Data Glacier

Your Deep Learning Partner

Healthcare

Persistency of a Drug

12th May 2023

Data Glacier

Your Deep Learning Partner

Group Name: One

Name: Hira Fahim

Email: hirashahidd26@yahoo.com

Country: United Kingdom

Company: Unemployed

Specialization: Data Science

Batch Code: LISUM19

Submission Date: 12th May 2023

Submitted to: Data Glacier

Background – Healthcare – Persistency of the Drug

Problem Statement:

One of the challenge for all Pharmaceutical companies is to understand the persistency of drug as per the physician prescription. To solve this problem ABC pharma company approached an analytics company to automate this process of identification.

ML Problem:

With an objective to gather insights on the factors that are impacting the persistency, build a classification for the given dataset.

Approach

The analysis has been divided into following parts:

- Data Understanding
- Exploratory Data Analysis
- Univariate Analysis
- Bivariate Analysis
- Feature Engineering
- Feature Selection
- Model Building
- Model Evaluation
- Model Selection
- Model Deployment

Data Understanding

Dataset Information

The data is related to Pharmaceutical companies and to understand the persistency of drug as per the physician prescription. The dataset is based on demographic details of patients, provider attributes, clinical factors and disease and treatment factors.

Datatype of columns and Non-Null values

```
In [4]: # Datatypes of columns and non-null values
        d.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3424 entries, 0 to 3423
Data columns (total 69 columns):
 #   Column                                                              Non-Null Count  Dtype
---  ------                                                              --------------  -----
 0   Ptid                                                                3424 non-null   object
 1   Persistency_Flag                                                    3424 non-null   object
 2   Gender                                                              3424 non-null   object
 3   Race                                                                3424 non-null   object
 4   Ethnicity                                                           3424 non-null   object
 5   Region                                                              3424 non-null   object
 6   Age_Bucket                                                          3424 non-null   object
 7   Ntm_Speciality                                                      3424 non-null   object
 8   Ntm_Specialist_Flag                                                 3424 non-null   object
 9   Ntm_Speciality_Bucket                                               3424 non-null   object
 10  Gluco_Record_Prior_Ntm                                              3424 non-null   object
 11  Gluco_Record_During_Rx                                              3424 non-null   object
 12  Dexa_Freq_During_Rx                                                 3424 non-null   int64
 13  Dexa_During_Rx                                                      3424 non-null   object
 14  Frag_Frac_Prior_Ntm                                                 3424 non-null   object
 15  Frag_Frac_During_Rx                                                 3424 non-null   object
 16  Risk_Segment_Prior_Ntm                                              3424 non-null   object
 17  Tscore_Bucket_Prior_Ntm                                             3424 non-null   object
 18  Risk_Segment_During_Rx                                              3424 non-null   object
 19  Tscore_Bucket_During_Rx                                             3424 non-null   object
 20  Change_T_Score                                                      3424 non-null   object
 21  Change_Risk_Segment                                                 3424 non-null   object
 22  Adherent_Flag                                                       3424 non-null   object
 23  Idn_Indicator                                                       3424 non-null   object
 24  Injectable_Experience_During_Rx                                     3424 non-null   object
 25  Comorb_Encounter_For_Screening_For_Malignant_Neoplasms              3424 non-null   object
 26  Comorb_Encounter_For_Immunization                                   3424 non-null   object
 27  Comorb_Encntr_For_General_Exam_W_O_Complaint,_Susp_Or_Reprtd_Dx     3424 non-null   object
 28  Comorb_Vitamin_D_Deficiency                                         3424 non-null   object
 29  Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified                3424 non-null   object
 30  Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx  3424 non-null   object
 31  Comorb_Long_Term_Current_Drug_Therapy                               3424 non-null   object
```

```
 32  Comorb_Dorsalgia                                                    3424 non-null   object
 33  Comorb_Personal_History_Of_Other_Diseases_And_Conditions            3424 non-null   object
 34  Comorb_Other_Disorders_Of_Bone_Density_And_Structure                3424 non-null   object
 35  Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias     3424 non-null   object
 36  Comorb_Osteoporosis_without_current_pathological_fracture           3424 non-null   object
 37  Comorb_Personal_history_of_malignant_neoplasm                       3424 non-null   object
 38  Comorb_Gastro_esophageal_reflux_disease                             3424 non-null   object
 39  Concom_Cholesterol_And_Triglyceride_Regulating_Preparations         3424 non-null   object
 40  Concom_Narcotics                                                    3424 non-null   object
 41  Concom_Systemic_Corticosteroids_Plain                               3424 non-null   object
 42  Concom_Anti_Depressants_And_Mood_Stabilisers                        3424 non-null   object
 43  Concom_Fluoroquinolones                                             3424 non-null   object
 44  Concom_Cephalosporins                                               3424 non-null   object
 45  Concom_Macrolides_And_Similar_Types                                 3424 non-null   object
 46  Concom_Broad_Spectrum_Penicillins                                   3424 non-null   object
 47  Concom_Anaesthetics_General                                         3424 non-null   object
 48  Concom_Viral_Vaccines                                               3424 non-null   object
 49  Risk_Type_1_Insulin_Dependent_Diabetes                              3424 non-null   object
 50  Risk_Osteogenesis_Imperfecta                                        3424 non-null   object
 51  Risk_Rheumatoid_Arthritis                                           3424 non-null   object
 52  Risk_Untreated_Chronic_Hyperthyroidism                              3424 non-null   object
 53  Risk_Untreated_Chronic_Hypogonadism                                 3424 non-null   object
 54  Risk_Untreated_Early_Menopause                                      3424 non-null   object
 55  Risk_Patient_Parent_Fractured_Their_Hip                             3424 non-null   object
 56  Risk_Smoking_Tobacco                                                3424 non-null   object
 57  Risk_Chronic_Malnutrition_Or_Malabsorption                          3424 non-null   object
 58  Risk_Chronic_Liver_Disease                                          3424 non-null   object
 59  Risk_Family_History_Of_Osteoporosis                                 3424 non-null   object
 60  Risk_Low_Calcium_Intake                                             3424 non-null   object
 61  Risk_Vitamin_D_Insufficiency                                        3424 non-null   object
 62  Risk_Poor_Health_Frailty                                            3424 non-null   object
 63  Risk_Excessive_Thinness                                             3424 non-null   object
 64  Risk_Hysterectomy_Oophorectomy                                      3424 non-null   object
 65  Risk_Estrogen_Deficiency                                            3424 non-null   object
 66  Risk_Immobilization                                                 3424 non-null   object
 67  Risk_Recurring_Falls                                                3424 non-null   object
 68  Count_Of_Risks                                                      3424 non-null   int64
dtypes: int64(2), object(67)
memory usage: 1.8+ MB
```

Hira Fahim

Data Understanding

Numerical and Categorical Features

```
In [7]: display_numeric_categoric_feature(d)

Numeric Features:
Index(['Dexa_Freq_During_Rx', 'Count_Of_Risks'], dtype='object')
=====================================================================================
Categorical Features:
Index(['Ptid', 'Persistency_Flag', 'Gender', 'Race', 'Ethnicity', 'Region',
       'Age_Bucket', 'Ntm_Speciality', 'Ntm_Specialist_Flag',
       'Ntm_Speciality_Bucket', 'Gluco_Record_Prior_Ntm',
       'Gluco_Record_During_Rx', 'Dexa_During_Rx', 'Frag_Frac_Prior_Ntm',
       'Frag_Frac_During_Rx', 'Risk_Segment_Prior_Ntm',
       'Tscore_Bucket_Prior_Ntm', 'Risk_Segment_During_Rx',
       'Tscore_Bucket_During_Rx', 'Change_T_Score', 'Change_Risk_Segment',
       'Adherent_Flag', 'Idn_Indicator', 'Injectable_Experience_During_Rx',
       'Comorb_Encounter_For_Screening_For_Malignant_Neoplasms',
       'Comorb_Encounter_For_Immunization',
       'Comorb_Encntr_For_General_Exam_W_O_Complaint,_Susp_Or_Reprtd_Dx',
       'Comorb_Vitamin_D_Deficiency',
       'Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified',
       'Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx',
       'Comorb_Long_Term_Current_Drug_Therapy', 'Comorb_Dorsalgia',
       'Comorb_Personal_History_Of_Other_Diseases_And_Conditions',
       'Comorb_Other_Disorders_Of_Bone_Density_And_Structure',
       'Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias',
       'Comorb_Osteoporosis_without_current_pathological_fracture',
       'Comorb_Personal_history_of_malignant_neoplasm',
       'Comorb_Gastro_esophageal_reflux_disease',
       'Concom_Cholesterol_And_Triglyceride_Regulating_Preparations',
       'Concom_Narcotics', 'Concom_Systemic_Corticosteroids_Plain',
       'Concom_Anti_Depressants_And_Mood_Stabilisers',
       'Concom_Fluoroquinolones', 'Concom_Cephalosporins',
       'Concom_Macrolides_And_Similar_Types',
       'Concom_Broad_Spectrum_Penicillins', 'Concom_Anaesthetics_General',
       'Concom_Viral_Vaccines', 'Risk_Type_1_Insulin_Dependent_Diabetes',
       'Risk_Osteogenesis_Imperfecta', 'Risk_Rheumatoid_Arthritis',
       'Risk_Untreated_Chronic_Hyperthyroidism',
       'Risk_Untreated_Chronic_Hypogonadism', 'Risk_Untreated_Early_Menopause',
       'Risk_Patient_Parent_Fractured_Their_Hip', 'Risk_Smoking_Tobacco',
       'Risk_Chronic_Malnutrition_Or_Malabsorption',
       'Risk_Chronic_Liver_Disease', 'Risk_Family_History_Of_Osteoporosis',
       'Risk_Low_Calcium_Intake', 'Risk_Vitamin_D_Insufficiency',
       'Risk_Poor_Health_Frailty', 'Risk_Excessive_Thinness',
       'Risk_Hysterectomy_Oophorectomy', 'Risk_Estrogen_Deficiency',
       'Risk_Immobilization', 'Risk_Recurring_Falls'],
      dtype='object')
```

Null values

```
In [8]: # total null values in the dataset
        d.isnull().sum()

Out[8]: Ptid                              0
        Persistency_Flag                  0
        Gender                            0
        Race                              0
        Ethnicity                         0
                                         ..
        Risk_Hysterectomy_Oophorectomy    0
        Risk_Estrogen_Deficiency          0
        Risk_Immobilization               0
        Risk_Recurring_Falls              0
        Count_Of_Risks                    0
        Length: 69, dtype: int64
```

```
In [9]: d.isna().apply(pd.value_counts)
```

Out[9]:

	Ptid	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Speciality	Ntm_Specialist_Flag	Ntm_Speciality_Bucket	...	Risk_Family_Histo
False	3424	3424	3424	3424	3424	3424	3424	3424	3424	3424	...	

1 rows × 69 columns

There is no null values in the dataset.

Hira Fahim

Exploratory Data Analysis

Step:1 Drop Duplicate Rows

```
In [12]: # Remove duplicate rows
d=d.drop_duplicates()
d
```

Out[12]:

	Ptid	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Speciality	Ntm_Specialist_Flag	Ntm_Speciality_Bucket	...
0	P1	Persistent	Male	Caucasian	Not Hispanic	West	>75	GENERAL PRACTITIONER	Others	OB/GYN/Others/PCP/Unknown	...
1	P2	Non-Persistent	Male	Asian	Not Hispanic	West	55-65	GENERAL PRACTITIONER	Others	OB/GYN/Others/PCP/Unknown	...
2	P3	Non-Persistent	Female	Other/Unknown	Hispanic	Midwest	65-75	GENERAL PRACTITIONER	Others	OB/GYN/Others/PCP/Unknown	...
3	P4	Non-Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	GENERAL PRACTITIONER	Others	OB/GYN/Others/PCP/Unknown	...
4	P5	Non-Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	GENERAL PRACTITIONER	Others	OB/GYN/Others/PCP/Unknown	...
...	...	...	...	...	...	...	...	...	...	...	...
3419	P3420	Persistent	Female	Caucasian	Not Hispanic	South	>75	GENERAL PRACTITIONER	Others	OB/GYN/Others/PCP/Unknown	...
3420	P3421	Persistent	Female	Caucasian	Not Hispanic	South	>75	Unknown	Others	OB/GYN/Others/PCP/Unknown	...
3421	P3422	Persistent	Female	Caucasian	Not Hispanic	South	>75	ENDOCRINOLOGY	Specialist	Endo/Onc/Uro	...
3422	P3423	Non-Persistent	Female	Caucasian	Not Hispanic	South	55-65	Unknown	Others	OB/GYN/Others/PCP/Unknown	...
3423	P3424	Non-Persistent	Female	Caucasian	Not Hispanic	South	65-75	Unknown	Others	OB/GYN/Others/PCP/Unknown	...

3424 rows × 69 columns

There is no duplicate rows in the dataset.

Step:2 Drop Unnecessary Column

```
In [13]: #Drop ptid column
d.drop(columns='Ptid',axis=1,inplace=True)
```

'Ptid' has all unique values .

Univariate Analysis

Description of the data

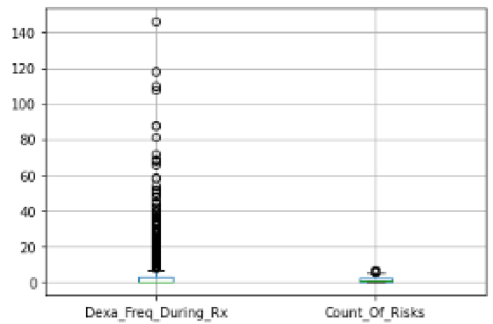
```
# Description of numerical columns
d.describe()
```

	Dexa_Freq_During_Rx	Count_Of_Risks
count	3424.000000	3424.000000
mean	3.016063	1.239486
std	8.136545	1.094914
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	1.000000
75%	3.000000	2.000000
max	146.000000	7.000000

From **description** and **boxplot**, we can see there are outliers in numerical input variables. **Histogram** shows uneven distribution and positively skewed data in both numerical variables.

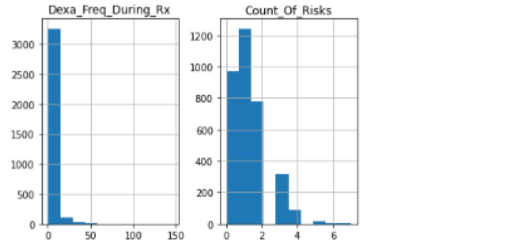
Visualization (boxplot) of Numerical Attributes

```
In [16]: d.boxplot()
Out[16]: <AxesSubplot:>
```



Histogram for Numerical Attributes

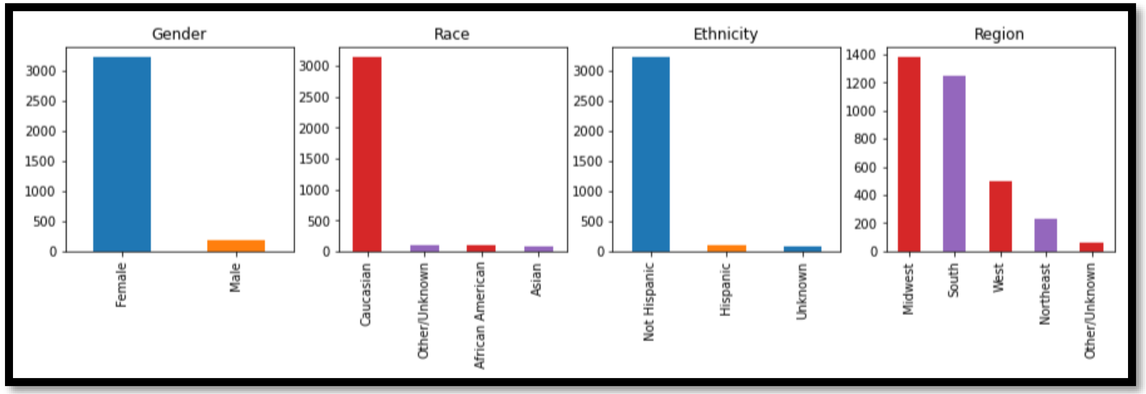
```
In [15]: d.hist()
Out[15]: array([[<AxesSubplot:title={'center':'Dexa_Freq_During_Rx'}>,
        <AxesSubplot:title={'center':'Count_Of_Risks'}>]], dtype=object)
```

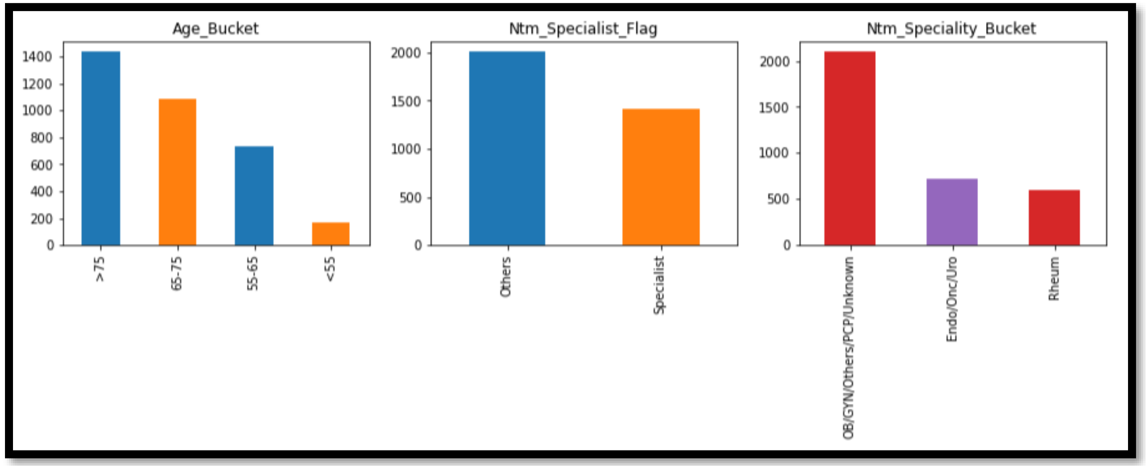


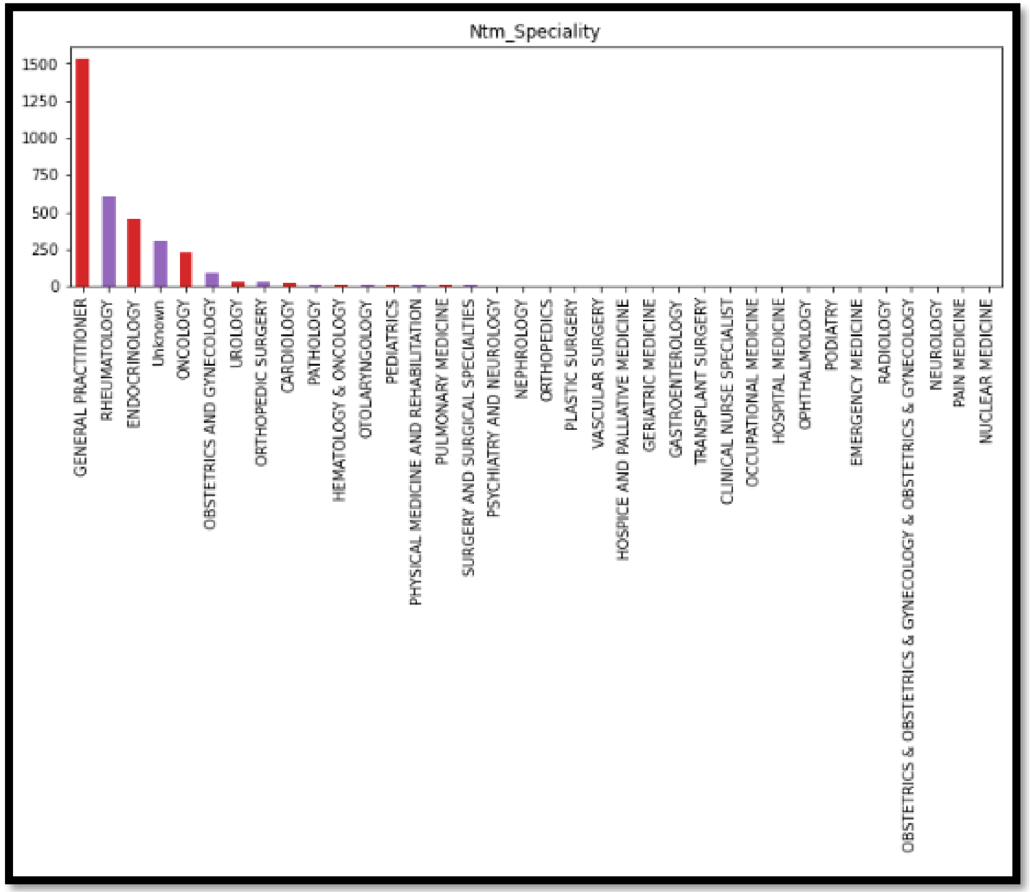
Hira Fahim

Univariate Analysis

Visualization of Categorical Attributes



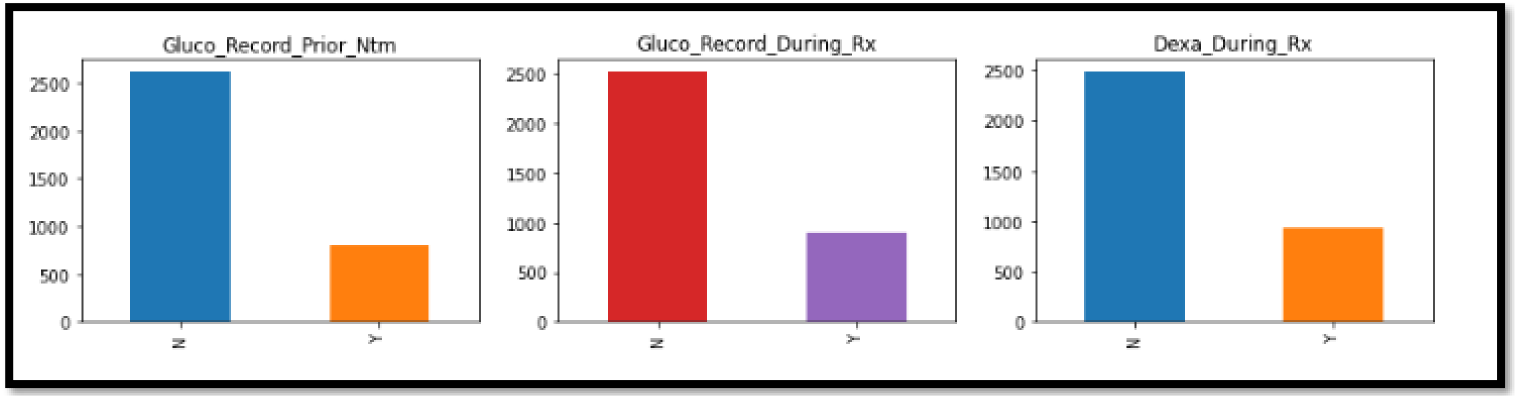


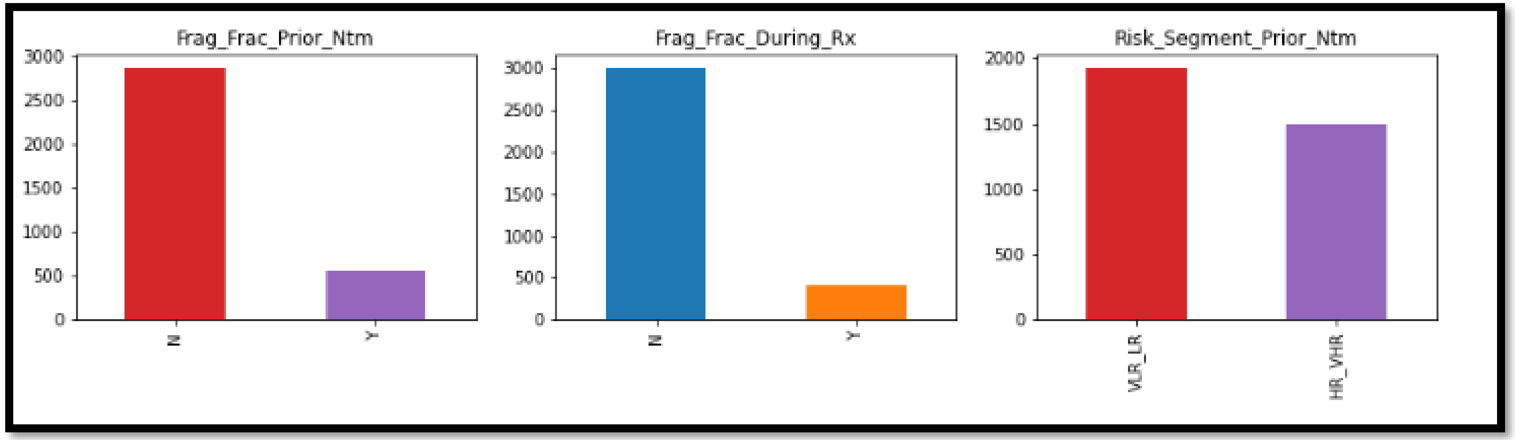


In **Bar chart** of categorical columns, we see uneven distribution of data in all the categorical columns.

Univariate Analysis

Visualization of Categorical Attributes





In **Bar chart** of categorical columns, we see uneven distribution of data in all the categorical columns.

Hira Fahim

Bivariate Analysis

Persistency of the drug based on Gender

Persistency of the drug based on Race

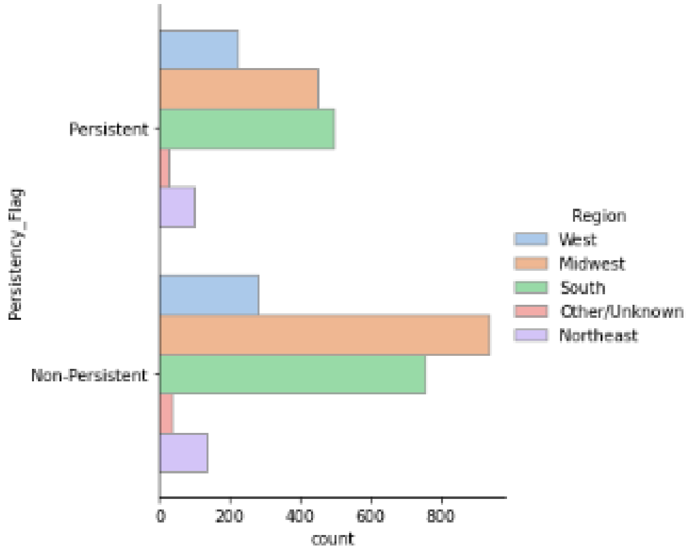
Persistency of the drug based on Ethnicity

There is no significant insight through bivariate Analysis.

Bivariate Analysis

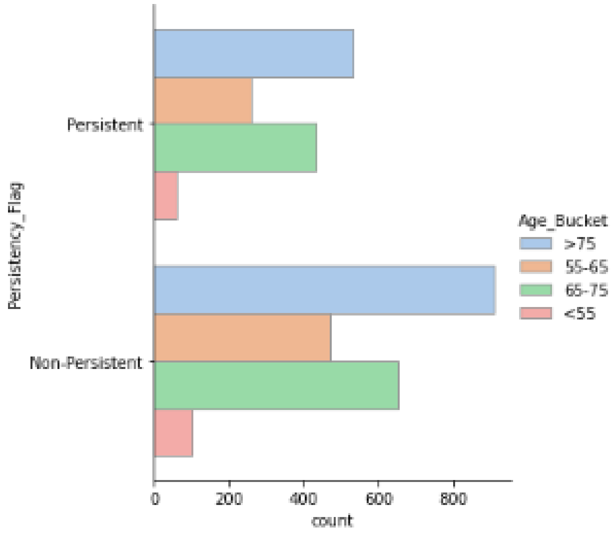
Persistency of the drug based on Region

```
In [31]: sns.catplot(
             data=d, y="Persistency_Flag", hue="Region", kind="count",
             palette="pastel", edgecolor=".6",)

Out[31]: <seaborn.axisgrid.FacetGrid at 0x24c3ff10460>
```



Persistency of the drug based on Age

```
In [32]: sns.catplot(
             data=d, y="Persistency_Flag", hue="Age_Bucket", kind="count",
             palette="pastel", edgecolor=".6",)

Out[32]: <seaborn.axisgrid.FacetGrid at 0x24c4007a490>
```



There is no significant insight through bivariate Analysis.

Feature Engineering

Convert datatype of Categorical Features

```
In [35]: # Convert object to category
         d[categorical_columns]=d[categorical_columns].astype("category")
```

Encode Categorical Columns into Numerical

```
In [37]: # encoding categorical features into numeric
         d[categorical_columns]=d[categorical_columns].apply(lambda x: x.cat.codes)
```

Feature Selection

Divide independent variables and target variable

```
In [39]: x=d[categorical_columns].drop(columns=['Persistency_Flag'])
         y=d['Persistency_Flag']
```

Categorical Feature Selection using sklearn library and chi2 and SelectKbest function.

```
In [40]: from sklearn.feature_selection import chi2, SelectKBest

In [41]: cs= SelectKBest (score_func = chi2, k= "all")
         cs.fit(x,y)
         feature_score = pd.DataFrame({"Score":cs.scores_, "P_Values": cs.pvalues_},index = x.columns)
         feature_score.nlargest(n=61, columns="Score")

Out[41]:
```

	Score	P_Values
Dexa_During_Rx	601.821735	6.722923e-133
Comorb_Long_Term_Current_Drug_Therapy	324.413431	1.583364e-72
Comorb_Encounter_For_Screening_For_Malignant_Neoplasms	196.456669	1.239081e-44
Comorb_Encounter_For_Immunization	189.482869	4.122973e-43
Comorb_Other_Disorders_Of_Bone_Density_And_Structure	177.698458	1.541551e-40
...	...	...
Risk_Untreated_Early_Menopause	0.095081	7.578140e-01
Gluco_Record_Prior_Ntm	0.086825	7.682533e-01
Risk_Family_History_Of_Osteoporosis	0.037398	8.466578e-01
Risk_Osteogenesis_Imperfecta	0.023770	8.774707e-01
Age_Bucket	0.011883	9.131938e-01

61 rows × 2 columns

Feature Selection

Eliminate Categorical features with less or no relationship with target variable considering the p-value> 0.05

```
In [42]: #drop categoricalcolumns with less or no relationship with traget varibale
         remove_columns = ['Ntm_Speciality','Gender','Risk_Low_Calcium_Intake','Risk_Segment_Prior_Ntm',
                           'Risk_Patient_Parent_Fractured_Their_Hip','Change_Risk_Segment','Risk_Untreated_Early_Menopause',
                           'Gluco_Record_Prior_Ntm','Risk_Family_History_Of_Osteoporosis','Risk_Osteogenesis_Imperfecta','Age_Bucket',
                           'Race','Risk_Segment_During_Rx','Ethnicity','Frag_Frac_Prior_Ntm']

         d.drop(columns=remove_columns,inplace=True)
```

Final dataset ready for Modelling

```
In [43]: d.sample(10)
```

Out[43]:

	Persistency_Flag	Region	Ntm_Specialist_Flag	Ntm_Speciality_Bucket	Gluco_Record_During_Rx	Dexa_Freq_During_Rx	Dexa_During_Rx	Frag_Frac_Durin
2632	1	0	0	1	0	0	0	
908	1	3	1	2	0	0	0	
723	0	0	0	1	0	0	0	
138	0	0	0	1	0	0	0	
2077	0	4	0	1	0	0	0	
2008	0	3	0	1	0	0	0	
1780	1	4	1	1	0	0	0	
1624	1	2	0	1	0	0	0	
3332	0	1	1	0	0	0	0	
3013	1	0	1	2	1	8	1	

10 rows × 53 columns

Model Building

1. The dataset is imbalance, so we will balance the dataset using SMOTE.
2. Divide the dataset into input variables and output variable then split the input and output into train and test sets (20% test and 80% train).
3. Different Machine Learning models to predict the persistency of the drug:
    - Logistic Regression
    - Random Forest
    - K- Nearest Neighbor
    - Gradient Boosting

Model Evaluation

Metrics of Evaluation

1. Accuracy, Precision, Recall and F1-Score
2. Scores of Test, Train and Complete dataset
3. Confusion Matrix
4. Lift and Gain
5. KS Statistics and ROC-AUC Score

Model Selection

Model selection based on Scores and Confusion Matrix

Model	Score All Dataset	Score Train Dataset	Score Test Dataset	TN	FP	FN	TP
Logistic Regression	0.8022	0.785	0.774	335	83	110	326
Random Forest	0.8028	0.768	0.752	369	49	163	273
KNNC	0.837	0.8635	0.787	332	86	96	340
KNNC + Hyperparameter Tuning	0.837	0.863	0.787	332	86	96	340
Gradient Boosting	0.857	0.868	0.77	333	85	111	325
Gradient Boosting + Hyperparameter Tuning	0.943	0.962	0.803	343	75	93	343

The highest score for all dataset is of Gradient Boosting with hyperparameter tuning. But the difference between the score train dataset and test dataset is 16% which means its over fitting model. For the persistency of the drug the FN can cost a lot to the healthcare business, so it is needed to be low. KNNC and KNNC with hyperparameter tuning have low FN. KNNC is the best model. Let's check other metrics for evaluation.

Model Selection

Model selection based on Lift and Gain Curve

Cumulative gains and lift charts are visual aids for measuring model performance.
The Greater the area between the Lift / Gain and Baseline, the Better the model.
By analysing Gain and Lift Curve, Random Forest Classification Model, KNNC model and Gradient Boosting Classification model with Hyperparameter Tuning are the best models.

Model Selection

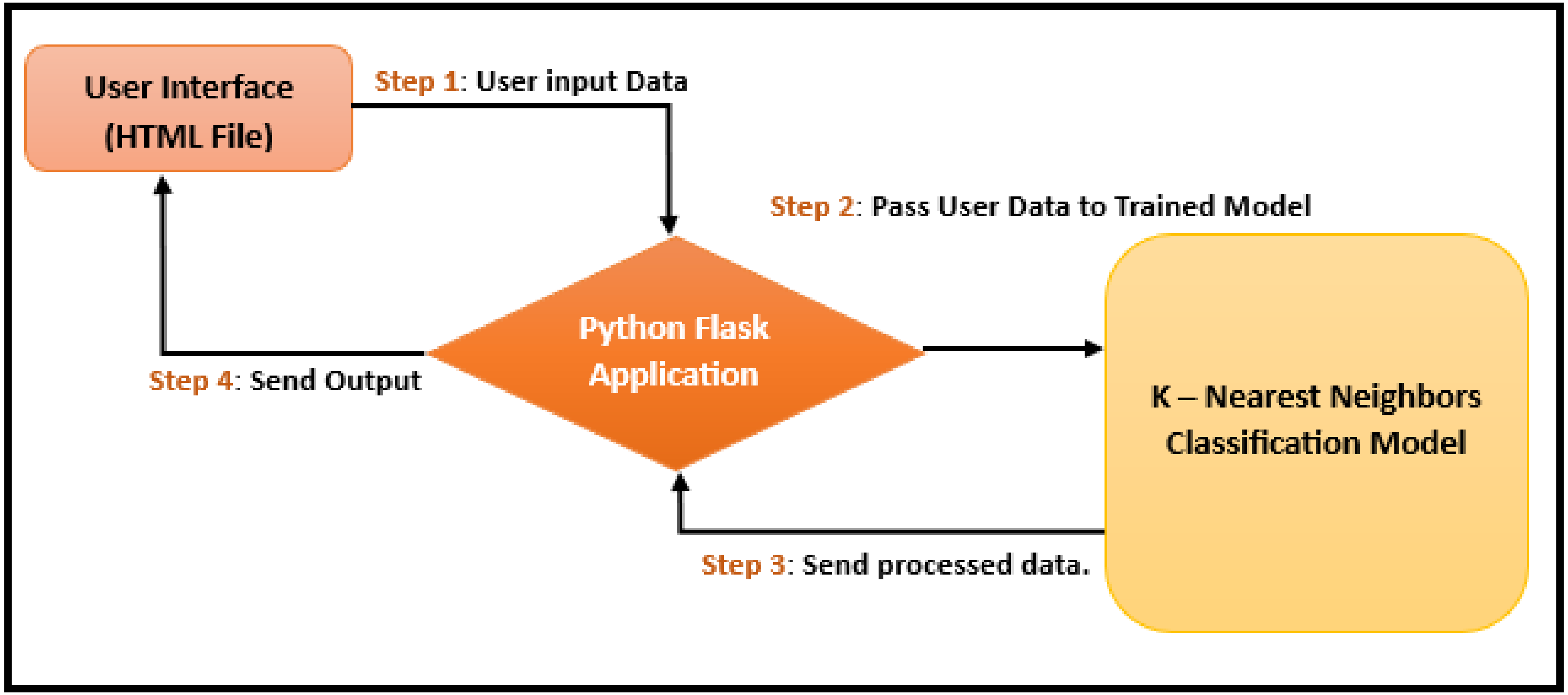
Model selection based on Accuracy, Precision, Recall, F1 Score, KS statistics and AUC-ROC

Model	Precision	Recall	F1 score	Accuracy	KS Statics	AUC- ROC
Logistic Regression	0 - 0.75 1 - 0.80	0.80 0.75	0.78 0.77	0.77	0.565	0.85
Random Forest	0 – 0.69 1 – 0.85	0.88 0.63	0.78 0.72	0.75	0.556	0.844
KNNC	0 – 0.78 1 – 0.80	0.79 0.78	0.78 0.79	0.79	0.574	0.85
KNNC + Hyperparameter Tuning	0 – 0.78 1 – 0.80	0.79 0.78	0.78 0.79	0.79	0.574	0.85

The F1- Score, KS statistics and AUC-ROC metrics of KNNC model are the best. Therefore, **The best model for deployment is K- Nearest Neighbor Classification model.**

Model Deployment

Given Workflow shows K- Nearest Neighbors Classifier model is used and Flask Framework for deployment. It represents the details of how the model works from user interface till the results.

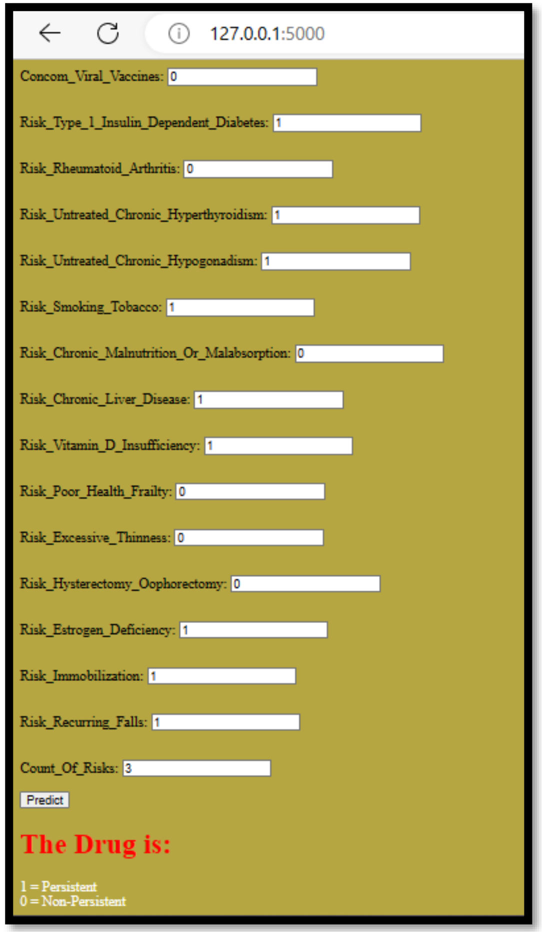


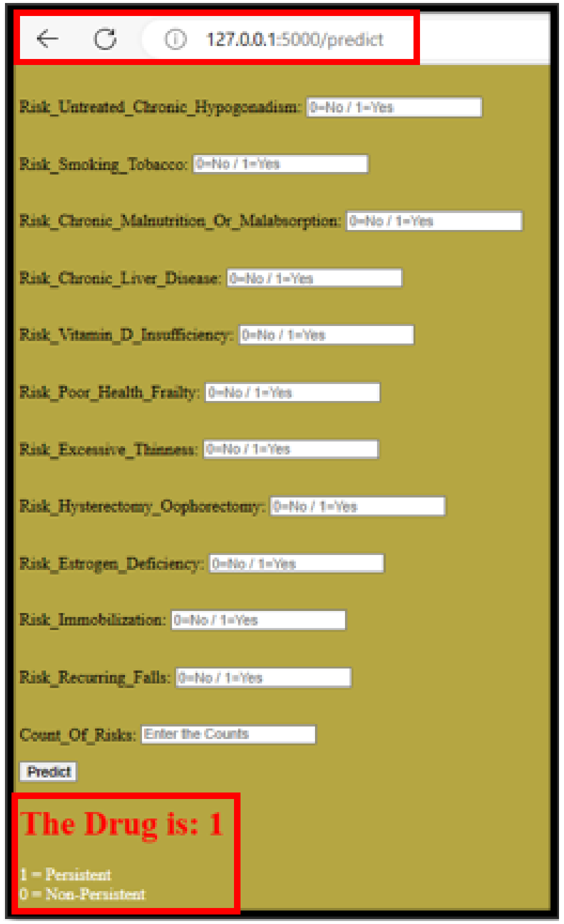
Model Deployment

1. Save the model using Pickle.
2. Deploy the model using Flask framework.
3. The app.py file contains the source code including the ML code for prediction and will be execute by the Python interpreter to run the Flask web application.
4. The Index.html file will render a text form where a user enter the details of required fields. Index.html file will be rendered via the render_template ('index.html', prediction_text="{}".format(output)), which is inside the predict function of app.py script to display the output as per the input submitted by the user.
5. The URL generate by 'app.py.' Open a web browser and navigate to http://127.0.0.1:5000/ following is output of Index.html.

```python
import numpy as np
from flask import Flask, request, render_template
import pickle

#Create an app object using the Flask class.
app = Flask(__name__)

#Load the trained model. (Pickle file)
model = pickle.load(open('models/model.pkl', 'rb'))

#Define the route to be home.
#The decorator below links the relative route of the URL to the function it is decorating.
#Here, home function is with '/', our root directory.
#Running the app sends us to index.html.
#Note that render_template means it looks for the file in the templates folder.

#use the route() decorator to tell Flask what URL should trigger our function.
@app.route('/')
def home():
  return render_template('index.html')

#You can use the methods argument of the route() decorator to handle different HTTP methods.
#GET: A GET message is send, and the server returns data
#POST: Used to send HTML form data to the server.
#Add Post method to the decorator to allow for form submission.
#Redirect to /predict page with the output
@app.route('/predict',methods=['POST'])
def predict():

    int_features = [float(x) for x in request.form.values()] #Convert string inputs to float.
    features = [np.array(int_features)]  #Convert to the form [[a, b,c]] for input to the model
    prediction = model.predict(features)  # features Must be in the form [[a, b,c]]

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text="{}".format(output))


#When the Python interpreter reads a source file, it first defines a few special variables.
#For now, we care about the __name__ variable.
#If we execute our code in the main program, like in our case here, it assigns
# __main__ as the name (__name__).
#So if we want to run our code right here, we can check if __name__ == __main__
#if so, execute it here.
#If we import this file (module) to another file then __name__ == app (which is the name of this python file).

if __name__ == "__main__":
    app.run()
```

Model Deployment

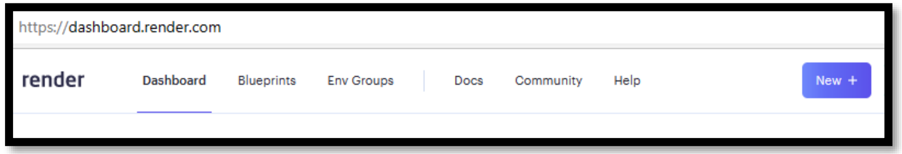


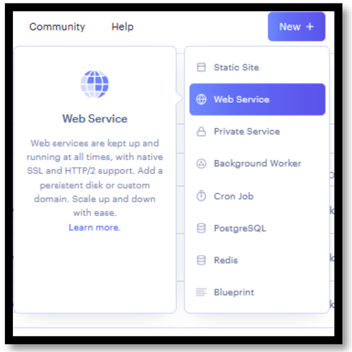


Select categorical fields as per their respective number in the given code and click the Predict button. The predicted result will be displayed at the bottom of the web page.

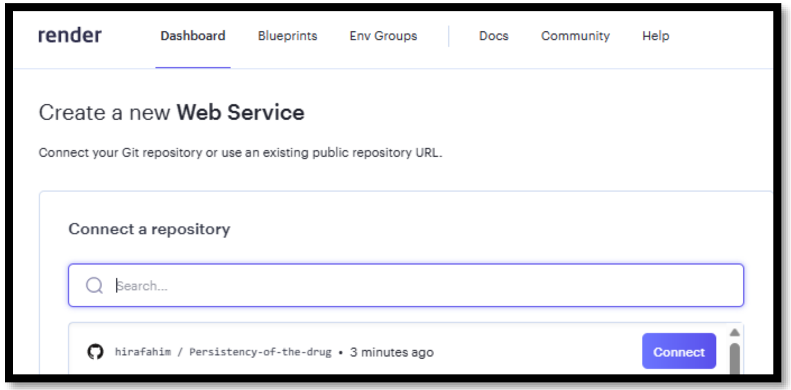
Model Deployment on Render (Open-Source Cloud Deployment)

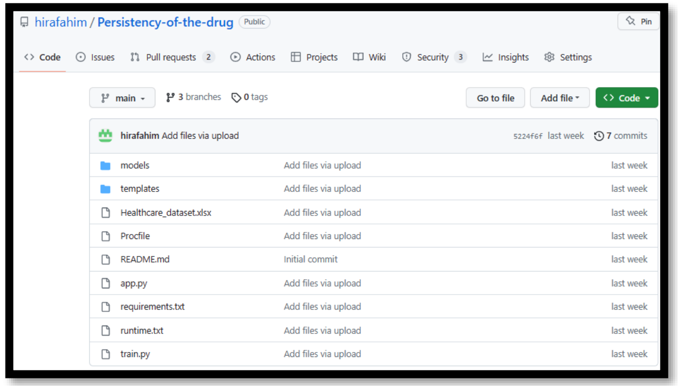
➢ After the model has been trained and deployed locally, now it is ready for deploy on open-source cloud "Render".





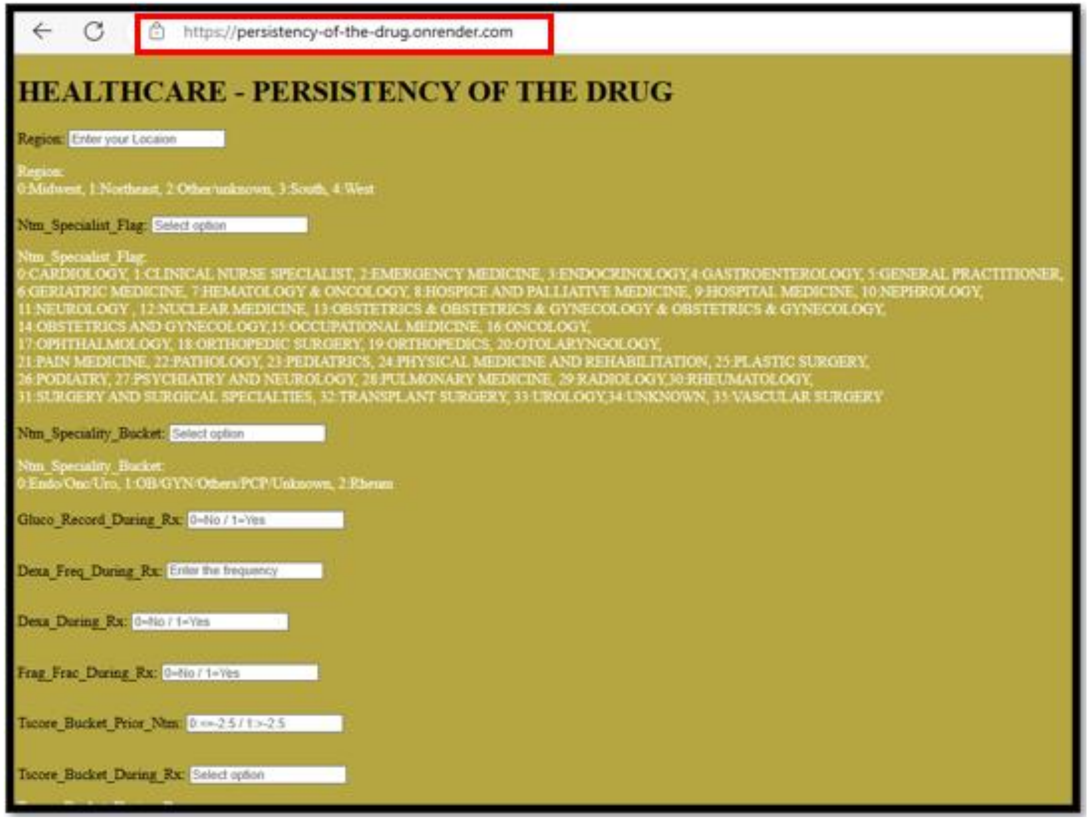
➢ Connect web service to GitHub Repository.

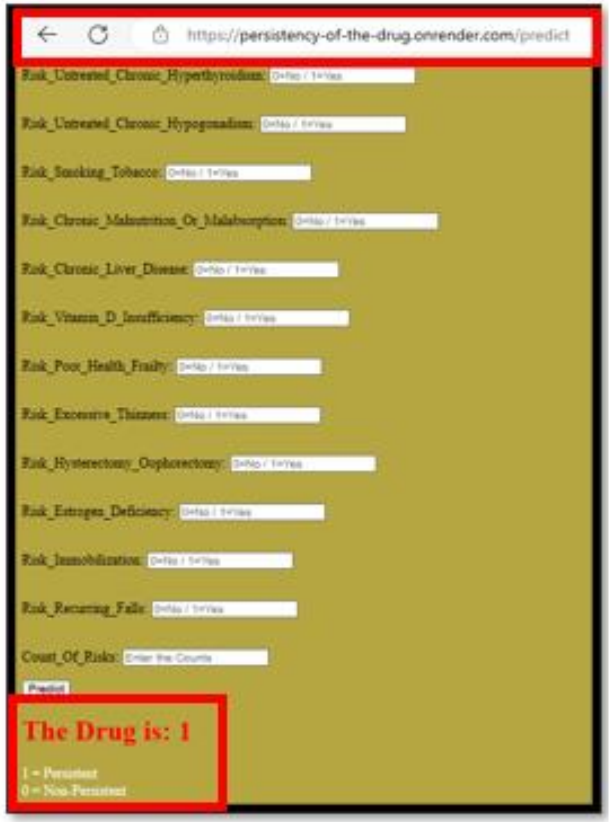




➢ Click and open the application for persistency of the drug. https://persistency-of-the-drug.onrender.com

Model Deployment on Render (Open-Source Cloud Deployment)





Select categorical fields as per their respective number in the given code and click the Predict button. The predicted result will be displayed at the bottom of the web page.

Challenges

- Feature selection was a challenging task, which is done by Chi2 from sklearn.feature_selection library.

- Selection of best model was also tricky but after carefully considering all parameters and metrics of evaluation choose ‘K-Nearest Neighbor Classification model’ as the best model.

Thank You

Data Glacier
Your Deep Learning Partner

Hira Fahim