Data Glacier

Your Deep Learning Partner

Healthcare - Persistency of Drugs

Groupe name: Ehiafarin
Country: Iran
Groupe member: Alireza Ehiaei
Email: arh.ehiaei@yahoo.com

Specialization: Data Science
Data Cleansing and Transformation

September-2021

Table Of Contents

Introduction

Every dataset has missing values that need to be treated appropriately to create a robust model. In this practice, I have discussed two ways to handle missing values. There is no unique rule to handle missing values in a specific manner.

One can use various methods on different features depending on how and what the data is about. Indeed, each dataset needs some specific approaches to handle missing values.

Problem Statement

The dataset in this research is related to drug treatment, and except two features which are integer and have no missing values, there are 67 other features with categorical type which some of them have unknown values.

The aim is predicting drug persistency of a patient. And there is Persistency-Flag column as target column, with two labels of Persistent and Non-Persistent.o, It is a binary classification problem.

Problem Statement

There are different ways to handle missing values including:

- Deleting Rows with missing values
- Impute missing values for continuous variable
- Impute missing values for categorical variable
- Using Algorithms that support missing values
- Prediction of missing values

https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e

Problem Statement

Delete coloumns or rows with missing values:

Missing values can be handled by deleting the rows or columns having null values. For example, one can choose If 50% or 60% of or more of the rows as null then the entire column can be dropped. Similarly, If a row has e.g. one or more columns values as null can also be dropped.

Dealing with missing values, method 1:

Dropping columns or rows including missing values:

For example, we can remove the columns with more than 30%, and drop the rows with 20% missing values. There are 3424 rows, we remove 4 columns which have more than 30% missing values.

```
list_column_with_nulls = []

for column in df.columns:
    n_missed = len(df[df.loc[:,column]=='
    percent = n_missed/3424
    if percent > 0.3:
        print(column)
        print("percentage of missing valu
        list_column_with_nulls.append(col
```

```
Risk_Segment_During_Rx
percentage of missing values : 0.44
Tscore_Bucket_During_Rx
percentage of missing values : 0.44
Change_T_Score
percentage of missing values : 0.44
Change_Risk_Segment
percentage of missing values : 0.65
```

```
len(list_column_with_nulls)
```

```
4
```

Dealing with missing values

New data set has 4 less columns.

```
drug_Data.head()
```

	Ptid	Persistency_Flag	Gender	Race	Ethnicity	Regi
0	P1	Persistent	Male	Caucasian	Not Hispanic	We
1	P2	Non-Persistent	Male	Asian	Not Hispanic	We
2	P3	Non-Persistent	Female	Other/Unknown	Hispanic	Midwe
3	P4	Non-Persistent	Female	Caucasian	Not Hispanic	Midwe
4	P5	Non-Persistent	Female	Caucasian	Not Hispanic	Midwe

```
drug_Data.shape
```

(3424, 69)

```
df = drug_Data
```

```
df.drop([col for col in list_colum
```

```
df.head()
```

	Ptid	Persistency_Flag	Gender	Ra
0	P1	Persistent	Male	Caucasi
1	P2	Non-Persistent	Male	Asi
2	P3	Non-Persistent	Female	Other/Unknov
3	P4	Non-Persistent	Female	Caucasi
4	P5	Non-Persistent	Female	Caucasi

```
df.shape
```

(3424, 65)

Dealing with missing values

Similarly, we drop the rows with just 2% missing values.

```
rows_with_missing_values = []
for i in range(len(df)) :
    n=0
    s = []
    for j in df.columns:
        if (df.loc[i,j]=='Other/Unknown') | (df.loc[i,j]=='Unknown'):
            n=n+1
            s.append(j)
    if (n/69 > 0.02):
        print('row :',i, '     number of missing values :', n)
        print('Columns including missing values :',s)
        rows_with_missing_values.append(i)
```

```
row : 154      number of missing values : 2
Columns including missing values : ['Race', 'Ethnicity']
row : 218      number of missing values : 2
Columns including missing values : ['Race', 'Ethnicity']
row : 221      number of missing values : 2
Columns including missing values : ['Race', 'Ntm_Speciality']
row : 406      number of missing values : 2
Columns including missing values : ['Race', 'Ethnicity']
```

Dealing with missing values

Finally, we have a data set with
3379 rows and 65 columns.

	Ptid	Persistency_Flag	Gender	Rac
0	P1	Persistent	Male	Caucasia
1	P2	Non-Persistent	Male	Asia
2	P3	Non-Persistent	Female	Other/Unknow
3	P4	Non-Persistent	Female	Caucasia
4	P5	Non-Persistent	Female	Caucasia

```
df.shape
```

```
(3379, 65)
```

Dealing with missing values, method 2

Imputation

Since the columns including missing values in this research are categorical, their missing values can be replace with the most frequent category. We could use this method as first step, but If the number of missing values is very large the accuracy of final prediction would be reduced. Now, we removed some rows and columns, and impute the rest of missing values in each column with the mode of that column.

Dealing with missing values

By assuming as nulls, still there are four columns including missing values with maximum 290 missed values:

```
Race                64
Ethnicity           58
Region              55
Age_Bucket           0
Ntm_Speciality     290
```

Dealing with missing values

After replacing nulls with the mode of each column, no missing value remains.

Filling nulls of each column with the most frequent (mode) of each column

```
df = df.fillna(df.mode().iloc[0])
```

Any null in data?

```
df.isnull().values.any()
```

```
False
```

Dealing with missing values, method 3

Prediction of missed values

Another way to treat with missing values is predicting them. For example, assume that we want to predict all null values of the column ‘Risk_Segment_During_Rx’ based on other columns. First, object data type have been converted to float type to be used in the LinearDiscriminantAnalysis model. Then, by finding missed values in target column and reloving their rows we have data set for trainning the model.

Dealing with missing values, method 3

Finally, by using the model the missed values of the last column are predicted and filled.

```
from numpy import nan
from pandas import read_csv
from sklearn.discriminant_analysis import LinearDiscriminantAnalysi
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

X = Data_without_nulls.iloc[:,:-1]
y = Data_without_nulls.iloc[:,-1]
# define the model
model = LinearDiscriminantAnalysis()
# define the model evaluation procedure
cv = KFold(n_splits=3, shuffle=True, random_state=1)
# evaluate the model
result = cross_val_score(model, X, y, cv=cv, scoring='accuracy')
# report the mean performance
print('Accuracy: %.3f' % result.mean())
```

```
Accuracy: 0.926
```

Dealing with missing values, method 3

Finally, there is no missed values:

Predicting the last column using the model

```
]:
data_ =[]
data_ =pd.DataFrame(data_)
data_ = pd.DataFrame(model.predict(data_without_null_1.iloc[:,:-1]))

```

Replacing predicted values just for missed values

```
]:
for i in range(len(data_without_null_1)) :
    if(pd.isnull(data_without_null_1.loc[i,'Risk_Segment_During_Rx'])):
        data_without_null_1.loc[i,'Risk_Segment_During_Rx']= data_.iloc[i,0]
```

Github link

Github link of notebook code:

https://github.com/Alireza-Ehiaei/Data_Sciences/tree/main/Drug_Persistency1/Cleaning_data

Thank You