Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Рубежный контроль №1
по дисциплине
«Методы машинного обучения»
на тему
«Обработка пропусков»

Выполнил:
студент группы РТ5-61Б
Корякин Д.

Москва — 2020 г.

0.1. РК - 1

Корякин Д.А. РТ5-61Б.

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

dataset - https://www.kaggle.com/san-francisco/sf-restaurant-scores-lives-standard

[21]:
```
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
```

[22]:
```
data = pd.read_csv("data/restaurant.csv")
```

[23]:
```
data.dtypes
```

[23]:
```
business_id                 int64
business_name              object
business_address           object
business_city              object
business_state             object
business_postal_code       object
business_latitude         float64
business_longitude        float64
business_location          object
business_phone_number     float64
inspection_id              object
inspection_date            object
inspection_score          float64
inspection_type            object
violation_id               object
violation_description      object
risk_category              object
dtype: object
```

[24]:
```
data.head()
```

[24]:
```
   business_id            business_name  business_address ␣
↪business_city  \
0        69618   Fancy Wheatfield Bakery  1362 Stockton St   San␣
↪Francisco
1        97975               BREADBELLY   1408 Clement St   San␣
↪Francisco
2        69487    Hakkasan San Francisco       1 Kearny St   San␣
↪Francisco
3        91044     Chopsticks Restaurant   4615 Mission St   San␣
↪Francisco
4        85987                   Tselogs      552 Jones St   San␣
↪Francisco
```

```
  business_state business_postal_code  business_latitude  
↪business_longitude  \
0             CA                94133                NaN          
↪       NaN
1             CA                94118                NaN          
↪       NaN
2             CA                94108                NaN          
↪       NaN
3             CA                94112                NaN          
↪       NaN
4             CA                94102                NaN          
↪       NaN

  business_location  business_phone_number   inspection_id  \
0               NaN                    NaN  69618_20190304
1               NaN           1.415724e+10  97975_20190725
2               NaN                    NaN  69487_20180418
3               NaN                    NaN  91044_20170818
4               NaN                    NaN  85987_20180412

           inspection_date  inspection_score           
↪inspection_type  \
0  2019-03-04T00:00:00.000               NaN              
↪Complaint
1  2019-07-25T00:00:00.000              96.0      Routine -
↪Unscheduled
2  2018-04-18T00:00:00.000              88.0      Routine -
↪Unscheduled
3  2017-08-18T00:00:00.000               NaN  Non-inspection site
↪visit
4  2018-04-12T00:00:00.000              94.0      Routine -
↪Unscheduled

            violation_id                              
↪violation_description  \
0  69618_20190304_103130           Inadequate sewage or
↪wastewater disposal
1  97975_20190725_103124  Inadequately cleaned or sanitized food
↪contact…
2  69487_20180418_103119  Inadequate and inaccessible handwashing
↪facili…
3                    NaN                                        
↪       NaN
4  85987_20180412_103132                              Improper
↪thawing methods

  risk_category
```

```
0    Moderate Risk
1    Moderate Risk
2    Moderate Risk
3              NaN
4    Moderate Risk
```

[25]:
```python
data.shape
```

[25]:
```
(53973, 17)
```

[26]:
```python
data.isnull().sum()
```

[26]:
```
business_id                 0
business_name               0
business_address            0
business_city               0
business_state              0
business_postal_code     1083
business_latitude       24095
business_longitude      24095
business_location       24095
business_phone_number   36539
inspection_id               0
inspection_date             0
inspection_score        14114
inspection_type             0
violation_id            13462
violation_description   13462
risk_category           13462
dtype: int64
```

[27]:
```python
d = data[["business_name", "inspection_score", "risk_category"]]
d = d.dropna(axis=0, how="any")
d.shape
```

[27]:
```
(37778, 3)
```

0.1.1. Преобразование категориальных признаков

Label encoding

[28]:
```python
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

[29]:
```python
le = LabelEncoder()
risk_le = le.fit_transform(d["risk_category"])
```

[30]:
```python
np.unique(risk_le)
```

[30]:
```
array([0, 1, 2])
```

[31]:
```python
le.inverse_transform(np.unique(risk_le))
```

[31]: array(['High Risk', 'Low Risk', 'Moderate Risk'], dtype=object)

[32]:
```
d["risk_category_index"] = risk_le
```

One Hot Encoding

[33]:
```
ohe = OneHotEncoder()
risk_ohe = ohe.fit_transform(d[["risk_category"]])
```

[34]:
```
risk_ohe.todense()[0:10]
```

[34]:
```
matrix([[0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 1., 0.],
        [0., 1., 0.],
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 1., 0.],
        [0., 1., 0.],
        [0., 0., 1.]])
```

[35]:
```
d["risk_category"].head(10)
```

[35]:
```
1     Moderate Risk
2     Moderate Risk
4     Moderate Risk
8          Low Risk
9          Low Risk
18    Moderate Risk
20    Moderate Risk
24         Low Risk
28         Low Risk
33    Moderate Risk
Name: risk_category, dtype: object
```

[36]:
```
ohe_names = ohe.get_feature_names()
ohe_names
```

[36]:
```
array(['x0_High Risk', 'x0_Low Risk', 'x0_Moderate Risk'],
 ↪dtype=object)
```

[37]:
```
for idx, name in enumerate(ohe_names):
    d[name] = risk_ohe[:, idx].todense()
```

Получившийся набор данных

[38]:
```
d.head(10)
```

[38]:
```
                  business_name  inspection_score  risk_category \
1                    BREADBELLY              96.0  Moderate Risk
```

```
2         Hakkasan San Francisco              88.0  Moderate Risk
4                        Tselogs              94.0  Moderate Risk
8        The Estate Kitchen, LLC              86.0       Low Risk
9                    Beloved Cafe             96.0       Low Risk
18                   Ahipoki Bowl             94.0  Moderate Risk
20             Kasa Indian Eatery             96.0  Moderate Risk
24              Burger King #6414             90.0       Low Risk
28             Kabob Trolley, LLC             72.0       Low Risk
33   HILLCREST ELEMENTARY SCHOOL              88.0  Moderate Risk

    risk_category_index  x0_High Risk  x0_Low Risk  x0_Moderate
  ↪Risk
1                     2           0.0          0.0
  ↪1.0
2                     2           0.0          0.0
  ↪1.0
4                     2           0.0          0.0
  ↪1.0
8                     1           0.0          1.0
  ↪0.0
9                     1           0.0          1.0
  ↪0.0
18                    2           0.0          0.0
  ↪1.0
20                    2           0.0          0.0
  ↪1.0
24                    1           0.0          1.0
  ↪0.0
28                    1           0.0          1.0
  ↪0.0
33                    2           0.0          0.0
  ↪1.0
```

0.1.2. Вывод

Нулевые строки признака `inspection_score` были удалены, Категориальный признак был закодирован с помощью `OneHotEncoder` и `LabelEncoder`. Оба признака можно использовать при дальнейшем построении модели.

```
[ ]:
```