Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №2

Вариант №15

Выполнил:
студент группы ИУ5-63
Миронова Александра
Подпись и дата:
28.05.22

Проверил:
Юрий Евгеньевич Гапанюк
Подпись и дата:

Москва, 2022 г.

Задание:

Для заданного набора данных провести обработку пропусков в данных для одного категориального и одного количественного признака. Указать использованные способы обработки пропусков в данных для категориальных и количественных признаков? Указать, какие признаки лучше использовать для дальнейшего построения моделей машинного обучения и почему?

Результат:

Лабораторная работа №2

Задание:

1) Выбрать набор данных (датасет), содержащий категориальные 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.)

2) Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:

<

ul>

- обработку пропусков в данных;
- кодирование категориальных признаков;
- масштабирование данных.

Ход выполнения работы

1) Текстовое описание набора данных

В качестве набора данных я буду использовать набор данных о шоколадных батончиках и конфетах. https://www.kaggle.com/datasets/rtatman/chocolate-bar-ratings

Датасет состоит из файла: flavors_of_cacao.csv

Файл содержит следующие колонки:

Name - название книги Author - Автор книги User rating - рейтинг книги Reviews - количество отзывов о книге Price - цена книги Year - год получения статуса бестселлер Genre - жанр

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")

data = pd.read_csv('flavors_of_cacao.csv', sep=",")
# Первые 5 строк датасета
data.head()
```

	Company_(Maker-if_known)	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percent	Company_Location	Rating	Bean_Type	Broad_Bean_Origin
0	A. Morin	Agua Grande	1876	2016.0	63%	France	3.75	NaN	Sao Tome
1	A. Morin	Kpime	1676	2015.0	70%	France	2.75	NaN	Togo
2	A. Morin	Atsane	1676	NaN	70%	France	3.00	NaN	Togo
3	A. Morin	Akata	1680	2015.0	70%	France	3.50	NaN	Togo
4	A. Morin	Quilla	1704	NaN	70%	France	3.50	NaN	Peru

```
# размер набора данных
data.shape
```

```
(1795, 9)
```

```
# типы колонок
data.dtypes
```

```
Company_(Maker-if_known)             object
Specific_Bean_Origin_or_Bar_Name     object
REF                                   int64
Review_Date                         float64
Cocoa_Percent                        object
Company_Location                     object
Rating                              float64
Bean_Type                            object
Broad_Bean_Origin                    object
dtype: object
```

```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
Company_(Maker-if_known) - 0
Specific_Bean_Origin_or_Bar_Name - 0
REF - 0
Review_Date - 53
Cocoa_Percent - 44
Company_Location - 62
Rating - 0
Bean_Type - 888
Broad_Bean_Origin - 1
```

Пропуски имеются в столбце числовых значений Review_Date и в столбцах категориальных данных Cocoa_Percent, Bean_Type, Broad_Bean_Origin

2) Обработка числовых значений.

Поскольку число пропущенных значений в столбце с оценками батончиков составляет всего 3 %, я воспользуюсь средствами импьютации библиотеки scikit-learn.

```
review_count_data=data[['Review_Date']]
review_count_data
```

	Review_Date
0	2016.0
1	2015.0
2	NaN
3	2015.0
4	NaN
...	...
1790	2011.0
1791	2011.0
1792	2011.0
1793	2011.0
1794	2010.0

1795 rows × 1 columns

```
np.unique(review_count_data)
```

```
array([2006., 2007., 2008., 2009., 2010., 2011., 2012., 2013., 2014.,
       2015., 2016., 2017.,   nan])
```

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
# Импьютация медианой
imputer1 = SimpleImputer(missing_values=np.nan, strategy='median')
full_review_count = imputer1.fit_transform(review_count_data)
full_review_count
```

```
array([[2016.],
       [2015.],
       [2013.],
       ...,
       [2011.],
       [2011.],
       [2010.]])
```

Убедимся, что пустые значения отсутствуют

```
np.unique(full_review_count)
```

```
array([2006., 2007., 2008., 2009., 2010., 2011., 2012., 2013., 2014.,
       2015., 2016., 2017.])
```

Теперь заменим в data столбец Review_Date новым стольбцом без пропусков.

```
data['Review_Date'] = full_review_count.reshape(-1)
data.head()
```

	Company_(Maker-if_known)	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percent	Company_Location	Rating	Bean_Type	Broad_Bean_Origin
0	A. Morin	Agua Grande	1876	2016.0	63%	France	3.75	NaN	Sao Tome
1	A. Morin	Kpime	1676	2015.0	70%	France	2.75	NaN	Togo
2	A. Morin	Atsane	1676	2013.0	70%	France	3.00	NaN	Togo
3	A. Morin	Akata	1680	2015.0	70%	France	3.50	NaN	Togo
4	A. Morin	Quilla	1704	2013.0	70%	France	3.50	NaN	Peru

3) Обработка категориальных значений.

В качестве категориальных данных был выбран столбец Bean_Type. Поскольку данный признак имеет значение лишь для производителей и некоторых пользователей, разбирающихся в стортах какао, он не влияет на оценку батончика. Количество пропущенных значений составляет 50% от всех значений. На основании этих данных я решила, что заполнять пропуски не имеет смысла, лучше просто удалить данный столбец из датасета.

```
data.drop(['Bean_Type'], inplace=True, axis=1)
data.head()
```

	Company_(Maker-if_known)	Specific_Bean_Origin_or_Bar_Name	REF	Review_Date	Cocoa_Percent	Company_Location	Rating	Broad_Bean_Origin
0	A. Morin	Agua Grande	1876	2016.0	63%	France	3.75	Sao Tome
1	A. Morin	Kpime	1676	2015.0	70%	France	2.75	Togo
2	A. Morin	Atsane	1676	2013.0	70%	France	3.00	Togo
3	A. Morin	Akata	1680	2015.0	70%	France	3.50	Togo
4	A. Morin	Quilla	1704	2013.0	70%	France	3.50	Peru

```
```