Базанова А.Г. ИУ5-21М

Номер задачи №1 - 1

Номер задачи №2 - 21

Задача №1.

Для набора данных проведите кодирование одного (произвольного) категориального признака с использованием метода "count (frequency) encoding".

Задача №21.

Для набора данных проведите масштабирование данных для одного (произвольного) числового признака с использованием масштабирования по медиане.

Дополнительные требования:

Для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Загрузка и первичный анализ данных

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [ ]:
```
data = pd.read_csv('Studentdata.csv', sep=",")
```

In [ ]:
```
data.head(5)
```

Out[ ]:

	Unnamed: 0	2.Gender	3. Year of Study	4. Branch	1. Type of Schooling till 10th Grade:	2. Percentage in Class 10th?	3. Location of Class 10th School:	4. Type of Schooling for class 12th:	Percenta in Cl 12
0	0	Male	Third	CSE/IT	Normal Schooling	>=80% and <90%	Town	Normal Schooling	>=80% <9
1	1	Female	Second	CSIT	Schooling with Tuitions	>= 90%	City	Schooling with Tuitions	>=9
2	2	Male	Second	CSIT	Normal Schooling	>= 90%	City	Coaching Based Schools	>=9
3	3	Male	Second	CSIT	Normal Schooling	>=70% and <80%	City	Normal Schooling	>=80% <9
4	4	Male	Second	CSIT	Normal Schooling	>=70% and <80%	City	Normal Schooling	>=80% <9

5 rows × 25 columns

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 256 entries, 0 to 255
Data columns (total 25 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Unnamed: 0
256 non-null    int64
 1   2.Gender
256 non-null    object
 2   3. Year of Study
256 non-null    object
 3   4. Branch
256 non-null    object
 4   1. Type of Schooling till 10th Grade:
256 non-null    object
 5   2. Percentage in Class 10th?
256 non-null    object
 6   3. Location of Class 10th School:
256 non-null    object
 7   4. Type of Schooling for class 12th:
256 non-null    object
 8   5. Percentage in Class 12th?
256 non-null    object
 9   6. Location of Class 12th School:
256 non-null    object
 10  7. Dropped an year after School?
256 non-null    object
 11  8. Are you about to become first generation graduate ?
256 non-null    object
 12  9. Involved in additional volunteer groups?(NCC,NSS, Social Welfare Groups
or any other)  256 non-null    object
 13  1. How many programming languages do you know?
256 non-null    object
 14  2.  Actively involved in Specific technology?
201 non-null    object
 15  3. Exposure to Data structure and algorithms?
256 non-null    object
 16  4. Exposure to GitHub ?
256 non-null    object
 17  5. Are you associated with any developer Community?
256 non-null    object
 18  6. Active in developer Communities
216 non-null    object
 19  7. Participating in Hackathons and other competitions?
256 non-null    object
 20  8. Have you made any Software based projects?
256 non-null    object
 21  9. Have you made any Hardware based project (Arduino, Raspberry Pi, Robots
or any other)?  256 non-null    object
 22  10. Have you ever pitched any idea?
256 non-null    object
 23  11. Do you have any additional skills?
219 non-null    object
 24  Achievements:
147 non-null    object
dtypes: int64(1), object(24)
memory usage: 50.1+ KB
```

In [ ]:
```
data.describe()
```

Out[ ]:

	Unnamed: 0
count	256.000000
mean	127.500000
std	74.045031
min	0.000000
25%	63.750000
50%	127.500000
75%	191.250000
max	255.000000

In [ ]:
```
data.drop(columns=['Unnamed: 0'])
```

Out[ ]:

	2.Gender	3. Year of Study	4. Branch	1. Type of Schooling till 10th Grade:	2. Percentage in Class 10th?	3. Location of Class 10th School:	4. Type of Schooling for class 12th:	5. Percentage in Class 12th?	Loca of C . Sch
0	Male	Third	CSE/IT	Normal Schooling	>=80% and <90%	Town	Normal Schooling	>=80% and <90%	T
1	Female	Second	CSIT	Schooling with Tuitions	>= 90%	City	Schooling with Tuitions	>=90%	
2	Male	Second	CSIT	Normal Schooling	>= 90%	City	Coaching Based Schools	>=90%	
3	Male	Second	CSIT	Normal Schooling	>=70% and <80%	City	Normal Schooling	>=80% and <90%	
4	Male	Second	CSIT	Normal Schooling	>=70% and <80%	City	Normal Schooling	>=80% and <90%	
...	...	...	...	...	...	...	...	...	
251	Female	Third	CSE	Schooling with Tuitions	>= 90%	Town	Schooling with Tuitions	>=90%	T
252	Male	Second	ECE	Normal Schooling	>=80% and <90%	City	Normal Schooling	>=70% and <80%	
253	Male	Second	CSIT	Normal Schooling	>= 90%	City	Normal Schooling	>=90%	
254	Male	Second	ECE	Normal Schooling	>=80% and <90%	City	Normal Schooling	>=80% and <90%	
255	Male	Second	MCA	Normal Schooling	>=80% and <90%	City	Normal Schooling	>=60% and <70%	

256 rows × 25 columns

Задача №1.

Для набора данных проведите кодирование одного (произвольного) категориального признака с использованием метода "count (frequency) encoding".

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

```
Всего строк: 256
```

In [ ]:
```
# создание словаря с частотами появления каждого значения признака
counts = data['2. Percentage in Class 10th?'].value_counts().to_dict()
# замена значений признака на их частоты
data['Percentage_count'] = data['2. Percentage in Class 10th?'].map(counts)
```

In [ ]:

Out[ ]:
```
Unnamed: 0
0
2.Gender
0
3. Year of Study
0
4. Branch
0
1. Type of Schooling till 10th Grade:
0
2. Percentage in Class 10th?
0
3. Location of Class 10th School:
0
4. Type of Schooling for class 12th:
0
5. Percentage in Class 12th?
0
6. Location of Class 12th School:
0
7. Dropped an year after School?
0
8. Are you about to become first generation graduate ?
0
9. Involved in additional volunteer groups?(NCC,NSS, Social Welfare Groups  or
any other)      0
1. How many programming languages do you know?
0
2.  Actively involved in Specific technology?
55
3. Exposure to Data structure and algorithms?
0
4. Exposure to GitHub ?
0
5. Are you associated with any developer Community?
0
6. Active in developer Communities
40
7. Participating in Hackathons and other competitions?
0
8. Have you made any Software based projects?
0
9. Have you made any Hardware based project (Arduino, Raspberry Pi, Robots or a
ny other)?      0
10. Have you ever pitched any idea?
0
11. Do you have any additional skills?
37
Achievements:
109
Percentage_count
0
dtype: int64
```

In [ ]:
```
data[['2. Percentage in Class 10th?','Percentage_count']].head(10)
```

Out[ ]:

	2. Percentage in Class 10th?	Percentage_count
0	>=80% and <90%	101
1	>= 90%	116
2	>= 90%	116
3	>=70% and <80%	28
4	>=70% and <80%	28
5	>= 90%	116
6	>=80% and <90%	101
7	>= 90%	116
8	<60%	2
9	>= 90%	116

Таким образом можно увидеть как закодирована успеваемость относительно данного датасета

Задача №21.

Для набора данных проведите масштабирование данных для одного (произвольного) числового признака с использованием масштабирования по медиане.

In [ ]:
```
from sklearn.datasets import load_diabetes
```

In [ ]:
```
diabet_dataset = load_diabetes()
data = pd.DataFrame(diabet_dataset.data,
                    columns=diabet_dataset.feature_names)
data['Y'] = diabet_dataset.target
data.shape
```

Out[ ]:
```
(442, 11)
```

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

Out[ ]:

	age	sex	bmi	bp	s1	s2	s3	s4	
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031

In [ ]:
```
# Нужно ли масштабирование
data.describe()
```

Out[ ]:

	age	sex	bmi	bp	s1	s2
count	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02
mean	-2.511817e-19	1.230790e-17	-2.245564e-16	-4.797570e-17	-1.381499e-17	3.918434e-17
std	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02
min	-1.072256e-01	-4.464164e-02	-9.027530e-02	-1.123988e-01	-1.267807e-01	-1.156131e-01
25%	-3.729927e-02	-4.464164e-02	-3.422907e-02	-3.665608e-02	-3.424784e-02	-3.035840e-02
50%	5.383060e-03	-4.464164e-02	-7.283766e-03	-5.670422e-03	-4.320866e-03	-3.819065e-03
75%	3.807591e-02	5.068012e-02	3.124802e-02	3.564379e-02	2.835801e-02	2.984439e-02
max	1.107267e-01	5.068012e-02	1.705552e-01	1.320436e-01	1.539137e-01	1.987880e-01

In [ ]:
```
# DataFrame не содержащий целевой признак
X_ALL = data.drop('Y', axis=1)
```

In [ ]:
```
from sklearn.preprocessing import RobustScaler
from sklearn.model_selection import train_test_split
```

In [ ]:
```
# Функция для восстановления датафрейма
# на основе масштабированных данных
def arr_to_df(arr_scaled):
    res = pd.DataFrame(arr_scaled, columns=X_ALL.columns)
    return res
```

In [ ]:
```
# Разделим выборку на обучающую и тестовую
X_train, X_test, y_train, y_test = train_test_split(X_ALL, data['Y'],
                                                    test_size=0.2,
                                                    random_state=1)
# Преобразуем массивы в DataFrame
X_train_df = arr_to_df(X_train)
X_test_df = arr_to_df(X_test)

X_train_df.shape, X_test_df.shape
```

Out[ ]:
```
((353, 10), (89, 10))
```

In [ ]:
```
cs41 = RobustScaler()
data_cs41_scaled_temp = cs41.fit_transform(X_ALL)
# формируем DataFrame на основе массива
data_cs41_scaled = arr_to_df(data_cs41_scaled_temp)
data_cs41_scaled.describe()
```

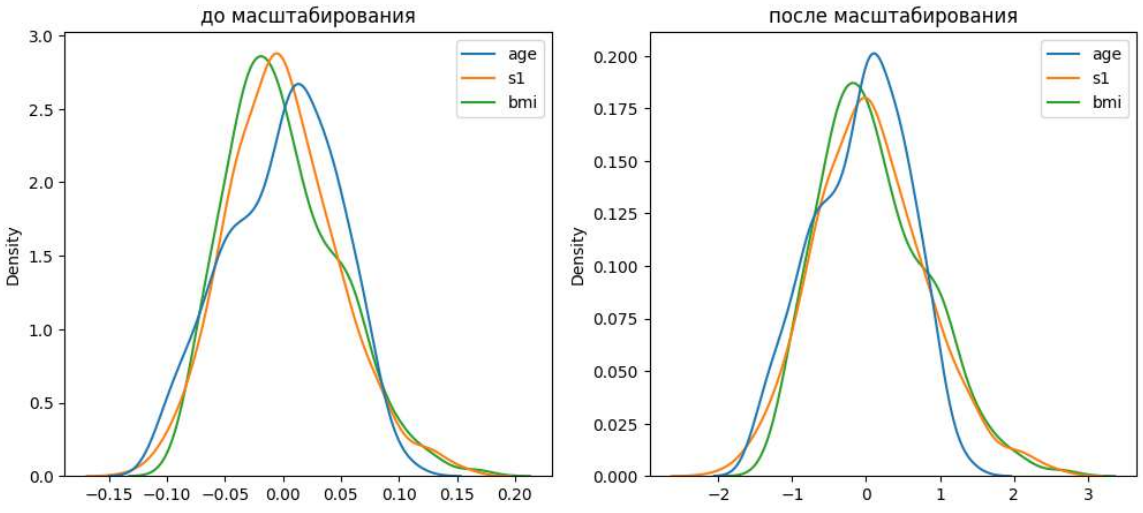
Out[ ]:

	age	sex	bmi	bp	s1	s2	s3	
count	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	44
mean	-0.071417	0.468326	0.111241	0.078429	0.069017	0.063437	0.102198	
std	0.631760	0.499561	0.727263	0.658633	0.760617	0.790977	0.739097	
min	-1.493976	0.000000	-1.267490	-1.476190	-1.956044	-1.856957	-1.485714	-
25%	-0.566265	0.000000	-0.411523	-0.428571	-0.478022	-0.440832	-0.442857	-
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	0.433735	1.000000	0.588477	0.571429	0.521978	0.559168	0.557143	
max	1.397590	1.000000	2.716049	1.904762	2.527473	3.365410	2.914286	

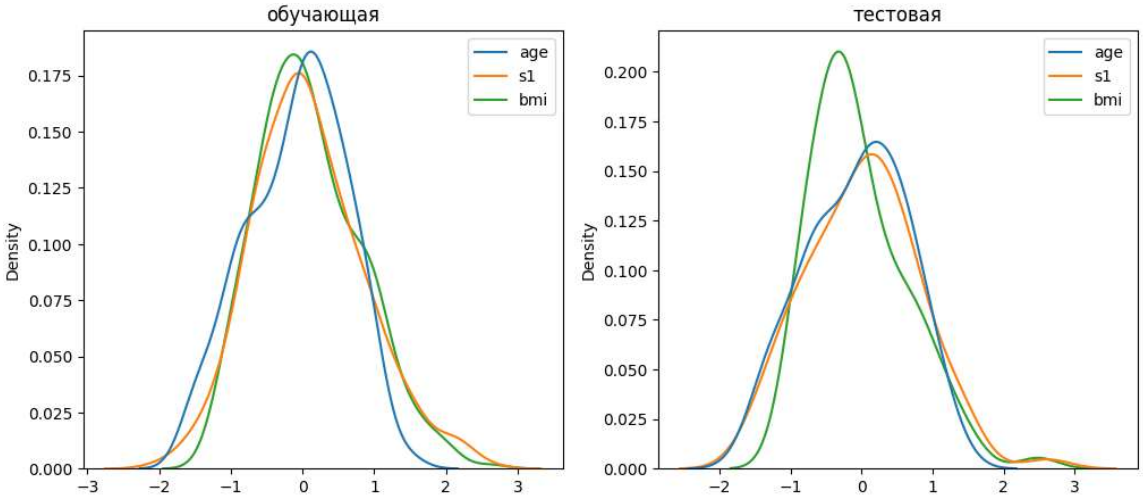
In [ ]:
```
cs42 = RobustScaler()
cs42.fit(X_train)
data_cs42_scaled_train_temp = cs42.transform(X_train)
data_cs42_scaled_test_temp = cs42.transform(X_test)
# формируем DataFrame на основе массива
data_cs42_scaled_train = arr_to_df(data_cs42_scaled_train_temp)
data_cs42_scaled_test = arr_to_df(data_cs42_scaled_test_temp)
```

In [ ]:
```
# Построение плотности распределения
def draw_kde(col_list, df1, df2, label1, label2):
    fig, (ax1, ax2) = plt.subplots(
        ncols=2, figsize=(12, 5))
    # первый график
    ax1.set_title(label1)
    sns.kdeplot(data=df1[col_list], ax=ax1)
    # второй график
    ax2.set_title(label2)
    sns.kdeplot(data=df2[col_list], ax=ax2)
    plt.show()
```

In [ ]:
```
draw_kde(['age', 's1', 'bmi'], data, data_cs41_scaled, 'до масштабирования', 'по
```



In [ ]:
```
draw_kde(['age', 's1', 'bmi'], data_cs42_scaled_train, data_cs42_scaled_test, 'c
```



Дополнительное задание

Для пары произвольных колонок данных построить график "Диаграмма рассеяния".

In [ ]:
```
data = pd.read_csv('India_rainfall_act_dep_1901_2016_1.csv', sep=",")
```

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

Out[ ]:

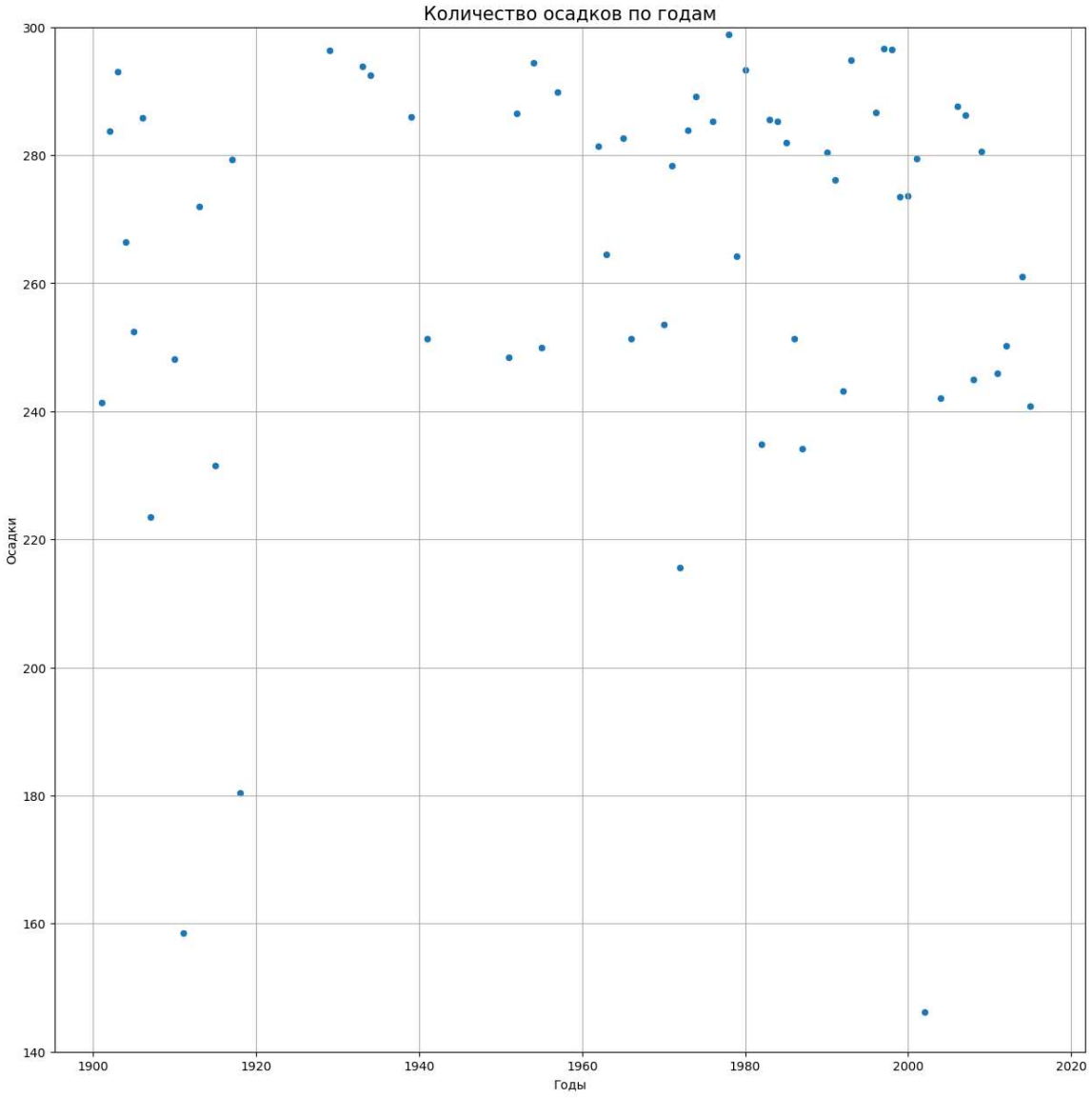
	YEAR	Actual Rainfall: JUN	Actual Rainfall: JUL	Actual Rainfall: AUG	Actual Rainfall: SEPT	Actual Rainfall: JUN-SEPT	Departure Percentage: JUN	Departure Percentage: JUL	Departure Percentage: AUG
0	1901	109.1	241.4	284.2	121.9	756.6	-29.9	-16.7	11.(
1	1902	104.0	283.7	202.6	201.9	792.1	-33.6	-2.0	-21.(
2	1903	114.8	293.0	279.6	204.4	891.9	-26.6	1.6	9.(
3	1904	158.8	266.4	210.4	129.6	765.2	2.8	-7.7	-17.4
4	1905	88.7	252.5	202.6	174.6	718.5	-43.6	-12.3	-21.2

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 116 entries, 0 to 115
Data columns (total 11 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   YEAR                            116 non-null    int64
 1   Actual Rainfall: JUN            116 non-null    float64
 2   Actual Rainfall: JUL            116 non-null    float64
 3   Actual Rainfall: AUG            116 non-null    float64
 4   Actual Rainfall: SEPT           116 non-null    float64
 5   Actual Rainfall: JUN-SEPT       116 non-null    float64
 6   Departure Percentage: JUN       116 non-null    float64
 7   Departure Percentage: JUL       116 non-null    float64
 8   Departure Percentage: AUG       116 non-null    float64
 9   Departure Percentage: SEP       116 non-null    float64
 10  Departure Percentage: JUN-SEPT  116 non-null    float64
dtypes: float64(10), int64(1)
memory usage: 10.1 KB
```

In [ ]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
data.plot(x='YEAR', y='Actual Rainfall: JUL', kind='scatter',  figsize=(15, 15))
plt.title(f'Количество осадков по годам', fontsize=15);
plt.ylim(140,300);
plt.ylabel('Осадки');
plt.xlabel('Годы');
plt.grid(True);
```


Количество осадков по годам
Осадки
Годы
300
280
260
240
220
200
180
160
140
1900
1920
1940
1960
1980
2000
2020

In [ ]: