In [1]:

```python
import pandas as pd
```

In [2]:

```python
c=pd.read_csv("HR_comma_sep.csv")
c
```

Out[2]:

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

14999 rows × 10 columns

In [3]:

```python
c.head()
```

Out[3]:

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

In [4]:

```python
c.tail()
```

Out[4]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company
14994	0.40	0.57	2	151	3
14995	0.37	0.48	2	160	3
14996	0.37	0.53	2	143	3
14997	0.11	0.96	6	280	4

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company
14998	0.37	0.52	2	158	3

In [6]:

```
c.shape
```

Out[6]:

```
(14999, 10)
```

In [7]:

```
c.describe()
```

Out[7]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.00000
mean	0.612834	0.716102	3.803054	201.050337	3.498233
std	0.248631	0.171169	1.232592	49.943099	1.460136
min	0.090000	0.360000	2.000000	96.000000	2.00000
25%	0.440000	0.560000	3.000000	156.000000	3.00000
50%	0.640000	0.720000	4.000000	200.000000	3.00000
75%	0.820000	0.870000	5.000000	245.000000	4.00000
max	1.000000	1.000000	7.000000	310.000000	10.00000

In [8]:

```
c.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   satisfaction_level     14999 non-null  float64
 1   last_evaluation        14999 non-null  float64
 2   number_project         14999 non-null  int64  
 3   average_montly_hours   14999 non-null  int64  
 4   time_spend_company     14999 non-null  int64  
 5   Work_accident          14999 non-null  int64  
 6   left                   14999 non-null  int64  
 7   promotion_last_5years  14999 non-null  int64  
 8   Department             14999 non-null  object 
 9   salary                 14999 non-null  object 
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

In [9]:

```
c.isnull().sum()
```

Out[9]:

```
satisfaction_level       0
last_evaluation          0
number_project           0
average_montly_hours     0
time_spend_company       0
Work_accident            0
left                     0
promotion_last_5years    0
```

```
Department              0
salary                  0
dtype: int64
```

In [44]:

```
d=c.drop(['last_evaluation','number_project','time_spend_company','Work_accident','D
d
```

Out[44]:

	satisfaction_level	average_montly_hours	left	promotion_last_5years	salary
0	0.38	157	1	0	low
1	0.80	262	1	0	medium
2	0.11	272	1	0	medium
3	0.72	223	1	0	low
4	0.37	159	1	0	low
...	...	...	...	...	...
14994	0.40	151	1	0	low
14995	0.37	160	1	0	low
14996	0.37	143	1	0	low
14997	0.11	280	1	0	low
14998	0.37	158	1	0	low

14999 rows × 5 columns

In [45]:

```
d.head()
```

Out[45]:

	satisfaction_level	average_montly_hours	left	promotion_last_5years	salary
0	0.38	157	1	0	low
1	0.80	262	1	0	medium
2	0.11	272	1	0	medium
3	0.72	223	1	0	low
4	0.37	159	1	0	low

In [46]:

```
d=pd.get_dummies(d,dtype=int)
d
```

Out[46]:

	satisfaction_level	average_montly_hours	left	promotion_last_5years	salary_high	salary_low
0	0.38	157	1	0	0	1
1	0.80	262	1	0	0	0
2	0.11	272	1	0	0	0
3	0.72	223	1	0	0	1
4	0.37	159	1	0	0	1
...	...	...	...	...	...	...

	satisfaction_level	average_montly_hours	left	promotion_last_5years	salary_high	salary_low
14994	0.40	151	1	0	0	1
14995	0.37	160	1	0	0	1
14996	0.37	143	1	0	0	1
14997	0.11	280	1	0	0	1
14998	0.37	158	1	0	0	1

14999 rows × 7 columns

In [47]:

```
corr_mat=d.corr()
corr_mat
```

Out[47]:

	satisfaction_level	average_montly_hours	left	promotion_last_5years	sala
satisfaction_level	1.000000	-0.020048	-0.388375	0.025605	(
average_montly_hours	-0.020048	1.000000	0.071287	-0.003544	-(
left	-0.388375	0.071287	1.000000	-0.061788	-(
promotion_last_5years	0.025605	-0.003544	-0.061788	1.000000	(
salary_high	0.029708	-0.007101	-0.120929	0.076756	1
salary_low	-0.047415	-0.001050	0.134722	-0.082832	-(
salary_medium	0.031367	0.005007	-0.068833	0.040985	-(

In [48]:

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

In [49]:

```
y
```

Out[49]:

```
0        1
1        1
2        1
3        1
4        1
        ..
14994    1
14995    1
14996    1
14997    1
14998    1
Name: left, Length: 14999, dtype: int64
```

In [50]:

```
x
```

Out[50]:

	satisfaction_level	average_montly_hours	promotion_last_5years	salary_high	salary_low	salary
0	0.38	157	0	0	1	
1	0.80	262	0	0	0	

	satisfaction_level	average_montly_hours	promotion_last_5years	salary_high	salary_low	salary
2	0.11	272	0	0	0	
3	0.72	223	0	0	1	
4	0.37	159	0	0	1	
...	...	...	...	...	...	
14994	0.40	151	0	0	1	
14995	0.37	160	0	0	1	
14996	0.37	143	0	0	1	
14997	0.11	280	0	0	1	
14998	0.37	158	0	0	1	

14999 rows × 6 columns

In [51]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5,random_state=42)
```

In [52]:

```
from sklearn.linear_model import LogisticRegression
eve=LogisticRegression()
eve.fit(x_train,y_train)
```

Out[52]:

```
LogisticRegression()
```

In [53]:

```
ypred=eve.predict(x_test)
ypred
```

Out[53]:

```
array([0, 0, 0, ..., 0, 0, 1], dtype=int64)
```

In [54]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,ypred)
```

Out[54]:

```
array([[5345,  388],
       [1322,  445]], dtype=int64)
```

In [55]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,ypred)
```

Out[55]:

```
0.772
```

In [56]:

```
res=pd.DataFrame(columns=['left','predicted'])
res['left']=y_test
res['predicted']=ypred
res=res.reset_index()
res['ID']=res.index
```

In [57]:

```
res.head()
```

Out[57]:

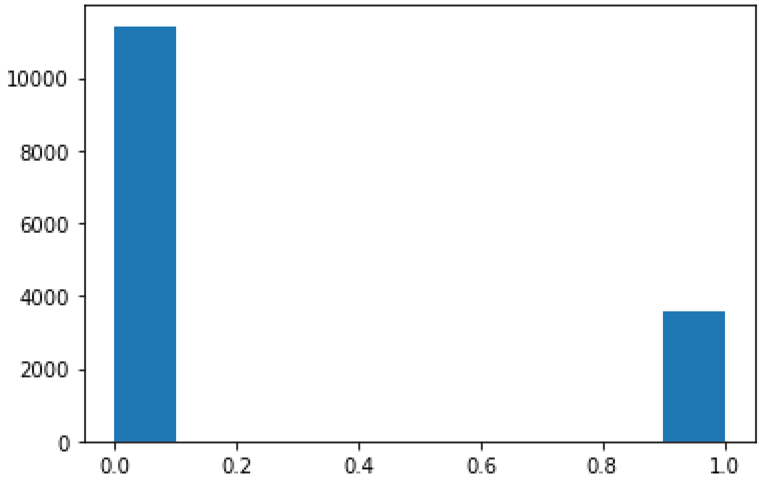
	index	left	predicted	ID
0	6723	0	0	0
1	6473	0	0	1
2	4679	0	0	2
3	862	1	0	3
4	7286	0	0	4

In [58]:

```
import matplotlib.pyplot as plt
plt.hist(d['left'])
```

Out[58]:

```
(array([11428.,     0.,     0.,     0.,     0.,     0.,     0.,     0.,
            0.,  3571.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```

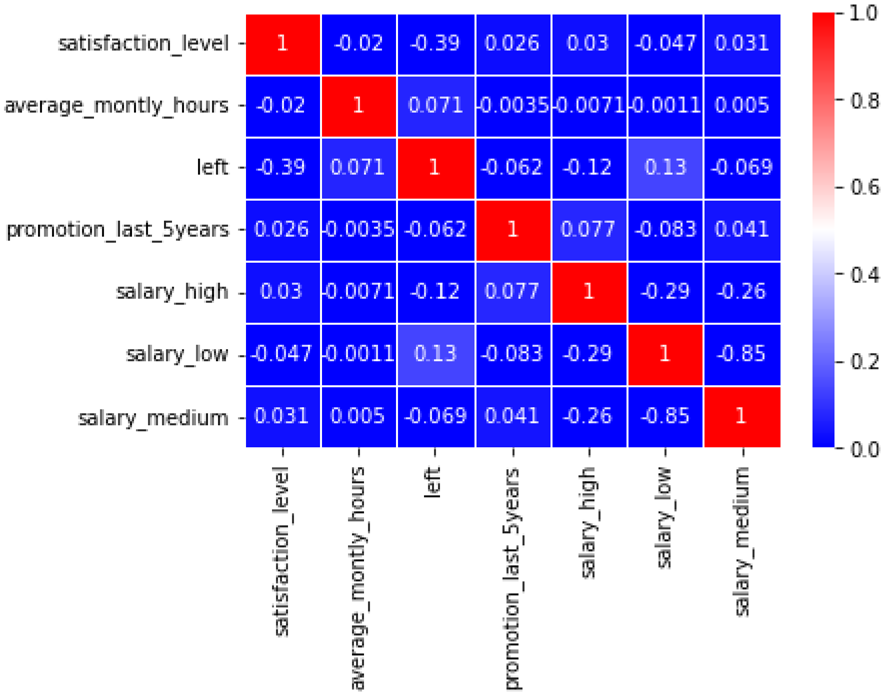


In [59]:

```
import seaborn as sns
sns.heatmap(corr_mat,vmax=1,vmin=0,annot=True,linewidth=1,cmap='bwr')
```

Out[59]:

```
<AxesSubplot:>
```



```
In [ ]:
```