OESON Project 5:
Medicine Details Analysis

By: Christopher Gonzalez

Introduction

- **The pharmaceutical industry often faces challenges of data related to medicine composition, uses, & side effects.**
- **Given how available medication continues to grow, providers are in need of reliable insights to then prescribe.**
- **Furthermore, understanding medical usage & patient satisfaction can help companies make improvements.**
- **This presentation will uncover significant patterns to improve the healthcare industry and patient outcomes.**

Background of the Dataset

- **The dataset contains over 11,000 medicines with their salt composition, uses, side effects, & patient satisfaction.**
- **Patient satisfaction is divided into three categories: Excellent Review %, Average Review %, & Poor Review %.**
- **Since the goal is to maximize satisfaction, it's encouraged to make observations using the Excellent Review %.**
- **Similarly, we can use Poor Review % to make observations into factors that lead to product dissatisfaction.**

A Portion of the Dataset

	Medicine Name	Composition	Uses	Side_effects	Image URL	Manufacturer	Excellent Review %
1	Avastin 400mg Injection	Bevacizumab (400mg)	an cancer Cervical cancer	Inflammation of the nose	199ab116a69a969be3.jpg	he Products India Pvt Ltd	22
2	ugmentin 625 Duo Tablet	Clavulanic Acid (125mg)	ent of Bacterial infections	ucocutaneous candidiasis	y2y9bdipmh6rgkrj0zm.jpg	line Pharmaceuticals Ltd	47
3	Azithral 500 Tablet	Azithromycin (500mg)	ent of Bacterial infections	Abdominal pain Diarrhea	kqkouvaqejbyk47dvjfu.jpg	mbic Pharmaceuticals Ltd	39
4	Ascoril LS Syrup	+ Guaifenesin (50mg/5ml)	ent of Cough with mucus	amp Increased heart rate	73ae66cbb0dbfded86.jpg	nark Pharmaceuticals Ltd	24
5	Aciloc 150 Tablet	Ranitidine (150mg)	nt of Peptic ulcer disease	astrointestinal disturbance	n7apngctvrtweencwi1.jpg	adila Pharmaceuticals Ltd	34
6	Allegra 120mg Tablet	Fexofenadine (120mg)	nent of Allergic conditions	siness Dizziness Nausea	63b5bbafb529df0736.jpg	Sanofi India Ltd	35
7	Avil 25 Tablet	Pheniramine (25mg)	tion of Meniere's disease	Sedation	msye6bf97tkcocat24j.jpg	Sanofi India Ltd	40
8	Aricep 5 Tablet	Donepezil (5mg)	Alzheimer's disease	ight loss Accidental injury	l7ojjdaw2gsm5sie1glu.jpg	rmaceuticals India Pvt Ltd	43
9	Amoxyclav 625 Tablet	Clavulanic Acid (125mg)	ent of Bacterial infections	ucocutaneous candidiasis	56f9cafcca2156ad3de.jpg	Abbott	36
10	Atarax 25mg Tablet	Hydroxyzine (25mg)	ith inflammation & itching	set stomach Constipation	l/v9py58kciridvbi7bqls.jpg	Reddy's Laboratories Ltd	35

Gaining Insight Into the Medicines

- **To make observations, the dataset was uploaded and analyzed using Tableau.**

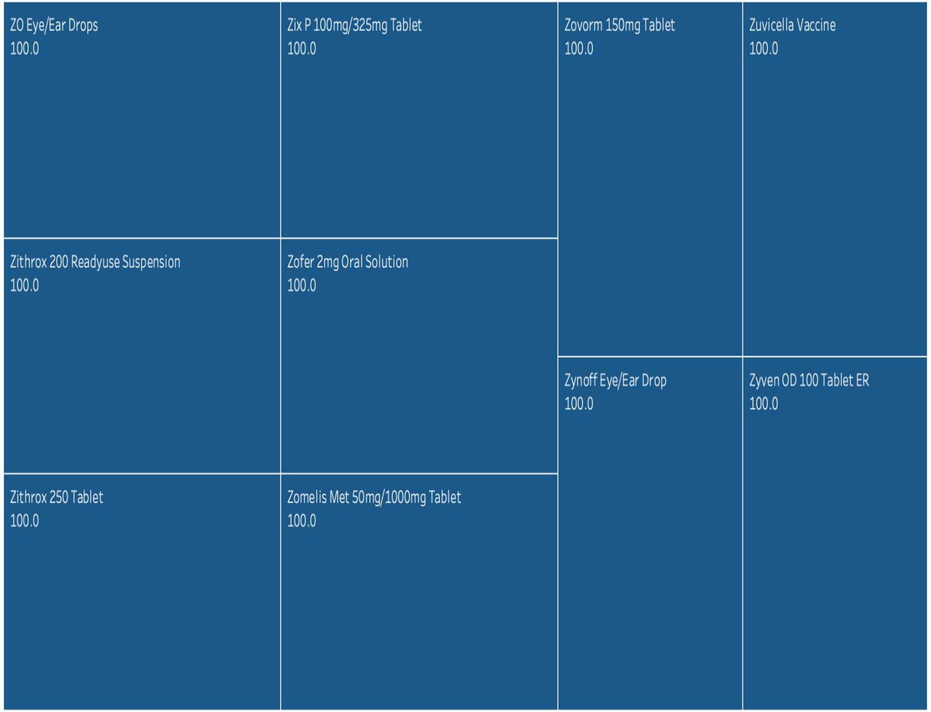
- **Using Tableau allows us to identify factors that influence medicine effectiveness & patient satisfaction.**

- **Afterwards, Tableau is used to identify common side effects with their associated medicines.**

- **Finally, user satisfaction is predicted using machine and deep learning models in Python.**

Patient Satisfaction by Medicine Name

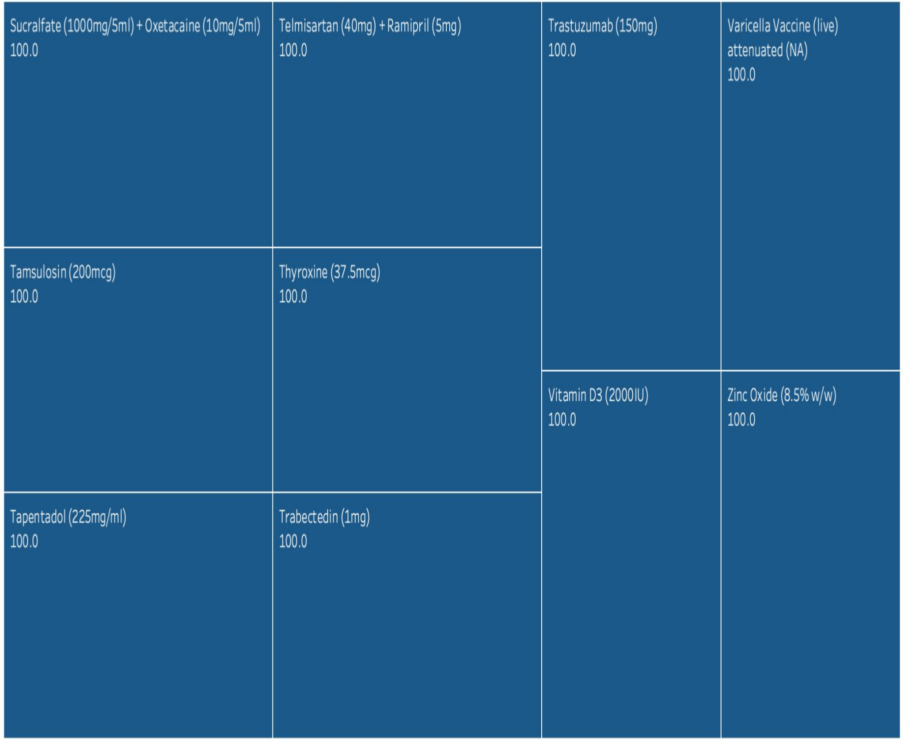
Satisfaction



Dissatisfaction

Patient Satisfaction by Composition

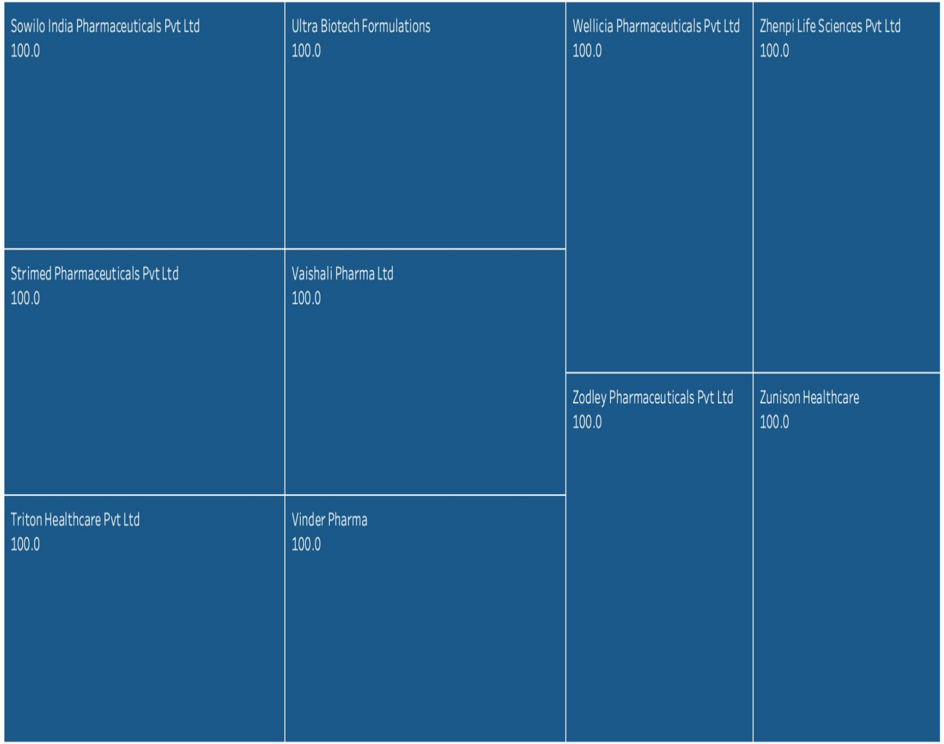
Satisfaction



Dissatisfaction

Patient Satisfaction by Manufacturer

Satisfaction



Dissatisfaction

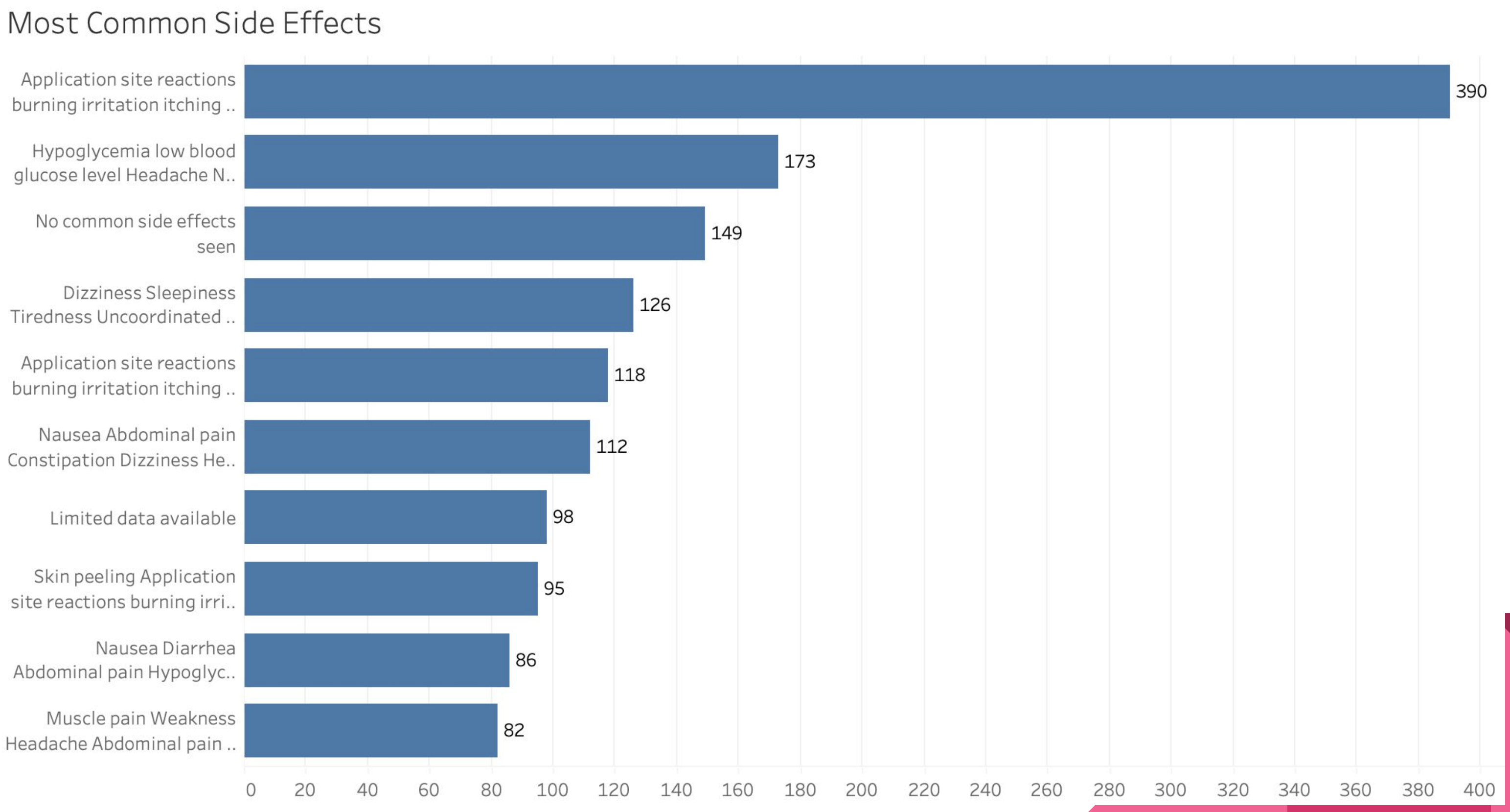
Significant Findings

- Some compositions that contributed to patient satisfaction were Sucralfate, Thyroxine, & Vitamin D3.
- On the contrast, medicines that contain Paracetamol, Metformin, & Vitamin E led to poor reviews.
- Medicine tablets appear to make the majority of the data, while other medicines include shampoo, eye, & ear drops.
- It should be noted that other medicines also received a excellent/poor rating of 100, however their inclusion would make the data harder to visualize.

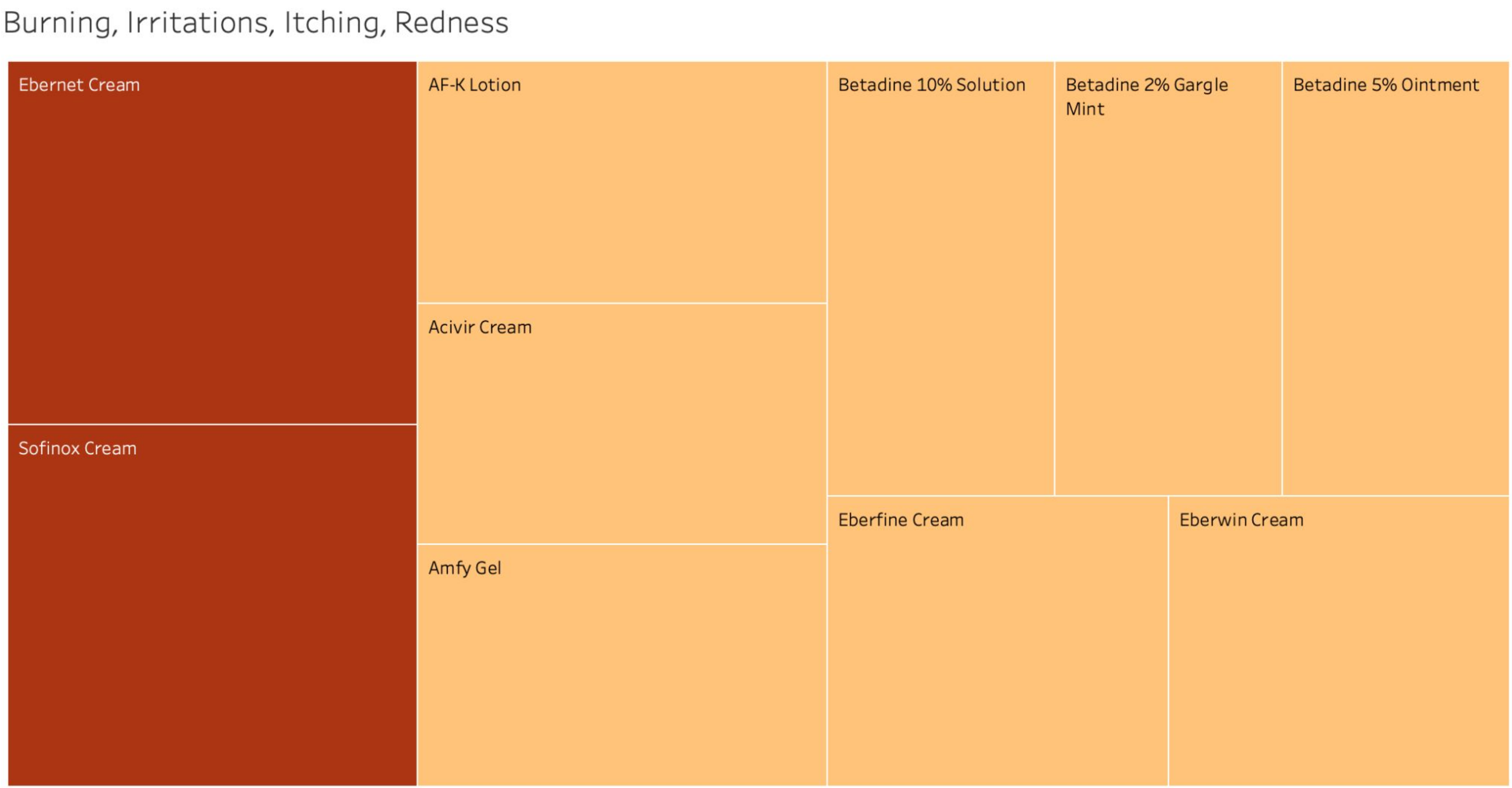
Visualizing Common Side Effects

- **Most, if not all medicines have side effects which can occur to the patient.**
- **Medicines with higher occurring side effects are less likely to receive good ratings & vice versa.**
- **Using the dataset, it is possible to determine the most common side effects, as well as the corresponding medicine.**

Common Side Effects



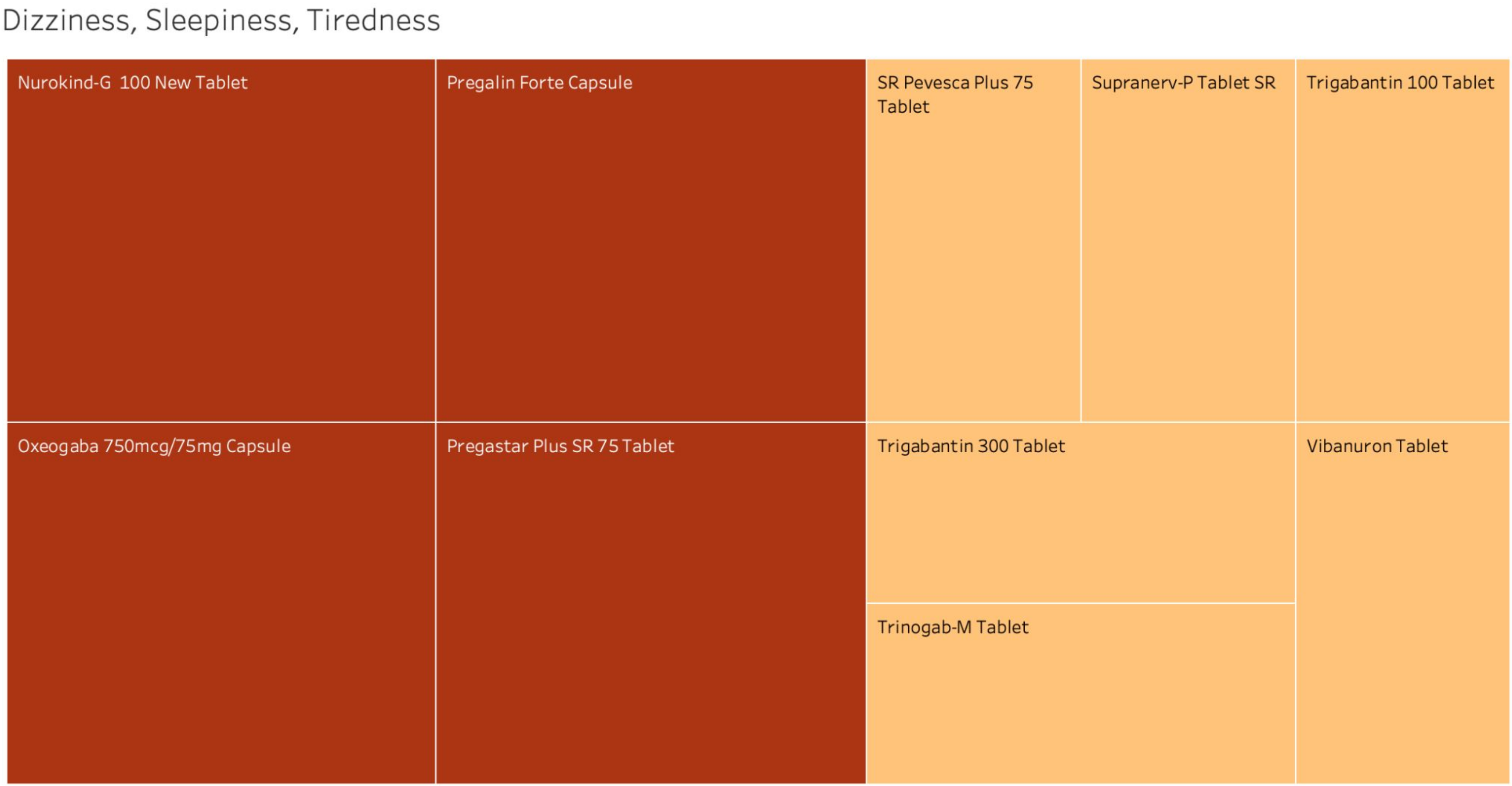
Medicines with Burning/Itching Side Effect



Medicines with Hypoglycemia Side Effect

Hypoglycemia, Low Blood Glucose Level

Medicines with Dizziness/Tiredness Side Effect



Significant Findings

- **The three most common side effects are application site reactions, hypoglycemia, & dizziness/tiredness.**
- **Application site reactions (burning, itching, irritation) are more common than the other two effects combined.**
- **More reports of application reactions are seen in creams compared to other medicine types like lotions & gels.**

Using Python to Make Predictions

- **The final task of this project is to create a model to predict user satisfaction ratings with accuracy.**
- **To accomplish this, the construction of machine & deep learning models are required.**
- **First, it is crucial to explore and determine the best machine model for the dataset so it can be incorporated.**
- **Once the deep learning model is built, its overall performance can be evaluated.**

Importing Libraries

```python
#Import Necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import (mean_squared_error, mean_absolute_error, r2_score)

from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
from keras.callbacks import EarlyStopping
from keras.optimizers import Adam
```

Loading the Dataset

```python
#Load the dataset
data = pd.read_csv('Medicine_Details.csv')
data.head(10)
```

	Medicine Name	Composition	Uses	Side_effects	Image URL	Manufacturer	Excellent Review %	Average Review %	Poor Review %
0	Avastin 400mg Injection	Bevacizumab (400mg)	Cancer of colon and rectum Non-small cell lun...	Rectal bleeding Taste change Headache Noseblee...	https://onemg.gumlet.io/l_watermark_346,w_480,...	Roche Products India Pvt Ltd	22	56	22
1	Augmentin 625 Duo Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	Treatment of Bacterial infections	Vomiting Nausea Diarrhea Mucocutaneous candidi...	https://onemg.gumlet.io/l_watermark_346,w_480,...	Glaxo SmithKline Pharmaceuticals Ltd	47	35	18
2	Azithral 500 Tablet	Azithromycin (500mg)	Treatment of Bacterial infections	Nausea Abdominal pain Diarrhea	https://onemg.gumlet.io/l_watermark_346,w_480,...	Alembic Pharmaceuticals Ltd	39	40	21
3	Ascoril LS Syrup	Ambroxol (30mg/5ml) + Levosalbutamol (1mg/5ml)...	Treatment of Cough with mucus	Nausea Vomiting Diarrhea Upset stomach Stomach...	https://onemg.gumlet.io/l_watermark_346,w_480,...	Glenmark Pharmaceuticals Ltd	24	41	35
4	Aciloc 150 Tablet	Ranitidine (150mg)	Treatment of Gastroesophageal reflux disease (...	Headache Diarrhea Gastrointestinal disturbance	https://onemg.gumlet.io/l_watermark_346,w_480,...	Cadila Pharmaceuticals Ltd	34	37	29
5	Allegra 120mg Tablet	Fexofenadine (120mg)	Treatment of Sneezing and runny nose due to al...	Headache Drowsiness Dizziness Nausea	https://onemg.gumlet.io/l_watermark_346,w_480,...	Sanofi India Ltd	35	42	23

Checking for Missing Values/Dropping

```
#Check for Missing Values
print(data.isnull().sum())
```

```
Medicine Name         0
Composition           0
Uses                  0
Side_effects          0
Image URL             0
Manufacturer          0
Excellent Review %    0
Average Review %      0
Poor Review %         0
dtype: int64
```

```
#Drop Irrelevant Columns
data = data.drop(columns = ['Image URL'])
data.head()
```

	Medicine Name	Composition	Uses	Side_effects	Manufacturer	Excellent Review %	Average Review %	Poor Review %
0	Avastin 400mg Injection	Bevacizumab (400mg)	Cancer of colon and rectum Non-small cell lun...	Rectal bleeding Taste change Headache Noseblee...	Roche Products India Pvt Ltd	22	56	22
1	Augmentin 625 Duo Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	Treatment of Bacterial infections	Vomiting Nausea Diarrhea Mucocutaneous candidi...	Glaxo SmithKline Pharmaceuticals Ltd	47	35	18
2	Azithral 500 Tablet	Azithromycin (500mg)	Treatment of Bacterial infections	Nausea Abdominal pain Diarrhea	Alembic Pharmaceuticals Ltd	39	40	21
3	Ascoril LS Syrup	Ambroxol (30mg/5ml) + Levosalbutamol (1mg/5ml)...	Treatment of Cough with mucus	Nausea Vomiting Diarrhea Upset stomach Stomach...	Glenmark Pharmaceuticals Ltd	24	41	35

Preparing the Data

```
#Separate Features and Target Variable
x = data.drop('Excellent Review %', axis = 1)
y = data['Excellent Review %']

#Identify Categorical and Numerical Columns
categorical_cols = x.select_dtypes(include = ['object']).columns
numerical_cols = x.select_dtypes(include = ['float64', 'int64']).columns

#Data Preprocessing for Numerical Data
numerical_transformer = Pipeline(steps = [('scaler', StandardScaler())])

#Data Preprocessing for Categorical Data
categorical_transformer = Pipeline(steps = [('onehot', OneHotEncoder(handle_unknown = 'ignore'))])

#Combine Numerical and Categorical Data
preprocessor = ColumnTransformer(
    transformers = [('num', numerical_transformer, numerical_cols), ('cat', categorical_transformer, categorical_cols)])

#Apply Transformations
x = preprocessor.fit_transform(x)
```

```
#Splitting the Dataset Into Training and Test sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
```

Determining the Best Model

```
#Linear Regression
lr_model = LinearRegression()
lr_model.fit(x_train, y_train)

#Make Predictions on the Test Set
lr_predict = lr_model.predict(x_test)

#Evaluate
print('Linear Regression Model: ')
print(f'Mean Absolute Error : {mean_absolute_error(y_test, lr_predict)}')
print(f'Mean Squared Error : {mean_squared_error(y_test, lr_predict)}')
print(f'R-Squared : {r2_score(y_test, lr_predict)}')
```

```
Linear Regression Model:
Mean Absolute Error : 0.0005110958701100271
Mean Squared Error : 5.401887283915776e-07
R-Squared : 0.9999999991433933
```

Determining the Best Model

```
#Decision Tree
dt_model = DecisionTreeRegressor()
dt_model.fit(x_train, y_train)

#Make Predictions
dt_predict = dt_model.predict(x_test)

#Evaluate
print('Decision Tree Model: ')
print(f'Mean Absolute Error : {mean_absolute_error(y_test, dt_predict)}')
print(f'Mean Squared Error : {mean_squared_error(y_test, dt_predict)}')
print(f'R-Squared : {r2_score(y_test, dt_predict)}')
```

```
Decision Tree Model:
Mean Absolute Error : 0.15687103594080337
Mean Squared Error : 0.4427061310782241
R-Squared : 0.9992979767658952
```

Determining the Best Model

```
#Random Forest
rfr_model = RandomForestRegressor()
rfr_model.fit(x_train, y_train)

#Make Predictions
rfr_predict = rfr_model.predict(x_test)

#Evaluate
print('Random Forest Regressor Model: ')
print(f'Mean Absolute Error : {mean_absolute_error(y_test, rfr_predict)}')
print(f'Mean Squared Error : {mean_squared_error(y_test, rfr_predict)}')
print(f'R-Squared : {r2_score(y_test, rfr_predict)}')
```

```
Random Forest Regressor Model:
Mean Absolute Error : 0.16651585623678655
Mean Squared Error : 0.322604947145874
R-Squared : 0.9994884277572981
```

Determining the Best Model

- The models are assessed with the following metrics: Mean Absolute Error, Mean Squared Error, & R-Squared.
- MAE & MSE reflects the average absolute difference between predicted/actual values.
- R-Squared is a value from 0-1, with higher values indicating better overall performance.
- Based on the results, it is evident that a linear regression model is best for the dataset.

Training Deep Learning Model

```python
#Data Preprocessing
features = data[['Excellent Review %', 'Average Review %', 'Poor Review %']]

#Scale data
scaler = MinMaxScaler(feature_range = (0, 1))
scaled_data = scaler.fit_transform(features)
```

```python
#Prepare Training Data
def create_sequences(data, seq_length):
    xs, ys = [], []

    for i in range(len(data) - seq_length):
        x = data[i:i + seq_length]
        y = data[i + seq_length][2]
        xs.append(x)
        ys.append(y)
    return np.array(xs), np.array(ys)

seq_length = 60
x, y = create_sequences(scaled_data, seq_length)
```

```python
#Split Data Into Training and Test Sets
split = int(0.8 * len(x))
x_train, x_test = x[:split], x[split:]
y_train, y_test = y[:split], y[split:]
```

Training Deep Learning Model

```
#Building the Model
model = Sequential()
model.add(LSTM(units = 50, return_sequences = True, input_shape = (x_train.shape[1], x_train.shape[2])))
model.add(Dropout(0.2))
model.add(LSTM(units = 50, return_sequences = False))
model.add(Dropout(0.2))
model.add(Dense(units = 1))

model.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

```
/opt/anaconda3/lib/python3.11/site-packages/keras/src/layers/rnn/rnn.py:204: UserWarning: Do not pass an `input_shape`/`input_dim` argument t
o a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)
```

```
#Training the Model
history = model.fit(x_train, y_train, epochs = 100, batch_size = 32, validation_split = 0.1)
```

```
Epoch 1/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 11s 32ms/step - loss: 0.0598 - val_loss: 0.0621
Epoch 2/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 8s 30ms/step - loss: 0.0574 - val_loss: 0.0620
Epoch 3/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 8s 30ms/step - loss: 0.0579 - val_loss: 0.0630
Epoch 4/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 8s 30ms/step - loss: 0.0572 - val_loss: 0.0620
Epoch 5/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 8s 31ms/step - loss: 0.0568 - val_loss: 0.0620
Epoch 6/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 8s 30ms/step - loss: 0.0574 - val_loss: 0.0620
Epoch 7/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 8s 31ms/step - loss: 0.0563 - val_loss: 0.0620
Epoch 8/100
265/265 ━━━━━━━━━━━━━━━━━━━━ 8s 31ms/step - loss: 0.0584 - val_loss: 0.0619
```

Evaluating the Model

```
#Evaluating the Model
predicted_review = model.predict(x_test)
predicted_review = scaler.inverse_transform(np.concatenate((np.zeros((predicted_review.shape[0], 2)), predicted_review), axis = 1))[:, 2]

#Inverse transform the actual medals
actual_review = scaler.inverse_transform(np.concatenate((np.zeros((y_test.shape[0], 2)), y_test.reshape(-1, 1)), axis = 1))[:, 2]
```

```
74/74 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step
```

```
#Calculate Performance Metrics
mae = mean_absolute_error(actual_review, predicted_review)
mse = mean_squared_error(actual_review, predicted_review)
r2 = r2_score(actual_review, predicted_review)

print(f'MSE: {mse}')
print(f'MAE: {mae}')
print(f'R2: {r2}')
```

```
MSE: 595.3192512063712
MAE: 18.575428377097584
R2: -0.002433855535449947
```

Conclusion

- **From the data, Sucralfate, Thyroxine, & Vitamin D3 contributed most to patient satisfaction.**
- **Additionally, application site reactions, hypoglycemia, & dizziness/tiredness are the most common side effects.**
- **Considering the evaluation of the deep learning model, further optimizations can & should be made my adjusting parameters.**