Chu and Chong, J Health Inform Manag 2020, 4:2
DOI: 10.37532/jhmm.2020.4(2).115

Journal of Health Informatics & Management

Research Article

A SCITECHNOL JOURNAL

# Big Data Analytics Using Python-Based Machine Learning: A Challenging Opportunity in Hospital Pharmacy

**Wilson W S Chu and Gary C H Chong***

*Pharmacy Department, United Christian Hospital, Hospital Authority, Hong Kong, China*

***Corresponding author:** Wilson W S Chu, Pharmacy Department, United Christian Hospital, Hospital Authority, Hong Kong, China, Tel: +852 3949 4181; E-mail: ww.chu@yahoo.com

**Received Date:** June 03, 2020; **Accepted Date:** June 19, 2020; **Published Date:** June 26, 2020

## Abstract

**Background:** In this study, we introduce the use of Python 3 as an effective tool to hold and analyse Big Data in Healthcare. The aim of this study is to explore how big data may be utilized to support a better hospital pharmacy service, provide potential benefits, and describe the potential future use. Furthermore, population description and pattern are to be explored.

**Methods:** We open Jupyter Notebook and then choose Python 3 to run the code. We used the Hospital Pharmacy Management System (PMS) Query Template System to extract data from 01 April 2019 to 31 March 2020. We used Python 3 to hold and analyse all the .CSV dataset.

**Results:** The total number of dispensing records retrieved was 1,765,910. 127,337 patients paid 358,802 visits to our pharmacy with 381,060 prescriptions handled. For the 127,337 individual patients, 57% fall into 60-100 while 21% are from 80-100. 43% of the prescriptions had 1 or 2 items. 48,000 patients visited our medical clinic, 47,000 patients visited the family clinic, and 36,000 patients visited the psychiatric clinic during the data collection period.

**Conclusions:** In this study, we demonstrate how data analytics could assist in decision making in healthcare. The analytics helps time-strapped pharmacist to monitor patient data easily in real-time and thus enable them to invest more time in better patient care. The huge volume of raw data we obtain during this research is organised to create insights into the behaviour of our patient populations, prescription type, prescribing pattern, services, and workflows.

**Keywords:** Pharmacists; Big data; Machine learning

## Introduction

The term "Big Data" refers to large amounts of structured and unstructured data that require rapid and often multifaceted computational analyses to tell patterns, trends, and associations, relating to various aspects of human and non-human entities [1,2]. The goal is to provide patterns for better strategic planning and resource exploitation [3]. The secret to using the vast big data in healthcare effectively is to find out how to look at the volume, the real-time need for and the variety of data, and use it to improve both the quality and to plan future services [4]. Pharmacy informatics, on the other hand, is defined as the use and integration of data, information, knowledge, technology, and automation in the medication-use process for the purpose of improving health outcomes [5].

Python is a very clear and powerful programming language [6]. It runs on Windows, Mac OSX, Unix, Linux, Solaris, and it even runs on your mobile phone [7]. Python is free to download and use, and it has a very large online user [8]. Throughout this project, we used the Anaconda Python distribution, which is a free and open-source distribution of the Python programming languages for scientific computing [9]. While it is not necessary to use this distribution of Python, it is highly recommended.

In this study, we introduce the use of Python 3 as an effective tool to hold and analyse Big Data in Healthcare. We use Medical Subject Headings (MeSH) for keywords and terms indexing. The Medical Subject Headings (MeSH) thesaurus is a controlled and hierarchically-organised vocabulary produced by the National Library of Medicine. It is used for indexing, cataloging, and searching for biomedical and health-related information [10].

The aim of this study is to explore how big data may be utilised to support a better hospital pharmacy service, provide potential benefits, and describe the potential future use. Furthermore, population description and pattern are to be explored. Advanced analytics are then applied to patient profiles, thus to offer a planning-based prediction. The original Python 3 coding is included so the reader could replicate this project to give results of their own.

## Methods

In order to run Python, we first install the Anaconda distribution of Python through the link: Anaconda.com/downloads, download the .exe installer and run the .exe installer. After the installation of Anaconda is completed, we go to the Windows start menu and select the Anaconda Navigator. On the Navigator's Home tab, we open Jupyter Notebook and then choose Python 3 to run the code.

In this study, we used the Hospital Pharmacy Management System (PMS) Query Template System to extract data from 01 April 2019 to 31 March 2020. The maximum number of rows of each Query Template System's search is equal to the "hexadecimal value" of $(FFFF)_{16}$ which is equivalent to the "decimal value" of $(15 \times 16^3)+(15 \times 16^2)+(15 \times 16^1)+(15 \times 16^0)$, and hence equal to $65535_{10}$. Therefore, we had to download the data weeks by weeks. For example, from 01 April 2019 to 07 April 2019, from 08 April 2019 to 14 April 2019, from 15 April 2019 to 21 April 2019 and so on. Each separated excel file was then converted to a comma-separated values file (.CSV). The total number of CSV files of our raw data was 52. The sample codes are as follows:

### Python coding

```
df1 = pd.read_csv (r'C:\File Location\filename1.csv')
df2 = pd.read_csv (r'C:\File Location\filename2.csv')
.
.
df52 = pd.read_csv (r'C:\File Location\filename52.csv')
```

frames = [df1 … df52]

merge = pd.concat(frames)

The search resulted in a dataset of (1765910)10 rows and this exceeded the Microsoft Excel limits of $(1048576)_{10}$ rows which is equivalent to hexadecimal value = $(100000)_{16}$). The "date of birth" format of the raw data is presented as YYYY. We converted to years (for example if the date of birth was 1945, the years would be 75 in 2020) and presented in age groups (0-9, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99). We adopted and modified the age group structure according to the Hong Kong Information Services Department and the World Fact Book published by the USA's The Central Intelligence Agency (CIA) [11,12].

## Python coding

DOB #Convert DOB to AGE

DOB.to_excel(r'C:\File Location\filename56.xlsx') #Export to Excel

DOB1 = 2020-DOB

cats1=pd.cut(DOB1,bins=[0,10,20,30,40,50,60,70,80,90,100])

cats1 #Report patients' age group distribution

DOB_N = mergeN ['DOB']

DOB_N1 = 2020-DOB_N

We used Python 3 to hold and analyse all the dataset. No manual data transfer was involved. The patients' demographic data, healthcare metrics, number of medications, and the top ten medications dispensed were reported. Further analysis was used to group individuals into basic categories, such as "patient who is aged from 80-100", "number of dispensing records", "number of prescriptions handled". These "buckets" were then be used to target specific interventions that are appropriate for the largest number of people in these specific groups.

## Overall python coding

import pandas as pd

importmatplotlib.pyplot as plt

frommatplotlib.pyplot import pie, axis, show%matplotlib inline

importos, glob

path = 'C:/File Location/'all_files = glob.glob(os.path.join(path, "_DOB*.csv"))

df1 = pd.read_csv (r'C:\File Location\filename1.csv')

df2 = pd.read_csv (r'C:\File Location\filename2.csv')

df3 = pd.read_csv (r'C:\File Location\filename3.csv')

df4 = pd.read_csv (r'C:\File Location\filename4.csv')

df5 = pd.read_csv (r'C:\File Location\filename5.csv')

df6 = pd.read_csv (r'C:\File Location\filename6.csv')

df7 = pd.read_csv (r'C:\File Location\filename7.csv')

df8 = pd.read_csv (r'C:\File Location\filename8.csv')

df9 = pd.read_csv (r'C:\File Location\filename9.csv')

df10 = pd.read_csv (r'C:\File Location\filename10.csv')

df11 = pd.read_csv (r'C:\File Location\filename11.csv')

df12 = pd.read_csv (r'C:\File Location\filename12.csv')

df13 = pd.read_csv (r'C:\File Location\filename13.csv')

df14 = pd.read_csv (r'C:\File Location\filename14.csv')

df15 = pd.read_csv (r'C:\File Location\filename15.csv')

df16 = pd.read_csv (r'C:\File Location\filename16.csv')

df17 = pd.read_csv (r'C:\File Location\filename17.csv')

df18 = pd.read_csv (r'C:\File Location\filename18.csv')

df19 = pd.read_csv (r'C:\File Location\filename19.csv')

df20 = pd.read_csv (r'C:\File Location\filename20.csv')

df21 = pd.read_csv (r'C:\File Location\filename21.csv')

df22 = pd.read_csv (r'C:\File Location\filename22.csv')

df23 = pd.read_csv (r'C:\File Location\filename23.csv')

df24 = pd.read_csv (r'C:\File Location\filename24.csv')

df25 = pd.read_csv (r'C:\File Location\filename25.csv')

df26 = pd.read_csv (r'C:\File Location\filename26.csv')

df27 = pd.read_csv (r'C:\File Location\filename27.csv')

df28 = pd.read_csv (r'C:\File Location\filename28.csv')

df29 = pd.read_csv (r'C:\File Location\filename29.csv')

df30 = pd.read_csv (r'C:\File Location\filename30.csv')

df31 = pd.read_csv (r'C:\File Location\filename31.csv')

df32 = pd.read_csv (r'C:\File Location\filename32.csv')

df33 = pd.read_csv (r'C:\File Location\filename33.csv')

df34 = pd.read_csv (r'C:\File Location\filename34.csv')

df35 = pd.read_csv (r'C:\File Location\filename35.csv')

df36 = pd.read_csv (r'C:\File Location\filename36.csv')

df37 = pd.read_csv (r'C:\File Location\filename37.csv')

df38 = pd.read_csv (r'C:\File Location\filename38.csv')

df39 = pd.read_csv (r'C:\File Location\filename39.csv')

df40 = pd.read_csv (r'C:\File Location\filename40.csv')

df41 = pd.read_csv (r'C:\File Location\filename41.csv')

df42 = pd.read_csv (r'C:\File Location\filename42.csv')

df43 = pd.read_csv (r'C:\File Location\filename43.csv')

df44 = pd.read_csv (r'C:\File Location\filename44.csv')

df45 = pd.read_csv (r'C:\File Location\filename45.csv')

df46 = pd.read_csv (r'C:\File Location\filename46.csv')

df47 = pd.read_csv (r'C:\File Location\filename47.csv')

df48 = pd.read_csv (r'C:\File Location\filename48.csv')

df49 = pd.read_csv (r'C:\File Location\filename49.csv')

df50 = pd.read_csv (r'C:\File Location\filename50.csv')

df51 = pd.read_csv (r'C:\File Location\filename51.csv')

```
df52 = pd.read_csv (r'C:\File Location\filename52.csv')

frames = [df1, df2, df3, df4, df5, df6, df7, df8, df9, df10, df11, df12, df13, df14, df15, df16, df17, df18, df19, df20, df21, df22, df23, df24, df25, df26, df27, df28, df29, df30, df31, df32, df33, df34, df35, df36, df37, df38, df39, df40, df41, df42, df43, df44, df45, df46, df47, df48, df49, df50, df51, df52]

#1732307 lines #1-52

merge = pd.concat(frames)

merge # total visits

Item_Rx = merge.filter(["PAT_KEY", "TICKNUM"])

Item_Rx

Item_Rx1=Item_Rx.groupby(['PAT_KEY','TICKNUM']).size().reset_index(name='Item Number')

Item_Rx1

Item1= Item_Rx1 ['Item Number']

Item_report=pd.cut(Item1,bins=[0,1,2,3,4,5,6,10,15,20,30])

Item_report

ax = Item_report.value_counts().plot(kind='bar', figsize=(14,8), title="Number of Item")

ax.set_xlabel("Number of Item")

ax.set_ylabel("Frequency")

Item_report.value_counts().plot(kind='bar')

importmatplotlib.pyplot as plt

Item = [Item1]

bins = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,30]

plt.hist(Item, bins, histtype='bar', rwidth=0.8)

plt.xlabel('Number of Items')

plt.title('Number of Items per Prescription')

plt.show()

mergeX = merge.drop_duplicates(subset='PAT_KEY', keep='first') #Report number of individual patient

sns.set(font_scale=1.4)

mergeX['SEX'].value_counts().plot(kind='bar', figsize=(7, 6), rot=0)

plt.xlabel("Gender", labelpad=14)

plt.ylabel("Count of Patient", labelpad=14)

plt.title("Count of Patient by Gender", y=1.02);

mergeX #127337 rows

mergeN = merge.drop_duplicates(subset='MOE_ORD_NO', keep='first') #number of visits, same pt can visit more than one time

DOB = mergeX ['DOB']

DOB #Convert DOB to AGE

DOB.to_excel(r'C:\File location\Filename53.xlsx') #Export to Excel

cats1=pd.cut(DOB1,bins=[0,14,25,34,44,54,64,85,120])

cats1 #Report patients' age group distribution

DOB_N = mergeN ['DOB']

DOB_N1 = 2020-DOB_N

cats2=pd.cut(DOB_N1,bins=[0,14,25,34,44,54,64,85,120])

cats2.value_counts().plot(kind='bar') #Report patients' age group distribution

plt.xlabel('age groups')

plt.ylabel('Number of patient')

plt.title('Histogram')

plt.show()

importmatplotlib.pyplot as plt

population_age = [DOB1]

bins = [0,10,20,30,40,50,60,70,80,90,100]

plt.hist(population_age, bins, histtype='bar', rwidth=0.8)

plt.xlabel('age groups')

plt.ylabel('Number of people')

plt.title('Histogram')

plt.show()

mergeX['PHS_SPECIALTY'].value_counts().plot(kind='bar')

ax = MergeNN.head(20).plot(kind='bar', figsize=(10,10), title="Specialty")

ax.set_xlabel("Specialty")

ax.set_ylabel("Frequency of patient visit")

med = mergeN.groupby(['DRUG_NAME','STRENGTH','FORM_DESC','BASEUNIT'])['ISS_QTY'].agg('sum').reset_index()

med.to_excel(r'C:\File location\Filename54.xlsx')

med #report

med_sorted = med.sort_values(by='ISS_QTY', ascending=False) # sort by ISS_QTY

med_sorted.to_excel(r'C:\File location\Filename55.xlsx')

med_sorted.head(10)
```

## Results

The data collection period was from 01 April 2019 to 31 March 2020. The total number of dispensing records retrieved was 1,765,910. Demographically, the sex ratio for the study period was 7 males to 6 females. 127,337 patients paid 358,802 visits to our pharmacy with 381,060 prescriptions handled. A total of 1,211 different drug items were dispensed.

For the 127,337 individual patients, 9% of them fall into the age group 0-19, 34% fall into 20-59 and 57% fall into 60-100 while 21% are from 80-100.Within the 381,060 prescriptions, 29% of the prescriptions had 1 dispensing item and 43% of the prescriptions had 1 or 2 items. And 3% of our prescriptions consisted of more than 15 items.Most patients (48,000) visited our medical clinic, 47,000 patients visited the

family clinic, and 36,000 patients visited the psychiatric clinic during the data collection period. Our top ten medication uses are also identified as listed in Tables 1 and 2.

| Data sources | Results |
|---|---|
| Data collection period | 01 April 2019 to 31 March 2020. |
| Number of dispensing records | 1,765,910 |
| Male : Female | 7:6 |
| Total number of individual patients | 127,337 Individuals |
| Total number of visits | 358,802 visits |
| Total number of prescriptions | 381,060 prescriptions |
| Total medications type | 1,211 different drug items |

**Table 1:** Study results.

| Drug Name | Strength | Form description | Unit dispensed | Quantity |
|---|---|---|---|---|
| Amlodipine (Besylate) | 5mg | Tablet | Tab | 3185879 |
| Aspirin | 80mg | Tablet | Tab | 2533148 |
| Amlodipine (Besylate) | 10mg | Tablet | Tab | 1827229 |
| Paracetamol | 500mg | Tablet | Tab | 1818509 |
| Allopurinol | 100mg | Tablet | Tab | 1695299 |
| Famotidine | 20mg | Tablet | Tab | 1000817 |
| Entecavir | 0.5mg | Tablet | Tab | 683605 |
| Calcium (Carbonate) + Vitamin D | 1000mg Ca + 800 iu | Chew Tab | Tab | 624217 |
| BenzhexolHcl | 2mg | Tablet | Tab | 551766 |
| Acetylcysteine (Sugar-Free) | 200mg/Sachet | Granules | Pack | 537463 |

**Table 2:** Top ten medications.

## Discussion

In this study, we demonstrate how data analytics could assist in decision making in healthcare. The analytics helps time-strapped pharmacist to monitor patient data easily in real-time and thus enable them to invest more time in better patient care. The huge volume of raw data we obtain during this research is organised to create insights into the behaviour of our patient populations, prescription type, prescribing pattern, services, and workflows. Big data in this study also uses patient demographic and medical data to predict the "patient orbit".

57% of our patients are aged 60 or above. That means that we should allocate resources for this age group. 21% of our patients are 80-year-olds or above, we could handle their prescriptions with priority and provide special counselling on medication use to improve the drug compliance.

According to our results, we have a significant number of patients visit the hospital psychiatric department (Figures 1-3). In view of this, more resources could be put such as encouraging more pharmacists to take part in the Board of Pharmacy Specialties (BPS) Board Certified Psychiatric Pharmacist (BCPP) program. According to the website of the BCPP program (available from https://www.bpsweb.org/bps-specialties/psychiatric-pharmacy/), the purpose of the BCPP program is to "validate that the pharmacist has the advanced knowledge and experience to optimize outcomes and recovery for patients with mental illness by:

- Designing, implementing, monitoring, and modifying treatment plans for patients.
- Educating patients, health care professionals, and other stakeholders.
- Providing leadership in the health system and public policy to improve the health of persons with mental illness." This is an opportunity to expand our pharmacists' role in this area. In addition, we could approach the key opinion psychiatrist leader to provide clinic pharmacist services and set up a pharmacist-led medications' support and counselling clinic.

*doi: 10.37532/jhmm.2020.4(2).115*

29% of our prescriptions are single-items and 43% of all prescriptions have 1 or 2 items. Currently, we are using the express queue management mode separating the single-item prescriptions from multi-item prescriptions. In view of the result, we could explore the possibility to modify our existing express queue management to include prescriptions of less than or equal to 2 items.

We are a busy pharmacy with 1.7 million dispensing in a year, of the top 10 medications we dispense, 8 of them we do not have any pre-packs to aid the dispensing process. As the duration of our prescriptions is usually 16 or 26 weeks, we could prepare 112 and 182 tablets pre-packs for all the top-10 medications. For Calcium and Vitamin D, the original pack size is 30 tablets, therefore, we could simply add a pre-pack bag of 22 tablets.Furthermore, we could consider deploying automation at our out-patient department to aid the dispensing process. In 2018, King Fahd Specialist Hospital in Saudi Arabia's city of Tabuk opened the first smart pharmacy run by a robot and the smart pharmacy can dispense as many as 240 prescriptions per hour [13].

Why do we use Python 3 in this research? Apparently, there is more than 20 spreadsheet software available and the more common use could be Microsoft Excel, Google Sheets, Libre Office Calc, Smartsheet, Quip, etc. [14,15]. Microsoft Excel is useful spreadsheet software, first introduced in 1987 [16]. By default, Microsoft Excel supports three worksheets in a file, and each worksheet could support up to 1,048,576 rows and 16,384 columns of data. The Excel could have more than three worksheets if the computer has enough memory for the additional data. On the other hand, Python's pandas could handle more than 1,048,576 rows, even 100 million rows. The library is highly optimized for dealing with large datasets through its Data Frame structure. The constraint is the amount of RAM on the computer. Using Python's NumPy, the single calculation could be applied for the entire dataset, instead of doing the calculations iteratively for every element, which could be error-prone.

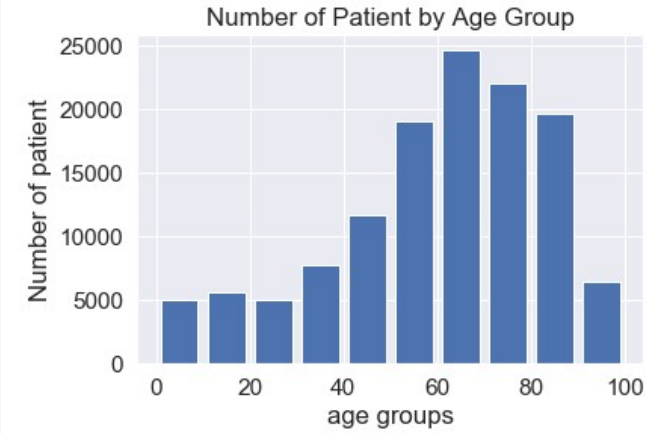


**Figure 1:** Age group distribution (Individual patients).

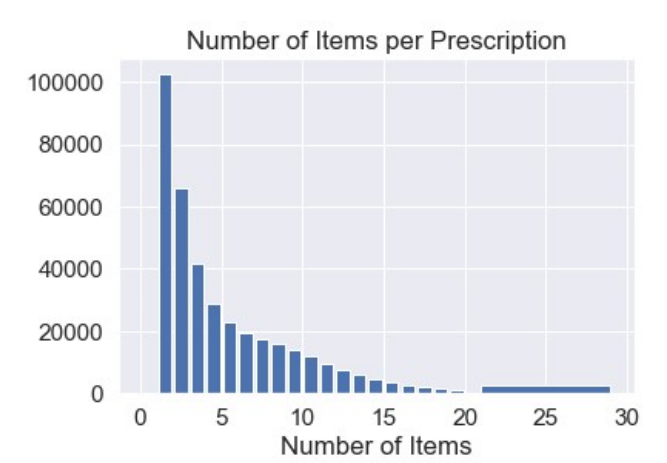


**Figure 2:** Number of items per prescription.

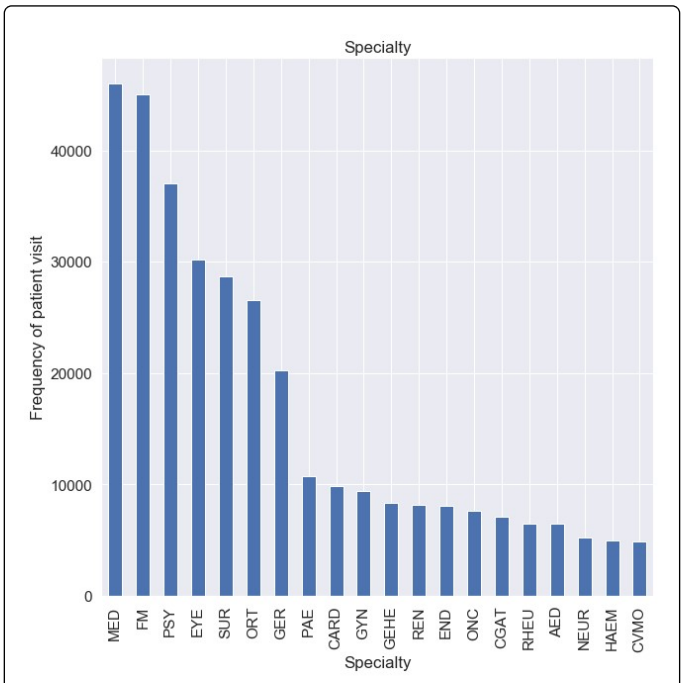


**Figure 3:** Specialty visited by patients.

There is a possibility to expand the use of machine learning in our hospital. First, it could predict side effect outcomes. In a study, Papachristou N., et al used Python 2.7, the Scikit-Learn, Theano, and Lasagne libraries to predict the three common and inter-related symptoms in oncology patients [17]. These kinds of predictive models can be used to identify high-risk patients, educate patients about their symptom experience, and improve the timing of pre-emptive and personalized symptom management interventions.

Big data could reduce prescription errors to improve outcomes and saves lives. According to the Network for Excellence in Health Innovation, prescription errors cost some $21 billion per year, affecting more than 7 million U.S. patients and leading to 7,000 deaths [18]. To overcome this, Israeli startup MedAware is partnering with healthcare

organizations to deploy their decision support tool that uses big data to spot prescription errors before they occur [19].

Big data analysis could help to identify high-risk and high-cost patients as it may be beneficial to patients from early medication interventions, thus saving them from visiting the emergency department [20]. From the data we obtain, we could identify high-risk patients by obtaining the number of discharge, in other word, hospital readmission rate and high-cost patients by acquiring the number of visits per year, number of drugs per prescription, drug-cost, etc.

## Conclusion

To conclude, in hospital pharmacy, big healthcare data is essential for understanding patterns and patient profiles. Big healthcare data are characterized by a large number of patients covered. Therefore, different potential big data uses are proposed in the healthcare setting and, of course, in hospital pharmacy in order to improve effectiveness, safety, efficiency, quality and eventually patient care.

## Funding

There is no funding source.

## Conflict of Interest

None declared

## Key Points

1. The data analytics helps pharmacist to monitor patient data easily in real-time.

2. In hospital pharmacy, big healthcare data is essential for understanding patterns and patient profiles.

3. Python 3 was used to hold and analyses all the dataset.

4. There is a possibility to expand the use of machine learning in our hospital.

## References

1. Rajalakshmi (2015) An Exclusive Study on Unstructured Data Mining with Big Data. International Journal of Applied Engineering Research 10: 3875-3886.
2. Lawrence J (2017) Could big data be the future of pharmacy? The Pharmaceutical Journal 298: 7900.
3. Raghupathi W, Raghupathi V (2014) Big data analytics in healthcare: Promise and potential. Health Information Science and Systems 2:3.
4. Dash A et al (2019) Big data in healthcare: management, analysis and future prospects. Journal of Big Data 6:54.
5. Ma C (2015) Big data in pharmacy practice: Current use, challenges, and the future. Integrated Pharmacy Research and Practice 4: 91-99.
6. Ekmekci B (2016) An Introduction to Programming for Bioscientists: A Python-Based Primer. PLOS Computational Biology 12: e1004867.
7. Download Python for Other Platforms (2020) Available from: https://www.python.org/download/other/
8. Kearney D (2020) First Steps with Python
9. Your data science toolkit (2020) Available from: https://www.anaconda.com/.
10. Medical Subject Headings (2020) National Library of Medicine.
11. The Fact. Published by the Information Services Department. Hong Kong Special Administrative Region Government (2015) Hong Kong.
12. The World Fact Book published by the USA's The Central Intelligence Agency (CIA) (2020). Available from: https://www.cia.gov/library/publications/the-world-factbook/fields/341.html
13. Shaykh M (2018) Saudi Arabia opens first smart pharmacy run by a robot."
14. List of spreadsheet software (2020) Available
15. Brookes T (2019) The Best Spreadsheet Software in 2019. Microsoft Excel vs. Google Sheets—and the other contenders.
16. Microsoft Excel (2020) Available from: https://www.microsoft.com/en-us/microsoft-365/excel?utm_source=zapier.com&utm_medium=referral&utm_campaign=zapier&utm_source=zapier.com&utm_medium=referral&utm_campaign=zapier
17. Papachristou N, Puschmann D, Barnaghi P, Cooper B, Hu X, et al. (2018) Learning from data to predict future symptoms of oncology patients. PLoS ONE 13: e0208808.
18. NEJM Catalyst (2018) Healthcare Big Data and the Promise of Value-Based Care.
19. MedAware (2020) Available from: https://www.medaware.com/.
20. Bates D (2014) Big Data In Health Care: Using Analytics To Identify And Manage High-Risk And High-Cost Patients. Health Affairs 33: 7