Lab2-prdom

March 22, 2021

[1]:
```
import random
import math
import pandas as pd
from category_encoders import TargetEncoder
from category_encoders import OneHotEncoder
from category_encoders import CountEncoder
from category_encoders import BinaryEncoder
df=pd.read_csv("allegro-api-transactions.csv")
import warnings
warnings.filterwarnings('ignore')
```

1 Zadanie domowe 2

Zajmiemy się przekształceniem bazy allegro w pierwszej części konując jej zmienne kategoryczne, a w drugiej uzupełniając brakujące dane.

1.1 Część pierwsza

Rozpoczniemy od zakodowania zmiennej kategorycznej *it_location* za pomocą target encoding.

[2]:
```
df['it_location'].describe()
```

[2]:
```
count       420020
unique       10056
top       Warszawa
freq         23244
Name: it_location, dtype: object
```

Zauważamy, że kolumna ta przyjmuje 10056 unikalnych wartości, co świadczy o tym, że metoda *one hot encoding* nie będzie tu najlepszym pomysłem, bowiem stworzyłaby ona właśnie tyle kolumn. Natomiast *target encoding* pozwoli zmniejszyć ilość zmiennych.

[5]:
```
te=TargetEncoder()
df1=df.copy()
df1['it_location_encoded']=te.fit_transform(df1['it_location'],df1['price'])
df1.head()
```

[5]:
```
   lp                 date     item_id  \
0   0  2016-04-03 21:21:08  4753602474
1   1  2016-04-03 15:35:26  4773181874
2   2  2016-04-03 14:14:31  4781627074
3   3  2016-04-03 19:55:44  4783971474
4   4  2016-04-03 18:05:54  4787908274

                                        categories  pay_option_on_delivery  \
0  ['Komputery', 'Dyski i napędy', 'Nośniki', 'No…                       1
1  ['Odzież, Obuwie, Dodatki', 'Bielizna damska',…                       1
2  ['Dom i Ogród', 'Budownictwo i Akcesoria', 'Śc…                       1
3  ['Książki i Komiksy', 'Poradniki i albumy', 'Z…                       1
4  ['Odzież, Obuwie, Dodatki', 'Ślub i wesele', '…                       1

   pay_option_transfer         seller   price  it_is_allegro_standard  \
0                    1    radzioch666   59.99                       1
1                    1  InwestycjeNET    4.90                       1
2                    1    otostyl_com  109.90                       1
3                    1        Matfel1   18.50                       0
4                    1      PPHU_RICO   19.90                       1

   it_quantity  it_is_brand_zone  it_seller_rating          it_location  \
0          997                 0             50177             Warszawa
1         9288                 0             12428             Warszawa
2          895                 0              7389               Leszno
3          971                 0             15006  Wola Krzysztoporska
4          950                 0             32975            BIAŁYSTOK

             main_category  it_location_encoded
0                Komputery            85.423398
1  Odzież, Obuwie, Dodatki            85.423398
2              Dom i Ogród            61.990914
3        Książki i Komiksy            35.433365
4  Odzież, Obuwie, Dodatki           117.191956
```

Stworzyliśmy w ramce danych nową kolumnę *it_location_encoded*, w której umieszczone są zakodowane wartości *it_location* za pomocą *target encoding*.

Target encoding polega na kodowaniu zmiennej kategorycznej jako: średniej wartości targetu(tu *price*), dla danej kategorii.

Zalety: - prosty i szybki - nie zwiększa wymiaru bazy danych

Wady: - jest zależny od rozkładu targetu, co oznacza, że ma skłonności do overfitting-u - jest specyficzny co do danych i żadko pokazuje znaczącą poprawę

[6]:
```
ohe=OneHotEncoder(use_cat_names=True)
df2=df.copy()
df2=df2.join(ohe.fit_transform(df2.main_category))
```

```
df2.head()
```

[6]:

```
   lp                 date     item_id  \
0   0  2016-04-03 21:21:08  4753602474
1   1  2016-04-03 15:35:26  4773181874
2   2  2016-04-03 14:14:31  4781627074
3   3  2016-04-03 19:55:44  4783971474
4   4  2016-04-03 18:05:54  4787908274

                                          categories  pay_option_on_delivery  \
0  ['Komputery', 'Dyski i napędy', 'Nośniki', 'No…                       1
1  ['Odzież, Obuwie, Dodatki', 'Bielizna damska',…                       1
2  ['Dom i Ogród', 'Budownictwo i Akcesoria', 'Śc…                       1
3  ['Książki i Komiksy', 'Poradniki i albumy', 'Z…                       1
4  ['Odzież, Obuwie, Dodatki', 'Ślub i wesele', '…                       1

   pay_option_transfer         seller   price  it_is_allegro_standard  \
0                    1    radzioch666   59.99                       1
1                    1  InwestycjeNET    4.90                       1
2                    1    otostyl_com  109.90                       1
3                    1        Matfel1   18.50                       0
4                    1      PPHU_RICO   19.90                       1

   it_quantity  …  main_category_Filmy  main_category_Fotografia  \
0          997  …                    0                         0
1         9288  …                    0                         0
2          895  …                    0                         0
3          971  …                    0                         0
4          950  …                    0                         0

  main_category_Biuro i Reklama main_category_Instrumenty  \
0                             0                          0
1                             0                          0
2                             0                          0
3                             0                          0
4                             0                          0

   main_category_Muzyka  main_category_Konsole i automaty  \
0                     0                                 0
1                     0                                 0
2                     0                                 0
3                     0                                 0
4                     0                                 0

   main_category_Sprzęt estradowy, studyjny i DJ-ski  \
0                                                  0
1                                                  0
```

```
2                                                    0
3                                                    0
4                                                    0

   main_category_Antyki i Sztuka  main_category_Bilety  \
0                              0                     0
1                              0                     0
2                              0                     0
3                              0                     0
4                              0                     0

   main_category_Nieruchomości
0                             0
1                             0
2                             0
3                             0
4                             0

[5 rows x 41 columns]
```

Stworzyliśmy w ramce danych nowe kolumny, które kodują wartości *main_category* za pomocą one hot encoding. Dokonuje tego poprzez stworzenie po jednej kolumnie dla każdej unikalnej wartości w kolumnie *main_category*

Zalety: - działa dobrze z nominalnymi danymi

Wady: - może stworzyć naprawdę duże ramki danych

[7]:
```
be=BinaryEncoder()
df3=df.copy()
df3=df3.join(be.fit_transform(df3.main_category))
df3.head()
```

[7]:
```
   lp                 date     item_id  \
0   0  2016-04-03 21:21:08  4753602474
1   1  2016-04-03 15:35:26  4773181874
2   2  2016-04-03 14:14:31  4781627074
3   3  2016-04-03 19:55:44  4783971474
4   4  2016-04-03 18:05:54  4787908274

                                          categories  pay_option_on_delivery  \
0  ['Komputery', 'Dyski i napędy', 'Nośniki', 'No…                       1
1  ['Odzież, Obuwie, Dodatki', 'Bielizna damska',…                       1
2  ['Dom i Ogród', 'Budownictwo i Akcesoria', 'Śc…                       1
3  ['Książki i Komiksy', 'Poradniki i albumy', 'Z…                       1
4  ['Odzież, Obuwie, Dodatki', 'Ślub i wesele', '…                       1

   pay_option_transfer         seller   price  it_is_allegro_standard  \
```

```
0                   1    radzioch666   59.99                      1
1                   1  InwestycjeNET    4.90                      1
2                   1    otostyl_com  109.90                      1
3                   1        Matfel1   18.50                      0
4                   1      PPHU_RICO   19.90                      1

   it_quantity  it_is_brand_zone  it_seller_rating          it_location  \
0          997                 0             50177             Warszawa
1         9288                 0             12428             Warszawa
2          895                 0              7389               Leszno
3          971                 0             15006  Wola Krzysztoporska
4          950                 0             32975            BIAŁYSTOK

             main_category  main_category_0  main_category_1  main_category_2  \
0                Komputery                0                0                0
1  Odzież, Obuwie, Dodatki                0                0                0
2              Dom i Ogród                0                0                0
3        Książki i Komiksy                0                0                0
4  Odzież, Obuwie, Dodatki                0                0                0

   main_category_3  main_category_4  main_category_5
0                0                0                1
1                0                1                0
2                0                1                1
3                1                0                0
4                0                1                0
```

Stworzyliśmy w ramce danych nowe kolumny, które kodują wartości *main_category* za pomocą *binary encoding*. Działa ono podobnie jak *one hot*, ale przechowuje wartości jako binarne bitstringi.

Co prawda metoda ta nie tworzy tak wielu kolumn jak one hot, ale przypomina przez to zbytnio *ordinal encoder* tyle, że w postaci binarnej.

[8]:
```
ce=CountEncoder()
df4=df.copy()
df4['main_category_encoded']=ce.fit_transform(df4.main_category)
df4.head()
```

[8]:
```
   lp                date     item_id  \
0   0  2016-04-03 21:21:08  4753602474
1   1  2016-04-03 15:35:26  4773181874
2   2  2016-04-03 14:14:31  4781627074
3   3  2016-04-03 19:55:44  4783971474
4   4  2016-04-03 18:05:54  4787908274

                                          categories  pay_option_on_delivery  \
0  ['Komputery', 'Dyski i napędy', 'Nośniki', 'No…                       1
```

```
1  ['Odzież, Obuwie, Dodatki', 'Bielizna damska',…                    1
2  ['Dom i Ogród', 'Budownictwo i Akcesoria', 'Śc…                    1
3  ['Książki i Komiksy', 'Poradniki i albumy', 'Z…                    1
4  ['Odzież, Obuwie, Dodatki', 'Ślub i wesele', '…                    1

   pay_option_transfer         seller   price  it_is_allegro_standard  \
0                    1    radzioch666   59.99                       1
1                    1  InwestycjeNET    4.90                       1
2                    1    otostyl_com  109.90                       1
3                    1        Matfel1   18.50                       0
4                    1      PPHU_RICO   19.90                       1

   it_quantity  it_is_brand_zone  it_seller_rating          it_location  \
0          997                 0             50177              Warszawa
1         9288                 0             12428              Warszawa
2          895                 0              7389                Leszno
3          971                 0             15006  Wola Krzysztoporska
4          950                 0             32975             BIAŁYSTOK

             main_category  main_category_encoded
0                Komputery                  14491
1  Odzież, Obuwie, Dodatki                  54257
2              Dom i Ogród                  91042
3        Książki i Komiksy                  11572
4  Odzież, Obuwie, Dodatki                  54257
```

Stworzyliśmy w ramce danych nowe kolumny, które kodują wartości *main_category* za pomocą *count encoding*. Dokonuje tego zamieniając każdą wartość kategoryczną ilością jej wystąpień.

Metoda ta może powodować wiele problemów np. gdy kolumna zawiera tylko dwie wartości kategoryczne i każda z nich występuje dokładnie tyle samo razy.

1.2 Część druga

Rozpoczniemy od ograniczenia bazy danych do zmiennych numerycznych.

[9]:
```
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from sklearn.metrics import mean_squared_error
import seaborn as sns
import numpy as np
```

[10]:
```
def seller_rating():
    answer1=[0 for i in range(10)]
    answer=pd.DataFrame()
    for i in range(10):
        copy=df.loc[:,['price', 'it_seller_rating', 'it_quantity']].copy()
```

```
        random_sample1=random.sample(range(0,df.shape[0]),math.floor(df.
↪shape[0]*0.1))
        copy.iloc[random_sample1,1]=None

        imp=IterativeImputer(max_iter=10,random_state=21)
        ans=pd.DataFrame(imp.fit_transform(copy),columns=['price',␣
↪'it_seller_rating', 'it_quantity'])
        answer1[i] = math.sqrt(mean_squared_error(df['it_seller_rating'],␣
↪ans['it_seller_rating']))

    answer['it_seller_rating']=answer1
    return answer
```

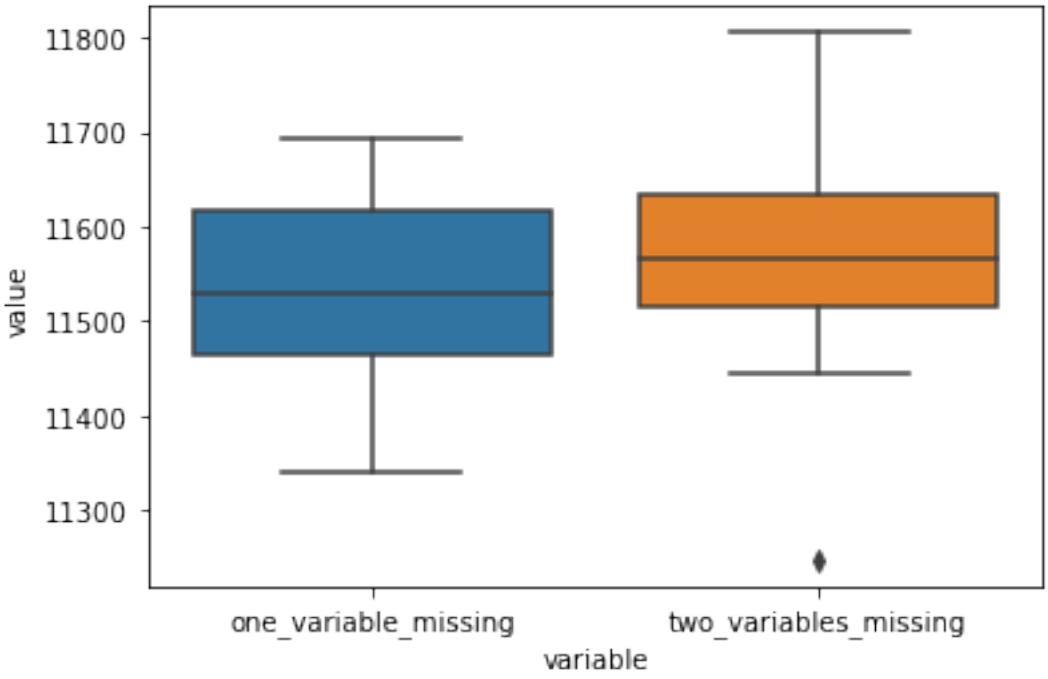
[11]:
```
def both():
    answer1=[0 for i in range(10)]
    answer2=[0 for i in range(10)]
    answer=pd.DataFrame()
    for i in range(10):
        copy=df.loc[:,['price', 'it_seller_rating', 'it_quantity']].copy()
        random_sample1=random.sample(range(0,df.shape[0]),math.floor(df.
↪shape[0]*0.1))
        random_sample2=random.sample(range(0,df.shape[0]),math.floor(df.
↪shape[0]*0.1))
        copy.iloc[random_sample1,1]=None
        copy.iloc[random_sample2,2]=None
        imp=IterativeImputer(max_iter=10,random_state=21)
        ans=pd.DataFrame(imp.fit_transform(copy),columns=['price',␣
↪'it_seller_rating', 'it_quantity'])
        answer1[i] = math.sqrt(mean_squared_error(df['it_seller_rating'],␣
↪ans['it_seller_rating']))
        answer2[i]=math.sqrt(mean_squared_error(df['it_quantity'],␣
↪ans['it_quantity']))
    answer['it_seller_rating']=answer1
    answer['it_quantity']=answer2
    return answer
```

[12]:
```
answer1=seller_rating()
```

[13]:
```
answer=both()
```

[14]:
```
plotdata=pd.DataFrame()
plotdata['one_variable_missing']=answer1['it_seller_rating']
plotdata['two_variables_missing']=answer['it_seller_rating']
sns.boxplot(data=pd.melt(plotdata),x='variable', y='value')
```

[14]:
```
<AxesSubplot:xlabel='variable', ylabel='value'>
```



Z powyższego wykresu łatwo wywnioskować, że algorytm gorzej działa, gdy brakuje więcej niż jednego argumentu.