TEAM MEMBERS DETAILS

GROUP NAME: PACHACUTEAM

SPECIALIZATION: Data Science

Name	Email	Country	College/Company
Andersson Romero	andromdez@gmail.com	Perú	Freelancer

Data Clean and Transform

In this section we focus on cleaning outliers, replacing unknown data, and transforming categorical data.

We start by looking to complete the unknown data, we first locate the features that contain unknown data.

```
# list columns with Unknown
columns = df.columns.to_list()
l_Unknown = []
for col in columns:
  arr = df[col].unique()
  if np.where(arr=='Unknown')[0].size != np.array(0):
    l_Unknown.append(col)
  else:
    continue
```

```
df[l_Unknown].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3424 entries, 0 to 3423
Data columns (total 6 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Ethnicity               3424 non-null   object
 1   Ntm_Speciality          3424 non-null   object
 2   Risk_Segment_During_Rx  3424 non-null   object
 3   Tscore_Bucket_During_Rx 3424 non-null   object
 4   Change_T_Score          3424 non-null   object
 5   Change_Risk_Segment     3424 non-null   object
dtypes: object(6)
memory usage: 160.6+ KB
```

Once located, if these data exceed 40% of the total, the column is deleted. Otherwise, we replace with the mode (since they are categorical data).

```
obj_rem=[]
# if Unknown values are 40% more than total size .... remove , if not ... replace with mode
for l in l_Unknown:
    val = dfn[l].value_counts().Unknown
    if val>len(dfn)*0.4:
        #l_Unknown.remove(l)
        dfn.drop(l , axis=1 , inplace=True)
        obj_rem.append(l)
    else:
        dfn[l].replace(to_replace='Unknown', value=dfn[l].mode()[0], inplace=True)
#print(l_Unknown)
dfn['Race'].replace(to_replace='Other/Unknown', value=dfn['Race'].mode()[0], inplace=True)
```

For the second part, we use quantiles to remove outliers. Additionally, remove them from the Ptid column.

```
cols = numerical_df

Q1 = dfo[cols].quantile(0.25)
Q3 = dfo[cols].quantile(0.75)
IQR = Q3 - Q1

dfo = dfo[~((dfo[cols] < (Q1 - 1.5 * IQR)) |(dfo[cols] > (Q3 + 1.5 * IQR))).any(axis=1)]
```

```
dfo.reset_index(drop=True, inplace=True)

dfo.shape
dfo.drop('Ptid', axis=1, inplace=True)
```

On the other hand, we convert the categorical variables to integers.

```
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from sklearn.compose import make_column_selector as selector
numerical_columns_selector = selector(dtype_exclude=object)
categorical_columns_selector = selector(dtype_include=object)

numerical_df = numerical_columns_selector(dfo)
categorical_df = categorical_columns_selector(dfo)

dff = dfo.copy()
```

We first store the features with two categories and three or more categories.

```
# list columns with number
columnso = categorical_df
l_2v = []
l_3vm = []
for col in columnso:
  arr = dfo[col].unique()
  if len(arr) == 2:
    l_2v.append(col)
  else:
    l_3vm.append(col)
```

We convert, with LabelEncoder()

2 var

```
[ ]  for dv in l_2v:
        le = LabelEncoder()
        dff[dv] = le.fit_transform(dff[dv])
```

3 var+

```
[ ]  for dv in l_3vm:
        le = LabelEncoder()
        dff[dv] = le.fit_transform(dff[dv])
```

```
dff.head()
```

	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntm_Speciality	Ntm_Specialist_Flag	Ntm_Speciality_Bucket	Gluco_Record_Prior_Ntm	...	R
0	1	1	2	1	4	3	5	0	1	0	...	
1	0	1	1	1	4	0	5	0	1	0	...	
2	0	0	2	0	0	1	5	0	1	0	...	
3	0	0	2	1	0	3	5	0	1	0	...	
4	0	0	2	1	0	3	5	0	1	1	...	

5 rows × 64 columns

We save the clean and transformed dataset.

Save clean and transform data

```
[ ]  dff.to_csv("healthcare_clean_transf_data.csv", index=False)
```

DATA INTAKE REPORT

Name: Healthcare project

Report date: 08/04/2022

Internship Batch: LISUM07

Version: 1.0

Data intake by: PACHACUTEAM

Data intake reviewer:<intern who reviewed the report>

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	.xlsx
Size of the data	899 KB

GITHUB LINK: https://github.com/And2300/Healthcare_PersistencyDrug