Final Project Report
Data Science-HealthCare: Persistency of a Drug

AMNA NAEEM
Email: amnanaeem675@gmail.com

PROBLEM DESCRIPTION:

- One of the greatest problems faced by Pharmaceutical companies these days is of persistency related to drugs. Persistency is related to the behavior of patient towards the medication advised by doctor. This term reflects the duration of time a drug is taken by a patient from the start of treatment to the end of treatment as maintained by doctor. The persistency is important because it can save not only the health of patient but also the money endured on health care system on illness due to discontinuity of drugs.

- **OBJECTIVE OF THE STUDY:**

The objective is to gather insights on the factors that are impacting the persistency, build a classification for the given dataset

Project Lifecycle:

The project is sub-divided into following categories:

- Loading the necessary libraries.
- Data Understanding
- Data cleansing and Transformation
- Exploratory Data Analysis
- Modeling Development and Evaluation

Data Set:

- The data set is downloaded from the link: https://drive.google.com/file/d/1P_oMc6gOBlhw6dY5PxaqxV2swdHMUooK/view. Data contained information related to demographics, doctor specialty, clinical factors and disease/treatment factors mainly related to non-tuberculous mycobacteria (NTM).

```
data=pd.read_csv("datahealth.csv")
```

```
df=data.copy()
df.head()
```

	Ptid	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Speciality	Ntm_Specialist_Flag	Ntm_Speciality_Bucket	...	Risk_F
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	...	

5 rows × 69 columns

Data Understanding:

- The main features are further sub-categorized hence in total there are 3424 observations related to 69 features including the Patient's id in data.

```
df.shape
```

```
(3424, 69)
```

- However, 67 variables are non-numerical in nature, consist information regarding different categories and only two variables namely; Dexa_Freq_During_Rx, Count_Of_Risks are of numerical nature.

```
dtypes: int64(2), object(67)
memory usage: 949.7+ KB
```

Data Cleansing And Transformation

- **Missing value analysis:**
  - Data set is divided into two subsets on the basis of datatypes and evaluated for missing values. There was no missing value found related to any feature.

```
in_df=df.columns[df.dtypes!='object']
obj_df=df.columns[df.dtypes=='object']
```

```
df[in_df].isnull().sum()
```

```
Dexa_Freq_During_Rx    0
Count_Of_Risks         0
dtype: int64
```

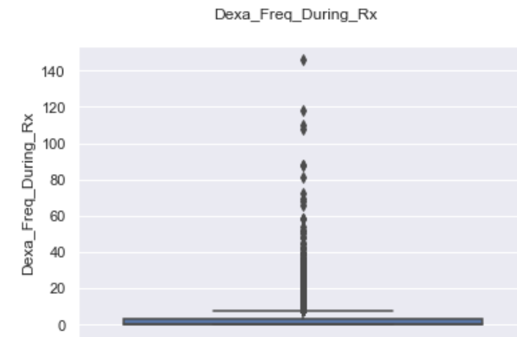
```
df[obj_df].isnull().sum()
```

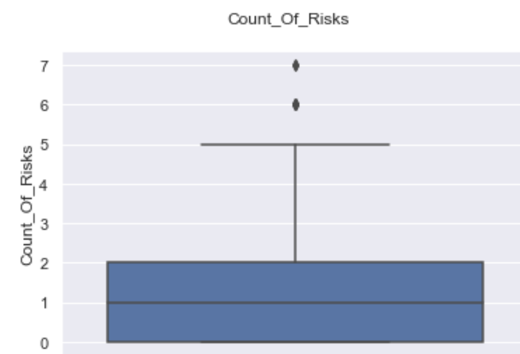
```
Ptid                               0
Persistency_Flag                   0
Gender                             0
Race                               0
Ethnicity                          0
                                  ..
Risk_Excessive_Thinness            0
Risk_Hysterectomy_Oophorectomy     0
Risk_Estrogen_Deficiency           0
Risk_Immobilization                0
Risk_Recurring_Falls               0
Length: 67, dtype: int64
```

Duplicate values and Outliers Detection:

No duplicate values were present in the dataset. However, box-plot is used for outlier detection and are found in both numerical variables. The outliers are removed from both variables using the inter quartile method. The results are plotted using the box plot.

```
for x in df.columns[df.dtypes!=object]:
    fig=plt.figure()
    sns.boxplot(y=df[x],data=df)
    fig.suptitle(x)
```






```
duplicate=df[df.duplicated()]
duplicate
```

Ptid	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Speciality	Ntm_Specialist_Flag	Ntm_Speciality_Bucket	...	Risk_Family_History_Of_

0 rows × 69 columns

Variables after removal of Outliers:

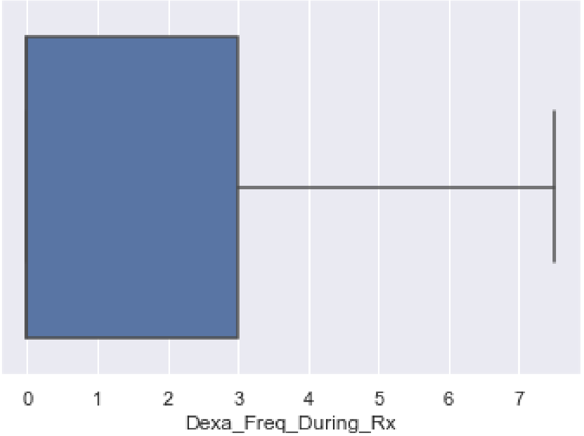
```
percentile25 = df['Dexa_Freq_During_Rx'].quantile(0.25)
percentile75 = df['Dexa_Freq_During_Rx'].quantile(0.75)
upper_limit = percentile75 + 1.5 * (percentile75 -percentile25 )
lower_limit = percentile25 - 1.5 * (percentile75 -percentile25 )
df[df['Dexa_Freq_During_Rx'] > upper_limit]
df[df['Dexa_Freq_During_Rx'] < lower_limit]
df3 = df[df['Dexa_Freq_During_Rx'] < upper_limit]
df3.shape
```

```
(2964, 69)
```

```
df3_cap = df.copy()
df3_cap['Dexa_Freq_During_Rx'] = np.where(
    df3_cap['Dexa_Freq_During_Rx'] > upper_limit,
    upper_limit,
    np.where(
        df3_cap['Dexa_Freq_During_Rx'] < lower_limit,
        lower_limit,
        df3_cap['Dexa_Freq_During_Rx']
    )
)
```

```
sns.boxplot(new_df_cap['Dexa_Freq_During_Rx'])
```

```
<AxesSubplot:xlabel='Dexa_Freq_During_Rx'>
```



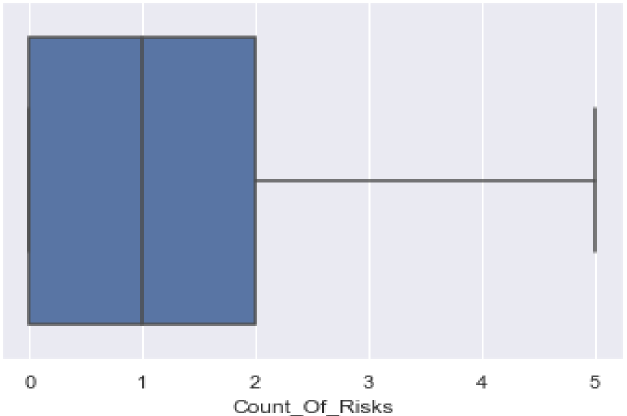

```
percentile25 = df3_cap['Count_Of_Risks'].quantile(0.25)
percentile75 = df3_cap['Count_Of_Risks'].quantile(0.75)
upper_limit = percentile75 + 1.5 * (percentile75 -percentile25 )
lower_limit = percentile25 - 1.5 * (percentile75 -percentile25 )
df3_cap[df3_cap['Count_Of_Risks'] > upper_limit]
df3_cap[df3_cap['Count_Of_Risks'] < lower_limit]
new_df = df3_cap[df3_cap['Count_Of_Risks'] < upper_limit]
new_df.shape
```

```
(3401, 69)
```

```
new_df_cap = df3_cap.copy()
new_df_cap['Count_Of_Risks'] = np.where(
    new_df_cap['Count_Of_Risks'] > upper_limit,
    upper_limit,
    np.where(
        new_df_cap['Count_Of_Risks'] < lower_limit,
        lower_limit,
        new_df_cap['Count_Of_Risks']
    )
)
```

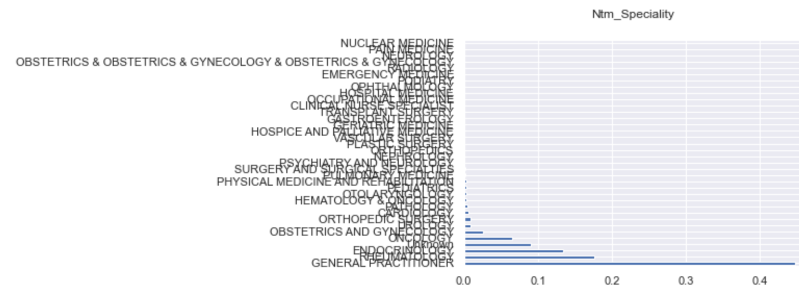
```
sns.boxplot(new_df_cap['Count_Of_Risks'])
```

```
<AxesSubplot:xlabel='Count_Of_Risks'>
```



Rare Category Problem in Categorical Variables:

The categorical variables are evaluated for rare category problem that results when there exists a large difference between frequency of categories. This is found in one variable name NTM_Speciality. This problem is solved by merging the categories with frequency less than 100 into one category and named as others.




```
conditions=[
    (df['Ntm_Speciality'] == 'GENERAL PRACTITIONER'),
(df['Ntm_Speciality'] == 'RHEUMATOLOGY'),
(df['Ntm_Speciality'] == 'ENDOCRINOLOGY'),
(df['Ntm_Speciality'] == 'ONCOLOGY')
]
```

```
choices=['GENERAL PRACTITIONER','RHEUMATOLOGY','ENDOCRINOLOGY','ONCOLOGY']
```

```
df['Ntm_Speciality_Cat'] = np.select(conditions, choices, default='other')
```

```
df['Ntm_Speciality_Cat'].value_counts()
```

```
GENERAL PRACTITIONER    1535
RHEUMATOLOGY             604
other                    602
ENDOCRINOLOGY            458
ONCOLOGY                 225
Name: Ntm_Speciality_Cat, dtype: int64
```

Transformation and Correlation:

```
def number_encode_features(df):
    result = df.copy()
    encoders = {}
    for column in result.columns:
        if result.dtypes[column] == np.object:
            encoders[column] = preprocessing.LabelEncoder()
            result[column] = encoders[column].fit_transform(result[column]
)
    return result, encoders
# Calculate the correlation and plot it
encoded_data, _ = number_encode_features(df)
encoded_data.drop(['Ptid'],axis=1).corr()
```

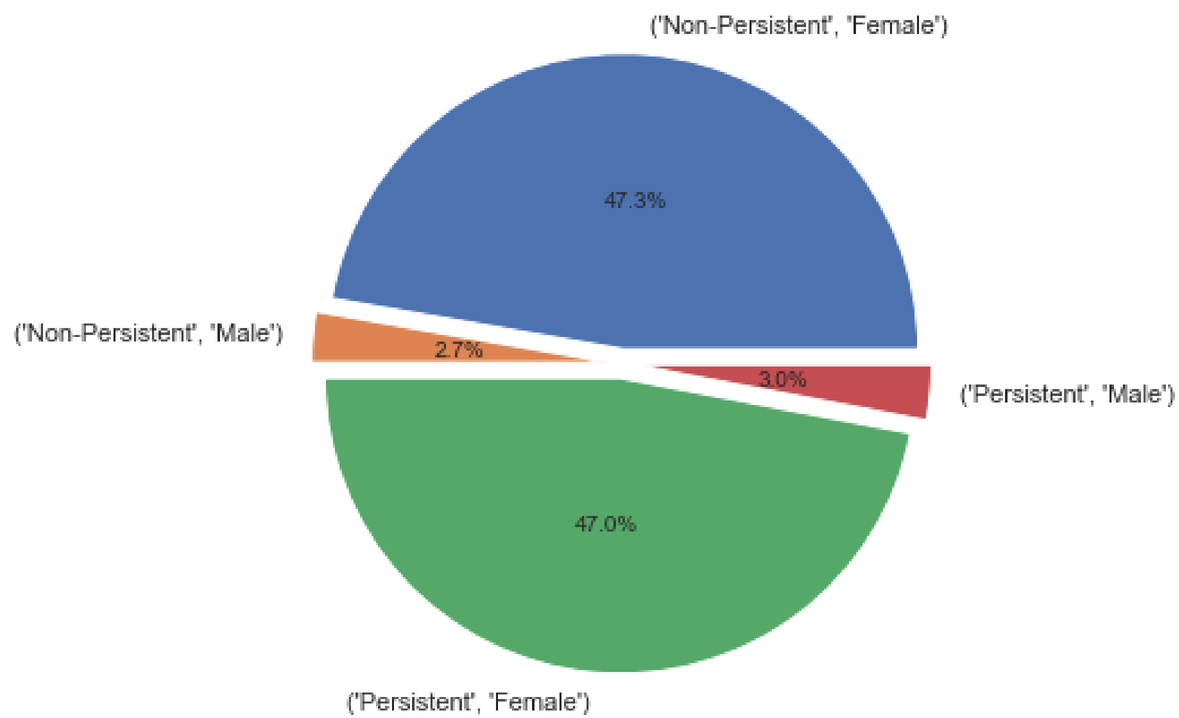
```
def get_redundant_pairs(encoded_data):
    '''Get diagonal and lower triangular pairs of correlation matrix'''
    pairs_to_drop = set()
    cols = encoded_data.columns
    for i in range(0, encoded_data.shape[1]):
        for j in range(0, i+1):
            pairs_to_drop.add((cols[i], cols[j]))
    return pairs_to_drop

def get_top_abs_correlations(encoded_data, n=5):
    au_corr = encoded_data.corr().abs().unstack()
    labels_to_drop = get_redundant_pairs(encoded_data)
    au_corr = au_corr.drop(labels=labels_to_drop).sort_values(ascending=False)
    return au_corr[0:n]

print("Top Absolute Correlations")
print(get_top_abs_correlations(encoded_data, 3))
```

```
Top Absolute Correlations
Dexa_Freq_During_Rx      Dexa_During_Rx            0.948994
Ntm_Speciality           Ntm_Speciality_Cat        0.868479
Risk_Segment_Prior_Ntm   Tscore_Bucket_Prior_Ntm   0.866841
dtype: float64
```

The correlation for all variables are computed after converting them into codes. For this the label encoder method from sklearing package has been used. The highest correlation is found to exist between six variables. The three of them namely Dexa_Freq_During_Rx, Ntim_Speciality and Risk_Segment_Prior_Ntm are excluded from the dataset.

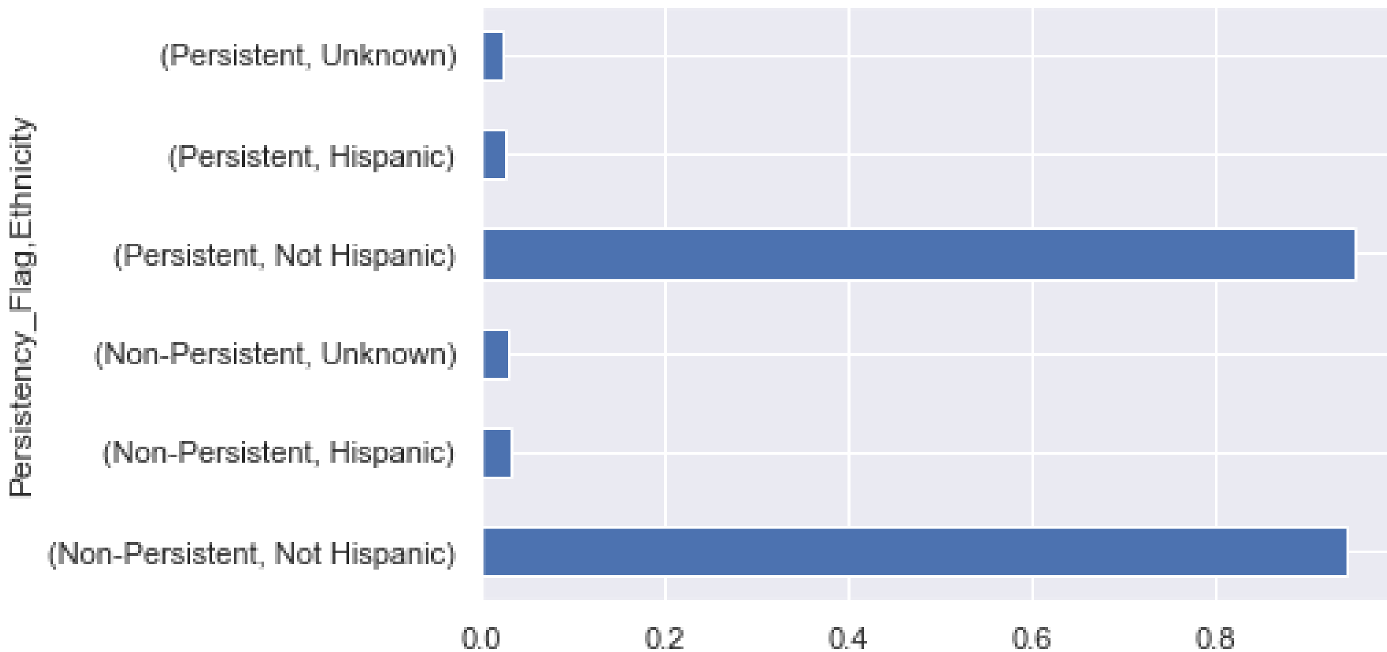
EDA: Persistency and Gender

The persistency in genders are found to be same as **0.3%** difference exist between persistency and non-persistent male and female.



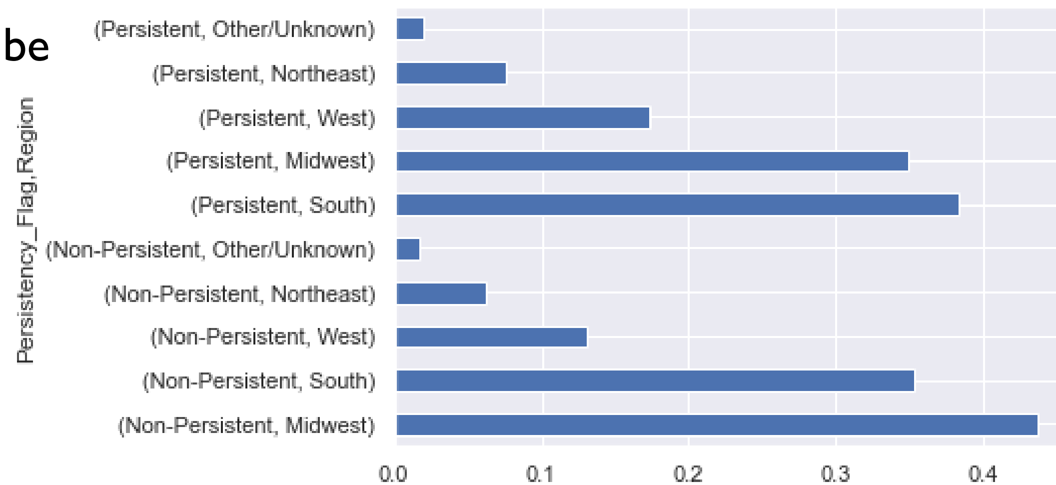
Persistency and Ethnicity:

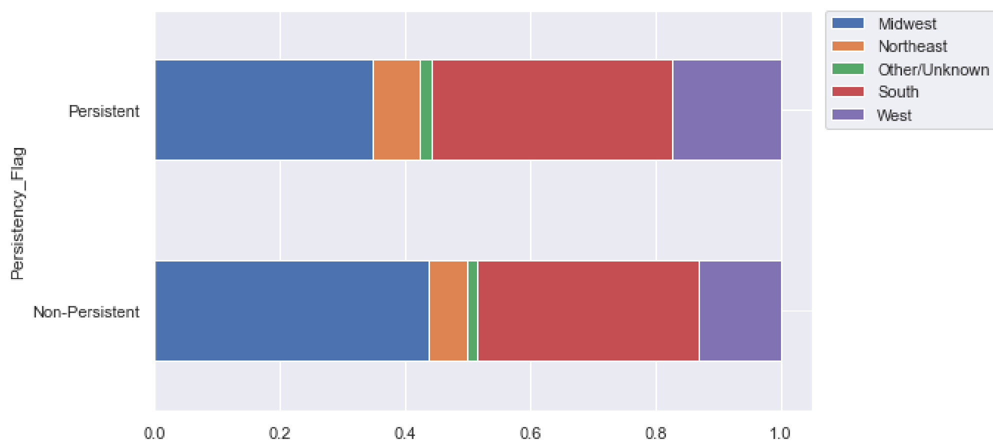
The people belongs to Not Hispanic group are found to be more persistent as compare to Non-Hispanic.



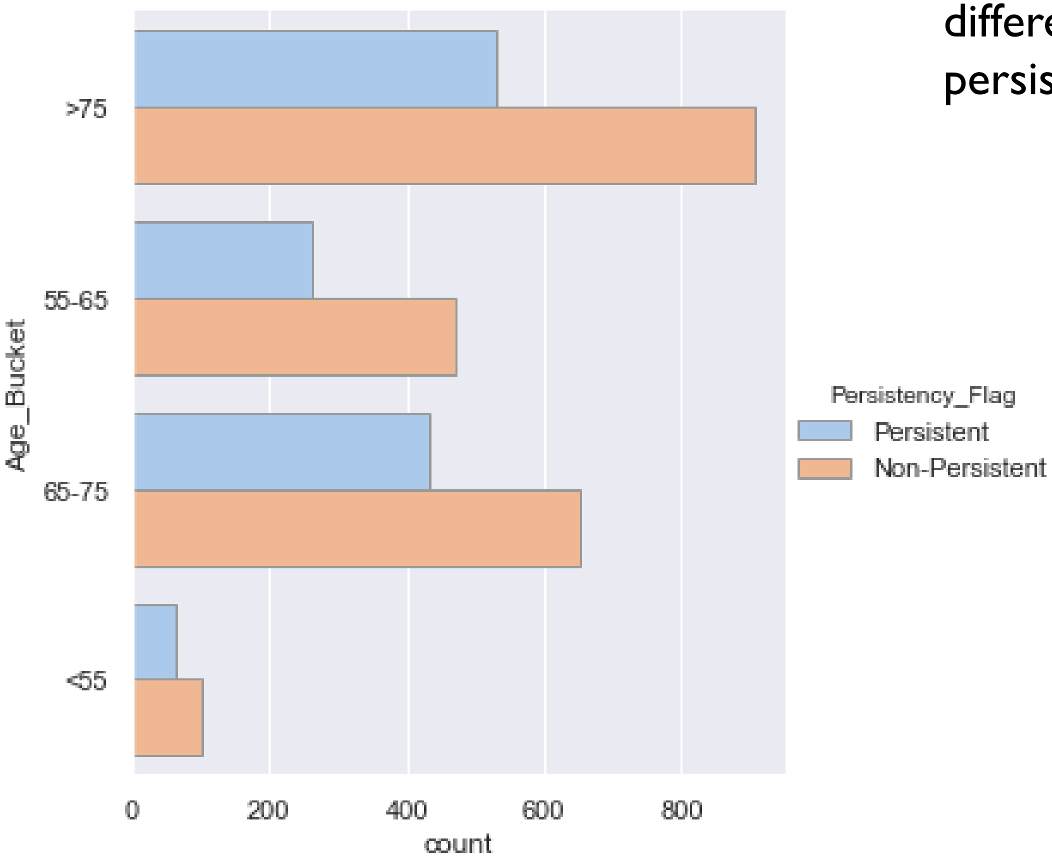
Persistency and Region

People from all regions are found to be mostly non-persistent however the people from **west regions are appeared to be more persistent** than non persistent people.



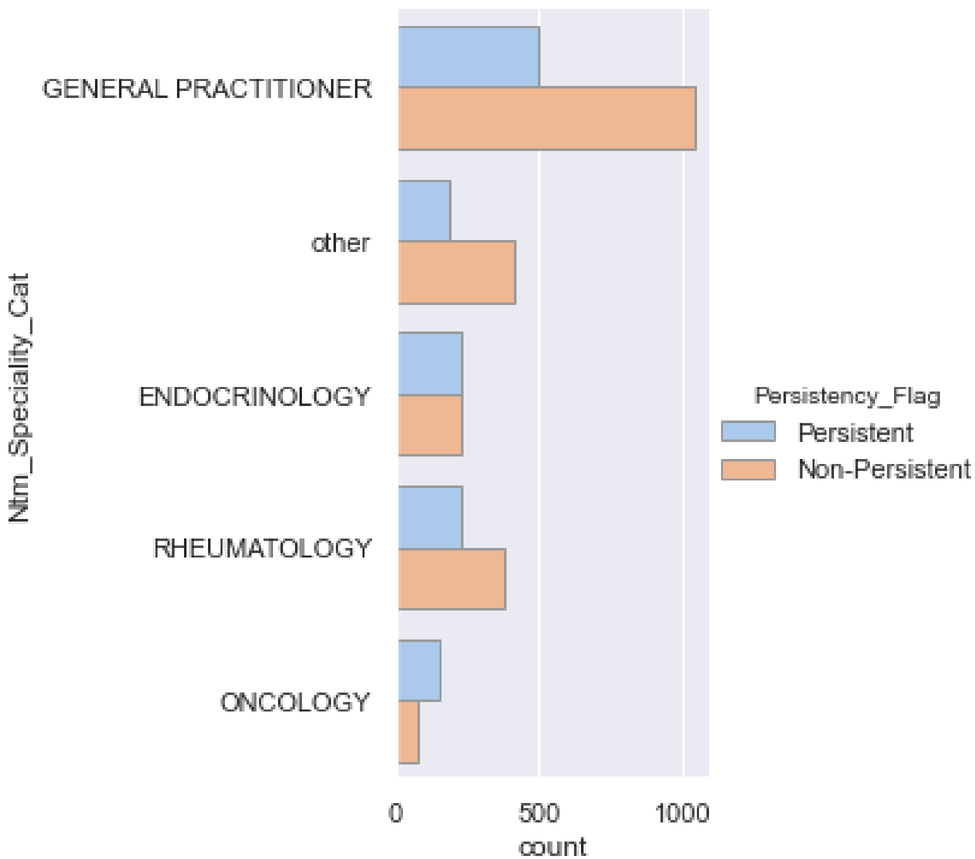


Persistency and Age



People from all age groups are found to be more non-persistent however the difference between persistent and non-persistent for age group<55 is small.

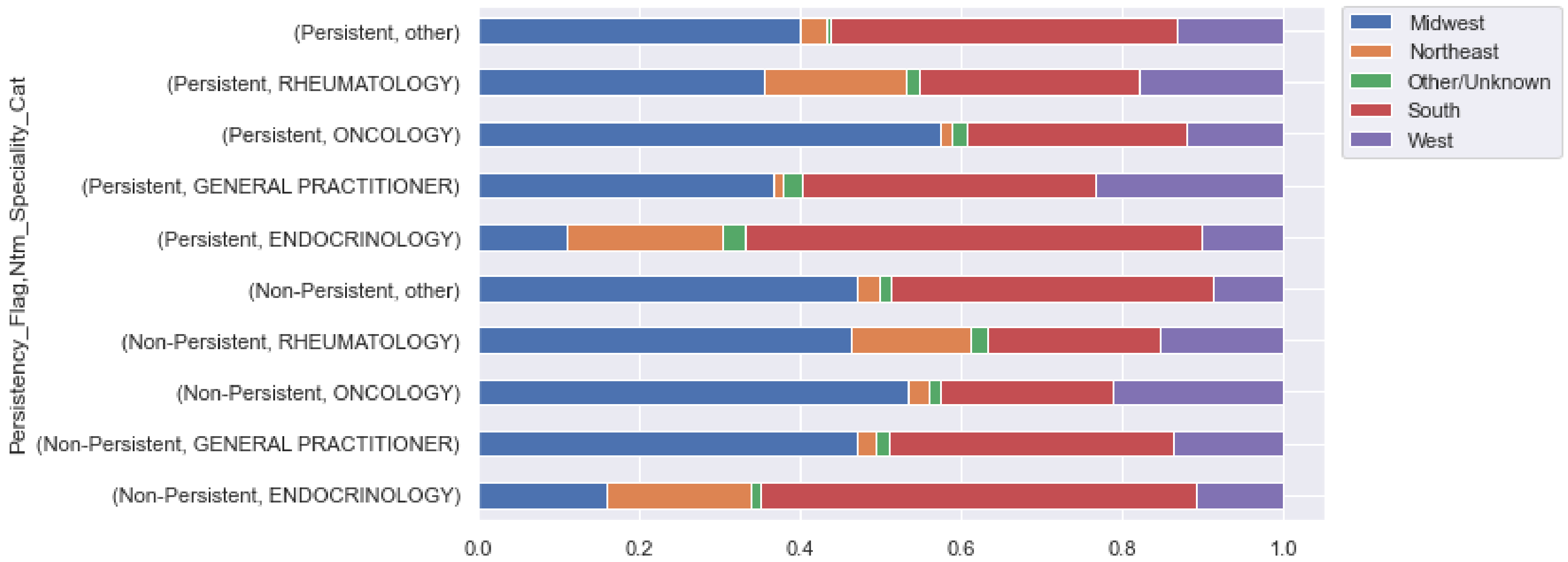
Persistency and Ntm-Speciality



The medicine prescribed by oncology specialist seems to have more persistency and endocrinology specialists are found to have same number for both categories. Whereas large number of non persistent in have been found patients in case of general practitioner.

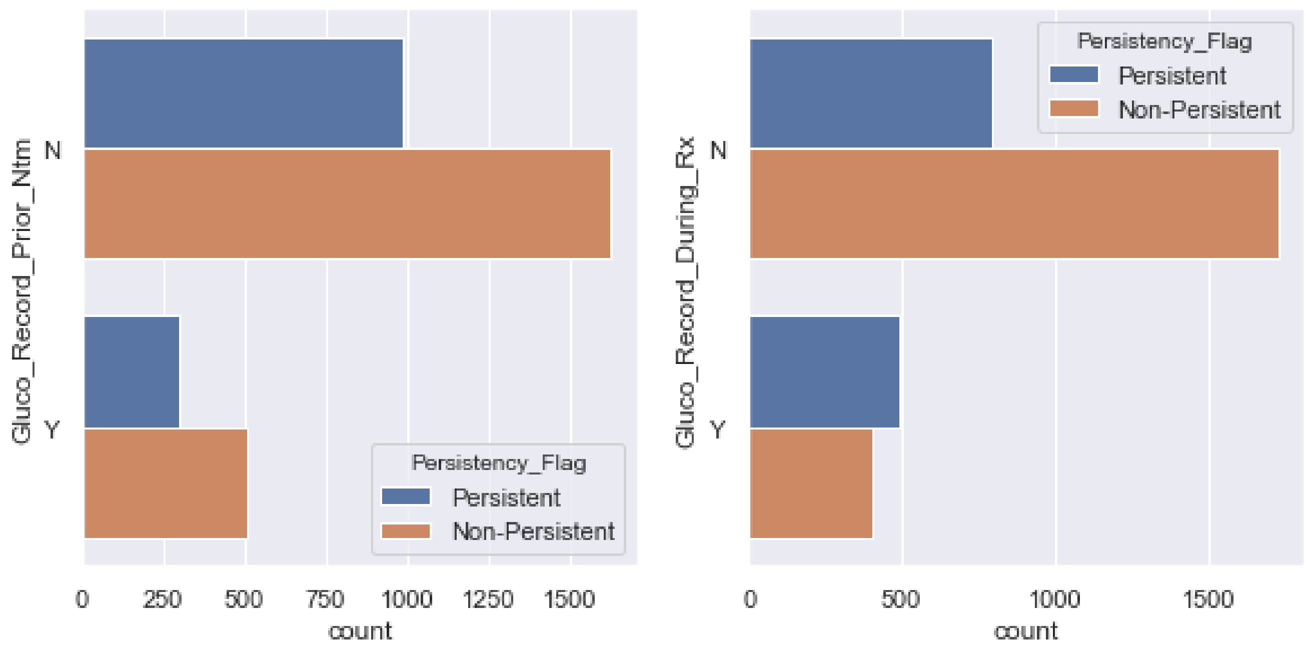
Persistency, Ntm-Speciality and Region

The west people are found to be more persistent in case of general practitioner where people from Midwest region are found to be more persistent when it comes to Oncology. The people from south are more persistent when it comes to endocrinology.

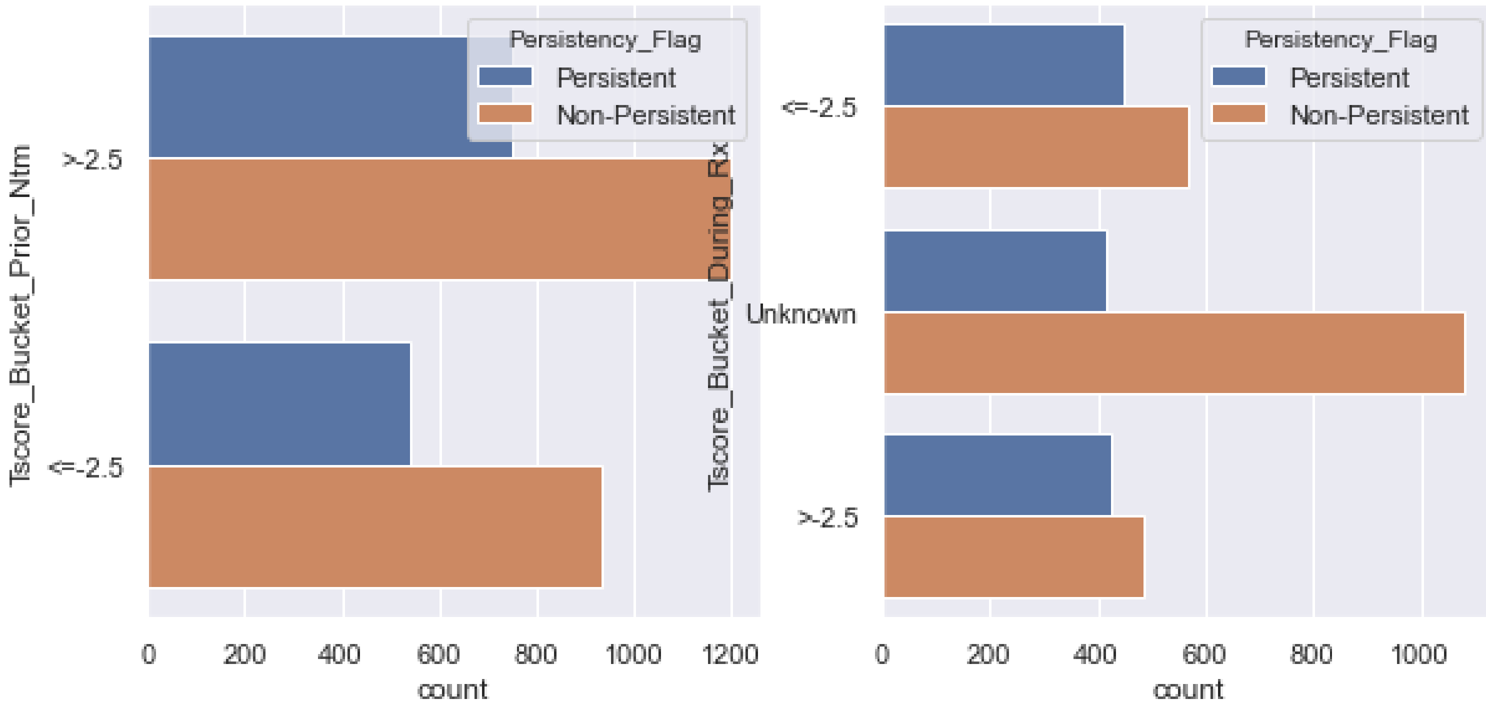


Persistency and Glucocorticoid

The Glucocorticoid record during Rx showed persistency.

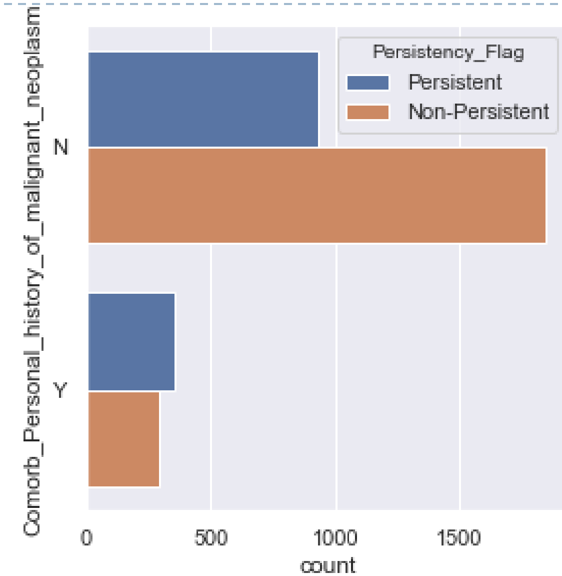


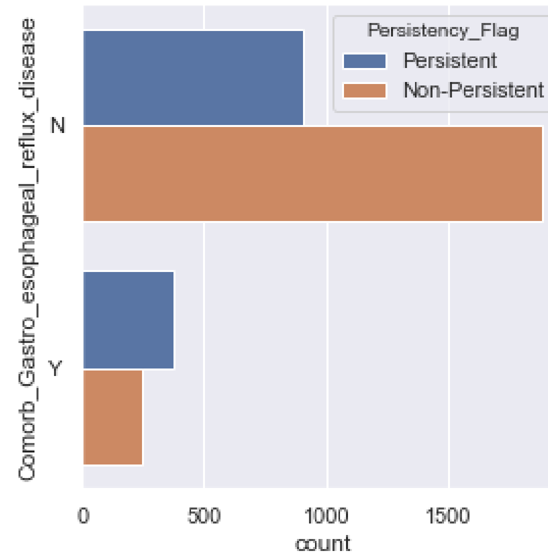
T-score and Persistency

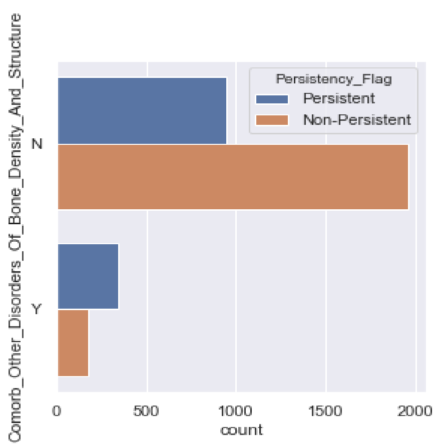


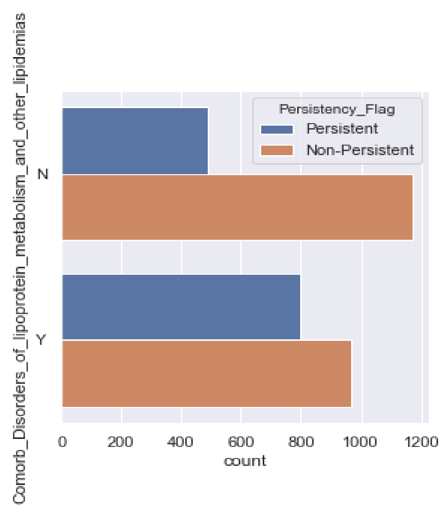
Persistency and NTM-Comorbidity

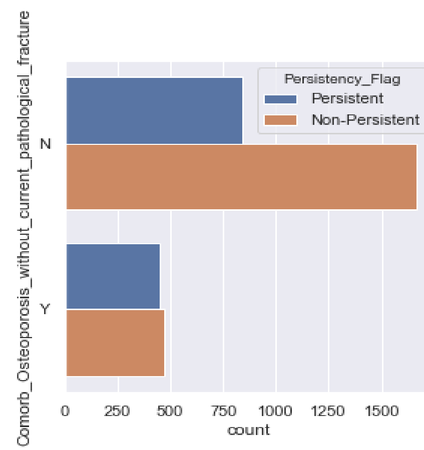
Persistency is examined for different comorbidities and persistency has been found in most people suffering from different diseases.

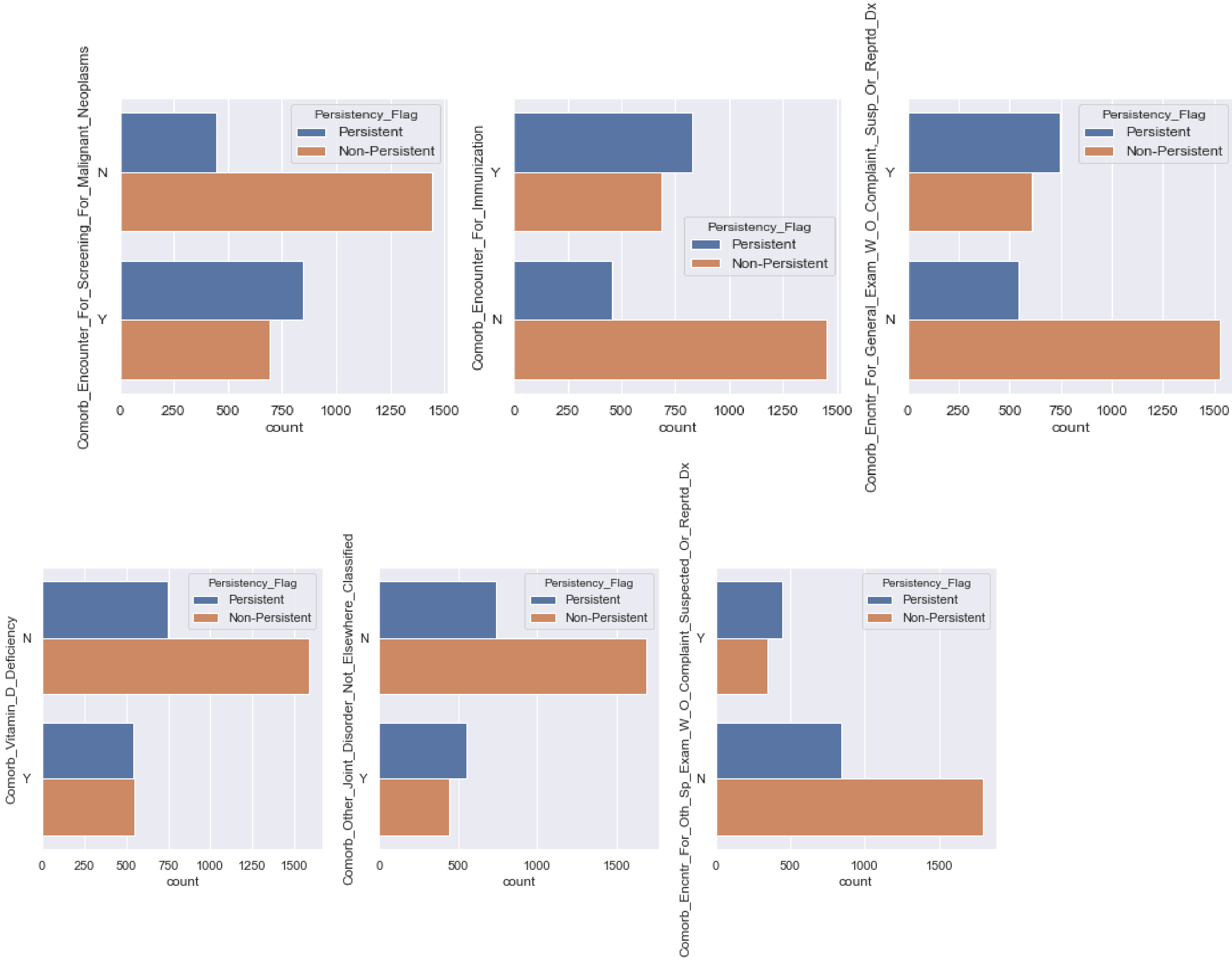








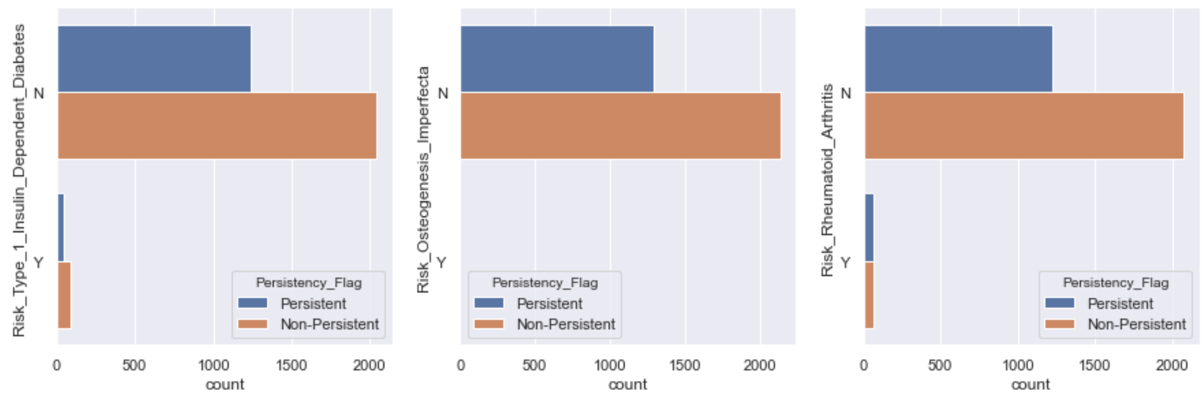



Comorb_Encounter_For_Screening_For_Malignant_Neoplasms
Comorb_Encounter_For_Immunization
Comorb_Encntr_For_General_Exam_W_O_Complaint,_Susp_Or_Reprtd_Dx
Comorb_Vitamin_D_Deficiency
Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified
Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx
Persistency_Flag
Persistent
Non-Persistent
N
Y
count
0
250
500
750
1000
1250
1500

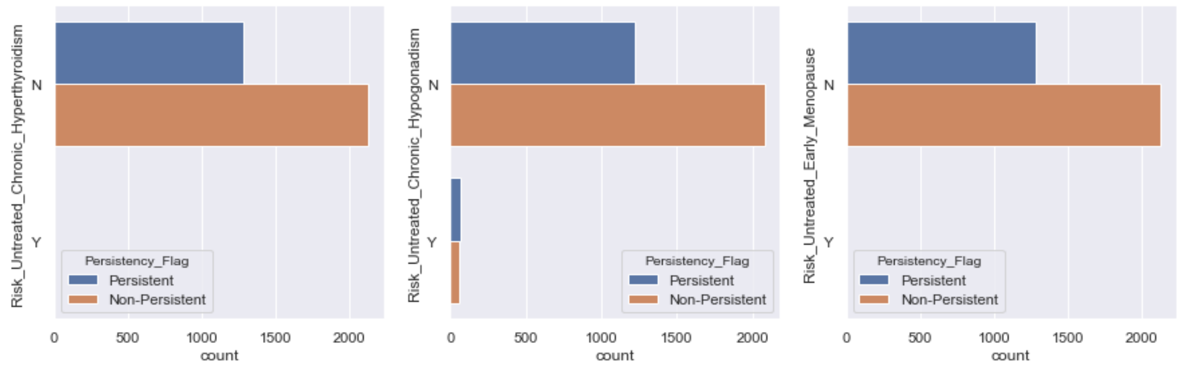
Persistency and NTM-Risk

Risk factor has showed less influence on the persistency of people .

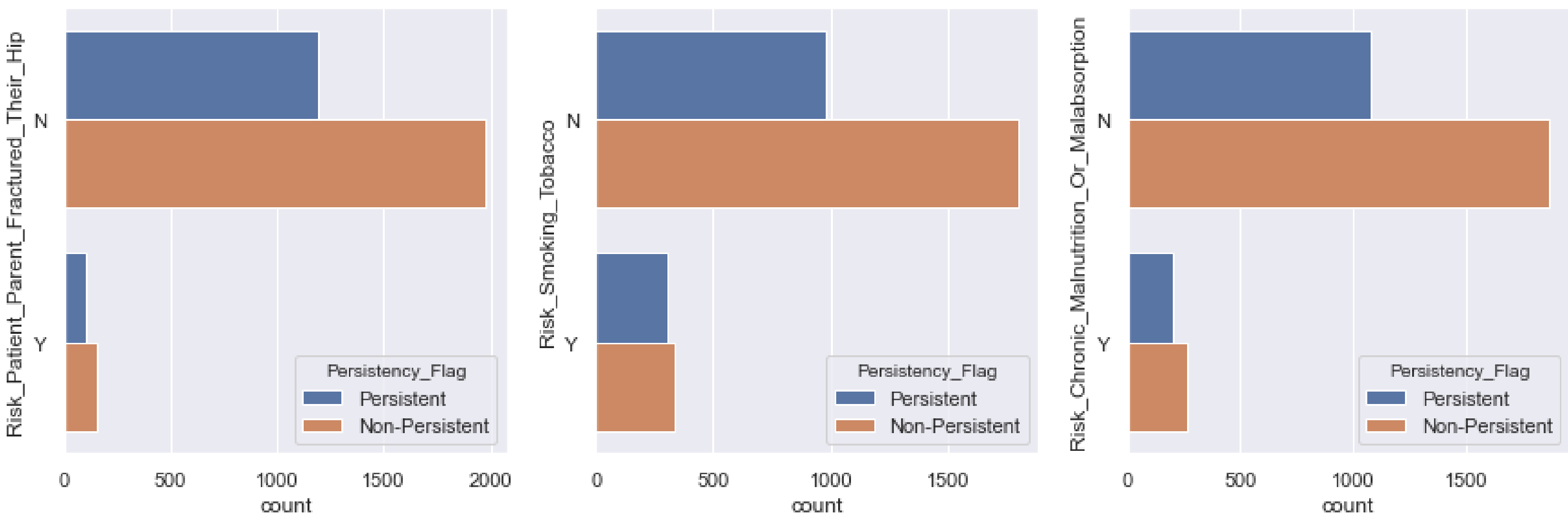
Risk_Rheumatoid_Arthritis



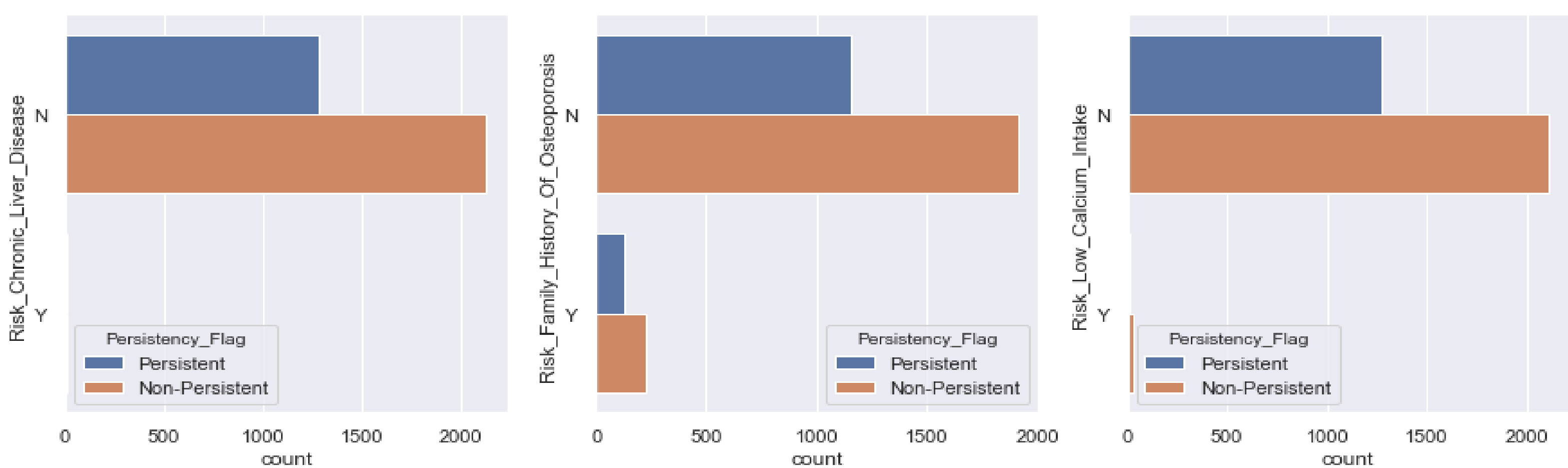
Risk_Untreated_Early_Menopause



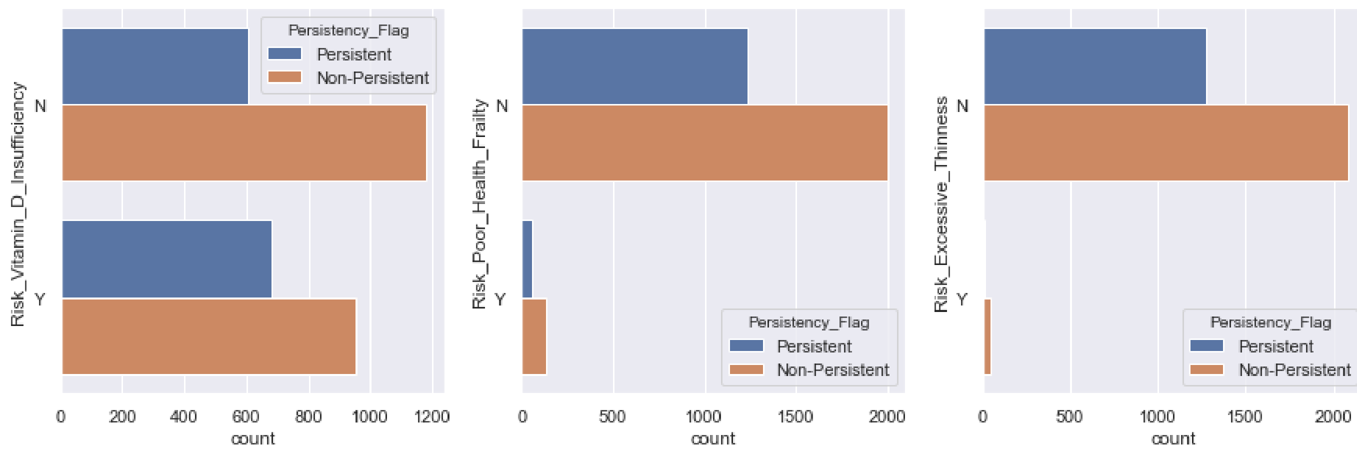
Risk_Chronic_Malnutrition_Or_Malabsorption



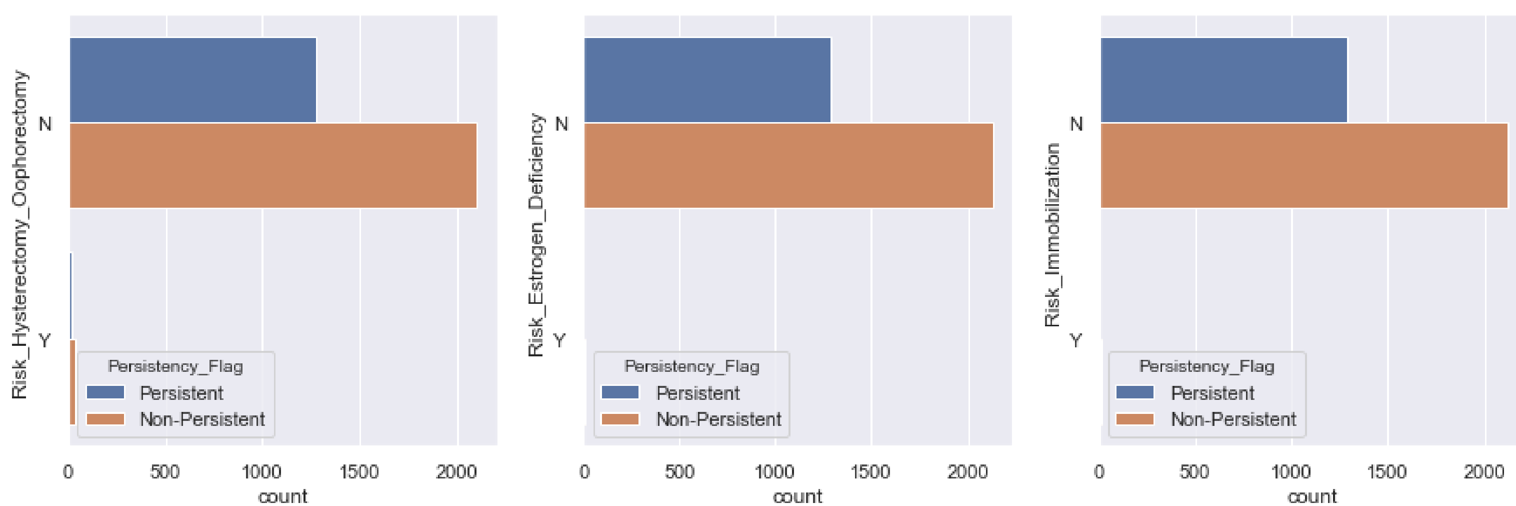
Risk_Low_Calcium_Intake



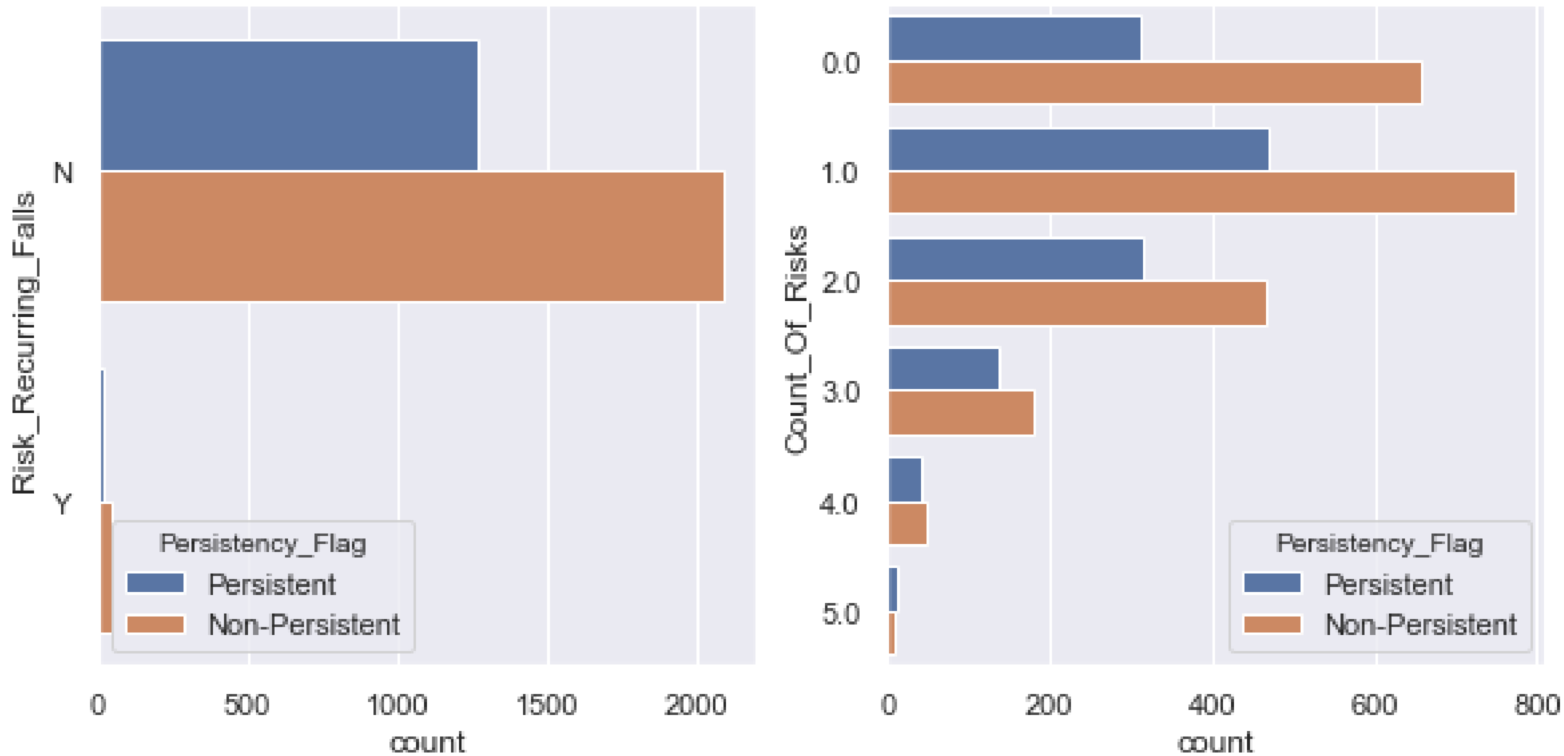
Risk_Excessive_Thinness



Risk_Immobilization



Count_Of_Risks



Summary of EDA

- Many factors are found to be related to persistency and also people from different regions has different behavior related to persistency that depends on the type of specialty.

- The number of people included in dataset varies and also the specialty that is found to have different behavior in regions .

Modeling Techniques:

- Considering the nature of target variable the classification modeling techniques are most suitable for present study. This is a problem of binary classification and models logistic regression , decision tree can be used easily.

- We conduct our experiment by implementing the following classification models:

    - Logistic Regression
    - Decision Tree
    - Random Forest
    - AdaBoost
    - XGBoost

Model Development and Evaluation

The five models are fitted to the data by splitting them into training and testing data set into 70 and 30 percent ratio. The performance of all models are compared using the Accuracy, Precision, Recall, AUC and F1 score. The results are displayed in figure below. The model logistic regression is found to perform best in all models.

Algorithm	Accuracy	Precision	Recall	AUC	F1 Score
Decision Tree	0.765564	0.809524	0.513854	0.817766	0.628659
Random Forest	0.774319	0.766990	0.596977	0.839901	0.671388
Logistic Regression	0.804475	0.796970	0.662469	0.878546	0.723521
Ada Boost	0.807393	0.797015	0.672544	0.878153	0.729508
XGBoost	0.788911	0.747253	0.685139	0.862814	0.714849

ROC Curve of five models: