# DRUG SENTIMENT ANALYSIS & RECOMMENDATION

EXCELR
Raising Excellence

## Presented by group 3


1. Benna Rose
2. Rutuja Hande
3. Tanmay Bhor
4. Surya Chaitanya
5. Ajith Singh
6. Vivek Gangurde
7. Vaibhav Mane

# Project Overview

The objective of this model is to do the sentiment analysis based on review and recommend drug

## Business Objective:

This is a sample dataset which consists of 161297 drug name, condition reviews and ratings from different patients and our goal is to examine how patients are feeling using the drugs their positive and negative experiences so that we can recommend him a suitable drug. By analyzing the reviews, we can understand the drug effectiveness and its side effects.

The dataset provides patient reviews on specific drugs along with related conditions and a 10 star patient rating reflecting overall patient satisfaction.

So in this dataset, we can see many patients conditions but we will focus only on the below, classify the below conditions from the patients reviews

a. Depression

c. High Blood Pressure

d. Diabetes, Type 2

# Features

Attribute Information:
1. DrugName (categorical): name of drug
2. condition (categorical): name of condition
3. review (text): patient review
4. rating (numerical): 10 star patient rating
5. date (date): date of review entry
6. usefulCount (numerical): number of users who found review useful

# Project process

| TIME | Week 1 | Week 2 | Week 3 | Week 4 |
|---|---|---|---|---|
| Task name | EDA | TEXT PROCESSING | MODEL BUILDING | DEPLOYMENT |

# EDA

Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.

01

## Step one

understood the  objective and checked data's datatypes

02

## Step two

did descriptive analysis

03

## Step three

made visualizations

# 1. changed date column to date-time

```
# Convert date to datetime
df['date'] = pd.to_datetime(df['date'])
```

```
# Select rows where the condition is depression, high blood pressure, or ty
condition_list = ['depression', 'high blood pressure', 'diabetes, type 2']
df_selected = df[df['condition'].isin(condition_list)]

# Save the resulting dataframe to a CSV file
df_selected.to_csv('drug_data.csv', index=False)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11549 entries, 0 to 11548
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   UniqueID     11549 non-null  int64
 1   drugName     11549 non-null  object
 2   condition    11549 non-null  object
 3   review       11549 non-null  object
 4   rating       11549 non-null  float64
 5   date         11549 non-null  datetime64[ns]
 6   usefulCount  11549 non-null  float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 631.7+ KB
```

# 2.renamed unnamed column to uniqueID

```
# renaming the unknown  column to UniqueID
df= df.rename({'unnamed: 0': 'UniqueID'}, axis=1)
```

# EDA

```
df.shape
```

```
(11549, 7)
```

```
[ ]  # checking duplicate values
     df.duplicated().sum()
```

```
0
```

```
#check null values in dataset
df.isnull().sum()
```

```
UniqueID      0
drugName      0
condition     0
review        0
rating        0
date          0
usefulCount   0
dtype: int64
```

total 11549 patients
zero duplicate values
no null values

EDA

# Statistical Analysis of all features

```
# statistical values for all column
df.describe(include= 'all')
```

| | UniqueID | drugName | condition | review | rating | date | usefulCount |
|---|---|---|---|---|---|---|---|
| **count** | 11549.000000 | 11549 | 11549 | 11549 | 11549.000000 | 11549 | 11549.000000 |
| **unique** | NaN | 323 | 3 | 8718 | NaN | 3058 | NaN |
| **top** | NaN | Bupropion | depression | "Good" | NaN | 2016-03-03 00:00:00 | NaN |
| **freq** | NaN | 462 | 7501 | 3 | NaN | 22 | NaN |
| **first** | NaN | NaN | NaN | NaN | NaN | 2008-02-25 00:00:00 | NaN |
| **last** | NaN | NaN | NaN | NaN | NaN | 2017-12-11 00:00:00 | NaN |
| **mean** | 125493.346437 | NaN | NaN | NaN | 6.830202 | NaN | 45.732877 |
| **std** | 64386.100877 | NaN | NaN | NaN | 3.224350 | NaN | 51.304835 |
| **min** | 870.000000 | NaN | NaN | NaN | 1.000000 | NaN | 0.000000 |
| **25%** | 74703.000000 | NaN | NaN | NaN | 4.000000 | NaN | 16.000000 |
| **50%** | 124678.000000 | NaN | NaN | NaN | 8.000000 | NaN | 31.000000 |
| **75%** | 182981.000000 | NaN | NaN | NaN | 10.000000 | NaN | 59.000000 |
| **max** | 232218.000000 | NaN | NaN | NaN | 10.000000 | NaN | 1291.000000 |

top drug is Bupropion

3-3-2016

had highest reviews

top review is "Good"

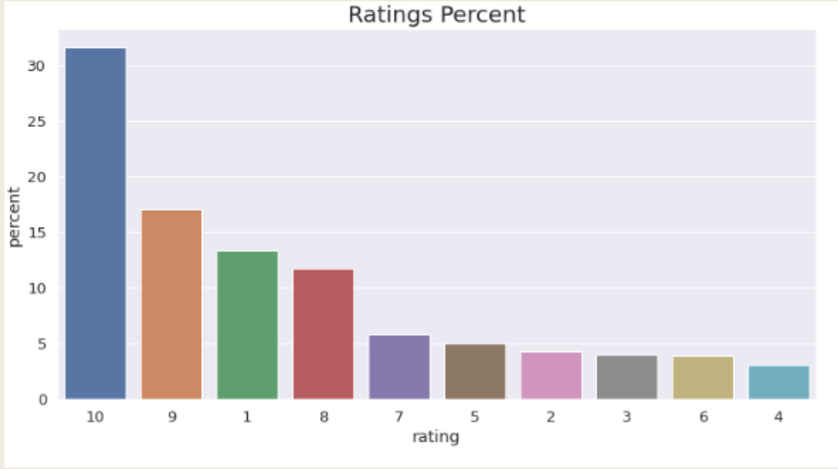
usefulCount has outliers

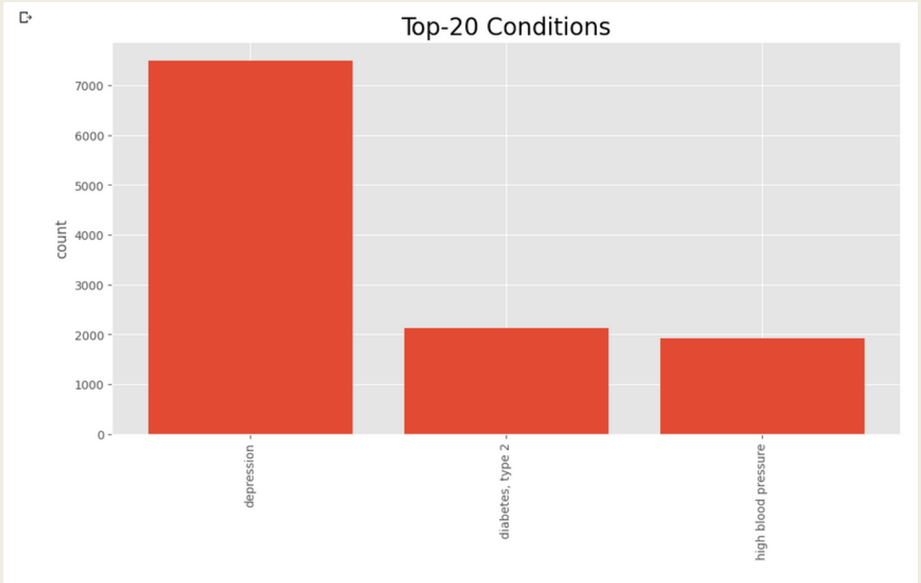

Visualizations

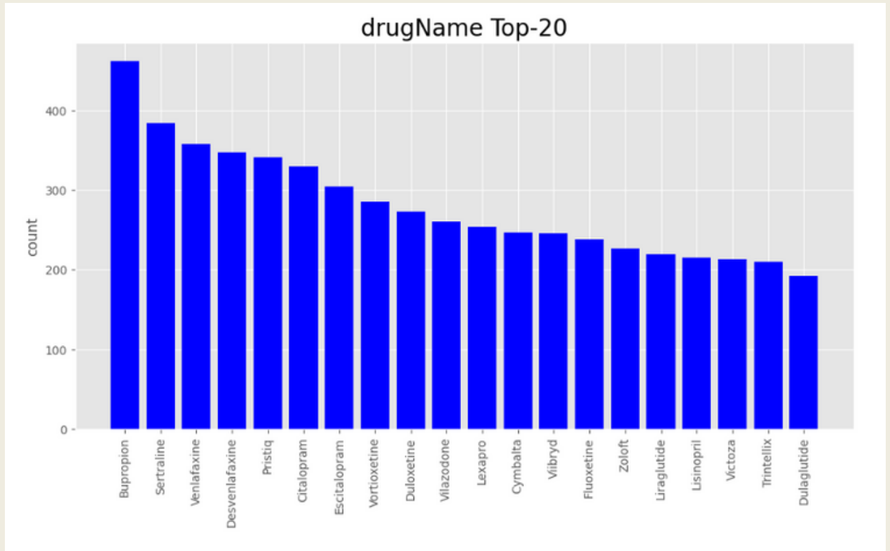
# Visualization

depression count is high and other two has nearly same count

bupropion is most occured drugname we saw that in describe function

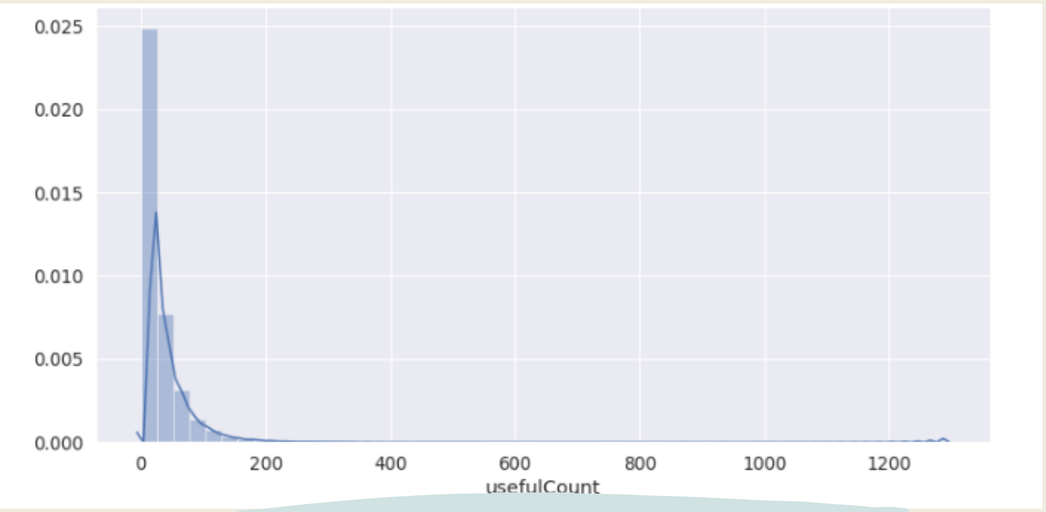


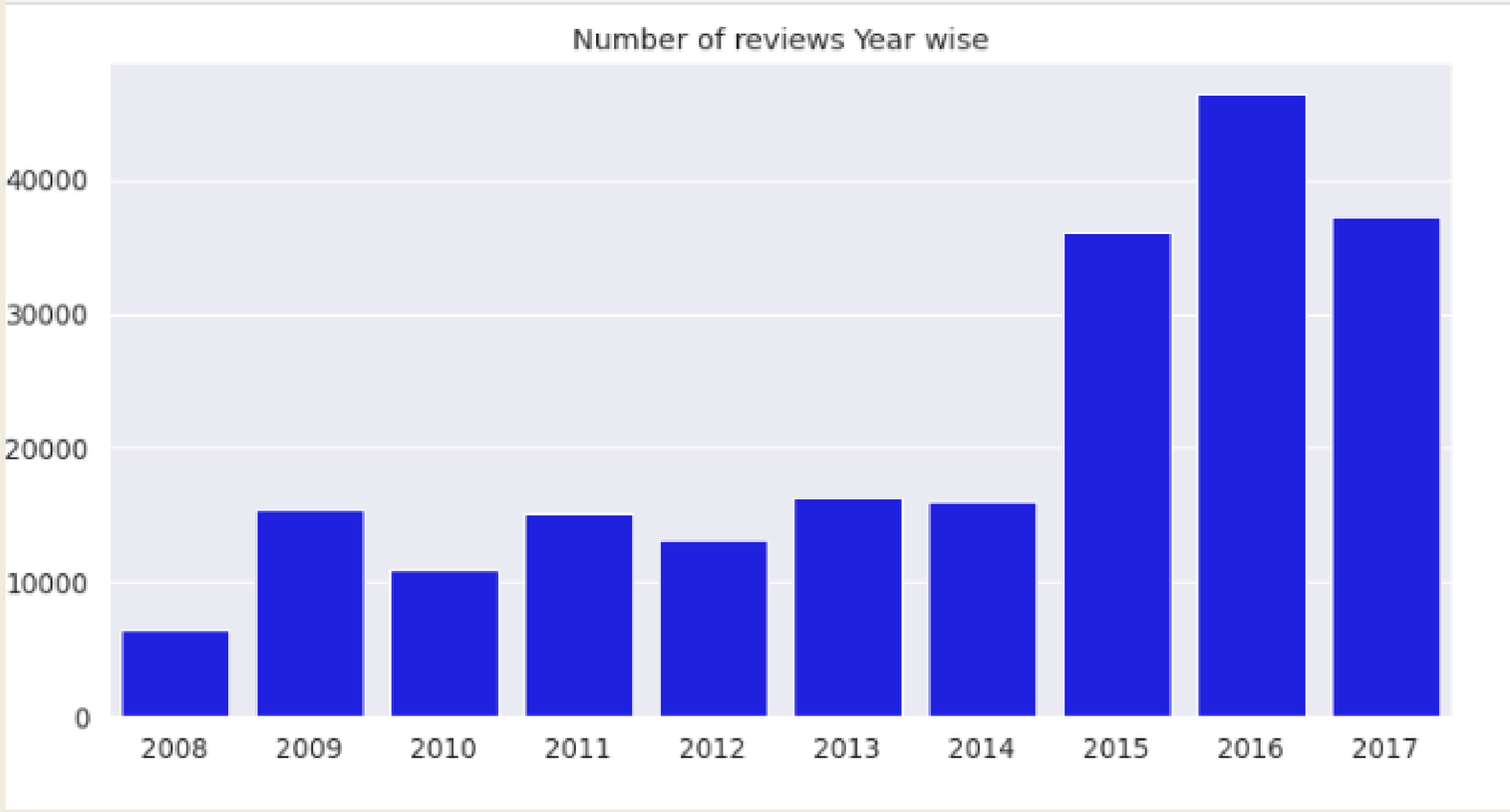




most of drug has rating 10 and 9 & rating one is also high for some of drugs

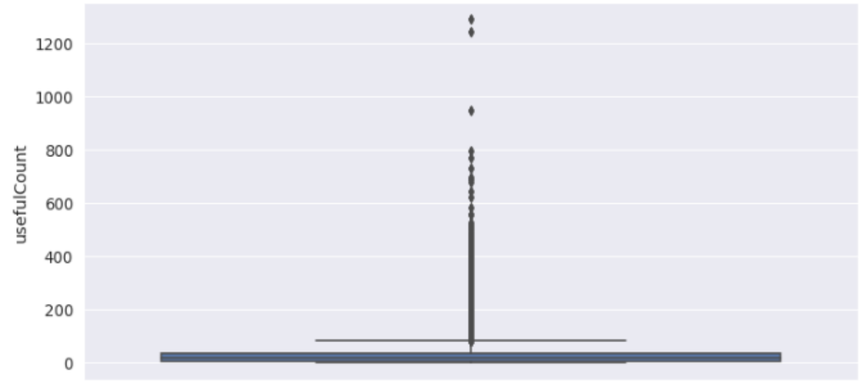
# Visualization




some of drugs reviews is useful to 200 patients

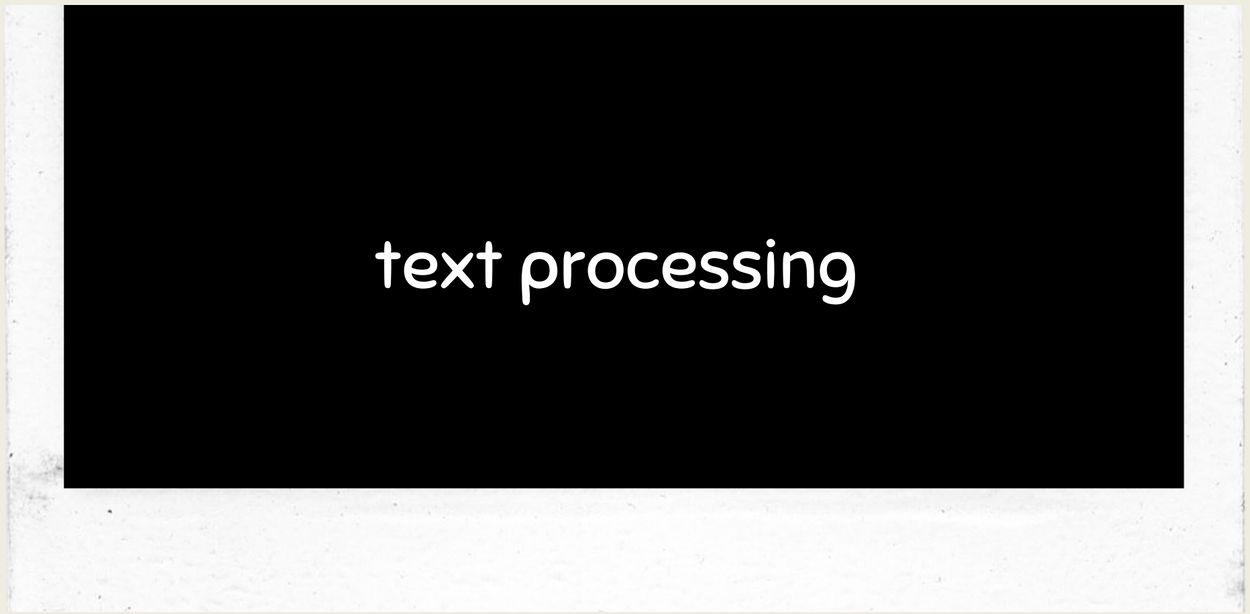



```
#check the descriptive summary
sns.boxplot(y = merged_data['usefulCount'])
plt.show()
```




and some were useful to 1200 patients

year 2015 , 16 , 17 has most reviews


text processing

# Text Processing

checked part of speech for some reviews

```
#parts of speech for each word
tagged = nltk.pos_tag(tokens)
tagged[:10]
```

```
[('third', 'JJ'),
 ('week', 'NN'),
 ('pristiq', 'NN'),
 ('far', 'RB'),
```

removed # , Punctuations , and regular english words

```
# Define the process_review function
def process_review(review):
    # Remove hashtags
    review = review.replace('#', '')

    # Expand contractions
    words = review.split()
    words = [contractions_dict[word] if word in contractions_dict else word for word in words]
    review = ' '.join(words)

    # Tokenize the text
    tokens = word_tokenize(review)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if not token in stop_words]

    # Remove punctuation
    tokens = [token for token in tokens if token.isalpha()]

    # Return the processed tokens
    return ' '.join(tokens)

# Clean the drug reviews
df['processed_review'] = df['review'].apply(process_review)
```

expanded words like can't to can not it is easy for model

# created tokens for each review

```
#tokenize this cleaned review and show top 10 words
tokens = nltk.word_tokenize(example)
tokens[:10]
```

```
['third',
 'week',
 'pristiq',
 'far',
 'feeling',
```

# model building

1. vader model
2. roberta  pretrained by huggins face
3. some other ML Algorithms'

# Model Building

1. vader model (bag of words )

cons – uses sentiment only

```
# for positive statement
sia.polarity_scores('i am so happy')
```

```
{'neg': 0.0, 'neu': 0.334, 'pos': 0.666, 'compound': 0.6115}
```

```
# for negative statement
sia.polarity_scores('this is the worst thing ever')
```

```
{'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}
```

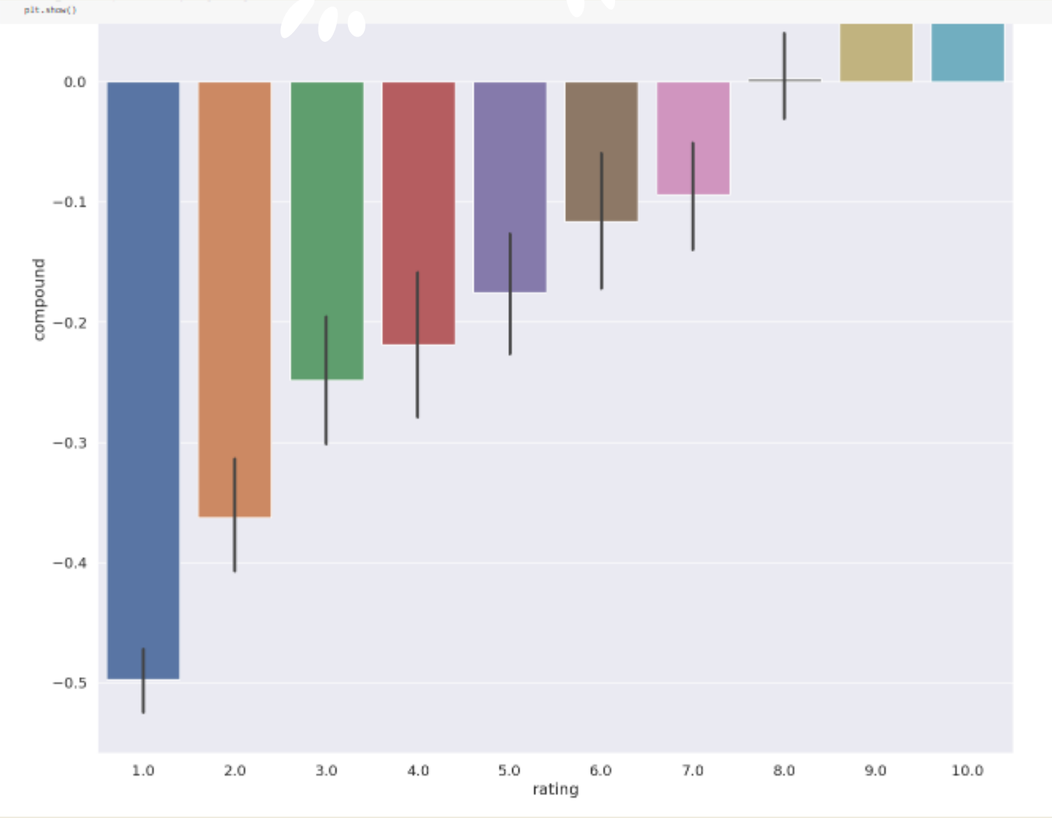
based on sentiment gives +ve or –ve score

```
#now we have sentiment score
veders.head()
```

| | UniqueID | neg | neu | pos | compound | drugName | condition | rating | processed_review |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 75612 | 0.221 | 0.490 | 0.289 | 0.6369 | l-methylfolate | depression | 10.0 | taken years improvement mostly moderate severe... |
| 1 | 96233 | 0.062 | 0.833 | 0.105 | 0.2960 | sertraline | depression | 8.0 | week zoloft anxiety mood swings take mornings ... |
| 2 | 121333 | 0.241 | 0.526 | 0.233 | -0.4215 | venlafaxine | depression | 4.0 | gp started venlafaxine yesterday help depressi... |
| 3 | 156544 | 0.069 | 0.843 | 0.089 | 0.3400 | dulaglutide | diabetes, type 2 | 10.0 | hey guys months since last post wanted give mo... |
| 4 | 131909 | 0.152 | 0.555 | 0.293 | 0.8101 | effexor xr | depression | 10.0 | medicine saved life wits end ready give doctor... |

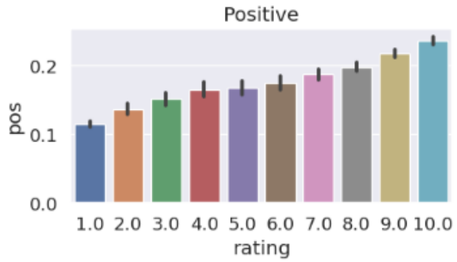
# Model Building

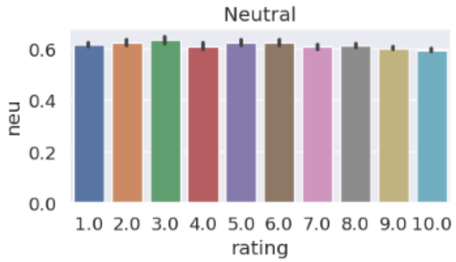
upto rating review
will be –Ve

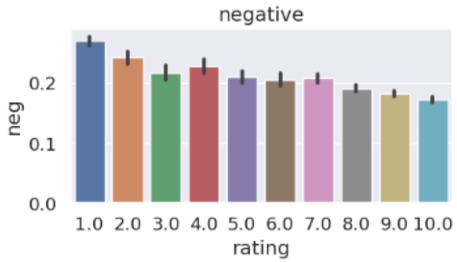



rating increases
positiveness increases

rating decreases
negativeness increases



# Model Building

## 2. Roberta pretrained model

pipeline makes model prediction easy

both model results

```
#run for roberta model
encoded_text = tokenizer(example , return_tensors = 'pt')
output = model(**encoded_text)
scores = output [0][0].detach().numpy()
scores = softmax(scores)
scores_dict = {
    'roberta_neg': scores[0],
    'roberta_neu': scores[1],
    'roberta_pos': scores[2]
}
print(scores_dict)
```

```
{'roberta_neg': 0.0035646246, 'roberta_neu': 0.08046401, 'roberta_pos': 0.9159714}
```

```
[ ] results_df.head()
```

| | UniqueID | neg | neu | pos | compound | roberta_neg | roberta_neu | roberta_pos | drugName | condition | rating | processed_review |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 75612 | 0.221 | 0.490 | 0.289 | 0.6369 | 0.027062 | 0.260905 | 0.712033 | l-methylfolate | depression | 10.0 | taken years improvement mostly moderate severe... |
| 1 | 96233 | 0.062 | 0.833 | 0.105 | 0.2960 | 0.049863 | 0.313331 | 0.636807 | sertraline | depression | 8.0 | week zoloft anxiety mood swings take mornings ... |
| 2 | 121333 | 0.241 | 0.526 | 0.233 | -0.4215 | 0.122782 | 0.466128 | 0.411090 | venlafaxine | depression | 4.0 | gp started venlafaxine yesterday help depressi... |
| 3 | 156544 | 0.069 | 0.843 | 0.089 | 0.3400 | 0.273285 | 0.637052 | 0.089663 | dulaglutide | diabetes, type 2 | 10.0 | hey guys months since last post wanted give mo... |
| 4 | 131909 | 0.152 | 0.555 | 0.293 | 0.8101 | 0.052658 | 0.232951 | 0.714391 | effexor xr | depression | 10.0 | medicine saved life wits end ready give doctor... |

```
sent_pipeline('this drug is best')
```

```
[{'label': 'POSITIVE', 'score': 0.9968098998069763}]
```

# Model Building

best predictions drug has rating 1 but review is +ve

best predictions drug has rating 10 but review is –ve

## review examples

positive 1 _star and negative 5 star review

lets look at some examples where the model scoring and review score differs the most

```
[ ] results_df.query('rating == 1').sort_values('roberta_pos' ,ascending = False)['processed_review'].values[0]
```

```
'experienced many side effects blood sugar higher placed actos doctor took immediately felt great'
```

```
[ ] results_df.query('rating == 1').sort_values('pos' ,ascending = False)['processed_review'].values[0]
```

```
'feel better'
```

```
[ ] results_df.query('rating == 10').sort_values('roberta_neg' ,ascending = False)['processed_review'].values[0]
```

```
'started diovan mg today norvasc awhile diovan dropped blood pressure diovan literally felt like crap weak shortness breath legs hurt walking dogs felt terrible call dr tomorrow let know'
```

```
[ ] results_df.query('rating == 10').sort_values('neg' ,ascending = False)['processed_review'].values[1]
```

```
'autumn depression'
```

# Model Building

## 3. ML Algorithms

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, OneHotEncoder


# Select the columns for the features and target variable
feature_cols = ['drugName', 'condition', 'processed_review']
target_col = 'sentiment'

# Separate the features and target variable into separate DataFrames
X = df[feature_cols]
y = df[target_col]


# Convert categorical features to numerical using one-hot encoding
X_encoded = pd.get_dummies(X, columns=['drugName', 'condition','processed_review'])

# Convert categorical target variable to numerical using label encoding
le = LabelEncoder()
y_encoded = le.fit_transform(y)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded, test_size=0.2, random_state=42)
```

```
# Create a new column called "sentiment" based on the "rating" column
df['sentiment'] = df['rating'].apply(lambda x: 'positive' if x > 5 else 'negative')
```

```
df.head()
```

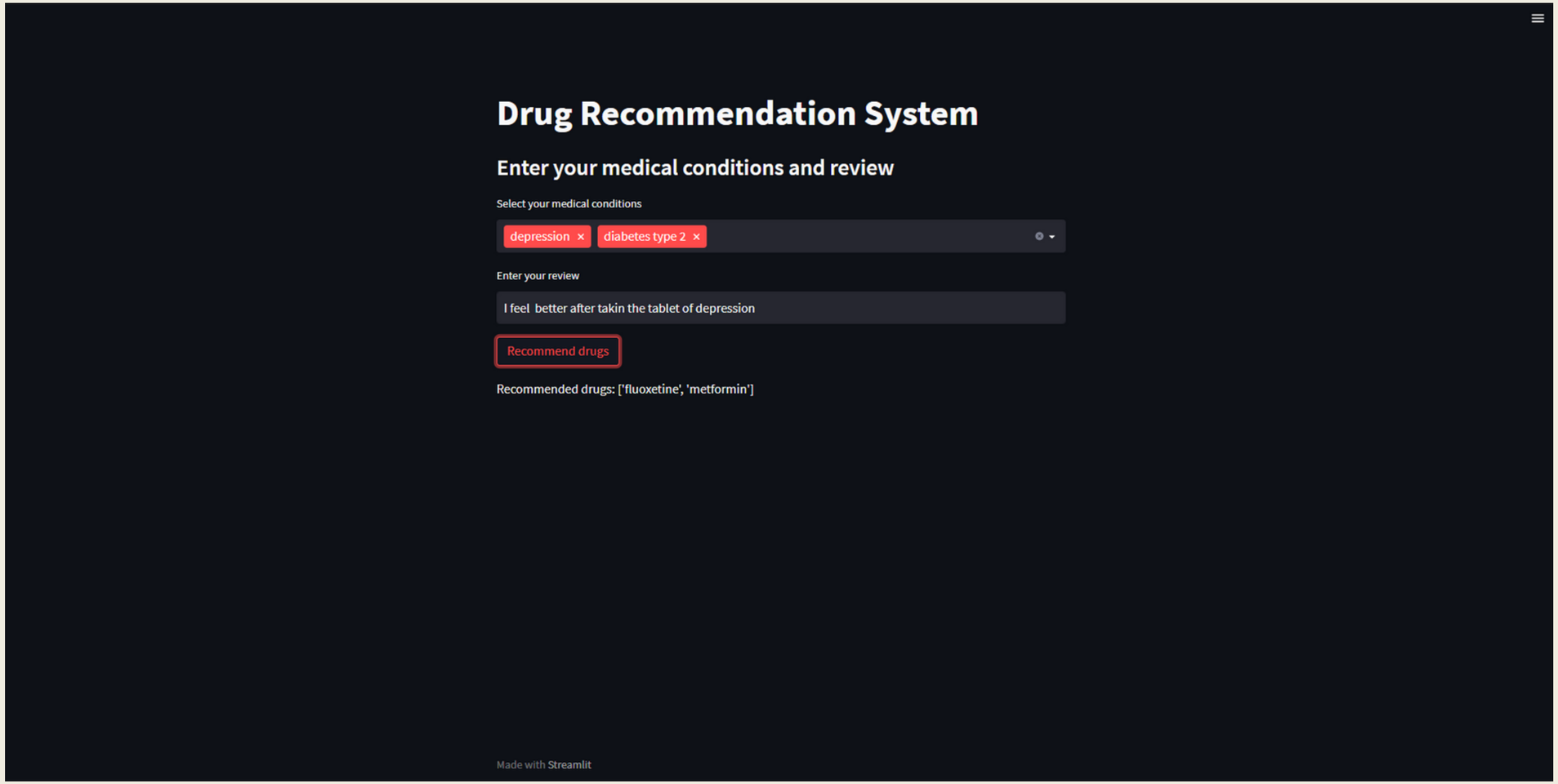
| | UniqueID | drugName | condition | rating | processed_review | sentiment |
|---|---|---|---|---|---|---|
| 0 | 75612 | l-methylfolate | depression | 10.0 | taken years improvement mostly moderate severe... | positive |
| 1 | 96233 | sertraline | depression | 8.0 | week zoloft anxiety mood swings take mornings ... | positive |
| 2 | 121333 | venlafaxine | depression | 4.0 | gp started venlafaxine yesterday help depressi... | negative |
| 3 | 156544 | dulaglutide | diabetes, type 2 | 10.0 | hey guys months since last post wanted give mo... | positive |
| 4 | 131909 | effexor xr | depression | 10.0 | medicine saved life wits end ready give doctor... | positive |

# ML Algorithms

| | | Accuracy |
|---|---|---|
| 1. | Logistic Regression | Accuracy: 0.73 |
| 2. | RandomForestClassifier | Accuracy: 0.70 |
| 3. | Naive Bayes classifier | Accuracy: 0.73 |
| 4. | support vector machine | Accuracy: 0.81 |
| 5. | GradientBoostingClassifier | Accuracy: 0.69 |

# Deployment Result

# Recommendation app



# Conclusion

1. In conclusion, drug sentiment analysis is a useful tool for understanding how people perceive and react to different medications. By analyzing large amounts of data from , reviews, and other sources, sentiment analysis can provide valuable insights into the efficacy, safety, and overall patient satisfaction with different drugs.

2. Based on this analysis, we can make informed recommendations for individuals who are considering taking a particular drug or seeking treatment for a specific condition. For example, if the sentiment analysis shows that a particular drug is associated with a high incidence of negative side effects or poor treatment outcomes, we may recommend that individuals seek alternative treatment options.

# Challenges

- Some errors were faced while deploying the model using streamlit;
- Main challenge was with the ‘deployment of model’ due to lack of sufficient knowledge;
- Some of the team members had training on ‘Heroku’ only, which is a paid version presently.

Does anyone
have questions?

# Thank You