In [1]:

```
# Importing Libraries

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

In [2]:

```
# loading and reading dataset

performance = pd.read_csv("Desktop/Datasets/case-study-hr-analytics-in-power-bi/
performance
```

Out[2]:

	ï»¿PerformanceID	EmployeeID	ReviewDate	EnvironmentSatisfaction	JobSatisfac
0	PR01	79F7-78EC	1/2/2013	5	
1	PR02	B61E-0F26	1/3/2013	5	
2	PR03	F5E3-48BB	1/3/2013	3	
3	PR04	0678-748A	1/4/2013	5	
4	PR05	541F-3E19	1/4/2013	5	
...	...	...	...	...	
6704	PR995	4F28-CFAF	3/14/2016	5	
6705	PR996	7C80-94E0	3/14/2016	3	
6706	PR997	8233-2483	3/14/2016	3	
6707	PR998	8A5B-3D6E	3/15/2016	5	
6708	PR999	4500-37EB	3/16/2016	4	

6709 rows × 11 columns

In [3]:

```
# Change column'name: 'ï»¿EmployeeID' to 'EmployeeID'

performance.rename(columns = {'ï»¿PerformanceID':'PerformanceID'}, inplace = Tru
```

In [4]:

```
# Check if column name changed

performance.columns
```

Out[4]:

```
Index(['PerformanceID', 'EmployeeID', 'ReviewDate', 'EnvironmentSatisfaction',
       'JobSatisfaction', 'RelationshipSatisfaction',
       'TrainingOpportunitiesWithinYear', 'TrainingOpportunitiesTaken',
       'WorkLifeBalance', 'SelfRating', 'ManagerRating'],
      dtype='object')
```

In [5]:

```
# Overall information about this dataset

performance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6709 entries, 0 to 6708
Data columns (total 11 columns):
 #   Column                           Non-Null Count  Dtype 
---  ------                           --------------  ----- 
 0   PerformanceID                    6709 non-null   object
 1   EmployeeID                       6709 non-null   object
 2   ReviewDate                       6709 non-null   object
 3   EnvironmentSatisfaction          6709 non-null   int64 
 4   JobSatisfaction                  6709 non-null   int64 
 5   RelationshipSatisfaction         6709 non-null   int64 
 6   TrainingOpportunitiesWithinYear  6709 non-null   int64 
 7   TrainingOpportunitiesTaken       6709 non-null   int64 
 8   WorkLifeBalance                  6709 non-null   int64 
 9   SelfRating                       6709 non-null   int64 
 10  ManagerRating                    6709 non-null   int64 
dtypes: int64(8), object(3)
memory usage: 576.7+ KB
```

In [6]:

```
# Dataset statistics for numerical columns

performance.describe()
```

Out[6]:

	EnvironmentSatisfaction	JobSatisfaction	RelationshipSatisfaction	TrainingOpport
count	6709.000000	6709.000000	6709.000000	
mean	3.872559	3.430616	3.427336	
std	0.940701	1.152565	1.156753	
min	1.000000	1.000000	1.000000	
25%	3.000000	2.000000	2.000000	
50%	4.000000	3.000000	3.000000	
75%	5.000000	4.000000	4.000000	
max	5.000000	5.000000	5.000000	

In [7]:

```
# Check if there is any missing value

performance.isnull().sum()
```

Out[7]:

```
PerformanceID                      0
EmployeeID                         0
ReviewDate                         0
EnvironmentSatisfaction            0
JobSatisfaction                    0
RelationshipSatisfaction           0
TrainingOpportunitiesWithinYear    0
TrainingOpportunitiesTaken         0
WorkLifeBalance                    0
SelfRating                         0
ManagerRating                      0
dtype: int64
```

In [8]:

```
# Check duplicates in Column 'Film'
```

```
performance.duplicated().sum()
```

Out[8]: 0

In [9]:
```
# Check top rows of dataset

performance.head()
```

Out[9]:

	PerformanceID	EmployeeID	ReviewDate	EnvironmentSatisfaction	JobSatisfaction
0	PR01	79F7-78EC	1/2/2013	5	4
1	PR02	B61E-0F26	1/3/2013	5	4
2	PR03	F5E3-48BB	1/3/2013	3	4
3	PR04	0678-748A	1/4/2013	5	3
4	PR05	541F-3E19	1/4/2013	5	2

In [10]:
```
# Check bottom rows of dataset

performance.tail()
```

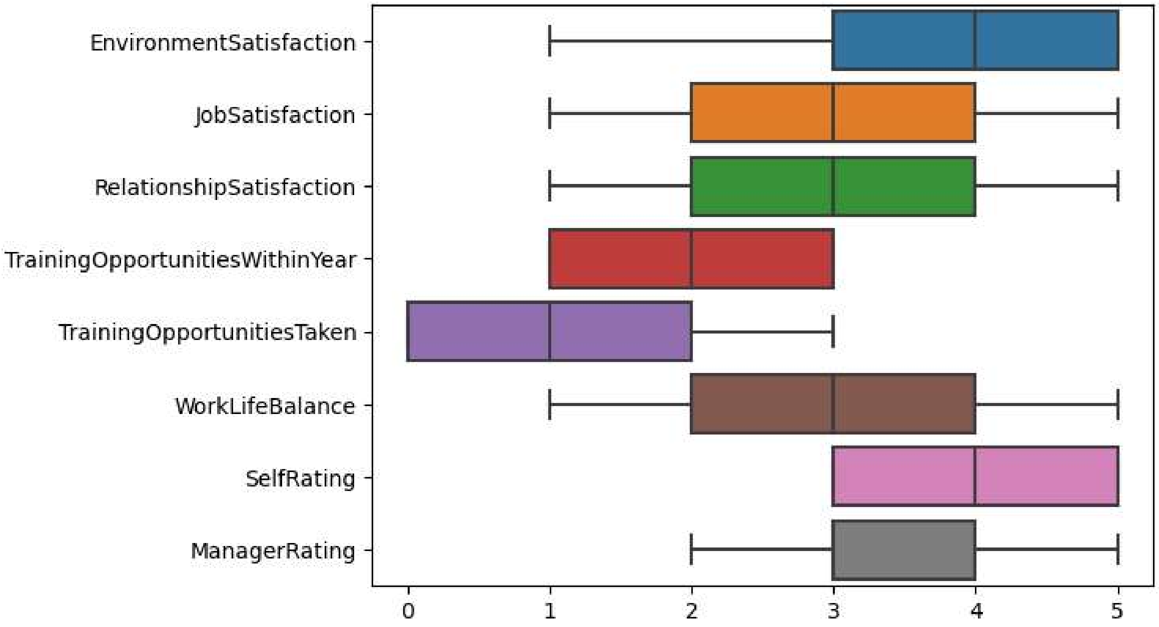
Out[10]:

	PerformanceID	EmployeeID	ReviewDate	EnvironmentSatisfaction	JobSatisfactio
6704	PR995	4F28-CFAF	3/14/2016	5	
6705	PR996	7C80-94E0	3/14/2016	3	
6706	PR997	8233-2483	3/14/2016	3	
6707	PR998	8A5B-3D6E	3/15/2016	5	
6708	PR999	4500-37EB	3/16/2016	4	

In [11]:
```
#Check outliers using Boxplot for numerical columns

sns.boxplot(performance,orient='h')
```

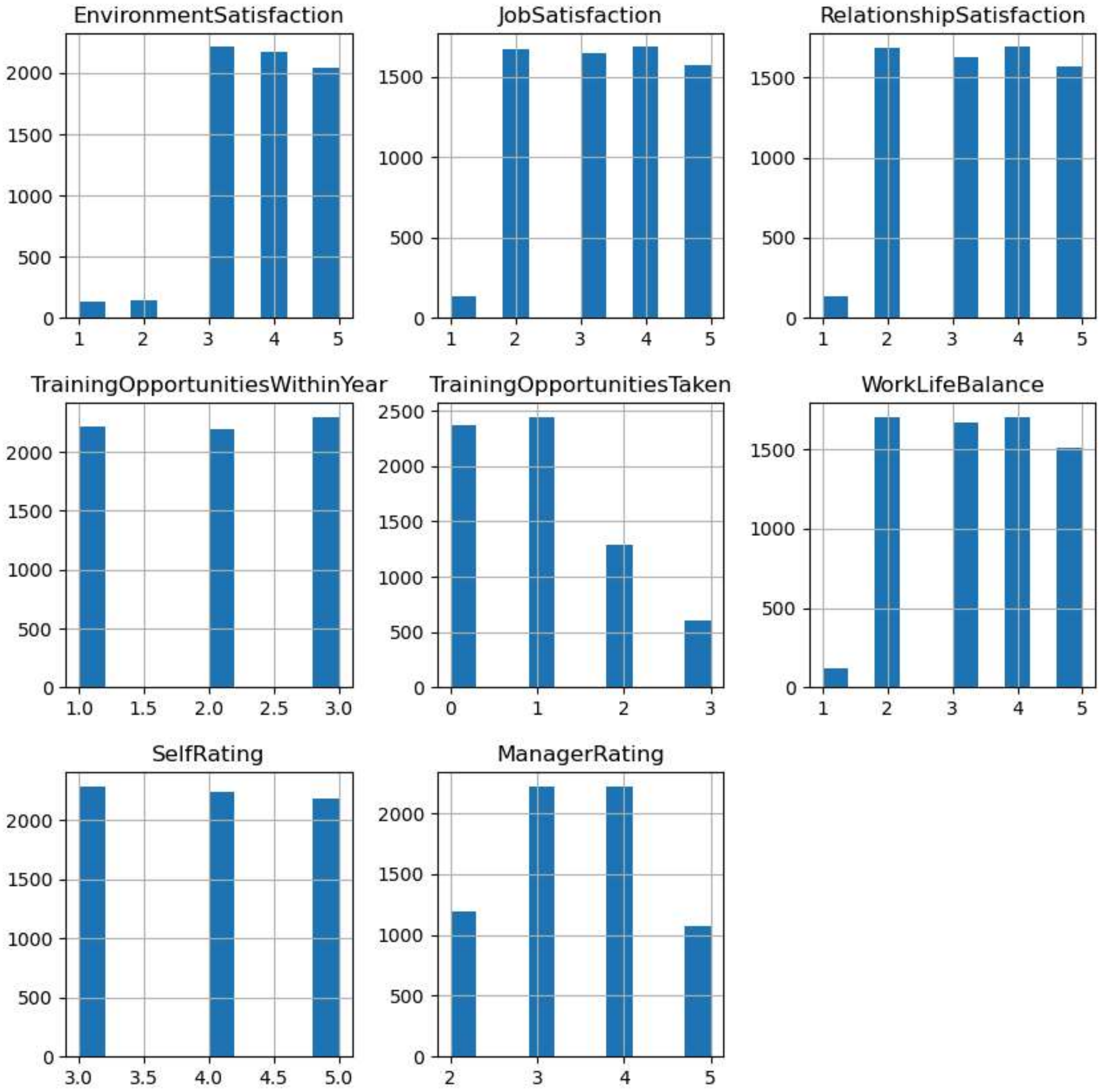
Out[11]: <Axes: >



```
In [12]: # check data distribution using histograms for numerical columns

         performance.hist(figsize=(10,10))
```

```
Out[12]: array([[<Axes: title={'center': 'EnvironmentSatisfaction'}>,
                 <Axes: title={'center': 'JobSatisfaction'}>,
                 <Axes: title={'center': 'RelationshipSatisfaction'}>],
                [<Axes: title={'center': 'TrainingOpportunitiesWithinYear'}>,
                 <Axes: title={'center': 'TrainingOpportunitiesTaken'}>,
                 <Axes: title={'center': 'WorkLifeBalance'}>],
                [<Axes: title={'center': 'SelfRating'}>,
                 <Axes: title={'center': 'ManagerRating'}>, <Axes: >]],
               dtype=object)
```



In [13]:

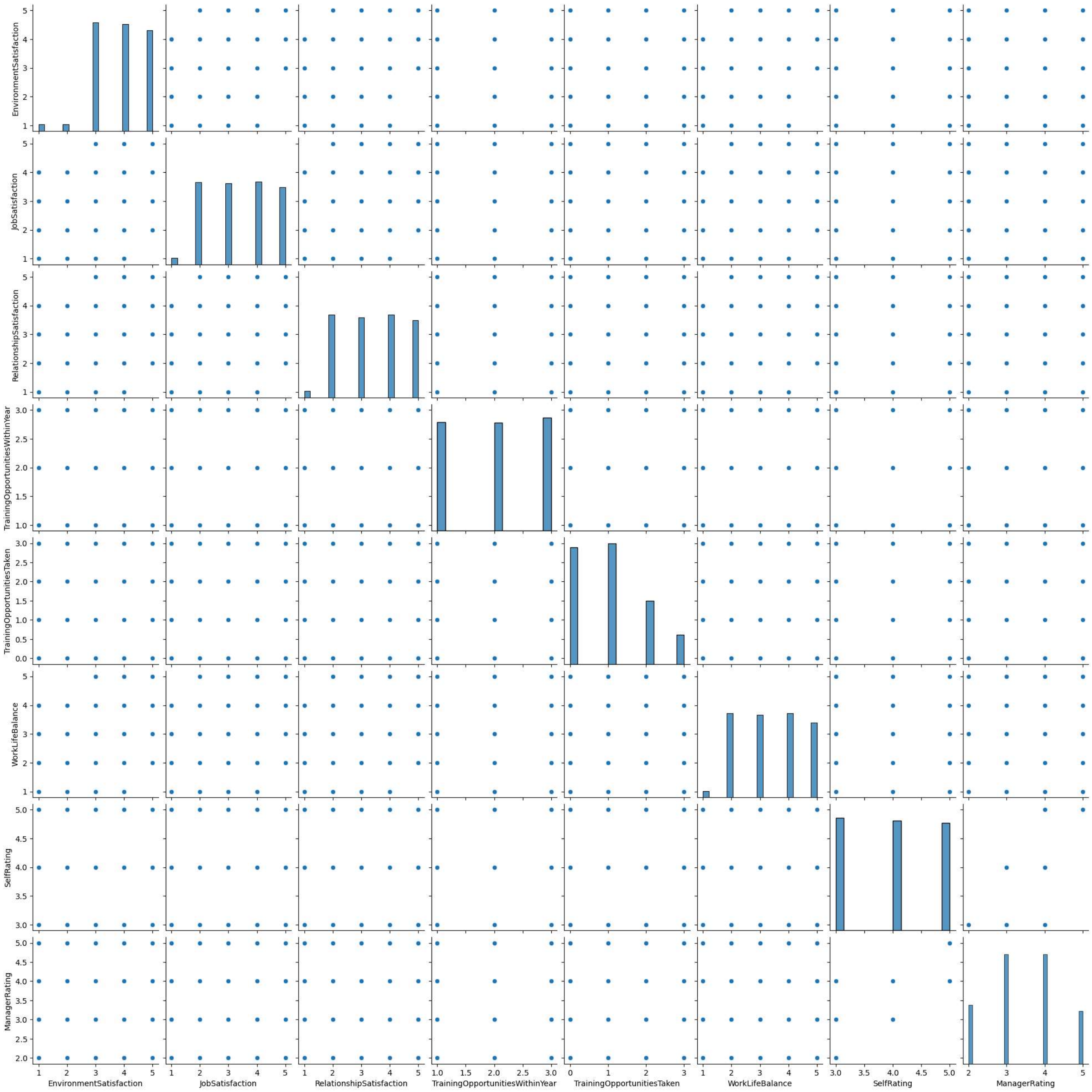
```
# Explore the relationships between variables

sns.pairplot(performance)
```

```
C:\Users\thaop\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[13]:

```
<seaborn.axisgrid.PairGrid at 0x232057eb650>
```



In [14]:

```
performance.to_csv('Atlas Performance.csv')
```