные слова, и важность каждого ключевого слова обозначается размером шрифта или цветом. Такое представление удобно для быстрого восприятия наиболее известных терминов и для распределения терминов по популярности относительно друг друга. При использовании ключевых слов для упрощения навигации по веб-сайтам, термины снабжаются гиперссылками на понятия, которые подразумеваются под ключевыми словами. Тем не менее, облако не может быть единственным элементом навигации, а пользоваться текстовым поиском удобнее для поиска при наличии более конкретной информации об искомом.



Для начала посмотрим полный список слов навыков во всём датафрейме без предварительной обработки.





Можно заметить, что самое большое слово - Не указано. Это из-за того, что многие работодатели не указывали требуемые навыки. Также одно из больших слов это аналитическое мышление и анализ данных, что довольно логично, т.к. у нас вакансии - аналитик, и это действительно одни из самых важных и ключевых навыков в этой профессии. Остальное рассмотрим в предобработанном виде.

Теперь посмотрим облако слов на обработанном тексте.





Тут уже список ключевых навыков расширился. Можно заметить, что больших слов стало больше, например системный анализ, обработка данных, управление проектов, информационная грамотность, технические задания и т.д. Для большинства BI-аналитиков это действительно важные и ключевые навыки.

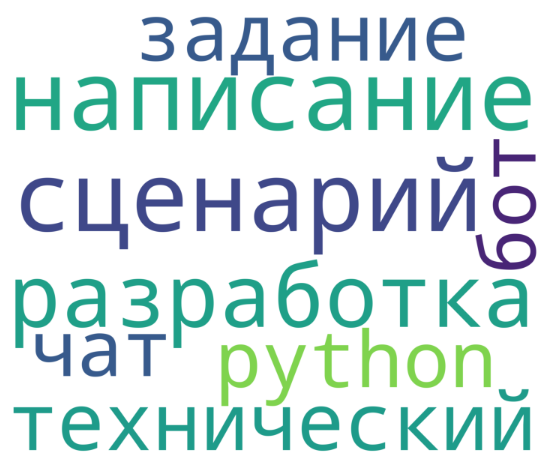
Теперь посмотрим отдельно леммитизацию (в виде облака слов) по идексам.





Это облако формирует некое представление того, что будет требоваться в данной вакансии. Можно предположить, что скорее всего эта вакансия как то связана с с работой с документами и какой-то информацией на пк в коллективе.

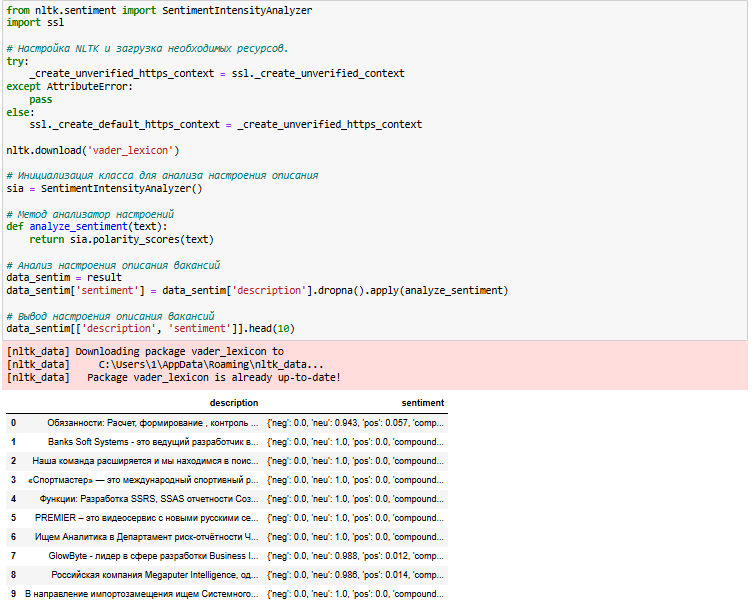
Давайте посмотрим следующий индекс.



Тут уже совсем иной случай. Тут уже заметно что профессия связана непосредственно с разработкой чат бота на python.

3.1.3 Выявление эмоциональной окраски фраз

Можно так же посмотреть на эмоциональную окраску описания вакансий, посмотрим, как чаще всего работодатели любят строить предложения (с каким эмоциональным тоном).



В итоге мы получили признак sentiment (настроение), которое показывает четыре основные эмоциональные окраски:

* neg - негативный эмоциональный окрас
* neu - нейтральный эмоциональый окрас
* pos - позитивный эмоциональный окрас
* compound - смешанный (сложносоставляющий) эмоциональый окрас

Чем ближе к 1.0 - тем больше встречается тот или иной эмоциональный окрас.

3.1.4 Векторизация текстовых данных

Векторизация текста или преобразование текста в численную форму. Алгоритмы машинного обучения не умеют работать с текстом, поэтому необходимо превратить текст в цифры.

Прямое кодирование (one-hot encoding) считается самым простым способом преобразования токенов в тензоры и выполняется следующим образом:

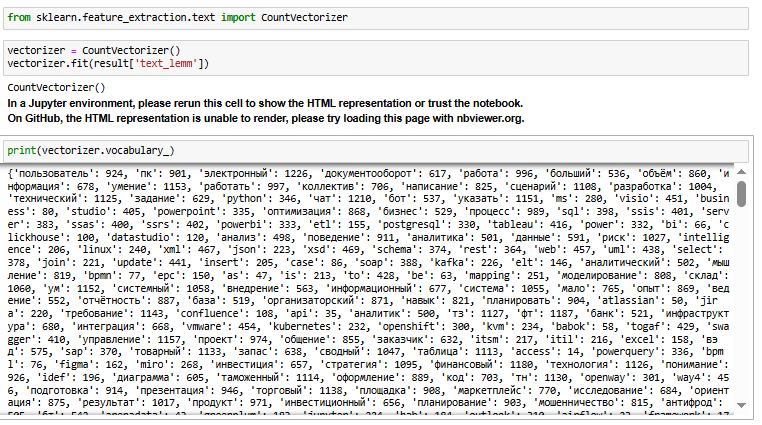
1. каждый токен представляет бинарный вектор (значения 0 или 1);
2. единица ставится тому элементу, который соответствует номеру токена в словаре.

Проблемой прямого кодирования является размерность.

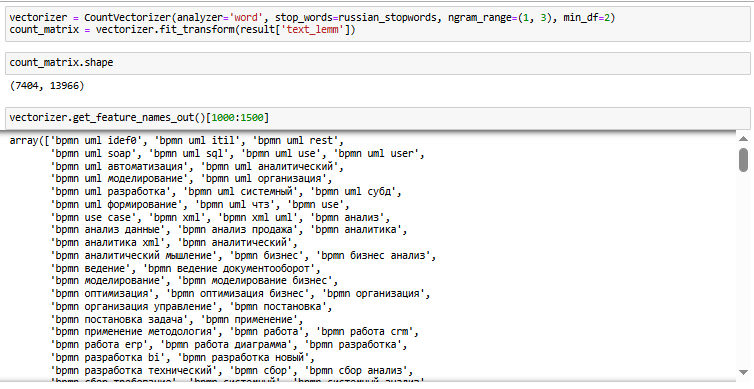
В отличие от прямого кодирования, мешок слов (Bag of words) выделяет вектору весь документ, и каждый элемент кодируется 1 по порядку следования слов в словаре.

Bag of words решает проблему размерности по одной оси. Количество строк определяется количеством документов. Однако, этот метод не учитывает важность того или иного токена, ведь одно слово может повторятся по несколько раз.

Мешок слов - решает проблему размерности по одной оси. Количество строк определяется количеством документов. Однако, этот метод не учитывает важность того или иного токена, ведь одно слово может повторятся по несколько раз. В пакете scikit-learn есть модуль CountVectorizer, который преобразовывает входной текст в матрицу, значениями которой являются количества вхождения данного ключа(слова) в текст. Таким образом, мы получим матрицу, размерность которой будет равна количеству всех слов, умноженных на количество документов. И элементами матрицы будут числа, которые означают, сколько раз всего слово встретилось в тексте.



Можно увидеть, что получился словарь с самим словом и кол-вом этого слова.

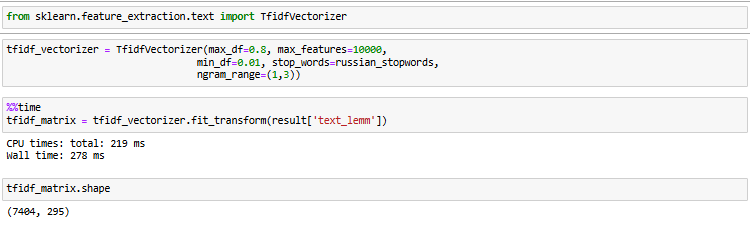


Получили массив значимых (feature) слов.

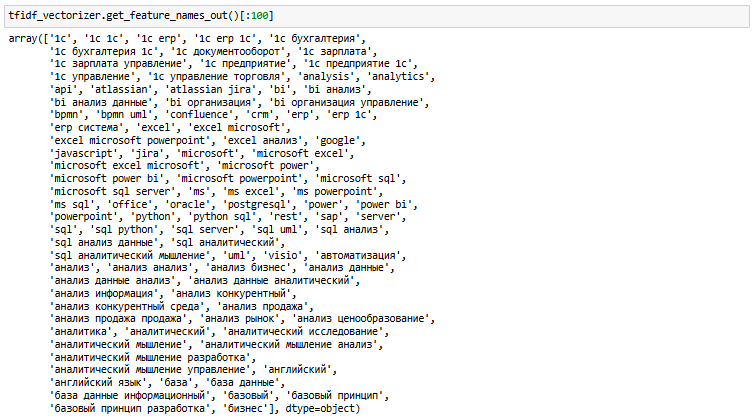
TF-IDF (от англ. TF — term frequency, IDF — inverse document frequency) — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно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

TF-IDF состоит из двух компонентов: Term Frequency (частотность слова в документе) и Inverse Document Frequency (инверсия частоты документа). Они считаются следующим образом:

TF-ID — это способ векторизации текста, отражающий важность слова в документе, а не только частоту его появления. В тексте большого объема некоторые слова могут присутствовать очень часто, но при этом не нести никакой значимой информации о фактическом содержании текста (документа). Если такие данные передавать непосредственно классификатору, то такие частые термины могут затенять частоты более редких, но при этом более интересных терминов. Для того, чтобы этого избежать, достаточно разделить количество употреблений каждого слова в документе на общее количество слов в документе, это есть TF — частота термина. Термин IDF (inverse document frequency) обозначает обратную частоту термина (инверсия частоты), с которой некоторое слово встречается в документах. IDF позволяет измерить непосредственную важность термина.



В итоге мы получили матрицу TF-IDF, посмотрим какие тут получились значисые слова.



По этой матрице что-то сказать сложно, но в будущем она нам пригодится для нахождения частоты (важности) навыков.

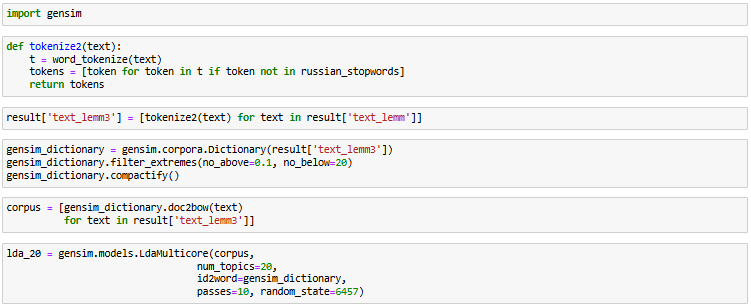
3.2 Тематическое моделирование

Тематическое моделирование позволяет выделять из текстов темы, связанные с определенными множествами слов, и затем смотреть, с какой вероятностью тексты соотносятся с этими темами.

Библиотека gensim. Это популярная открытая библиотека для тематического моделирования, в которой есть нужная модель — LDA (Латентном размещении Дирихле (LDA)). Затем следует создать словарь для тематического моделирования из лемматизированного текста.

После создания словаря лучше всего отфильтровать те слова, которые встречаются в слишком большом количестве текстов, и те, которые встречаются слишком редко. Для этого есть метод filter\_extremes, который принимает в себя аргументы no\_above (только слова, которые встречаются не более чем в указанной доле текстов) и no\_below (слова, которые встречаются не менее чем в указанном количестве текстов).

После удаления лишних слов словарь лучше всего ужать в размерах, убрав пропуски с помощью метода compactify.

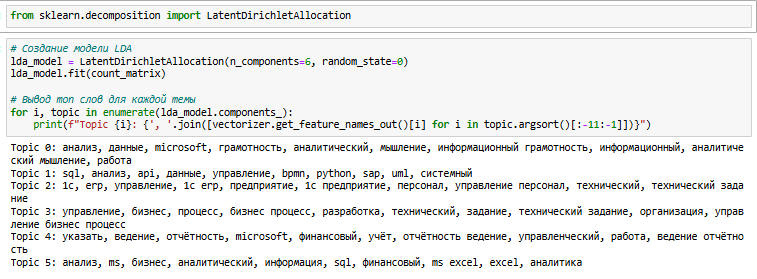


Посмотрим какие топики у нас получились.



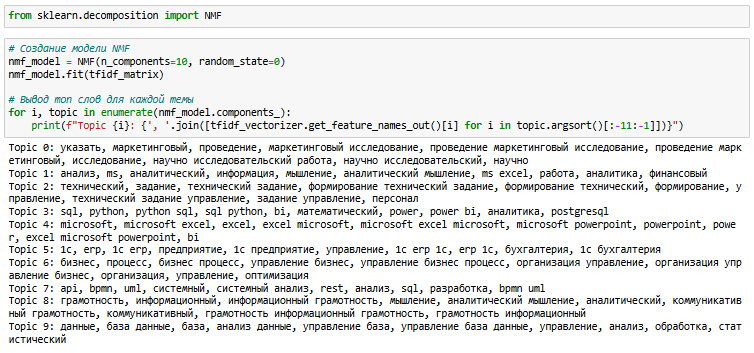
Получилось 20 топиков, в котором обозначено каждое входящее слово в этот топик и его коэф. влияния.

LDA — это метод машинного обучения, который используется для анализа тематической структуры в больших наборах документов. Который использует модель вероятности, чтобы определить, какие темы присутствуют в каждом документе, и какие слова связаны с каждой темой. LDA может быть полезен в случаях, когда нужно опредлить тематическую структуру в большом наборе документов, таких как статьи и т.д.



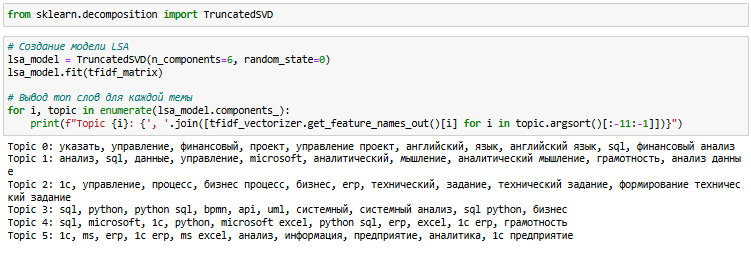
В итоге у нас получилось 6 топиков (мы указали) и модель распределила слова так. В принципе можно представить за что отвечает каждый топик (какая профессия и т.д. но лучше посмотреть более подробные топики дальше).

NMF — это метод машинного обучения, который используется для разложения матрицы на две с неотрицательными элементами. Он может быть полезен в случаях, когда нужно найти скрытые признаки в данных, таких как тексты и т.д. NMF может использоваться в различных областях, таких как обработка изображений, текстовый анализ и т.д.



Тут уже заметно более точное распределение слов. Можно заметить, что слова группируются по смыслу и по английским словам. Например, седьмой топик явно указывает на работу с microsoft продукцией.

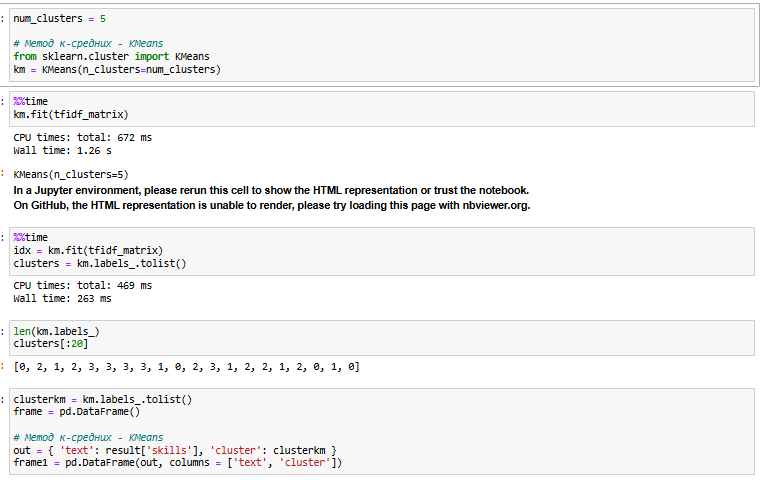
LSA — это метод машинного обучения, который используется для анализа структуры в больших наборах текстовых данных. Он использует матричные вычисления для определения близости между словами и документами. LSA может быть полезен в случаях, когда вы хотите определить структуру в большом наборе текстовых данных, таких как статьи, отчеты и т.д.



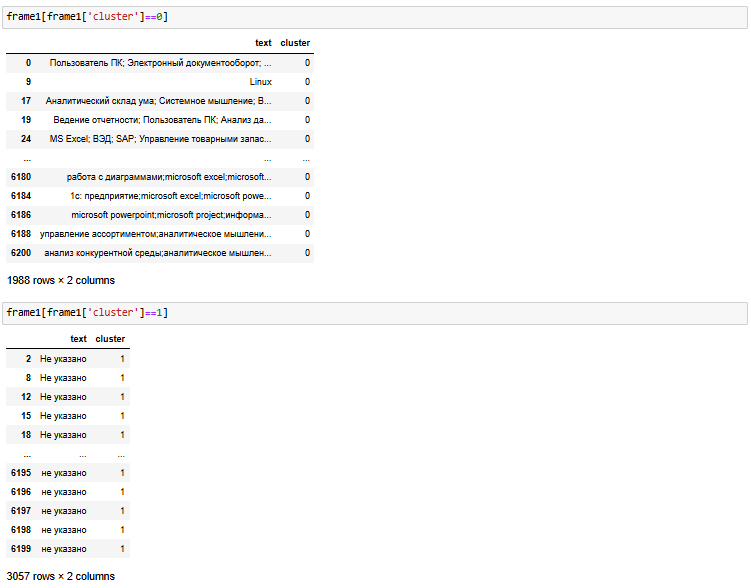
LSA модель тоже неплохо справилась, но для большей точности лучше полагаться на модель NMF.

3.3 Кластеризация

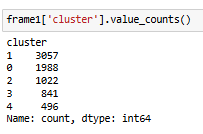
Возьмем кол-во кластеров 5. Когда мы рассматривали топики, мы нашли несколько схожих по тематике группы навыков. (языки, 1с, питон, microsoft, не указано) и т.д.



Кластеризация прошла успешно, теперь посмотрим на наши кластеры: изучим входящие данные по каждому кластеру







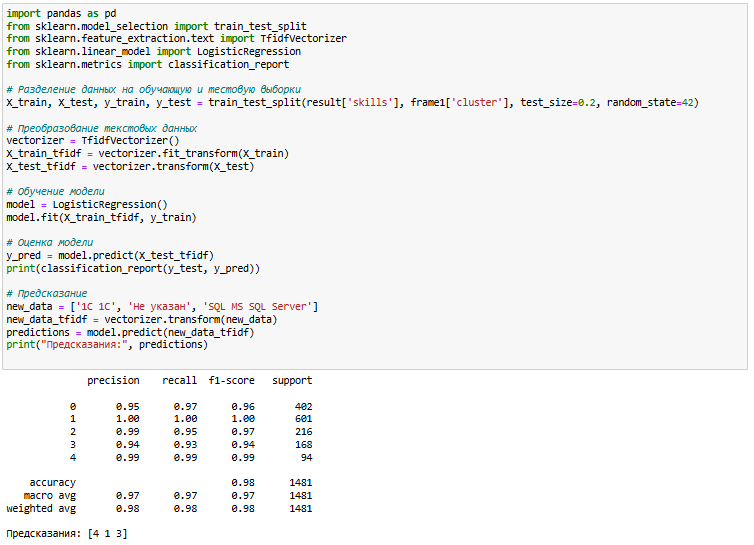
В итоге у нас получилось пять групп навыков:

* Навыки, которые тесно связаны с разработкой и управлением различных систем (кластер 0)
* Навыки, которые не указали работодатели (кластер 1)
* Навыки, которые связаны с работой с базами данных SQL (кластер 2)
* Навыки, которые связаны с 1C разработкой (кластер 3)
* Навыки, которые тесно связаны с bpmn и организацией и управлением биз (кластер 4)

Так же можно поиграться с кол-вом кластеров и посмотреть, как модель распределит навыки, но пока что будет достаточно и этого.

3.4 Классификация

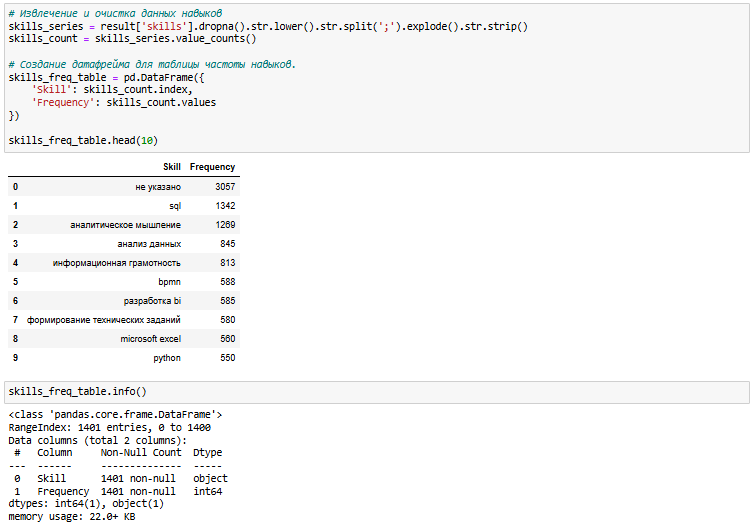
Теперь на основе наших кластеров можно сделать классификацию, для опеределения класса навыков.



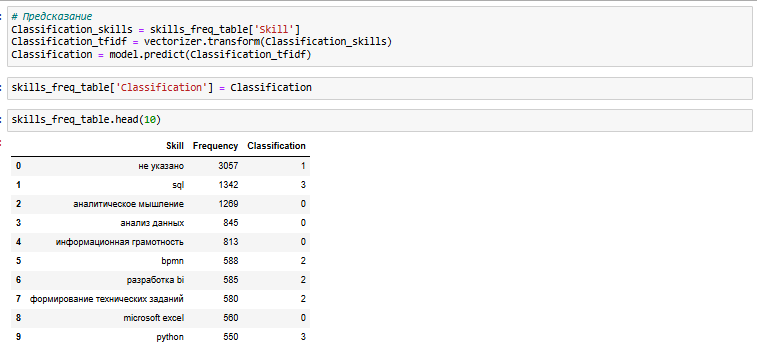
Модель отработала хорошо. Показатели точности очень высокие. На новом датафрейме отработала хорошо, всё предсказала верно: третий кластер связан с 1С, первый кластер не указан, второй кластер связан с разработкой и ЯП. Модель всё предсказала верно.

3.5 Таблица с классами навыком, частотой и содержанием

Теперь благодаря модели на основе классификации, можно построить требуемую таблицу с классами навыков, содержанием и частотой встречаемости.



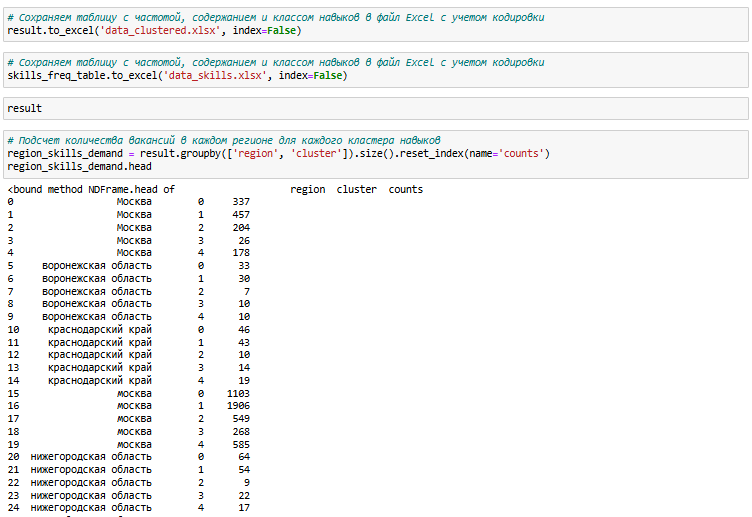
Всего получилось 1401 уникальных навыков, теперь определим для каждого из них свой класс навыков (зададим им кластер через классификацию)

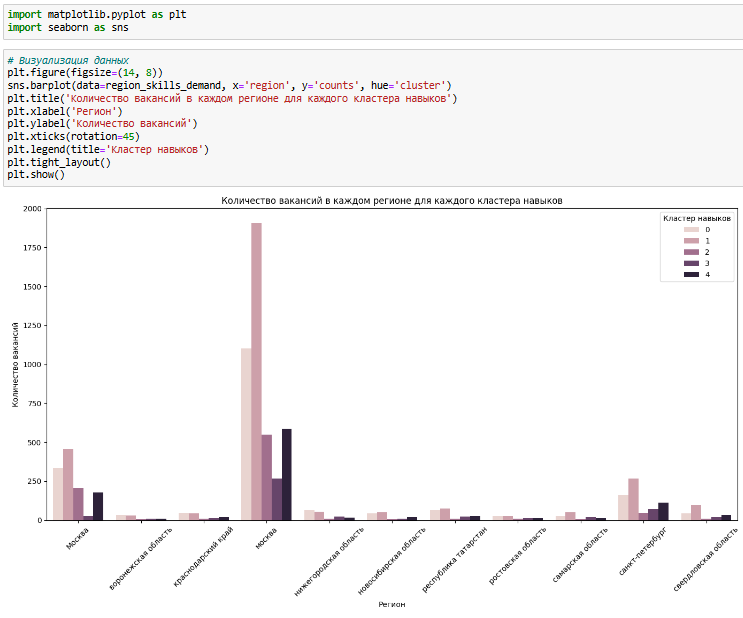


В итоге получилась таблица с классами навыков, содержанием и частотой встречаемости навыков. В дальнейшем её можно будет использовать для построения графов (сетевой анализ).

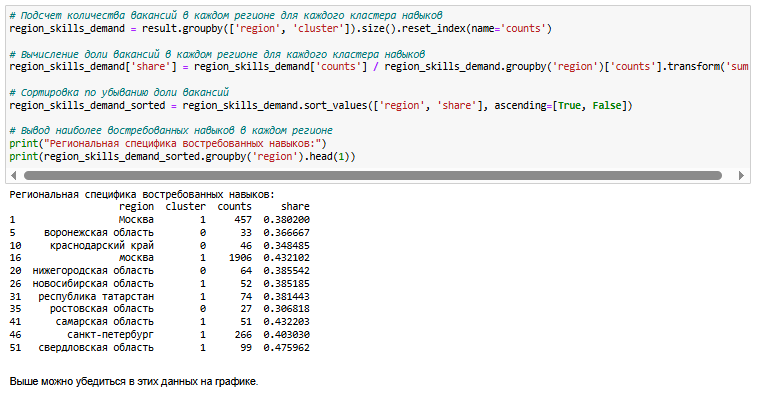
3.6 Проведение анализа классов навыков

Проанализируем полученные классы навыков для более широкого представления о них.

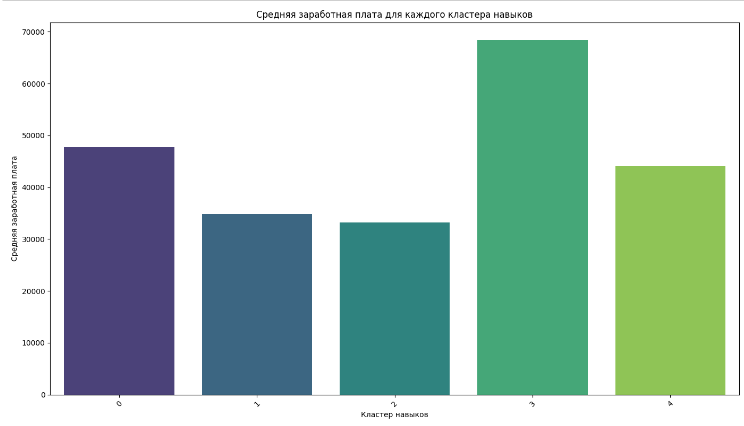




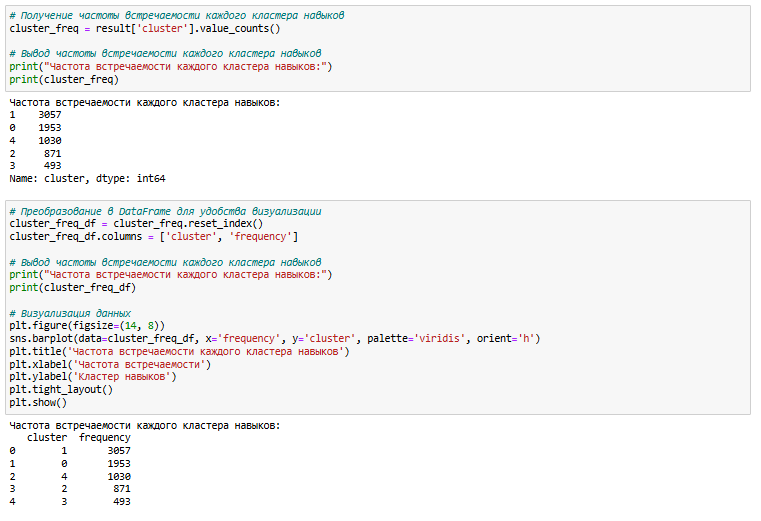
На данном графике отображаются регионы (по оси X) и количество навыков (по оси Y). Как можно заметить – в Москве больше всего вакансий, значит и количество навыков там будет больше всего. В остальных регионах приобладают навыки группы 0 и 1 (Навыки, которые тесно связаны с разработкой и управлением различных систем и навыки, которые не указали работодатели). Можно сделать вывод что навыки связанные с разработкой и управлением различных систем очень востребованы в различных регионах Росиии.

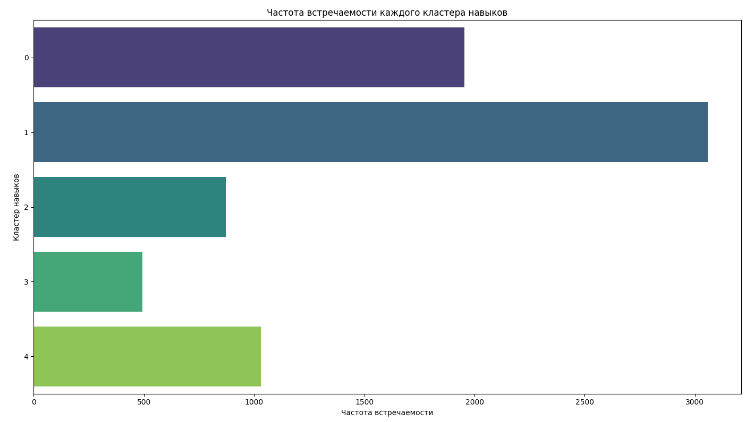




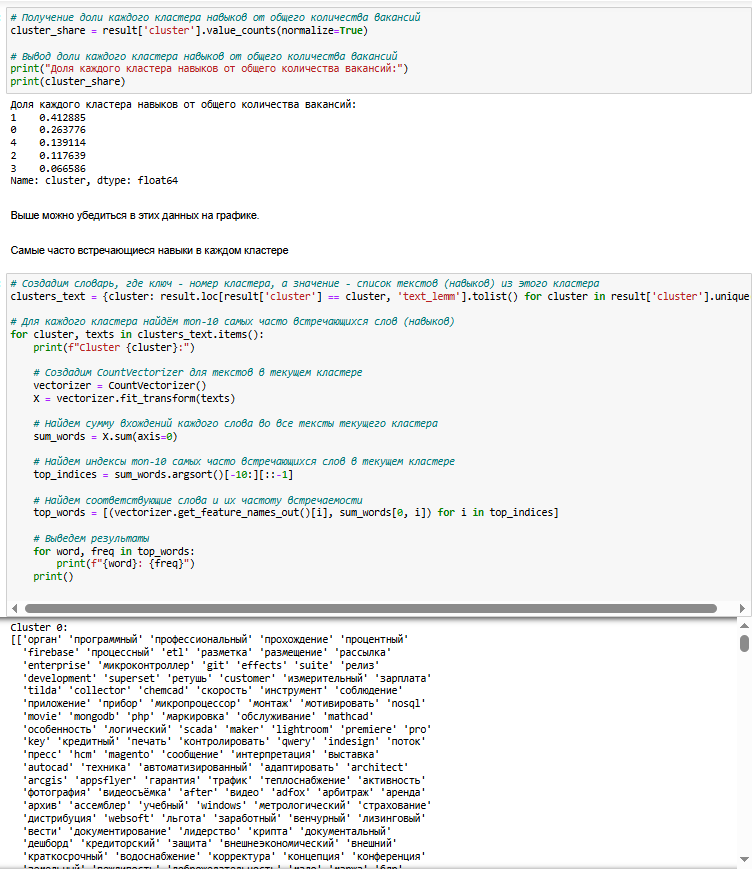


На данном графике отображаются классы навыков (по оси X) и количество средней заработной платы (по оси Y). Как можно заметить – класс навыков 3-ей группы имеет более высокую средную заработную плату, а класс навыков 2-ой группы имеет наименьшую среднюю заработную плату. Можно сделать вывод что классы навыков 3-ей группы (Навыки, которые связаны с 1C разработкой) более высокооплачиваемые чем остальные группы.





На данном графике отображаются классы навыков (по оси Y) и количество встречаемых навыков (по оси X). Как можно заметить – класс навыков 1-ей группы имеет самую высокую частоту встречаемости, но если не учитывать 1-ую группу (т.к. это не указанные навыки), то навыки 0-ой группы (Навыки, которые тесно связаны с разработкой и управлением различных систем) встречаются чаще всего. Навыки 3-ей группы (Навыки, которые связаны с 1C разработкой) встречаются реже всего.



Почему-то выводятся все навыки, но главное, что они отсортированы и можно посмотреть часто встречающиеся навыки.

Топ навыков по встречаемости в каждом кластере:

* 0 кластер - орган, программный, процессный, прохождение, системный, приложение, разметка, рассылка, etl, микроконтроллер.
* 1 кластер - указать, не указано
* 2 кластер - dynamics, внимательность, операционный, взаимодействие, devops
* 3 кластер - язык, 1c, внимательность, английский, docs
* 4 кластер - процессный, финанс, качество, клиентоориентированность, контроль, docker

3.7 Файл в формате xlsx

Теперь сохраним таблицу с частотой, содержанием и классом навыков в файл Excel для работы с графами в следующем модуле.



Итоги

В результате было спарсено ещё почти 2к вакансий и добавили их к существующим. Смогли сделать семантический анализ текста. Построили облако слов. Обучили несколько моделей для векторизации. Вывели и посмотрели на топики. Сделали систему для поиска ключевых навыков по топикам. Сделали кластеризацию. Сделали классификацию. Определили группы навыков.

В итоге получилось пять групп навыков:

* Навыки, которые тесно связаны с разработкой и управлением различных систем (кластер 0)
* Навыки, которые не указали работодатели (кластер 1)
* Навыки, которые связаны с работой с базами данных SQL (кластер 2)
* Навыки, которые связаны с 1C разработкой (кластер 3)
* Навыки, которые тесно связаны с bpmn и организацией и управлением биз (кластер 4)