Symbiosis Skills and Professional University
Kiwale, Pune

CAPSTONE PROJECT REPORT

On

TOPIC-

"Use of Artificial Intelligence in PharmacoVigilance for Social Networking sites"

Submitted by

Subash Nadar

PRN: 1901201012

PDG DS & AI

AY-2019-2020

Under the Guidance of

Faculty Mentor's Name: Dr. Ruby Jain

STUDENT DECLARATION AND ATTESTATION BY FACULTY MENTOR

This is to declare that this report has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

Signature 

Name: Subash Nadar

PRN: 1901201012

School/Program: SSPU, PGD DS & AI

Signature

Name of faculty mentor:

School/Program:

ACKNOWLEDGEMENTS

I would like to Thank SSOU for providing the necessary support in my pursuit to this capstone project. I would also like to express my gratitude to Dr. Ruby Jain for the guidance through this capstone project. Her valuable advice and constructive feedback have helped me to carry out the activities of the course in a timely manner

I am also grateful to Dr Ajanta Devi for providing me with the necessary support and guidance in selecting the topic and further helping with constructing the overall scope of the project. Her valuable inputs helped me with data analysis and report preparation

I would like to express my special Thanks to my Teachers for the guidance and support provided during the course

1. Mr Kumarsanjay Bhorekar – Machine Learning
2. Dr. Ruby Jain – Machine Learning
3. Ms Preeti Pandu – Business Intelligence

Finally, I would like to Thank my Family, Friends and Colleagues for encouraging me throughout my PGD program

With this project, I have gained wonderful experience in using Data Science for Business insights and helped me in doing a lot of research to unveil new findings

Contents

1. PLAN OF CAPSTONE PROJECT

i. Purpose of the Project

With an ever changing lifestyle and eating habits, there comes new type of diseases and viruses. And to keep humans safe from these newfound diseases and viruses, there is a need to discover new medicines and drugs on a regular basis. Though before any drug is released in the market, it must go over a Drug Discovery Lifecycle. In Pharmacology world, it is termed a Clinical Trials. A typical Clinical Trial phases consists of below four phases.

Once the Drug is approved by the regulatory authorities, it hits the market and doctors starts prescribing it. Now the thing with Clinical trials is that it is at max tested on some thousands of volunteers within specific geography, gender, age group etc. But when it is available in the market, it goes beyond the tested profiles and can show some side effects that were not expected and are adverse I nature.

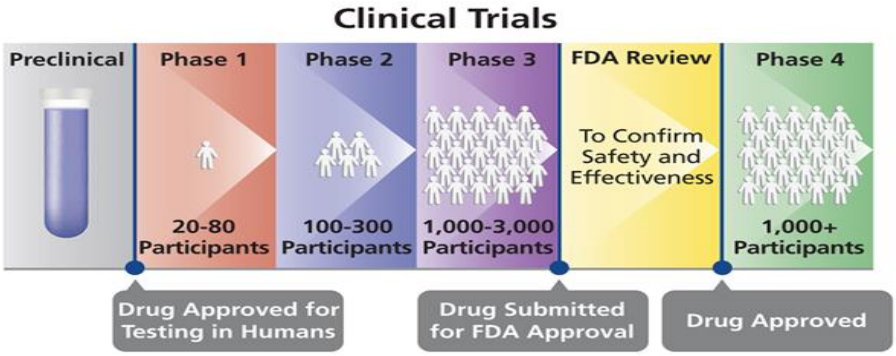

Fig. 1 Clinical Trial Phases

"Clinical trials are a way to test new methods of diagnosing, treating, or preventing health conditions. The goal is to determine whether something is both safe and effective."
- *www.healthline.com*

Whenever a patient takes a drug and has an unexpected side effect, which could be anything from minor problems like a runny nose to more serious ones that require hospitalization, pharma companies are legally required to report this information, known as an adverse event (AE), to the respective regulatory agencies (FDA for US, EMA for Europe, ICMR for India etc). This is called as ICSR (Individual Case Safety Report).

Within the pharmaceutical industry, AE reporting is a critical and time-consuming part of ensuring the safe and effective use of medicines by patients. Pharmacovigilance employees process a growing number of cases from sources as varied as patients' social media accounts to reports from investigators overseeing clinical trials. One of the Large Pharma company based in US has processed approximately 1.4 million AEs globally in 2019 alone for their Worldwide Safety organization. Looking forward across the industry, the volume of AEs is expected to increase by 20% annually.

ii. Problem statement

There are two problem statements:

a) Though ADR reporting is a very matured process for traditional channels like clinical trials and Customer care reporting, it is largely unexplored for reporting on social networking sites and hence they tend to be missed out i.e. **not reported**

b) With an ever increasing volume of such safety reports, Pharma companies are struggling to keep pace with managing it on a timely basis and hence there is a huge demand to **automate ADR reporting process**

2. OBJECTIVE OF THE PROJECT

i. General Objective

With increased digital penetration, Social Networking site has become a de facto place for most of the new generation population to go interact with like-minded people. This has become so integrated in every life that they are sharing all their pleasant as well as unpleasant experiences online, as it is accessible at their fingertip

This capstone project is aiming to explore social network reporting of ADRs for specific drug and build a framework that any Pharma company can use in their pursuit of automating the social network reporting of ADRs

3. INTRODUCTION

For the purpose of this project, we have extracted the data from twitter with reference for some of the drugs. These tweets were then used as input data set for the processing of the tweets and identifying the tweets Adverse event and subsequently extracting drug information for those

An adverse drug reaction (ADR) is an unwanted or harmful reaction experienced following the administration of a drug or combination of drugs under normal conditions of use and is suspected to be related to the drug. ADR (Adverse Drug Reaction) monitoring is an important activity for any Pharma company from drug safety perspective.

i. Technical Specification

Operating System	Windows 10 Pro
RAM	12 GB
Integrated Development Environment (IDE).	Jupyter Notebook
Coding Language	Python 3.7
Packages used for Data Processing	• Pandas *(for data manipulation and analysis)* • Wordcloud *(data visualization technique used for representing text data in which the size of each word indicates its frequency or importance)*
Packages used for Data Visualization	• Matplotlib *(plotting library for the Python programming language and its numerical mathematics extension NumPy)* • Seaborn *(data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics)*
Packages used for Text Processing (Natural Language Processing)	• Textblob *(for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more)* • Spacy *(library for Natural Language Processing in Python. It features NER, POS tagging, dependency parsing, word vectors and more)*

	• NLTK *(suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language)*
Packages used for Medical Terms Processing	• Med7 *(a transferable clinical natural language processing model for electronic health records, compatible with spaCy, for clinical named-entity recognition (NER) tasks)*

ii. Data Description

The extracted tweeter data has following information:

- Brand Name

  This field has name of the Drug Brands that are in consideration. Example – Strattera, Vyvanse, Adderall, Concerta etc
- ID_Drug

  This is a unique ID which identifies the specific Drug in consideration
- Generic Name

  This is the generic name of the Drug like atomoxetine (Strattera), lisdexamfetamine (Vyvanse), amphetamine and dextroamphetamine (Adderall) etc
- Review

  This is the most important field in this project. This includes Review, in free text form, by the tweeterrati. This review would contain any Adverse Reaction and related Drug information, which are vital for the final submission to the Regulatory Authority

Source of the Data set is https://www.twitter.com/

After loading the data and removing unwanted information, following Master Data Frame is created:

	Brand Name	ID_Drug	Generic Name	Review
0	Adderall	1	amphetamine and dextroamphetamine	"Hi everyone. I am 25 years old married with a...
1	Adderall	1	amphetamine and dextroamphetamine	"I'm a 31yr old female, 5'4, 115lbs. I was 1st...
2	Adderall	1	amphetamine and dextroamphetamine	"This drug was prescribed to after ritilin did...
3	Adderall	1	amphetamine and dextroamphetamine	"My face breaks out in itchy blotchy spots and...
4	Adderall	1	amphetamine and dextroamphetamine	"Adderall has improved my life drastically! So...

4. METHODOLOGY AND ANALYSIS

i. Solution Approach

Research Methodology for this project is depicted in the flow diagram below:

Fig. 2 Research Methodology

1. **Identify the Subject –** First step in the process is to identify the name of the drug which will be the subject of this study
2. **Identify posts on web –** Second step is to identify the posts on social networking sites like Facebook or Twitter or any other relevant organization, related to the subject
3. **Sentiment Analysis –** Derive sentiment of the post, i.e. identify if adverse event or positive outcome
4. **Extract relevant information –** Once post is identified, need to extract the relevant information from it like:
    a. Drug
    b. Dosage
    c. Form
    d. Route
    e. Frequency
    f. Duration
5. **Adverse Event Report –** If adverse event identified, create Adverse Reaction Report and share it with the respective Pharmaceutical Company or Regulatory Authority, for further reporting

ii. Data Cleaning

Data cleaning is an important step in the Data Analysis exercise to ensures noises from the data is removed which helps in achieving near correct solution

Step 1: Remove unwanted columns that may not add any value to the analysis

After loading the data, below data frame is created.

```
data = pd.read_csv("AHDH.csv", sep=",")
data.head()
```

	Unnamed: 0	Brand Name	ID_Drug	Generic Name	Review	Rating	Unnamed: 6
0	1	Adderall	1	amphetamine and dextroamphetamine	"Hi everyone. I am 25 years old married with a...	10	NaN
1	2	Adderall	1	amphetamine and dextroamphetamine	"I'm a 31yr old female, 5'4, 115lbs. I was 1st...	9	NaN
2	3	Adderall	1	amphetamine and dextroamphetamine	"This drug was prescribed to after ritilin did...	7	NaN
3	4	Adderall	1	amphetamine and dextroamphetamine	"My face breaks out in itchy blotchy spots and...	7	NaN
4	5	Adderall	1	amphetamine and dextroamphetamine	"Adderall has improved my life drastically! So...	10	NaN

Now let's remove the *unwanted columns* from the data set and create a Master Data Frame

```
data.drop(['Unnamed: 6', 'Unnamed: 0', 'Rating'],axis=1, inplace=True)
data.head()
```

	Brand Name	ID_Drug	Generic Name	Review
0	Adderall	1	amphetamine and dextroamphetamine	"Hi everyone. I am 25 years old married with a...
1	Adderall	1	amphetamine and dextroamphetamine	"I'm a 31yr old female, 5'4, 115lbs. I was 1st...
2	Adderall	1	amphetamine and dextroamphetamine	"This drug was prescribed to after ritilin did...
3	Adderall	1	amphetamine and dextroamphetamine	"My face breaks out in itchy blotchy spots and...
4	Adderall	1	amphetamine and dextroamphetamine	"Adderall has improved my life drastically! So...

Step 2: Remove Null data, if any

```
data['Review'].isnull().sum()
```

```
0
```

No Null data found in the data set

Step 3: Check for Spaces in the data

```
spaces = []
for i, x in enumerate(data['Review']):
    if type(x) == str:
        if x.isspace():
            spaces.append(i)

print(len(spaces), 'spaces in index: ', spaces)
```

```
0 spaces in index:  []
```

Step 4: Remove Duplicate Tweets

```
data.drop_duplicates('Review', inplace=True)
data.reset_index(drop = True, inplace=True)
```

iii. Exploratory Data Analysis

Dimension of the Data:

```
data.shape
```

```
(1231, 4)
```

The dataset contains 4 columns and 1231 number of rows (i.e. reviews from Tweet)

Let's visualize the distribution of D=Brand Names (Drug Names) in the dataset

```
# Count each Brand occurance in the data
brand_data = data['Brand Name'].value_counts()
```

```
brand_data.values
```

```
array([399, 198, 190, 152, 116,  55,  46,  44,  31], dtype=int64)
```

```
brand_data.index
```

```
Index(['Vyvanse', 'Strattera', 'Concerta', 'Adderall', 'Adderall XR ',
       'Ritalin', 'Dexedrine', 'Daytrana', 'Metadate CD'],
      dtype='object')
```

```
#Visualizing the spread of Brands
plt.bar(brand_data.index, brand_data.values, align='edge')
plt.xticks(rotation=45)
plt.title('Brands Distribution')
plt.figure(figsize=(30,22), dpi=80)
plt.show()
```

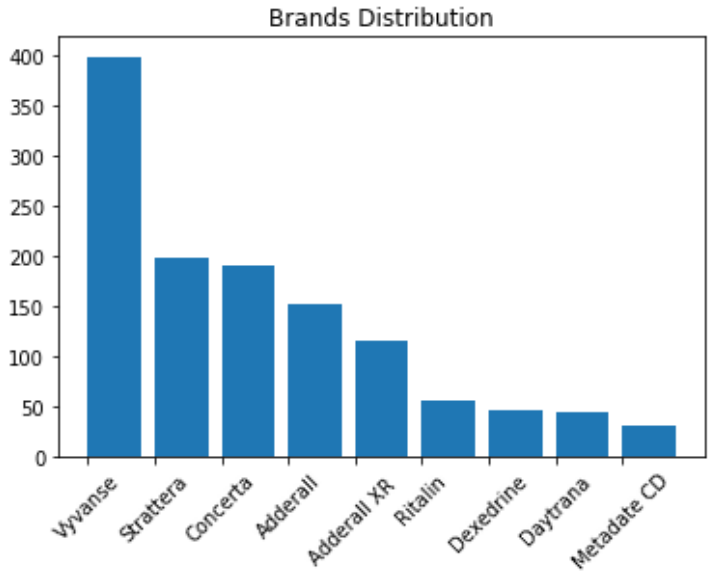

Fig. 3 Brand Distribution

From the graph above, we can see that the dataset contains highest number of Tweets for **Vyvanse** followed by **Strattera** and so on. Least review found for Metadate CD

a) Expand the Contraction words

Contractions are the shorthand versions of words like "don't" for "do not" and "how'll" for "how will". These are commonly used in casual chatting, including posts in social networking sites, to reduce the writing time of words.

For better analysis of the reviews, we need to expand these contractions

```
#Let's create contraction dictionary

contractions_dict = { "ain't": "are not","'s":" is","aren't": "are not",
                     "can't": "cannot","can't've": "cannot have",
                     "'cause": "because","could've": "could have","couldn't": "could not",
                     "couldn't've": "could not have", "didn't": "did not","doesn't": "does not",
                     "don't": "do not","hadn't": "had not","hadn't've": "had not have",
                     "hasn't": "has not","haven't": "have not","he'd": "he would",
                     "he'd've": "he would have","he'll": "he will", "he'll've": "he will have",
                     "how'd": "how did","how'd'y": "how do you","how'll": "how will",
                     "I'd": "I would", "I'd've": "I would have","I'll": "I will",
                     "I'll've": "I will have","I'm": "I am","I've": "I have", "isn't": "is not",
                     "it'd": "it would","it'd've": "it would have","it'll": "it will",
                     "it'll've": "it will have", "let's": "let us","ma'am": "madam",
                     "mayn't": "may not","might've": "might have","mightn't": "might not",
                     "mightn't've": "might not have","must've": "must have","mustn't": "must not",
                     "mustn't've": "must not have", "needn't": "need not",
                     "needn't've": "need not have","o'clock": "of the clock","oughtn't": "ought not",
                     "oughtn't've": "ought not have","shan't": "shall not","sha'n't": "shall not",
                     "shan't've": "shall not have","she'd": "she would","she'd've": "she would have",
                     "she'll": "she will", "she'll've": "she will have","should've": "should have",
                     "shouldn't": "should not", "shouldn't've": "should not have","so've": "so have",
                     "that'd": "that would","that'd've": "that would have", "there'd": "there would",
                     "there'd've": "there would have", "they'd": "they would",
                     "they'd've": "they would have","they'll": "they will",
                     "they'll've": "they will have", "they're": "they are","they've": "they have",
                     "to've": "to have","wasn't": "was not","we'd": "we would",
                     "we'd've": "we would have","we'll": "we will","we'll've": "we will have",
                     "we're": "we are","we've": "we have", "weren't": "were not","what'll": "what will",
                     "what'll've": "what will have","what're": "what are", "what've": "what have",
                     "when've": "when have","where'd": "where did", "where've": "where have",
                     "who'll": "who will","who'll've": "who will have","who've": "who have",
                     "why've": "why have","will've": "will have","won't": "will not",
                     "won't've": "will not have", "would've": "would have","wouldn't": "would not",
                     "wouldn't've": "would not have","y'all": "you all", "y'all'd": "you all would",
                     "y'all'd've": "you all would have","y'all're": "you all are",
                     "y'all've": "you all have", "you'd": "you would","you'd've": "you would have",
                     "you'll": "you will","you'll've": "you will have", "you're": "you are",
                     "you've": "you have"}
```

```
# Regular expression for finding contractions
contractions_re=re.compile('(%s)' % '|'.join(contractions_dict.keys()))
```

```
# Function for expanding contractions
def expand_contractions(text,contractions_dict=contractions_dict):
  def replace(match):
    return contractions_dict[match.group(0)]
  return contractions_re.sub(replace, text)
```

above code logic for contraction is referred to from the blog site:
source: https://www.analyticsvidhya.com/blog/author/abhishek-shrm/

Data before applying Contractions:

```
data['Review'][0:5]
```

```
0    "Hi everyone. I am 25 years old married with a...
1    "I'm a 31yr old female, 5'4, 115lbs. I was 1st...
2    "This drug was prescribed to after ritilin did...
3    "My face breaks out in itchy blotchy spots and...
4    "Adderall has improved my life drastically! So...
Name: Review, dtype: object
```

Data after applying Contractions:

```
# Modify the review by Expanding Contractions
data['Rev_mod']=data['Review'].apply(lambda x:expand_contractions(x))
```

```
data['Rev_mod'][0:5]
```

```
0    "Hi everyone. I am 25 years old married with a...
1    "I am a 31yr old female, 5'4, 115lbs. I was 1s...
2    "This drug was prescribed to after ritilin did...
3    "My face breaks out in itchy blotchy spots and...
4    "Adderall has improved my life drastically! So...
Name: Rev_mod, dtype: object
```

In the above code, you can see that the short form like “I’m” is converted to “I am”

b) Convert all text to Lower case

I n general Python is case sensitive, and all NLP models follow this principle. Hence any text analysis package would treat words like 'Student' and 'student' differently.

Since there is no difference in the meaning of both the words in a statement, we can convert all the texts to lower case

```
data['Rev_mod'] = [w.lower() for w in data['Rev_mod']]
```

```
data['Rev_mod'][0:5]
```

```
0    "hi everyone. i am 25 years old married with a...
1    "i am a 31yr old female, 5'4, 115lbs. i was 1s...
2    "this drug was prescribed to after ritilin did...
3    "my face breaks out in itchy blotchy spots and...
4    "adderall has improved my life drastically! so...
Name: Rev_mod, dtype: object
```

c) Remove Punctuations

Punctuations are the marks in English like commas, hyphens, full stops, etc. These add to the volume of the text for processing, even though not used in the Text analysis exercise

```
data['Rev_mod']=data['Rev_mod'].apply(lambda x: re.sub('[%s]' % re.escape(string.punctuation), '', x))
```

```
data['Rev_mod'][0:5]
```

```
0    hi everyone i am 25 years old married with a 6...
1    i am a 31yr old female 54 115lbs i was 1st pre...
2    this drug was prescribed to after ritilin did ...
3    my face breaks out in itchy blotchy spots and ...
4    adderall has improved my life drastically so d...
Name: Rev_mod, dtype: object
```

d) Lemmatization with stopwords removal

Lemmatisation in linguistics is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's lemma, or dictionary form.

-*Wikipedia*

Stopwords are commonly used word in a sentence (such as “the”, “a”, “an”, “in”). These are useful in a conversation but adds no value to the Text analysis. These would also to take up space in the database or taking up valuable processing time. H=We should hence remove these stopwords from processing

```
data['Rev_mod']=data['Rev_mod'].apply(lambda x: ' '.join([token.lemma_ for token in list(nlp(x))
                                                          if (token.is_stop==False)]))
```

```
data[['Review', 'Rev_mod']].head()
```

	Review	Rev_mod
0	"Hi everyone. I am 25 years old married with a...	hi 25 year old marry 6 month old son diagnose ...
1	"I'm a 31yr old female, 5'4, 115lbs. I was 1st...	31yr old female 54 115lbs 1st prescribe addera...
2	"This drug was prescribed to after ritilin did...	drug prescribe ritilin work medication help ad...
3	"My face breaks out in itchy blotchy spots and...	face break itchy blotchy spot sleep
4	"Adderall has improved my life drastically! So...	adderall improve life drastically drastically ...

iv. Sentiment Analysis

Next step is to identify reviews that may have any Adverse Events reported. This is the very core of this project. These are typically any side effects that the customer is not happy with

We may use Sentiment Analysis algorithm to identify such reporting, which can be further used to extract Event related information to be used for reporting to regulatory authorities

What is Sentiment Analysis:

Sentiment analysis (also known as opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion is *positive, negative, or neutral*. For instance, at emotional states such as "angry", "sad", and "happy".

https://en.wikipedia.org/wiki/Sentiment_analysis

Sentiment Analysis with TextBlob:

TextBlob aims to provide access to common text-processing operations through a familiar interface. You can treat TextBlob objects as if they were Python strings that learned how to do Natural Language Processing. The sentiment property returns a namedtuple of the form Sentiment (polarity, subjectivity). The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

https://textblob.readthedocs.io/en/dev/quickstart.html

Let's check the Sentiment Polarity of the reviews & Subjectivity of respective Sentiments

```
reviews = data["Review"]

for i in range(0,len(reviews)):
    reviewsBlob = TextBlob(reviews[i])
    sentiment = reviewsBlob.sentiment.polarity
    subjectivity = reviewsBlob.sentiment.subjectivity
    data.set_value(i, 'Sentiment', sentiment)
    if sentiment < -0.00:
        SentimentClass = 'Negative'
        data.set_value(i, 'SentimentClass', SentimentClass )
    elif sentiment > 0.15:
        SentimentClass = 'Positive'
        data.set_value(i, 'SentimentClass', SentimentClass )
    else:
        SentimentClass = 'Neutral'
        data.set_value(i, 'SentimentClass', SentimentClass )

    data.set_value(i, 'Subjectivity', subjectivity)
    if subjectivity < 0.45:
        SubjecivityClass = 'Objective'
        data.set_value(i, 'SubjecivityClass', SubjecivityClass )
    elif sentiment > 0.55:
        SubjecivityClass = 'Subjective'
        data.set_value(i, 'SubjecivityClass', SubjecivityClass )
    else:
        SubjecivityClass = 'Neutral'
        data.set_value(i, 'SubjecivityClass', SubjecivityClass )

data.head()
```

	Brand Name	ID_Drug	Generic Name	Review	Rev_mod	Sentiment	SentimentClass	Subjectivity	SubjecivityClass
0	Adderall	1	amphetamine and dextroamphetamine	"Hi everyone. I am 25 years old married with a...	hi 25 year old marry 6 month old son diagnose ...	-0.200000	Negative	0.530303	Neutral
1	Adderall	1	amphetamine and dextroamphetamine	"I'm a 31yr old female, 5'4, 115lbs. I was 1st...	31yr old female 54 115lbs 1st prescribe addera...	0.071429	Neutral	0.357143	Objective
2	Adderall	1	amphetamine and dextroamphetamine	"This drug was prescribed to after ritilin did...	drug prescribe ritilin work medication help ad...	0.443182	Positive	0.702273	Neutral
3	Adderall	1	amphetamine and dextroamphetamine	"My face breaks out in itchy blotchy spots and...	face break itchy blotchy spot sleep	0.000000	Neutral	0.000000	Objective
4	Adderall	1	amphetamine and dextroamphetamine	"Adderall has improved my life drastically! So...	adderall improve life drastically drastically ...	-0.068750	Negative	0.499851	Neutral

We can see average Polarity for each Drug:

```
product_polarity_sorted=pd.DataFrame(data.groupby('Brand Name')['Sentiment'].mean().sort_values(ascending=True))

plt.figure(figsize=(16,8))
plt.xlabel('Sentiments')
plt.ylabel('Drug Name')
plt.title('Polarity of Drug Reviews')
polarity_graph=plt.barh(np.arange(len(product_polarity_sorted.index)),product_polarity_sorted['Sentiment'],color='teal',)

# Writing Drug names on bar
for bar,product in zip(polarity_graph,product_polarity_sorted.index):
  plt.text(0.005,bar.get_y()+bar.get_width(),'{}'.format(product),verticalalignment='bottom', horizontalalignment='left',
           fontsize=12,color='white',style='italic')

# Writing polarity values on graph
for bar,polarity in zip(polarity_graph,product_polarity_sorted['Sentiment']):
  plt.text(bar.get_width()+0.001,bar.get_y()+bar.get_width(),'%.3f'%polarity,va='bottom',fontsize=11,color='black')

plt.yticks([])
plt.show()
```

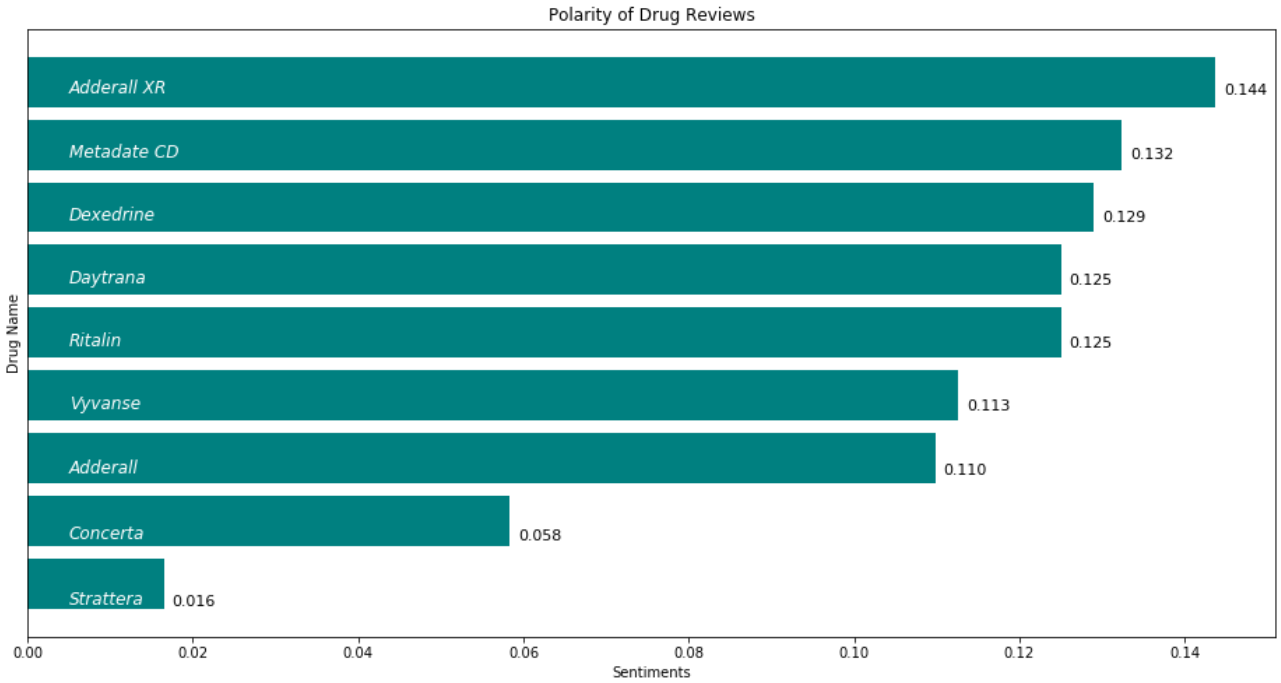

Fig. 4 Average Polarity for each Drug

From the graph above, it is evident that Adderall XR has the highest positivity in the review and Strattera has lowest positivity

Let's see the spread of the Sentiments across the data set

```
data['SentimentClass'].value_counts()
```

```
Neutral     447
Positive    438
Negative    346
Name: SentimentClass, dtype: int64
```

```
#Visualizing the spread of Sentiments
review_data = data['SentimentClass'].value_counts()
plt.pie(review_data, autopct='%1.1f%%', shadow=True,
        labels=['Positive Sentiments','Neutral Sentiments', 'Negative Sentiments'])
plt.title('Spread of Sentiment Polarity');
plt.show()
```

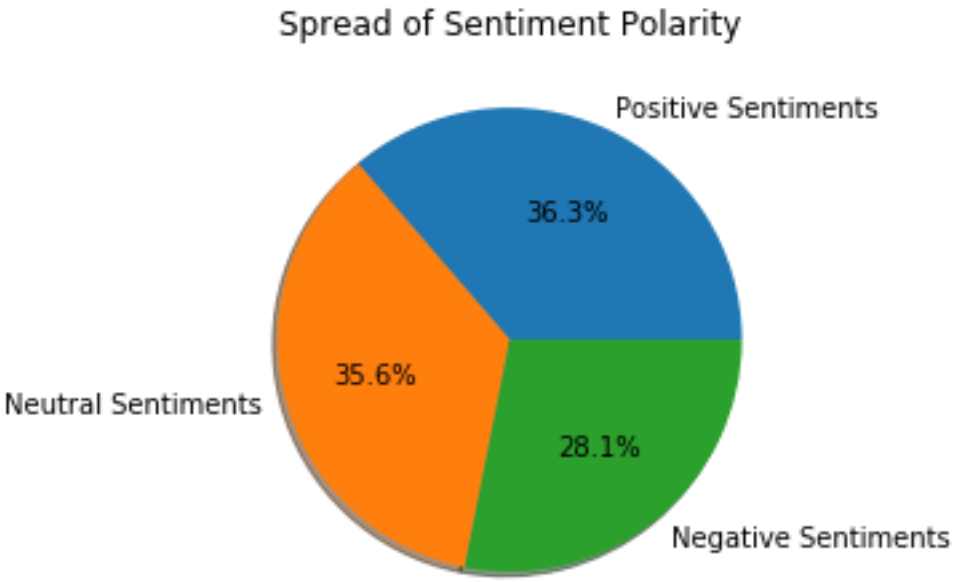

Fig. 5 Spread of Sentiment

From the graph above, we can see that 28 % of Tweets are having Negative Sentiments, i.e. mentioning of Adverse Events. 36% posted positive sentiments around the Drug and 36% were neutral in their comments

We can analyse each Drug to see which one has received more Negative Sentiments than Positive in the reviews. This would give an indication which Drug needs to review its Drug side effects and improve upon them

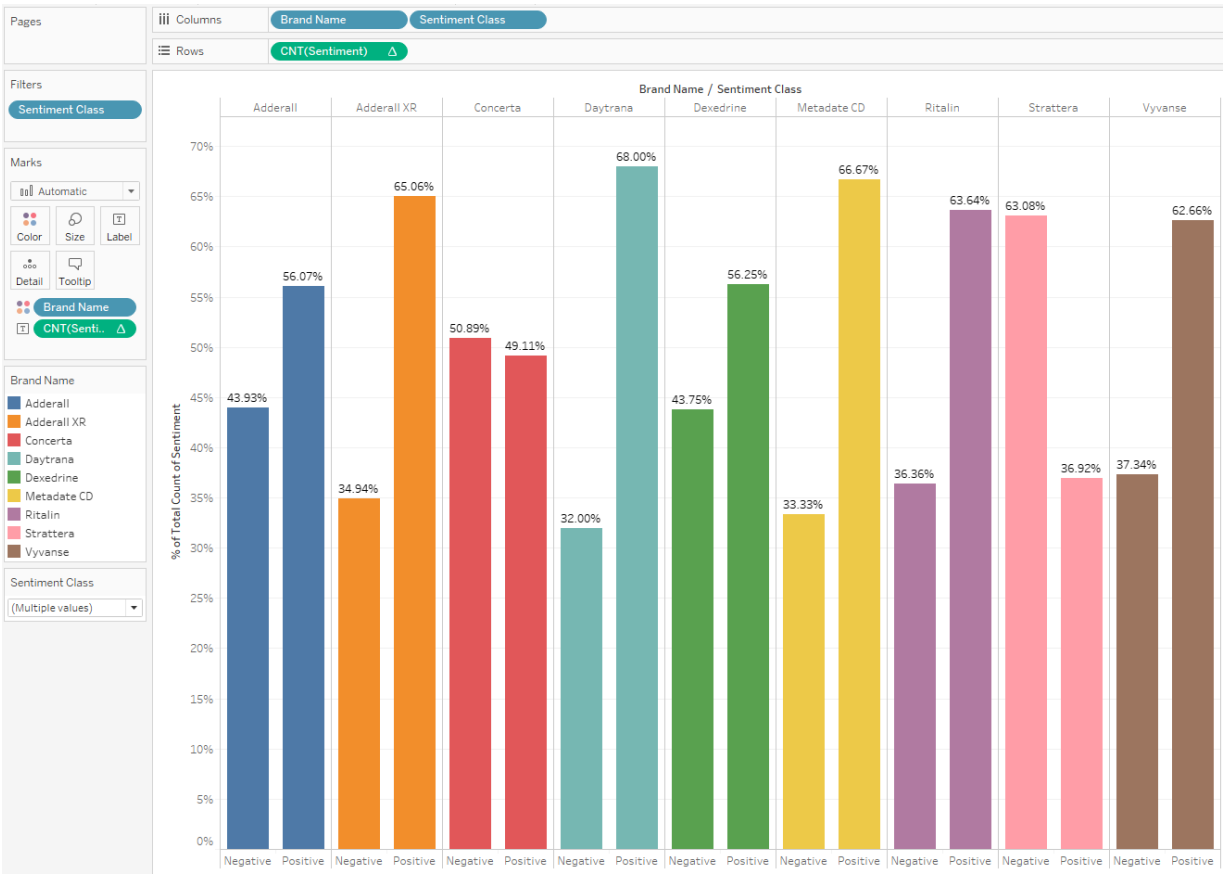

Fig.6 Spread of Sentiment for each Drug

From the above graph, we can see that Concerta and Strattera are the only Drugs which has more Negative Sentiments than Positive Sentiments.

Let's see Word cloud (most commonly used terms) for each of the Sentiments

Wordcloud – Positive Sentiment:

```
#Word cloud of Positive Sentiments
from wordcloud import WordCloud
wordcloud = WordCloud(width=1600, height=800,max_font_size=200).generate(str(data[data['SentimentClass']=='Positive']['Review']))
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title('Word Cloud - Positive Sentiments',fontsize=20,fontweight='bold')
plt.show()
```

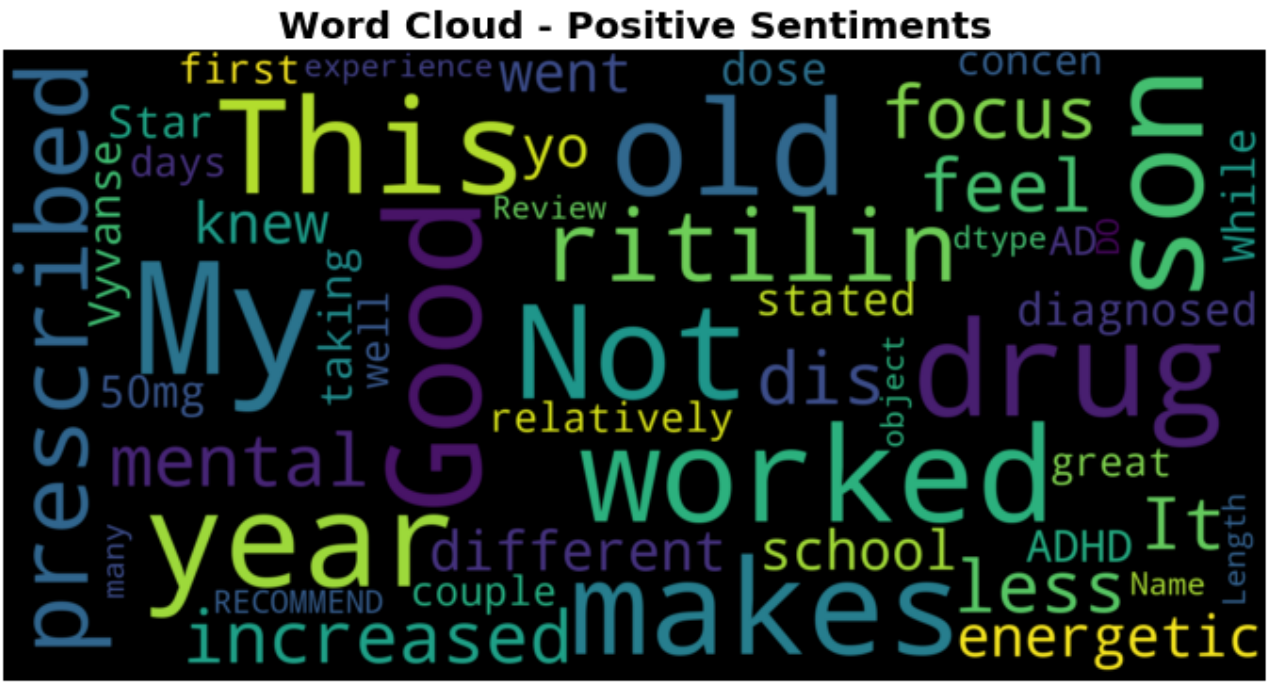

Fig. 7 Positive Sentiments Word Cloud

Wordcloud – Negative Sentiment:

```
#Word cloud of Negative Sentiments
wordcloud = WordCloud(width=1600, height=800,max_font_size=200).generate(str(data[data['SentimentClass']=='Negative']['Review']))
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title('Word Cloud - Negative Sentiments',fontsize=20,fontweight='bold')
plt.show()
```

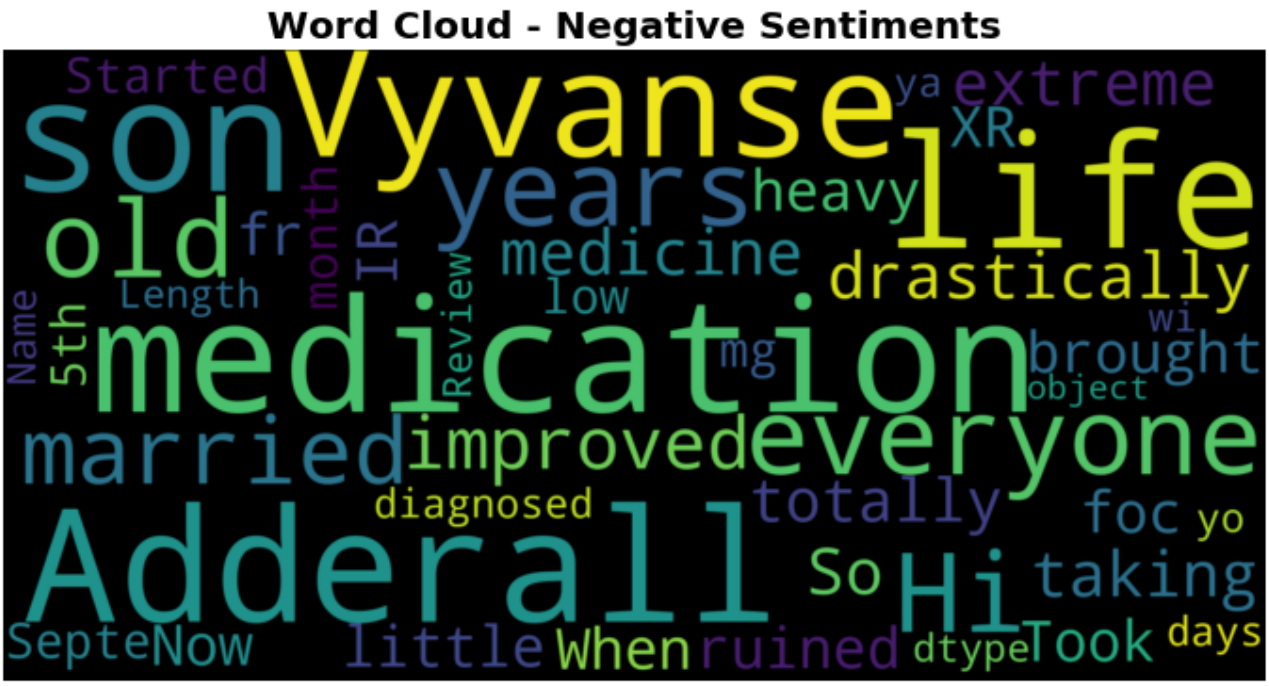

Fig. 8 Negative Sentiments Word Cloud

Wordcloud – Neutral Sentiment:

```
#Word cloud of Neutral Sentiments
wordcloud = WordCloud(width=1600, height=800,max_font_size=200).generate(str(data[data['SentimentClass']=='Neutral']['Review']))
plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title('Word Cloud - Neutral Sentiments',fontsize=20,fontweight='bold')
plt.show()
```

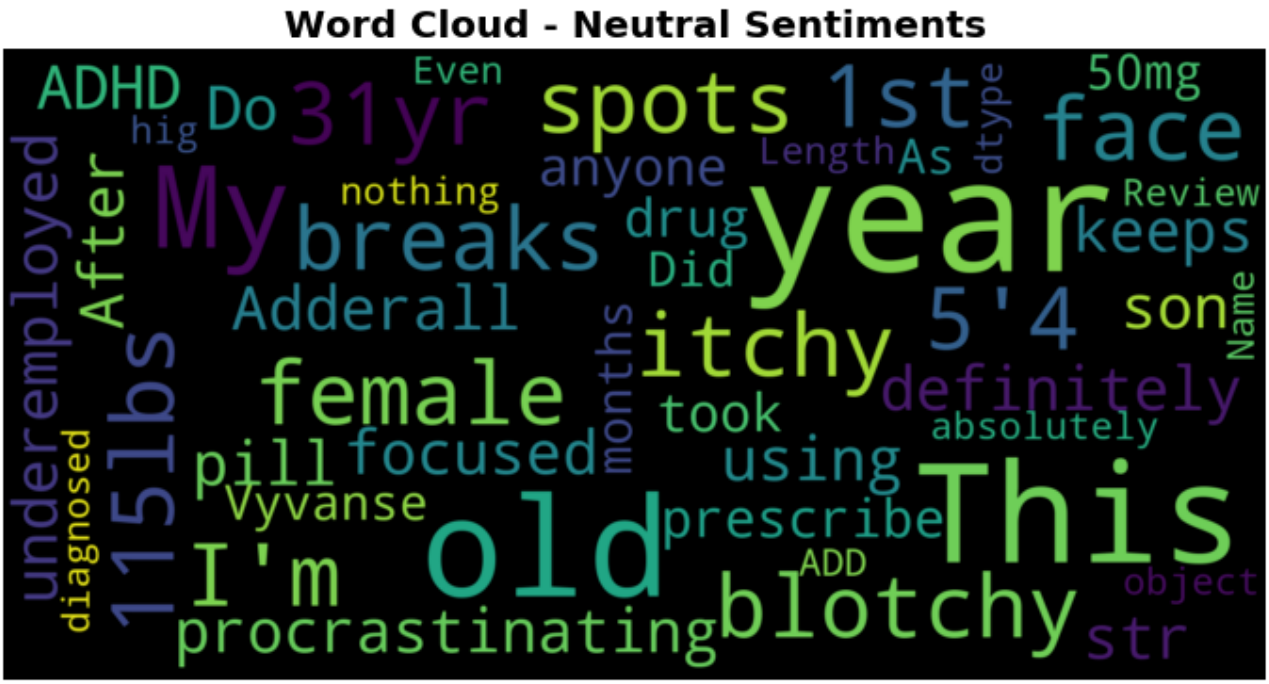

Fig. 9 Neutral Sentiments Word Cloud

v. Adverse Reaction Report

All the reviews identified with Negative Polarity is considered to be having some form of reported Adverse Event. We need to select these for further processing and subsequent submission to Regulatory Authority

Step 1: Select all the reviews with Sentiment Class as Negative

```
adr_data = data[data['SentimentClass']=='Negative']
adr_data = adr_data[['Brand Name', 'Generic Name', 'Review']]
adr_data.head()
```

	Brand Name	Generic Name	Review
0	Adderall	amphetamine and dextroamphetamine	"Hi everyone. I am 25 years old married with a...
4	Adderall	amphetamine and dextroamphetamine	"Adderall has improved my life drastically! So...
6	Adderall	amphetamine and dextroamphetamine	"I do not like taking medicine but had extreme...
9	Adderall	amphetamine and dextroamphetamine	"Adderall has totally brought my life into foc...
13	Adderall	amphetamine and dextroamphetamine	"Took XR but IR is a little heavy for me. Now ...

Now reset the Index

```
adr_data.reset_index(inplace = True, drop = True)
adr_data.head()
```

	Brand Name	Generic Name	Review
0	Adderall	amphetamine and dextroamphetamine	"Hi everyone. I am 25 years old married with a...
1	Adderall	amphetamine and dextroamphetamine	"Adderall has improved my life drastically! So...
2	Adderall	amphetamine and dextroamphetamine	"I do not like taking medicine but had extreme...
3	Adderall	amphetamine and dextroamphetamine	"Adderall has totally brought my life into foc...
4	Adderall	amphetamine and dextroamphetamine	"Took XR but IR is a little heavy for me. Now ...

Step 2: Let's see distribution of Brands with Negative Sentiments

```
# Count each Brand occurance in the data
adr_brand_data = adr_data['Brand Name'].value_counts()
```

```
#Visualizing the spread of Brands
plt.bar(adr_brand_data.index, adr_brand_data.values, align='edge')
plt.xticks(rotation=45)
plt.title('Brands Distribution for Adverse Events')
plt.figure(figsize=(30,22), dpi=80)
plt.show()
```

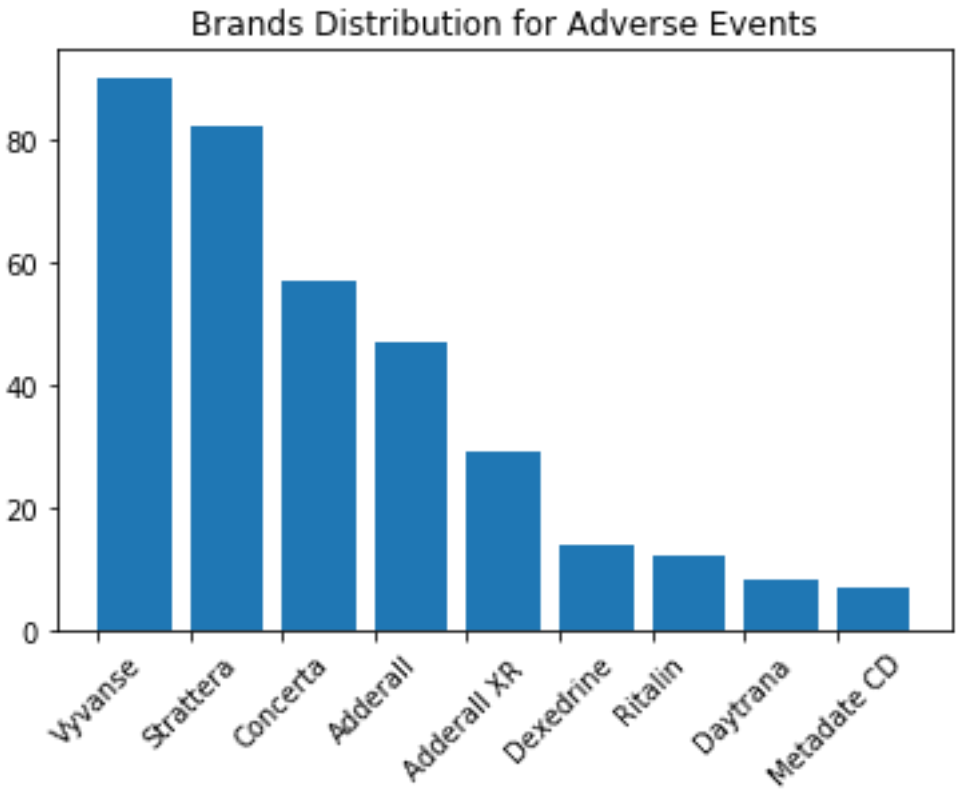

Fig. 10 Brand Distribution for Adverse Events

We see that Strattera has more Negative Sentiments compared to Vyvanse, which has more tweets

Step 3: Extract relevant Drug information from the review comments

Drug related information form an important set of data that is required to be sent to regulatory authorities as a part of Adverse Event Reporting (ADR)

We will use the **med7** package for this purpose

This repository is a transferable clinical natural language processing model for electronic health records, compatible with spaCy, for clinical named-entity recognition (NER) tasks. The en_core_med7_lg model is trained on MIMIC-III free-text electronic health records and is able to recognise 7

https://github.com/kormilitzin/med7

Various Categories/Tables that this package is able to extract are:

Label	Concept	Description
DOSAGE	1-2, sliding scale, taper, bolus, thirty (30) ml	The total amount of a drug administered
DRUG	aspirin, lisinopril, prednisone, vitamin b, Elegy!	Generic or brand name of the medication
DURATION	for 3 days, 7 days, chronic, x5 days, for five more days	The length &time that the drug was prescribed for
FORM	tablet, capsule, solution, puff, adhesive patch, disk with device	A particular configuration of the drug which it is marketed for use
FREQUENCY	once a day, b.i.d., pm, q6h, hs, every six (6) hours as needed	The dosage regimen at which the medication should be administered
ROUTE	iv, p.a. (by mouth), gtt, nasal canula, injection,	The path by which the drug N taken into the body
STRENGTH	15mg, 100 unit/ml, 50mg/2m1, 0.05%, 25-50mg	The amount of drug in a given dosage

Load all the libraries and sample any text extraction to see med7 extraction capabilities

```
import spacy
import en_core_med7_lg
import en_core_web_sm

med7 = en_core_med7_lg.load()
nlp = en_core_web_sm.load()

# create distinct colours for labels
col_dict = {}
seven_colours = ['#e6194B', '#3cb44b', '#ffe119', '#ffd8b1', '#f58231', '#f032e6', '#42d4f4']
for label, colour in zip(med7.pipe_labels['ner'], seven_colours):
    col_dict[label] = colour

options = {'ents': med7.pipe_labels['ner'], 'colors':col_dict}
```

```
text = '''A patient was prescribed Magnesium hydroxide, aimovig & repatha 400mg/5ml
suspension PO of total 30ml bid for the next 5 days.'''

doc = med7(text)

spacy.displacy.render(doc, style='ent', jupyter=True, options=options)
```

We see that med7 package is able to extract the relevant Drug information from the sample text processed

Now let's extract the Drug information from each of the Tweeter review

```
adr_data = adr_data.astype('object')
reviews = adr_data['Review']

for i in range(0,len(reviews)):
    doc = med7(reviews[i])
    drug_detail = [(ent.label_, ent.text) for ent in doc.ents]
    adr_data.set_value(i, 'Drug Detail', drug_detail)
```

```
adr_data.head()
```

	Brand Name	Generic Name	Review	Drug Detail
0	Adderall	amphetamine and dextroamphetamine	hi everyone i am 25 years old married with a 6...	[(DRUG, adderall), (STRENGTH, 30mg), (ROUTE, i...
1	Adderall	amphetamine and dextroamphetamine	adderall has improved my life drastically so d...	[(DRUG, adderall)]
2	Adderall	amphetamine and dextroamphetamine	adderall has totally brought my life into focu...	[(DRUG, adderall)]
3	Adderall	amphetamine and dextroamphetamine	i have been taking adderall for two years and ...	[(DRUG, adderall), (DURATION, for two years), ...
4	Adderall	amphetamine and dextroamphetamine	when i was diagnosed with adhd in the 3rd grad...	[(DRUG, adderall), (DRUG, adderall)]

Here we have the Drug details extracted from the Review

We now have the Adverse Reaction Report ready to be submitted for Regulatory Authority, after converting it to the ICH recommended format of E2BR2

Note that the formatting to E2BR2 is not in scope of this project

5. RESULT & DISCUSSION

During the Sentiment Analysis of the data, we found that approximately 28% of the review in the given data had negative sentiments. Reporters (Tweeterrati) were unhappy with the Drug in consideration and were reporting an Adverse Event.

A larger portion of reviews, over 36%, were with positive sentiments and remaining Neutral. It is hence important to have a very efficient Sentiment Analysis in place so that we are not reporting any incorrect event (on-adverse)

A quick analysis of the Adverse event reviews suggests that, it contains highest number of Tweets for Vyvanse followed by Strattera, even though Strattera had the highest number of reviews in the original data set. Least review found for Metadate CD. With this, we can see the trend in the adverse reaction reporting for various drug over a period of time.

Extraction of the Drug detail from the review suggests that information related to Drugs are also easily shared by the customers and hence it is important to have a robust medical extraction package to do the job well

For Live analysis of the Twitter feeds, we can subscribe for paid API access and integrate the solution with it

6. FUTURE SCOPE

We presented the identification of the adverse events from social network posts and subsequent extraction of the relevant Drug information from them before creating an Adverse Events Report

The scope can be extended to extract **Live feed** of the Tweets as it happens (posted) using the Tweets API provided by https://developer.twitter.com/ site and integrating it with the proposed model.

Another feature that can be added is creating a **Python executable program** (using package like Tkinter) so that this program can be used by any Business users from PharmacoVigilance space and get the report in Business user format

Scope of this capstone project was limited to extracting Drug details from the post. Other relevant critical information that is required from Regulatory Authority submission perspective are Patient Detail, Reporter Detail and Adverse Event information

Temporal Sequencing is another critical feature that can be added in future scope, which will help focus on extracting current events and not historical events. This ensures only current relevant conditions are reported and not the historical medical conditions

7. CONCLUSION

To conclude, basis data analysis was performed on a limited set of data to identify any adverse events reported in social networking like, twitter. During analysis, several statistical important features were explored and visualized. All the free package available in the market has been able to do a good job in identifying the adverse event and related drug information