Final Project Report

Submitted by: Cyprian Nyamwaro

Email: cypriannyamwaro@gmail.com

Country: Kenya

Submission date: August 12, 2021

Internship Batch: LISUM01

Problem Description:

ABC is a pharmaceutical company that wants to understand the persistency of a drug as per the physician's prescription for a patient. This company has approached an Analytics company to automate this process of identification. This Analytics company has given responsibility to CNN and has asked to come up with a solution to automate the persistency of a drug for the client ABC.

Data Understanding

The healthcare dataset is considerable size data having 69 columns altogether. The label is the Persistency Flag which is a binary data having value as True or False depending on the other features. The first column being the unique id of the patient is of no use for us as this is not going to help us in training the model. Hence the first thing we do is drop that column. We have analyzed the dataset and upon analysis we found that there are only few numerical data column and rest are either binary or string value.

Business Understanding

The pharma company ABC wants to understand about the persistency of a drug for a patient. There are a bunch of Non-Tuberculous Mycobacterial (NTM) infection data. ABC company wants to know whether a patient is persistent or not depending on the prescription data. Depending on the persistency count, ABC pharma company would produce medicines in that quantity so that they can run their business strategically.

DATASETS

Bucket	Variable	Variable Description
Unique Row Id	Patient ID	Unique ID of each patient
Target Variable	Persistency_Flag	Flag indicating if a patient was persistent or not
Demographics	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
	IDN Indicator	Flag indicating patients mapped to IDN
Provider Attributes	NTM - Physician Specialty	Specialty of the HCP that prescribed the NTM Rx
Clinical Factors	NTM - T-Score	T Score of the patient at the time of the NTM Rx (within 2 years prior from rxdate)
	Change in T Score	Change in Tscore before starting with any therapy and after receiving therapy (Worsened, Remained Same, Improved, Unknown)
	NTM - Risk Segment	Risk Segment of the patient at the time of the NTM Rx (within 2 years days prior from rxdate)
	Change in Risk Segment	Change in Risk Segment before starting with any therapy and after receiving therapy (Worsened, Remained Same, Improved, Unknown)
	NTM - Multiple Risk Factors	Flag indicating if patient falls under multiple risk category (having more than 1 risk) at the time of the NTM Rx (within 365 days prior from rxdate)
	NTM - Dexa Scan Frequency	Number of DEXA scans taken prior to the first NTM Rx date (within 365 days prior from rxdate)
	NTM - Dexa Scan Recency	Flag indicating the presence of Dexa Scan before the NTM Rx (within 2 years prior from rxdate or between their first Rx and Switched Rx; whichever is smaller and applicable)
Disease/Treatment Factor	Dexa During Therapy	Flag indicating if the patient had a Dexa Scan during their first continuous therapy
	NTM - Fragility Fracture Recency	Flag indicating if the patient had a recent fragility fracture (within 365 days prior from rxdate)
	Fragility Fracture During Therapy	Flag indicating if the patient had fragility fracture during their first continuous therapy
	NTM - Glucocorticoid Recency	Flag indicating usage of Glucocorticoids (>=7.5mg strength) in the one year look-back from the first NTM Rx
	Glucocorticoid Usage During Therapy	Flag indicating if the patient had a Glucocorticoid usage during the first continuous therapy
	NTM - Injectable Experience	Flag indicating any injectable drug usage in the recent 12 months before the NTM OP Rx
	NTM - Risk Factors	Risk Factors that the patient is falling into. For chronic Risk Factors complete lookback to be applied and for non-chronic Risk Factors, one year lookback from the date of first OP Rx
	NTM - Comorbidity	Comorbidities are divided into two main categories - Acute and chronic, based on the ICD codes. For chronic disease we are taking complete look back from the first Rx date of NTM therapy and for acute diseases, time period before the NTM OP Rx with one year lookback has been applied
	NTM - Concomitancy	Concomitant drugs recorded prior to starting with a therapy(within 365 days prior from first rxdate)
	Adherence	Adherence for the therapies

DATA INTAKE REPORT

Total number of observations	3424
Total number of files	1
Total number of features	26
Base format of the file	.xlsx
Size of the data	898KB

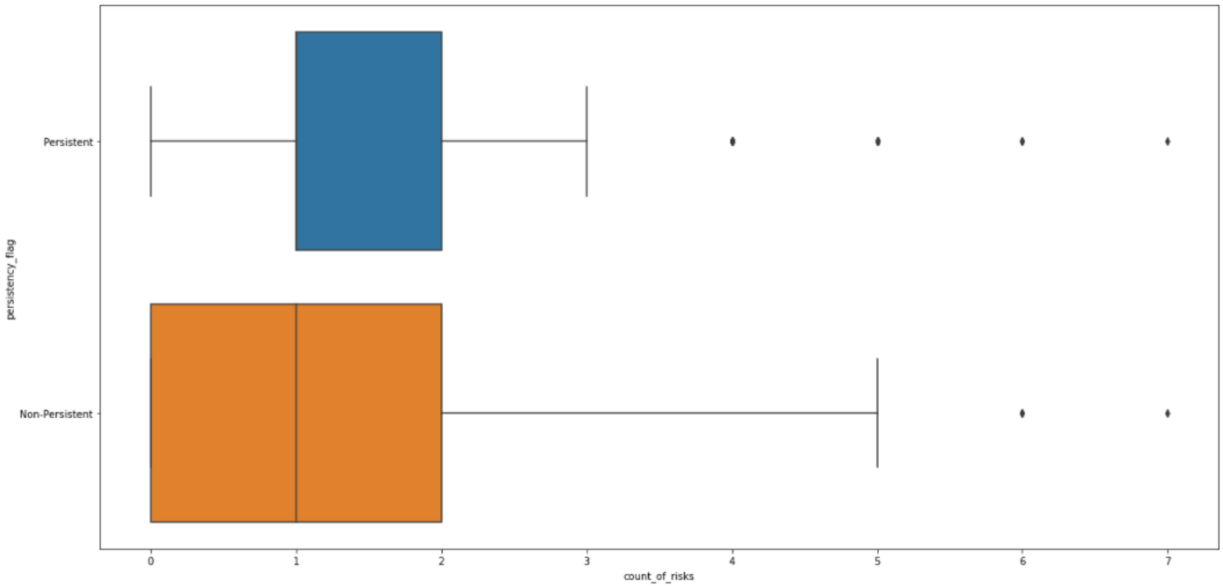
Data Types

Ptid	object
Persistency_Flag	object
Gender	object
Race	object
Ethnicity	object
Region	object
Age_Bucket	object
Ntm_Speciality	object
Ntm_Specialist_Flag	object
Ntm_Speciality_Bucket	object
Gluco_Record_Prior_Ntm	object
Gluco_Record_During_Rx	object
Dexa_Freq_During_Rx	int64
Dexa_During_Rx	object
Frag_Frac_Prior_Ntm	object
Frag_Frac_During_Rx	object
Risk_Segment_Prior_Ntm	object
Tscore_Bucket_Prior_Ntm	object
Risk_Segment_During_Rx	object
Tscore_Bucket_During_Rx	object
Change_T_Score	object
Change_Risk_Segment	object
Adherent_Flag	object
Idn_Indicator	object
Injectable_Experience_During_Rx	object
Comorb_Encounter_For_Screening_For_Malignant_Neoplasms	object
Comorb_Encounter_For_Immunization	object
Comorb_Encntr_For_General_Exam_W_O_Complaint,_Susp_Or_Reprtd_Dx	object
Comorb_Vitamin_D_Deficiency	object
Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified	object
Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx	object
Comorb_Long_Term_Current_Drug_Therapy	object
Comorb_Dorsalgia	object
Comorb_Personal_History_Of_Other_Diseases_And_Conditions	object
Comorb_Other_Disorders_Of_Bone_Density_And_Structure	object
Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias	object
Comorb_Osteoporosis_without_current_pathological_fracture	object
Comorb_Personal_history_of_malignant_neoplasm	object
Comorb_Gastro_esophageal_reflux_disease	object
Concom_Cholesterol_And_Triglyceride_Regulating_Preparations	object
Concom_Narcotics	object
Concom_Systemic_Corticosteroids_Plain	object
Concom_Anti_Depressants_And_Mood_Stabilisers	object
Concom_Fluoroquinolones	object
Concom_Cephalosporins	object
Concom_Macrolides_And_Similar_Types	object
Concom_Broad_Spectrum_Penicillins	object
Concom_Anaesthetics_General	object
Concom_Viral_Vaccines	object
Risk_Type_1_Insulin_Dependent_Diabetes	object
Risk_Osteogenesis_Imperfecta	object
Risk_Rheumatoid_Arthritis	object
Risk_Untreated_Chronic_Hyperthyroidism	object
Risk_Untreated_Chronic_Hypogonadism	object

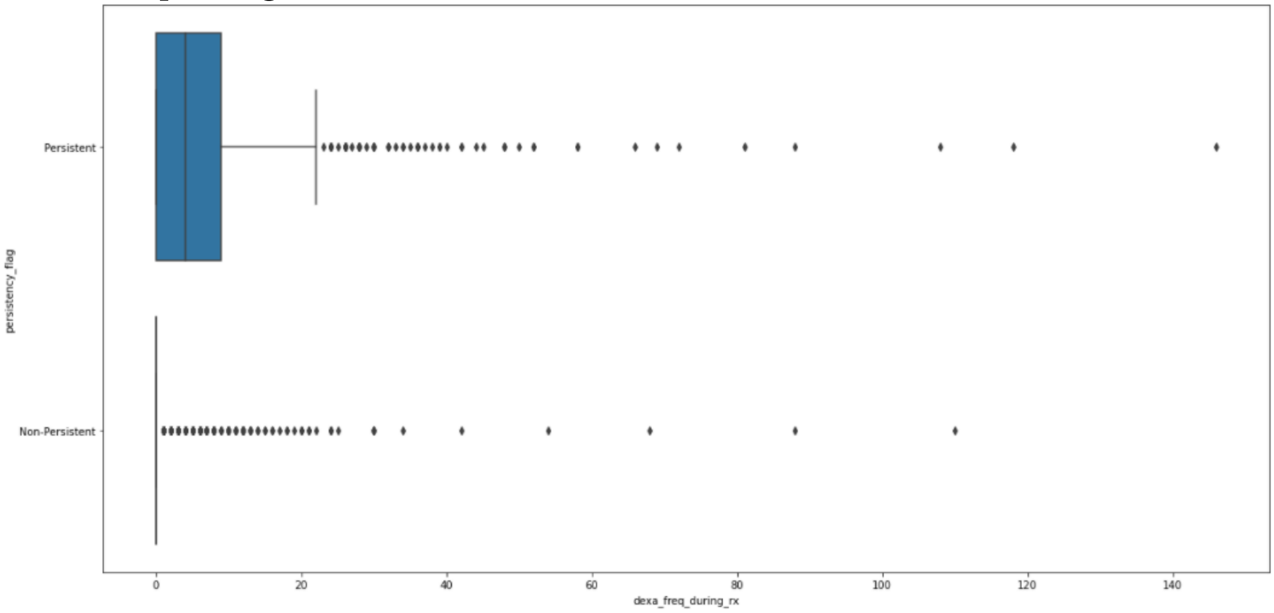
Data Problems

Null Values: This dataset has no Null values
Outliers: We have only two numerical columns and both of them have some outliers.

Count of risks



Dexa_freq_during_rx



Skewness and Kurtosis: We have only two numerical columns and both of them have some outliers.

- `count_of_risks:`

  Count of risks skweness: 0.8797905232898707

  Count of risks Kurtosis: 0.9004859968892842

- `dexa_freq_during_rx:`

  dexa_freq_during_rx skweness: 6.8087302112992285

  dexa_freq_during_rx Kurtosis: 74.75837754795428

Data Transformation

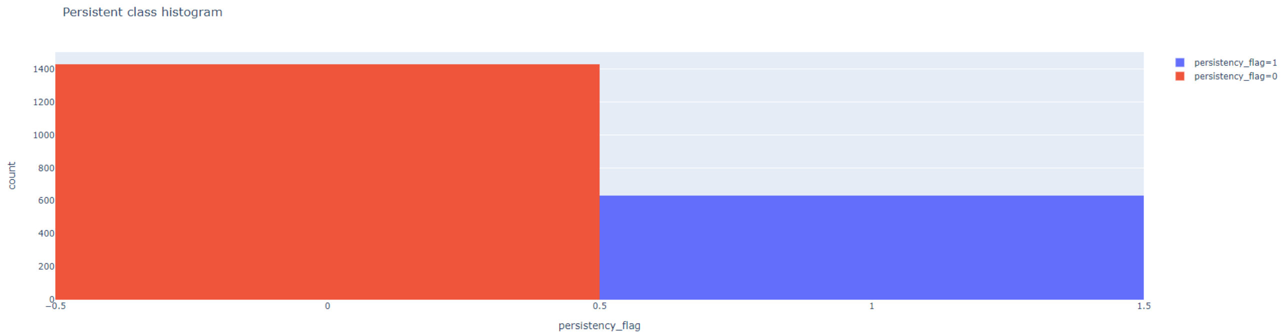
As we did not have any Null values, so we have nothing to do in this regard. We have some skewness and Kurtosis in our two numerical features, so we will scaled their values by RobustScaler() and after that remove their outliers by calculating IQR and remove data smaller/greater than two whiskers. After removing outliers from “dexa_freq_during_rx” we can check how much we have decrease in the shape of the data:

Old Shape: (3424, 69)

New Shape: (2964, 69)

We have changed all the [‘Y’, ‘N’] values to [1, 0] to train models on the data, and also we change the values of target feature in this way : ['Non-Persistent', 'Persistent'] to [0, 1].

The other thing that we had to overcome on this dataset is the unbalancing of the target feature:



since imbalanced datasets make predicting hard and don't let models work well on them! One of good things that we can do is "Up sampling", in this method we increase the records of the minority class, at last we have same count of records of each class.

The other thing that we performed on the dataset is "one hot encoding", For using classifiers we need numerical values, to do this I used One Hot Encoding that implemented by "get_dummies()" function from Pandas library, it works like this:

ID	Gender
1	Male
2	Female
3	Not Specified
4	Not Specified
5	Female

ID	Male	Female	Not Specified
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	1
5	0	1	0