Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Лабораторная работа №2 по дисциплине**

**«Методы машинного обучения»**

**По теме «Обработка признаков, часть 1»**

**ИСПОЛНИТЕЛЬ:**

Позняк А.А.

Группа ИУ5-22M

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2023 г.

Москва 2023

Оглавление

[Задание 3](#_Toc137078282)

[Текст программы 4](#_Toc137078283)

[Экранные формы 8](#_Toc137078284)

Задание

1. 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.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:

* устранение пропусков в данных;
* кодирование категориальных признаков;
* нормализация числовых признаков.

Текст программы

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import scipy.stats as stats

%matplotlib inline

sns.set(style="ticks")

data = pd.read\_csv('Flipkart\_mobile\_brands.csv')

data.head()

data\_features = list (zip(

#признаки

[i for i in data.columns],

zip(

#типы колонок

[str(i) for i in data.dtypes],

#проверим есть ли пропущенные значения

[i for i in data.isnull().sum() ]

)))

data\_features

# процент пропусков

[(c, data[c].isnull().mean()) for c in data.columns]

#удаление

data.dropna(subset=['Model', 'Selling Price', 'Original Price'], inplace=True)

#заполнение

data['Color'] = data['Color'].astype(str).str[0]

data['Memory'] = data['Memory'].astype(str).str[0]

data['Storage'] = data['Storage'].astype(str).str[0]

data.isnull().sum()

data.head()

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

cat\_enc\_le = le.fit\_transform(data['Color'])

data['Color'].unique()

np.unique(cat\_enc\_le)

le.inverse\_transform([0,1,2,3])

data['Brand'].unique()

from sklearn.preprocessing import OneHotEncoder

ohe = OneHotEncoder()

cat\_enc\_ohe = ohe.fit\_transform(data[['Brand']])

cat\_enc\_ohe

cat\_enc\_ohe.todense()[0:20]

#TargetEncoder

#!pip install category\_encoders

from category\_encoders.target\_encoder import TargetEncoder as ce\_TargetEncoder

ce\_TargetEncoder1 = ce\_TargetEncoder()

data\_MEAN\_ENC = ce\_TargetEncoder1.fit\_transform(data[data.columns.difference(['Original Price'])], data['Original Price'])

data\_MEAN\_ENC.head()

def check\_mean\_encoding(field):

for s in data[field].unique():

data\_filter = data[data[field]==s]

if data\_filter.shape[0] > 0:

prob = sum(data\_filter['Original Price']) / data\_filter.shape[0]

print(s, '-' , prob)

check\_mean\_encoding('Brand')

check\_mean\_encoding('Model')

check\_mean\_encoding('Color')

def sales\_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.hist(figsize=(20,20))

plt.show()

sales\_plots(data, 'Rating')

#Логарифмическое преобразование

data['Rating'] = np.log(data['Rating'])

sales\_plots(data, 'Rating')

#Обратное преобразование

data['Rating\_rec'] = 1 / (data['Rating'])

sales\_plots(data, 'Rating')

#Квадратный корень

data['Rating\_sqr'] = data['Rating']\*\*(1/2)

sales\_plots(data, 'Rating\_sqr')

#Возведение в степень

data['Rating\_exp1'] = data['Rating']\*\*(1/1.5)

sales\_plots(data, 'Rating\_exp1')

data['Rating\_exp2'] = data['Rating']\*\*(2)

sales\_plots(data, 'Rating\_exp2')

data['Rating\_exp3'] = data['Rating']\*\*(0.333)

sales\_plots(data, 'Rating\_exp3')

#Преобразование Йео-Джонсона

# Необходимо преобразовать данные к действительному типу

data['Rating'] = data['Rating'].astype('float')

data['Rating\_yeojohnson'], param = stats.yeojohnson(data['Rating'])

print('Оптимальное значение λ = {}'.format(param))

sales\_plots(data, 'Rating\_yeojohnson')

Экранные формы

  

       