Лабораторная работа 2

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

Цель лабораторной работы: изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Задание: 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.) Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи: обработку пропусков в данных; кодирование категориальных признаков; масштабирование данных.

Ввод [1]:

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

Ввод [2]:

```
data = pd.read_csv('../datasets/merged_data_cleaned.csv', sep=",")
```

Ввод [3]:

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

Out[3]:

```
(1339, 44)
```

Ввод [4]:

```
# типы колонок
data.dtypes
```

Out[4]:

```
Unnamed: 0                 int64
Species                   object
Owner                     object
Country.of.Origin         object
Farm.Name                 object
Lot.Number                object
Mill                      object
ICO.Number                object
Company                   object
Altitude                  object
Region                    object
Producer                  object
Number.of.Bags             int64
Bag.Weight                object
In.Country.Partner        object
Harvest.Year              object
Grading.Date              object
Owner.1                   object
Variety                   object
Processing.Method         object
Aroma                    float64
Flavor                   float64
Aftertaste               float64
Acidity                  float64
Body                     float64
Balance                  float64
Uniformity               float64
Clean.Cup                float64
Sweetness                float64
Cupper.Points            float64
Total.Cup.Points         float64
Moisture                 float64
Category.One.Defects       int64
Quakers                  float64
Color                     object
Category.Two.Defects       int64
Expiration                object
Certification.Body        object
Certification.Address     object
Certification.Contact     object
unit_of_measurement       object
altitude_low_meters      float64
altitude_high_meters     float64
altitude_mean_meters     float64
dtype: object
```

Ввод [5]:

```
# проверим есть ли пропущенные значения
data.isnull().sum()
```

Out[5]:

```
Unnamed: 0                  0
Species                     0
Owner                       7
Country.of.Origin           1
Farm.Name                 359
Lot.Number               1063
Mill                      318
ICO.Number                157
Company                   209
Altitude                  226
Region                     59
Producer                  232
Number.of.Bags              0
Bag.Weight                  0
In.Country.Partner          0
Harvest.Year               47
Grading.Date                0
Owner.1                     7
Variety                   226
Processing.Method         170
Aroma                       0
Flavor                      0
Aftertaste                  0
Acidity                     0
Body                        0
Balance                     0
Uniformity                  0
Clean.Cup                   0
Sweetness                   0
Cupper.Points               0
Total.Cup.Points            0
Moisture                    0
Category.One.Defects        0
Quakers                     1
Color                     218
Category.Two.Defects        0
Expiration                  0
Certification.Body          0
Certification.Address       0
Certification.Contact       0
unit_of_measurement         0
altitude_low_meters       230
altitude_high_meters      230
altitude_mean_meters      230
dtype: int64
```

Ввод [6]:

```
# Первые 5 строк датасета
data.head()
```

Out[6]:

	Unnamed: 0	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mill	ICO.N
0	0	Arabica	metad plc	Ethiopia	metad plc	NaN	metad plc	201
1	1	Arabica	metad plc	Ethiopia	metad plc	NaN	metad plc	201
2	2	Arabica	grounds for health admin	Guatemala	san marcos barrancas "san cristobal cuch	NaN	NaN	
3	3	Arabica	yidnekachew dabessa	Ethiopia	yidnekachew dabessa coffee plantation	NaN	wolensu	
4	4	Arabica	metad plc	Ethiopia	metad plc	NaN	metad plc	201

5 rows × 44 columns

Ввод [7]:

```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 1339
```

Обработка пропусков в данных

Простые стратегии - удаление или заполнение нулями

Ввод [8]:

```
# Удаление колонок, содержащих пустые значения
data_new_1 = data.dropna(axis=1, how='any')
(data.shape, data_new_1.shape)
```

Out[8]:

```
((1339, 44), (1339, 25))
```

Ввод [11]:

```
data_new_1.head()
```

Out[11]:

	Unnamed: 0	Species	Number.of.Bags	Bag.Weight	In.Country.Partner	Grading.Date	Aroma	Fl
0	0	Arabica	300	60 kg	METAD Agricultural Development plc	April 4th, 2015	8.67	
1	1	Arabica	300	60 kg	METAD Agricultural Development plc	April 4th, 2015	8.75	
2	2	Arabica	5	1	Specialty Coffee Association	May 31st, 2010	8.42	
3	3	Arabica	320	60 kg	METAD Agricultural Development plc	March 26th, 2015	8.17	
4	4	Arabica	300	60 kg	METAD Agricultural Development plc	April 4th, 2015	8.25	

5 rows × 25 columns

Ввод [9]:

```
# Удаление строк, содержащих пустые значения
data_new_2 = data.dropna(axis=0, how='any')
(data.shape, data_new_2.shape)
```

Out[9]:

```
((1339, 44), (132, 44))
```

Ввод [10]:

```
data_new_2.head()
```

Out[10]:

	Unnamed: 0	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mill
29	29	Arabica	lin, che-hao krude 林哲豪	Taiwan	tsoustructive garden 鄒築園	Tsoustructive 2015 Sumatra Typica	tsoustructive garden 鄒築園
115	115	Arabica	lin, che-hao krude 林哲豪	Taiwan	shi fang yuan 十方源	2016 Tainan Coffee Cupping Event Micro Lot 臺南市...	shi fang yuan 十方源
125	125	Arabica	consejo salvadoreño del café	El Salvador	monterrey	1-71	j.j. borja nathan
128	128	Arabica	rodrigo soto	Costa Rica	rio jorco	Tarrazu	rio jorco
129	129	Arabica	juan luis alvarado romero	Guatemala	san diego buena vista	11/08/0109	san diego buena vista

5 rows × 44 columns

"Внедрение значений" - импьютация (imputation)

Ввод [12]:

```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
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=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col,
```

```
Колонка Quakers. Тип данных float64. Количество пустых значений 1, 0.07%.
Колонка altitude_low_meters. Тип данных float64. Количество пустых значений 2
30, 17.18%.
Колонка altitude_high_meters. Тип данных float64. Количество пустых значений
230, 17.18%.
Колонка altitude_mean_meters. Тип данных float64. Количество пустых значений
230, 17.18%.
```

Ввод [13]:

```
# Фильтр по колонкам с пропущенными значениями
data_num = data[num_cols]
data_num
```

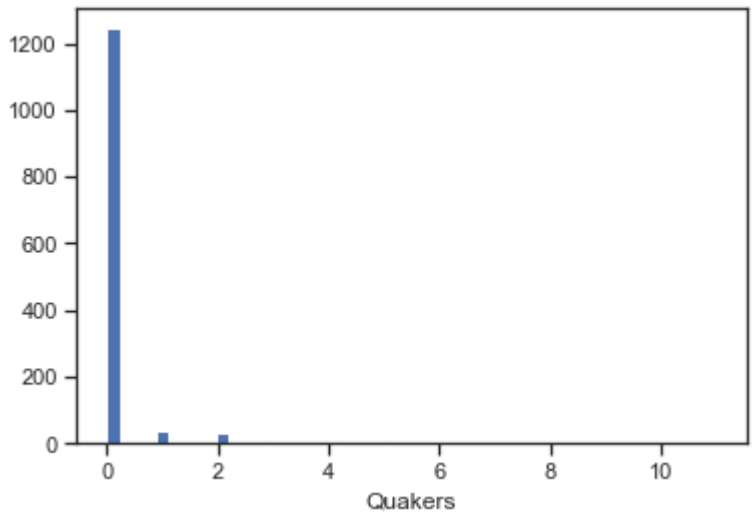
Out[13]:

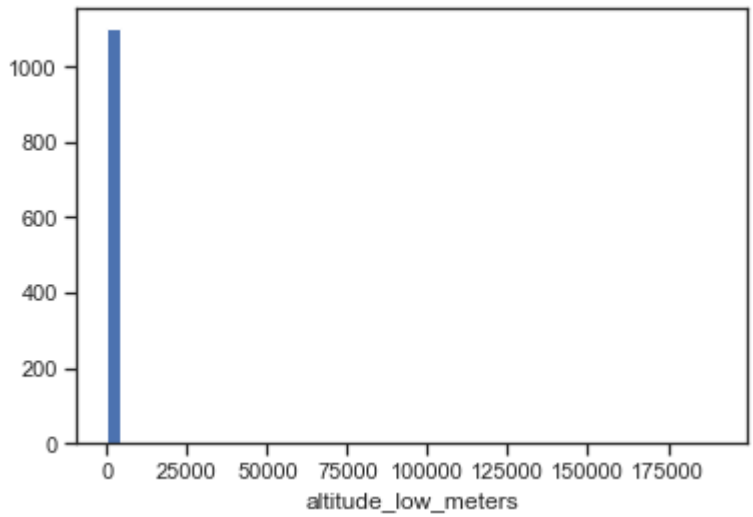
	Quakers	altitude_low_meters	altitude_high_meters	altitude_mean_meters
0	0.0	1950.0	2200.0	2075.0
1	0.0	1950.0	2200.0	2075.0
2	0.0	1600.0	1800.0	1700.0
3	0.0	1800.0	2200.0	2000.0
4	0.0	1950.0	2200.0	2075.0
...	...	...	...	...
1334	0.0	NaN	NaN	NaN
1335	0.0	40.0	40.0	40.0
1336	0.0	795.0	795.0	795.0
1337	0.0	NaN	NaN	NaN
1338	0.0	NaN	NaN	NaN

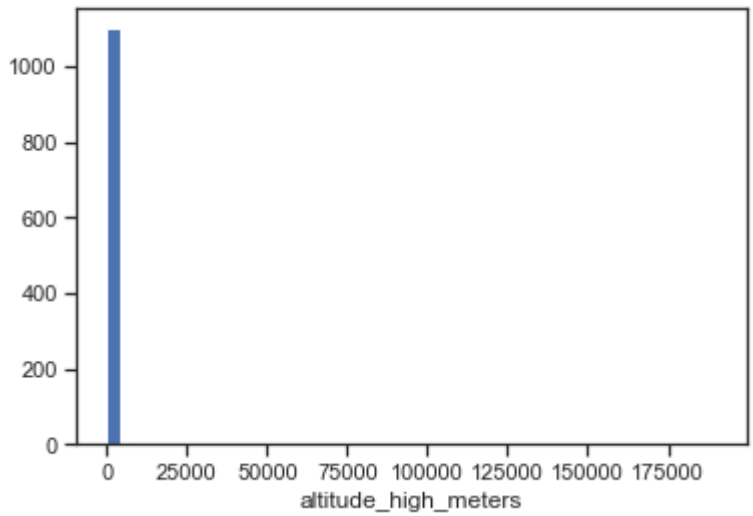
1339 rows × 4 columns

Ввод [17]:

```
# Гистограмма по признакам
for col in data_num:
    plt.hist(data[col], 50)
    plt.xlabel(col)
    plt.show()
```







Ввод [18]:

```
data_num_Quakers = data_num[['Quakers']]
data_num_Quakers.head()
```

Out[18]:

	Quakers
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0

Ввод [19]:

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

Ввод [20]:

```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_Quakers)
mask_missing_values_only
```

Out[20]:

```
array([[False],
       [False],
       [False],
       ...,
       [False],
       [False],
       [False]])
```

Ввод [23]:

```
strategies=['mean', 'median', 'most_frequent']
```

Ввод [24]:

```
def test_num_impute(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_num_Quakers)
    return data_num_imp[mask_missing_values_only]
```

Ввод [25]:

```
strategies[0], test_num_impute(strategies[0])
```

Out[25]:

```
('mean', array([0.17339312]))
```

Ввод [26]:

```
# Более сложная функция, которая позволяет задавать колонку и вид импьютации
def test_num_impute_col(dataset, column, strategy_param):
    temp_data = dataset[[column]]

    indicator = MissingIndicator()
    mask_missing_values_only = indicator.fit_transform(temp_data)

    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(temp_data)

    filled_data = data_num_imp[mask_missing_values_only]

    return column, strategy_param, filled_data.size, filled_data[0], filled_data[fil
```

Ввод [27]:

```
data[['altitude_low_meters']].describe()
```

Out[27]:

	altitude_low_meters
count	1109.000000
mean	1750.713315
std	8669.440545
min	1.000000
25%	1100.000000
50%	1310.640000
75%	1600.000000
max	190164.000000

Ввод [28]:

```
test_num_impute_col(data, 'altitude_low_meters', strategies[0])
```

Out[28]:

```
('altitude_low_meters', 'mean', 230, 1750.7133150586112, 1750.71331505
86112)
```

Обработка пропусков в категориальных данных

Ввод [39]:

```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
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'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col,
```

```
Колонка Owner. Тип данных object. Количество пустых значений 7, 0.52%.
Колонка Country.of.Origin. Тип данных object. Количество пустых значений 1,
0.07%.
Колонка Farm.Name. Тип данных object. Количество пустых значений 359, 26.81%.
Колонка Lot.Number. Тип данных object. Количество пустых значений 1063, 79.3
9%.
Колонка Mill. Тип данных object. Количество пустых значений 318, 23.75%.
Колонка ICO.Number. Тип данных object. Количество пустых значений 157, 11.7
3%.
Колонка Company. Тип данных object. Количество пустых значений 209, 15.61%.
Колонка Altitude. Тип данных object. Количество пустых значений 226, 16.88%.
Колонка Region. Тип данных object. Количество пустых значений 59, 4.41%.
Колонка Producer. Тип данных object. Количество пустых значений 232, 17.33%.
Колонка Harvest.Year. Тип данных object. Количество пустых значений 47, 3.5
1%.
Колонка Owner.1. Тип данных object. Количество пустых значений 7, 0.52%.
Колонка Variety. Тип данных object. Количество пустых значений 226, 16.88%.
Колонка Processing.Method. Тип данных object. Количество пустых значений 170,
12.7%.
Колонка Color. Тип данных object. Количество пустых значений 218, 16.28%.
```

Ввод [44]:

```
cat_temp_data = data[['Farm.Name']]
cat_temp_data.head()
```

Out[44]:

	Farm.Name
0	metad plc
1	metad plc
2	san marcos barrancas "san cristobal cuch
3	yidnekachew dabessa coffee plantation
4	metad plc

Ввод [45]:

```
cat_temp_data['Farm.Name'].unique()
```

```
       'llano hermoso', 'hani', 'finca la fortuna', 'el regadito',
       'dongshan gaoyuan village chief manor coffee tainan, taiwan 台灣
台南東山高原村長莊園咖啡',
       'rancho los laureles', 'finca los andes',
       'producer group (approx. 1,000 farmers)', 'agua de la maripos
a',
       '1', 'zaragoza, montelibano, pamal navil', 'peña campana',
       'la cruz', 'el limón', 'la orduña', 'sierra madre', 'las ceiba
s',
       'la marina-el orizabeño', 'rancho vigia', '2000 farmers',
       'la vuelta', 'la morena', 'el centenario', '200 farms',
       'kyangundu cooperative society', 'sethuraman estate kaapi royal
e',
       'sethuraman estate', 'ugacof project area',
       'katikamu capca farmers association', 'sethuraman estates',
       'ishaka', 'kasozi coffee farmers', 'kyangundu coop society',
       'bushenyi', 'kigezi coffee farmers association',
       'mannya coffee project', 'robustasa', 'fazenda cazengo'],
      dtype=object)
```

Ввод [46]:

```
cat_temp_data[cat_temp_data['Farm.Name'].isnull()].shape
```

Out[46]:

```
(359, 1)
```

Ввод [47]:

```
# Импьютация наиболее частыми значениями
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp2 = imp2.fit_transform(cat_temp_data)
data_imp2
```

Out[47]:

```
array([['metad plc'],
       ['metad plc'],
       ['san marcos barrancas "san cristobal cuch'],
       ...,
       ['fazenda cazengo'],
       ['various'],
       ['various']], dtype=object)
```

Ввод [48]:

```
# Пустые значения отсутствуют
np.unique(data_imp2)
```

```
ta',
       'santa laura', 'santa maria', 'santa maria temaxcalapa',
       'santa mariana', 'santa mariana/são vicente', 'santa martha',
       'santa matilde', 'santa rosa', 'santa teresa',
       'santo tomas pachuj', 'savan coffee bean', 'saveg coffee farm',
       'seid damtew coffee planation', 'selian coffee estate',
       'selva negra', 'sertao', 'sertao farm', 'sethuraman estate',
       'sethuraman estate kaapi royale', 'sethuraman estates', 'severa
l',
       'several farmers', 'several farms',
       'several, bukidnon, mindanao, philippines', 'shah plantation',
       'shangrilla estate', 'sheng he shang pin coffee 聖荷上品咖啡坊',
       'shi fang yuan 十方源', 'sierra madre', 'sierra nevada',
       'sipi organic coffee project', 'sithar coffee farm', 'sitio cla
ro',
       'sitío corrego da olaria/são caetano', 'sitío santa luzia',
       'sitío são geraldo', 'small holders farmer', 'sogestal kayanz
a',
       'song yue coffee 嵩岳咖啡', 'sran temanggung plantation',
       'su-zhen huang 黃素真', 'suma ikt itende'
```

Ввод [49]:

```
# Импьютация константой
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value='NA')
data_imp3 = imp3.fit_transform(cat_temp_data)
data_imp3
```

Out[49]:

```
array([['metad plc'],
       ['metad plc'],
       ['san marcos barrancas "san cristobal cuch'],
       ...,
       ['fazenda cazengo'],
       ['NA'],
       ['NA']], dtype=object)
```

Ввод [50]:

```
np.unique(data_imp3)
```

Out[50]:

```
array(['-', '1', '200 farms', '2000 farmers', '2000 farms', 'NA',
       'a shu she coffee 阿束社咖啡莊園', 'acacia hills', 'ada farm',
       'agropecuaria quiagral', 'agua caliente', 'agua de la maripos
a',
       "alicia's farm", 'alishan zou zhu yuan 阿里山鄒築園', 'amkeni',
'ampcg',
       'ano family', 'aolme', 'apollo co., ltd.', 'apollo estate',
       'aprocafi', 'arianna farms', 'aricha coop',
       'arroyo triste, arroyo triste, san jose vista hermosa',
       'asefa dukamo coffee plantation',
       'asociación aldea global jinotega', 'asoperc', 'bacofa',
       'bai he lin coffee 白鶴林咖啡莊園', 'baijiada coffee farm佰加達咖啡
莊園',
       'baishencun coffee farm百勝村咖啡莊園', 'baishengcun coffee 百勝村
咖啡莊園',
       'barranca de las flores', 'beneficio el torreon', 'bethel',
       'bi yun si shi yi 碧云四十一咖啡坊', 'blend', 'blue lake', 'bola d
e oro',
```

Ввод [51]:

```
data_imp3[data_imp3=='NA'].size
```

Out[51]:

```
359
```

Преобразование категориальных признаков в числовые

Ввод [52]:

```
cat_enc = pd.DataFrame({'c1':data_imp2.T[0]})
cat_enc
```

Out[52]:

	c1
0	metad plc
1	metad plc
2	san marcos barrancas "san cristobal cuch
3	yidnekachew dabessa coffee plantation
4	metad plc
...	...
1334	robustasa
1335	robustasa
1336	fazenda cazengo
1337	various
1338	various

1339 rows × 1 columns

Кодирование категорий целочисленными значениями (label encoding)

Использование LabelEncoder

Ввод [53]:

```
from sklearn.preprocessing import LabelEncoder
```

Ввод [54]:

```
cat_enc['c1'].unique()
```

```
       'el mirador', 'finca nueva linda', 'campo das flores',
       'são francisco da serra', 'fazenda do lobo', "olhos d'agua",
       'tega and tula special coffee farm',
       'kan tou mountain coffee 崁頭山咖啡館', 'alishan zou zhu yuan 阿里
山鄒築園',
       'santa maria', 'lintong nihuta', 'isule farmers group',
       'ibanda farmers group', 'el chile', 'fazenda santo antonio', '雅
慕伊',
       'kabeywa county', 'grupo medina (pequeños productores)',
       'sertao farm', 'vegas', 'finca la estancia', 'las lomas',
       'santa rosa', 'santa josefita', '佐佑品咖啡莊園',
       'kalugmanan agri development corp.',
       'several, bukidnon, mindanao, philippines', 'pinzoneno',
       'fiech (mixed producers)', 'fazenda pantano',
       'nyapea coffee farmers', 'el regalito', 'apollo co., ltd.',
       'el jabali', 'el barbaro', 'gicumbi', 'santa fé 2',
       'kampung keling, jumaraj, gurusinga, gongsol',
       'burka coffee estate', 'providencia', 'las merceditas',
       'las delicias', 'la esmeralda',
```

Ввод [55]:

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

Ввод [56]:

```
# Наименования категорий в соответствии с порядковыми номерами

# Свойство называется classes, потому что предполагается что мы решаем
# задачу классификации и каждое значение категории соответствует
# какому-либо классу целевого признака

le.classes_
```

Out[56]:

```
array(['-', '1', '200 farms', '2000 farmers', '2000 farms',
       'a shu she coffee 阿束社咖啡莊園', 'acacia hills', 'ada farm',
       'agropecuaria quiagral', 'agua caliente', 'agua de la maripos
a',
       "alicia's farm", 'alishan zou zhu yuan 阿里山鄒築園', 'amkeni',
'ampcg',
       'ano family', 'aolme', 'apollo co., ltd.', 'apollo estate',
       'aprocafi', 'arianna farms', 'aricha coop',
       'arroyo triste, arroyo triste, san jose vista hermosa',
       'asefa dukamo coffee plantation',
       'asociación aldea global jinotega', 'asoperc', 'bacofa',
       'bai he lin coffee 白鶴林咖啡莊園', 'baijiada coffee farm佰加達咖啡
莊園',
       'baishencun coffee farm百勝村咖啡莊園', 'baishengcun coffee 百勝村
咖啡莊園',
       'barranca de las flores', 'beneficio el torreon', 'bethel',
       'bi yun si shi yi 碧云四十一咖啡坊', 'blend', 'blue lake', 'bola d
e oro',
```

Ввод [57]:

```
cat_enc_le
```

Out[57]:

```
array([380, 380, 448, ..., 160, 528, 528])
```

Ввод [58]:

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

Out[58]:

```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  1
2,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  2
5,
        26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  3
8,
        39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  5
1,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  6
4,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  7
7,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  9
0,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 10
3,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 11
6,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 12
9,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 14
2,
       143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 15
5,
       156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 16
8,
       169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 18
1,
       182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 19
4,
       195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 20
7,
       208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 22
0,
       221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 23
3,
       234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 24
6,
       247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 25
9,
       260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 27
2,
       273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 28
5,
       286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 29
8,
       299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 31
1,
       312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 32
4,
       325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 33
7,
       338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 35
0,
       351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 36
```

```
3,
       364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 37
6,
       377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 38
9,
       390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 40
2,
       403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 41
5,
       416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 42
8,
       429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 44
1,
       442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 45
4,
       455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 46
7,
       468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 48
0,
       481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 49
3,
       494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 50
6,
       507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 51
9,
       520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 53
2,
       533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 54
5,
       546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 55
8,
       559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570])
```

Кодирование категорий наборами бинарных значений

Ввод [59]:

```
from sklearn.preprocessing import OneHotEncoder
```

Ввод [60]:

```
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['c1']])
```

Ввод [61]:

```
cat_enc.shape
```

Out[61]:

```
(1339, 1)
```

Ввод [62]:

```
cat_enc_ohe.shape
```

Out[62]:

```
(1339, 571)
```

Ввод [63]:

```
cat_enc_ohe.todense()[0:10]
```

Out[63]:

```
matrix([[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

Ввод [64]:

```
cat_enc.head(10)
```

Out[64]:

	c1
0	metad plc
1	metad plc
2	san marcos barrancas "san cristobal cuch
3	yidnekachew dabessa coffee plantation
4	metad plc
5	various
6	various
7	aolme
8	aolme
9	tulla coffee farm

Ввод [65]:

```
pd.get_dummies(cat_enc).head()
```

Out[65]:

	c1_-	c1_1	c1_200 farms	c1_2000 farmers	c1_2000 farms	c1_ashu she coffee 阿束社咖啡莊園	c1_acacia hills	c1_ada farm	c1_agropecuaria quiagral	c1_agua caliente
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0

5 rows × 571 columns

Ввод [66]:

```
pd.get_dummies(cat_temp_data, dummy_na=True).head()
```

Out[66]:

	Farm.Name_-	Farm.Name_1	Farm.Name_200 farms	Farm.Name_2000 farmers	Farm.Name_2000 farms	Farm.Name shu s coffee 阿束咖啡莊
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 × 572 columns

Масштабирование данных

Ввод [67]:

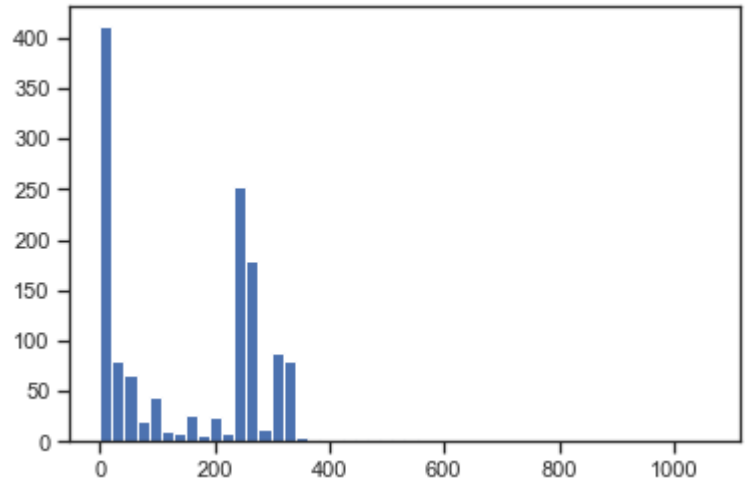
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

Ввод [68]:

```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['Number.of.Bags']])
```

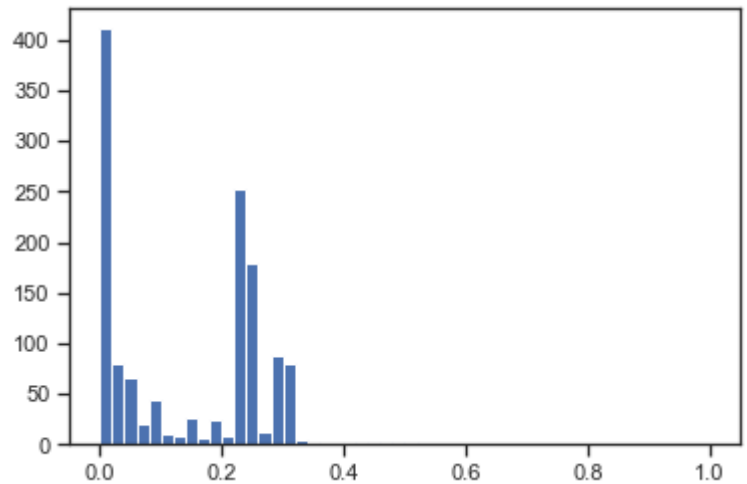
Ввод [70]:

```
plt.hist(data['Number.of.Bags'], 50)
plt.show()
```



Ввод [71]:

```
plt.hist(sc1_data, 50)
plt.show()
```



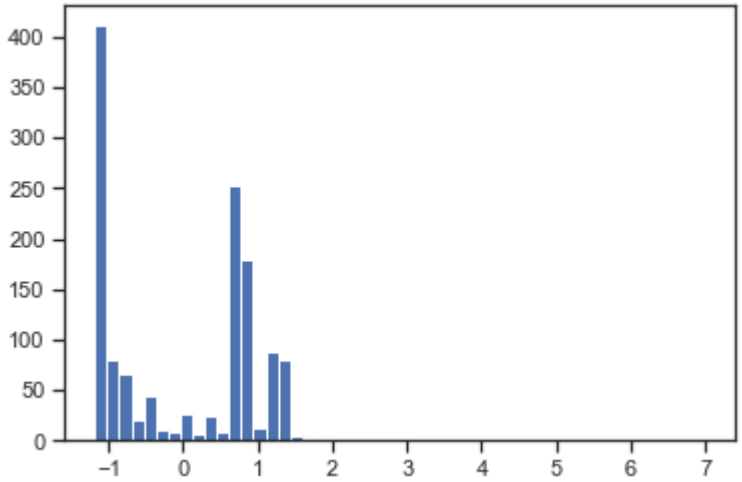
Масштабирование данных на основе Z-оценки - StandardScaler

Ввод [72]:

```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['Number.of.Bags']])
```

Ввод [73]:

```
plt.hist(sc2_data, 50)
plt.show()
```



Ввод [ ]:

```

```