Group FTR

Week 9: Deliverables (Pls Scroll down)

Team Details

Name	Email	Country	College/Company	Specialization
Fabian Umeh	Fabianumeh335@gmail.com	UK	Teesside University	Data Science
Rukevwe Ovuowo	rukevwe10@gmail.com	Nigeria	GBG Data science Academy	Data Science
Olutayo Oladeinbo	oladeinboolutayo@yahoo.com	UK	Teesside University	Data Science

Problem statement

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. With an objective to gather insights on the factors that are impacting the persistency, it is necessary to build a classification for the given dataset, using the variable 'Persistency_Flag' as target variable and other attributes as prediction variables.

Project Lifecycle
Two weeks—deadline (13/11/2022)

Data Intake Report

Name: Healthcare – Persistency of a drug
Report date: 11/12/2022
Internship Batch: LISUM11: 30
Version:<1.0>
Data intake by: Fabian Umeh, Rukevwe Ovuowo, and Olutayo Oladeinbo
Data intake reviewer: Group members
Data storage location: Github
Tabular data details:
Total number of observations: 3424
Total number of files: 1
Total number of features:69
Base format of the file: .csv
Size of the data: 898 KB

PROBLEMS IN DATA (Week 8)

1.1 Missing values:

The dataset didn't contain any null values. Hence we just proceeded to encoding the columns.

1.2 Approach:

In order to encode the data, the label encoder library was imported which we applied on our non-numeric columns.

Import LabelEncoder library

```
#Data Encoding
from sklearn.preprocessing import LabelEncoder
```

Encode our target column y

```
# Select our Dependent variable 'Persistency_Flag'
y = data.Persistency_Flag
```

```
y[:5]
```

```
0        Persistent
1    Non-Persistent
2    Non-Persistent
3    Non-Persistent
4    Non-Persistent
Name: Persistency_Flag, dtype: object
```

```
# Now that we have selected our dependent and independent variable,
# It is important to note that we need to encode our columns as computer only works well with numerical data
# Since we already have a dataset with 60 columns, using pandas One-Hot_Encoding wouldn't be smart.
# Hence, we use the LabelEncoder library.
```

```
#Create an object of the Label Encoder library
Le = LabelEncoder()
```

```
# Encode our target column
y = Le.fit_transform(y)
```

```
y[:5]
```

```
array([1, 0, 0, 0, 0])
```

Encoding other non-numeric column:

```
#Columns to Encode
CTE = ['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']
```

```
# Encode the other cartegorical variables
for col in CTE:
    X[col] = Le.fit_transform(X[col])
```

```
X.head()
```

	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Speciality	Ntm_Specialist_Flag	Ntm_Speciality_Bucket	Gluco_Record_Prior_Ntm	Gluco_Record_Duri
0	1	2	1	4	3	5	0	1	0	
1	1	1	1	4	0	5	0	1	0	

It was later noticed that the number of persistent cases are fewer compared to the number of people with Non-persistent cases. To solve this imbalance in the dataset, we imported the SMOTE library. The SMOTE library is used create random samples of the minority class and add to the dataset, so there is an even number between the minority and majority class.

```
#There is an imbalance in the dataset.
#We use SMOTE to settle the imbalance in the dataset
```

```
# Create an object of our SMOTE library
sm = SMOTE(random_state=2)
```

```
# Performing oversampling on our train set
X_train,y_train = sm.fit_resample(X_train,y_train)
```