Student Name: Palash Vijay Tayade

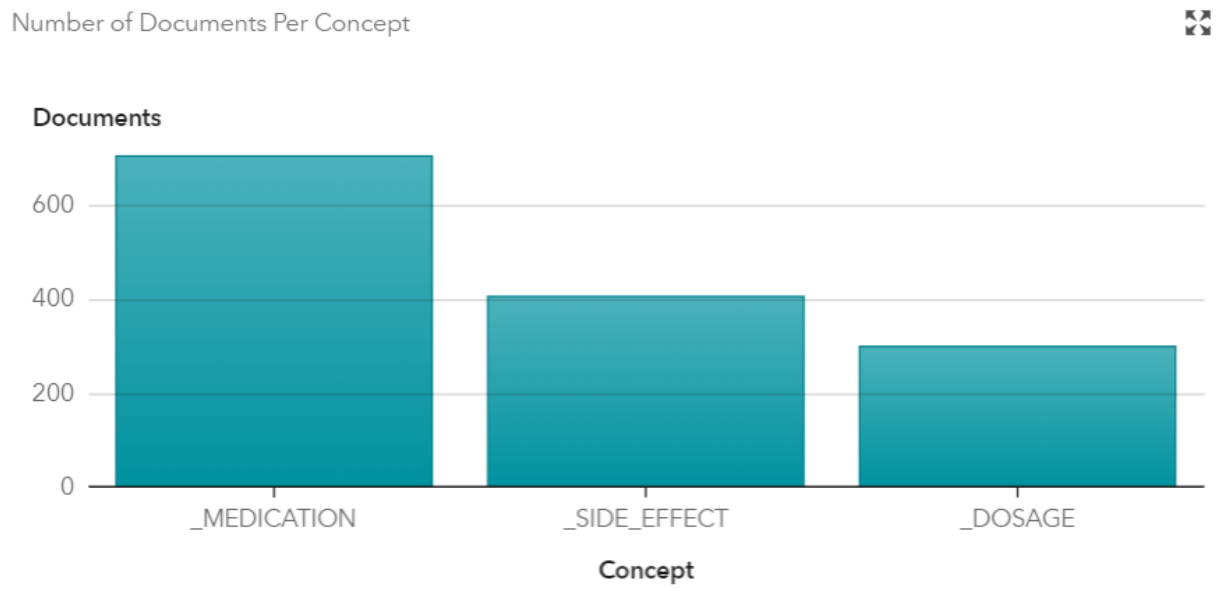
Student ID: 44498632

Assignment No. 3

Answer 1:

Chart 1

Following is a bar chart displaying the number of documents that mention a medication, dosage or a side effect.



How the chart was produced

To produce this chart, I first create a new project in SAS VIYA. I then entered the title as Assignment 3 Q1, type as Text Analytics and DRUG_REPORT as the data source.

Once created, I added the role of Text to the DrugReport variable. This was done because DrugReport variable holds the unstructured data that we will analyze later.

Next, I ran the pipeline with default settings. This step is necessary to Open any one of the nodes in the pipeline.

I then right clicked on Concepts and selected Open. Next, I created four custom concepts called _MEDICATION, _SIDE_EFFECT and _DOSAGE. I then copied and pasted the required code provided to us in the MedicationConcept.txt, SideEffectsConcept.txt and DosageConcept.txt

files. The code for _MEDICATION and _SIDE_EFFECT concepts included a list of CLASSIFIER: value pair, that would identify the mentioned word (value) in the documents. The code for _DOSAGE included two regular expressions used to find the pattern "<<Numeric value>> Mg" or "<<Numeric value>> mg".

Finally, to produce the chart, run the pipeline. If it runs successfully, right click on the Concept and click Results. This would produce the above chart along with some other charts.

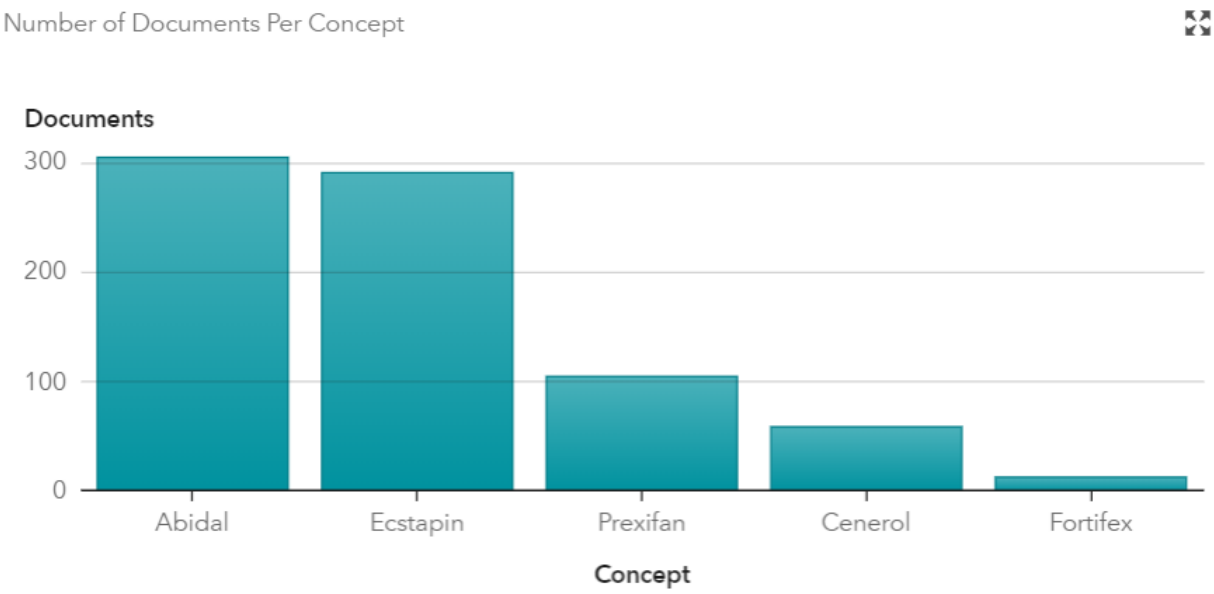
Discussion

The bar chart shown above clearly indicates there were 704 documents that mentioned medication of some type. Similarly, there were 406 and 300 documents that had mentioned some type of side effects and dosage respectively. I later confirmed these results by manually searching the documents for each medication, side effect and dosage.

Answer 2

Chart – Displays the count of number of documents that mentioned the following medication

- **Abidal**
- **Cenerol**
- **Ecstapin**
- **Fortifex**
- **Prexifan**



How the chart was produced

To produce this chart, I first create a new project in SAS VIYA. I then entered the title as Assignment 3 Q2, type as Text Analytics and DRUG_REPORT as the data source.

Once created, I added the role of Text to the DrugReport variable. This was done because DrugReport variable holds the unstructured data that we will analyze later.

Next, I ran the pipeline with default settings. This step is necessary to Open any one of the nodes in the pipeline.

I then right clicked on Concepts and selected Open. Next, I created five custom concepts called Abidal, Ecstapin and Prexifan, Cenerol and Fortifex. I then wrote the classifier rule for each medication (for example CLASSIFIER: Abidal etc.).

Finally, to produce the chart, run the pipeline. If it runs successfully, right click on the Concept and click Results. This would produce the above chart along with some other charts.

Discussion

The bar chart shown above clearly indicates there were 305, 59, 291, 13 and 105 documents that mentioned medication Abidal, Cenerol, Ecstapin, Fortifex and Prexifan respectively. I later confirmed these results by manually searching the documents for each medication.

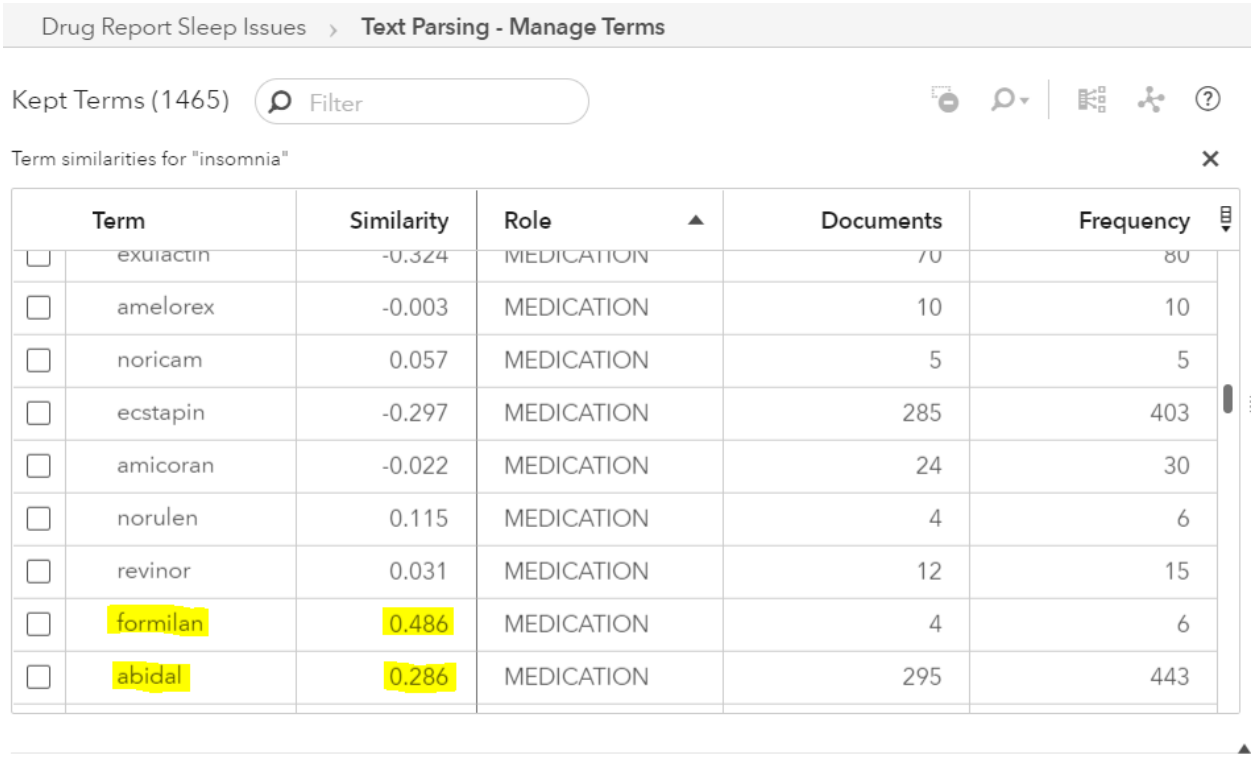
Answer 3

Assumptions – For this question, I have considered Insomnia, Restlessness, Sleeplessness as sleep issues and identified that Formilan, Norulen, Perxifan, Imitap, Captalan are the medications that can cause sleeping disorders in patients.

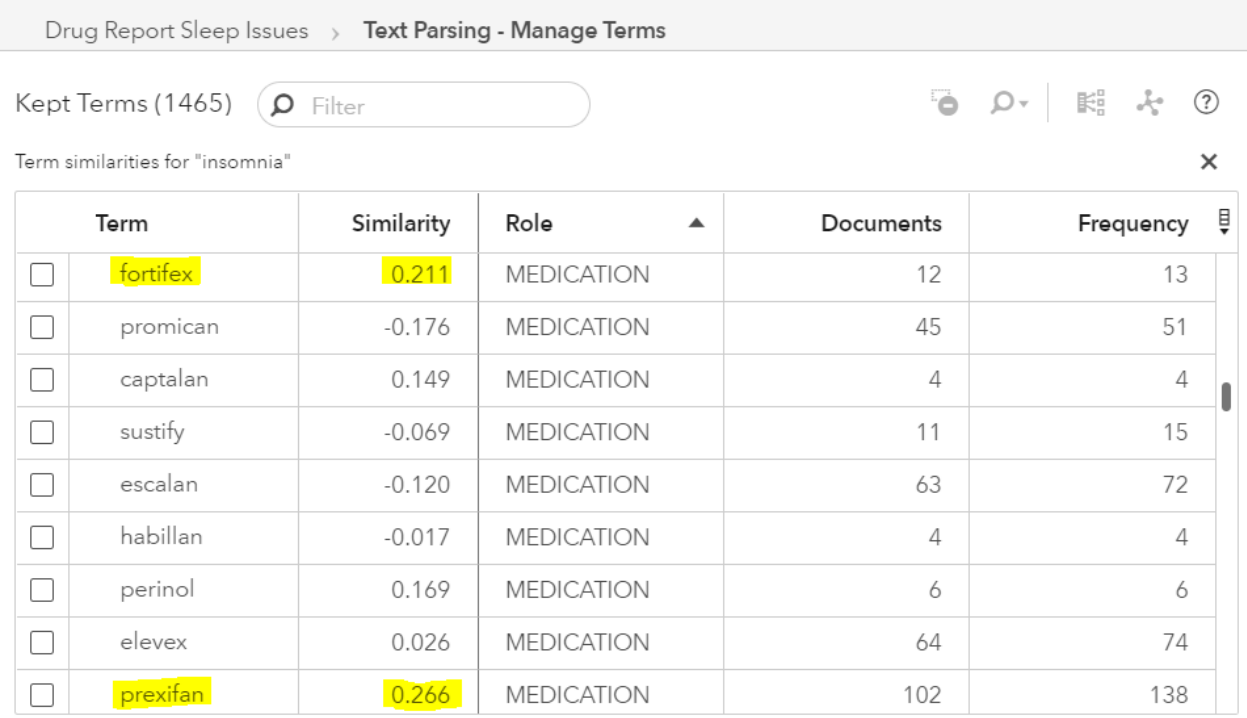
Discussion

Investigating Insomnia

To identify the medications associated with insomnia, I analyzed the similarity score for the term "insomnia" for each medication. The results revealed that the medications Formilan, Abidal, Prexifan and Fortifax each had a score of 0.486, 0.286, 0.266, 0.211 respectively.

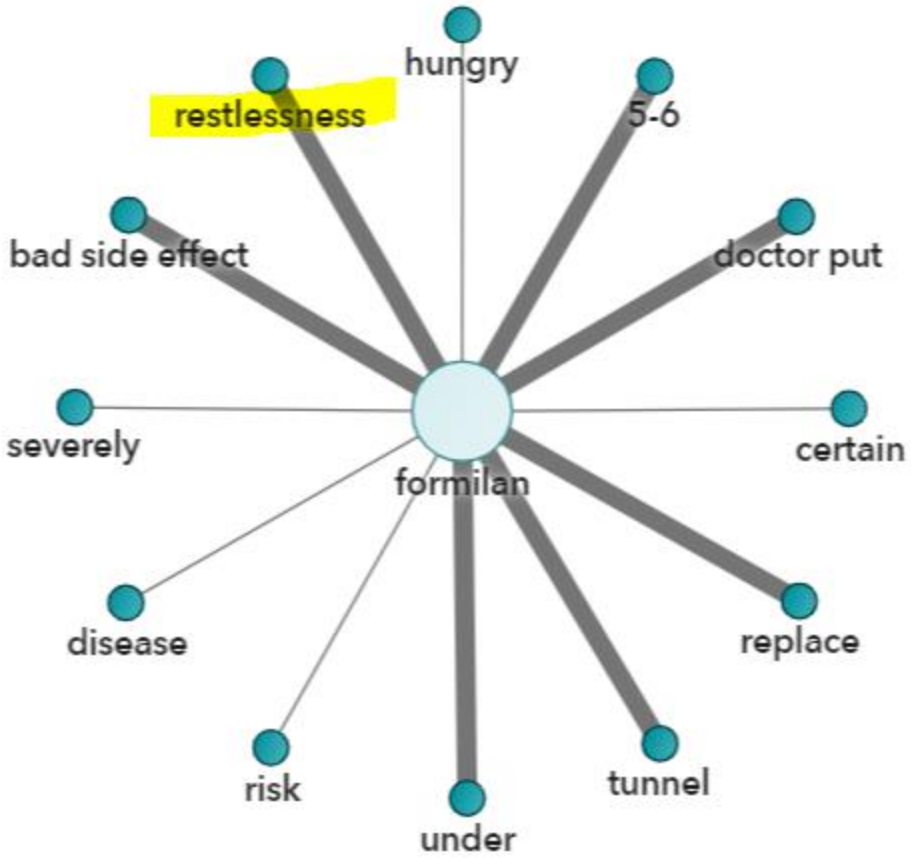


Term	Similarity	Role	Documents	Frequency
exuiactin	-0.324	MEDICATION	70	80
amelorex	-0.003	MEDICATION	10	10
noricam	0.057	MEDICATION	5	5
ecstapin	-0.297	MEDICATION	285	403
amicoran	-0.022	MEDICATION	24	30
norulen	0.115	MEDICATION	4	6
revinor	0.031	MEDICATION	12	15
formilan	0.486	MEDICATION	4	6
abidal	0.286	MEDICATION	295	443



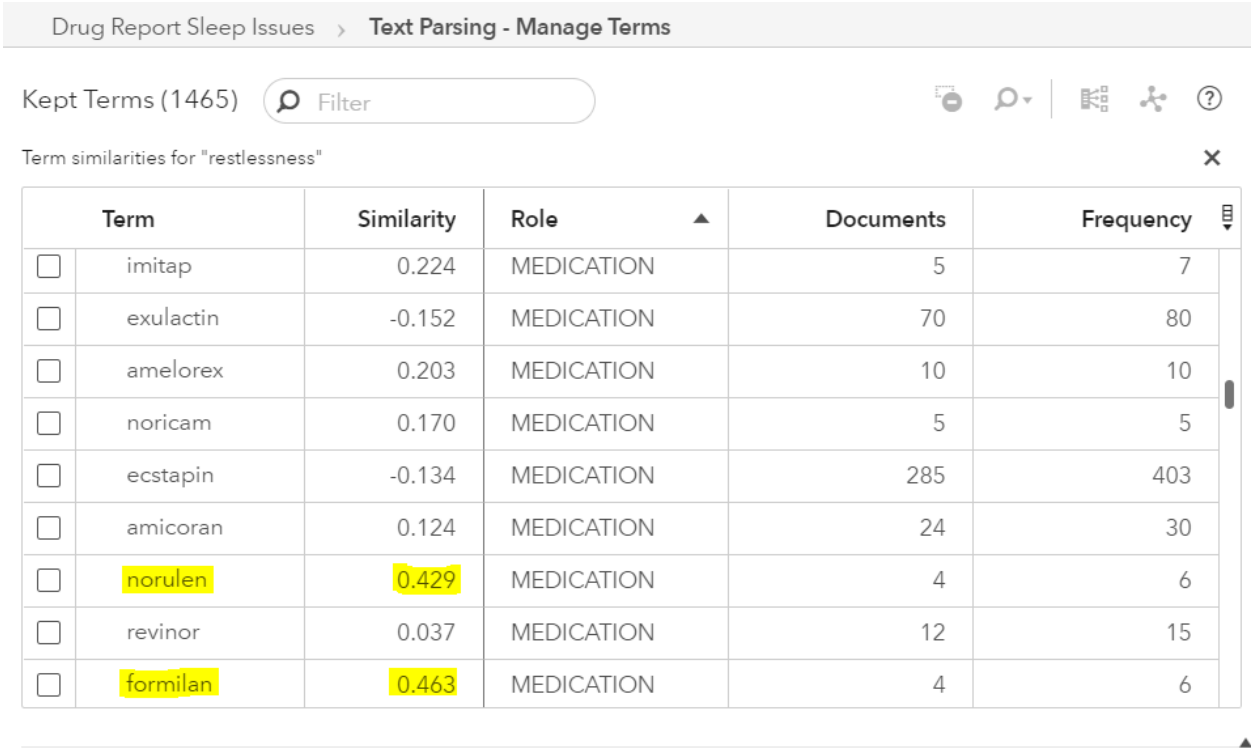
Term	Similarity	Role	Documents	Frequency
fortifex	0.211	MEDICATION	12	13
promican	-0.176	MEDICATION	45	51
captalan	0.149	MEDICATION	4	4
sustify	-0.069	MEDICATION	11	15
escalan	-0.120	MEDICATION	63	72
habillan	-0.017	MEDICATION	4	4
perinol	0.169	MEDICATION	6	6
elevex	0.026	MEDICATION	64	74
prexifan	0.266	MEDICATION	102	138

Now that I had narrowed down the list of possible medication causing insomnia, I further investigated the term map of each above-mentioned medication. Although, other medications did not have any key term related to any sleeping disorder, I noticed that Formilan (which also had the highest similarity score) had a direct relationship with restlessness (which is a symptom of sleeping disorder). The information gain between Formilan and restless was observed to be 7.523 which further raises the suspicion that Formilan might be responsible for causing sleeping disorder.

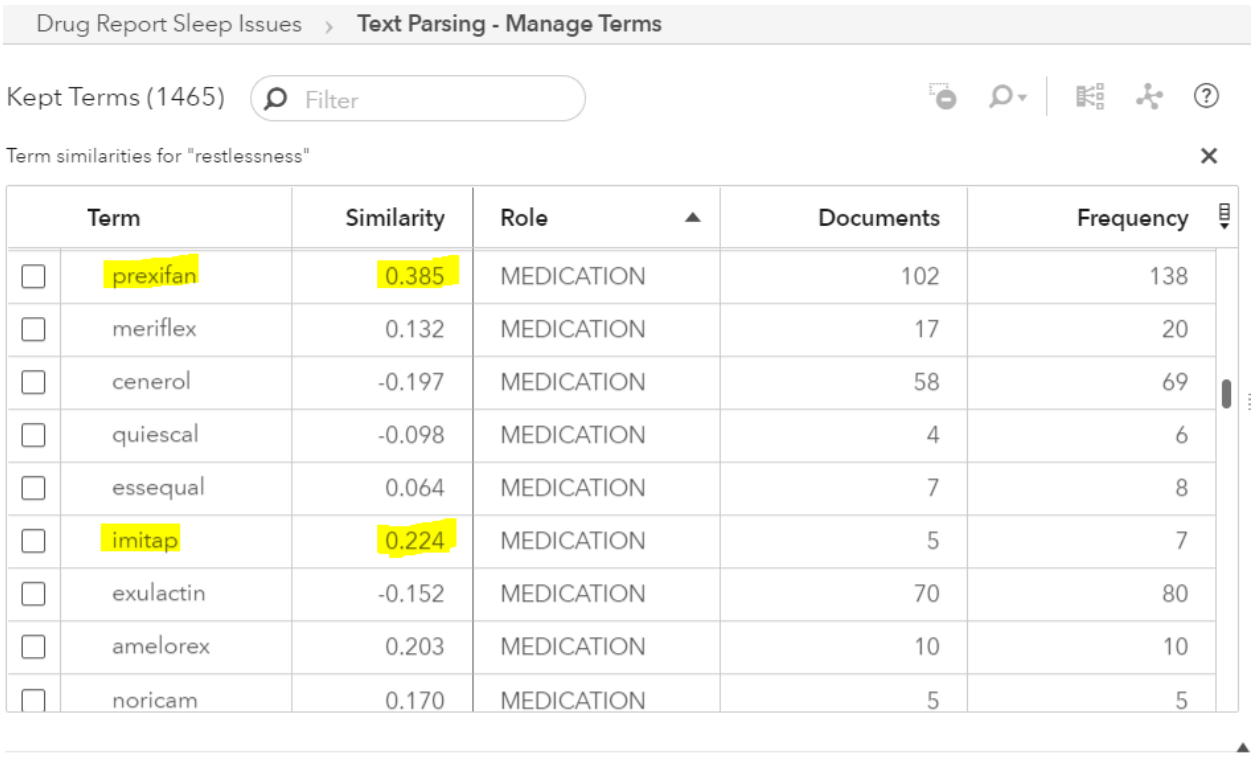


Investigating Restlessness

Using the similar processes as that of insomnia, I first analyzed the similarity score for the term restlessness. The results revealed that the medications Formilan, Norulen, Prexifan and Imitap each had a score of 0.463, 0.429, 0.385, 0.224 respectively.

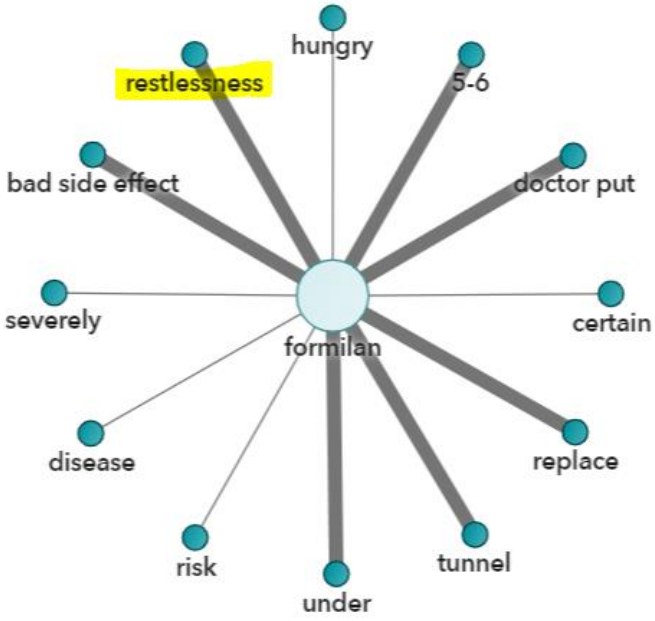


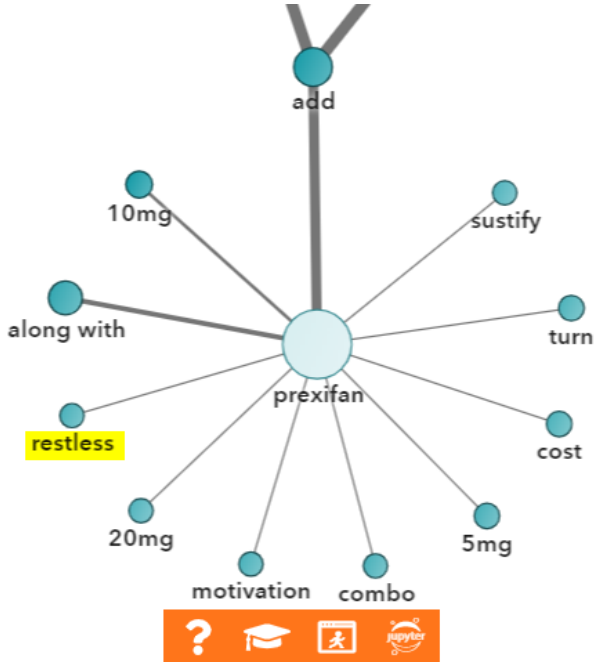
Term	Similarity	Role	Documents	Frequency
imitap	0.224	MEDICATION	5	7
exulactin	-0.152	MEDICATION	70	80
amelorex	0.203	MEDICATION	10	10
noricam	0.170	MEDICATION	5	5
ecstapin	-0.134	MEDICATION	285	403
amicoran	0.124	MEDICATION	24	30
norulen	0.429	MEDICATION	4	6
revinor	0.037	MEDICATION	12	15
formilan	0.463	MEDICATION	4	6



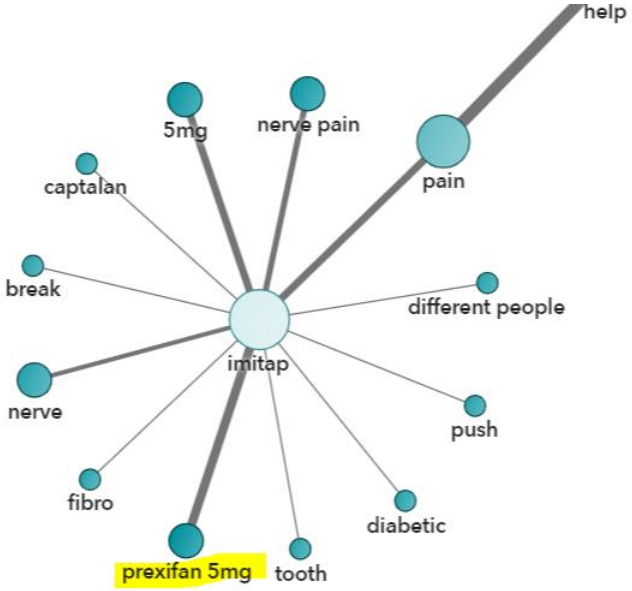
Term	Similarity	Role	Documents	Frequency
prexifan	0.385	MEDICATION	102	138
meriflex	0.132	MEDICATION	17	20
cenerol	-0.197	MEDICATION	58	69
quiescal	-0.098	MEDICATION	4	6
essequal	0.064	MEDICATION	7	8
imitap	0.224	MEDICATION	5	7
exulactin	-0.152	MEDICATION	70	80
amelorex	0.203	MEDICATION	10	10
noricam	0.170	MEDICATION	5	5

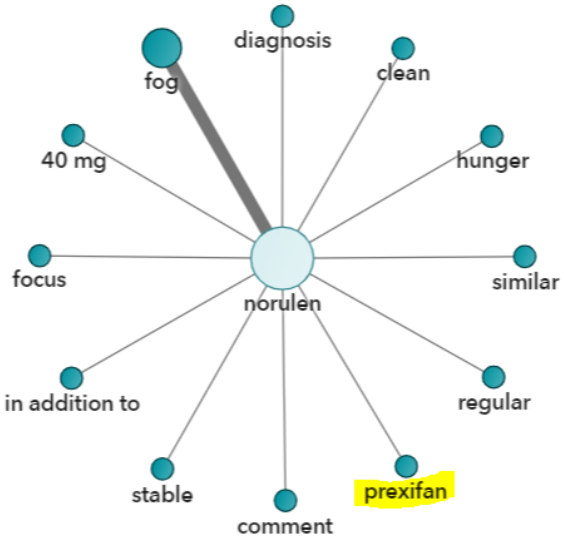
Further investigating the term map for each above-mentioned medication revealed that Formilan had a direct relationship with restlessness with an information gain of 7.523. Also, Prexifan had a direct relationship with restless with an information gain of 8.5874.





Interestingly, Imitap and Norulen term map did not include any direct relation with sleeping disorders, both had a direct relationship with Prexifan with an information gain of 17.975 and 7.022 respectively.





Investigating Sleeplessness

Using the same approach, I first created a list of top 4 medication having the highest similarity score with the term “Sleeplessness”. This revealed that medications Ecstapin, Escalan, Abidal, Captalan had similarity scores of 0.233, 0.202, 0.147, 0.138 respectively.

Kept Terms (1465) Filter

Term similarities for "sleeplessness"

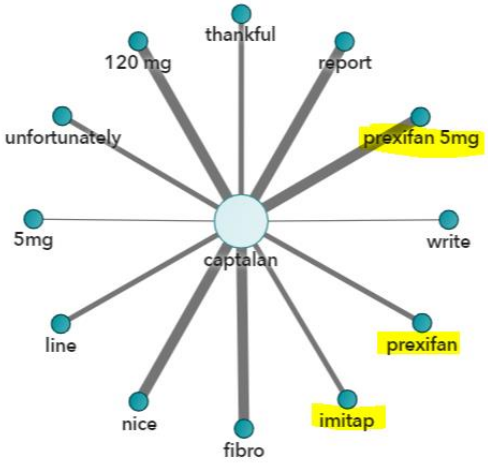
Term	Similarity	Role	Documents	Frequency
captalan	0.138	MEDICATION	4	4
sustify	0.104	MEDICATION	11	15
escalan	0.202	MEDICATION	63	72
habillan	-0.122	MEDICATION	4	4
perinol	-0.149	MEDICATION	6	6
elevex	-0.022	MEDICATION	64	74
prexifan	0.038	MEDICATION	102	138
meriflex	-0.012	MEDICATION	17	20
cenerol	-0.027	MEDICATION	58	69

Kept Terms (1465) Filter

Term similarities for "sleeplessness"

Term	Similarity	Role	Documents	Frequency
noricam	0.058	MEDICATION	5	5
ecstapin	0.233	MEDICATION	285	403
amicoran	-0.031	MEDICATION	24	30
norulen	0.115	MEDICATION	4	6
revinor	0.057	MEDICATION	12	15
formilan	0.110	MEDICATION	4	6
abidal	0.147	MEDICATION	295	443
life	-0.232	N	160	197
disease	0.030	N	5	5

Further, the term map analysis indicated that Ecstapin, Abidal and Escalan did not really had a connection with sleeping disorders. However, Captalan's temp map interestingly had direct connections with Prexifan and Imitap which were earlier identified as the medication that caused restlessness. This is shown in the following snapshot.



Discussion

By making use of similarity score and term map, I was able to identify the medications associated to sleep issues. In certain cases, I was able to identify overlapping medication while investigating different side effects. For example, the medication Prexifan and Imitap were identified in both restlessness and sleeplessness side effects. As the result of my investigation, I was able to identify medications (Formilan, Norulen, Perxifan, Imitap, Captalan) that were associated with sleep issues.

Answer 4

Chart 1 – The following charts represents top 20 concerns mentioned by the user in their post. On the X axis are the side effects and on Y axis are the number of document (with negative sentiments) that mentioned the side effect.

Chart 2 – The following charts represents all concerns mentioned by the user in their post. On the X axis are the side effects and on Y axis are the number of document (with negative sentiments) that mentioned the side effect.

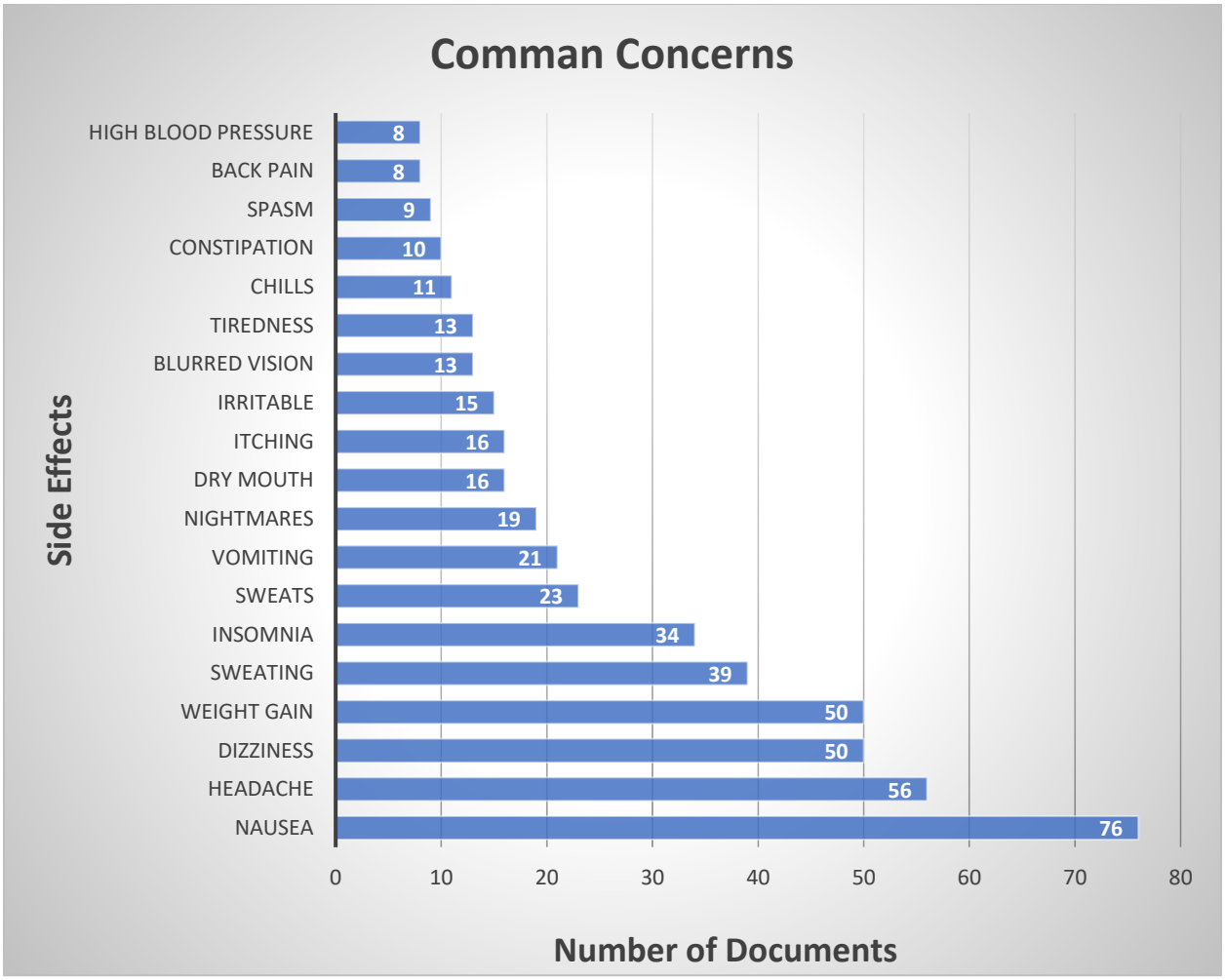


Chart 1

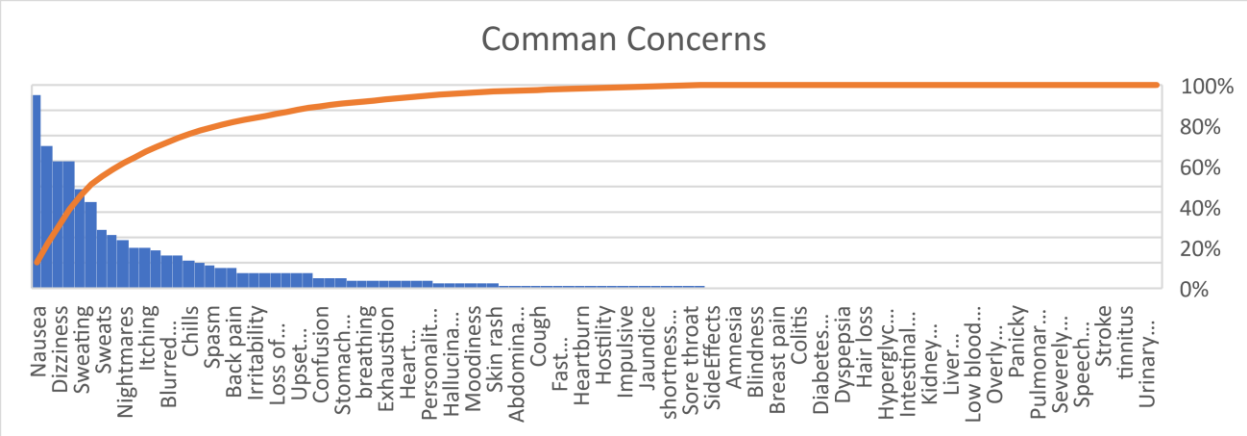


Chart 2

How the chart was produced

Following are the steps taken to produce the above charts.

Step 1 – Perform Sentiment Analysis

To perform the sentiment analysis of each document, I first created a new project in SAS VIYA. I then entered the title as Assignment 3 Q4, type as Text Analytics and DRUG_REPORT as the data source.

Once created, I added the role of Text to the DrugReport variable. This was done because DrugReport variable holds the unstructured data that we will analyze later.

Next, right click on “Topics” node and click “Run”. This runs the pipeline with default settings. This step is necessary to Open any one of the nodes in the pipeline.

Next, right click on “Topics” node and click “Open”. Below in the “Documents” section, verify that sentiments are available for each document.

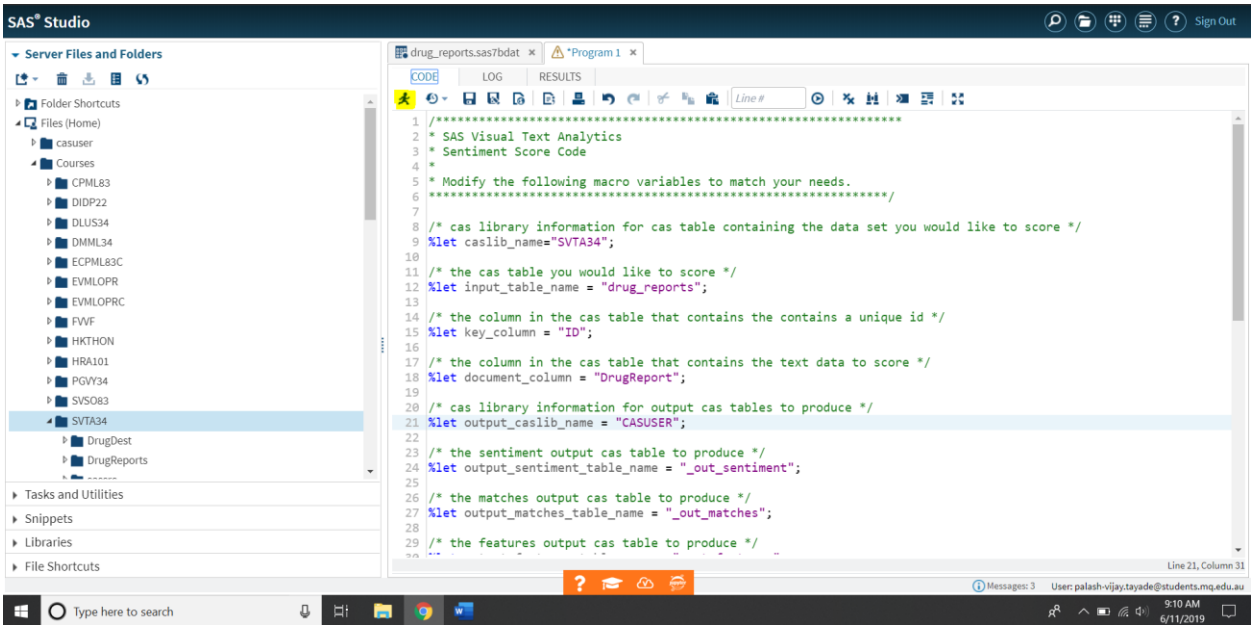
Step 2 – Retrieve Sentiment Analysis data

Right click on “Sentiment” node and click “Result”. Copy the code displayed in the “Sentiment Source Code” window.

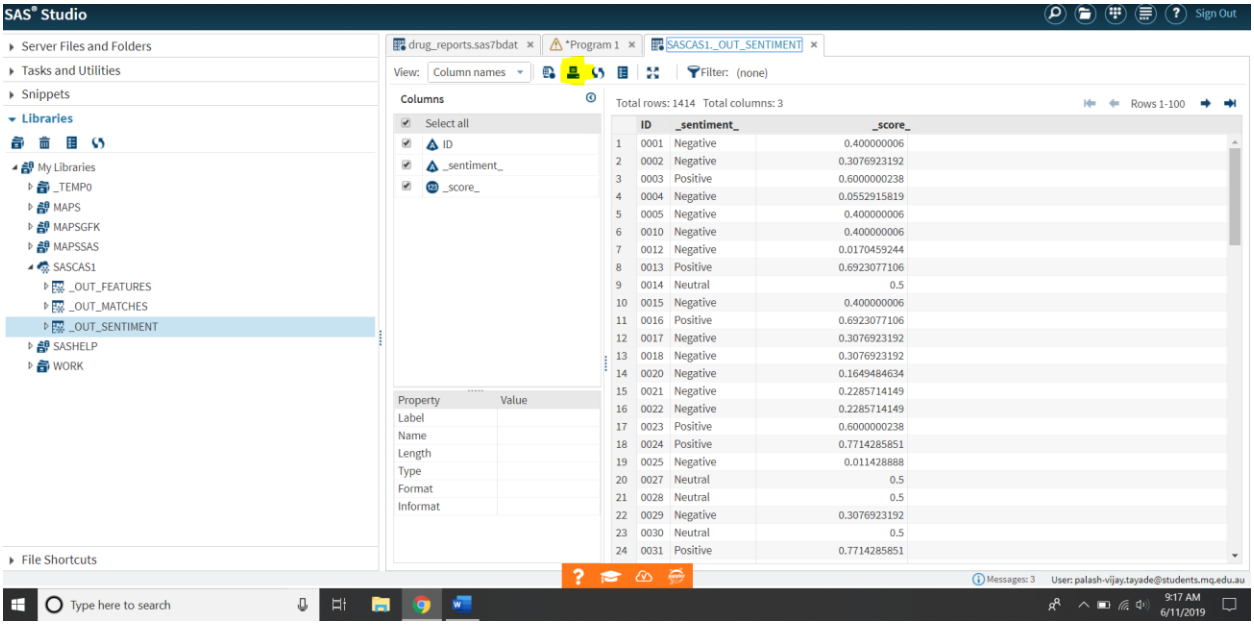
Open SAS studio, in “Servers Files and Folder” click new “SAS program” or press F4 to create a new program. Paste the copied code in the program window. Now set the following variable to the indicated values.

- caslib_name="SVTA34";
- input_table_name = "drug_reports";
- key_column = "ID";
- document_column = "DrugReport";
- output_caslib_name = "CASUSER";

The program should look like follows, click the “Submit” button highlighted in the snapshot below.



This should create 3 tables which will be displayed in the “Result” tab. Now click “Libraries” from the left pane. Next navigate to “My Library” – > ”SASCAS1”. Double click on table “_OUT_SENTIMENT”. This table displays the sentiment value associated with each document. Next click “Print” button indicated in the following snapshot to print to a pdf file.



Finally, since the PDF generated in the last step are just images of the table, use an online OCR (Optical Character Recognition) tool and convert the PDF into a CSV file.

Step 3 – Filter Negative Sentiment Documents

The next step is to filter negative sentiments from the sentiments CSV file generated in the last step. (Why this step is important is explained in the Discussion section below). To do this I wrote a Python program that reads the Documents.csv file and SentimentsData.csv file.

Note- To limit this section of report, I have intentionally not included the step to export DrugReport documents data into Documents.csv file since they were post by Prof. Diego on ILearn.

The program, "ExtractConcern.py" consist of the function "extractNegativeDocuments" which hold the logic to delete all the documents whose sentiments are not Negative. To achieve this, it reads gets the document_ID of the documents from the SentimentData.csv whose sentiment value is not negative. It then deletes such documents that have the same document_ID from the Documents.csv file.

Step 4 – Extract the most common side-effects

Now that we have all negative sentiment documents, the program read all side effects mentioned in the "SideEffectConcepts.txt" file and scans all the documents. It maintains a count for each side-effect which is incremented for every document that contains the side-effect.

Finally, it writes this data into a file called "OutputAfterSentimentAnalysis.csv". Following is the snippet of the program.

```
import csv

def extractNegativeDocuments(documents):
    #function that removes all positive and neutral documents
    with open('SentimentData.csv',"r") as csv_file:
        csv_reader = csv.reader(csv_file)
        csvFile_rows= list(csv_reader)

       #delete all documents that do not have negative sentiment
        for i in range(0,len(csvFile_rows)):
            if csvFile_rows[i][2] != "Negative":
                id=csvFile_rows[i][1]
                del documents[id]
        return documents

#Read the Document.csv file and store it into a dictionary.
with open("Documents.csv", "r") as f:
    lines = f.read().splitlines()
    temp = list(lines)
    documents={}
```

```
    #creating a dictionary of documents where the key is the documentID and the v
alue is the feedback posted by the user.
    for i in range(len(temp)):
        documents[temp[i].split("\",\"")[2]]=temp[i].split("\",\"")[0]

    extractNegativeDocuments(documents)

    #Read the Side effects of medication provided in the SideEffectsConcept.txt f
ile
    with open('SideEffectsConcept.txt',"r") as f:
        sideEffect = f.read().split('\n')
    sideEffectCount =  {}

    #create a dictionary that store the count of documents that mention a side ef
fect
    for row in sideEffect:
        sideEffectCount[row]=0

    #For each side effect, count the number of documents that mentions it and sto
re the count in sideEffectCount dictionary
    for key in sideEffectCount.keys():
        count=0
        for value in documents.values():
            if key.upper() in value.upper():
                count+=1
        sideEffectCount[key] = count

#Export the data from sideEffectCount dictionary into a file
w = csv.writer(open("OutputAfterSentimentAnalysis.csv", "w"))
for key, val in sideEffectCount.items():
    w.writerow([key, val])
```

Step 5 – Plot the data from "OutputAfterSentimentAnalysis.csv"

The OutputAfterSentimentAnalysis.csv file hold the count of the number of documents that mentioned the give side effect. This file was then open in Microsoft Excel and was the data was plotted using built in charts. Following is a partial snapshot of the OutputAfterSentimentAnalysis.csv file.

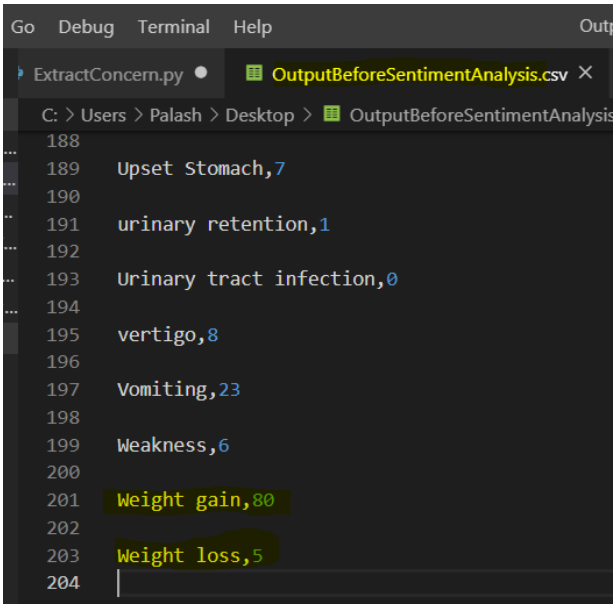
```
Abdominal pain,1
Aggression,0
Agitation,1
```

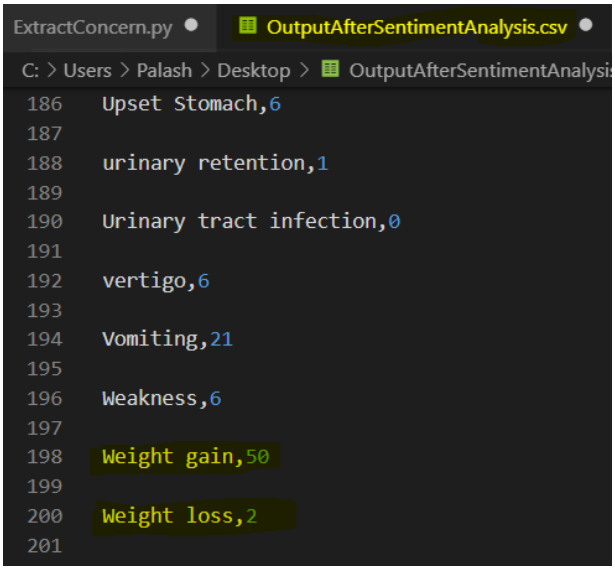
Discussion

To uncover the most common concerns expressed by the user in their post, the approach that I have used identifies the most common side-effects mentioned in the posts. To achieve this, I followed the following steps.

1. Perform Sentiment Analysis
2. Retrieve Sentiment Analysis data
3. Filter Negative Sentiment Documents
4. Extract the most common side-effects
5. Plot the extracted data

The steps 1,2,4 and 5 are self-explanatory and are discussed in detail in the above sections. For this section, lets focus on step 3. The approach that I have used to answer this question is by identifying the most common side effect mentioned in the posts by different the users. To understand the importance of step 3, consider the following example. Weight gain can be a type of side effect of a medication. But this is not necessarily true, since there can be other medications whose goal is to promote weight gain. So, it becomes crucial to analyze the sentiment of the post before regarding it as a side-effect. In other words, posts/documents that have a positive or neutral tone should be disregarded since they induce error in our analysis. For example, a post, "This medication has helped me with weight gain …", should be ignored as in this case "weight gain" is not a side effect. On the other hand, posts/documents with a negative tone mentioning a side-effect should be included. This increases the accuracy of the overall analysis. Following snapshot indicate the difference in results obtained before and after performing sentiment analysis.





Answer 5

Chart 1 – The following chart displays the ratio of negative, positive and neutral review posted by the user for each medication.

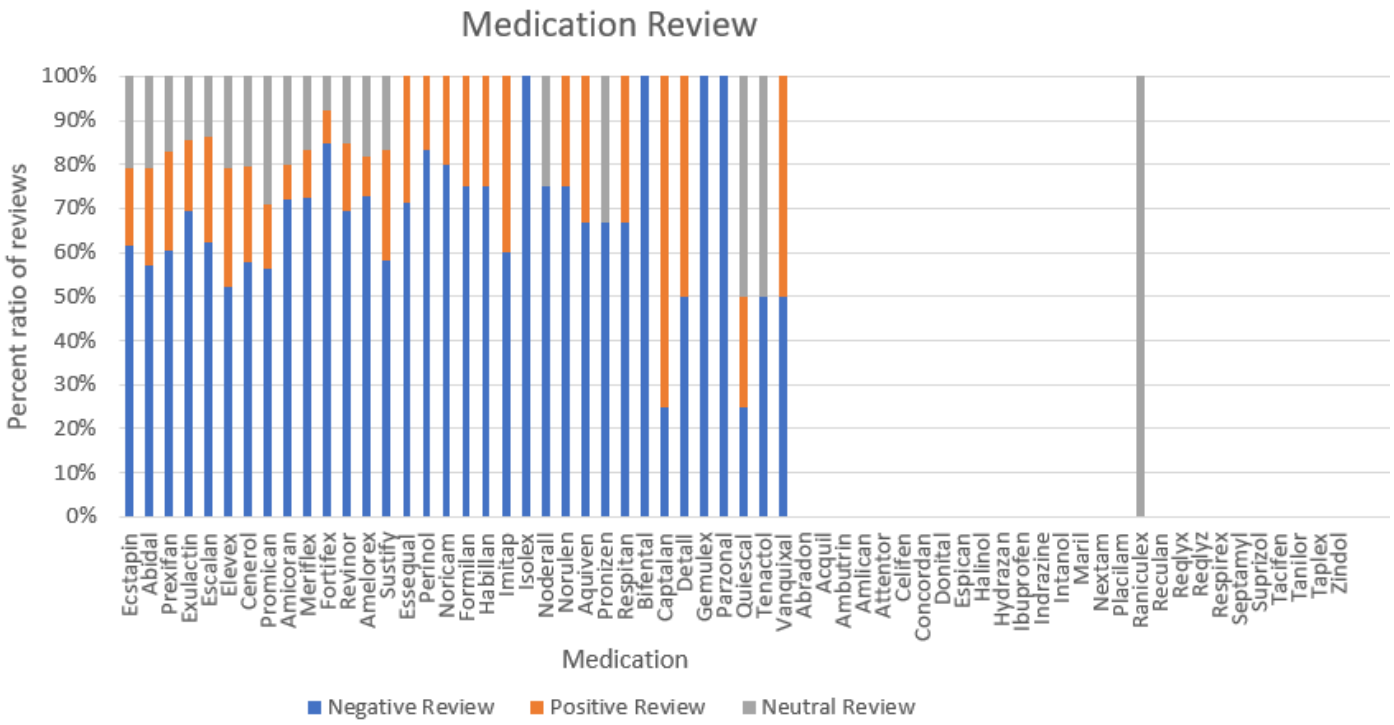


Chart 2 – The following chart displays the number of negative, positive and neutral review posted by the user for each medication.

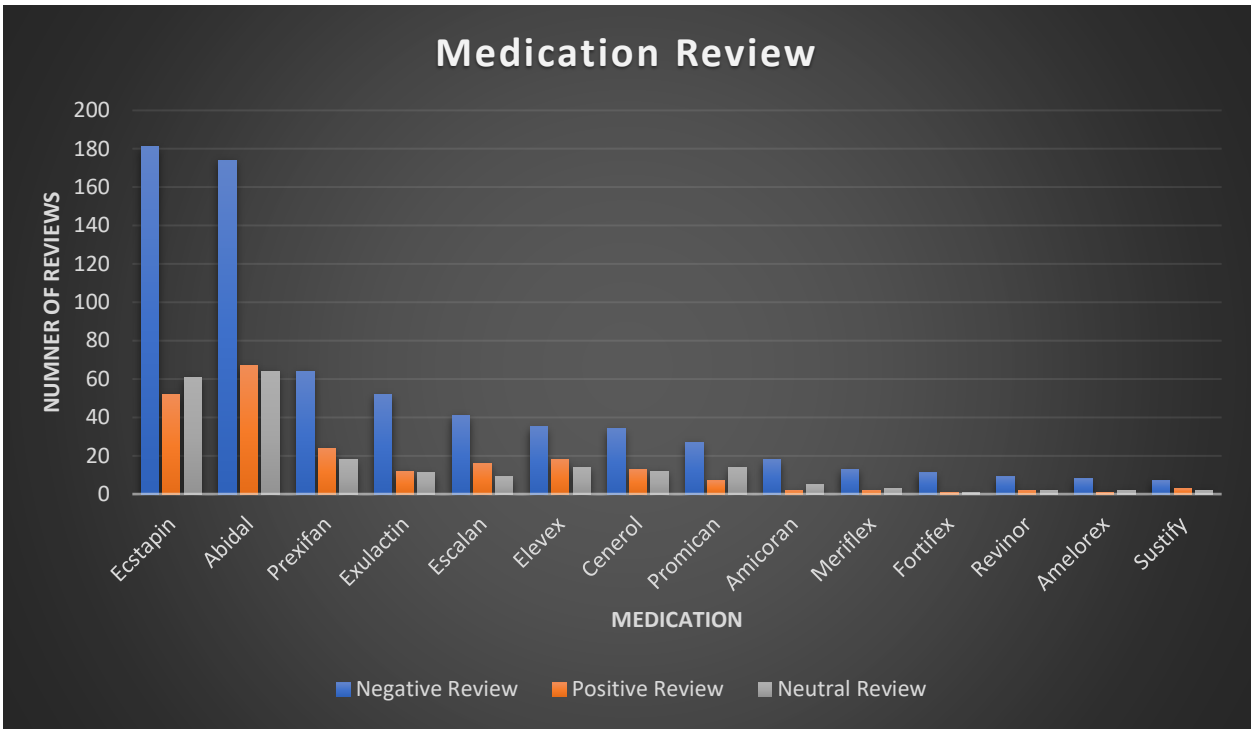
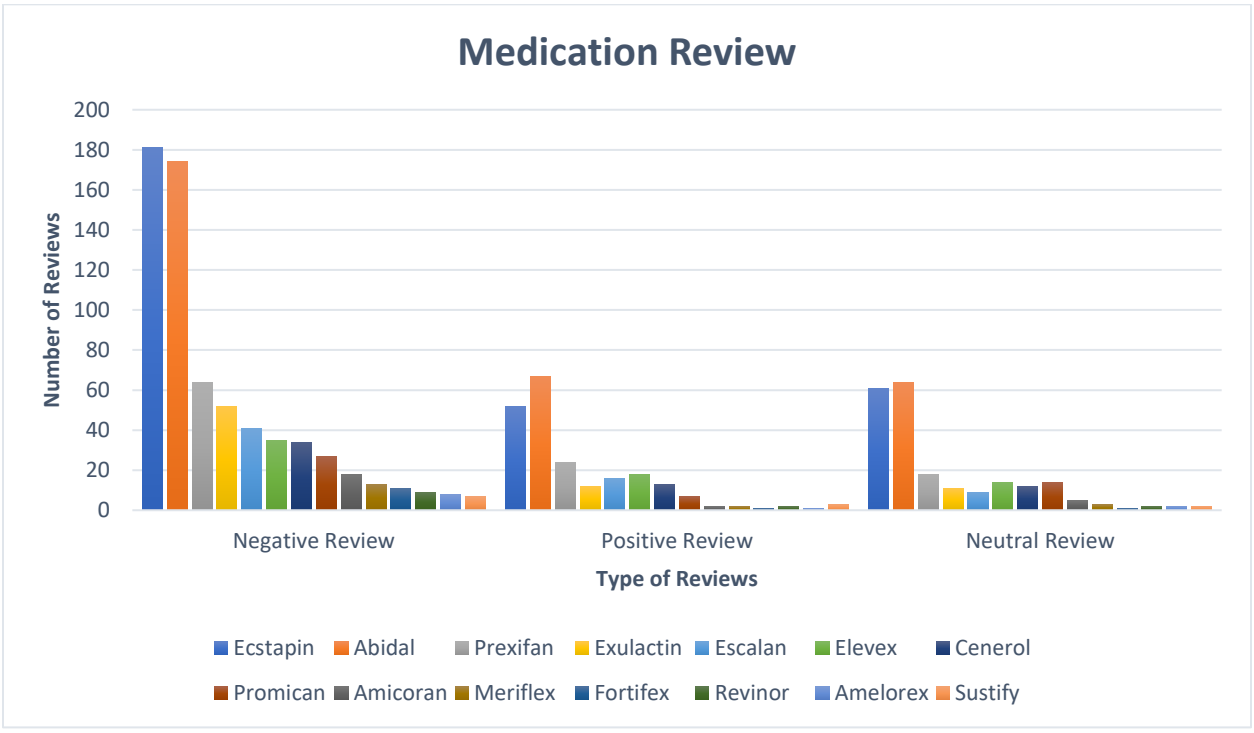


Chart 3 – The following chart displays the comparison of medication review of each type.



How the chart was produced

The charts produced in this answer used the data retrieved while answering question 4. For this answer, I wrote a python program (ExtractMedicationReview.py) that extracted negative, positive and neutral reviews for each medication. The program consists of 4 functions out of which 3 functions; extractNegativeDocuments, extractPositiveDocuments and extractNeutralDocuments are used to filter documents having negative, positive and neutral sentiments respectively.

The function getMedicReview takes to parameters; document and medicationName. It then identifies all the documents that mentions the supplied medicationName. This way we get the count of documents for each medication mentioned in negative, positive and neutral documents. This process is repeated for each medication mentioned in the MedicationConcept.txt file. In the end, the program outputs a CSV file called "MedicationReview.csv" which contain the medicationName and count of document mentions in each type of review (negative, positive and neutral). The following is the snapshot of the MedicationReview.csv produced by the program.

```
Abidal,"174,67,64"

Abradon,"0,0,0"

Acquil,"0,0,0"

Ambutrin,"0,0,0"

Amelorex,"8,1,2"

Amicoran,"18,2,5"

Amlican,"0,0,0"

Aquiven,"2,1,0"

Attentor,"0,0,0"

Bifental,"1,0,0"
```

This data was later plotted in using Microsoft Excel's built in bar charts. Following is the program ExtractMedicationReview.py used to extract the data.

```
import csv
import copy
```

```
def extractNegativeDocuments(documents,csvFile_rows):
    #function that removes all positive and neutral sentiment documents

    negativeDocuments = copy.deepcopy(documents)

   #delete all documents that do not have negative sentiment
    for i in range(0,len(csvFile_rows)):
        if csvFile_rows[i][2] == "Positive" or csvFile_rows[i][2] == "Neutral":
            id=csvFile_rows[i][1]
            del negativeDocuments[id]
    return negativeDocuments


def extractPositiveDocuments(documents,csvFile_rows):
    #function that removes all negative and neutral documents

    positiveDocuments = copy.deepcopy(documents)

    #delete all documents that do not have positive sentiment
    for i in range(0,len(csvFile_rows)):
        if csvFile_rows[i][2] == "Neutral" or csvFile_rows[i][2] == "Negative":
            id=csvFile_rows[i][1]
            del positiveDocuments[id]
    return positiveDocuments

def extractNeutralDocuments(documents,csvFile_rows):
    #function that removes all positive and negative sentiment documents

    neutralDocuments = copy.deepcopy(documents)

   #delete all documents that do not have neutral sentiment
    for i in range(0,len(csvFile_rows)):
        if csvFile_rows[i][2] == "Positive" or csvFile_rows[i][2] == "Negative":
            id=csvFile_rows[i][1]
            del neutralDocuments[id]

    return neutralDocuments

def getMedicReview(medication,doc):
    #this function deletes all documents that does not contain the supplied medic
ation
    docCopy = copy.deepcopy(doc)
    tempDocCopy = copy.deepcopy(doc)
    for key,value in docCopy.items():
        if not medication.upper() in value.upper():
```

```
            del tempDocCopy[key]
    return tempDocCopy

#Read the Document.csv file and store it into a dictionary.
with open("Documents.csv", "r") as f:
    lines = f.read().splitlines()
    temp = list(lines)
    documents={}


    #creating a dictionary of documents where the key is the documentID and the v
alue is the feedback posted by the user.
    for i in range(len(temp)):
        documents[temp[i].split("\",\"")[2]]=temp[i].split("\",\"")[0]

#Get sentiment data from SentimentData.csv file
with open('SentimentData.csv',"r") as csv_file:
    csv_reader = csv.reader(csv_file)
    csvFile_rows= list(csv_reader)

    #Get documents with negative sentiment
    negativeDocs = extractNegativeDocuments(documents,csvFile_rows)

    #Get documents with positive sentiment
    positiveDocs = extractPositiveDocuments(documents,csvFile_rows)

    #Get documents with neutral sentiment
    neutralDocs = extractNeutralDocuments(documents,csvFile_rows)

    #Get the name of medication names from MedicationConcept.txt file
    with open('MedicationConcept.txt',"r") as f:
        medication = f.read().split('\n')

        medicReview={}

        #For each medication retrieve the count of negative,positive and neutral
review
        for row in medication:
            negReview = len(getMedicReview(row,negativeDocs))
            posReview = len(getMedicReview(row,positiveDocs))
            neuReview = len(getMedicReview(row,neutralDocs))

            #Store the computed data as a dictionary
            medicReview[row]=str(negReview)+","+str(posReview)+","+str(neuReview)
```

```
#Export the data from medicReview dictionary into a file
w = csv.writer(open("MedicationReview.csv", "w"))
for key, val in medicReview.items():
    w.writerow([key,val])
```

Discussion

Since this question was open-ended, I decided to explore the effects of medication on patients. For this purpose, I used the sentiment analysis data for each post/document submitted by the user. To understand how effective a drug is, I decided to filter the given documents into negative, positive and neutral category. Further, for each medication I investigated its association with documents each category of documents.

The first chart is an example of stacked type of bar chart that displays the ratio of negative, positive and neutral reviews for each medication. Such types of charts are useful to quickly identify the effects (good and bad) of a given medication. However, they can be sometimes misleading since they do not show the number of reviews used to compute the analysis. For example, if there are only two reviews, a positive and a negative for a medication the chart would display an “alarming” output since in this case 50% of reviews are negative.

To overcome this drawback, the second chart displays the number of reviews/documents that were used to base the analysis. Furthermore, the chart is properly color-coded, similar to first chart that allows the viewer to quickly grasp the data. The third chart displays a consolidated view of types of review for each medication. This chart assists the analyst to compare performance of each medication in each negative, positive and neutral category.

From the business point of view, these charts help the analyst to identify medications that are having negative impact on the patients. This is crucial since these medications affect the health of a patient and may cause further complications if left unnoticed. This way the company can recall poorly performing medication from the market to prevent damage to its reputation.