Data Science Intern at Data Glacier

Project: Healthcare - Persistency of a drug

Week 12: Deliverables

Name: Chenyu Wang

Email: ryan.wang0924@gmail.com

University: University of Ottawa

Country: Canada

Specialization: Data Science

Batch code: LISUM 13:30

Submission date: Nov 23, 2022

Submitted to: Data Glacier

1. **Problem description:**

   One of the challenges 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.

2. **Problem understanding:**

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

3. **Project lifecycle:**

Weeks	Due Date	Task
Week 7	19 Oct, 2022	● Problem understanding ● data intake report ● Data Understanding
Week 8	26 Oct, 2022	● Data Cleaning and Feature engineering
Week 9	02 Nov, 2022	● Model Development
Week 10	9 Nov, 2022	● Model Selection ● Model Evaluation
Week 11	16 Nov, 2022	● Report the accuracy, precision and recall of both the class of target variable ● Report ROC-AUC as well
Week 12	23 Nov, 2022	● Deploy the model
Week 13	30 Nov, 2022	● Final Submission (Report + Code + Presentation)

4. GitHub Repo link

   https://github.com/chenyu-wang55/Data_Scientist_Intern_Data_Glacier/tree/main/Healthcare_project

5. Data Report

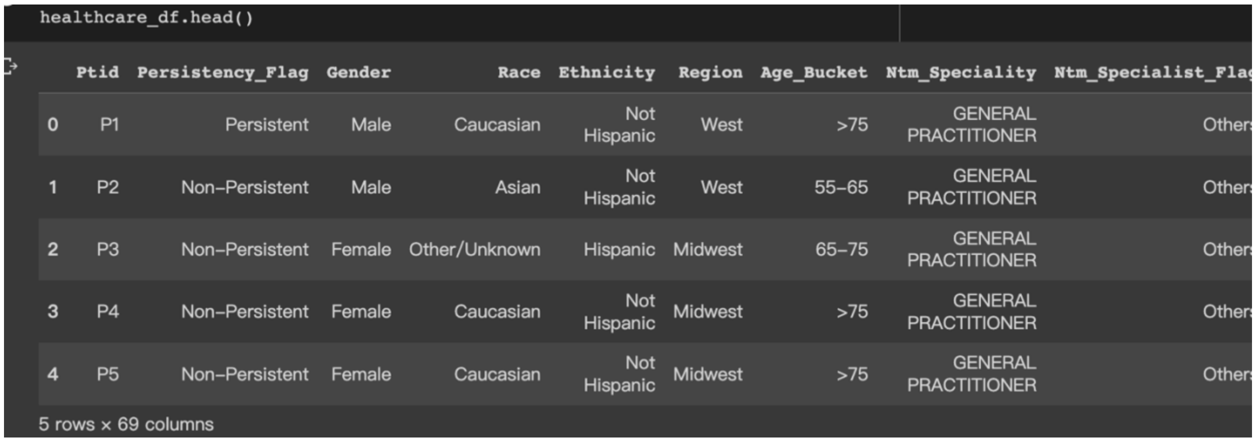
This dataset about the persistency of drug which contains 69 features and 3424 observations. The target feature in this dataset is ‘Persistency_Flag’ which classify the dataset as persistent and non-persistent.

Tabular data details: Healthcare Data

Total number of observations	3424
Total number of files	1
Total number of features	69
Base format of the file	xlsx
Size of the data	1.8 MB

6. Data Understanding & Cleaning

6.1 Used the head function in Pandas package to display the top 5 data records.

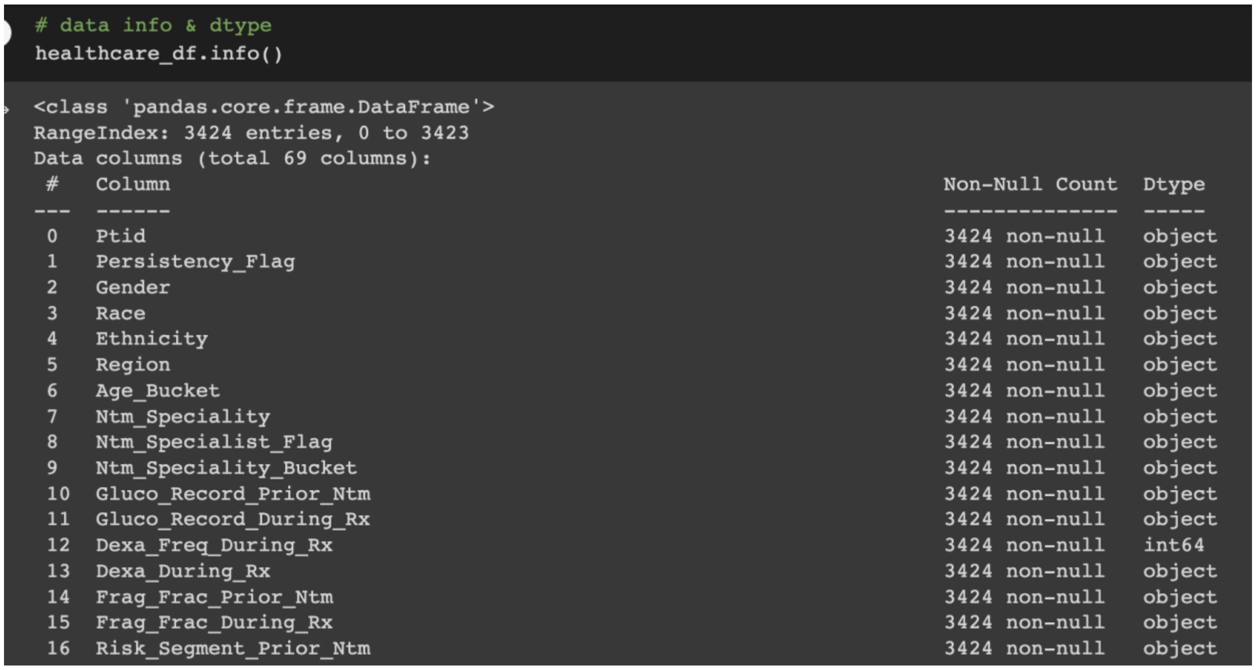


```
healthcare_df.head()
```

	Ptid	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Speciality	Ntm_Specialist_Fla
0	P1	Persistent	Male	Caucasian	Not Hispanic	West	>75	GENERAL PRACTITIONER	Other
1	P2	Non–Persistent	Male	Asian	Not Hispanic	West	55–65	GENERAL PRACTITIONER	Other
2	P3	Non–Persistent	Female	Other/Unknown	Hispanic	Midwest	65–75	GENERAL PRACTITIONER	Other
3	P4	Non–Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	GENERAL PRACTITIONER	Other
4	P5	Non–Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	GENERAL PRACTITIONER	Other

5 rows × 69 columns

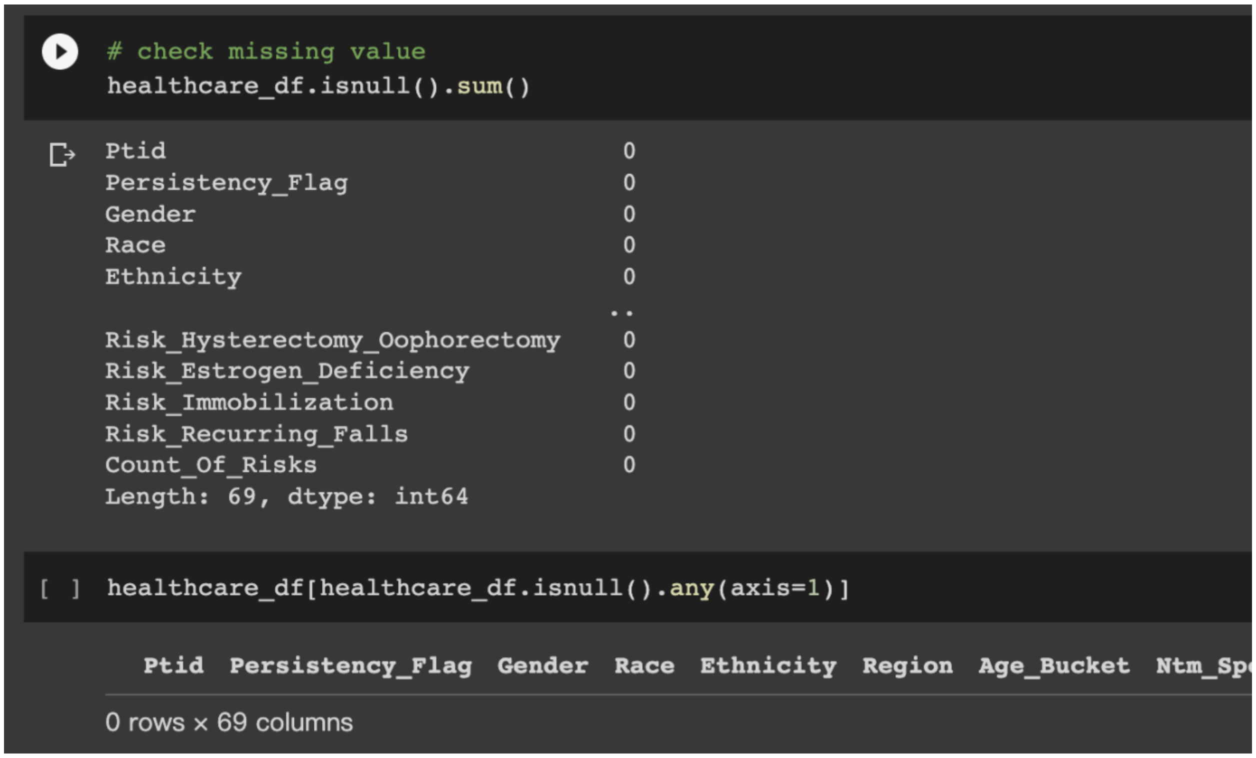
6.2 Check the type of columns



```
# data info & dtype
healthcare_df.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
```

6.3 check the missing value.



```
# check missing value
healthcare_df.isnull().sum()
```

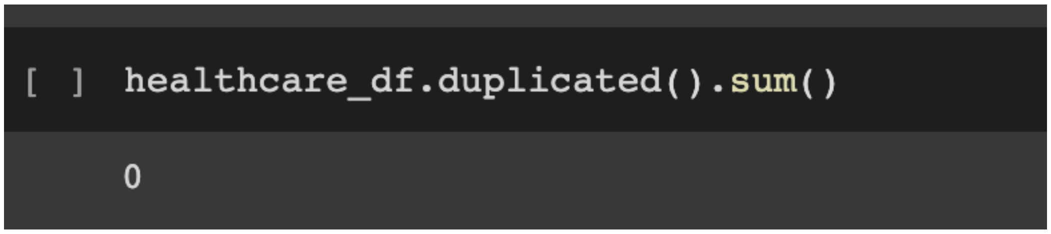
```
Ptid                                0
Persistency_Flag                    0
Gender                              0
Race                                0
Ethnicity                           0
                                   ..
Risk_Hysterectomy_Oophorectomy      0
Risk_Estrogen_Deficiency      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

```
[ ] healthcare_df[healthcare_df.isnull().any(axis=1)]
```

Ptid	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Spe

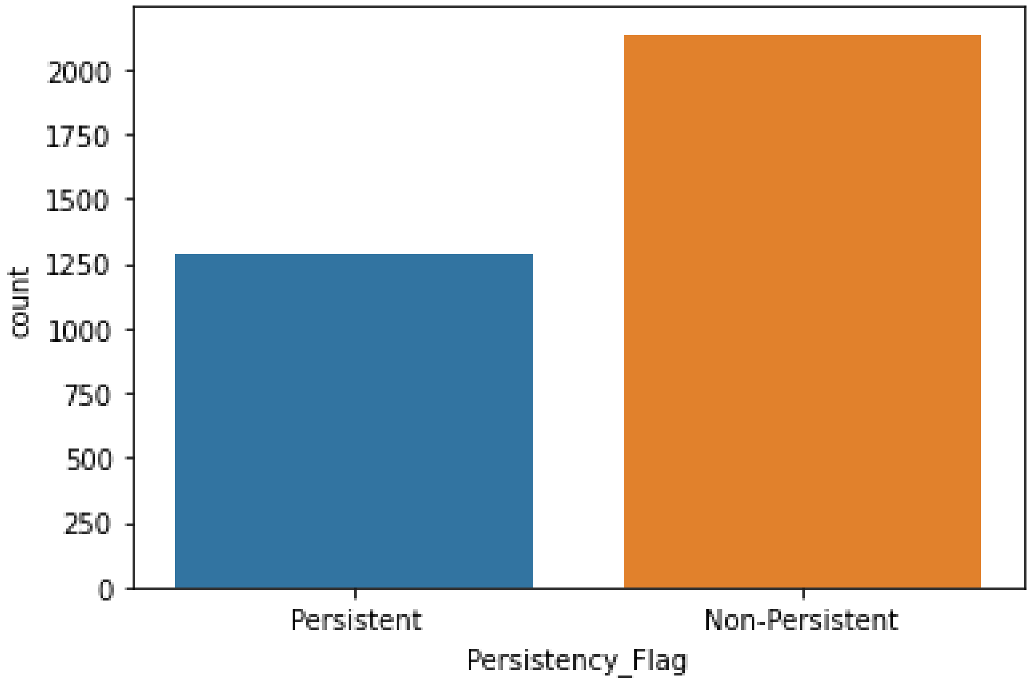
0 rows × 69 columns

6.4 check the duplicate values.

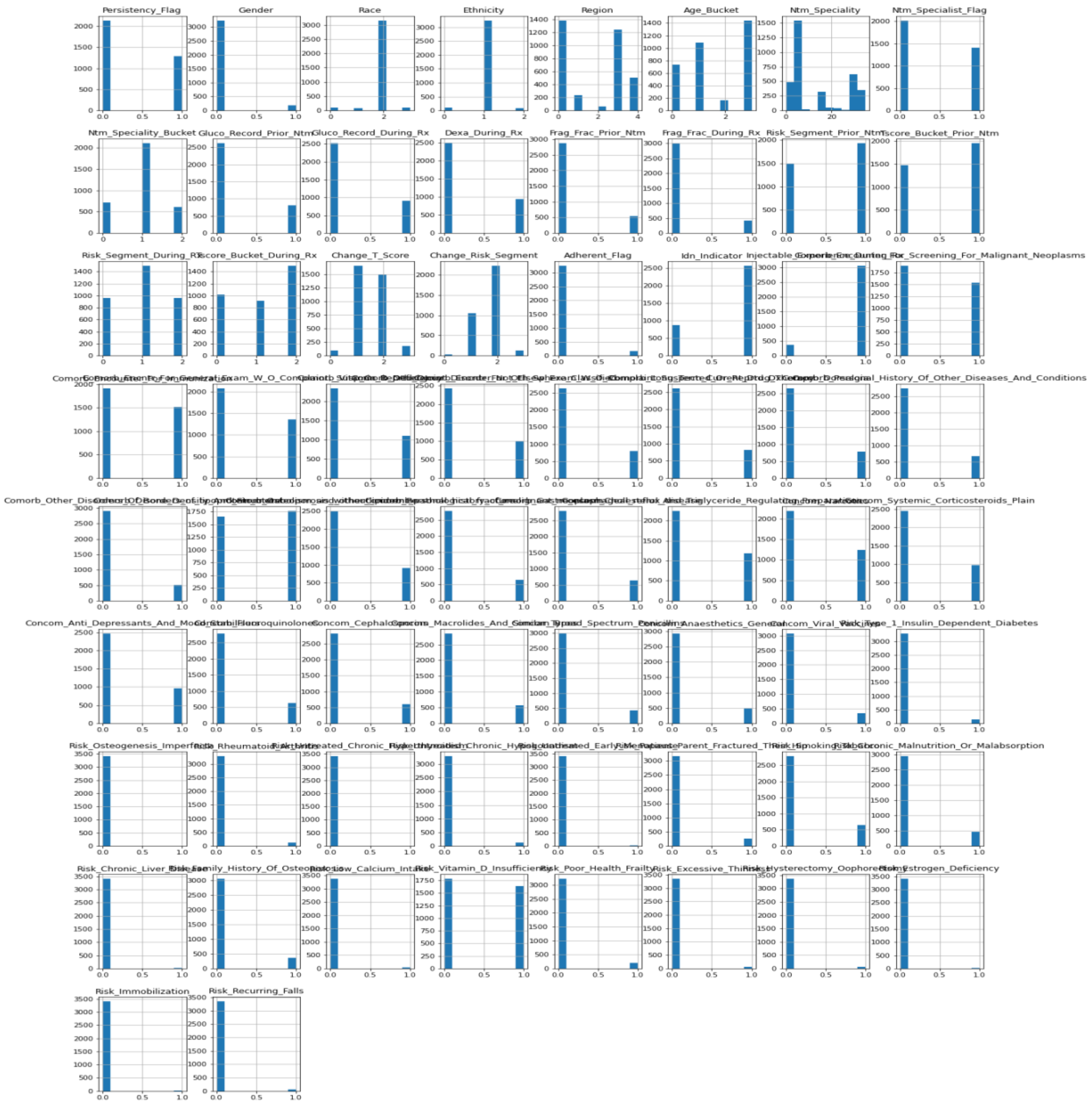


```
[ ] healthcare_df.duplicated().sum()

    0
```

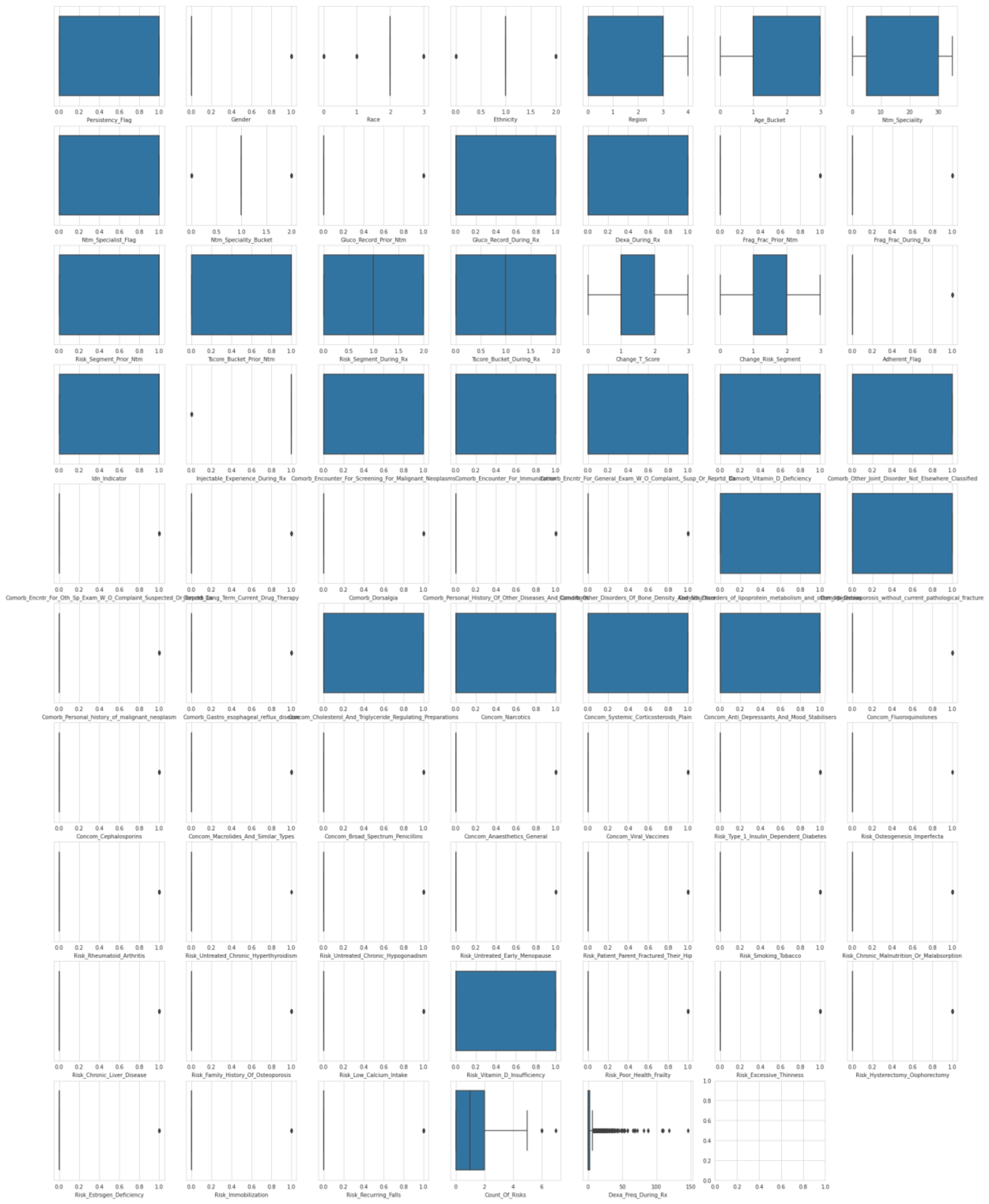
6.5 check whether the dataset is balanced or not.



6.6 Check the distribution of each feature.



6.7 check the outliers in each feature.

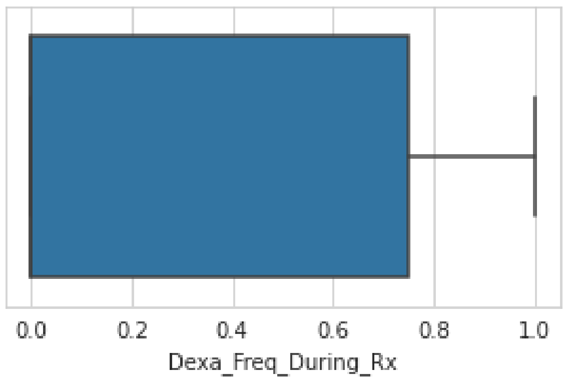


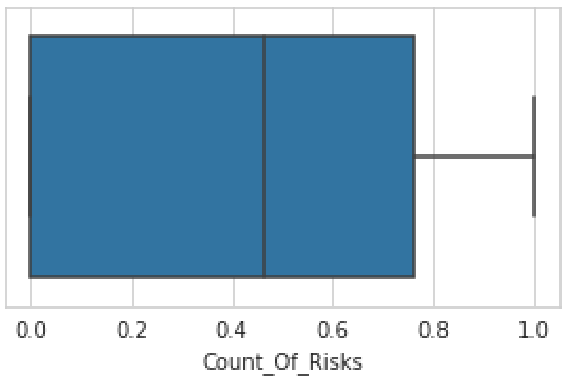
6.8 Conclusion:

1. There are not missing value and duplicate value.
2. 67 features are categorical values. I used label encoder covert them to numerical values.
3. The dataset is imbalanced. I used SMOTE technology to deal with this problem.
4. There are outliers in feature ‘count_of_risks’ and ‘Dexa_freq_during_Rx’. I used Quantile Transformer transforms the features to follow a uniform or a normal distribution.

7. Handle outliers

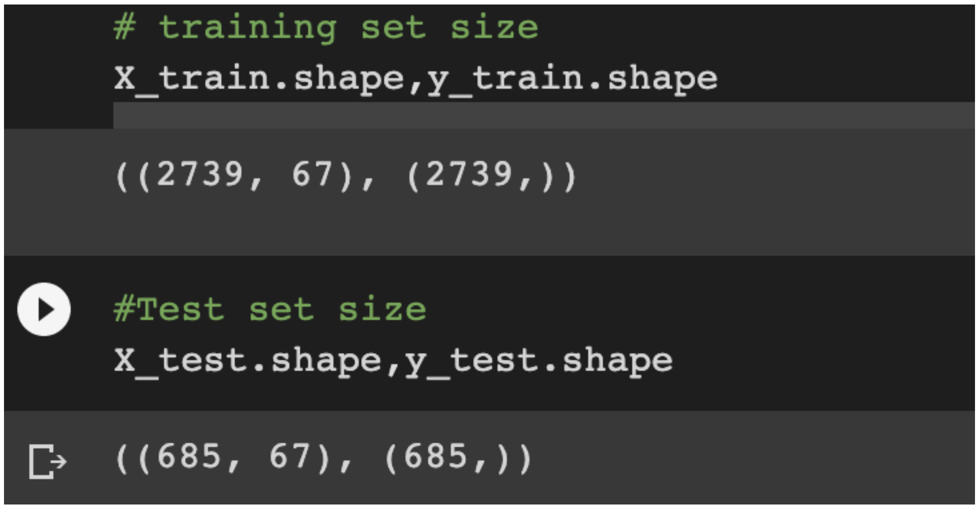
Quantile Transformer was used to handle outliers in two features. Quantile Transformer transforms the features to follow a uniform or a normal distribution. Therefore, for a given feature, this transformation tends to spread out the most frequent values. It also reduces the impact of (marginal) outliers. Following picture is the result after using Quantile Transformer.





8. Split the Data Frame

I randomly choose 80% of records as the training set and the remainder as the test set.



```
# training set size
X_train.shape,y_train.shape

((2739, 67), (2739,))

#Test set size
X_test.shape,y_test.shape

((685, 67), (685,))
```

9. Model Development

9.1 Decision Tree

I built the Decision Tree model. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Then, to prevent the overfitting problem, I used a 10-fold cross-validation method. According to the result of Grid Search, I employ the max_depth as 7 and min_samples_leaf as 20 to build decision tree model.

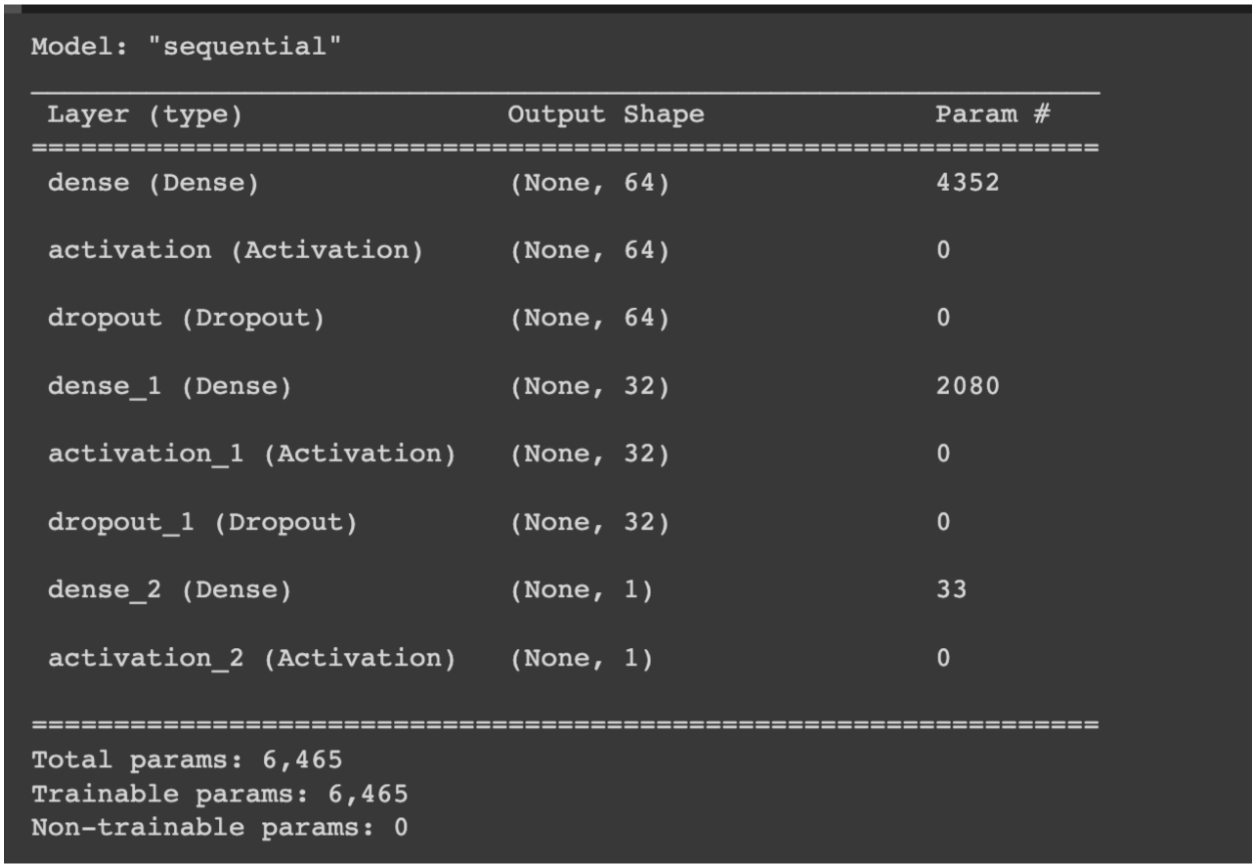
9.2 LightGBM

I built a LightGBM model. LightGBM is a gradient boosting framework that uses tree-based learning algorithms. Then, to prevent the overfitting problem, I used a 10-fold cross-validation method. According to the result of Grid Search, I employ the n_estimators as 40 and num_leaves as 31 to build LightGBM model.

9.3 DNN

I built a Deep Neural Network (DNN) model. Figure depicts the parameters of my 2 layers Deep Neural Network model. The first layer in our Neural Network consists of 64 neurons. The input shape of the first layer is 67 which is the value of the X_train.shape[1]. The second layer is the hidden layer which consists of 32 neurons. The relu activation function is used for these two layers. For preventing the overfitting problem, dropout, one of the regularization techniques, was used. It is a technique where randomly selected neurons are ignored during training. They are “dropped out” randomly. This means that their contribution to the activation of downstream neurons is temporarily removed. In our model, two dropout layers are deployed behind the first layer and the second layer. The drop rate is set as 0.5 at two layers. Finally, the Sigmoid activation function is used for the output layer.

For configuring the learning process, I use the compile method. I set the optimizer, loss function, and metrics. First, the model uses the ADAM optimizer. Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions. It is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Second, binary Cross-entropy loss was used as the loss function in the neural network model. Finally, In order to assess the performance of the model, I considered two metrics: loss and accuracy. Once the learning process is configured, I trained the model and set the hyperparameters which are the number of epochs as 100 for the training process.

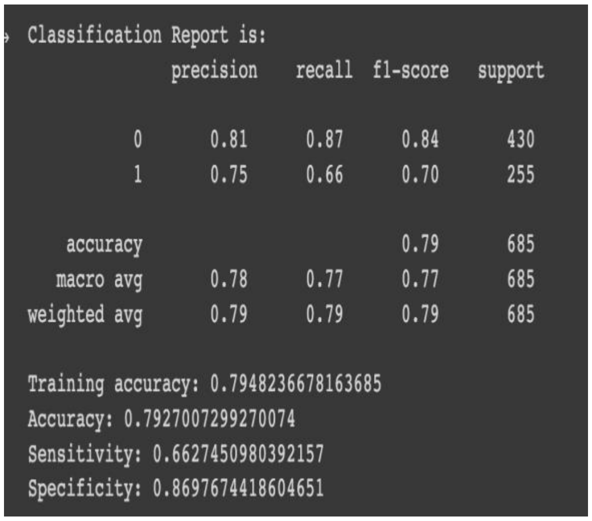


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                4352

 activation (Activation)     (None, 64)                0

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 32)                2080

 activation_1 (Activation)   (None, 32)                0

 dropout_1 (Dropout)         (None, 32)                0

 dense_2 (Dense)             (None, 1)                 33

 activation_2 (Activation)   (None, 1)                 0

=================================================================
Total params: 6,465
Trainable params: 6,465
Non-trainable params: 0
```

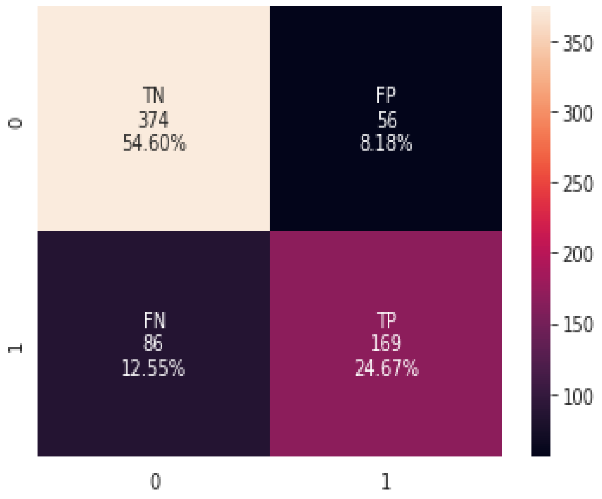
10. Evaluation

The definition of each measure:
Precision: Precision is the ability of a classifier not to label an instance positive that is actually negative. It is defined as the ratio of true positives to the sum of a true positive and false positive for each class.
Precision = TP/(TP + FP)
Recall: Recall is the ability of a classifier to find all positive instances. Each class is defined as the ratio of true positives to the sum of true positives and false negatives. The recall is the fraction of positives that were correctly identified.
Recall = TP/(TP+FN)
F1 score: The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.
F1 Score = 2(Recall * Precision) / (Recall + Precision)*

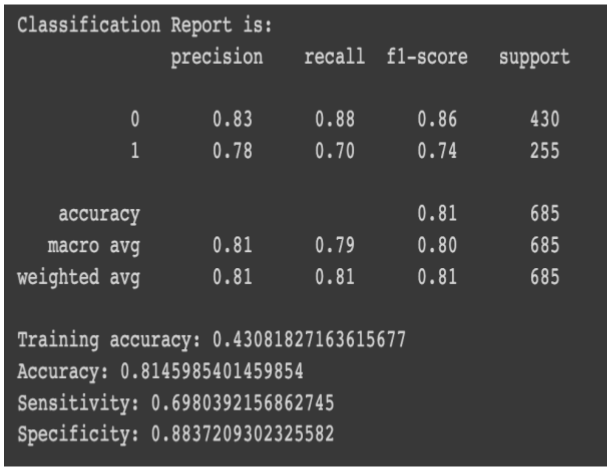
10.1 Evaluation of Decision Tree



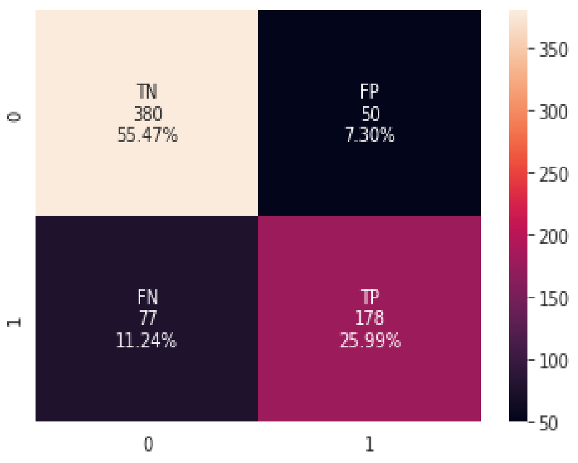
```
Classification Report is:
              precision    recall  f1-score   support

           0       0.81      0.87      0.84       430
           1       0.75      0.66      0.70       255

    accuracy                           0.79       685
   macro avg       0.78      0.77      0.77       685
weighted avg       0.79      0.79      0.79       685

Training accuracy: 0.7948236678163685
Accuracy: 0.7927007299270074
Sensitivity: 0.6627450980392157
Specificity: 0.8697674418604651
```



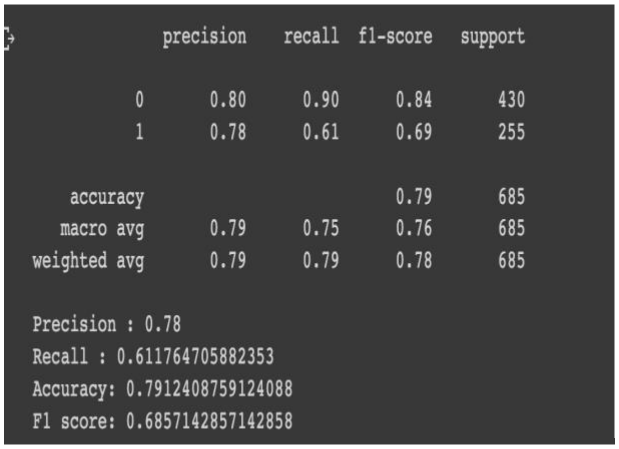
10.2 Evaluation of Lightgbm



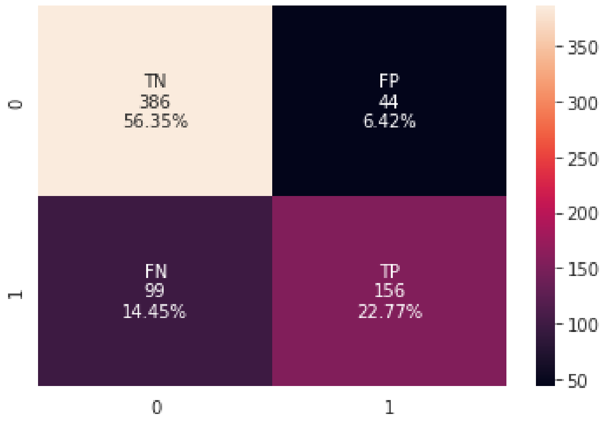
```
Classification Report is:
              precision    recall  f1-score   support

           0       0.83      0.88      0.86       430
           1       0.78      0.70      0.74       255

    accuracy                           0.81       685
   macro avg       0.81      0.79      0.80       685
weighted avg       0.81      0.81      0.81       685

Training accuracy: 0.43081827163615677
Accuracy: 0.8145985401459854
Sensitivity: 0.6980392156862745
Specificity: 0.8837209302325582
```



10.3 Evaluation of DNN



```
              precision    recall  f1-score   support

           0       0.80      0.90      0.84       430
           1       0.78      0.61      0.69       255

    accuracy                           0.79       685
   macro avg       0.79      0.75      0.76       685
weighted avg       0.79      0.79      0.78       685

Precision : 0.78
Recall : 0.611764705882353
Accuracy: 0.7912408759124088
F1 score: 0.6857142857142858
```



According to the classification report and confusion matrix, Lightgbm is the champion model. The accuracy of Lightgbm is the highest one with 0.81.

11. Conclusion

In this project, I analysis the persistency of drug. Decision tree, Lightgbm, and DNN models were built to classify the persistency of drug. Based on evalution, Lightgbm is the champion model. The accuracy of Lightgbm is the highest one with 0.81. In the furuter, the Lightgbm will deploy to automatically identify persistency of drug.