Лабораторная работа №2 по курсу "Технологии машинного обучения"

Выполнила Попова Дарья, студентка группы РТ5-61Б

Кодирование категориальных признаков

In [1]:

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

In [2]:

```
import numpy as np
import pandas as pd
chocolate = pd.read_csv('C:\\Users\\Дасупс\\Downloads\\flavors_of_cacao.csv')
```

In [3]:

```
chocolate.head()
```

Out[3]:

	Company \n(Maker-if known)	Specific Bean Origin\nor Bar Name	REF	Review\nDate	Cocoa\nPercent	Company\nLocation	Rating	Bean\nType	Broad Bean\nOrigin
0	A. Morin	Agua Grande	1876	2016	63%	France	3.75		Sao Tome
1	A. Morin	Kpime	1676	2015	70%	France	2.75		Togo
2	A. Morin	Atsane	1676	2015	70%	France	3.00		Togo
3	A. Morin	Akata	1680	2015	70%	France	3.50		Togo
4	A. Morin	Quilla	1704	2015	70%	France	3.50		Peru

Переименуем для начала колонки и избавимся от пробелов и \n.

In [4]:

```
chocolate = chocolate.rename(columns={'Company \n(Maker-if known)':'company',
                                      'Specific Bean Origin\nor Bar Name':'specific_bean_origin',
                                      'Review\nDate':'review_date',
                                      'Cocoa\nPercent':'cocoa_percentage',
                                      'Company\nLocation':'company_location',
                                      'Bean\nType':'bean_type',
                                      'Broad Bean\nOrigin':'bean_origin'})
```

In [5]:

```
chocolate.tail()
```

Out[5]:

	Company \n(Maker-if known)	specific_bean_origin	REF	review_date	cocoa_percentage	company_location	Rating	bean_type	bean_origin
1790	Zotter	Peru	647	2011	70%	Austria	3.75		Peru
1791	Zotter	Congo	749	2011	65%	Austria	3.00	Forastero	Congo
1792	Zotter	Kerala State	749	2011	65%	Austria	3.50	Forastero	India
1793	Zotter	Kerala State	781	2011	62%	Austria	3.25		India
1794	Zotter	Brazil, Mitzi Blue	486	2010	65%	Austria	3.00		Brazil

Немного не понимаю, почему первая колонка не переименовалась...

Посмотрим, какие у нас есть категориальные признаки и сколько в них уникальных значений.

In [6]:

```
chocolate.dtypes
```

Out[6]:

```
Company \n(Maker-if known)     object
specific_bean_origin           object
REF                             int64
review_date                     int64
cocoa_percentage               object
company_location               object
Rating                        float64
bean_type                      object
bean_origin                    object
dtype: object
```

In [7]:
```
chocolate.company_location.nunique()
```
Out[7]:
```
60
```
In [8]:
```
chocolate.specific_bean_origin.nunique()
```
Out[8]:
```
1039
```
In [9]:
```
chocolate.bean_type.nunique()
```
Out[9]:
```
41
```
In [10]:
```
chocolate.isnull().any()
```
Out[10]:
```
Company \n(Maker-if known)    False
specific_bean_origin          False
REF                           False
review_date                   False
cocoa_percentage              False
company_location              False
Rating                        False
bean_type                      True
bean_origin                    True
dtype: bool
```
In [11]:
```
chocolate = chocolate.dropna(axis=0, how='any')
```
In [12]:
```
chocolate.isnull().any()
```
Out[12]:
```
Company \n(Maker-if known)    False
specific_bean_origin          False
REF                           False
review_date                   False
cocoa_percentage              False
company_location              False
Rating                        False
bean_type                     False
bean_origin                   False
dtype: bool
```
LabelEncoder

Для колонки с локацией компании воспользуемся кодированием целочисленными значениями.

In [13]:
```
lbl_enc = LabelEncoder()
```
In [14]:
```
comp_loc_encryption = lbl_enc.fit_transform(chocolate.company_location)
```
In [15]:
```
inverse_array_comp_loc = np.unique(comp_loc_encryption)
```
In [16]:
```
inverse_array_comp_loc
```
Out[16]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55, 56, 57, 58, 59])
```
Теперь проверим, что скрывается за этим массивом и точно ли LabelEncoder всё верно нашаманил.

In [17]:
```
lbl_enc.inverse_transform(inverse_array_comp_loc)
```

Out[17]:

```
array(['Amsterdam', 'Argentina', 'Australia', 'Austria', 'Belgium',
       'Bolivia', 'Brazil', 'Canada', 'Chile', 'Colombia', 'Costa Rica',
       'Czech Republic', 'Denmark', 'Domincan Republic', 'Ecuador',
       'Eucador', 'Fiji', 'Finland', 'France', 'Germany', 'Ghana',
       'Grenada', 'Guatemala', 'Honduras', 'Hungary', 'Iceland', 'India',
       'Ireland', 'Israel', 'Italy', 'Japan', 'Lithuania', 'Madagascar',
       'Martinique', 'Mexico', 'Netherlands', 'New Zealand', 'Niacragua',
       'Nicaragua', 'Peru', 'Philippines', 'Poland', 'Portugal',
       'Puerto Rico', 'Russia', 'Sao Tome', 'Scotland', 'Singapore',
       'South Africa', 'South Korea', 'Spain', 'St. Lucia', 'Suriname',
       'Sweden', 'Switzerland', 'U.K.', 'U.S.A.', 'Venezuela', 'Vietnam',
       'Wales'], dtype=object)
```

One-Hot Encoding

Колонке с происхождением сырья для шоколада повезло намного меньше: на ней мы будем испытывать кодирование наборами бинарных значений...

In [18]:

```
ohe = OneHotEncoder()
specific_bean_origin_ohe_enc = ohe.fit_transform(chocolate[['specific_bean_origin']])
```

In [19]:

```
specific_bean_origin_ohe_enc.shape
```

Out[19]:

```
(1793, 1038)
```

Уже видим что-то невообразимо страшное во втором элементе кортежа с количеством столбцов...

In [20]:

```
specific_bean_origin_ohe_enc
```

Out[20]:

```
<1793x1038 sparse matrix of type '<class 'numpy.float64'>'
 with 1793 stored elements in Compressed Sparse Row format>
```

То же самое, только с помощью встроенное в Pandas функции get_dummies.

In [21]:

```
pd.get_dummies(chocolate[['specific_bean_origin']]).head()
```

Out[21]:

	specific_bean_origin_"heirloom", Arriba Nacional	specific_bean_origin_100 percent	specific_bean_origin_2009 Hapa Nibby	specific_bean_origin_A case of the Xerces Blues, triple roast	specific_bean_origin_ABOCFA Coop	specific_
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

5 rows × 1038 columns

Кошмар...