Лабораторная работа №2 по курсу "Методы машинного обучения"

Выполнил: Саврасов П.А. группа ИУ5-24М

Задание

In [41]:

In [43]:

Out[43]:

In [44]:

Out[44]:

In [45]:

Out[45]:

In [46]:

Out[46]:

In [47]:

In [57]:

In [68]:

400

200

In []:

8

4

5

Uneducated

768805383

Выбрать набор данных (датасет), содержащий категориальные и числовые признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.) Просьба не использовать датасет, на котором данная задача решалась в лекции.<\p>

High School

Graduate

Graduate

High School

Uneducated

5

3

3

print("\t", column,"(", len(data[column].unique()), "):\t", data[column].isnull().sum())

60K-80K

Less than \$40K

80K-120K

Less than \$40K

60K-80K

Blue

Blue

Blue

Blue

Blue

Married

Single

Married

Married

NaN

39 ...

44 ...

36 ...

34 ...

21 ...

26 ...

31 ...

23 ...

21 ...

8 ...

23 ...

33 ...

14 ...

23 ...

23 ...

518

5723

3718

3612

4463

4080

3747

3135

2206

332

12691.0

8256.0

3418.0

3313.0

4716.0

Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи: • устранение пропусков в данных;

- кодирование категориальных признаков;
- нормализацию числовых признаков.

import numpy as np import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt from sklearn.impute import SimpleImputer from sklearn.preprocessing import LabelEncoder import scipy.stats as stats import warnings warnings.filterwarnings('ignore') In [42]:

def diagnostic_plots(df, variable): plt.figure(figsize=(15,6)) # гистограмма plt.subplot(1, 2, 1)

df[variable].hist(bins=30) ## Q-Q plot plt.subplot(1, 2, 2) stats.probplot(df[variable], dist="norm", plot=plt)

plt.show()

data = pd.read csv('BankChurners.csv', sep=",")

data.head() CLIENTNUM Attrition_Flag Customer_Age Gender Dependent_count Education_Level Marital_Status Income_Category Card_Category Months_on_book ... Credit_Limit Total Existing

45

60K-80K

80K-120K

60K-80K

60K-80K

40K-60K

60K-80K

60K-80K

80K-120K

19

23

25

14

14

18

25

11

22

CLIENTNUM

7.2 7.4 7.6 7.8 8.0 8.2

Total Relationship Count

Total_Revolving_Bal

0

1

0

1

1

1

1

2000

1500

1000

500

3000

2000

3000

2000

1000

5000

3000

\$120K +

M

Customer Existing 49 818770008 F Customer

Existing 51 713982108 M Customer

Existing 40 769911858 F Customer Existing 709106358 40 M Customer

columnsWithNull.append(column)

5 rows × 23 columns Заполнение пустых значений

columnsWithNull = [] print("Столбцы с пустыми значениями (название (число уникальных значений): число пустых):\n") for column in data.columns: if data[column].isnull().sum() > 0:

Столбцы с пустыми значениями (название (число уникальных значений): число пустых): Education Level (7): 1519 Marital \overline{Status} (4):

data[columnsWithNull].head(10)

Income_Category (6): 1112 **Education_Level Marital_Status Income_Category** 0

High School Married Less than \$40K Graduate Single Graduate 2 Married

3 High School Less than \$40K NaN 60K-80K Uneducated Married Graduate 40K-60K 5 Married 6 NaN Married \$120K + High School 60K-80K NaN

Single

80K-120K Single 9 Graduate imputer = SimpleImputer(missing_values=np.nan, strategy="constant", fill_value = "Unknown") nullFixedData = pd.DataFrame(data = imputer.fit_transform(data), columns=data.columns) nullFixedData[columnsWithNull].head(10)

High School Married 60K-80K Graduate Single Less than \$40K 2 Graduate Married 80K-120K 3 High School Unknown Less than \$40K

Education_Level Marital_Status Income_Category

Married

Married

Unknown Married 6 7 High School Unknown 8 Uneducated Single 9 Graduate Single

Кодирование категориальных признаков

Uneducated

Graduate

labelEnc = LabelEncoder() for column in nullFixedData.columns: if nullFixedData[column].dtype not in ['float', 'int']: nullFixedData[[column]] = pd.DataFrame(labelEnc.fit transform(nullFixedData[column].astype(str)), columns=[column]) nullFixedData.head(10)

CLIENTNUM Attrition_Flag Customer_Age Gender Dependent_count Education_Level Marital_Status Income_Category Card_Category Months_on_book ... Credit_Limit Total 7152 0

9812 1 1 2 3053

7204 3 1 501

5 2544 1 9493 6

7 9818 1401 8

9 5893

10 rows × 23 columns

plt.show()

Нормализация числовых признаков data.hist(figsize=(20,20))

> 6000 -5000 -4000 3000

2000 1000

2000

1500

1000

2500 -

2000 -

1500 -

1000

500

1500

1000

500

6000

4000

2000

1000 1500 Total_Trans_Ct

> 2000 1000

Total_Ct_Chng_Q4_Q1 Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2

2

2

3

5

2

6

2500

2000

1500

1000

500

3500 -

3000

2500

2000

1500

1000

500

5000

2

Customer Age

Months_Inactive_12_mon

Avg_Open_To_Buy

2

1

3

1

1

3

2

2

Dependent_count

Contacts_Count_12_mon

Total_Amt_Chng_Q4_Q1

4

3

4

2

1

0

2

3

3500

2500

2000

1500

1000

500

5000 +

4000 -

3000 -

2000 -

1000 -

3000

2500

2000

1000

0

0

0

0

0

0

Months on book

Credit Limit

20000

Total_Trans_Amt

10000

Naive_Bayes (Glastilization_Level_Months_Inactive_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1

800 600

diagnostic_plots(data, 'Total_Trans_Ct')

400 200 100 120 140 data['Total_Trans_Ct_sqrt'] = data['Total_Trans_Ct']**(1/2) diagnostic_plots(data, 'Total_Trans_Ct_sqrt') 1000 800 600

150 125 100 Ordered Values 50 25 -25 Theoretical quantiles Probability Plot 12

Probability Plot

10 Ordered Values Theoretical quantiles