УДК 004.89

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ DATA CLEANING ДЛЯ ОЧИСТКИ БОЛЬШОГО ОБЪЁМА ДАННЫХ

Акилина М. В., студентка группы ИТб-162, IV курс
Пылов П. А., студент группы ИТб-162, IV курс
Протодьяконов А.В., к.т.н., доцент
Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

Коронавирус 2019 года (2019-нКоВ) – это вирус (специализированное именование – коронавирус), идентифицированный как причина вспышки респираторного заболевания, впервые обнаруженной в Ухани[1]. В настоящее время неясно, насколько легко или устойчиво этот вирус распространяется между людьми[2]. Особенно важно в настоящее время получать «чистые» данные, в виду того, что только такие данные составляют фундамент для медицинских исследований, проектирования моделей искусственного интеллекта для решения разных связанных задач, построения диаграмм и графиков распространения коронавируса.

В рассматриваемой статье представлен набор данных, содержащий ежедневную информацию о количестве зарегистрированных случаев, смертности и восстановлении от коронавируса[3]. Следует обратить внимание, что данные представляют ряд, зависящий от времени, поэтому число регистраций в любой день является накопительным числом.

Целью работы является создание алгоритма очистки данных, для правильного формирования типов данных столбцов, удаления аномалий из наборов, если в наборе присутствуют «выбросы», и сохранения очищенных данных в новый файл, для дальнейшего применения данных в исследованиях.

Платформой реализации алгоритма выступает язык программирования высокого уровня Python, так как он очень удобен для аналитики большого объёма данных и позволяет быстро выполнять обработку информации.

Для работы с данными их необходимо загрузить в проект решения, оптимальный способ сделать это – использовать метод чтения файла для преобразования данных в табличный вид (рисунок 1)

---

[1] Ухань, Китай
[2] Center for Disease Control and Prevention https://www.cdc.gov/
[3] Информация получена из открытых источников Всемирной Организации Здравоохранения https://www.who.int/

```
dataframeCoronaVirus = pd.read_csv('2019_nCoV_data.csv')
dataframeCoronaVirus.head()
```

Рисунок 1. Загрузка данных для создания алгоритм очистки.

Проверка данных, с которыми предстоит работать, является начальным шагом очищения данных. Изучим состав столбцов представлением первых пяти строк датасета (рисунок 2)

	Серийный номер регистрации	Дата события	Провинция/Штат/Город	Страна	Последнее обновление	Подтверждённое заражение	Летальный исход	Выздоровевшие
0	1	01/22/2020 12:00:00	Anhui	China	01/22/2020 12:00:00	1.0	0.0	0.0
1	2	01/22/2020 12:00:00	Beijing	China	01/22/2020 12:00:00	14.0	0.0	0.0
2	3	01/22/2020 12:00:00	Chongqing	China	01/22/2020 12:00:00	6.0	0.0	0.0
3	4	01/22/2020 12:00:00	Fujian	China	01/22/2020 12:00:00	1.0	0.0	0.0
4	5	01/22/2020 12:00:00	Gansu	China	01/22/2020 12:00:00	0.0	0.0	0.0

Рисунок 2. Первые пять строк датасета исследуемых данных.

Столбцы:

- Серийный номер регистрации – это код, который записывается как нумератор для каждого нового случая регистрации. По нему отслеживается конкретная регистрация болезни;
- Дата события – представляет дату и время наблюдения события в журнале отчёта Всемирной Организации Здравоохранения;
- Провинция/Штат/Город – это обозначение конкретного города или округа, принадлежащего территориально к городу, в котором произошла регистрация события;
- Страна – страна, в которой было зарегистрировано событие;
- Последнее обновление - обновление по всемирному времени координирования Земли – UTC;
- Подтвержденное заражение - это случаи, которые были подтверждены клиническими тестами;
- Летальный исход - это представление числа погибших от вируса людей;
- Выздоровевшие - это число полностью излечившихся от заражения людей, не учитывающих частичное облегчение симптомов.

Только после полного изучения всех данных можно приступать к этапу очистки набора. Стоит отметить, что все индексируемые датами столбцы должны относиться к соответствующим типам данных, точно также как и все числовые столбцы иметь отношение к свойственному для чисел типу данных.

Это необходимо для последующего полноценного оперирования над данными (например, выполнить вычитание дат, умножение чисел, сложение процентных отношений). В большинстве случаев, все данные изначально будут

представлены в виде типа данных «object». Набор данных несложно проверить особым методом определения типов данных «dtype» (рисунок 3)

```
dataframeCoronaVirus['Последнее обновление'].dtype
```

```
dtype('O')
```

Рисунок 3. Идентификация типов данных.

Типы данных «object» встречаются у нескольких столбцов. Следует скорректировать типы данных всех столбцов до правильных типов (рисунок 4)

```
dataframeCoronaVirus['Дата события'] = pd.to_datetime(dataframeCoronaVirus['Дата события'])
dataframeCoronaVirus['Последнее обновление'] = pd.to_datetime(dataset['Последнее обновление'])
dataframeCoronaVirus['День'] = dataframeCoronaVirus['Дата события'].dt.day
dataframeCoronaVirus['Месяц'] = dataframeCoronaVirus['Дата события'].dt.month
dataframeCoronaVirus['Неделя'] = dataframeCoronaVirus['Дата события'].dt.week
dataframeCoronaVirus['День недели'] = dataframeCoronaVirus['Дата события'].dt.weekday
```

Рисунок 4. Изменение типов данных столбцов в датафрейме.

Ежедневный отчет показывает, что число случаев увеличивается день ото дня, поэтому расчет количества случаев сегодняшнего дня приравнивается к числу случаев на сегодня с прибавлением числа за предыдущие дни включительно. Таким образом, возьмем наиболее актуальную дату в качестве обновленного количества случаев (рисунок 5)

```
fresh_data = dataframeCoronaVirus['Дата события'].iloc[-1]
last_updated = dataframeCoronaVirus[dataframeCoronaVirus['Дата события'].dt.date == fresh_data]
```

Рисунок 5. Привязка актуальной даты.

Важнейшим этапом является проверка на пропущенные значения (рисунок 6) – такие значения могут существенно повлиять на работу алгоритмов искусственного интеллекта и негативно сказаться на исследованиях в медицине.

```
# проверка пропущенных значений
dataframeCoronaVirus.isnull().sum()
```

```
Серийный номер регистрации      0
Дата события                    0
Провинция/Штат/Город            0
Страна                          0
Последнее обновление            0
Подтверждённое заражение        0
Летальный исход                 0
Выздоровевшие                   0
День                            0
Месяц                           0
Неделя                          0
День недели                     0
dtype: int64
```

Рисунок 6. Пропущенные значения и их детекция.

Проверка всех значений показала отличный результат – нет ни одного отсутствующего значения.

Теперь следует объединить данные, представляющие равнозначные данные – например, материковый Китай и Китай – это одна и та же страна и территориальное местоположение, однако следует проверить, что все данные находятся под одним названием и не содержат в себе повторений.

Объединение таких значений выполняется объединением данных с положительным аргументом замены повторов (рисунок 7)

```
dataframeCoronaVirus['Страна'] = dataframeCoronaVirus['Страна'].replace({'Mainland China':'China'},inplace=True)
updated_dataframe['Страна'].replace({'Mainland China':'China'},inplace=True)
```

Рисунок 7. Объединение данных материкового Китая и Китая.

Итогом всех действий будет сохранение очищенного файла с данными в новый файл «CoronaVirusCleanData.csv» (рисунок 8)

```
dataframeCoronaVirus = pd.to_csv('CoronaVirusCleanData.csv', encoding = 'utf-8')
```

Рисунок 8. Сохранение файла с данными.

Выполненные действия впоследствии во многом помогут сохранить ресурсы многим медикам и исследователям, так как очищенные данные позволяют сразу перейти экспертам к выполнению профессиональных действий над данными, позволяя экономить драгоценное время на исследования, в которых особенно остро нуждается современное мировое сообщество.

Список литературы:

1. Christopher Bishop. Pattern Recognition and Machine Learning, Plenum press, New York – London, 1971
2. A. Geron. Hands-on Machine learning with Scikit-Learn, Keras, and Tensor-Flow, OREILLY Sebastopol, California – USA, 2019
3. P. Bruce, A. Bruce. Practical statistics for data scientists, OREILLY Sebasto-pol, California – USA, 2017
4. В.В. Мещеряков. Моделирование и визуализация случайных данных на языке Python, Интерактивное учебное пособие, Москва, 2015