In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import preProcessing_uniTeh as pu
```

In [2]:

```
from scipy import stats
from IPython.core.display import display, HTML
from pylab import rcParams
```

In [3]:

```
import warnings
warnings.filterwarnings("ignore")
```

: ساختار بندی و مجتمع کرگلان 1داده

In [4]:

```
data = pd.read_csv('F:/O_C/T_U_C/dS_C9/7_Py(T)/3T/projects_classification/HR/HR.csv')
```

In [5]:

```
# pd.set_option('display.max_rows', 700)
```

In [6]:

```
data
```

Out[6]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
14994	0.40	0.57	2	151	3	0	1	
14995	0.37	0.48	2	160	3	0	1	
14996	0.37	0.53	2	143	3	0	1	
14997	0.11	0.96	6	280	4	0	1	
14998	0.37	0.52	2	158	3	0	1	

14999 rows × 10 columns

In [7]:

```
df = data.copy()
```

In [8]:

```
df.columns
```

Out[8]:

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'department', 'salary'],
      dtype='object')
```

In [9]:

```
df.dtypes
```

Out[9]:

```
satisfaction_level       float64
last_evaluation          float64
number_project             int64
average_montly_hours       int64
time_spend_company         int64
Work_accident              int64
left                       int64
promotion_last_5years      int64
department                object
salary                    object
dtype: object
```

In [10]:

```python
duplicate = df[df.duplicated(keep = 'last')]
duplicate
```

Out[10]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
12658	0.38	0.53	2	146	3	0	1	
12659	0.77	0.91	5	221	6	0	1	
12660	0.44	0.50	2	130	3	0	1	
12661	0.39	0.46	2	136	3	0	1	
14234	0.46	0.57	2	139	3	0	1	

3008 rows × 10 columns

In [11]:

```python
duplicate = df[df.duplicated(keep = 'last')]
duplicate
```

Out[11]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
12658	0.38	0.53	2	146	3	0	1	
12659	0.77	0.91	5	221	6	0	1	
12660	0.44	0.50	2	130	3	0	1	
12661	0.39	0.46	2	136	3	0	1	
14234	0.46	0.57	2	139	3	0	1	

3008 rows × 10 columns

In [12]:

```python
df = data.drop_duplicates(keep='first')
```

In [13]:

```python
df.reset_index(inplace=True)
```

In [14]:

```python
df = df.drop(columns='index')
```

In [15]:

```python
df
```

Out[15]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	

11989	0.33	0.65	3	172	10	0	0
11990	0.50	0.73	4	180	3	0	0

11991 rows × 10 columns

In [16]:

```
df['left'].value_counts()
```

Out[16]:

```
0    10000
1     1991
Name: left, dtype: int64
```

In [17]:

```
col = df.columns
```

In [18]:

```
for i in col:
    print('+++++ {} ++++++++++++++'.format(i))
    print(df[i].value_counts())
```

```
+++++ satisfaction_level ++++++++++++++
0.74    214
0.10    203
0.73    201
0.50    200
0.72    199
       ...
0.25     29
0.26     28
0.12     26
0.28     24
0.27     23
Name: satisfaction_level, Length: 92, dtype: int64
+++++ last_evaluation ++++++++++++++
0.55    281
0.50    269
0.51    264
0.57    258
0.54    252
       ...
0.42     45
0.43     44
0.38     42
0.44     35
0.36     19
Name: last_evaluation, Length: 65, dtype: int64
+++++ number_project ++++++++++++++
4    3685
3    3520
5    2233
2    1582
6     826
7     145
Name: number_project, dtype: int64
+++++ average_montly_hours ++++++++++++++
156    112
149    112
160    111
151    107
135    104
      ...
298      5
302      5
297      5
299      5
303      5
Name: average_montly_hours, Length: 215, dtype: int64
+++++ time_spend_company ++++++++++++++
3     5190
2     2910
4     2005
5     1062
6      542
10     107
7       94
8       81
Name: time_spend_company, dtype: int64
+++++ Work_accident ++++++++++++++
0    10141
1     1850
Name: Work_accident, dtype: int64
+++++ left ++++++++++++++
0    10000
1     1991
Name: left, dtype: int64
+++++ promotion_last_5years ++++++++++++++
0    11788
1      203
Name: promotion_last_5years, dtype: int64
+++++ department ++++++++++++++
sales          3239
technical      2244
support        1821
IT              976
RandD           694
product_mng     686
marketing       673
accounting      621
hr              601
management      436
Name: department, dtype: int64
+++++ salary ++++++++++++++
low       5740
medium    5261
high       990
Name: salary, dtype: int64
```

In [19]:

```
df.columns
```

Out[19]:

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'department', 'salary'],
      dtype='object')
```

In [20]:

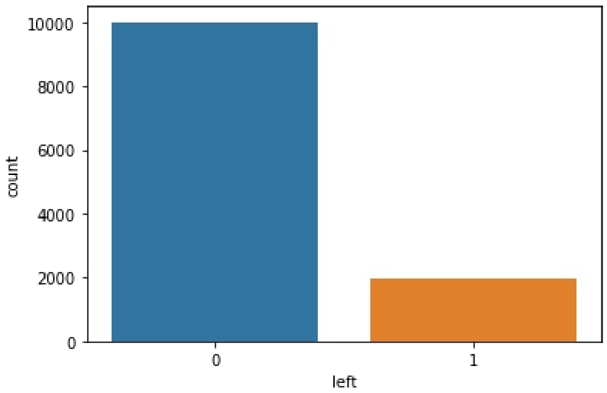
```
col_cat = ['number_project', 'time_spend_company', 'Work_accident', 'left', 'promotion_last_5years', 'department'
```

In [21]:

```
sns.countplot(x='left', data=df)
df.loc[:, 'left'].value_counts()
```

Out[21]:

```
0    10000
1     1991
Name: left, dtype: int64
```

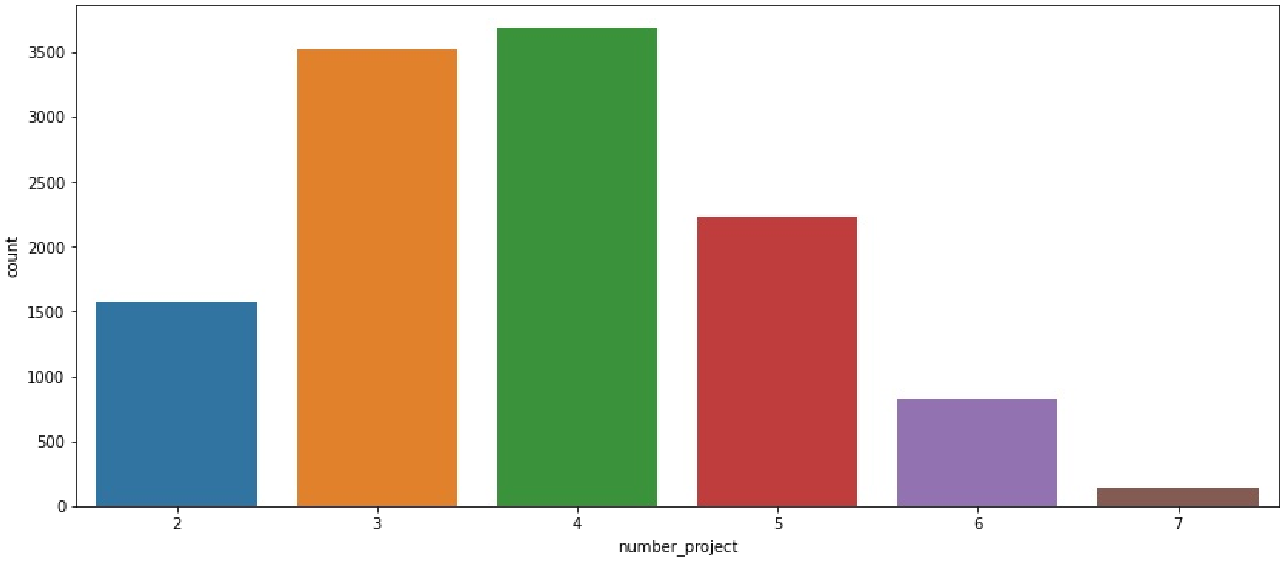


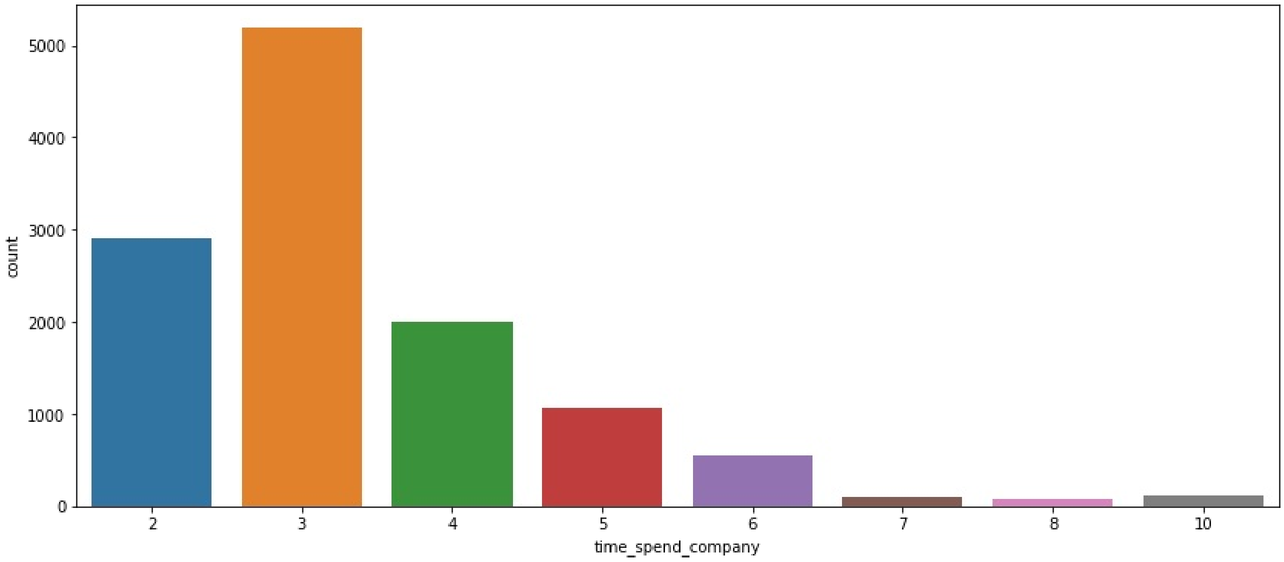
In [22]:

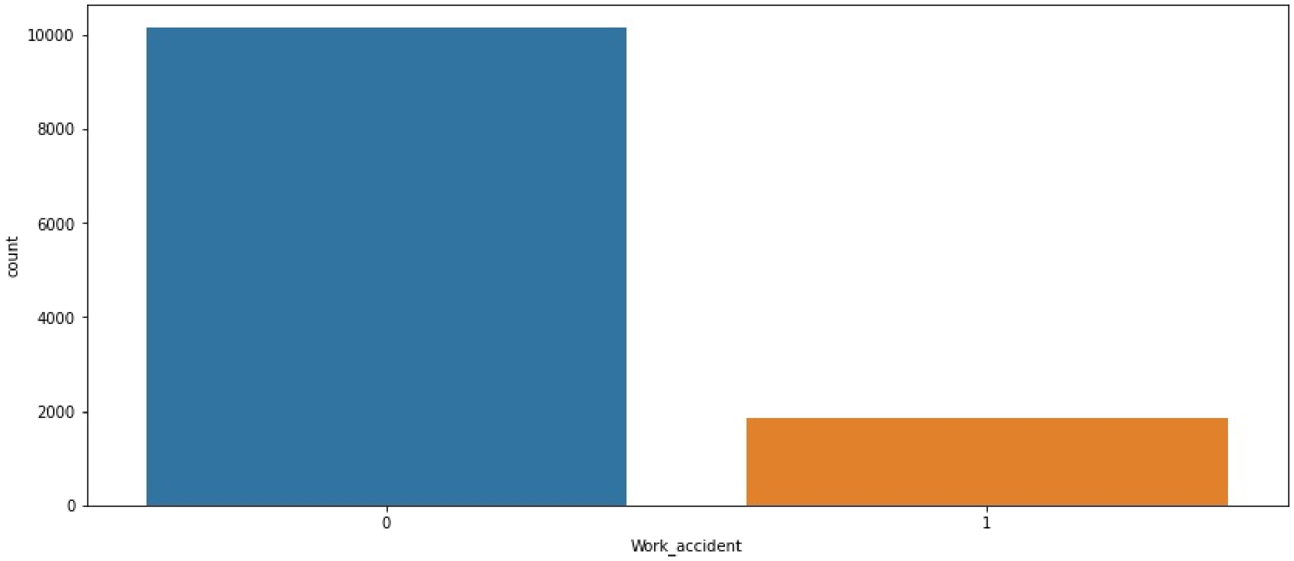
```
# from pylab import rcParams
rcParams["figure.figsize"] = 14, 6

def histplot(df, col_cat):
    for i in col_cat:
        sns.countplot(x=i, data=df)
        df.loc[:, i].value_counts()
        plt.show()
```

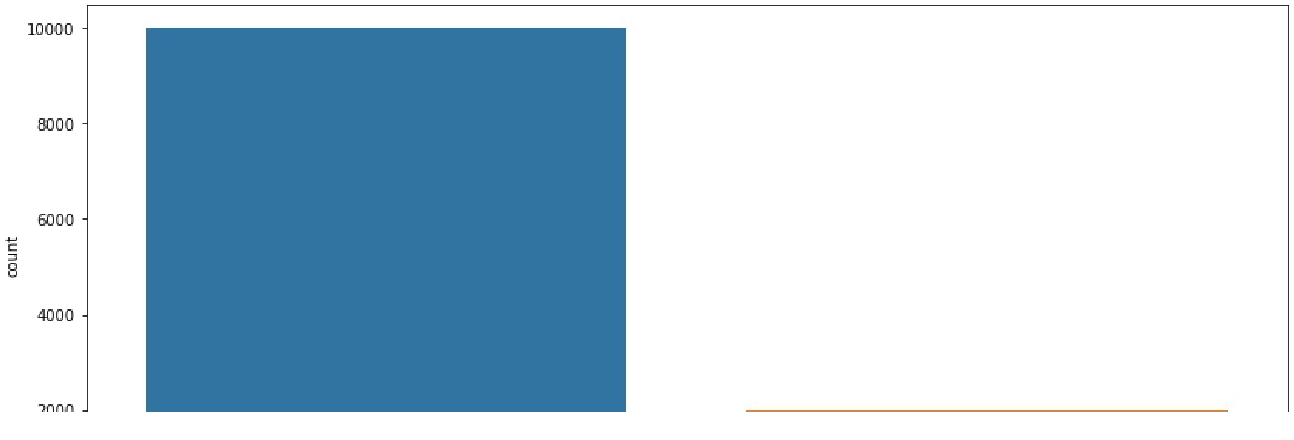
In [23]:

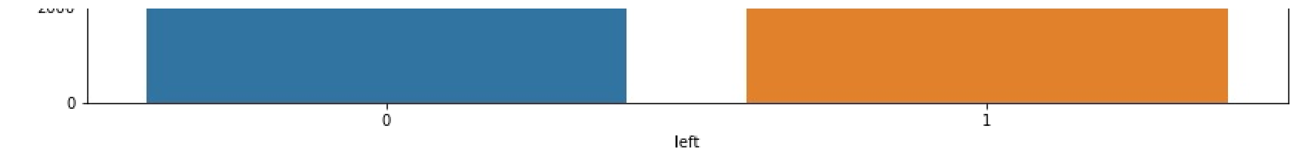
```
histplot(df, col_cat)
```

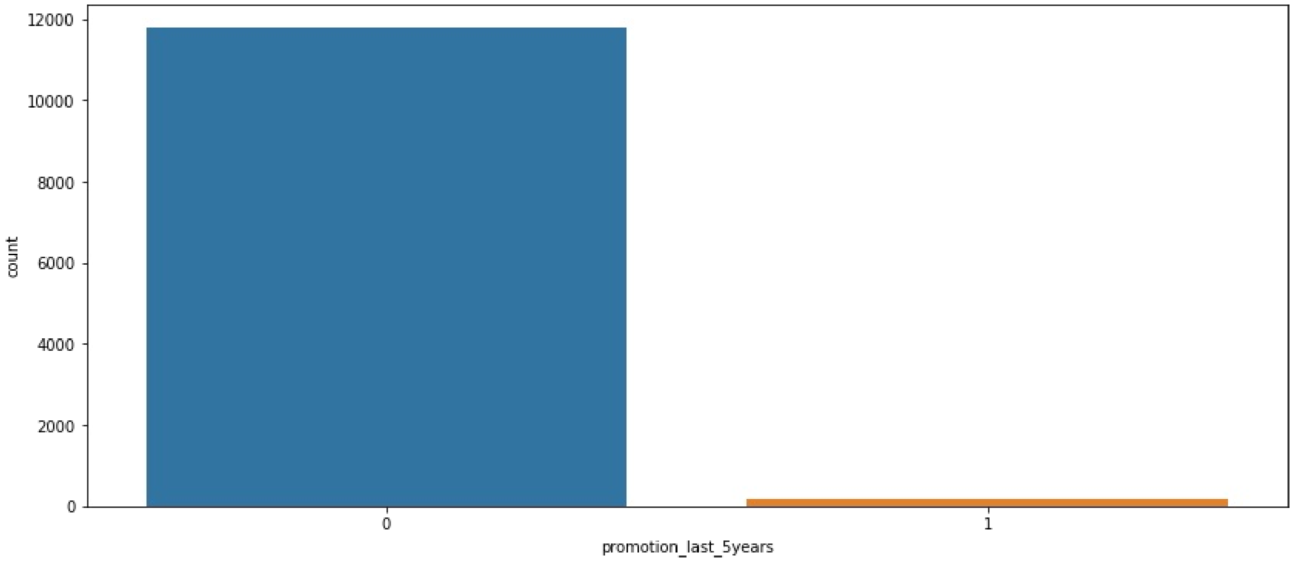

count
0
500
1000
1500
2000
2500
3000
3500
2
3
4
5
6
7
number_project

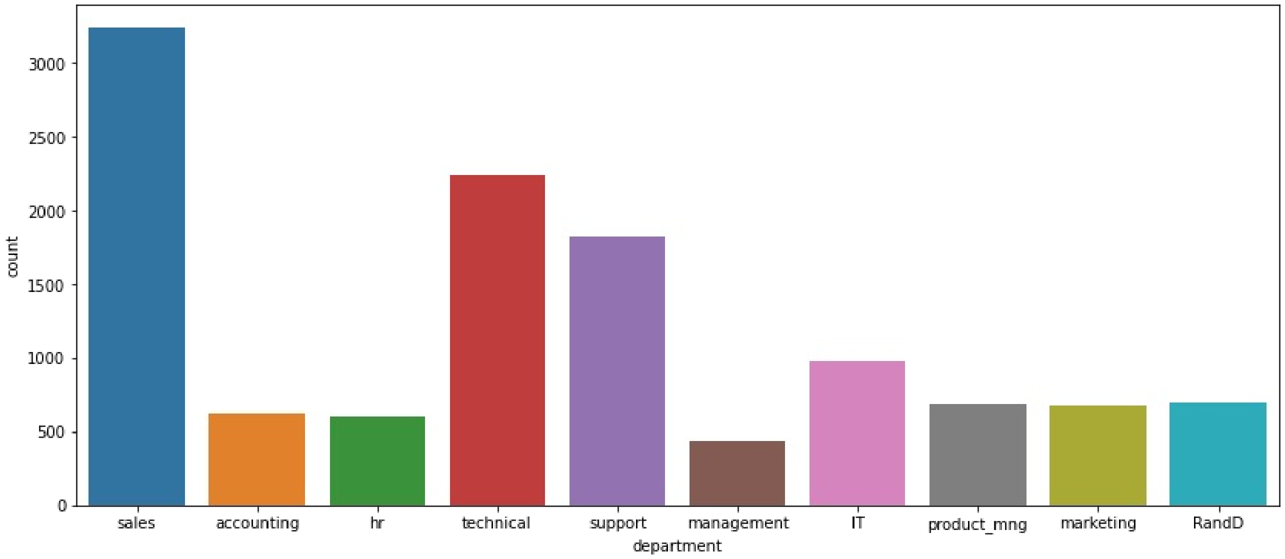

count
0
1000
2000
3000
4000
5000
2
3
4
5
6
7
8
10
time_spend_company

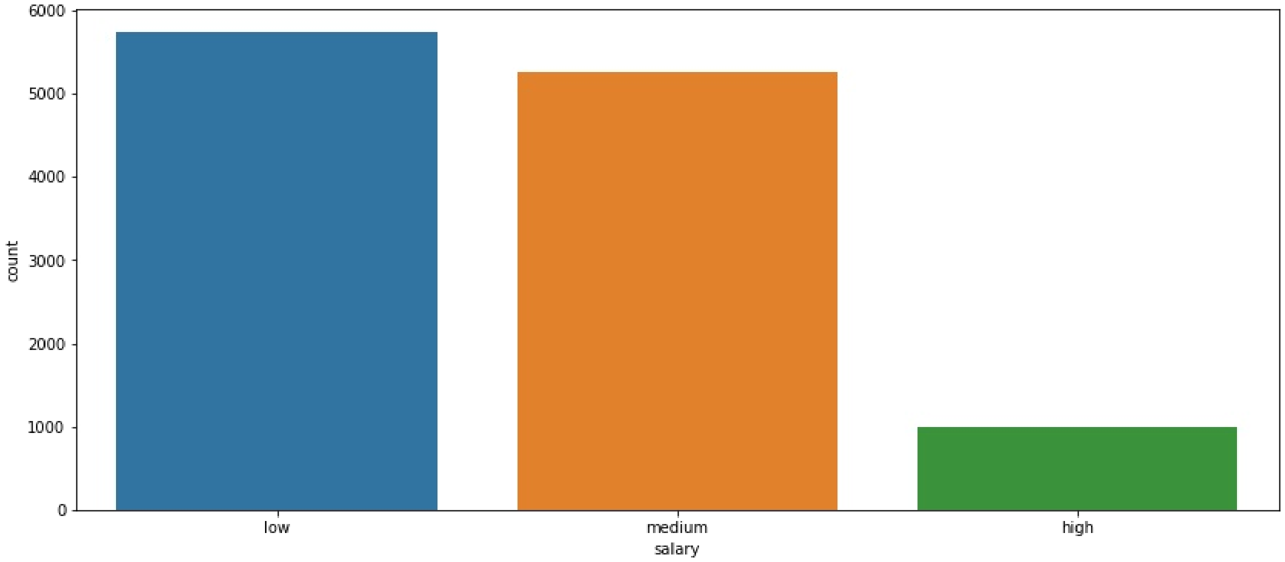

count
0
2000
4000
6000
8000
10000
0
1
Work_accident


count
2000
4000
6000
8000
10000









In [24]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11991 entries, 0 to 11990
```

```
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   satisfaction_level     11991 non-null  float64
 1   last_evaluation        11991 non-null  float64
 2   number_project         11991 non-null  int64  
 3   average_montly_hours   11991 non-null  int64  
 4   time_spend_company     11991 non-null  int64  
 5   Work_accident          11991 non-null  int64  
 6   left                   11991 non-null  int64  
 7   promotion_last_5years  11991 non-null  int64  
 8   department             11991 non-null  object 
 9   salary                 11991 non-null  object 
dtypes: float64(2), int64(6), object(2)
memory usage: 936.9+ KB
```

In [25]:

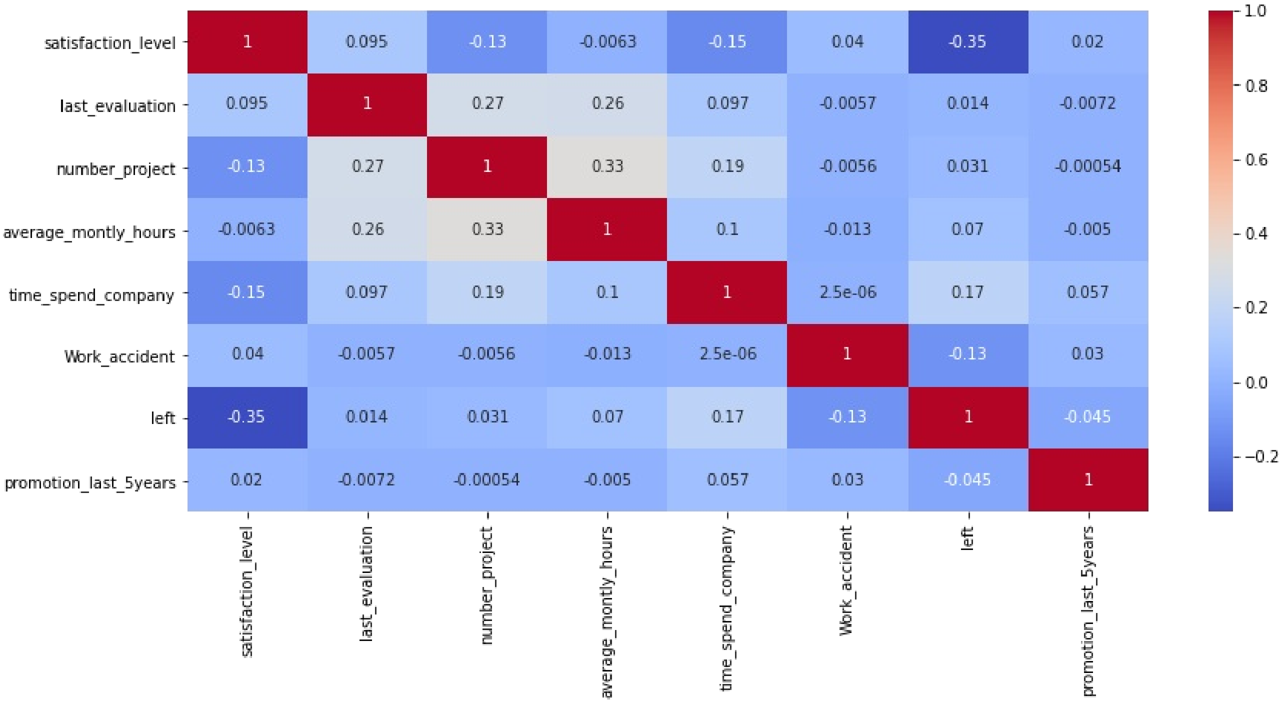
```
df.isnull().sum().sum()
```

Out[25]:

```
0
```

In [26]:

```
corr = df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm');
```



In [27]:

```
df.corr('spearman')
```

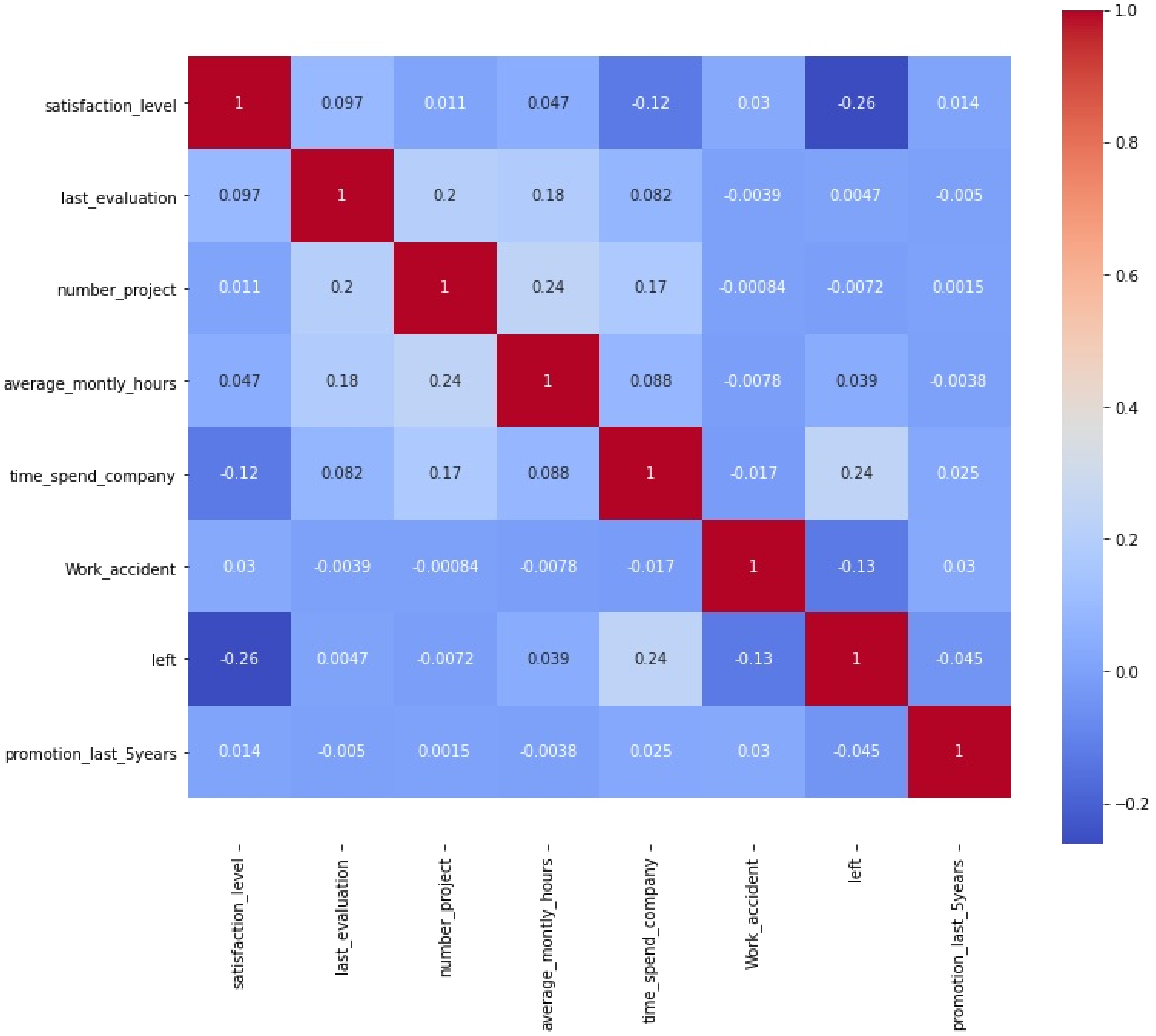
Out[27]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	lef
satisfaction_level	1.000000	0.139972	-0.000679	0.061647	-0.162049	0.036668	-0.31843
last_evaluation	0.139972	1.000000	0.267199	0.265949	0.110306	-0.004792	0.00576
number_project	-0.000679	0.267199	1.000000	0.310578	0.214167	-0.000930	-0.00800
average_montly_hours	0.061647	0.265949	0.310578	1.000000	0.122229	-0.009513	0.04763
time_spend_company	-0.162049	0.110306	0.214167	0.122229	1.000000	-0.019088	0.25935
Work_accident	0.036668	-0.004792	-0.000930	-0.009513	-0.019088	1.000000	-0.12543
left	-0.318436	0.005765	-0.008000	0.047631	0.259352	-0.125436	1.00000
promotion_last_5years	0.016499	-0.006012	0.001616	-0.004631	0.027375	0.029852	-0.04465

In [28]:

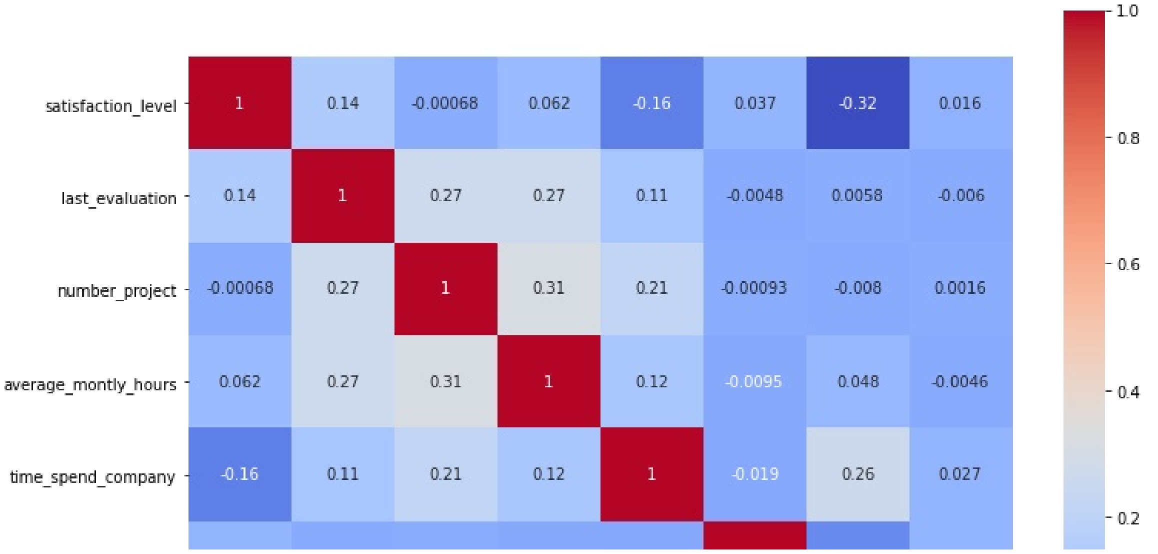
```
corr = df.corr('kendall')

plt.figure(figsize=(12, 10))
sns.heatmap(corr, annot=True, cmap='coolwarm')
```

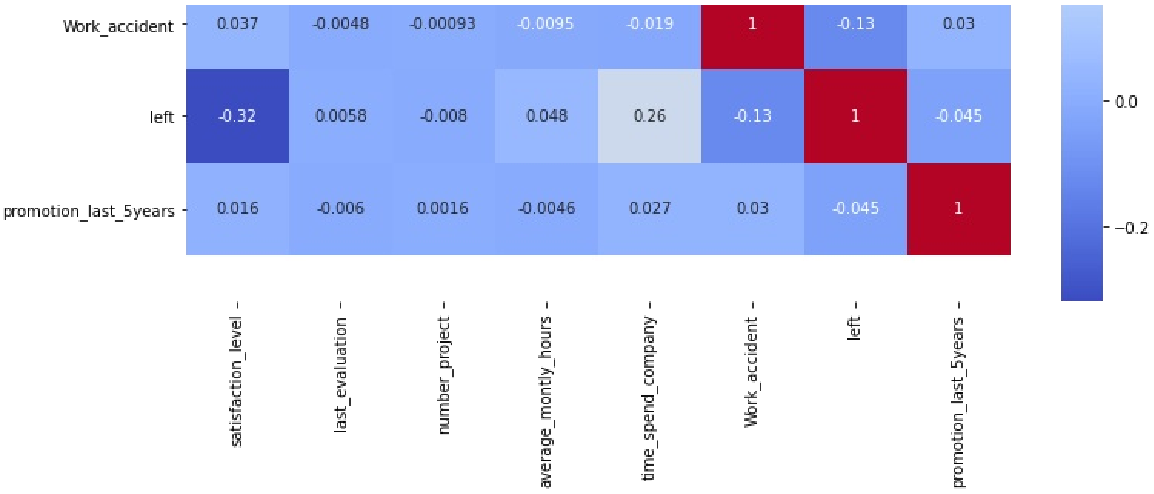
```python
plt.xticks(rotation=90)

b, t = plt.ylim()
b += 0.5
t -= 0.5
plt.ylim(b, t)
plt.show()
```



In [29]:

```python
corr = df.corr(method='spearman')

plt.figure(figsize=(12, 10))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.xticks(rotation=90)

b, t = plt.ylim()
b += 0.5
t -= 0.5
plt.ylim(b, t)
plt.show()
```





In [30]:

```python
from sklearn.feature_selection import chi2
```

In [31]:

```python
df[col_cat]
```

Out[31]:

	number_project	time_spend_company	Work_accident	left	promotion_last_5years	department	salary
0	2	3	0	1	0	sales	low
1	5	6	0	1	0	sales	medium
2	7	4	0	1	0	sales	medium
3	5	5	0	1	0	sales	low
4	2	3	0	1	0	sales	low
...	...	...	...	...	...	...	...
11986	3	10	1	0	1	management	high
11987	5	10	0	0	1	management	high
11988	3	10	0	0	1	management	high
11989	3	10	0	0	1	marketing	high
11990	4	3	0	0	0	IT	low

11991 rows × 7 columns

In [32]:

```python
x = df[col_cat].drop(['left', 'salary', 'department'] , axis=1)
y = df[col_cat].pop('left')
```

In [33]:

```python
x
```

Out[33]:

	number_project	time_spend_company	Work_accident	promotion_last_5years
0	2	3	0	0
1	5	6	0	0
2	7	4	0	0
3	5	5	0	0
4	2	3	0	0
...	...	...	...	...
11986	3	10	1	1
11987	5	10	0	1
11988	3	10	0	1
11989	3	10	0	1
11990	4	3	0	0

11991 rows × 4 columns

In [34]:

```python
y
```

Out[34]:
```
0        1
1        1
2        1
3        1
4        1
        ..
11986    0
11987    0
11988    0
11989    0
11990    0
Name: left, Length: 11991, dtype: int64
```

In [35]:
```
chi_scores = chi2(x, y)
chi_scores
```

Out[35]:
```
(array([  4.0807551 , 189.35860178, 159.56143856,  23.50849266]),
 array([4.33742713e-02, 4.38869734e-43, 1.41081324e-36, 1.24363595e-06]))
```

In [36]:
```
p_values = pd.Series(chi_scores[1],index = x.columns)
p_values.sort_values(ascending = False , inplace = True)
p_values
```

Out[36]:
```
number_project           4.337427e-02
promotion_last_5years    1.243636e-06
Work_accident            1.410813e-36
time_spend_company       4.388697e-43
dtype: float64
```

In [37]:
```
from sklearn.feature_selection import SelectKBest
test = SelectKBest(score_func=chi2, k=4)
```

In [38]:
```
fit = test.fit(x, y)
fit.scores_
```

Out[38]:
```
array([  4.0807551 , 189.35860178, 159.56143856,  23.50849266])
```

In [39]:
```
fit.pvalues_
```

Out[39]:
```
array([4.33742713e-02, 4.38869734e-43, 1.41081324e-36, 1.24363595e-06])
```

In [40]:
```
# import dtale

# dtale.show(data)
```

تبدیل داده کیفی به کمی :

In [41]:
```
df_c = df.copy()
```

In [42]:
```
def my_dummies(dataFrame, col_name):
    temp = pd.get_dummies(dataFrame[col_name], drop_first = True)
    dataFrame = pd.concat([dataFrame, temp], axis = 1)
    dataFrame.drop([col_name], axis = 1, inplace = True)
    return dataFrame
```

In [43]:
```
df_c = my_dummies(df_c, 'department')
```

In [44]:
```
df_c['salary'].value_counts()
```

Out[44]:
```
low       5740
```

```
Out[44]: low       5740
         medium    5261
         high       990
         Name: salary, dtype: int64
```

In [45]:

```
df_c['salary'] = df_c['salary'].map({'low' : 0, 'medium' : 1, 'high':2})
```

In [46]:

```
df_c
```

Out[46]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	
11989	0.33	0.65	3	172	10	0	0	
11990	0.50	0.73	4	180	3	0	0	

11991 rows × 18 columns

A little EDA

In [47]:

```
df_eda_encode = df_c.copy()
```

In [48]:

```
df_eda_encode
```

Out[48]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	
11989	0.33	0.65	3	172	10	0	0	
11990	0.50	0.73	4	180	3	0	0	

11991 rows × 18 columns

In [49]:

```
df_eda_encode.dtypes
```

Out[49]:

```
satisfaction_level       float64
last_evaluation          float64
number_project             int64
average_montly_hours       int64
time_spend_company         int64
Work_accident              int64
left                       int64
promotion_last_5years      int64
salary                     int64
RandD                      uint8
```

```
accounting                 uint8
hr                         uint8
management                 uint8
marketing                  uint8
product_mng                uint8
sales                      uint8
support                    uint8
technical                  uint8
dtype: object
```

In [50]:

```
# import dtale

# dtale.show(df_eda_encode)
```

In [51]:

```
df_eda = df.copy()
```

just for EDA - using ordinal for department field

In [52]:

```
df_eda
```

Out[52]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	
11989	0.33	0.65	3	172	10	0	0	
11990	0.50	0.73	4	180	3	0	0	

11991 rows × 10 columns

In [53]:

```
# df_eda['salary'] = df_eda['salary'].map({'low' : 0, 'medium' : 1, 'high':2})
```

In [54]:

```
df_eda['department'].value_counts()
```

Out[54]:

```
sales          3239
technical      2244
support        1821
IT              976
RandD           694
product_mng     686
marketing       673
accounting      621
hr              601
management      436
Name: department, dtype: int64
```

In [55]:

```
# df_eda['department'] = df_eda['department'].map({'sales':0, 'technical':1, 'support':2, 'IT':3, 'product_mng':4
#                                                  'marketing':5, 'RandD':6, 'accounting':7, 'hr':8, 'management':
```

In [56]:

```
df_eda
```

Out[56]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	

3	0.72	0.87	5	223	5	0	1
4	0.37	0.52	2	159	3	0	1
...	...	...	...	...	...	...	...
11986	0.90	0.55	3	259	10	1	0
11987	0.74	0.95	5	266	10	0	0
11988	0.85	0.54	3	185	10	0	0
11989	0.33	0.65	3	172	10	0	0
11990	0.50	0.73	4	180	3	0	0

11991 rows × 10 columns

In [57]:

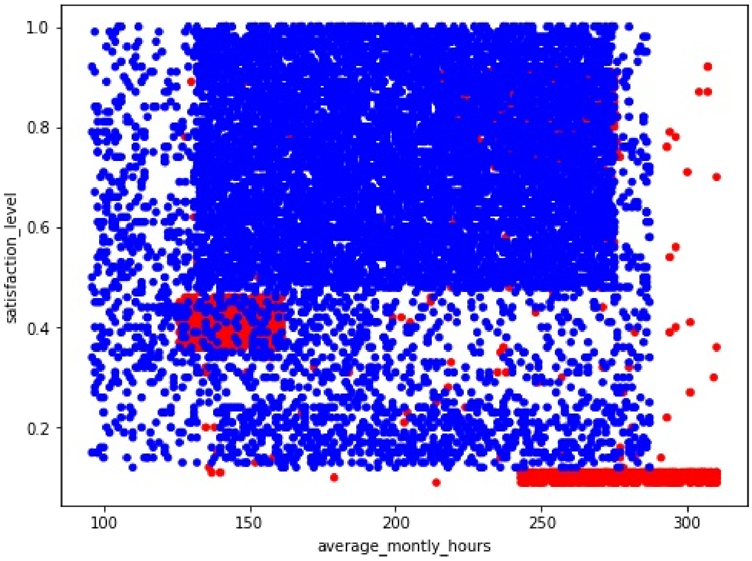
```
df_eda['department'].value_counts()
```

Out[57]:

```
sales          3239
technical      2244
support        1821
IT              976
RandD           694
product_mng     686
marketing       673
accounting      621
hr              601
management      436
Name: department, dtype: int64
```

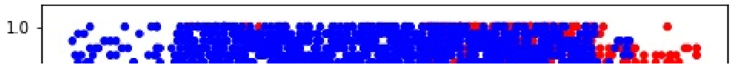
In [58]:

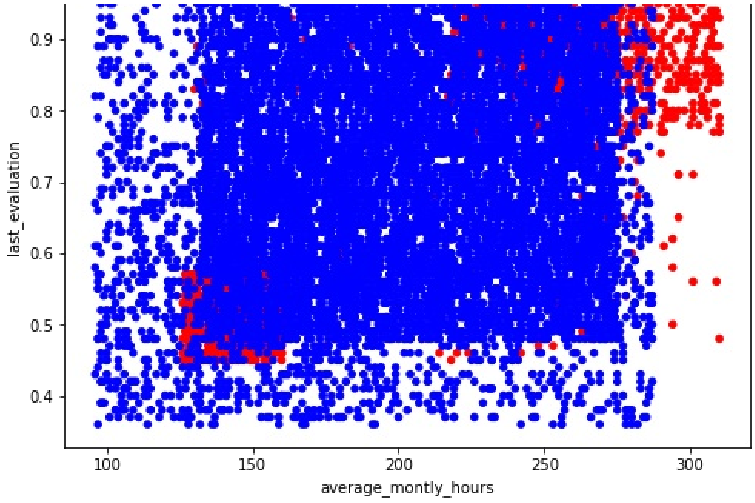
```
rcParams["figure.figsize"] = 8, 6
df_eda['color'] = df['left'] #□□□□□ □□□□□□ □□□□□ □□□□□ □□


df_eda.color[df.left == 0] ='b'
df_eda.color[df.left == 1] ='r'

df_eda.plot.scatter(x='average_montly_hours',y = 'satisfaction_level', c = df_eda['color'] )

plt.savefig('peiman.png', format = 'png')
```



In [59]:

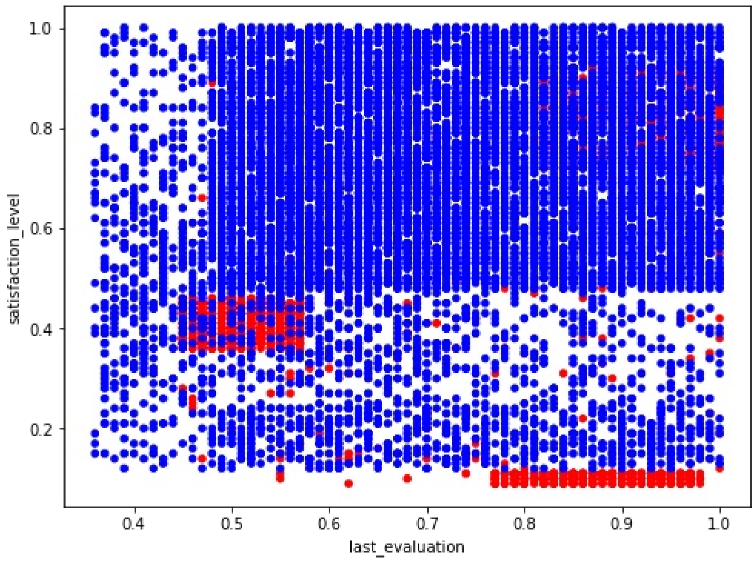
```
rcParams["figure.figsize"] = 8, 6
df_eda['color'] = df['left'] #□□□□□ □□□□□□ □□□□□ □□□□□ □□


df_eda.color[df.left == 0] ='b'
df_eda.color[df.left == 1] ='r'

df_eda.plot.scatter(x='average_montly_hours',y = 'last_evaluation', c = df_eda['color'] )

plt.savefig('peiman.png', format = 'png')
```





In [60]:

```
rcParams["figure.figsize"] = 8, 6
df_eda['color'] = df['left'] #□□□□ □□□□□ □□□□ □□□□ □□


df_eda.color[df.left == 0] ='b'
df_eda.color[df.left == 1] ='r'

df_eda.plot.scatter(x='last_evaluation',y = 'satisfaction_level', c = df_eda['color'] )

plt.savefig('peiman.png', format = 'png')
```



In [61]:

```
df_eda
```

Out[61]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	
11989	0.33	0.65	3	172	10	0	0	
11990	0.50	0.73	4	180	3	0	0	

11991 rows × 11 columns

In [62]:

```
df_eda.columns
```

Out[62]:

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'department', 'salary', 'color'],
      dtype='object')
```

In [63]:

```
col_cat = ['number_project', 'time_spend_company', 'Work_accident', 'left', 'promotion_last_5years', 'salary']
```

```
==========================================================
```

خواسته های مسئله

تعداد افراد استخدام شده در بر اساس سطح حقوق پرداختی

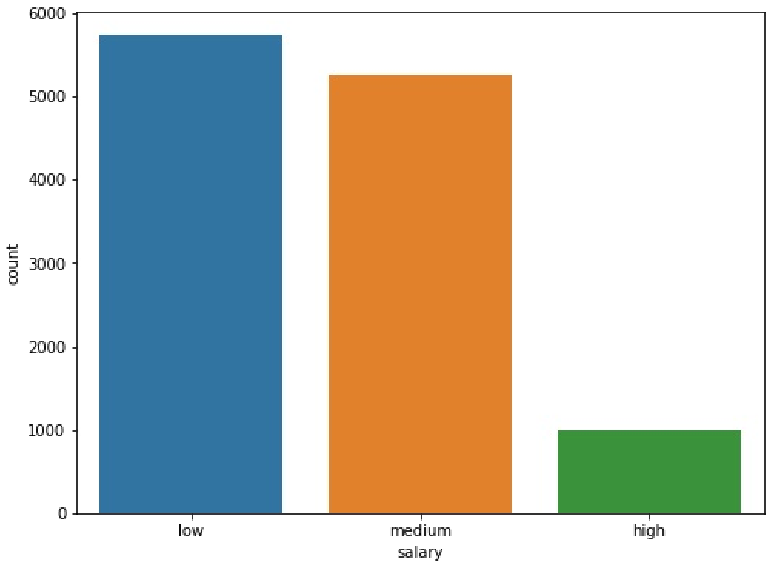
In [64]:

```
q1 = df_eda['salary'].value_counts()
print('تعداد افراد استخدام شده در بر اساس سطح حقوق پرداختی')
print(q1)
```

```
تعداد افراد استخدام شده در بر اساس سطح حقوق پرداختی
low       5740
medium    5261
high       990
Name: salary, dtype: int64
```

In [65]:

```
histplot(df_eda, ['salary'])
```



تعداد افراد استخدامی در هر دپارتمان

In [66]:

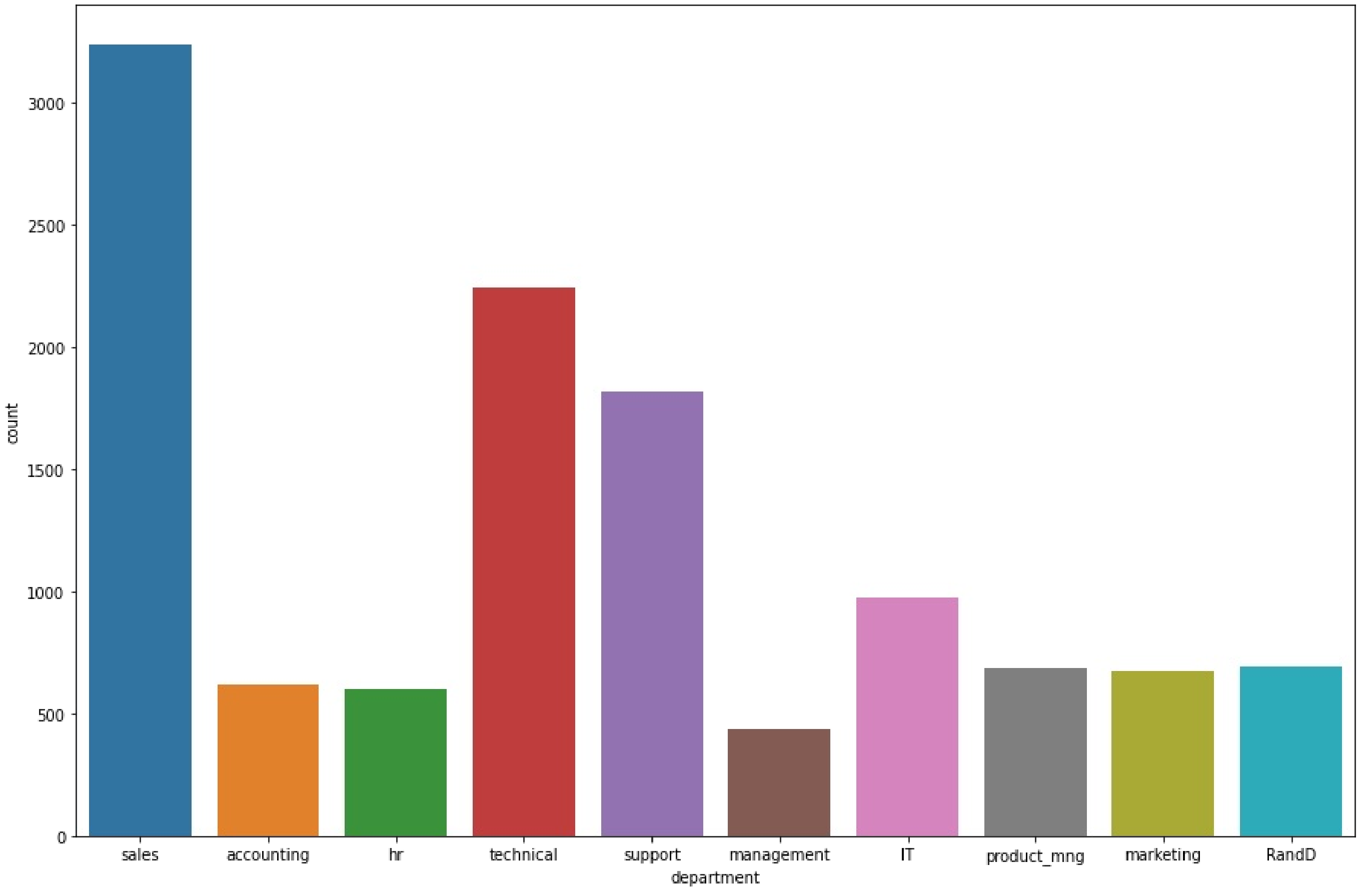
```
q2 = df_eda['department'].value_counts()
print('تعداد افراد استخدامی در هر دپارتمان')
print(q2)
```

```
تعداد افراد استخدامی در هر دپارتمان
sales          3239
technical      2244
support        1821
IT              976
RandD           694
product_mng     686
marketing       673
accounting      621
hr              601
management      436
```

```
Name: department, dtype: int64
```

In [67]:

```
rcParams["figure.figsize"] = 15, 10
histplot(df_eda, ['department'])
```



تعداد افراد استخدام شده بر اساس سطح حقوق و نوع دپارتمان (راهنمایی: Pivot Table)

In [68]:

```
Q3 = df.groupby(['department', 'salary']).agg({'department' : np.size})
Q3
```

Out[68]:

		department
department	**salary**	
IT	**high**	71
	low	476
	medium	429
RandD	**high**	47
	low	322
	medium	325
accounting	**high**	63
	low	296
	medium	262
hr	**high**	38
	low	296
	medium	267
management	**high**	128
	low	139
	medium	169
marketing	**high**	62
	low	310
	medium	301
product_mng	**high**	52

	low	343
	medium	291
sales	high	237
	low	1553
	medium	1449
support	high	126
	low	867
	medium	828
technical	high	166
	low	1138
	medium	940

In [69]:

```
Q3 = df.groupby(['salary', 'department']).agg({'salary' : np.size})
Q3
```

Out[69]:

		salary
salary	department	
high	IT	71
	RandD	47
	accounting	63
	hr	38
	management	128
	marketing	62
	product_mng	52
	sales	237
	support	126
	technical	166
low	IT	476
	RandD	322
	accounting	296
	hr	296
	management	139
	marketing	310
	product_mng	343
	sales	1553
	support	867
	technical	1138
medium	IT	429
	RandD	325
	accounting	262
	hr	267
	management	169
	marketing	301
	product_mng	291
	sales	1449
	support	828
	technical	940

In [70]:

```
df_eda
```

Out[70]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	

2	0.11	0.88	7	272	4	0	1
3	0.72	0.87	5	223	5	0	1
4	0.37	0.52	2	159	3	0	1
...	...	...	...	...	...	...	...
11986	0.90	0.55	3	259	10	1	0
11987	0.74	0.95	5	266	10	0	0
11988	0.85	0.54	3	185	10	0	0
11989	0.33	0.65	3	172	10	0	0
11990	0.50	0.73	4	180	3	0	0

11991 rows × 11 columns

```
In [71]: df_eda.columns
```

```
Out[71]: Index(['satisfaction_level', 'last_evaluation', 'number_project',
                'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
                'promotion_last_5years', 'department', 'salary', 'color'],
               dtype='object')
```

```
In [72]: df_q3 = df_eda.loc[:, ['salary', 'department'] ]
```

```
In [73]: df_q3
```

Out[73]:

	salary	department
0	low	sales
1	medium	sales
2	medium	sales
3	low	sales
4	low	sales
...	...	...
11986	high	management
11987	high	management
11988	high	management
11989	high	marketing
11990	low	IT

11991 rows × 2 columns

```
In [74]: low_salary = df_q3[df_q3['salary']==0].groupby('department').count()
         medium_salary = df_q3[df_q3['salary']==1].groupby('department').count()
         high_salary = df_q3[df_q3['salary']==2].groupby('department').count()
```

```
In [75]: low_salary
```

Out[75]:

	salary
department	

```
In [76]: salary_department = pd.merge(low_salary, medium_salary, on='department', suffixes=('_low', '_medium'))
         salary_department = pd.merge(salary_department, high_salary, on='department', suffixes=('_medium', '_high'))
```

```
In [77]: salary_department
```

Out[77]:

	salary_low	salary_medium	salary
department			

In [78]:

```
salary_department = salary_department.rename(columns = {'low_salary' : 'low', 'salary_medium': 'medium', 'salary'
```

In [79]:

```
salary_department
```

Out[79]:

	salary_low	medium	high
department			

In [80]:

```
salary_department.sum().sum()
```

Out[80]:

```
0.0
```

بیشترین تعداد پروژه ای که پرسنل دچار ریزش داشته اند بر اساس سطح رضایتمندی

In [81]:

```
df.columns
```

Out[81]:

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'department', 'salary'],
      dtype='object')
```

In [82]:

```
left = df_eda[df_eda['left']==1]
```

In [83]:

```
Q4 = left.groupby(['satisfaction_level']).agg({'number_project': np.max})
Q4
```

Out[83]:

	number_project
satisfaction_level	
0.09	7
0.10	7
0.11	7
0.12	5
0.13	6
...	...
0.88	5
0.89	5
0.90	5
0.91	5
0.92	5

81 rows × 1 columns

In [84]:

```
left
```

Out[84]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5year
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	.
1986	0.37	0.57	2	147	3	0	1	
1987	0.11	0.92	7	293	4	0	1	

1988	0.41	0.53	2	157	3	0	1
1989	0.84	0.96	4	247	5	0	1
1990	0.40	0.51	2	148	3	0	1

1991 rows × 11 columns

In [85]:

```
left.columns
```

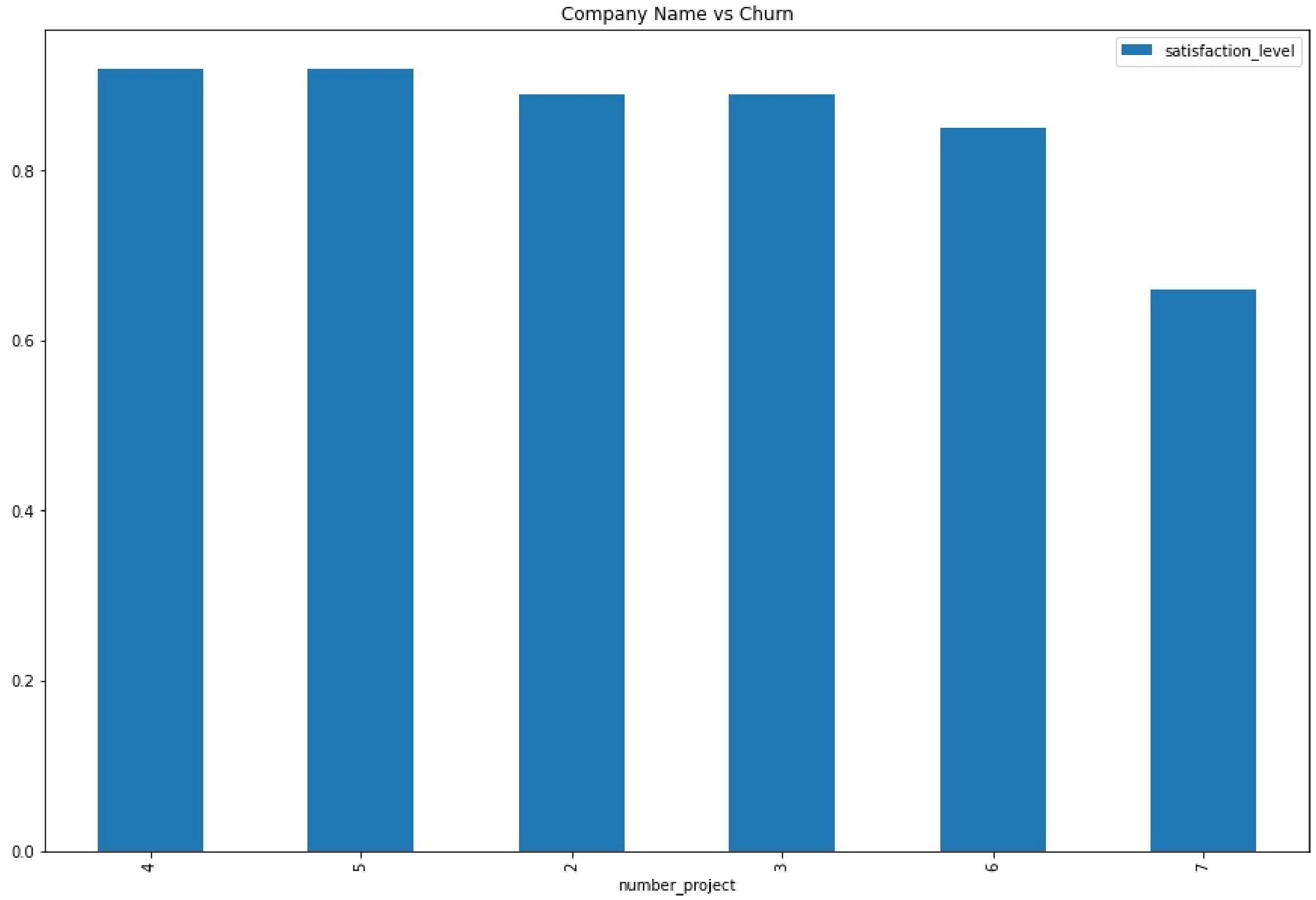
Out[85]:

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'department', 'salary', 'color'],
      dtype='object')
```

In [86]:

```
plt.figure(figsize=(25, 6))

left2 = pd.DataFrame(left.groupby(['number_project'])['satisfaction_level'].max().sort_values(ascending = False))
left2.plot.bar()
plt.title('Company Name vs Churn')
plt.show()
```

```
<Figure size 1800x432 with 0 Axes>
```



اخرین وضعیت ارزیابی پرسنلی که در شرکت دچار ریزش نشدند بصورت نزولی

In [87]:

```
stay = df_eda[df_eda['left']==0]
```

In [88]:

```
stay.sort_values('last_evaluation', ascending=True)
```

Out[88]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
6560	0.62	0.36	2	137	4	1	0	
6728	0.83	0.36	4	242	3	0	0	
3492	0.65	0.36	2	282	3	0	0	

10421	0.71	0.36	2	132	5	0	0
5621	0.40	0.36	4	128	4	0	0
...	...	...	...	...	...	...	...
6476	0.63	1.00	5	241	4	0	0
8005	0.88	1.00	5	190	2	0	0
6870	0.40	1.00	6	206	2	0	0
8226	0.56	1.00	3	141	2	1	0
6890	0.86	1.00	3	166	3	1	0

10000 rows × 11 columns

تاثیرگذارترین مولفه بر روی تارگت مسئله بر اساس تشخیص همبستگی

in all different kind solved and there are a lot of this problem in continue

I used chi2, kendal, spearman, pearson & MI(Muture Information) correlation in this notebook

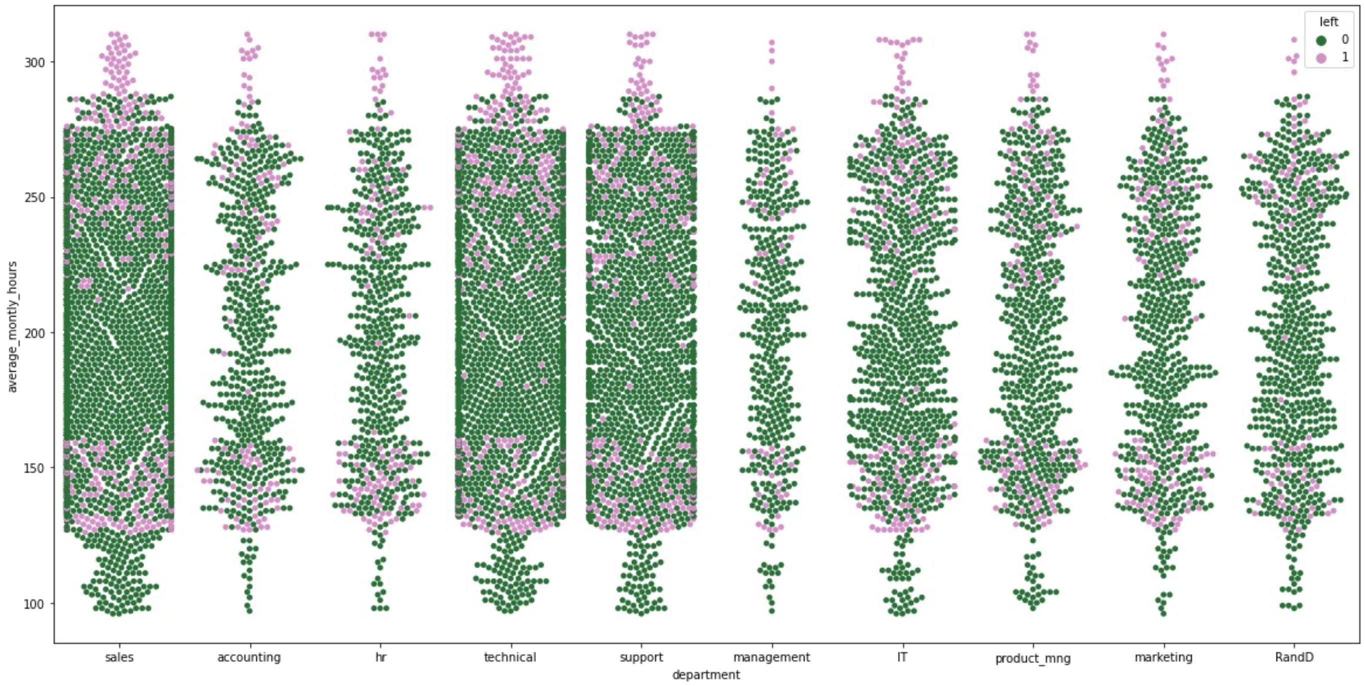
ریزش پرسنل بر اساس نوع دپارتمان و میانگین زمان حضور در سازمانSwarm( ) تحلیل نمودار ازدحامی

In [188…

```
sns.swarmplot(x=df_eda.department , y=df_eda.average_montly_hours, hue=df_eda.left, data=df_eda, palette=("cubehe
```

Out[188…

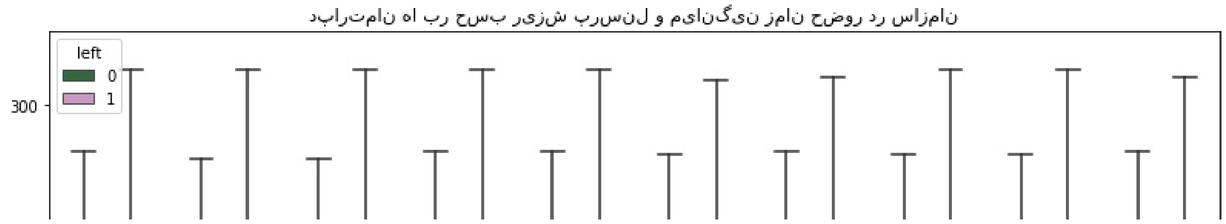
```
<AxesSubplot:xlabel='department', ylabel='average_montly_hours'>
```

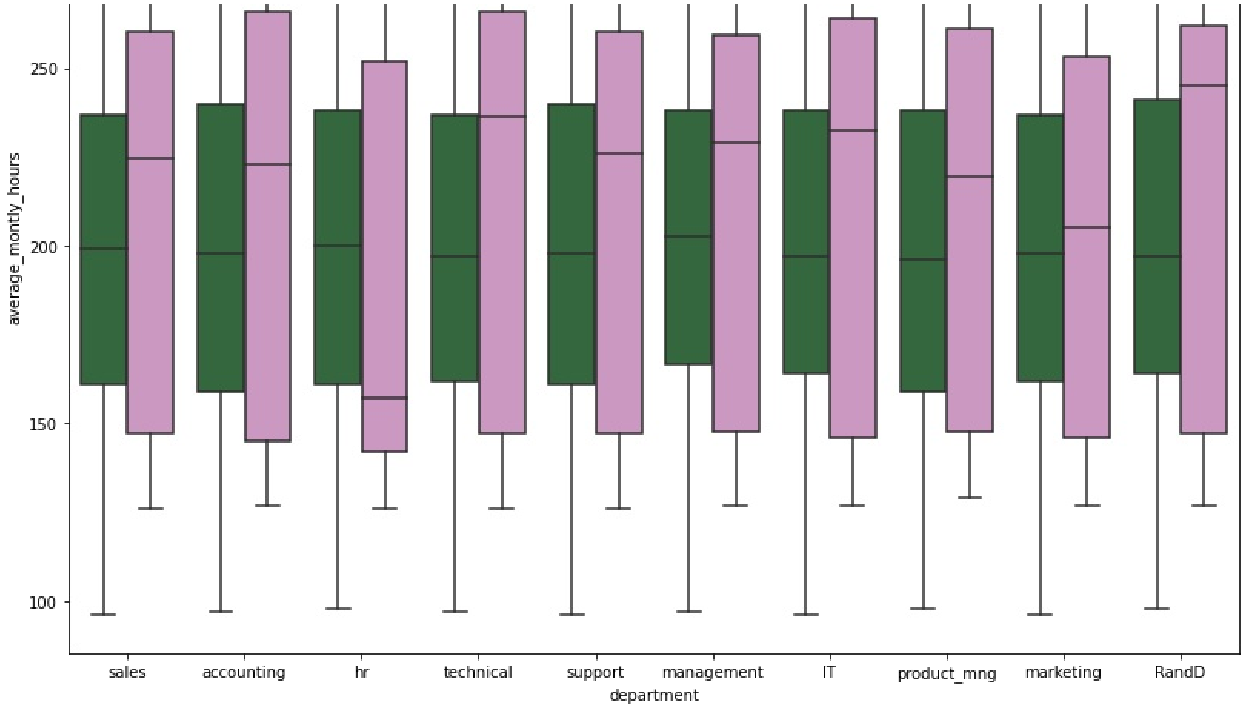


نمودار جعبه ای تمامی دپارتمان ها بر حسب ریزش پرسنل و میانگین زمان حضور در سازمان در یک قاب

In [89]:

```
plt.figure(figsize=(30, 10))

plt.subplot(1,2,2)
plt.title('دپارتمان ها  بر حسب ریزش پرسنل و میانگین زمان حضور در سازمان')
sns.boxplot(x=df_eda.department, y=df_eda.average_montly_hours, hue=df_eda.left, palette=("cubehelix"))

plt.show()
```





In [90]:

```
df_eda
```

Out[90]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	
11989	0.33	0.65	3	172	10	0	0	
11990	0.50	0.73	4	180	3	0	0	

11991 rows × 11 columns

In [91]:

```
df_eda.columns
```

Out[91]:

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'department', 'salary', 'color'],
      dtype='object')
```

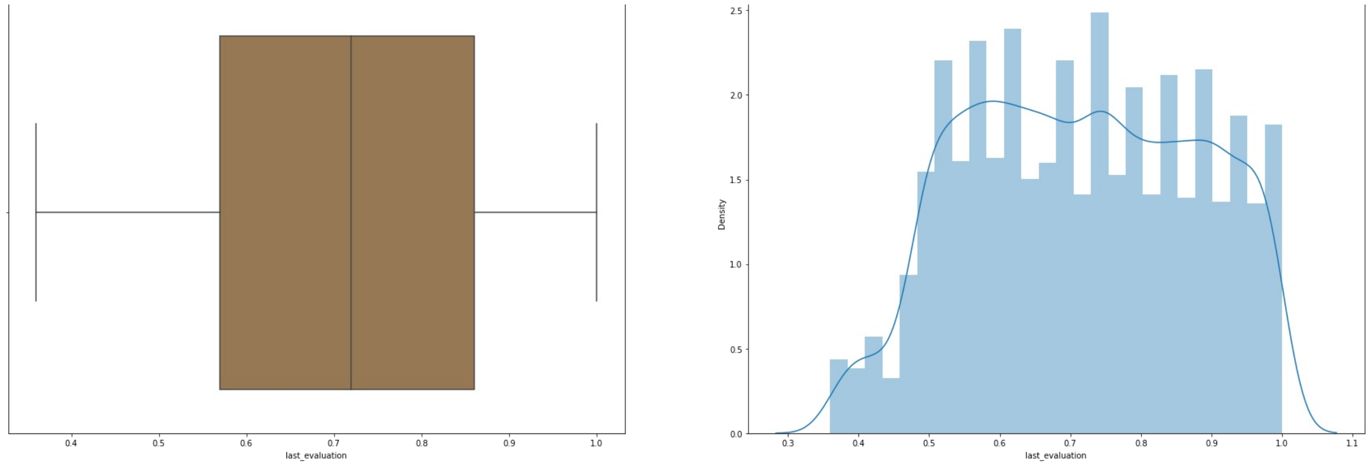
مقایسه توزیع و نمودار جعبه ای پرسنل وفادار سازمان بر اساس اخرین وضعیت ارزیابی افراد سازمان در یک قاب

In [92]:

```
stay = df_eda[df_eda['left']==0]
```

In [93]:

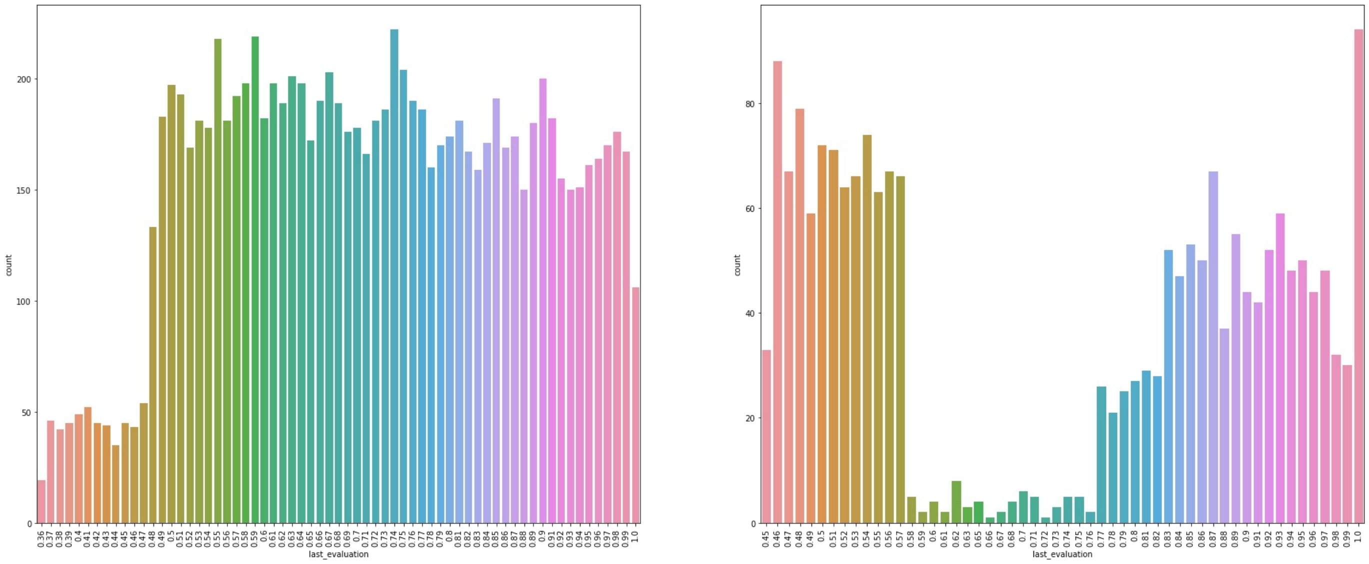
```
plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.title('پرسنل وفادار سازمان بر اساس اخرین وضعیت ارزیابی افراد سازمان')
sns.boxplot(x=df_eda.last_evaluation, palette=("cubehelix"))

plt.subplot(1, 2, 2)
pu.plot_single_distplot_fitWithNormalLine(stay, ['last_evaluation'])
```



مقایسه هیستوگرام اخرین وضعیت ارزیابی پرسنلی که دچار ریزش و پرسنلی که در شرکت فعال خواهند بود در یک قالب

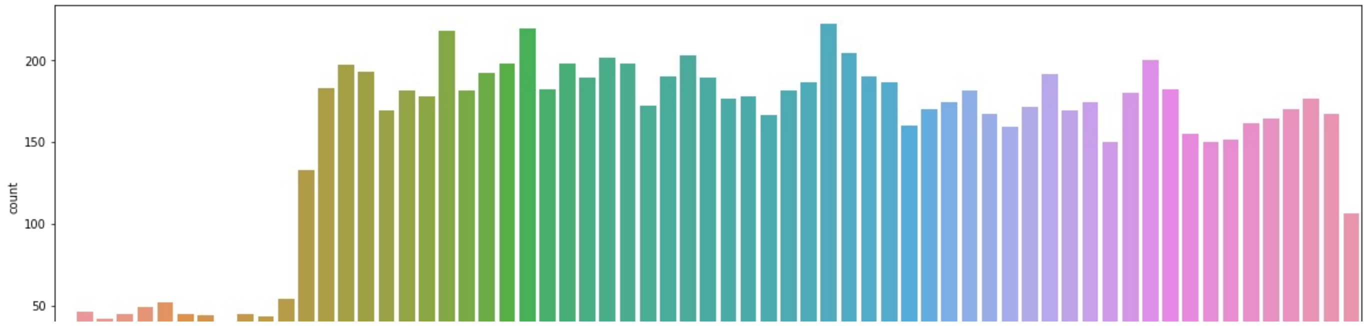
In [94]:
```
stay = df_eda[df_eda['left']==0]
left = df_eda[df_eda['left']==1]
```

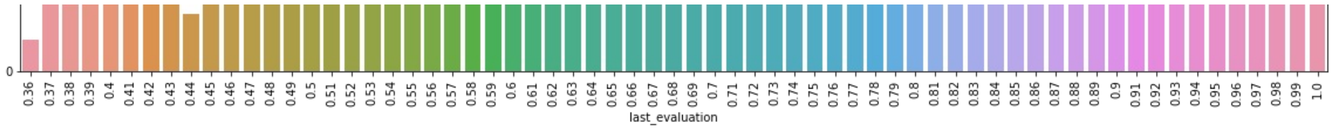
In [95]:
```
fig=plt.subplots(figsize=(30,12))

plt.subplot(1, 2, 1)
sns.countplot(x=stay.last_evaluation);
plt.xticks(rotation=90);

plt.subplot(1, 2, 2)
sns.countplot(x=left.last_evaluation);
plt.xticks(rotation=90);
```



In [96]:
```
def count_plot(y):
    fig=plt.subplots(figsize=(20,6));
    sns.countplot(x=y.last_evaluation);
    plt.xticks(rotation=90);
```

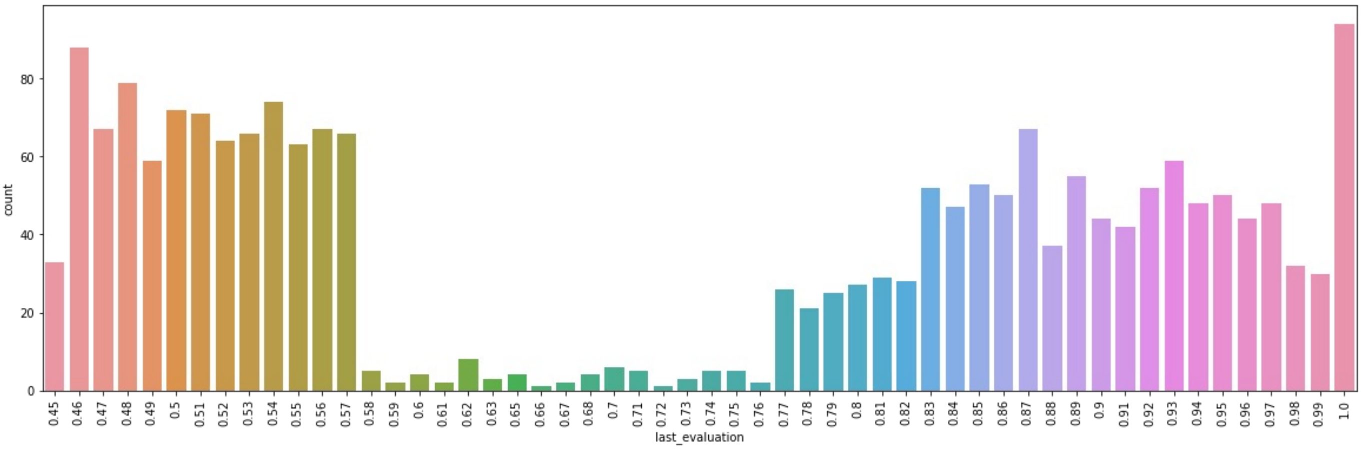
In [97]:
```
count_plot(stay)
```





In [98]:

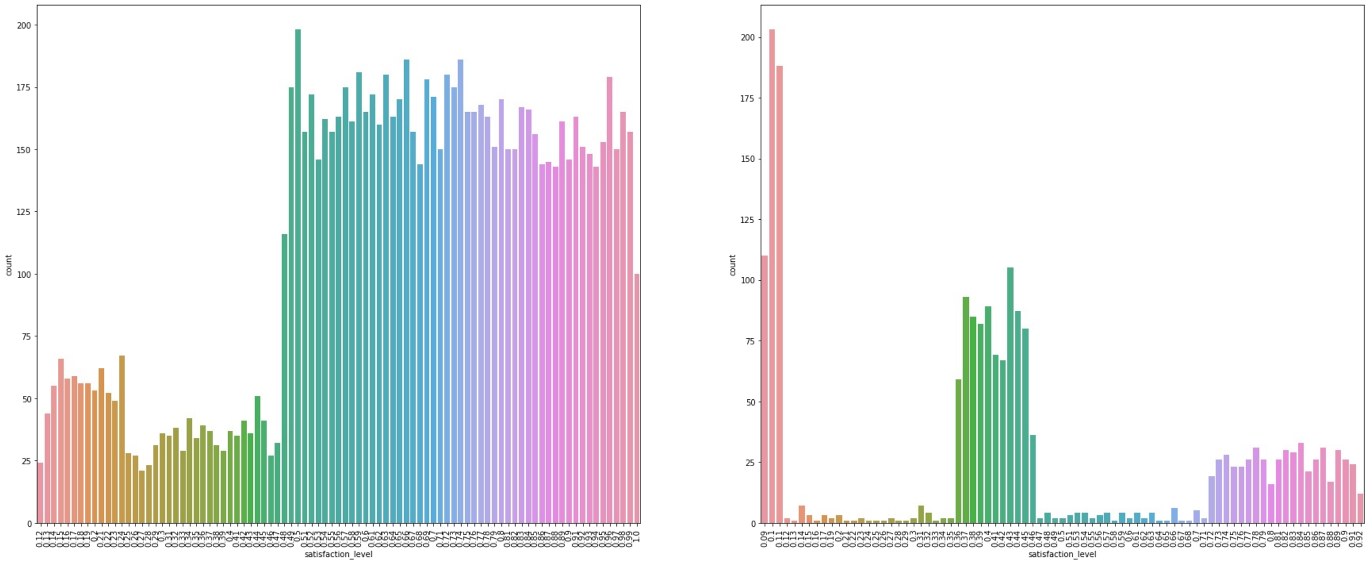
```
count_plot(left)
```



مقایسه هیستوگرام رضایتمندی پرسنلی که دچار ریزش و پرسنلی که در شرکت فعال خواهند بود در یک قالب

In [99]:

```
fig=plt.subplots(figsize=(30,12))

plt.subplot(1, 2, 1)
sns.countplot(x=stay.satisfaction_level);
plt.xticks(rotation=90);

plt.subplot(1, 2, 2)
sns.countplot(x=left.satisfaction_level);
plt.xticks(rotation=90);
```

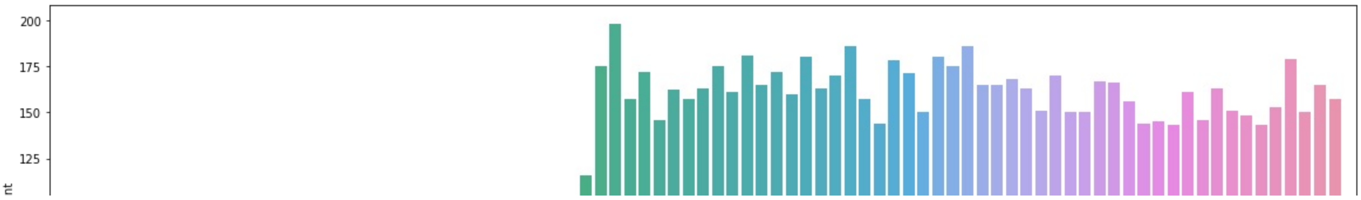


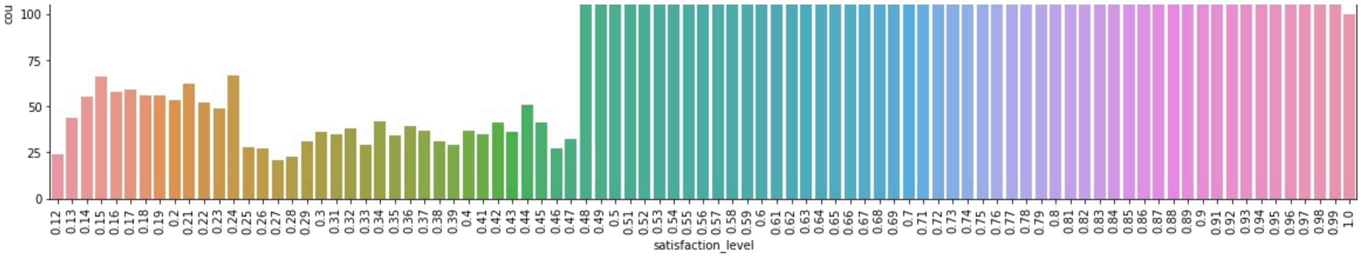
In [100…

```
def count_plot(y):
    fig=plt.subplots(figsize=(20,6));
    sns.countplot(x=y.satisfaction_level);
    plt.xticks(rotation=90);
```

In [101…

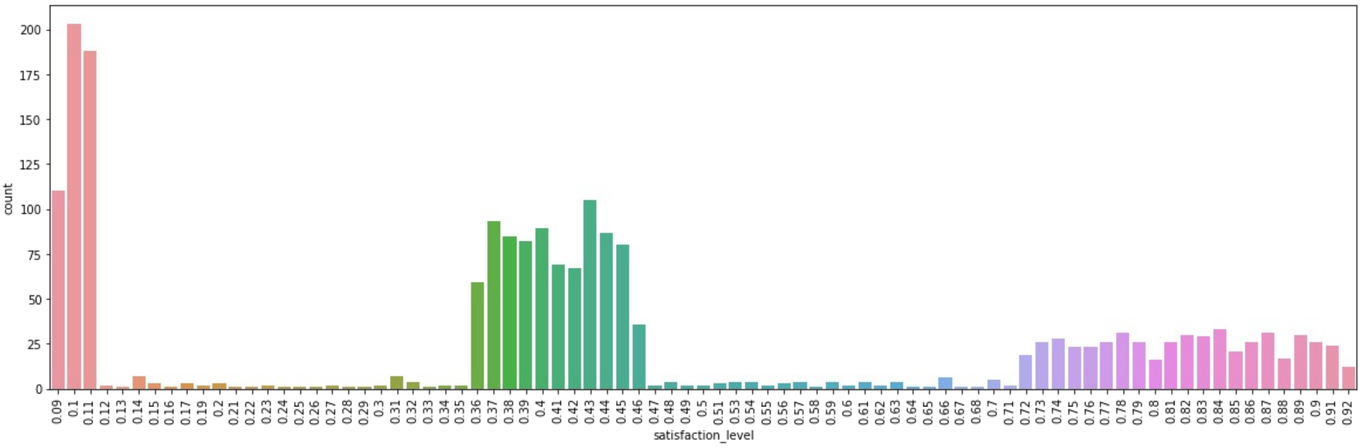
```
count_plot(stay)
```





In [102…

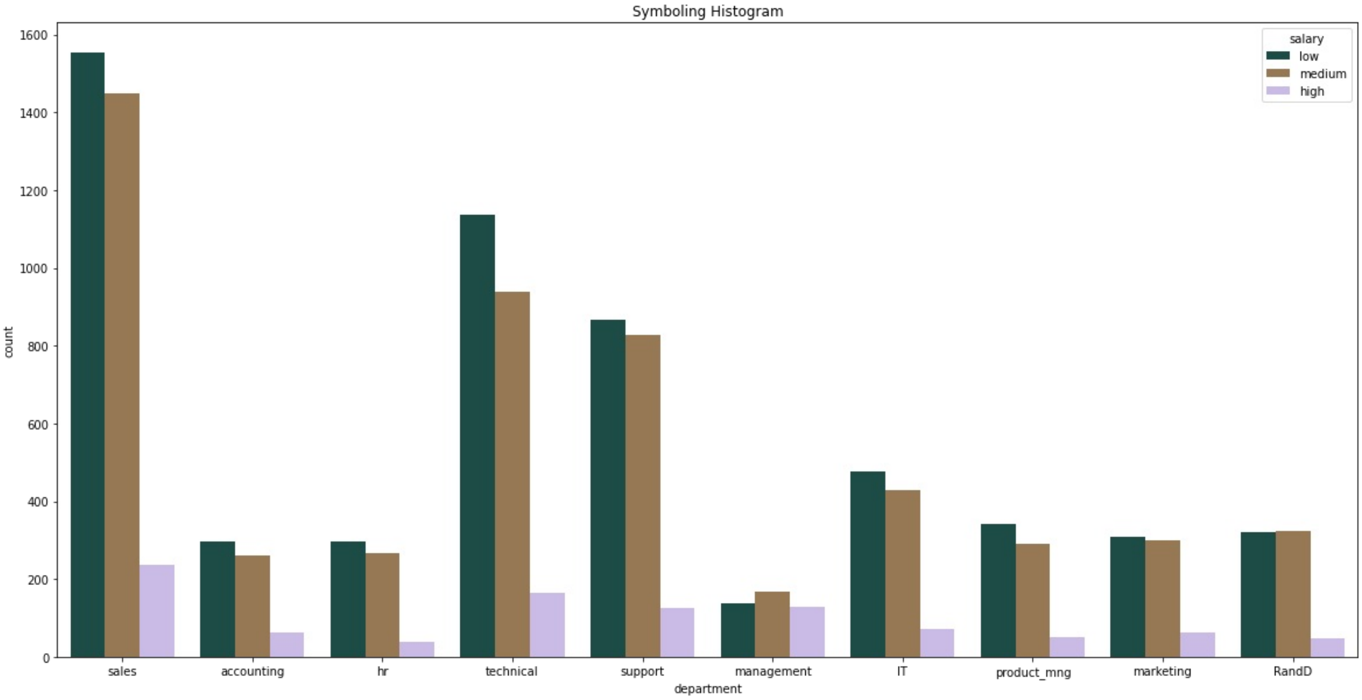
```
count_plot(left)
```



ترسیم نمودار تعداد افراد مشغول درهر دپارتمان دیتاست برحسب میزان درآمد (نمودار میله ای)

In [103…

```
fig=plt.subplots(figsize=(20, 10))
plt.title('Symboling Histogram')
sns.countplot(x = df_eda.department, hue=df_eda.salary, palette=("cubehelix"));
```



In [104…

```
df_eda
```

Out[104…

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	

...	...	...	...	...	...	...	...
11986	0.90	0.55	3	259	10	1	0
11987	0.74	0.95	5	266	10	0	0
11988	0.85	0.54	3	185	10	0	0
11989	0.33	0.65	3	172	10	0	0
11990	0.50	0.73	4	180	3	0	0

11991 rows × 11 columns

In [105…

```
df_eda2 = df_eda.copy()
```

In [106…

```
df_eda2['salary'] = df_eda2['salary'].map({'low' : 0, 'medium' : 1, 'high':2})
```

In [107…

```
df_eda2
```

Out[107…

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	
11989	0.33	0.65	3	172	10	0	0	
11990	0.50	0.73	4	180	3	0	0	

11991 rows × 11 columns

In [108…

```
df_eda2.salary.value_counts()
```

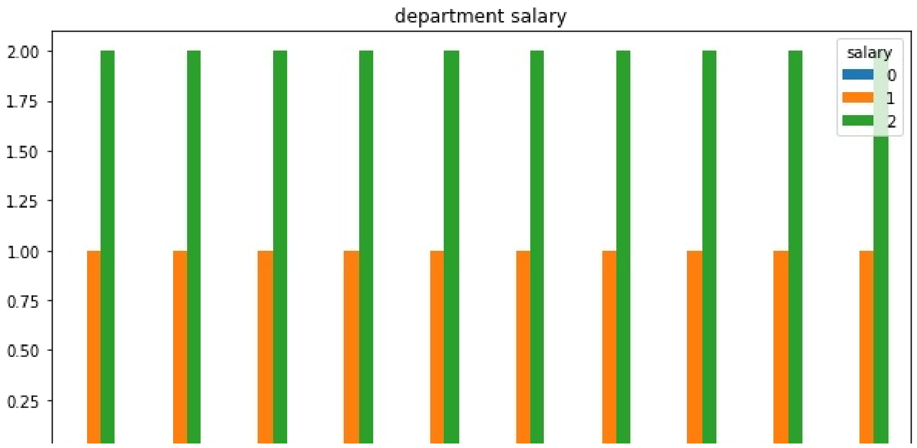
Out[108…

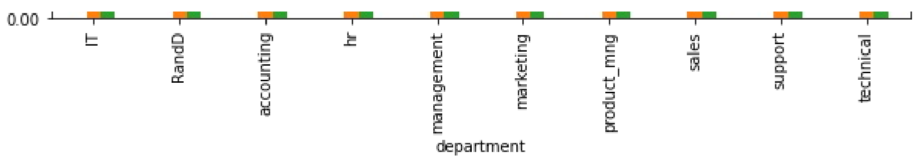
```
0    5740
1    5261
2     990
Name: salary, dtype: int64
```

In [109…

```
plt.figure(figsize=(25, 6))

df  = pd.DataFrame(df_eda2.groupby(['department','salary'])['salary'].mean().unstack(fill_value=0))
df.plot.bar()
plt.title('department salary')
plt.show()
```

```
<Figure size 1800x432 with 0 Axes>
```





```
==============================================================
```

شناسایی داده های پرت :

In [110…

```
df_o = df_c.copy()
```

In [111…

```
df_o
```

Out[111…

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2	157	3	0	1	
1	0.80	0.86	5	262	6	0	1	
2	0.11	0.88	7	272	4	0	1	
3	0.72	0.87	5	223	5	0	1	
4	0.37	0.52	2	159	3	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3	259	10	1	0	
11987	0.74	0.95	5	266	10	0	0	
11988	0.85	0.54	3	185	10	0	0	
11989	0.33	0.65	3	172	10	0	0	
11990	0.50	0.73	4	180	3	0	0	

11991 rows × 18 columns

In [112…

```
df_o.dtypes
```

Out[112…

```
satisfaction_level        float64
last_evaluation           float64
number_project              int64
average_montly_hours        int64
time_spend_company          int64
Work_accident               int64
left                        int64
promotion_last_5years       int64
salary                      int64
RandD                       uint8
accounting                  uint8
hr                          uint8
management                  uint8
marketing                   uint8
product_mng                 uint8
sales                       uint8
support                     uint8
technical                   uint8
dtype: object
```

In [113…

```
col = df_eda_encode.columns
```

In [114…

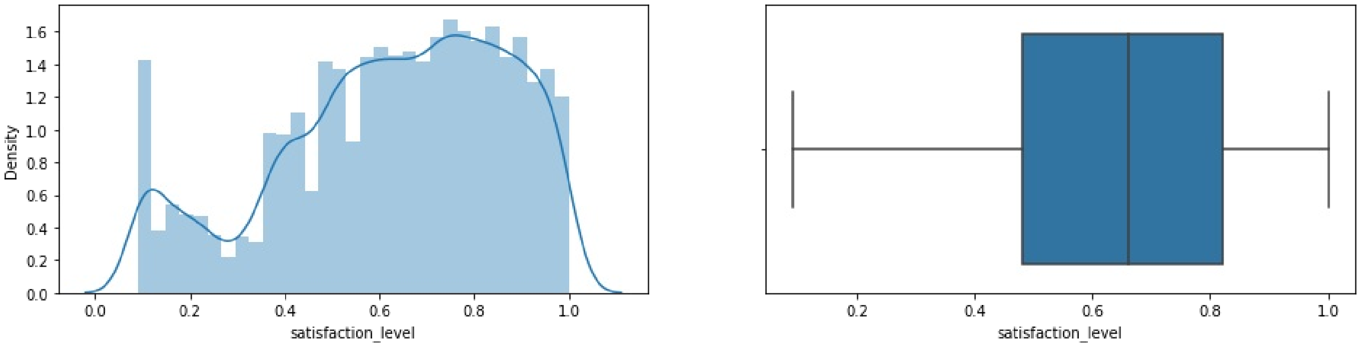
```
display(HTML("<style>div.output_scroll { height: 44em; }</style>"))
```

In [115…

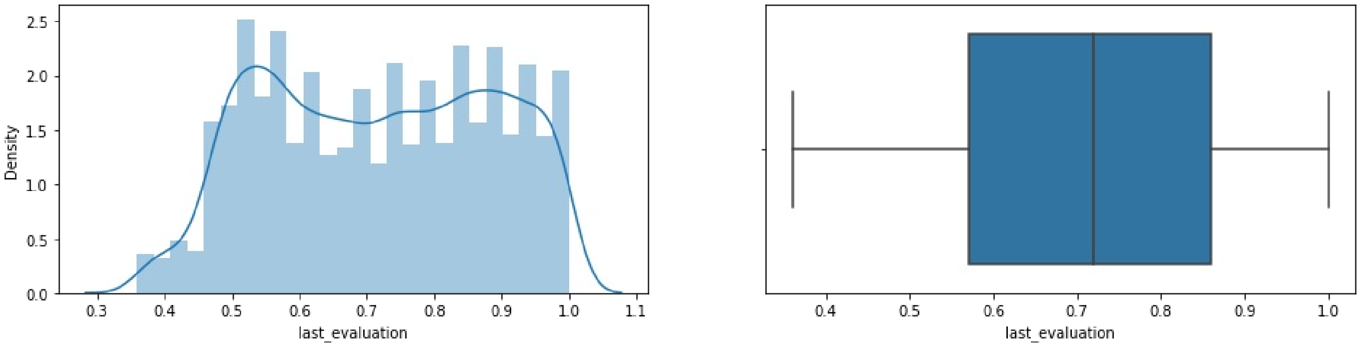
```
pu.plot_distBox_single_df(df_o)
```

```
**********************************************************
```

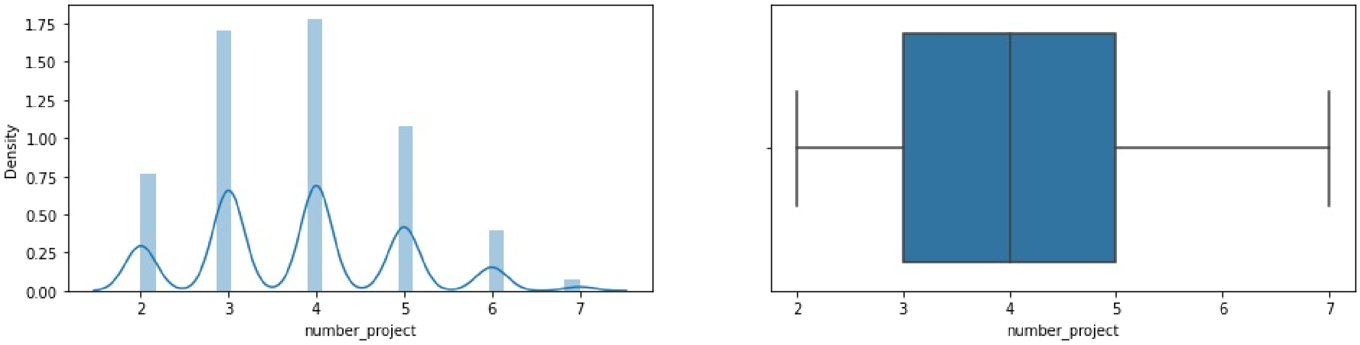
satisfaction_level



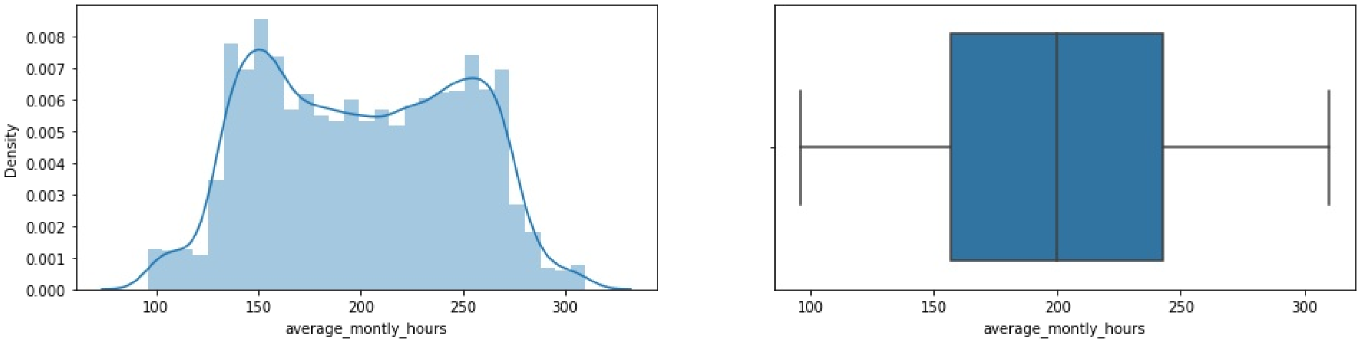
****
last_evaluation



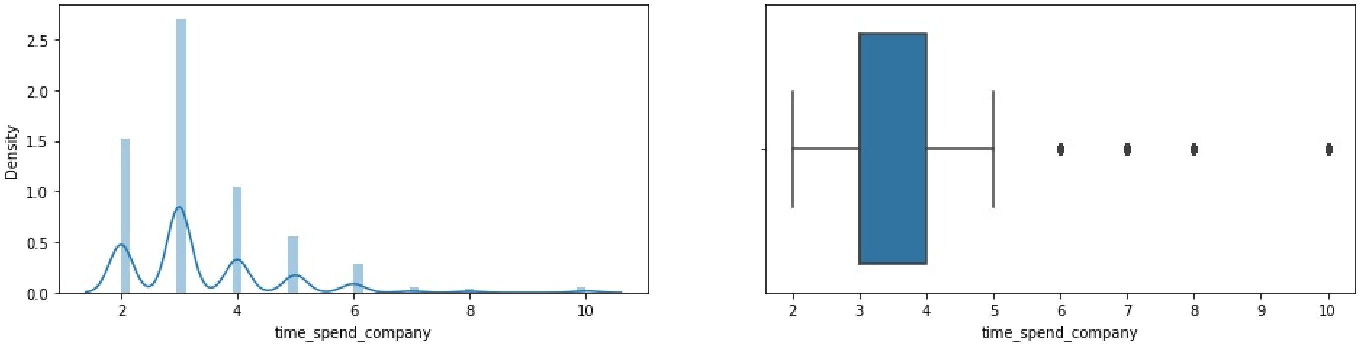
****
number_project



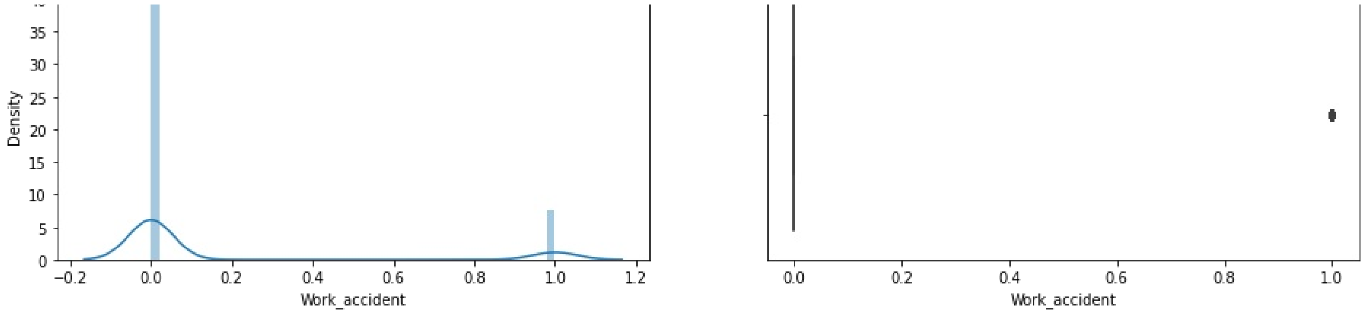
****
average_montly_hours



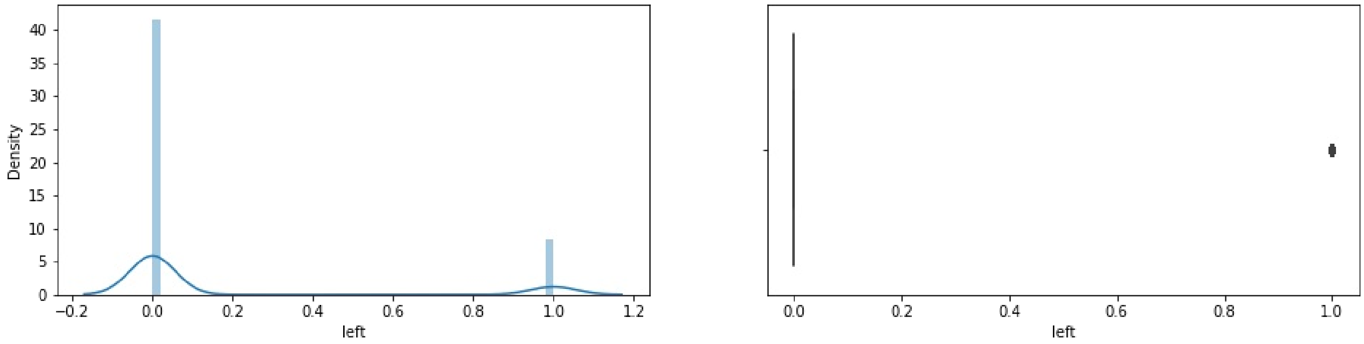
****
time_spend_company



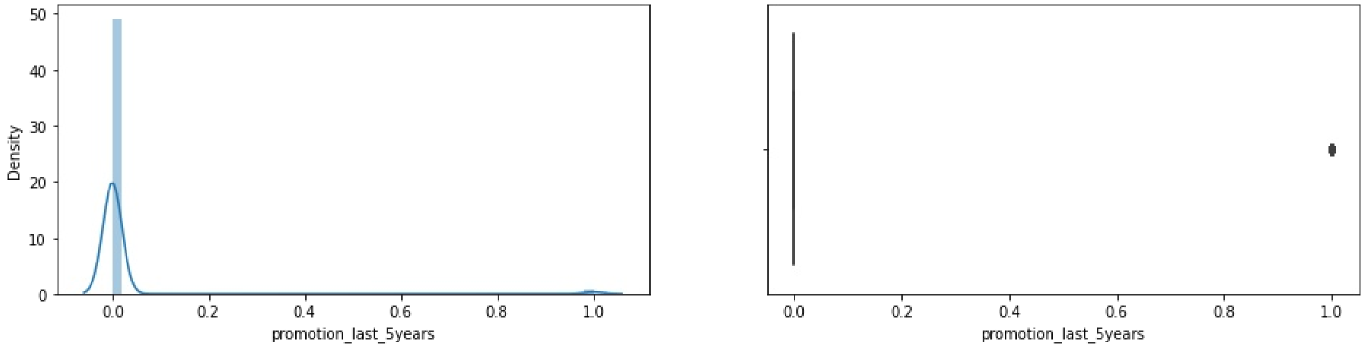
****
Work_accident



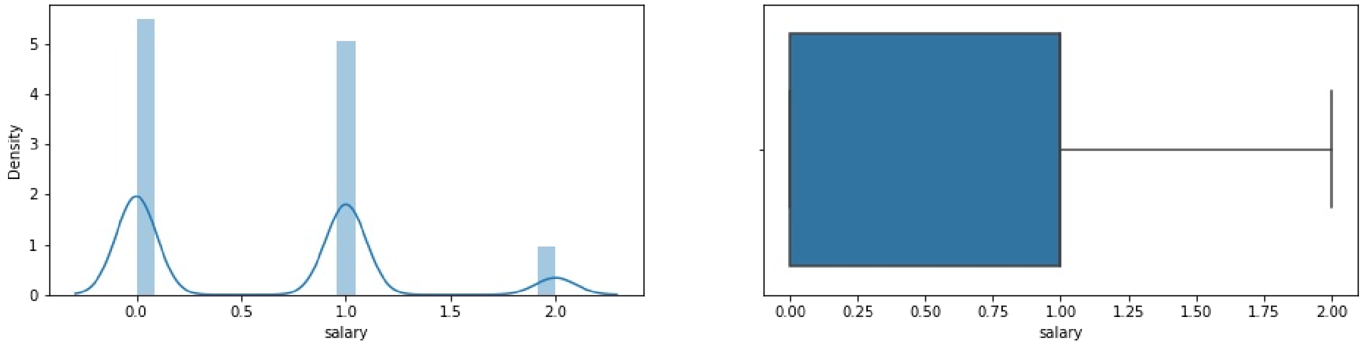
```
**********************************************************
left
```



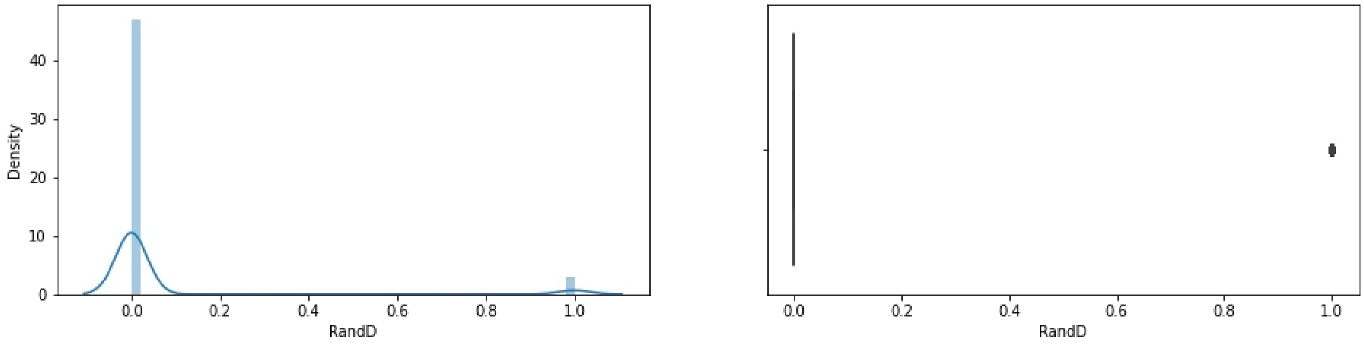
```
**********************************************************
promotion_last_5years
```



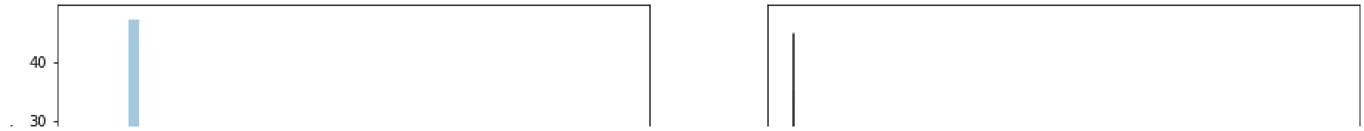
```
**********************************************************
salary
```

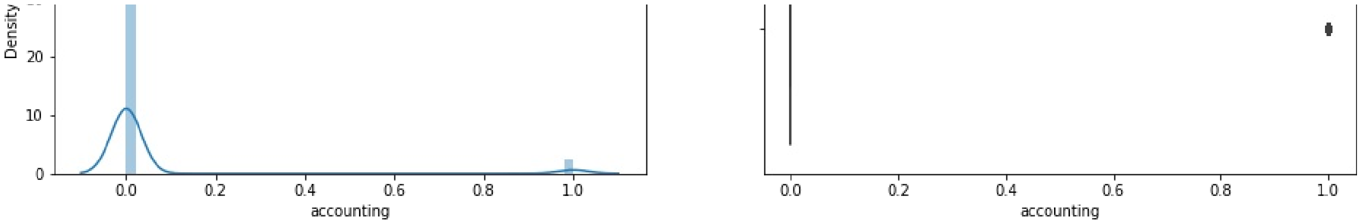


```
**********************************************************
RandD
```

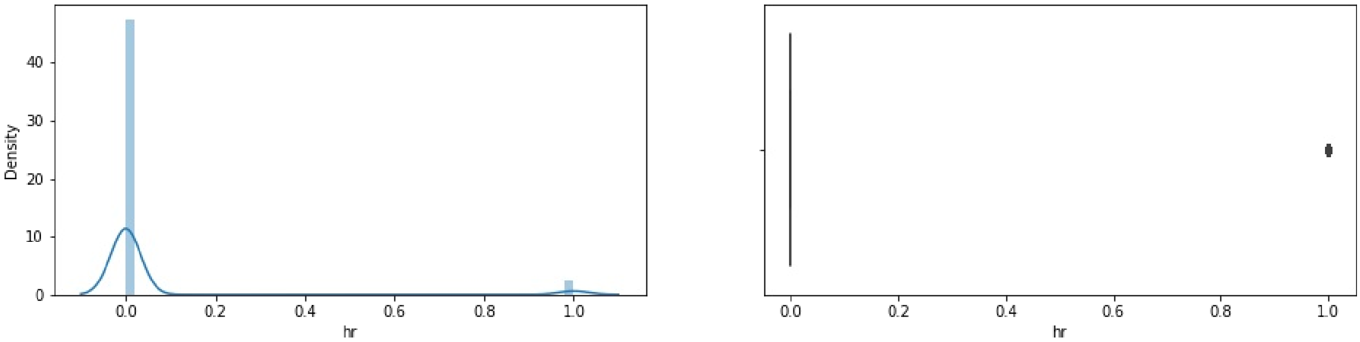


```
**********************************************************
accounting
```

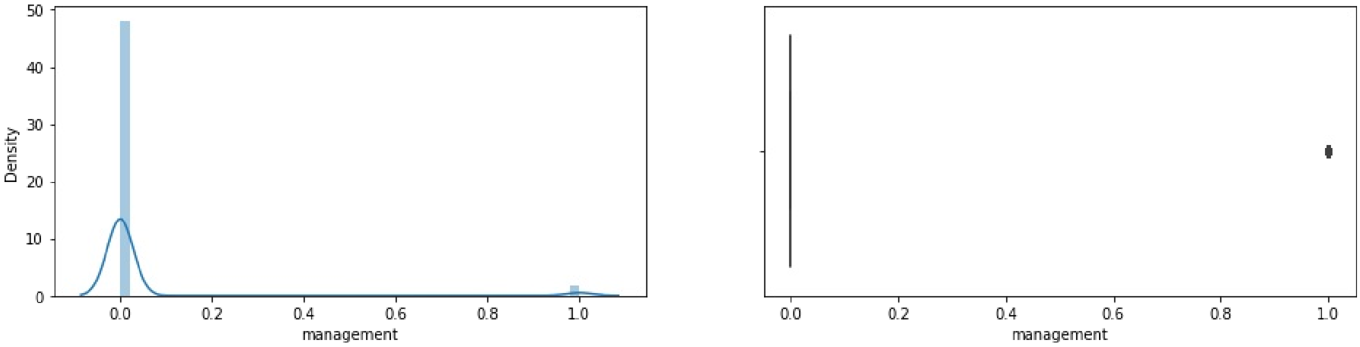




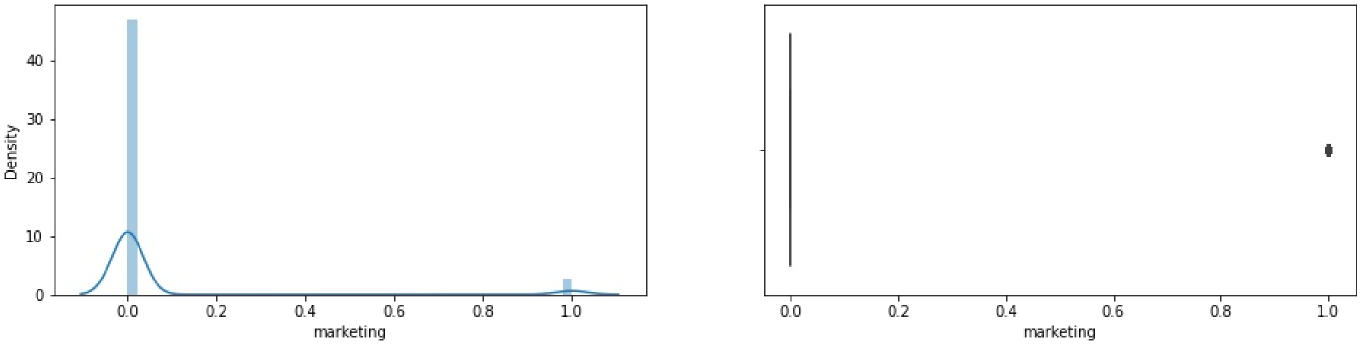
```
***********************************************************
hr
```



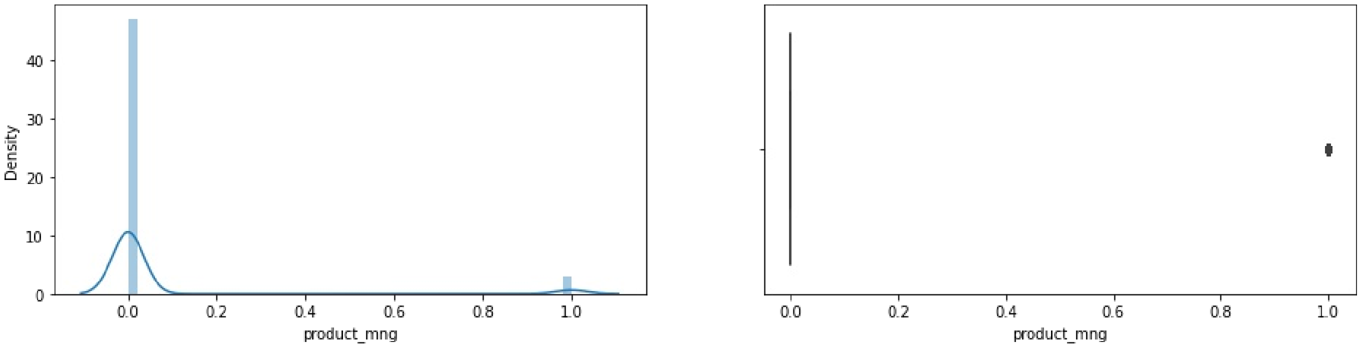
```
***********************************************************
management
```



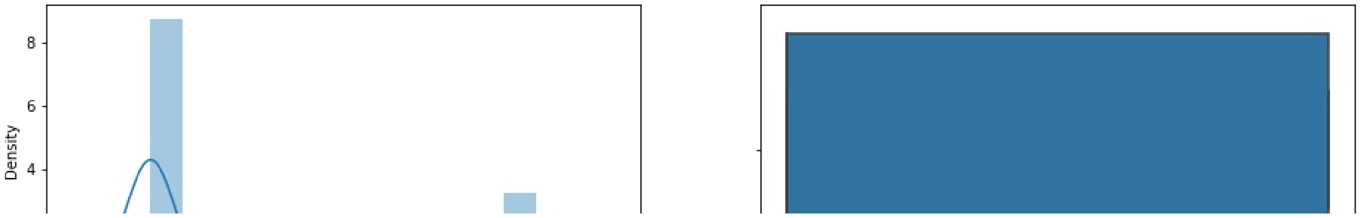
```
***********************************************************
marketing
```

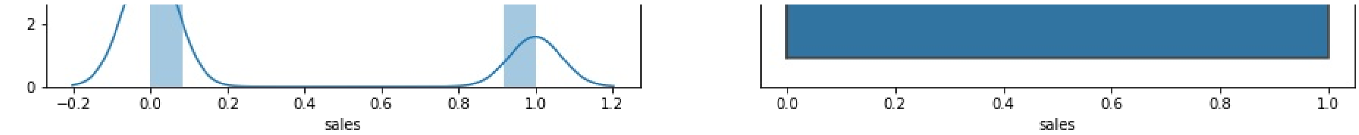


```
***********************************************************
product_mng
```

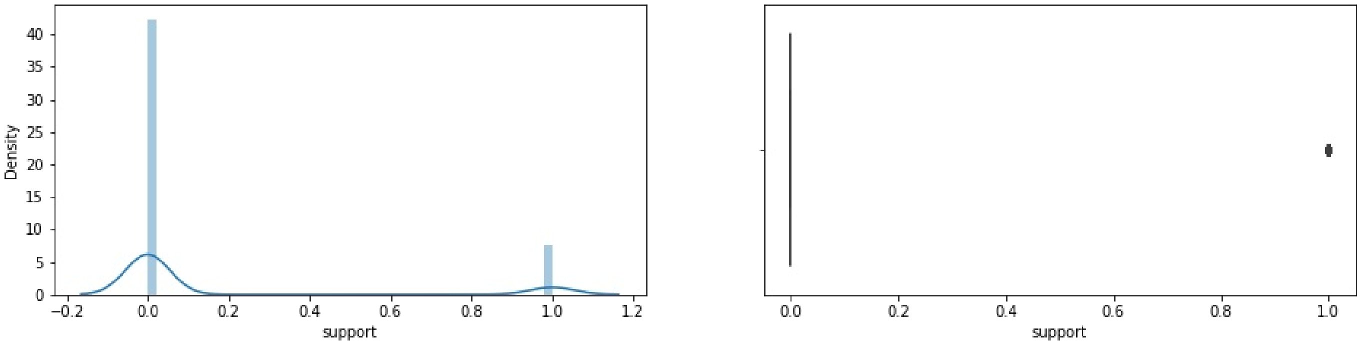


```
***********************************************************
sales
```

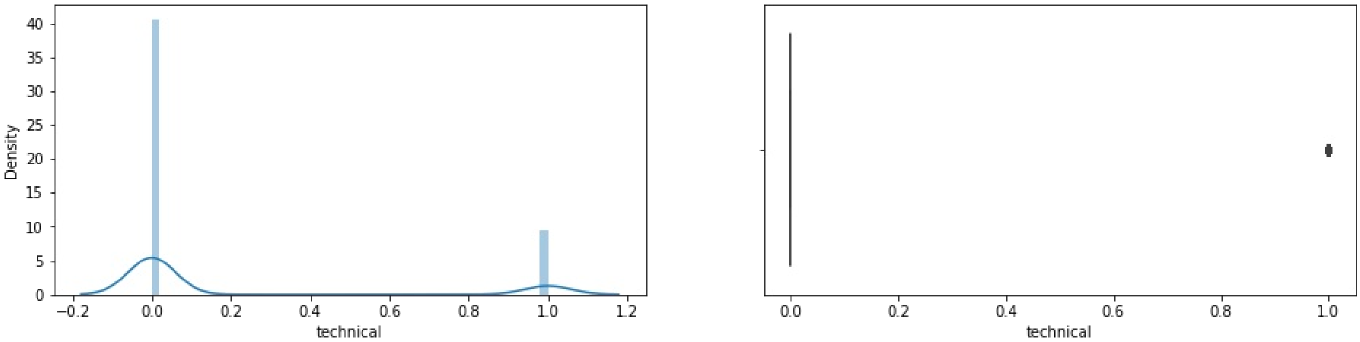




```
***********************************************************
support
```



```
***********************************************************
technical
```



In [116…

```
pu.normalityTest(df_o, col)
```

```
---------------------------------------------------
satisfaction_level
```

```
NormaltestResult(statistic=770.1049674073481, pvalue=5.940610579928223e-168)
k2 =  770.1049674073481 ,  p-value =  5.940610579928223e-168
NOT normal(Guassian)
--------------------------------------------------
last_evaluation

NormaltestResult(statistic=8734.41841212707, pvalue=0.0)
k2 =  8734.41841212707 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
number_project

NormaltestResult(statistic=332.832093896116, pvalue=5.326342755722184e-73)
k2 =  332.832093896116 ,  p-value =  5.326342755722184e-73
NOT normal(Guassian)
--------------------------------------------------
average_montly_hours

NormaltestResult(statistic=4414.000309899644, pvalue=0.0)
k2 =  4414.000309899644 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
time_spend_company

NormaltestResult(statistic=4512.754504540594, pvalue=0.0)
k2 =  4512.754504540594 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
Work_accident

NormaltestResult(statistic=3816.515250478078, pvalue=0.0)
k2 =  3816.515250478078 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
left

NormaltestResult(statistic=3426.6849212214593, pvalue=0.0)
k2 =  3426.6849212214593 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
promotion_last_5years

NormaltestResult(statistic=15398.837296966494, pvalue=0.0)
k2 =  15398.837296966494 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
salary

NormaltestResult(statistic=1006.291527901393, pvalue=3.065992747538444e-219)
k2 =  1006.291527901393 ,  p-value =  3.065992747538444e-219
NOT normal(Guassian)
--------------------------------------------------
RandD

NormaltestResult(statistic=9036.58000951614, pvalue=0.0)
k2 =  9036.58000951614 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
accounting

NormaltestResult(statistic=9612.038230734379, pvalue=0.0)
k2 =  9612.038230734379 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
hr

NormaltestResult(statistic=9781.196883701068, pvalue=0.0)
k2 =  9781.196883701068 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
management

NormaltestResult(statistic=11435.345248055768, pvalue=0.0)
k2 =  11435.345248055768 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
marketing

NormaltestResult(statistic=9195.874937805253, pvalue=0.0)
k2 =  9195.874937805253 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
product_mng
```

```
NormaltestResult(statistic=9096.707834525998, pvalue=0.0)
k2 =  9096.707834525998 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
sales

NormaltestResult(statistic=3382.597625816634, pvalue=0.0)
k2 =  3382.597625816634 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
support

NormaltestResult(statistic=3901.327394944481, pvalue=0.0)
k2 =  3901.327394944481 ,  p-value =  0.0
NOT normal(Guassian)
--------------------------------------------------
technical

NormaltestResult(statistic=2823.5243693465254, pvalue=0.0)
k2 =  2823.5243693465254 ,  p-value =  0.0
NOT normal(Guassian)
```

In [117…

```
def value_counts(dataFrame, col_name):
    for i in col_name:
        print('+++++ {} ++++++++++++++'.format(i))
        print(dataFrame[i].value_counts())
```

In [118…

```
value_counts(df_o, col)
```

```
+++++ satisfaction_level ++++++++++++++
0.74    214
0.10    203
0.73    201
0.50    200
0.72    199
       ...
0.25     29
0.26     28
0.12     26
0.28     24
0.27     23
Name: satisfaction_level, Length: 92, dtype: int64
+++++ last_evaluation ++++++++++++++
0.55    281
0.50    269
0.51    264
0.57    258
0.54    252
       ...
0.42     45
0.43     44
0.38     42
0.44     35
0.36     19
Name: last_evaluation, Length: 65, dtype: int64
+++++ number_project ++++++++++++++
4    3685
3    3520
5    2233
2    1582
6     826
7     145
Name: number_project, dtype: int64
+++++ average_montly_hours ++++++++++++++
156    112
149    112
160    111
151    107
135    104
      ...
298      5
302      5
297      5
299      5
303      5
Name: average_montly_hours, Length: 215, dtype: int64
+++++ time_spend_company ++++++++++++++
3     5190
```

```
2     2910
4     2005
5     1062
6      542
10     107
7       94
8       81
Name: time_spend_company, dtype: int64
+++++ Work_accident ++++++++++++++
0    10141
1     1850
Name: Work_accident, dtype: int64
+++++ left ++++++++++++++
0    10000
1     1991
Name: left, dtype: int64
+++++ promotion_last_5years ++++++++++++++
0    11788
1      203
Name: promotion_last_5years, dtype: int64
+++++ salary ++++++++++++++
0    5740
1    5261
2     990
Name: salary, dtype: int64
+++++ RandD ++++++++++++++
0    11297
1      694
Name: RandD, dtype: int64
+++++ accounting ++++++++++++++
0    11370
1      621
Name: accounting, dtype: int64
+++++ hr ++++++++++++++
0    11390
1      601
Name: hr, dtype: int64
+++++ management ++++++++++++++
0    11555
1      436
Name: management, dtype: int64
+++++ marketing ++++++++++++++
0    11318
1      673
Name: marketing, dtype: int64
+++++ product_mng ++++++++++++++
0    11305
1      686
Name: product_mng, dtype: int64
+++++ sales ++++++++++++++
0    8752
1    3239
Name: sales, dtype: int64
+++++ support ++++++++++++++
0    10170
1     1821
Name: support, dtype: int64
+++++ technical ++++++++++++++
0    9747
1    2244
Name: technical, dtype: int64
```

In [119…

```
df_o.columns
```

Out[119…

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'salary', 'RandD', 'accounting', 'hr',
       'management', 'marketing', 'product_mng', 'sales', 'support',
       'technical'],
      dtype='object')
```

In [120…

```
col_outlierDetection = ['satisfaction_level', 'last_evaluation', 'number_project',
                        'average_montly_hours', 'time_spend_company']
```

In [121…

```
pu.kolmogorov_smirnov_test(df_o)
```

```
satisfaction_level
```

```
satisfaction_level



p value for norm = 1.631889109238576e-51
p value for exponweib = 1.6198802864214826e-42
p value for weibull_max = 6.849895420283212e-14
p value for weibull_min = 2.794049269989194e-18
p value for pareto = 0.0
p value for genextreme = 6.527327344933816e-14
Best fitting distribution: weibull_max
Best p value: 6.849895420283212e-14
Parameters for the best fit: (1.5808993798778324, 1.0135063084783094, 0.42636673487871435)


**************************************************************


last_evaluation



p value for norm = 5.183639476589461e-63
p value for exponweib = 1.3384837927085421e-32
p value for weibull_max = 4.454011454573993e-75
p value for weibull_min = 1.0449828002938513e-58
p value for pareto = 0.0
p value for genextreme = 4.778208323459196e-75
Best fitting distribution: exponweib
Best p value: 1.3384837927085421e-32
Parameters for the best fit: (0.011473577120748566, 117.13595607357009, 0.35887811534275976, 0.6415666147581649)


**************************************************************


number_project



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 3.310149437330311e-303
Best fitting distribution: genextreme
Best p value: 3.310149437330311e-303
Parameters for the best fit: (0.1697663156835818, 3.334968755632744, 1.0699603146664618)


**************************************************************


average_montly_hours



p value for norm = 1.2395061042006772e-55
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 2.970409712565525e-54
Best fitting distribution: genextreme
Best p value: 2.970409712565525e-54
Parameters for the best fit: (0.3385281081439747, 184.5560373963241, 49.0368273890837)


**************************************************************


time_spend_company



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
```

```
Parameters for the best fit: (3.3648569760653824, 1.3301840485480776)


**************************************************************


Work_accident



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.15428237845050455, 0.3612192217340598)


**************************************************************


left



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.1660411975648403, 0.3721176134988425)


**************************************************************


promotion_last_5years



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.016929363689433742, 0.1290068228215261)


**************************************************************


salary



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.6038695688432991, 0.6358733801241162)


**************************************************************


RandD



p value for norm = 0.0
```

```
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.057876740889000085, 0.2335102219455663)


**************************************************************


accounting



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.051788841631223416, 0.22160044565325104)


**************************************************************


hr



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.0501209240263531, 0.218194447686227)


**************************************************************


management



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.036360603786172965, 0.18718576409139107)


**************************************************************


marketing



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.05612542740388625, 0.23016377604353241)


**************************************************************
```

```
product_mng



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.057209573847051956, 0.23224262853165145)


**************************************************************


sales



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.2701192561087482, 0.44402121974969255)


**************************************************************


support



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.15186389792344257, 0.3588889165618531)


**************************************************************


technical



p value for norm = 0.0
p value for exponweib = 0.0
p value for weibull_max = 0.0
p value for weibull_min = 0.0
p value for pareto = 0.0
p value for genextreme = 0.0
Best fitting distribution: norm
Best p value: 0.0
Parameters for the best fit: (0.18714035526644984, 0.3900241565559712)


**************************************************************
```

In [122…

```
pu.outliers_detection_IQR_with_Coarse(df_o, col_outlierDetection)
```

In [123…

```
pu.printNull(df_o, col)
```

```
-------
```

```
satisfaction_level has 0 null
-------
last_evaluation has 0 null
-------
number_project has 0 null
-------
average_montly_hours has 0 null
-------
time_spend_company has 824 null
-------
Work_accident has 0 null
-------
left has 0 null
-------
promotion_last_5years has 0 null
-------
salary has 0 null
-------
RandD has 0 null
-------
accounting has 0 null
-------
hr has 0 null
-------
management has 0 null
-------
marketing has 0 null
-------
product_mng has 0 null
-------
sales has 0 null
-------
support has 0 null
-------
technical has 0 null
```

In [124…

```
df_o['time_spend_company'].isnull().sum()
```

Out[124…

```
824
```

گام 4 : مدیریت داده های مفقوده

In [125…

```
df_m = df_o
```

choosing the best way to fill this field

In [126…

```
df_m2 = df_m.copy()
df_m3 = df_m.copy()
df_m4 = df_m.copy()
df_m5 = df_m.copy()
```

In [127…

```
df_m['time_spend_company']
```

Out[127…

```
0        3.0
1        NaN
2        4.0
3        5.0
4        3.0
        ...
11986    NaN
11987    NaN
11988    NaN
11989    NaN
11990    3.0
Name: time_spend_company, Length: 11991, dtype: float64
```

In [128…

```
df_m2['time_spend_company'].fillna(value=6, inplace=True)
```

In [129…

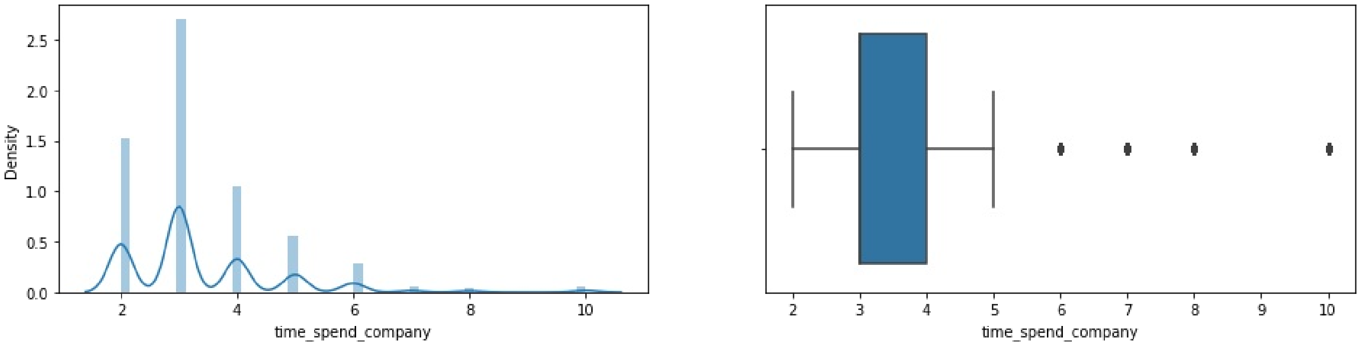
```
df_m2['time_spend_company'].isnull().sum()
```

Out[129… 0

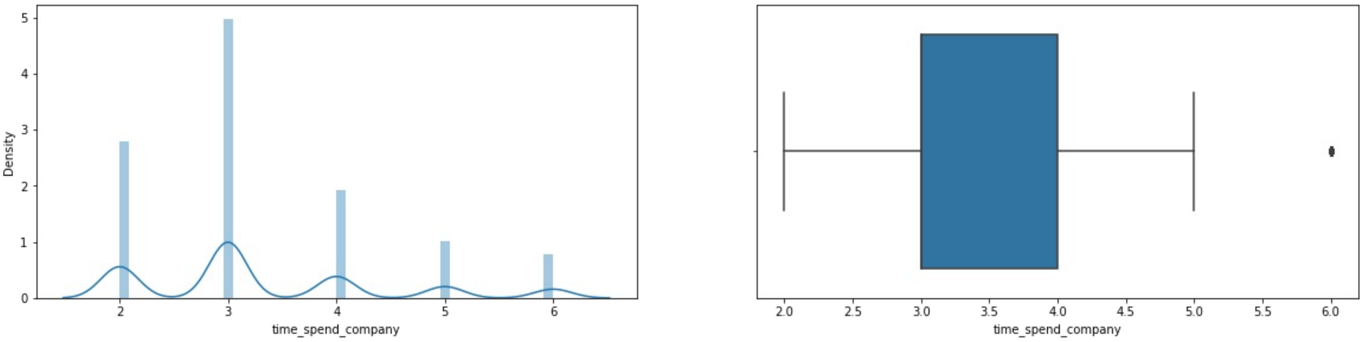
In [130…

```
pu.plot_distBox_two_df(df_c, df_m2, ['time_spend_company'], 'filling miss values')
```

```
**********************************************************
time_spend_company          before  filling miss values
```



```
time_spend_company          after  filling miss values
```



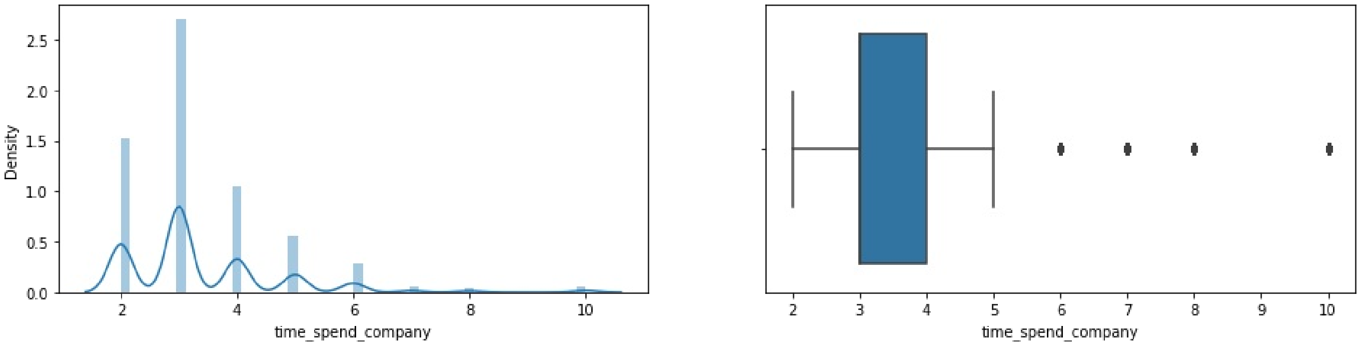
In [131…

```
from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors=3)
df_m3.iloc[:,:] = imputer.fit_transform(df_m3)
```

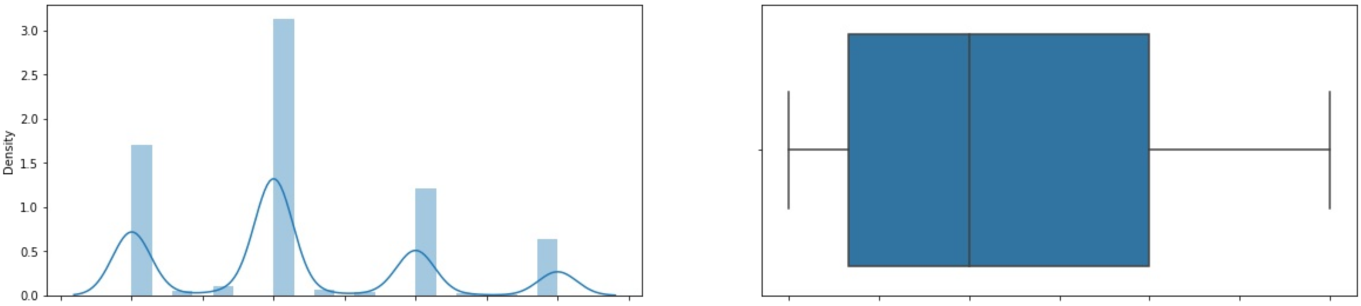
In [132…

```
pu.plot_distBox_two_df(df_c, df_m3, ['time_spend_company'], 'filling miss values')
```

```
**********************************************************
time_spend_company          before  filling miss values
```



```
time_spend_company          after  filling miss values
```



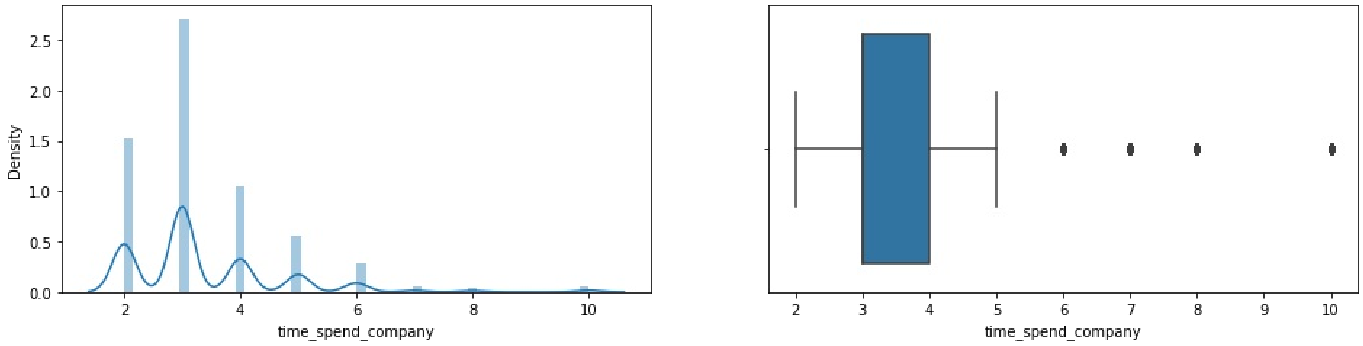
In [133…

```
from fancyimpute import IterativeImputer

II = IterativeImputer()
df_m3.iloc[:, :] = II.fit_transform(df_m4)
```

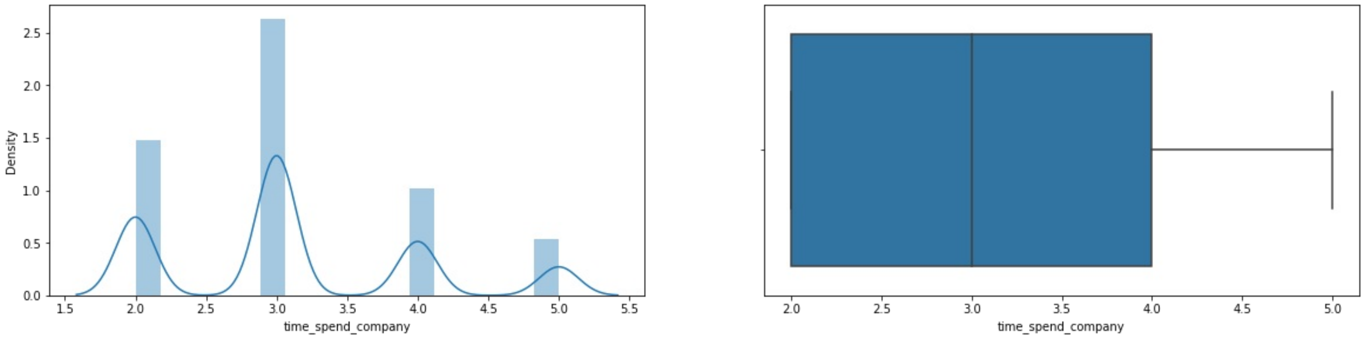
In [134…

```
pu.plot_distBox_two_df(df_c, df_m4, ['time_spend_company'], 'filling miss values')
```

```
**********************************************************
time_spend_company          before  filling miss values
```



```
time_spend_company          after  filling miss values
```



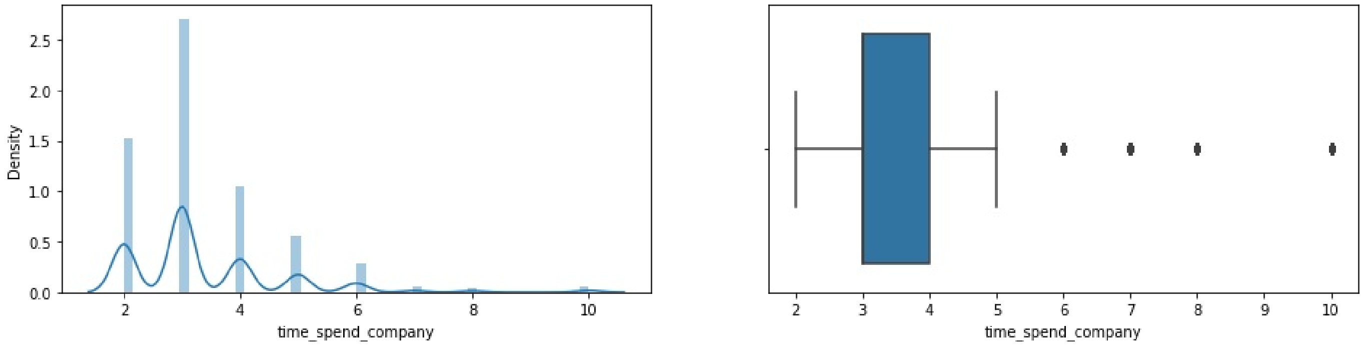
In [135…

```
df_m5['time_spend_company'] = df_m5['time_spend_company'].fillna(df_m5['time_spend_company'].ffill())
```

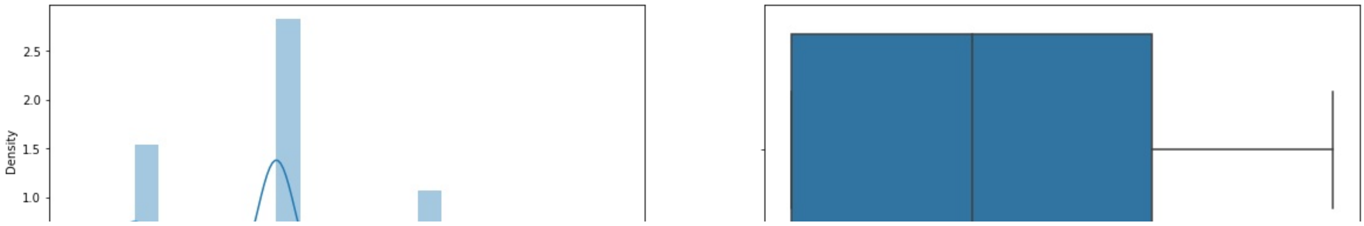
In [136…

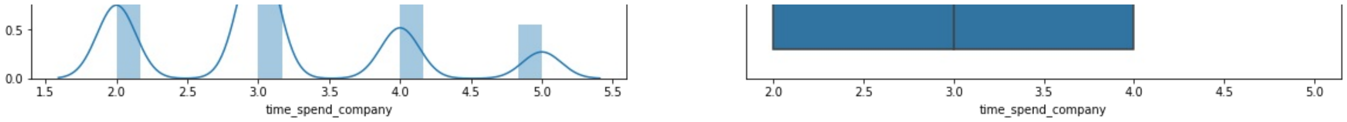
```
pu.plot_distBox_two_df(df_c, df_m5, ['time_spend_company'], 'filling miss values')
```

```
**********************************************************
time_spend_company          before  filling miss values
```



```
time_spend_company          after  filling miss values
```





In [137…

```
df_m = df_m2.copy()
```

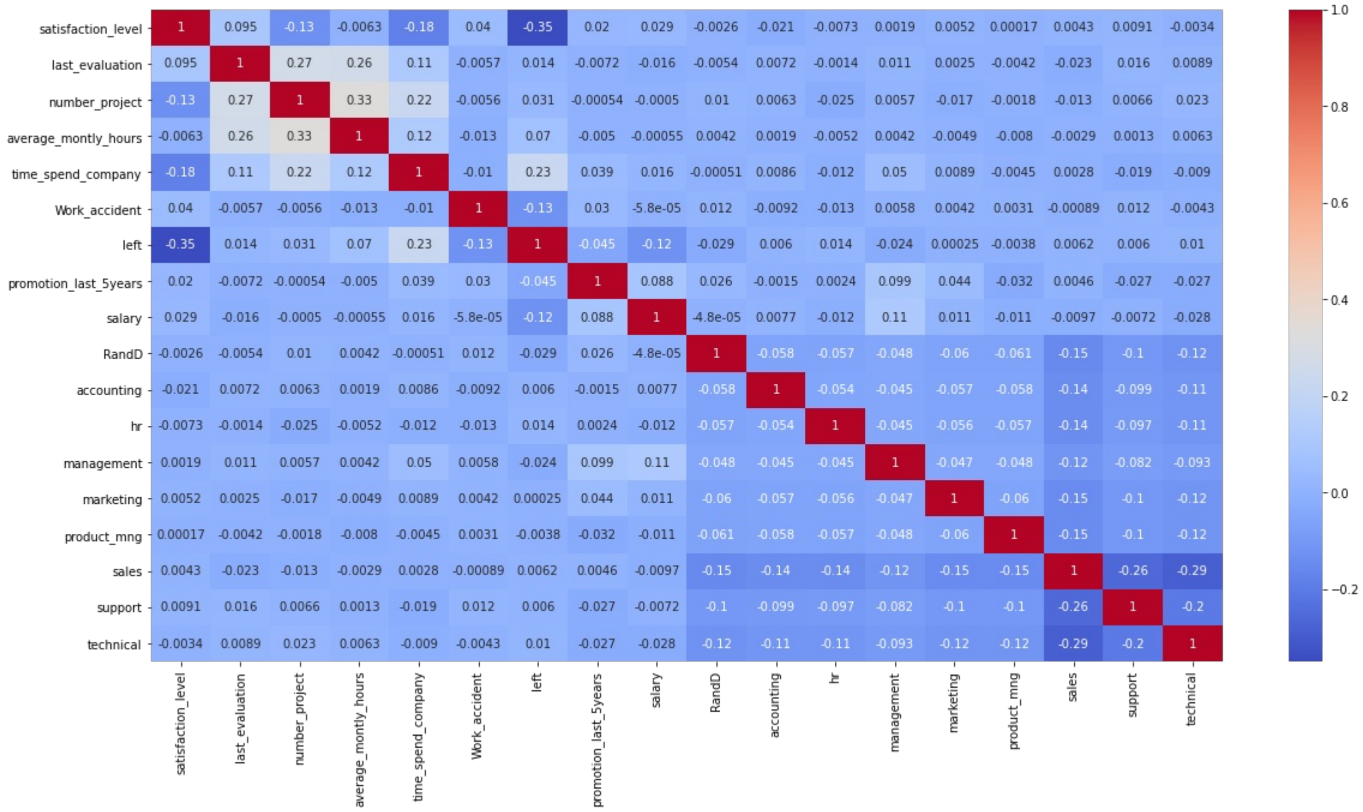
A little EDA

In [138…

```
# import dtale

# dtale.show(df_m)
```

In [139…

```
corr = df_m.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm');
```



In [140…

```
value_counts(df_m, col)
```

```
+++++ satisfaction_level ++++++++++++++
0.74    214
0.10    203
0.73    201
0.50    200
0.72    199
       ...
0.25     29
0.26     28
0.12     26
0.28     24
0.27     23
Name: satisfaction_level, Length: 92, dtype: int64
+++++ last_evaluation ++++++++++++++
0.55    281
0.50    269
0.51    264
0.57    258
0.54    252
       ...
```

```
0.42     45
0.43     44
0.38     42
0.44     35
0.36     19
Name: last_evaluation, Length: 65, dtype: int64
+++++ number_project ++++++++++++++
4.0    3685
3.0    3520
5.0    2233
2.0    1582
6.0     826
7.0     145
Name: number_project, dtype: int64
+++++ average_montly_hours ++++++++++++++
156.0    112
149.0    112
160.0    111
151.0    107
135.0    104
        ...
298.0      5
302.0      5
297.0      5
299.0      5
303.0      5
Name: average_montly_hours, Length: 215, dtype: int64
+++++ time_spend_company ++++++++++++++
3.0    5190
2.0    2910
4.0    2005
5.0    1062
6.0     824
Name: time_spend_company, dtype: int64
+++++ Work_accident ++++++++++++++
0    10141
1     1850
Name: Work_accident, dtype: int64
+++++ left ++++++++++++++
0    10000
1     1991
Name: left, dtype: int64
+++++ promotion_last_5years ++++++++++++++
0    11788
1      203
Name: promotion_last_5years, dtype: int64
+++++ salary ++++++++++++++
0    5740
1    5261
2     990
Name: salary, dtype: int64
+++++ RandD ++++++++++++++
0    11297
1      694
Name: RandD, dtype: int64
+++++ accounting ++++++++++++++
0    11370
1      621
Name: accounting, dtype: int64
+++++ hr ++++++++++++++
0    11390
1      601
Name: hr, dtype: int64
+++++ management ++++++++++++++
0    11555
1      436
Name: management, dtype: int64
+++++ marketing ++++++++++++++
0    11318
1      673
Name: marketing, dtype: int64
+++++ product_mng ++++++++++++++
0    11305
1      686
Name: product_mng, dtype: int64
+++++ sales ++++++++++++++
0    8752
1    3239
Name: sales, dtype: int64
+++++ support ++++++++++++++
0    10170
1     1821
Name: support, dtype: int64
```

```
+++++ technical ++++++++++++++
0    9747
1    2244
Name: technical, dtype: int64
```

In [141…

```
df_m.columns
```

Out[141…

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'salary', 'RandD', 'accounting', 'hr',
       'management', 'marketing', 'product_mng', 'sales', 'support',
       'technical'],
      dtype='object')
```

In [142…

```
col_cat = ['average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'salary', 'RandD', 'accounting', 'hr',
       'management', 'marketing', 'product_mng', 'sales', 'support',
       'technical']
```

In [143…

```
df_m[col_cat]
```

Out[143…

	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	salary	RandD	accounting	hr	management
0	157.0	3.0	0	1	0	0	0	0	0	0
1	262.0	6.0	0	1	0	1	0	0	0	0
2	272.0	4.0	0	1	0	1	0	0	0	0
3	223.0	5.0	0	1	0	0	0	0	0	0
4	159.0	3.0	0	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
11986	259.0	6.0	1	0	1	2	0	0	0	1
11987	266.0	6.0	0	0	1	2	0	0	0	1
11988	185.0	6.0	0	0	1	2	0	0	0	1
11989	172.0	6.0	0	0	1	2	0	0	0	0
11990	180.0	3.0	0	0	0	0	0	0	0	0

11991 rows × 15 columns

In [144…

```
x = df_m[col_cat].drop('left', axis=1)
y = df_m.left
```

In [145…

```
chi_scores = chi2(x, y)
chi_scores
```

Out[145…

```
(array([7.04006071e+02, 2.37510708e+02, 1.59561439e+02, 2.35084927e+01,
        1.13470974e+02, 9.51110150e+00, 4.03222688e-01, 2.09662497e+00,
        6.88900142e+00, 6.93790781e-04, 1.60469771e-01, 3.31450347e-01,
        3.68461138e-01, 9.74748368e-01]),
 array([4.02310410e-155, 1.37244127e-053, 1.41081324e-036, 1.24363595e-006,
        1.70154015e-026, 2.04232532e-003, 5.25429603e-001, 1.47624683e-001,
        8.67277039e-003, 9.78986224e-001, 6.88724378e-001, 5.64806317e-001,
        5.43844277e-001, 3.23498798e-001]))
```

In [146…

```
p_values = pd.Series(chi_scores[1],index = x.columns)
p_values.sort_values(ascending = False , inplace = True)
p_values
```

Out[146…

```
marketing                9.789862e-01
product_mng              6.887244e-01
sales                    5.648063e-01
support                  5.438443e-01
accounting               5.254296e-01
technical                3.234988e-01
hr                       1.476247e-01
```

```
management                 8.672770e-03
RandD                      2.042325e-03
promotion_last_5years      1.243636e-06
salary                     1.701540e-26
Work_accident              1.410813e-36
time_spend_company         1.372441e-53
average_montly_hours      4.023104e-155
dtype: float64
```

هرچی پی ولیو کمتر باشه کای2 بیشتر

کمتر از پنج صدم باعث میشه فرض اچ صفر که استقلال فیچرهاست ریجکت شده و فیچر به تارگت که وای هست وابسته تر باشه که خوبه

In [147…

```
### hr                        7.426460e-04
### management                3.422997e-08
### RandD                     2.778371e-08
### promotion_last_5years     7.083597e-14
### salary                    1.837026e-57
### time_spend_company        6.598399e-68
### Work_accident             1.121772e-68
### average_montly_hours     1.206710e-207

## □1□□□□□□□□ □ □□□□ □□□□□ □□□ □□ □□□ □□□□ □□□□ □□□□ □□□ □□□ □□ □□□□□□ 2 □□□□□□□□ □□ □□□□ □□□
###  □□□□ □□ □□□□□ □□□ □□□□□ □□ □□□□□□ □□□□□□ □□□□□ □
```

Mutu Information Correlation

In [148…

```
# pip install ennemi
```

In [149…

```
from ennemi import pairwise_mi
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [150…

```
df_MI = df_m.copy()
```

In [151…

```
df_MI
```

Out[151…

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.38	0.53	2.0	157.0	3.0	0	1	
1	0.80	0.86	5.0	262.0	6.0	0	1	
2	0.11	0.88	7.0	272.0	4.0	0	1	
3	0.72	0.87	5.0	223.0	5.0	0	1	
4	0.37	0.52	2.0	159.0	3.0	0	1	
...	...	...	...	...	...	...	...	
11986	0.90	0.55	3.0	259.0	6.0	1	0	
11987	0.74	0.95	5.0	266.0	6.0	0	0	
11988	0.85	0.54	3.0	185.0	6.0	0	0	
11989	0.33	0.65	3.0	172.0	6.0	0	0	
11990	0.50	0.73	4.0	180.0	3.0	0	0	

11991 rows × 18 columns

In [152…

```
col_cat
```

Out[152…

```
['average_montly_hours',
 'time_spend_company',
 'Work_accident',
 'left',
 'promotion_last_5years',
 'salary',
 'RandD',
 'accounting',
 'hr',
 'management',
 'marketing',
 'product_mng',
 'sales',
 'support',
 'technical']
```

In [153…

```
df_MI_cat = df_MI[col_cat]
```

In [154…

```
pairwise_cat = pairwise_mi(df_MI_cat, discrete=True)
```

In [155…

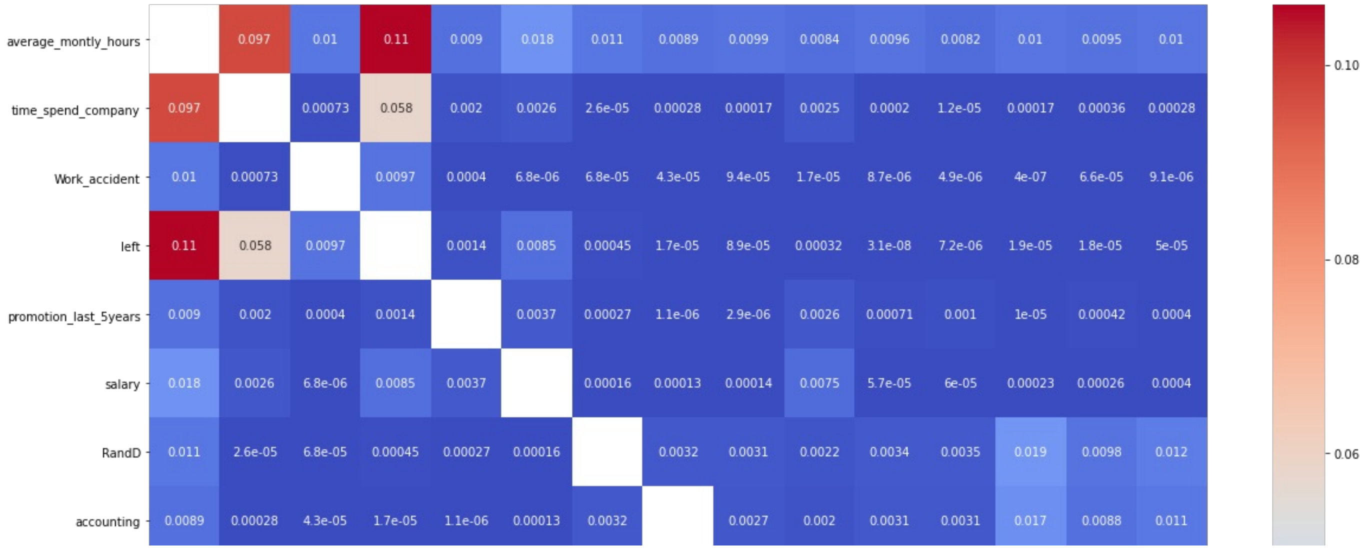
```
pairwise_cat
```

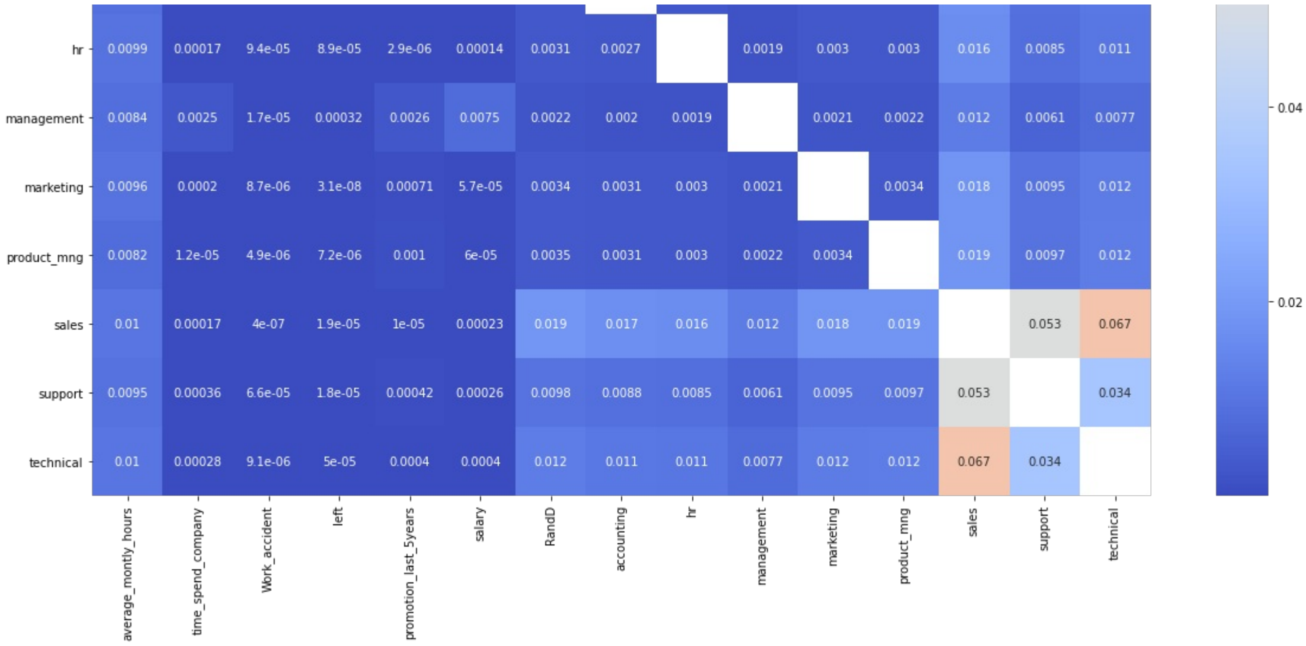
Out[155…

	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	salary	RandD	acc
average_montly_hours	NaN	0.097290	1.037501e-02	1.061262e-01	0.009045	0.017702	0.010526	0.
time_spend_company	0.097290	NaN	7.253219e-04	5.771170e-02	0.001997	0.002640	0.000026	0.
Work_accident	0.010375	0.000725	NaN	9.697385e-03	0.000395	0.000007	0.000068	0.
left	0.106126	0.057712	9.697385e-03	NaN	0.001356	0.008498	0.000453	0.
promotion_last_5years	0.009045	0.001997	3.954894e-04	1.356175e-03	NaN	0.003724	0.000270	0.
salary	0.017702	0.002640	6.778331e-06	8.497990e-03	0.003724	NaN	0.000161	0.
RandD	0.010526	0.000026	6.766800e-05	4.531807e-04	0.000270	0.000161	NaN	0.
accounting	0.008905	0.000282	4.316611e-05	1.748373e-05	0.000001	0.000128	0.003173	
hr	0.009877	0.000169	9.408788e-05	8.915624e-05	0.000003	0.000142	0.003068	0.
management	0.008446	0.002494	1.673488e-05	3.235238e-04	0.002639	0.007507	0.002209	0.
marketing	0.009614	0.000199	8.651498e-06	3.063239e-08	0.000712	0.000057	0.003447	0.
product_mng	0.008159	0.000012	4.904033e-06	7.159421e-06	0.001006	0.000060	0.003515	0.
sales	0.010002	0.000170	4.004415e-07	1.886453e-05	0.000010	0.000234	0.018874	0.
support	0.009494	0.000365	6.649079e-05	1.799069e-05	0.000419	0.000256	0.009846	0.
technical	0.010054	0.000281	9.147007e-06	4.953475e-05	0.000400	0.000402	0.012395	0.

In [156…

```
plt.figure(figsize=(20, 16))
sns.heatmap(pairwise_cat, annot=True, cmap='coolwarm');
```





In [157…

```
df_MI.columns
```

Out[157…

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'salary', 'RandD', 'accounting', 'hr',
       'management', 'marketing', 'product_mng', 'sales', 'support',
       'technical'],
      dtype='object')
```

In [158…

```
col_num = ['satisfaction_level', 'last_evaluation', 'number_project']
```

In [159…

```
df_MI[col_num]
```

Out[159…

	satisfaction_level	last_evaluation	number_project
0	0.38	0.53	2.0
1	0.80	0.86	5.0
2	0.11	0.88	7.0
3	0.72	0.87	5.0
4	0.37	0.52	2.0
...	...	...	...
11986	0.90	0.55	3.0
11987	0.74	0.95	5.0
11988	0.85	0.54	3.0
11989	0.33	0.65	3.0
11990	0.50	0.73	4.0

11991 rows × 3 columns

In [160…

```
df_MI_num = df_MI[col_num]
```

In [161…

```
pairwise_num = pairwise_mi(df_MI_num)
```

In [162…

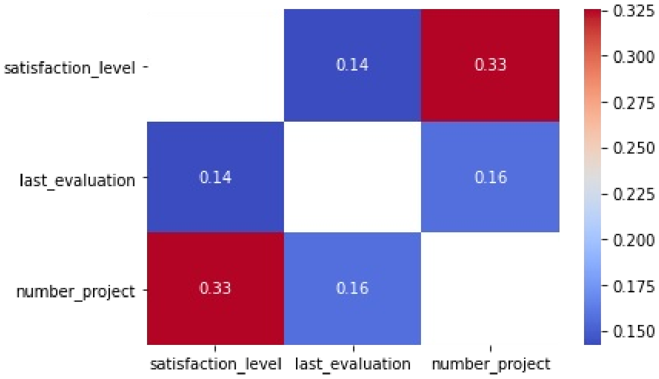
```
pairwise_num
```

Out[162…

	satisfaction_level	last_evaluation	number_project
satisfaction_level	NaN	0.142451	0.325207
last_evaluation	0.142451	NaN	0.157274
number_project	0.325207	0.157274	NaN

In [163…

```
plt.figure(figsize=(6,4))
sns.heatmap(pairwise_num, annot=True, cmap='coolwarm');
```



جداسازی داده تست و ترین

In [164…

```
df_st = df_m.copy()
```

In [165…

```
df_st.shape
```

Out[165…

```
(11991, 18)
```

In [166…

```
X = df_st.drop('left', axis=1)
y = df_st.left
```

In [167…

```
print(X.shape)
print(y.shape)
```

```
(11991, 17)
(11991,)
```

In [168…

```
from sklearn.model_selection import train_test_split
```

In [169…

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=2020)

print(X_train.shape, X_test.shape)
```

```
(9592, 17) (2399, 17)
```

In [170…

```
print(X_train.shape)
print(X_test.shape)

print(y_train.shape)
print(y_test.shape)
```

```
(9592, 17)
(2399, 17)
(9592,)
(2399,)
```

نرمال و استاندارگامسازی :

(xi-min) / (max-min) min-max scaler

In [171…

```
df_s = df_m.copy()
```

In [172…

```
df_s.describe().T
```

Out[172…

	count	mean	std	min	25%	50%	75%	max
satisfaction_level	11991.0	0.629658	0.241070	0.09	0.48	0.66	0.82	1.0
last_evaluation	11991.0	0.716683	0.168343	0.36	0.57	0.72	0.86	1.0
number_project	11991.0	3.802852	1.163238	2.00	3.00	4.00	5.00	7.0
average_montly_hours	11991.0	200.473522	48.727813	96.00	157.00	200.00	243.00	310.0
time_spend_company	11991.0	3.307814	1.134891	2.00	3.00	3.00	4.00	6.0
Work_accident	11991.0	0.154282	0.361234	0.00	0.00	0.00	0.00	1.0
left	11991.0	0.166041	0.372133	0.00	0.00	0.00	0.00	1.0
promotion_last_5years	11991.0	0.016929	0.129012	0.00	0.00	0.00	0.00	1.0
salary	11991.0	0.603870	0.635900	0.00	0.00	1.00	1.00	2.0
RandD	11991.0	0.057877	0.233520	0.00	0.00	0.00	0.00	1.0
accounting	11991.0	0.051789	0.221610	0.00	0.00	0.00	0.00	1.0
hr	11991.0	0.050121	0.218204	0.00	0.00	0.00	0.00	1.0
management	11991.0	0.036361	0.187194	0.00	0.00	0.00	0.00	1.0
marketing	11991.0	0.056125	0.230173	0.00	0.00	0.00	0.00	1.0
product_mng	11991.0	0.057210	0.232252	0.00	0.00	0.00	0.00	1.0
sales	11991.0	0.270119	0.444040	0.00	0.00	0.00	1.00	1.0
support	11991.0	0.151864	0.358904	0.00	0.00	0.00	0.00	1.0
technical	11991.0	0.187140	0.390040	0.00	0.00	0.00	0.00	1.0

In [173…

```
from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler()

df_s = pd.DataFrame(sc.fit_transform(df_s), columns=df_s.columns, index=df_s.index)
```

In [174…

```
sc = MinMaxScaler()

X_train = pd.DataFrame(sc.fit_transform(X_train), columns=X.columns, index=X_train.index)
X_test = pd.DataFrame(sc.transform(X_test), columns=X.columns, index=X_test.index)
```

In [175…

```
df_s.describe().T
```

Out[175…

	count	mean	std	min	25%	50%	75%	max
satisfaction_level	11991.0	0.593031	0.264912	0.0	0.428571	0.626374	0.802198	1.0
last_evaluation	11991.0	0.557316	0.263035	0.0	0.328125	0.562500	0.781250	1.0
number_project	11991.0	0.360570	0.232648	0.0	0.200000	0.400000	0.600000	1.0
average_montly_hours	11991.0	0.488194	0.227700	0.0	0.285047	0.485981	0.686916	1.0
time_spend_company	11991.0	0.326954	0.283723	0.0	0.250000	0.250000	0.500000	1.0
Work_accident	11991.0	0.154282	0.361234	0.0	0.000000	0.000000	0.000000	1.0
left	11991.0	0.166041	0.372133	0.0	0.000000	0.000000	0.000000	1.0
promotion_last_5years	11991.0	0.016929	0.129012	0.0	0.000000	0.000000	0.000000	1.0
salary	11991.0	0.301935	0.317950	0.0	0.000000	0.500000	0.500000	1.0
RandD	11991.0	0.057877	0.233520	0.0	0.000000	0.000000	0.000000	1.0
accounting	11991.0	0.051789	0.221610	0.0	0.000000	0.000000	0.000000	1.0
hr	11991.0	0.050121	0.218204	0.0	0.000000	0.000000	0.000000	1.0
management	11991.0	0.036361	0.187194	0.0	0.000000	0.000000	0.000000	1.0
marketing	11991.0	0.056125	0.230173	0.0	0.000000	0.000000	0.000000	1.0
product_mng	11991.0	0.057210	0.232252	0.0	0.000000	0.000000	0.000000	1.0
sales	11991.0	0.270119	0.444040	0.0	0.000000	0.000000	1.000000	1.0
support	11991.0	0.151864	0.358904	0.0	0.000000	0.000000	0.000000	1.0

technical	11991.0	0.187140	0.390040	0.0	0.000000	0.000000	0.000000	1.0

In [176…

```
df_s
```

Out[176…

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5yea
0	0.318681	0.265625	0.0	0.285047	0.25	0.0	1.0	0
1	0.780220	0.781250	0.6	0.775701	1.00	0.0	1.0	0
2	0.021978	0.812500	1.0	0.822430	0.50	0.0	1.0	0
3	0.692308	0.796875	0.6	0.593458	0.75	0.0	1.0	0
4	0.307692	0.250000	0.0	0.294393	0.25	0.0	1.0	0
...	...	...	...	...	...	...	...	
11986	0.890110	0.296875	0.2	0.761682	1.00	1.0	0.0	1
11987	0.714286	0.921875	0.6	0.794393	1.00	0.0	0.0	1
11988	0.835165	0.281250	0.2	0.415888	1.00	0.0	0.0	1
11989	0.263736	0.453125	0.2	0.355140	1.00	0.0	0.0	1
11990	0.450549	0.578125	0.4	0.392523	0.25	0.0	0.0	0

11991 rows × 18 columns

A little EDA

In [177…

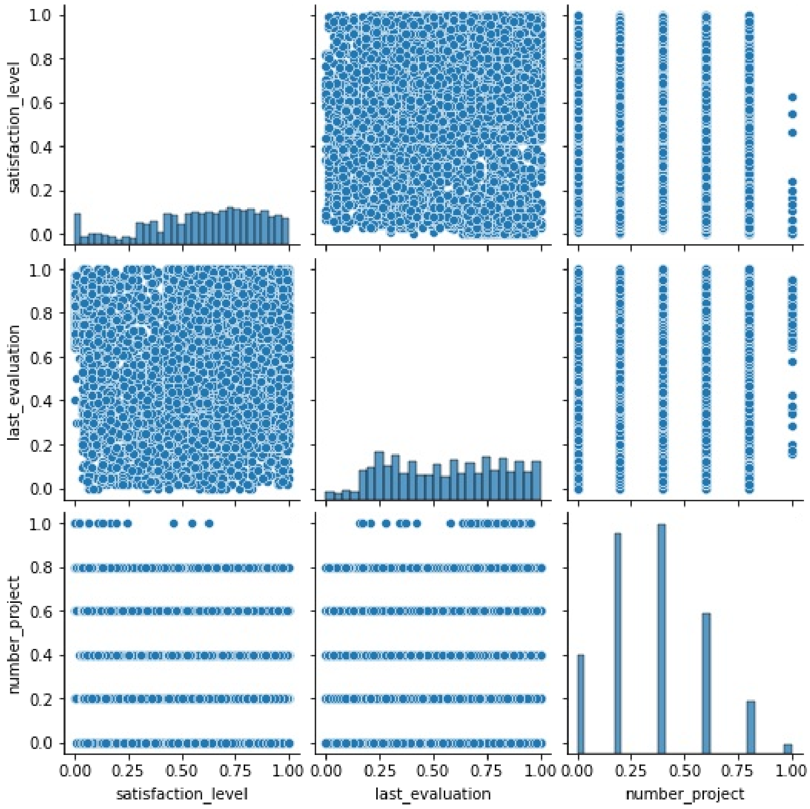
```
col = df_s.columns
col
```

Out[177…

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'salary', 'RandD', 'accounting', 'hr',
       'management', 'marketing', 'product_mng', 'sales', 'support',
       'technical'],
      dtype='object')
```

In [178…

```
ax = sns.pairplot(df_s[col_num])
```



In [182…

```
df_final = df_s.copy()
```

In [183…

```
df_final.to_csv('F:/O_C/T_U_C/dS_C9/7_Py(T)/3T/projects_classification/HR/cleanHRData.csv', index = False)
```

In [185…

```
X_train.to_csv('F:/O_C/T_U_C/dS_C9/7_Py(T)/3T/projects_classification/HR/X_train_cleanHRData.csv', index = False)
y_train.to_csv('F:/O_C/T_U_C/dS_C9/7_Py(T)/3T/projects_classification/HR/y_train_cleanHRData.csv', index = False)
X_test.to_csv('F:/O_C/T_U_C/dS_C9/7_Py(T)/3T/projects_classification/HR/X_test_cleanHRData.csv', index = False)
y_test.to_csv('F:/O_C/T_U_C/dS_C9/7_Py(T)/3T/projects_classification/HR/y_test_cleanHRData.csv', index = False)
```

In [ ]: