Лабораторная работа №2

Лабораторная работа №2

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

Введение

В качестве набора данных будем использовать датасет "HR Dataset". В датасете содержится информация о сотрудниках компании.
Ссылка на датасет: https://www.kaggle.com/datasets/imtiajemon/hr-dataset/data
Датасет содержит следующие поля.

1. **Employee_ID** - уникальный идентификатор сотрудника
2. **Name** - имя сотрудника
3. **Gender** - пол сотрудника
4. **Department** - отдел, в котором работает сотрудник
5. **EducationField** - область образования
6. **MaritalStatus** - семейное положение
7. **JobRole** - должность
8. **JobLevel** - уровень должности
9. **Age** - возраст
10. **MonthlyIncome** - месячный доход
11. **NumCompaniesWorked** - количество компаний, в которых работал сотрудник
12. **TotalWorkingYears** - общий стаж работы
13. **TrainingTimesLastYear** - количество тренингов за последний год
14. **YearsAtCompany** - стаж работы в компании
15. **YearsInCurrentRole** - стаж в текущей должности
16. **YearsSinceLastPromotion** - лет с последнего повышения
17. **YearsWithCurrManager** - лет с текущим менеджером
18. **Attrition** - увольнение (целевая переменная)

Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:

- обработку пропусков в данных;
- кодирование категориальных признаков;
- масштабирование данных.

Загрузка данных

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, OneHotEncoder,
StandardScaler

# Загрузка данных
df = pd.read_csv('HR_Data.csv')
print(df.head())
print("\nИнформация о данных:")
print(df.info())
print("\nПропуски в данных:")
print(df.isnull().sum())
```

```
   EmpID  Age AgeGroup Attrition     BusinessTravel              Department
\
0  RM297   18    18-25       Yes      Travel_Rarely  Research & Development
1  RM302   18    18-25        No      Travel_Rarely                   Sales
2  RM458   18    18-25       Yes  Travel_Frequently                   Sales
3  RM728   18    18-25        No         Non_Travel  Research & Development
4  RM829   18    18-25       Yes         Non_Travel  Research & Development

   DistanceFromHome EducationField EnvironmentSatisfaction  Gender  ...  \
0                 3  Life Sciences                 Average    Male  ...
1                10        Medical                    Good  Female  ...
2                 5      Marketing                    Poor    Male  ...
3                 5  Life Sciences                    Poor    Male  ...
4                 8        Medical                 Average    Male  ...

   MonthlyIncome Over18 OverTime PercentSalaryHike  PerformanceRating  \
0           1420      Y       No                13            Average
1           1200      Y       No                12            Average
2           1878      Y      Yes                14            Average
3           1051      Y       No                15            Average
4           1904      Y       No                12            Average

  TotalWorkingYears YearsAtCompany  YearsInCurrentRole  \
0                 0              0                   0
1                 0              0                   0
2                 0              0                   0
3                 0              0                   0
4                 0              0                   0
```

```
  YearsSinceLastPromotion  YearsWithCurrManager
0                       0                   0.0
1                       0                   0.0
2                       0                   0.0
3                       0                   0.0
4                       0                   0.0

[5 rows x 24 columns]

Информация о данных:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1473 entries, 0 to 1472
Data columns (total 24 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   EmpID                     1473 non-null   object
 1   Age                       1473 non-null   int64
 2   AgeGroup                  1473 non-null   object
 3   Attrition                 1473 non-null   object
 4   BusinessTravel            1473 non-null   object
 5   Department                1473 non-null   object
 6   DistanceFromHome          1473 non-null   int64
 7   EducationField            1473 non-null   object
 8   EnvironmentSatisfaction   1473 non-null   object
 9   Gender                    1473 non-null   object
 10  JobLevel                  1473 non-null   int64
 11  JobRole                   1473 non-null   object
 12  JobSatisfaction           1473 non-null   object
 13  MaritalStatus             1473 non-null   object
 14  MonthlyIncome             1473 non-null   int64
 15  Over18                    1473 non-null   object
 16  OverTime                  1473 non-null   object
 17  PercentSalaryHike         1473 non-null   int64
 18  PerformanceRating         1473 non-null   object
 19  TotalWorkingYears         1473 non-null   int64
 20  YearsAtCompany            1473 non-null   int64
 21  YearsInCurrentRole        1473 non-null   int64
 22  YearsSinceLastPromotion   1473 non-null   int64
 23  YearsWithCurrManager      1473 non-null   float64
dtypes: float64(1), int64(9), object(14)
memory usage: 276.3+ KB
None
```

```
Пропуски в данных:
EmpID                     0
Age                       0
AgeGroup                  0
Attrition                 0
BusinessTravel            0
Department                0
DistanceFromHome          0
EducationField            0
EnvironmentSatisfaction   0
Gender                    0
JobLevel                  0
JobRole                   0
JobSatisfaction           0
MaritalStatus             0
MonthlyIncome             0
Over18                    0
OverTime                  0
PercentSalaryHike         0
PerformanceRating         0
TotalWorkingYears         0
YearsAtCompany            0
YearsInCurrentRole        0
YearsSinceLastPromotion   0
YearsWithCurrManager      0
dtype: int64
```

Обработка пропусков

```python
# Заполнение пропусков (как в предыдущем коде)
categorical_cols = df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])

numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns
for col in numeric_cols:
    df[col] = df[col].fillna(df[col].median())

# Проверка пропусков после обработки
print("\n=== Пропуски после обработки ===")
print(df.isnull().sum())
```

```
=== Пропуски после обработки ===
EmpID                     0
Age                       0
AgeGroup                  0
Attrition                 0
BusinessTravel            0
Department                0
DistanceFromHome          0
EducationField            0
EnvironmentSatisfaction   0
Gender                    0
JobLevel                  0
JobRole                   0
JobSatisfaction           0
MaritalStatus             0
MonthlyIncome             0
Over18                    0
OverTime                  0
PercentSalaryHike         0
PerformanceRating         0
TotalWorkingYears         0
YearsAtCompany            0
YearsInCurrentRole        0
YearsSinceLastPromotion   0
YearsWithCurrManager      0
dtype: int64
```

Кодирование категориальных признаков

```python
# Label Encoding для признаков с порядковой зависимостью (если есть)
label_encoder = LabelEncoder()
ordinal_cols = ['JobLevel', 'PerformanceRating']  # Пример
for col in ordinal_cols:
    df[col] = label_encoder.fit_transform(df[col])

# One-Hot Encoding для номинальных признаков
nominal_cols = ['Department', 'Gender', 'MaritalStatus']  # Пример
df = pd.get_dummies(df, columns=nominal_cols, drop_first=True)  # drop_first для избежания дамми-ловушки

print("\nДанные после кодирования:")
print(df.head())
```

Данные после кодирования:

	EmpID	Age	AgeGroup	Attrition	BusinessTravel	DistanceFromHome	\
0	RM297	18	18-25	Yes	Travel_Rarely	3	
1	RM302	18	18-25	No	Travel_Rarely	10	
2	RM458	18	18-25	Yes	Travel_Frequently	5	
3	RM728	18	18-25	No	Non_Travel	5	
4	RM829	18	18-25	Yes	Non_Travel	8	

	EducationField	EnvironmentSatisfaction	JobLevel	JobRole	\
0	Life Sciences	Average	0	Laboratory Technician	
1	Medical	Good	0	Sales Representative	
2	Marketing	Poor	0	Sales Representative	
3	Life Sciences	Poor	0	Research Scientist	
4	Medical	Average	0	Laboratory Technician	

	...	TotalWorkingYears	YearsAtCompany	YearsInCurrentRole	\
0	...	0	0	0	
1	...	0	0	0	
2	...	0	0	0	
3	...	0	0	0	
4	...	0	0	0	

	YearsSinceLastPromotion	YearsWithCurrManager	\
0	0	0.0	
1	0	0.0	
2	0	0.0	
3	0	0.0	
4	0	0.0	

	Department_Research & Development	Department_Sales	Gender_Male	\
0	True	False	True	
1	False	True	False	
2	False	True	True	
3	True	False	True	
4	True	False	True	

	MaritalStatus_Married	MaritalStatus_Single
0	False	True
1	False	True
2	False	True
3	False	True
4	False	True

```
[5 rows x 26 columns]
```

Масштабирование данных

```python
# Выбор числовых признаков для масштабирования
numeric_features = ['Age', 'MonthlyIncome', 'TotalWorkingYears',
'YearsAtCompany']
scaler = StandardScaler()
df[numeric_features] = scaler.fit_transform(df[numeric_features])

print("\nДанные после масштабирования:")
print(df[numeric_features].head())
```

```
Данные после масштабирования:
        Age  MonthlyIncome  TotalWorkingYears  YearsAtCompany
0 -2.072527      -1.079876          -1.450767       -1.144768
1 -2.072527      -1.126640          -1.450767       -1.144768
2 -2.072527      -0.982521          -1.450767       -1.144768
3 -2.072527      -1.158312          -1.450767       -1.144768
4 -2.072527      -0.976995          -1.450767       -1.144768
```