PROJECT - 209

Patient's Condition Classification Using Drug Reviews

Working Team

- Suraj Santosh Temkar
- Triveni Gunwant Chapekar
- Vaishnavi Tandle
- Snehal Sanjay Pophale
- Muktagucha Manisha

Dataset Details

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

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

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.

Project Architecture

Collection of data

NEXT

Business understanding

NEXT

EDA

Model Building

NEXT

Deployment

Table of Contents

Data Collection & EDA

- Imported Libraries & Dataset
- Found (0) Duplicate Values
- Found (899) Missing Values in Condition column
- Dropped the Null Values
- Removed Unnecessary Columns
- Converted Numerical dtypes to Same dtypes
- Applied Goal Conditions

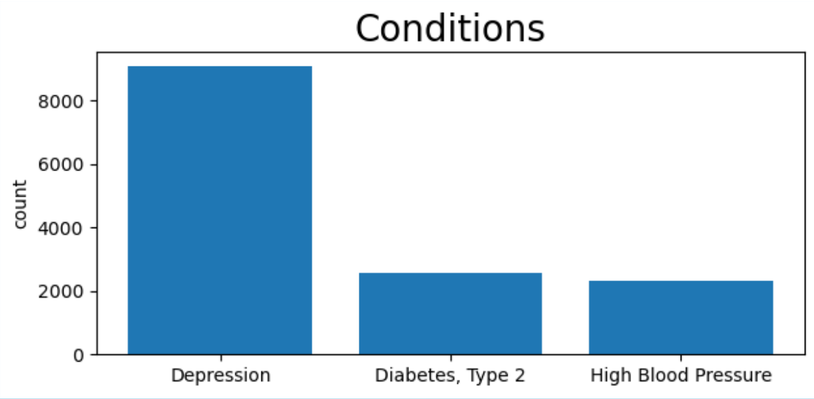
```
# Before: value count of 'condition'
df2['condition'].nunique()
```

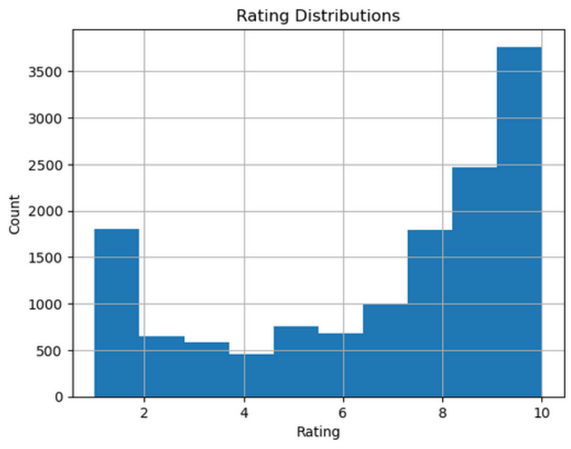
```
884
```

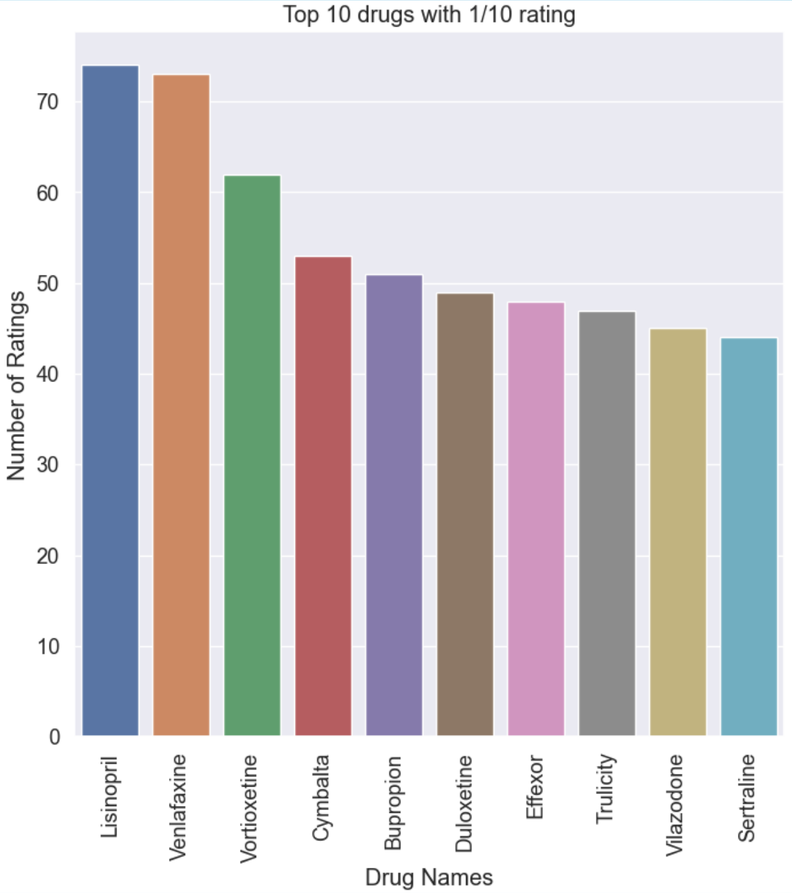
```
# After: names of 'condition'
df3['condition'].unique()
```

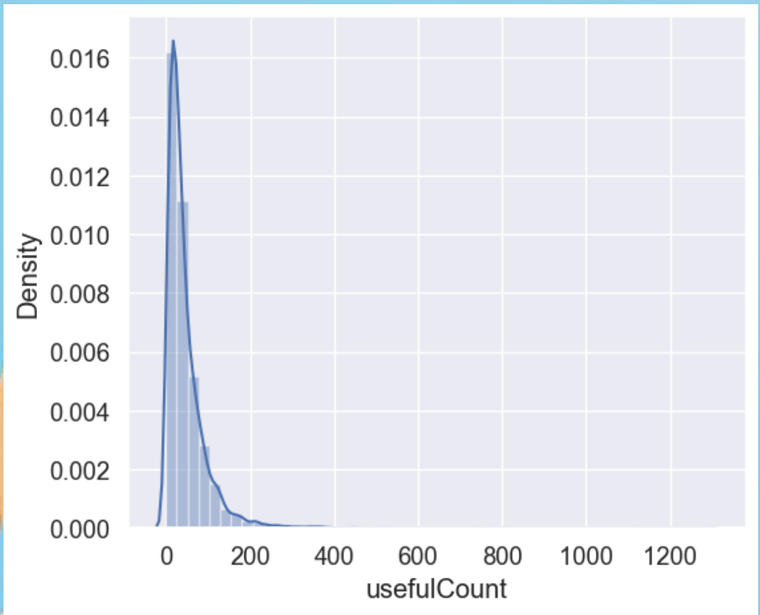
```
array(['Depression', 'Diabetes, Type 2', 'High Blood Pressure'],
      dtype=object)
```

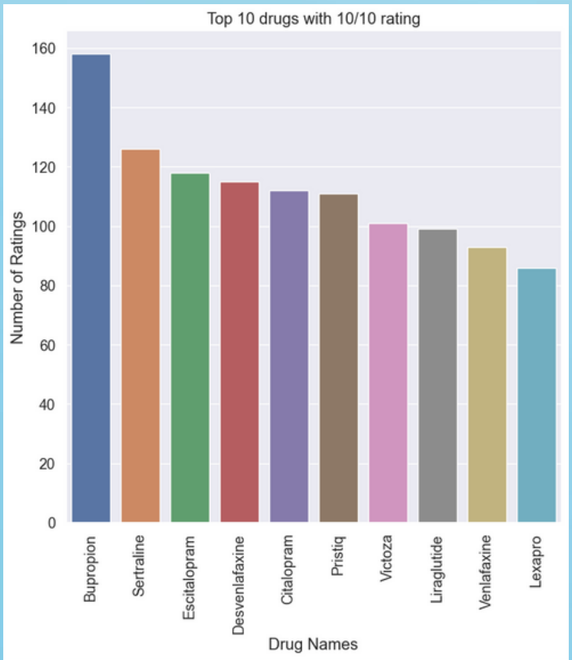
Visualizations











WordCloud of Conditions

Word Cloud for Depression

Word Cloud for Diabetes, Type 2

Word Cloud for High Blood Pressure

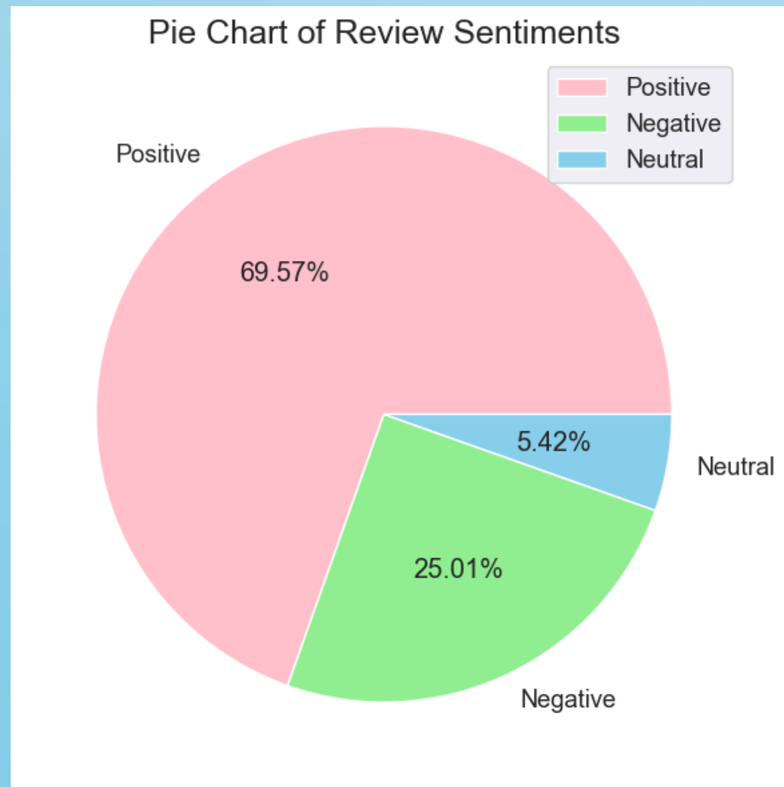
NLP & Text Mining

```
# Import the libraries for pre-processing
from bs4 import BeautifulSoup
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem.snowball import SnowballStemmer

stops = set(stopwords.words('english')) #English stopwords

stemmer = SnowballStemmer('english') #SnowballStemmer

def review_clean(raw_review):
    # 1. Delete HTML
    review_text = BeautifulSoup(raw_review, 'html.parser').get_text()
    # 2. Make a space
    letters_only = re.sub('[^a-zA-Z]', ' ', review_text)
    # 3. lower letters
    words = letters_only.lower().split()
    # 5. Stopwords
    meaningful_words = [w for w in words if not w in stops]
    # 6. Stemming
    stemming_words = [stemmer.stem(w) for w in meaningful_words]
    # 7. Space join words
    return( ' '.join(stemming_words))
```

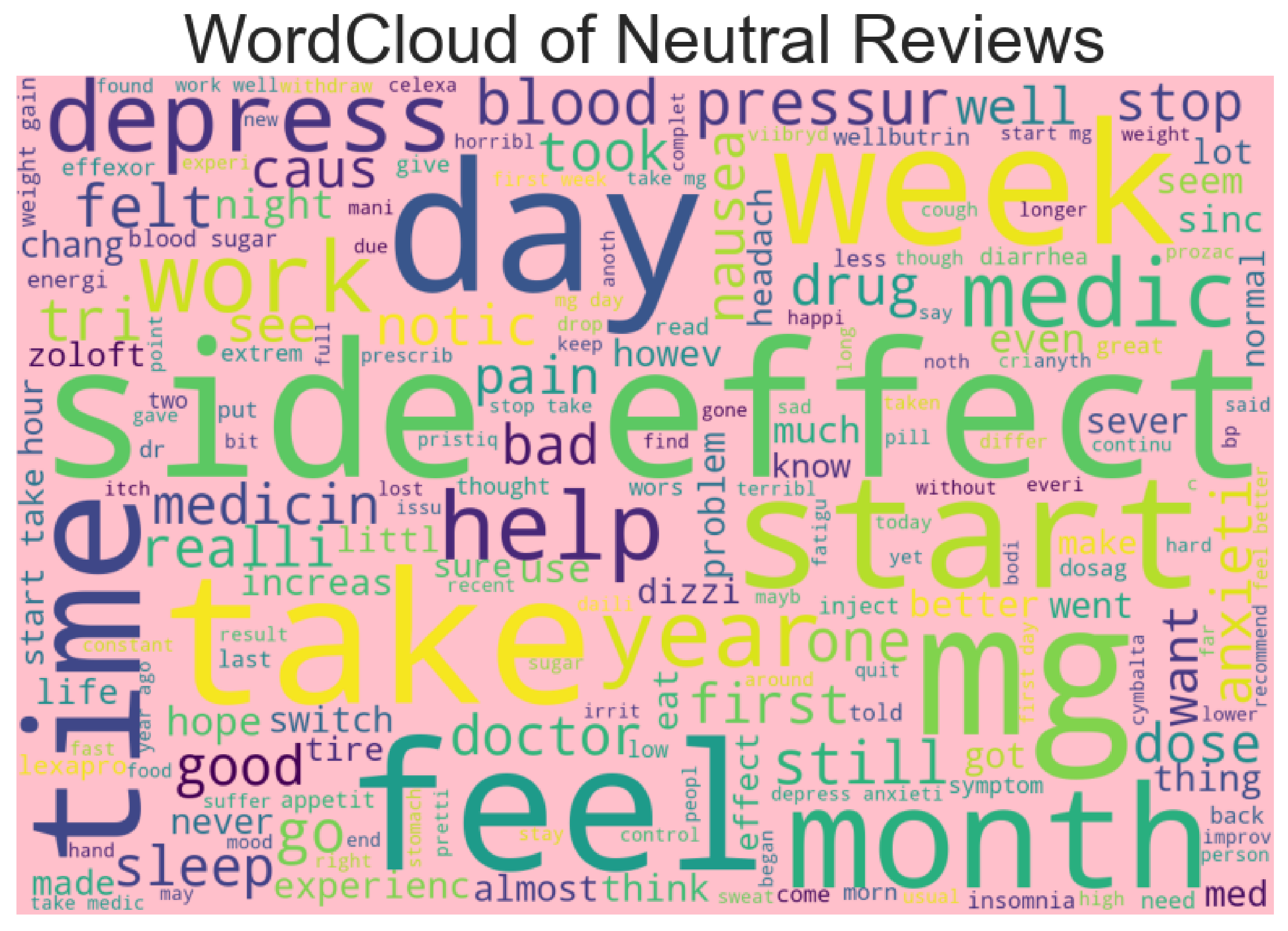
condition	review
Depression	taken anti depress year improv most moder seve...
Depression	week zoloft anxieti mood swing take mg morn br...
Depression	gp start venlafaxin yesterday help depress cha...
Diabetes, Type 2	hey guy month sinc last post want give month s...
Depression	medicin save life wit end anti depress readi g...
...	...
High Blood Pressure	fourth blood pressur pill feel like part work ...
High Blood Pressure	bystol feet arm numb blood sugar becam sever e...
Diabetes, Type 2	got diagnos type doctor prescrib invokana metf...
Depression	third med tri anxieti mild depress week hate m...
High Blood Pressure	tekturna day effect immedi also calcium channe...

Sentiment Analysis

```
# let's make a new column of review sentiment
df3.loc[(df3['rating'] > 5), 'Review_Sentiment'] = '1'
df3.loc[(df3['rating'] == 5), 'Review_Sentiment'] = '0'
df3.loc[(df3['rating'] < 5), 'Review_Sentiment'] = '-1'
df3
```



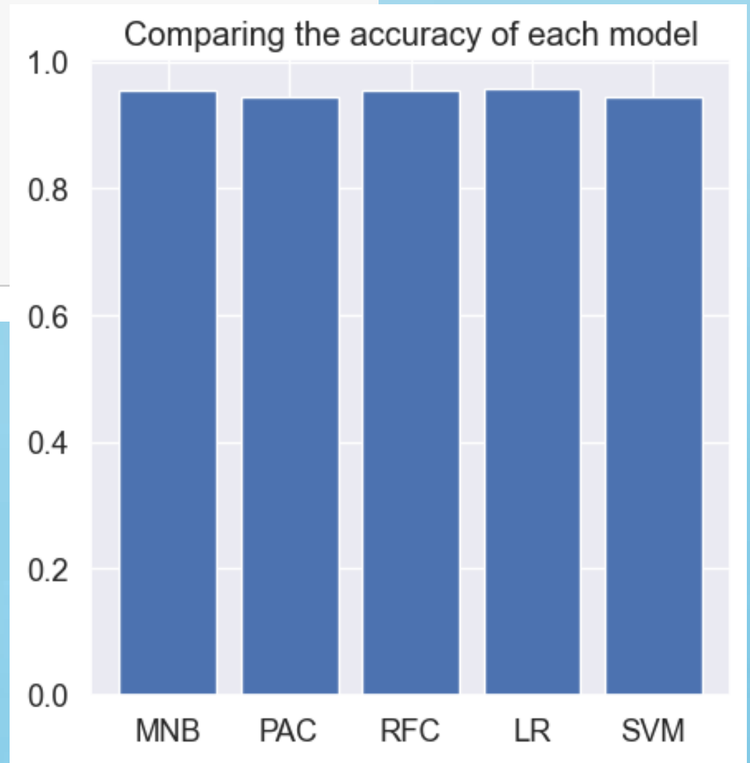
WordCloud Of Reviews



Model Building

Count Vectorizer (Bag of Words)

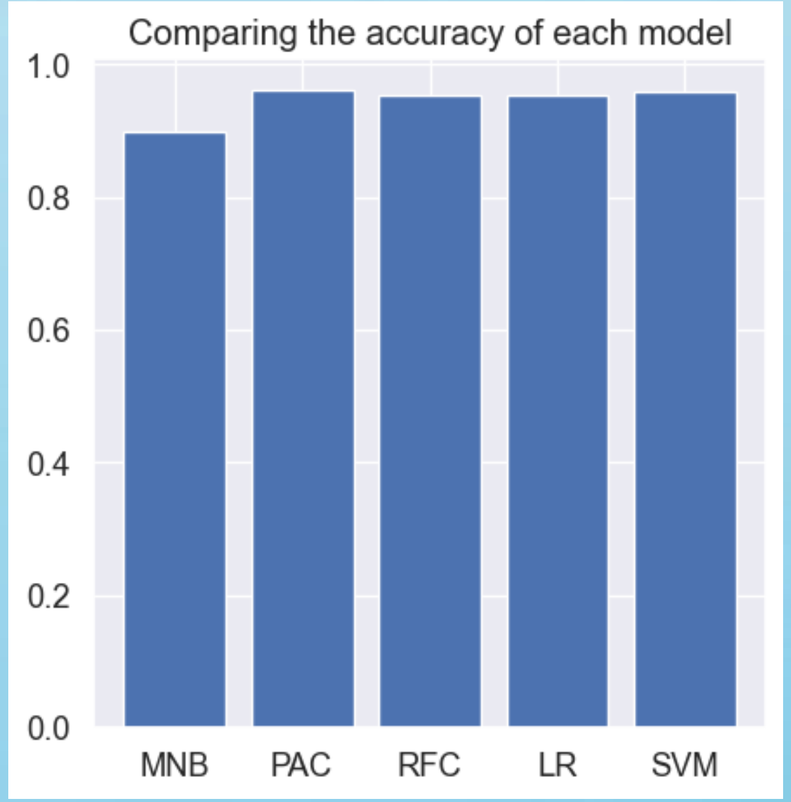
```
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
import itertools

count_vectorizer = CountVectorizer(stop_words='english')

count_train = count_vectorizer.fit_transform(x_train)

count_test = count_vectorizer.transform(x_test)
```



==> CountVectorizer simply counts the number of times a word appears in a document (using a bag-of-words approach), while TF-IDF Vectorizer takes into account not only how many times a word appears in a document but also how important that word is to the whole corpus

TF-IDF Vectorizer

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_
tfidf_train = tfidf_vectorizer.fit_transform(x_train)
tfidf_test = tfidf_vectorizer.transform(x_test)
```



==> From the above visualization, we can see that 'PAC' model has the high accuracy among all

Model Prediction

Model Prediction

Predictions Samples

```
text1 = ["I have taken anti-depressants for years, with some improvement but mostly moderate to severe side affects, which ma
test = tfidf_vectorizer.transform(text1)
pred1 = pac_tf.predict(test)[0]
pred1
```

```
'Depression'
```

```
text2 = ["my gp started me on Venlafaxine yesterday to help with depression and the change,a hour after taking them i was fee
test = tfidf_vectorizer.transform(text2)
pred2 = pac_tf.predict(test)[0]
pred2
```

```
'Diabetes, Type 2'
```

Model Deployment

```
import joblib
joblib.dump(tfidf_vectorizer,'tfidf_vectorizer.pkl')
joblib.dump(pac_tf,'Pac_tf.pkl')
```

```
['Pac_tf.pkl']
```

```
vectorizer = joblib.load('tfidf_vectorizer.pkl')
model = joblib.load('pac_tf.pkl')
```

pkl files

```
from flask import Flask, render_template ,request
import os
import joblib
import pandas as pd
import re
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from bs4 import BeautifulSoup

HTML_WRAPPER = """<div style="overflow-x: auto; border: 1px solid #e6e9ef; border-radiu

app = Flask(__name__, template_folder='template')

app.secret_key =os.urandom(24)
MODEL_PATH ='pac_tf.pkl'
TOKENIZER_PATH ='tfidf_vectorizer.pkl'
DATA_PATH =pd.read_csv('drugsCom_raw.tsv',sep="\t")

# loading vectorizer
vectorizer = joblib.load(TOKENIZER_PATH)
# loading model
model = joblib.load(MODEL_PATH)

# getting stopwords
stop = stopwords.words('english')
lemmatizer = WordNetLemmatizer()
```

app1.py

app1.py | home1.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
        integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
   <title>Medical Condition Predictor and Drug Recommender </title>

    <style>
        .highlight {background-color: #FFFF00}

        div {
font-size: 20px;
}
    </style>
    <script src="/scripts/snippet-javascript-console.min.js?v=1"></script>
<!--============================================================================================-->
      <style>
p.round1 {

  border-radius: 8px;

  padding:3px;

  text-align:center;
  color:black;
  background:#0C95F7;
}

p.round2 {

  border-radius: 8px;

  padding:3px;

  text-align:center;
```

Hyper Text Markup Language file length : 2,805 lines : 122 Ln : 1 Col : 1 Pos : 1

Home.html

app1.py | home1.html | predict.html

```
{%extends "home1.html" %}
{%block content%}


<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>


<div class="card text-center">
        <div class="card-header">
        <strong>   Original Entered Review : </strong>
        </div>
        <div class="card-body">

            <p id="new" class="card-text">

             {{ rawtext }}
            </p>

        </div>

    </div>



<br><br>
```

predict.html

Model Predictor Output

Original Entered Review :

I have taken anti-depressants for years, with some improvement but mostly moderate to severe side affects, which makes me go off them

Predicted Medical Condition

Depression

Top 3 Recommended Drugs:

Sertraline **Zoloft** **Viibryd**

Project Challenges

- Cleaning the dataset as there are null values and missing data
- As dataset is large forming Visualizations to applied 'goal conditions' is a task
- Text Mining, feature extraction by using Different techniques to clean the 'Reviews' in data is a task
- Building a model and choosing suitable type of algorithm without overfitting the data using 'CountVectorizor', 'TF-IDF', 'Bigrams', 'Trigrams' for Accuracy
- Used Deployment using Flask, HTML, Css ans adding pickle file to run the output and predict the data Succesfully

THANKYOU