# Week 10 Project Deliverable

**Group Name:** Mike
**Member Name:** Mike Wang
**Email**: yuqiao.wang@vanderbilt.edu
**Country:** USA
**College:** Vanderbilt University
**Batch Code**: LISUM21
**Specialization:** Data Science
**Date**: 7/09/2023
**Submitted to**: Data Glacier

## 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.

The main objective is to build a classification model to predict the NTM drug persistence of patients based on several factors.

# Project Schedule

| **Week** | **Date** | **Goal** |
|---|---|---|
| 07 | 06/19/2023 | Problem Statement, Data Preview |
| 08 | 06/26/2023 | Data Preprocessing |
| 09 | 07/02/2023 | Data Prepare and cleaning |
| 10 | 07/09/2023 | EDA |
| 11 | 07/16/2023 | Recommendation and Model Suggestion |
| 12 | 07/23/2023 | Model Building and Evaluation |

| | | |
|---|---|---|
| 13 | 07/30/2023 | Presentation |

# Data Understanding

## Predictor Variables

Demographic features

| | |
|---|---|
| Age | Age of the patient during their therapy |
| Race | Race of the patient from the patient table |
| Region | Region of the patient from the patient table |
| Ethnicity | Ethnicity of the patient from the patient table |
| Gender | Gender of the patient from the patient table |

Patient features

| | |
|---|---|
| IDN Indicator | Flag indicating patients mapped to IDN |
| NTM - Physician Specialty | Specialty of the HCP that prescribed the NTM Rx |
| NTM - T-Score | T Score of the patient at the time of the NTM ... |
| Change in T Score | Change in Tscore before starting with any ther... |
| NTM - Risk Segment | Risk Segment of the patient at the time of the... |
| Change in Risk Segment | Change in Risk Segment before starting with an... |
| NTM - Multiple Risk Factors | Flag indicating if patient falls under multip... |
| NTM - Dexa Scan Frequency | Number of DEXA scans taken prior to the first ... |
| NTM - Dexa Scan Recency | Flag indicating the presence of Dexa Scan befo... |
| Dexa During Therapy | Flag indicating if the patient had a Dexa Scan... |
| NTM - Fragility Fracture Recency | Flag indicating if the patient had a recent fr... |
| Fragility Fracture During Therapy | Flag indicating if the patient had fragility f... |
| NTM - Glucocorticoid Recency | Flag indicating usage of Glucocorticoids (>=7.... |
| Glucocorticoid Usage During Therapy | Flag indicating if the patient had a Glucocort... |
| NTM - Injectable Experience | Flag indicating any injectable drug usage in t... |
| NTM - Risk Factors | Risk Factors that the patient is falling into.... |
| NTM - Comorbidity | Comorbidities are divided into two main catego... |
| NTM - Concomitancy | Concomitant drugs recorded prior to starting w... |
| Adherence | Adherence for the therapies |

## Target Variable:

| | |
|---|---|
| Persistency_Flag | Flag indicating if a patient was persistent or... |

Boolean Predictor Variable List: (All the variables in the list have two values of Yes or No)

```
['Gluco_Record_Prior_Ntm',
 'Gluco_Record_During_Rx',
 'Dexa_During_Rx',
 'Frag_Frac_Prior_Ntm',
 'Frag_Frac_During_Rx',
 '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']
```

All these data will be dummy coded with 0 and 1 values for future classification model

Columns with quantitative data:

| | Dexa_Freq_During_Rx | Count_Of_Risks |
|---|---|---|
| **0** | 0 | 0 |
| **1** | 0 | 0 |
| **2** | 0 | 2 |
| **3** | 0 | 1 |
| **4** | 0 | 1 |
| **...** | ... | ... |
| **3419** | 0 | 1 |
| **3420** | 0 | 0 |
| **3421** | 7 | 1 |
| **3422** | 0 | 0 |
| **3423** | 0 | 1 |

| | 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 |

No outliers and missing values exist in the dataset

# Data Cleaning and Transformation

Since there is no missing data in the dataset, the only values that need to be cleaned was features that contains ‘Unknown’ Values:

```
# impute unknown value
df.loc[df['Change_Risk_Segment'] == 'Unknown', 'Change_Risk_Segment'] = 'No change'
df.loc[df['Change_T_Score'] == 'Unknown', 'Change_T_Score'] = 'No change'
```

For unknown changes, I grouped them all into unchanged group which is the majority of the variable

```
def transform_speciality(value):
#     transform medical speciality
    if 'MEDICINE' in value.split(' '):
        return 'MEDICINE'
    elif 'SURGERY' in value.split(' '):
        return 'SURGERY'
    elif df['Ntm_Speciality'].value_counts()[value] < 10 or value == 'Unknown':
        return 'OTHER'
    return value
```

Cleaned and transform medical specialty to decrease the variable numbers.

Use label encoding and one hot encoding to transform different types of categorical features

```
# label encoding variables with only two values to 0 and 1
label(df, 'Gender')
label(df, 'Ntm_Specialist_Flag')
label(df, 'Risk_Segment_Prior_Ntm')
label(df, 'Tscore_Bucket_Prior_Ntm')
label(df, 'Adherent_Flag')
```

```
ohe = onehot_encoder.fit(df[['Ethnicity','Age_Bucket','Ntm_Speciality_Bucket',
                                                      'Race','Risk_Segment_During_Rx','Tscore_Bucket_During_Rx',
                                                       'Change_T_Score','Change_Risk_Segment','Ntm_Speciality']])
```

```
# one hot encode other features into 0 or 1

new_df = pd.DataFrame(ohe.transform(df[['Ethnicity','Age_Bucket','Ntm_Speciality_Bucket',
                                                      'Race','Risk_Segment_During_Rx','Tscore_Bucket_During_Rx',
                                                       'Change_T_Score','Change_Risk_Segment', 'Ntm_Speciality']] ).toarray(),
                      columns=ohe.get_feature_names_out())

new_df.head()
```

| | Ethnicity_Hispanic | Ethnicity_Not Hispanic | Ethnicity_Unknown | Age_Bucket_55-65 | Age_Bucket_65-75 | Age_Bucket_<55 | Age_Bucket_>75 | Ntm_Speciality_Bucket_Endo/Onc/U |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0 |
| **1** | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0 |
| **2** | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0 |
| **3** | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0 |
| **4** | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0 |

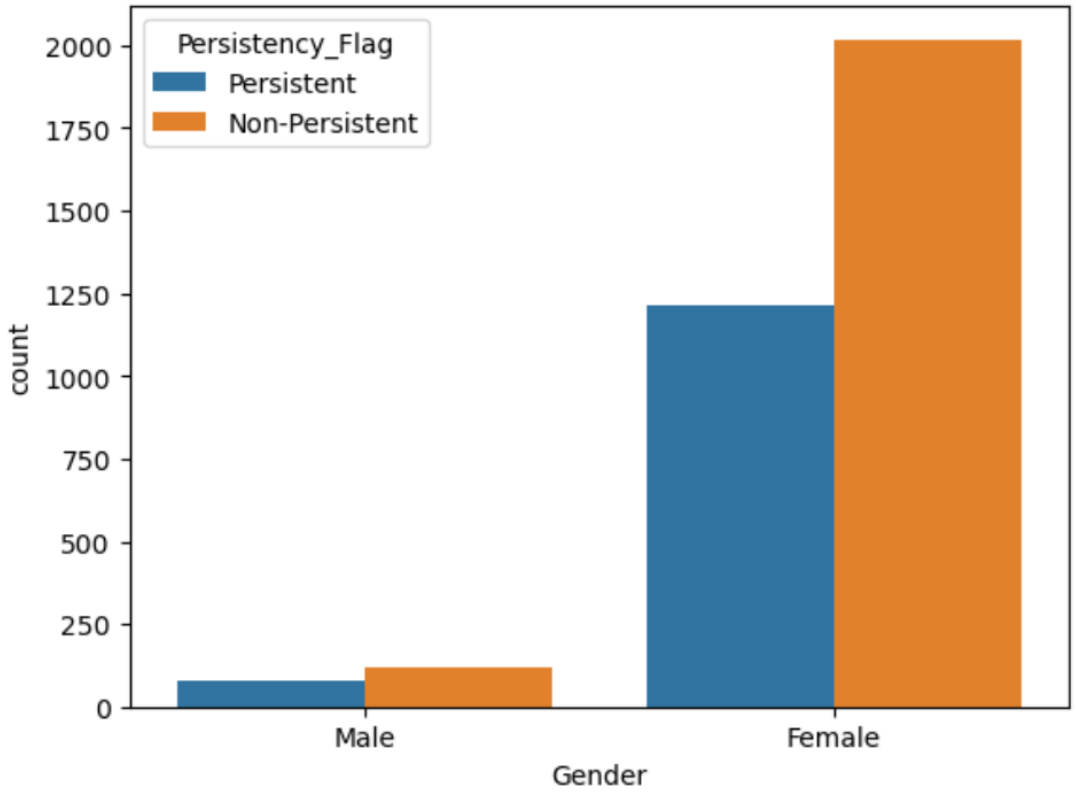
Code target variables

```python
# label encode other features
for col in bool_cols:
    label(df, col)
```

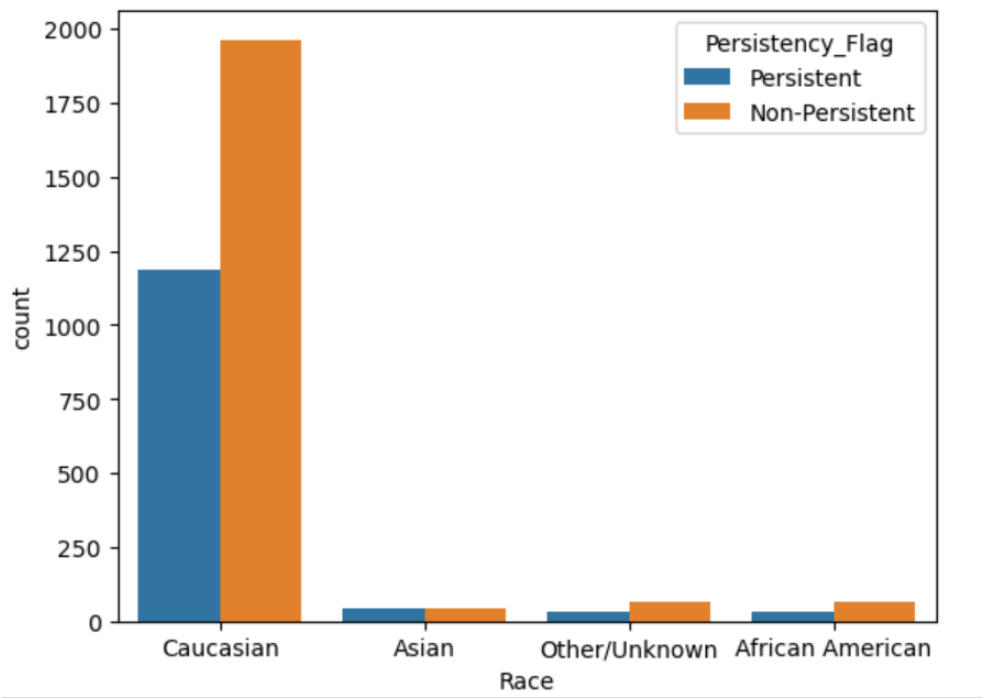
```python
# code target variables
df['target'] = np.where(df['Persistency_Flag'] == 'Persistent', 1, 0)
df['target']
```

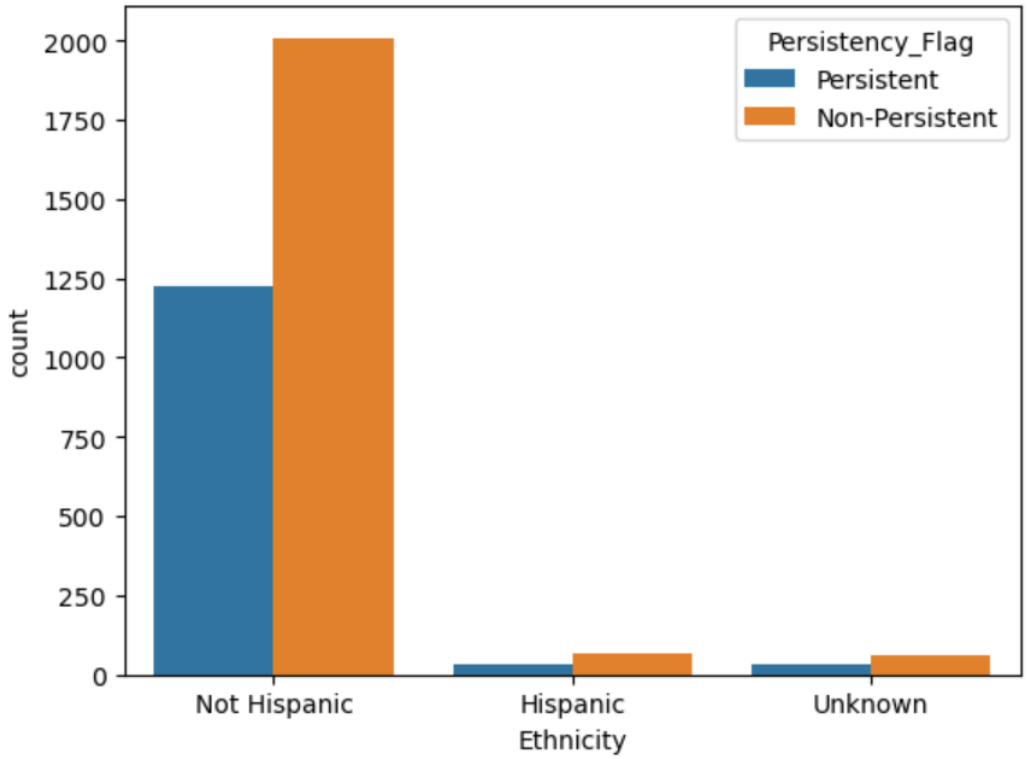
```
0       1
1       0
2       0
3       0
4       0
       ..
3419    1
3420    1
3421    1
3422    0
3423    0
Name: target, Length: 3424, dtype: int32
```

# Exploratory Data Analysis and Recommendations




There is a huge distinction between the number of Male and Female patients, we will recommend not use this feature or weight it less.


Persistency_Flag
Persistent
Non-Persistent
2000
1750
1500
1250
1000
750
500
250
0
count
Caucasian
Asian
Other/Unknown
African American
Race




Again, Race and Ethnicity groups are not balanced in the dataset, one way is that we can group other groups together into a new bigger group.

All the variables above have very unbalanced data distribution, as a result we recommend to group other values together into a new group for later model feature selection.


The healthcare dataset has many Boolean columns that contains YES or NO values. Again many of the unbalanced features will not be recommended for later use.