Data Science Intern at Data Glacier

Project: Healthcare - Persistency of a drug

Week 9: 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 2, 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_Flag
0	P1	Persistent	Male	Caucasian	Not Hispanic	West	>75	GENERAL PRACTITIONER	Others
1	P2	Non-Persistent	Male	Asian	Not Hispanic	West	55-65	GENERAL PRACTITIONER	Others
2	P3	Non-Persistent	Female	Other/Unknown	Hispanic	Midwest	65-75	GENERAL PRACTITIONER	Others
3	P4	Non-Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	GENERAL PRACTITIONER	Others
4	P5	Non-Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	GENERAL PRACTITIONER	Others

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_Sp

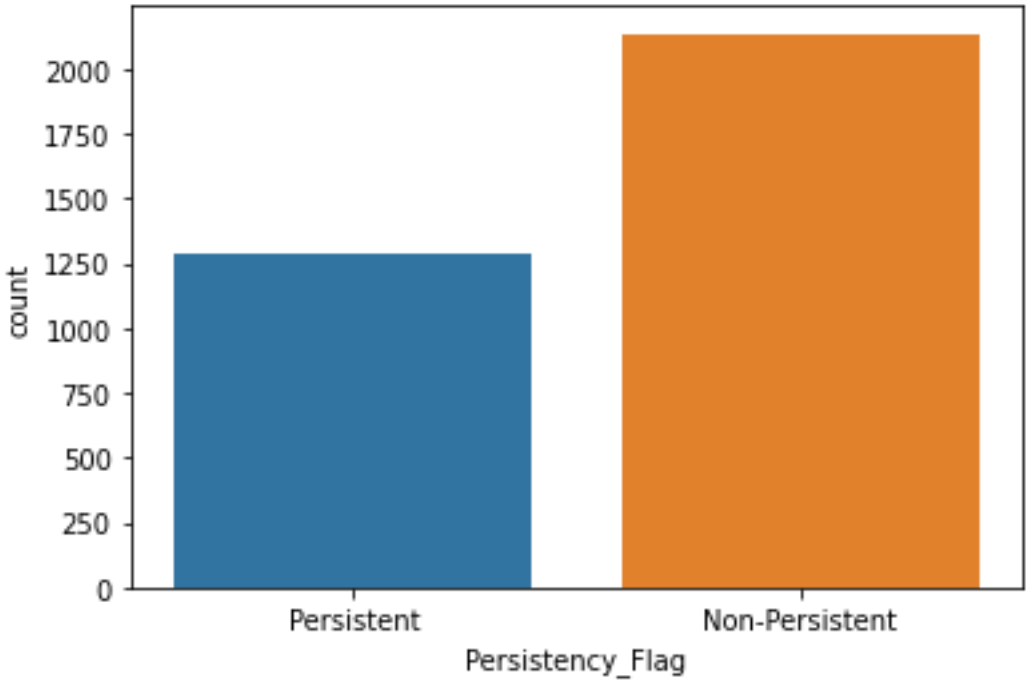
0 rows × 69 columns

6.4 check the duplicate values.

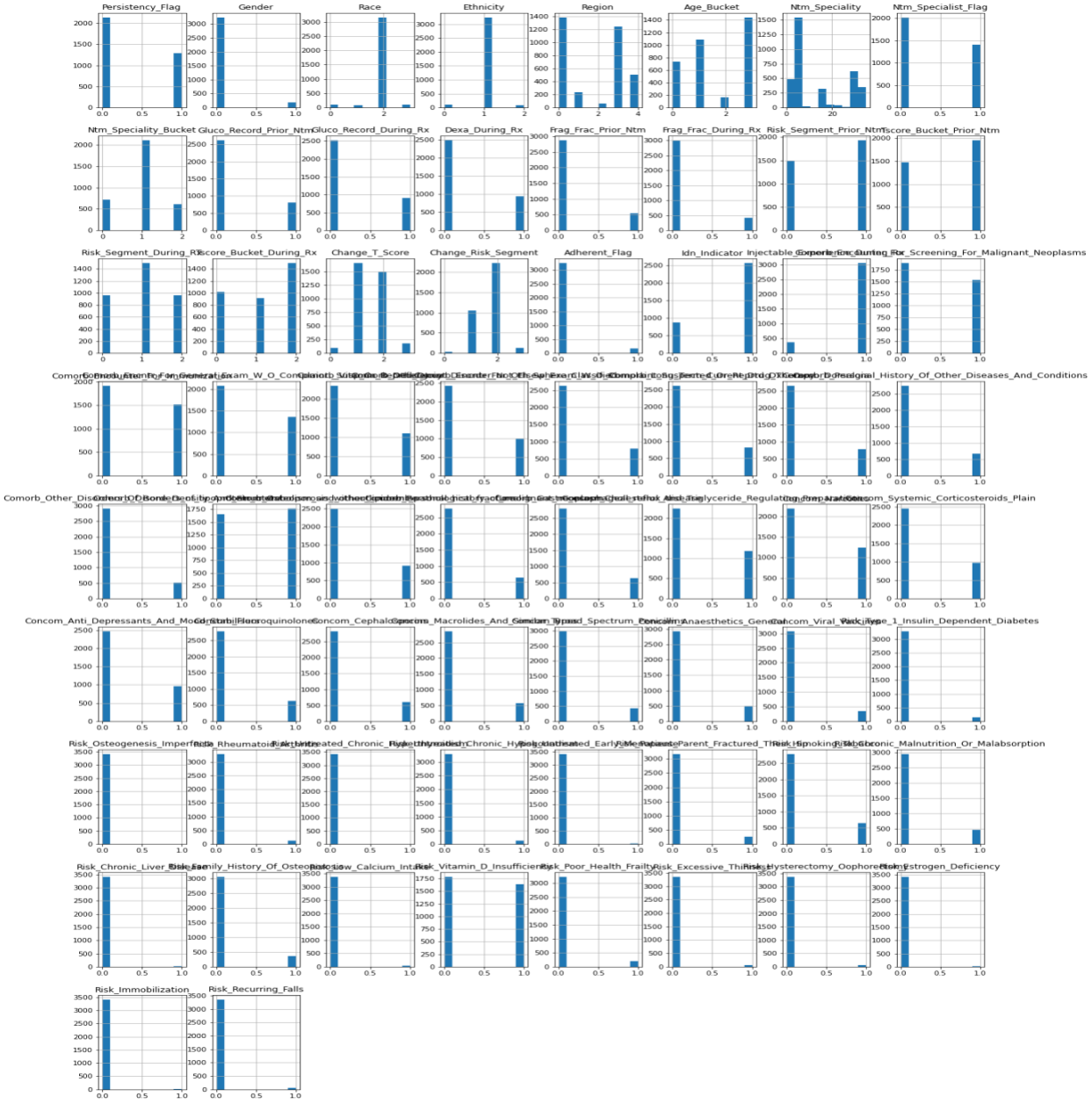
```
[ ]  healthcare_df.duplicated().sum()

     0
```

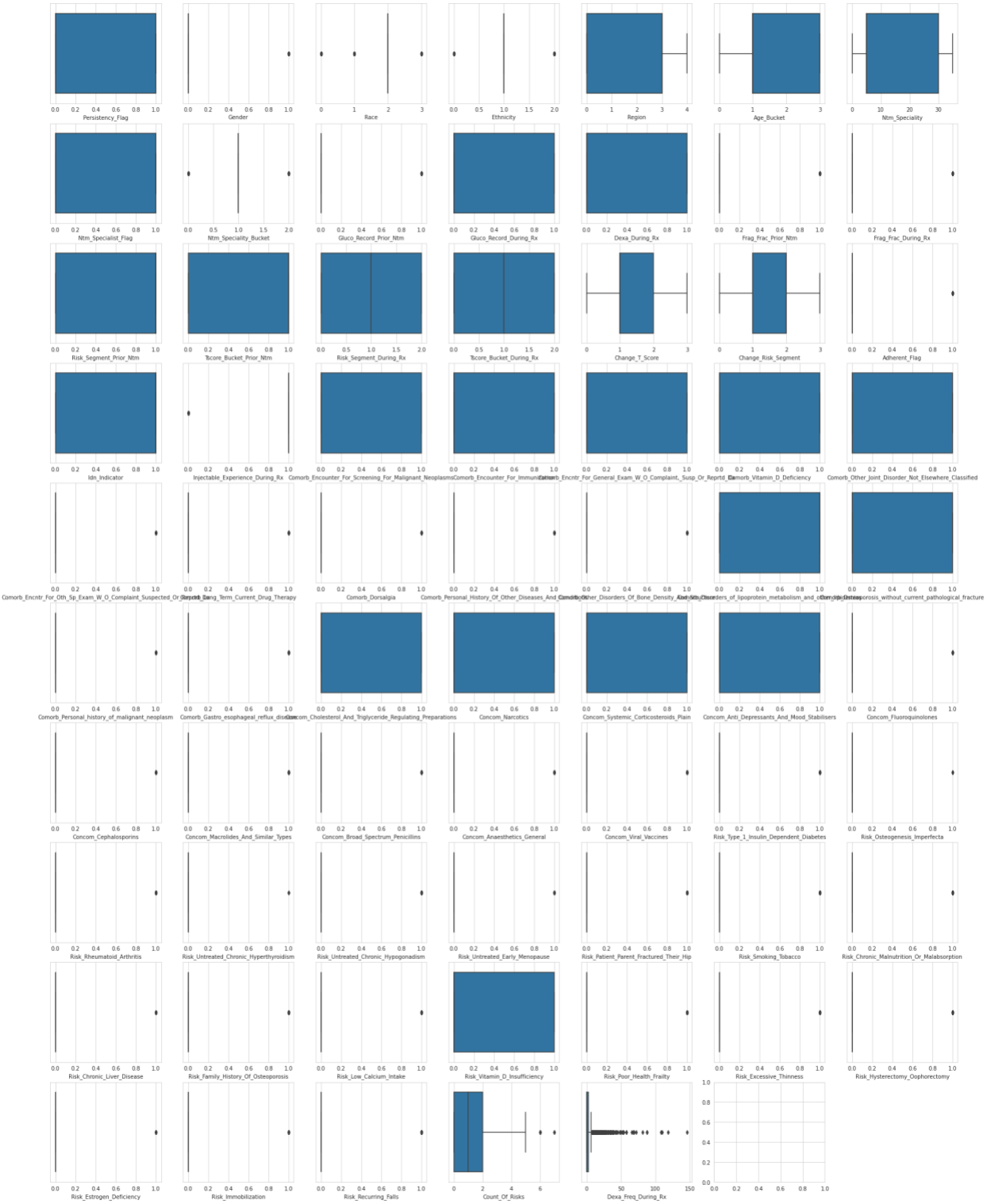
6.5 check whether the dataset is balanced or not.



6.6 Check the distribution of each feature.



6.7 check the outliers in each feature.

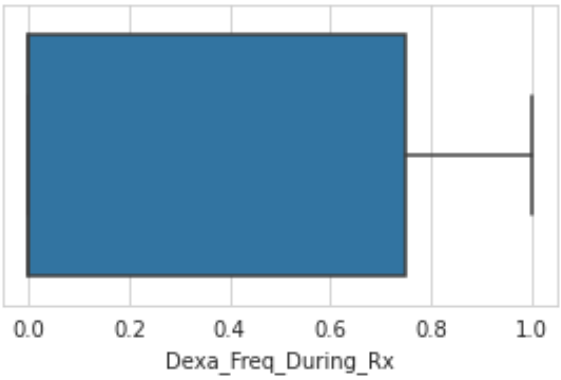


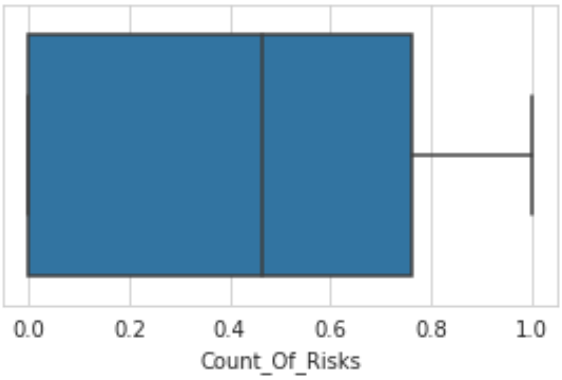
6.8 Conclusion:

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**
   - **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 5-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.