Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по рубежному контролю №1
Вариант 14

Выполнила:

студент группы ИУ5-61Б

Павловская А.А.

18.04.2021

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Москва, 2021 г.

Задание:

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?

Для студентов групп ИУ5-61Б, ИУ5Ц-81Б - для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Набор данных: Human Resources Data Set

https://www.kaggle.com/rhuebner/human-resources-data-set

Ответы на вопросы:

Для обработки пропусков данных для количественного признака Salary я использовала импьютацию различными показателями центра распределения 'mean', 'median', 'most_frequent' (среднее значение, медиана, мода) с помощью класса SimpleImputer библиотеки scikit-learn.
Для обработки пропусков данных для категориального признака State я также использовала класс SimpleImputer со стратегиями 'most_frequent' или 'constant' (мода и константа).

Кроме того, для всего датасета можно производить удаление строк или колонок, содержащих пустые значения, или заполнение пропусков нулями. Однако подобные методы оказались неэффективными для данного набора данных.

Для дальнейшего построения моделей машинного обучения я, скорее всего, буду использовать количественные признаки (и переводить категориальные в количественные), так как многие алгоритмы машинного обучения используют количественные признаки.

Текст программы и экранные формы с примерами выполнения программы (ячейки ноутбука):

ИУ5-61Б Павловская А.А. РК1 ТМО

Вариант 14

Набор данных:

Human Resources Data Set

In [1]:

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

In [2]:

```
# Загрузка данных
data = pd.read_csv('archive/HRDataset_v14.csv')
```

In [4]:

```
# Обзор датасета
data.head()
```

Out[4]:

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversity
Adinolfi	**Wilson K**	**10026.0**	**0.0**	**0.0**	**1.0**	**1.0**	**5.0**	**4.0**	
Ait Sidi	**Karthikeyan**	**10084.0**	**1.0**	**1.0**	**1.0**	**5.0**	**3.0**	**3.0**	
Akinkuolie	**Sarah**	**10196.0**	**1.0**	**1.0**	**0.0**	**5.0**	**5.0**	**3.0**	
Alagbe	**Trina**	**10088.0**	**1.0**	**1.0**	**0.0**	**1.0**	**5.0**	**3.0**	
Anderson	**Carol**	**10069.0**	**0.0**	**2.0**	**0.0**	**5.0**	**5.0**	**3.0**	

5 rows × 36 columns

In [5]:

```
data.shape
```

Out[5]:

```
(311, 36)
```

In [6]:

```
data.columns
```

Out[6]:

```
Index(['Employee_Name', 'EmpID', 'MarriedID', 'MaritalStatusID', 'GenderID',
       'EmpStatusID', 'DeptID', 'PerfScoreID', 'FromDiversityJobFairID',
       'Salary', 'Termd', 'PositionID', 'Position', 'State', 'Zip', 'DOB',
       'Sex', 'MaritalDesc', 'CitizenDesc', 'HispanicLatino', 'RaceDesc',
       'DateofHire', 'DateofTermination', 'TermReason', 'EmploymentStatus',
       'Department', 'ManagerName', 'ManagerID', 'RecruitmentSource',
       'PerformanceScore', 'EngagementSurvey', 'EmpSatisfaction',
       'SpecialProjectsCount', 'LastPerformanceReview_Date', 'DaysLateLast30',
```

```
       'Absences'],
      dtype='object')
```

In [7]:

```
data.dtypes
```

Out[7]:

```
Employee_Name                 object
EmpID                        float64
MarriedID                    float64
MaritalStatusID              float64
GenderID                     float64
EmpStatusID                  float64
DeptID                       float64
PerfScoreID                  float64
FromDiversityJobFairID       float64
Salary                       float64
Termd                        float64
PositionID                   float64
Position                      object
State                         object
Zip                          float64
DOB                           object
Sex                           object
MaritalDesc                   object
CitizenDesc                   object
HispanicLatino                object
RaceDesc                      object
DateofHire                    object
DateofTermination             object
TermReason                    object
EmploymentStatus              object
Department                    object
ManagerName                   object
ManagerID                    float64
RecruitmentSource             object
PerformanceScore              object
EngagementSurvey             float64
EmpSatisfaction              float64
SpecialProjectsCount         float64
LastPerformanceReview_Date    object
DaysLateLast30               float64
Absences                     float64
dtype: object
```

In [8]:

```
# Проверка наличия пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
Employee_Name - 4
EmpID - 4
MarriedID - 4
MaritalStatusID - 4
GenderID - 4
EmpStatusID - 4
DeptID - 4
PerfScoreID - 4
FromDiversityJobFairID - 4
Salary - 4
Termd - 4
PositionID - 4
Position - 4
State - 4
Zip - 4
DOB - 4
```

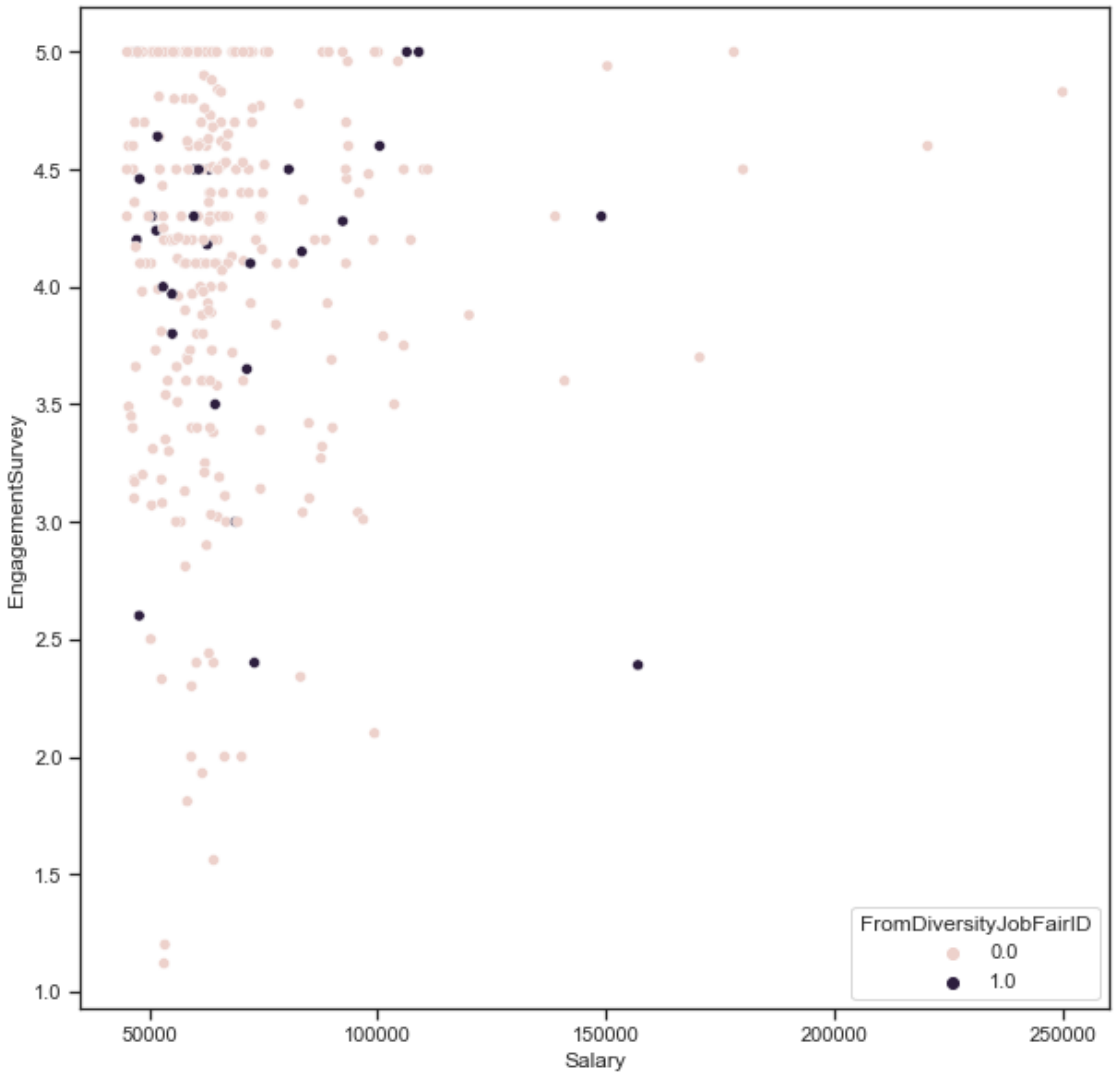
```
Sex - 4
MaritalDesc - 4
CitizenDesc - 4
HispanicLatino - 4
RaceDesc - 4
DateofHire - 4
DateofTermination - 211
TermReason - 4
EmploymentStatus - 4
Department - 4
ManagerName - 4
ManagerID - 12
RecruitmentSource - 4
PerformanceScore - 4
EngagementSurvey - 4
EmpSatisfaction - 4
SpecialProjectsCount - 4
LastPerformanceReview_Date - 4
DaysLateLast30 - 4
Absences - 4
```

In [9]:

```
# Диаграмма рассеивания
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Salary', y='EngagementSurvey', data=data,hue = 'FromDiversityJobFairID')
```

Out[9]:

```
<AxesSubplot:xlabel='Salary', ylabel='EngagementSurvey'>
```



In [10]:

```
total_count = data.shape[0]
```

Удаление или заполнение нулями

In [44]:

```
# Удаление колонок, содержащих пустые значения
# В данном случае такое удаление колонок некорректно, так как пропуски были во всех колон
ках
data_new_1 = data.dropna(axis=1, how='any')
(data.shape, data_new_1.shape)
```

Out[44]:

```
((311, 36), (311, 0))
```

In [40]:

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

Out[40]:

```
((311, 36), (100, 36))
```

Второй метод тоже не показал себя эффективным, так как была удалена значительная часть строк датасета

In [42]:

```
data.head()
```

Out[42]:

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversity
Adinolfi	**Wilson K**	**10026.0**	**0.0**	**0.0**	**1.0**	**1.0**	**5.0**	**4.0**	
Ait Sidi	**Karthikeyan**	**10084.0**	**1.0**	**1.0**	**1.0**	**5.0**	**3.0**	**3.0**	
Akinkuolie	**Sarah**	**10196.0**	**1.0**	**1.0**	**0.0**	**5.0**	**5.0**	**3.0**	
Alagbe	**Trina**	**10088.0**	**1.0**	**1.0**	**0.0**	**1.0**	**5.0**	**3.0**	
Anderson	**Carol**	**10069.0**	**0.0**	**2.0**	**0.0**	**5.0**	**5.0**	**3.0**	

5 rows × 36 columns

In [43]:

```
# Заполнение всех пропущенных значений нулями
# Для данного датасета способ не подходит, так как содержатся категориальные признаки с п
ропусками
data_new_3 = data.fillna(0)
data_new_3.head()
```

Out[43]:

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversity
Adinolfi	**Wilson K**	**10026.0**	**0.0**	**0.0**	**1.0**	**1.0**	**5.0**	**4.0**	
Ait Sidi	**Karthikeyan**	**10084.0**	**1.0**	**1.0**	**1.0**	**5.0**	**3.0**	**3.0**	
Akinkuolie	**Sarah**	**10196.0**	**1.0**	**1.0**	**0.0**	**5.0**	**5.0**	**3.0**	
Alagbe	**Trina**	**10088.0**	**1.0**	**1.0**	**0.0**	**1.0**	**5.0**	**3.0**	
Anderson	**Carol**	**10069.0**	**0.0**	**2.0**	**0.0**	**5.0**	**5.0**	**3.0**	

5 rows × 36 columns

"Внедрение значений" - импьютация

Обработка пропусков для количественного признака **Salary**

In [27]:

```
# 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'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(co
l, dt, temp_null_count, temp_perc))
```

```
Колонка EmpID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка MarriedID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка MaritalStatusID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка GenderID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка EmpStatusID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка DeptID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка PerfScoreID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка FromDiversityJobFairID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка Salary. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка Termd. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка PositionID. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка Zip. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка ManagerID. Тип данных float64. Количество пустых значений 12, 3.86%.
Колонка EngagementSurvey. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка EmpSatisfaction. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка SpecialProjectsCount. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка DaysLateLast30. Тип данных float64. Количество пустых значений 4, 1.29%.
Колонка Absences. Тип данных float64. Количество пустых значений 4, 1.29%.
```

In [28]:

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

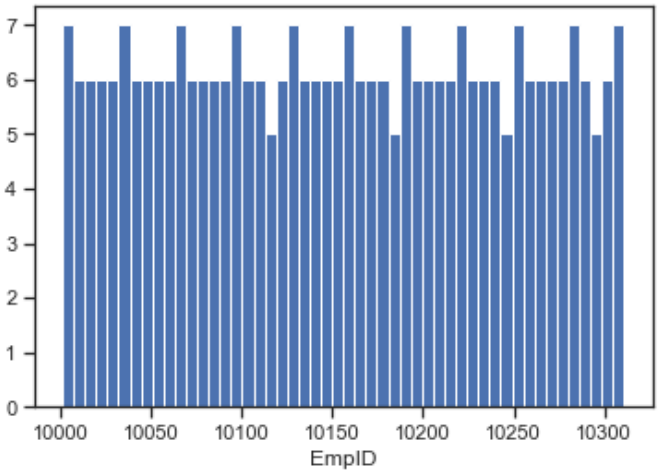
Out[28]:

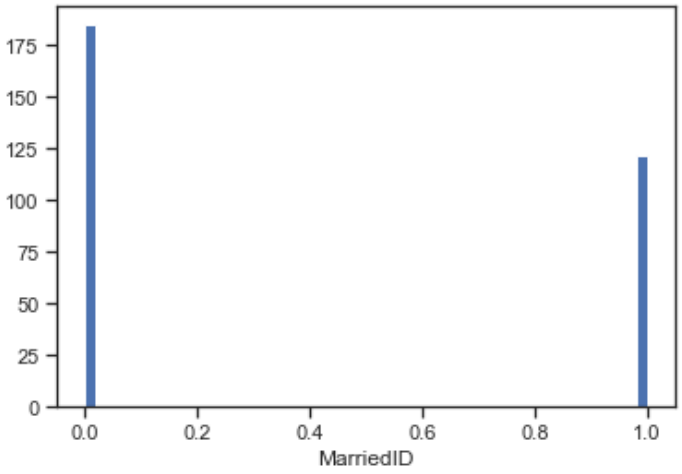
	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairID	Sala
Adinolfi	10026.0	0.0	0.0	1.0	1.0	5.0	4.0	0.0	62506
Ait Sidi	10084.0	1.0	1.0	1.0	5.0	3.0	3.0	0.0	104437
Akinkuolie	10196.0	1.0	1.0	0.0	5.0	5.0	3.0	0.0	64955
Alagbe	10088.0	1.0	1.0	0.0	1.0	5.0	3.0	0.0	64991
Anderson	10069.0	0.0	2.0	0.0	5.0	5.0	3.0	0.0	50825
...	...	...	...	...	...	...	...	...	
Woodson	10135.0	0.0	0.0	1.0	1.0	5.0	3.0	0.0	65893
Ybarra	10301.0	0.0	0.0	0.0	5.0	5.0	1.0	0.0	48513
Zamora	10010.0	0.0	0.0	0.0	1.0	3.0	4.0	0.0	220450
Zhou	10043.0	0.0	0.0	0.0	1.0	3.0	3.0	0.0	89292
Zima	10271.0	0.0	4.0	0.0	1.0	5.0	3.0	0.0	45046

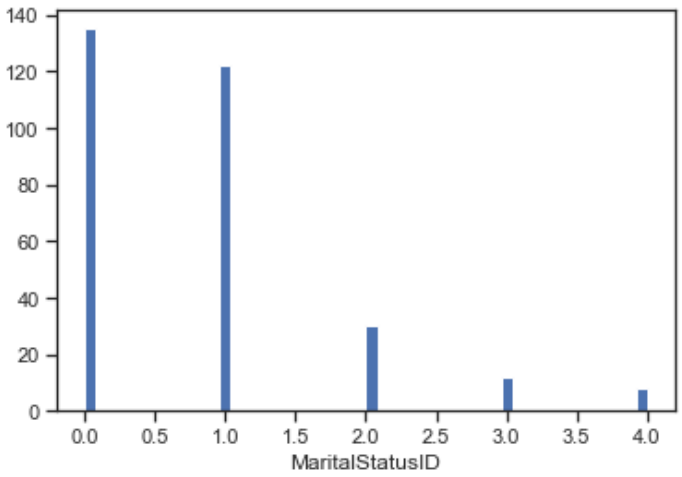
311 rows × 18 columns

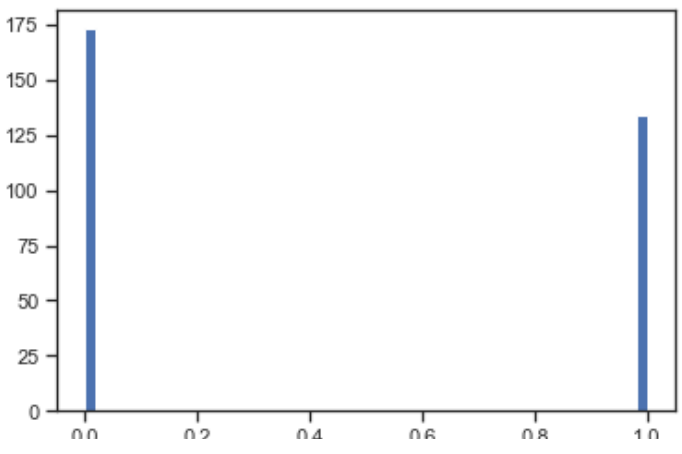
In [29]:

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

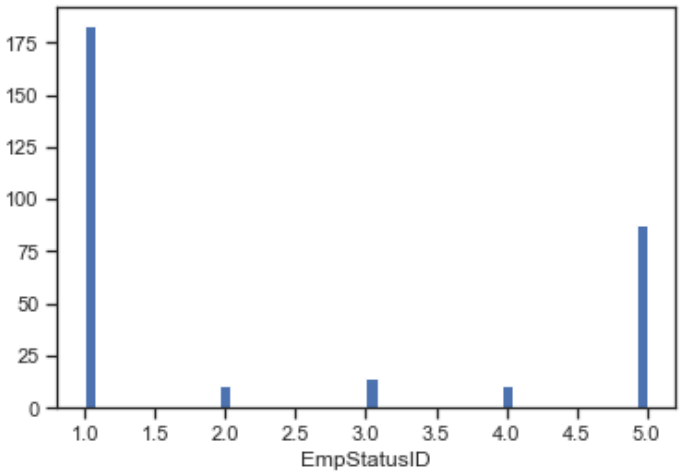


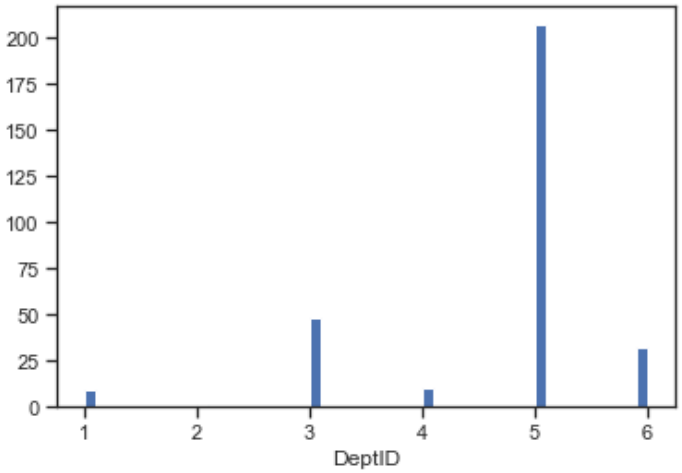


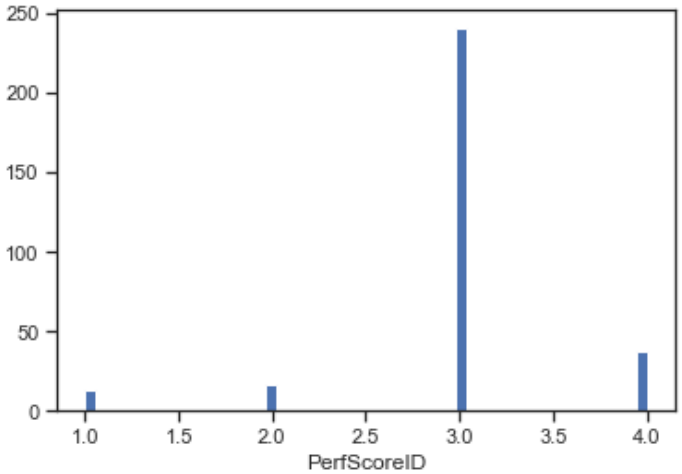


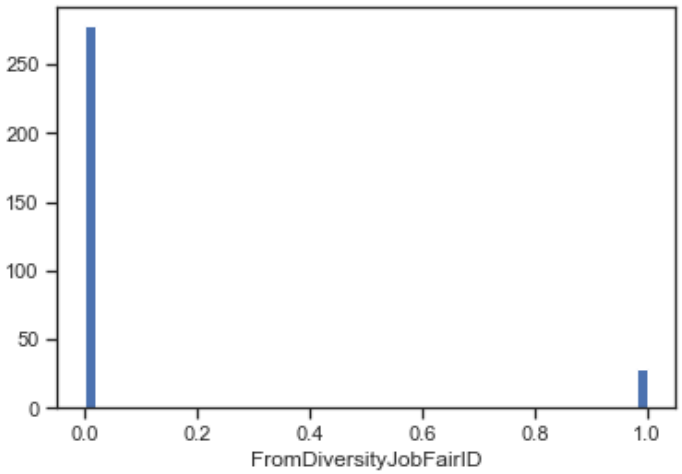



GenderID

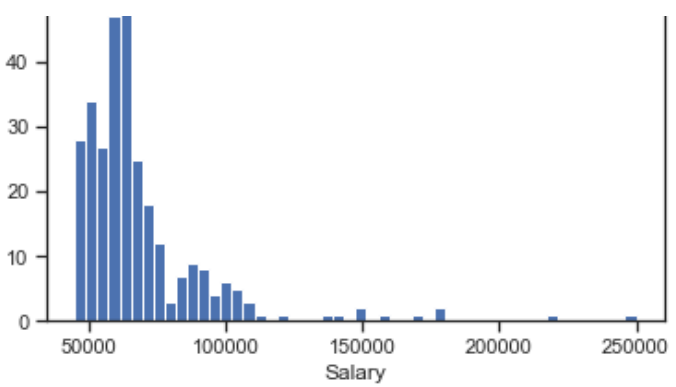

175
150
125
100
75
50
25
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
EmpStatusID

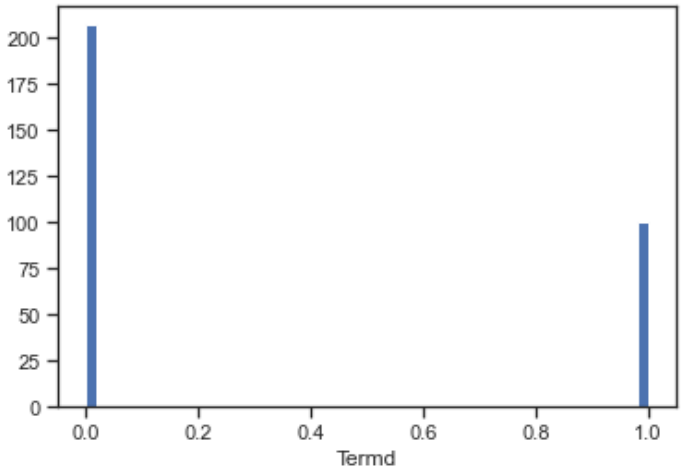

200
175
150
125
100
75
50
25
0
1
2
3
4
5
6
DeptID

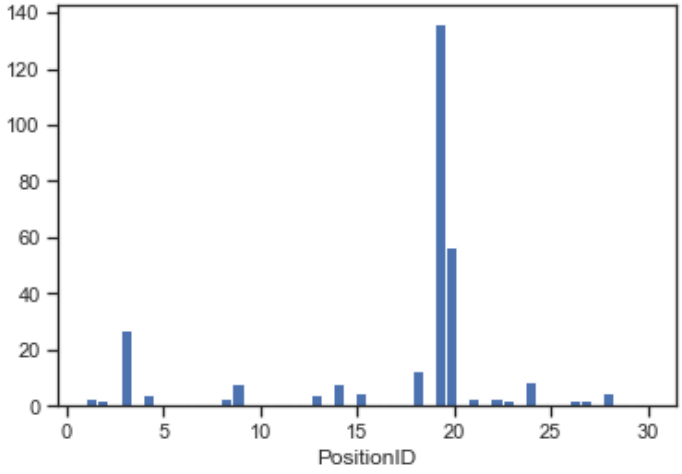

250
200
150
100
50
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
PerfScoreID

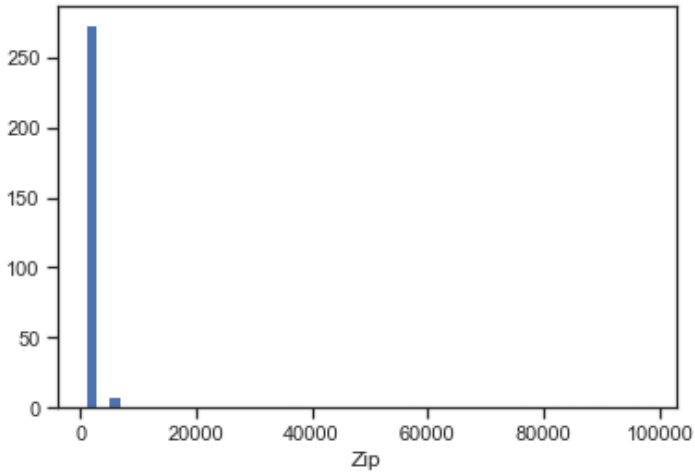

250
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8
1.0
FromDiversityJobFairID


60
50


40
30
20
10
0
50000
100000
150000
200000
250000
Salary

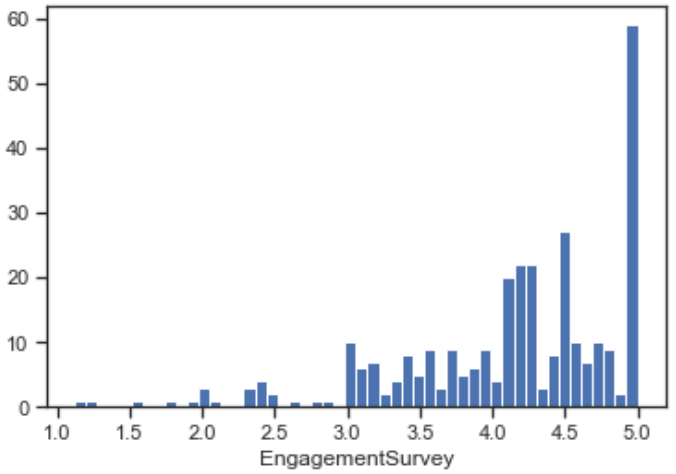

200
175
150
125
100
75
50
25
0
0.0
0.2
0.4
0.6
0.8
1.0
Termd

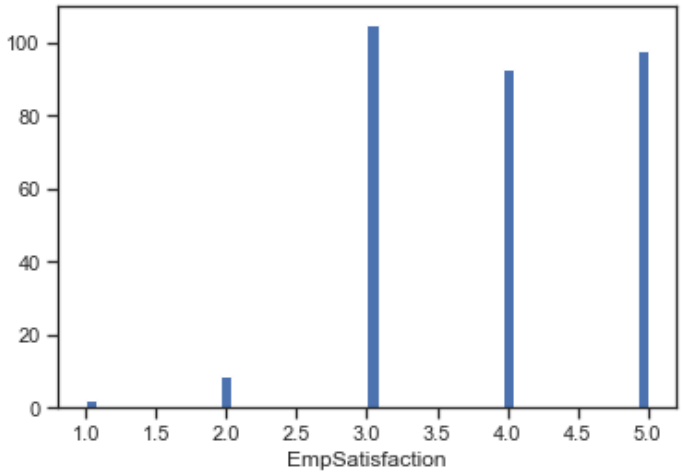

140
120
100
80
60
40
20
0
0
5
10
15
20
25
30
PositionID

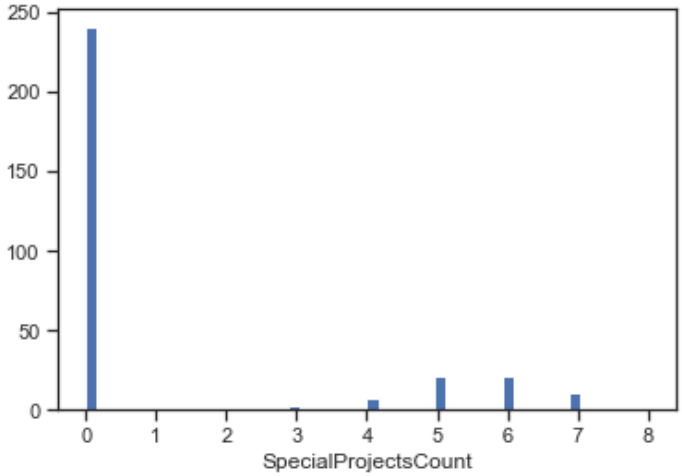

250
200
150
100
50
0
0
20000
40000
60000
80000
100000
Zip

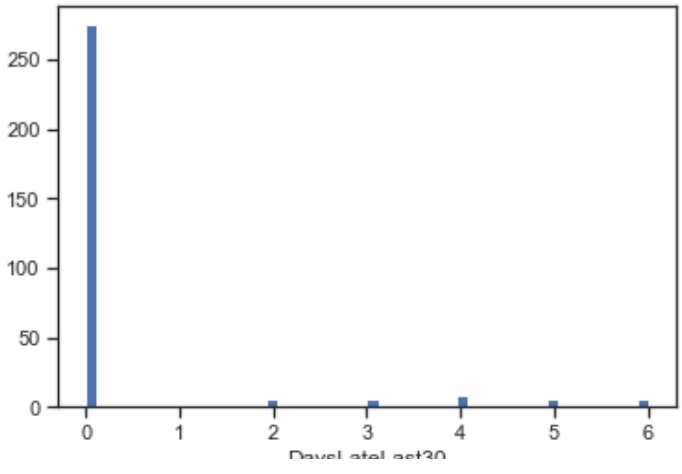

20
15
10

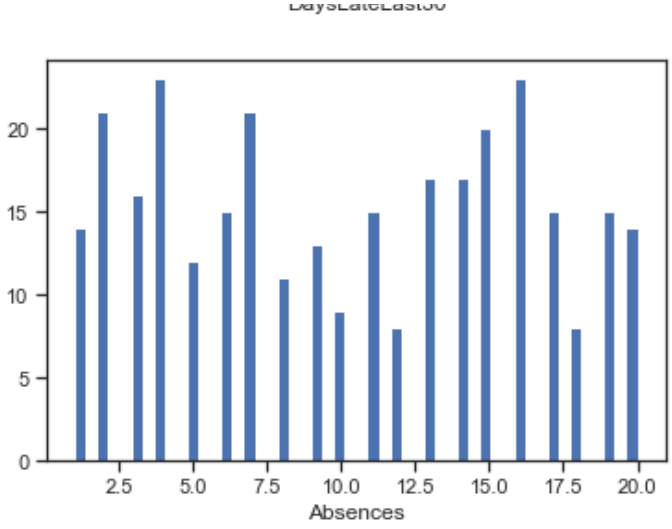

0
5
0
5
10
15
20
25
30
35
40
ManagerID


0
10
20
30
40
50
60
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
EngagementSurvey


0
20
40
60
80
100
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
EmpSatisfaction


0
50
100
150
200
250
0
1
2
3
4
5
6
7
8
SpecialProjectsCount


0
50
100
150
200
250
0
1
2
3
4
5
6



In [16]:

```
data_num_Salary = data_num[['Salary']]
data_num_Salary.head()
```

Out[16]:

	Salary
Adinolfi	62506.0
Ait Sidi	104437.0
Akinkuolie	64955.0
Alagbe	64991.0
Anderson	50825.0

In [17]:

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

In [18]:

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

Out[18]:

```
array([[False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [ True],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
          [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
        [False],
```

```
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [ True],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
       [False],
```

```
       [False]])
```

In [20]:

```
# Импьютация различными показателями центра распределения с помощью класса SimpleImputer
strategies=['mean', 'median', 'most_frequent']
```

In [21]:

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

In [22]:

```
# Среднее значение
strategies[0], test_num_impute(strategies[0])
```

Out[22]:

```
('mean',
 array([68819.51140065, 68819.51140065, 68819.51140065, 68819.51140065]))
```

In [23]:

```
# Медиана
strategies[1], test_num_impute(strategies[1])
```

Out[23]:

```
('median', array([62810., 62810., 62810., 62810.]))
```

In [24]:

```
# Мода
strategies[2], test_num_impute(strategies[2])
```

Out[24]:

```
('most_frequent', array([57815., 57815., 57815., 57815.]))
```

Обработка пропусков для категориального признака **State**

In [25]:

```
# Выбор категориальных колонок 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(co
l, dt, temp_null_count, temp_perc))
```

```
Колонка Employee_Name. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка Position. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка State. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка DOB. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка Sex. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка MaritalDesc. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка CitizenDesc. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка HispanicLatino. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка RaceDesc. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка DateofHire. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка DateofTermination. Тип данных object. Количество пустых значений 211, 67.85%.
Колонка TermReason. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка EmploymentStatus. Тип данных object. Количество пустых значений 4, 1.29%
```

Колонка EmploymentStatus. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка Department. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка ManagerName. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка RecruitmentSource. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка PerformanceScore. Тип данных object. Количество пустых значений 4, 1.29%.
Колонка LastPerformanceReview_Date. Тип данных object. Количество пустых значений 4, 1.29
%.

In [30]:

```
# Импьютация с помощью класса SimpleImputer со стратегиями "most_frequent" или "constant"
.
cat_temp_data = data[['State']]
cat_temp_data.head()
```

Out[30]:

	State
Adinolfi	**MA**
Ait Sidi	**MA**
Akinkuolie	**MA**
Alagbe	**MA**
Anderson	**MA**

In [31]:

```
cat_temp_data['State'].unique()
```

Out[31]:

```
array(['MA', 'TX', 'CT', 'VA', 'VT', 'AL', 'WA', 'CA', nan, 'OH', 'IN',
       'TN', 'NH', 'RI', 'PA', 'CO', 'NY', 'UT', 'GA', 'FL', 'NC', 'KY',
       'ID', 'NV', 'MT', 'OR', 'ND', 'AZ', 'ME'], dtype=object)
```

In [32]:

```
cat_temp_data[cat_temp_data['State'].isnull()].shape
```

Out[32]:

```
(4, 1)
```

In [33]:

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

Out[33]:

```
array([['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['TX'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
```

['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['CT'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['VA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['VT'],
['MA'],
['MA'],
['MA'],
['MA'],
['TX'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['CT'],
['VT'],
['CT'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['AL'],
['MA'],
['MA'],
['MA'],
['MA'],
['WA'],
['MA'],
['MA'],
['MA'],
['MA'],
['CA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],

['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['CT'],
['CT'],
['MA'],
['MA'],
['MA'],
['MA'],
['OH'],
['MA'],
['MA'],
['IN'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['TN'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['NH'],
['MA'],
['MA'],
['RI'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['PA'],
['MA'],
['CO'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['NY'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],

['MA'],
['MA'],
['MA'],
['UT'],
['MA'],
['MA'],
['MA'],
['CT'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['TX'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['GA'],
['MA'],
['MA'],
['MA'],
['MA'],
['FL'],
['MA'],
['MA'],
['NC'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['KY'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['ID'],
['MA'],
['MA'],
['MA'],
['MA'],

```
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['NV'],
['MA'],
['MA'],
['MT'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['OR'],
['MA'],
['MA'],
['MA'],
['ND'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['AZ'],
['MA'],
['MA'],
['ME'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
```

```
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA']], dtype=object)
```

In [34]:

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

Out[34]:

```
array(['AL', 'AZ', 'CA', 'CO', 'CT', 'FL', 'GA', 'ID', 'IN', 'KY', 'MA',
       'ME', 'MT', 'NC', 'ND', 'NH', 'NV', 'NY', 'OH', 'OR', 'PA', 'RI',
       'TN', 'TX', 'UT', 'VA', 'VT', 'WA'], dtype=object)
```

In [35]:

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

Out[35]:

```
array([['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['TX'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['CT'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['VA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['VT'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['TX'],
       ['MA'],
```

['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['CT'],
['VT'],
['CT'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['AL'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['WA'],
['MA'],
['MA'],
['MA'],
['MA'],
['CA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['N/A'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['CT'],
['CT'],
['MA'],
['MA'],
['MA'],
['MA'],
['OH'],
['MA'],
['MA'],
['IN'],
['N/A'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],

['MA'],
['MA'],
['TN'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['NH'],
['MA'],
['MA'],
['RI'],
['N/A'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['PA'],
['MA'],
['CO'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['NY'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['UT'],
['MA'],
['MA'],
['MA'],
['CT'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['TX'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],

```
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['GA'],
['MA'],
['MA'],
['MA'],
['MA'],
['FL'],
['MA'],
['MA'],
['NC'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['KY'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['ID'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
['MA'],
```

```
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['NV'],
       ['MA'],
       ['MA'],
       ['MT'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['OR'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['ND'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['AZ'],
       ['MA'],
       ['MA'],
       ['ME'],
       ['N/A'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA'],
       ['MA']], dtype=object)
```

In [36]:

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

Out[36]:

```
array(['AL', 'AZ', 'CA', 'CO', 'CT', 'FL', 'GA', 'ID', 'IN', 'KY', 'MA',
       'ME', 'MT', 'N/A', 'NC', 'ND', 'NH', 'NV', 'NY', 'OH', 'OR', 'PA',
       'RI', 'TN', 'TX', 'UT', 'VA', 'VT', 'WA'], dtype=object)
```

In [45]:

```
data_imp3[data_imp3=='N/A'].size
```

Out[45]:

```
4
```

Для признака **State** я считаю более корректным внедрение константы **N/A,** так как название штата может и не соответствовать моде

In [ ]: