Московский государственный технический университет имени Н. Э. Баумана

«Методы машинного обучения »

Отчет по

Лабораторной работе №3

Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.

Выполнил:
студент
Егоров С. А. ИУ5-22М
Проверил:
доцент, к.т.н.
Гапанюк Ю.Е.

Москва, 2020

Цель лабораторной работы

Изучение способов предварительной обработки данных для дальнейшего формирования моделей.

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:

   - обработку пропусков в данных (не менее 3 признаков);
   - кодирование категориальных признаков (не менее 3 признаков);
   - масштабирование данных (не менее 3 признаков).

Реализация задания

```
# Выберем  колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
total_count = data.shape[0]
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col, dt, temp_null_count, temp_perc))
```

```
Колонка director. Тип данных object. Количество пустых значений 1969, 31.58%.
Колонка cast. Тип данных object. Количество пустых значений 570, 9.14%.
Колонка country. Тип данных object. Количество пустых значений 476, 7.64%.
Колонка date_added. Тип данных object. Количество пустых значений 11, 0.18%.
Колонка rating. Тип данных object. Количество пустых значений 10, 0.16%.
```

Часть 1. Обработка пропусков в данных

```
cat_temp_data_reting = data_num[['rating']]
cat_temp_data_reting['rating'].unique()
```

```
array(['TV-PG', 'TV-MA', 'TV-Y7-FV', 'TV-Y7', 'TV-14', 'R', 'TV-Y', 'NR',
       'PG-13', 'TV-G', 'PG', 'G', nan, 'UR', 'NC-17'], dtype=object)
```

```
# Импьютация наиболее частыми значениями
imp1 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp1 = imp1.fit_transform(cat_temp_data_reting)
data_imp1
```

```
array([['TV-PG'],
       ['TV-MA'],
       ['TV-Y7-FV'],
       ...,
       ['TV-MA'],
       ['TV-MA'],
       ['TV-14']], dtype=object)
```

```
# Пустые значения отсутствуют
np.unique(data_imp1)
```

```
array(['G', 'NC-17', 'NR', 'PG', 'PG-13', 'R', 'TV-14', 'TV-G', 'TV-MA',
       'TV-PG', 'TV-Y', 'TV-Y7', 'TV-Y7-FV', 'UR'], dtype=object)
```

Часть 2. Кодирование категориальных признаков

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

```
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['c1'])
cat_enc['c1'].unique()
```

```
array(['TV-PG', 'TV-MA', 'TV-Y7-FV', 'TV-Y7', 'TV-14', 'R', 'TV-Y', 'NR',
       'PG-13', 'TV-G', 'PG', 'G', 'UR', 'NC-17'], dtype=object)
```

```
np.unique(cat_enc_le)
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13])
```

```
le.inverse_transform([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13])
```

```
array(['G', 'NC-17', 'NR', 'PG', 'PG-13', 'R', 'TV-14', 'TV-G', 'TV-MA',
       'TV-PG', 'TV-Y', 'TV-Y7', 'TV-Y7-FV', 'UR'], dtype=object)
```

Часть 3. Масштабирование данных

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['release_year']])
plt.hist(sc2_data, 50)
plt.show()
```