```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')
```

```
hr_data= pd.read_csv('HRData.csv')
print(hr_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64 
 9   EmployeeNumber            1470 non-null   int64 
 10  EnvironmentSatisfaction   1470 non-null   int64 
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64 
 13  JobInvolvement            1470 non-null   int64 
 14  JobLevel                  1470 non-null   int64 
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64 
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64 
 19  MonthlyRate               1470 non-null   int64 
 20  NumCompaniesWorked        1470 non-null   int64 
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64 
 24  PerformanceRating         1470 non-null   int64 
 25  RelationshipSatisfaction  1470 non-null   int64 
 26  StandardHours             1470 non-null   int64 
 27  StockOptionLevel          1470 non-null   int64 
 28  TotalWorkingYears         1470 non-null   int64 
 29  TrainingTimesLastYear     1470 non-null   int64 
 30  WorkLifeBalance           1470 non-null   int64 
 31  YearsAtCompany            1470 non-null   int64 
 32  YearsInCurrentRole        1470 non-null   int64 
 33  YearsSinceLastPromotion   1470 non-null   int64 
 34  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
None
```

```
print(hr_data.isnull().sum()) ##Since there are no null alues we dont have any NAN values to remove
```

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
```

```
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

```
hr_data.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumbe
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7

5 rows × 35 columns

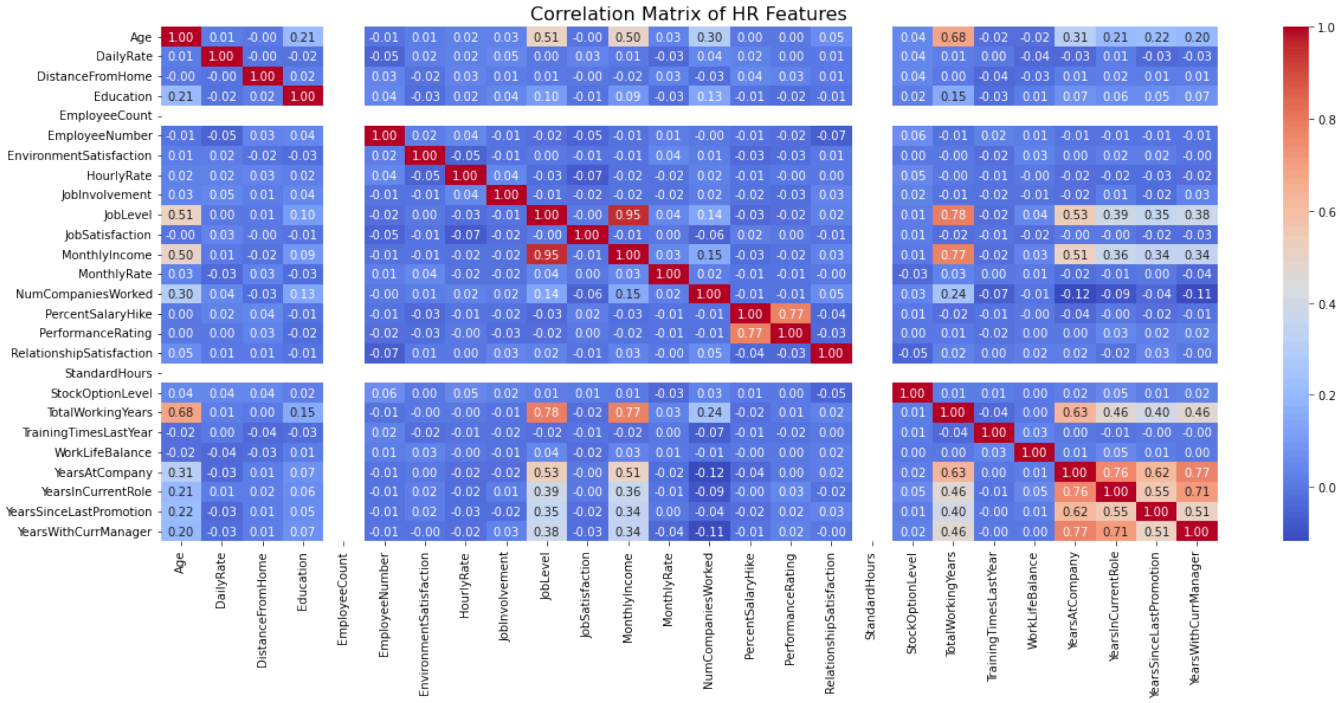
```
selected_columns=list(hr_data.columns)
corr_matrix = hr_data[selected_columns].corr()

# Set figure size for the heatmap
plt.figure(figsize=(20, 8))

# Create the heatmap using Seaborn
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')

# Add a title to the heatmap
plt.title('Correlation Matrix of HR Features', fontsize=16)

# Show the plot
plt.show()
```



```
#From the above correlation matrix for all numeric values we can keep those table columns which will be important for us and drop others
hr_data_new=hr_data.drop(["DailyRate","DistanceFromHome","Education","EmployeeCount","EnvironmentSatisfaction","HourlyRate","JobInvolver
```

```
hr_data_new.columns
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'EducationField',
       'EmployeeNumber', 'Gender', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked', 'OverTime',
       'PercentSalaryHike', 'PerformanceRating', 'TotalWorkingYears',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

```
hr_data_new.columns=["Age of Employee","Employee Attrition Needed","Business Travel", "Dept","Education Degree", "Emp. No.", "Gender","
```

```
##We will check if there are any duplicates on the basis of Emp. No. to check if there are any double entries of any employee; From the
duplicates_in_one_column = len(hr_data_new['Emp. No.']) - len(hr_data_new['Emp. No.'].drop_duplicates())
print(f"Number of duplicates on the basis of Emp. No. column: {duplicates_in_one_column}")
```

```
Number of duplicates on the basis of Emp. No. column: 0
```

```
hr_data_new.head()
```

	Age of Employee	Employee Attrition Needed	Business Travel	Dept	Education Degree	Emp. No.	Gender	Job Level	Role	Job Satisfaction Rate	...	Income per month	No. of Companies Worked
0	41	Yes	Travel_Rarely	Sales	Life Sciences	1	Female	2	Sales Executive	4	...	5993	8
1	49	No	Travel_Frequently	Research & Development	Life Sciences	2	Male	2	Research Scientist	2	...	5130	1
2	37	Yes	Travel_Rarely	Research & Development	Other	4	Male	1	Laboratory Technician	3	...	2090	6
3	33	No	Travel_Frequently	Research & Development	Life Sciences	5	Female	1	Research Scientist	3	...	2909	1
4	27	No	Travel_Rarely	Research & Development	Medical	7	Male	1	Laboratory Technician	2	...	3468	9

5 rows × 21 columns

```
#Standardizing of columns
columns_to_normalize = hr_data_new.select_dtypes(include=['float64', 'int64']).columns
train_X, test_X=train_test_split(hr_data_new[columns_to_normalize],test_size=0.3, random_state=1)
scaler=StandardScaler()
scaler.fit(train_X)
train_X=scaler.transform(train_X)
test_X=scaler.transform(test_X)
```

```
Q1 = hr_data_new.quantile(0.25)
Q3 = hr_data_new.quantile(0.75)
IQR = Q3 - Q1

outliers = ((hr_data_new < (Q1 - 1.5 * IQR)) | (hr_data_new > (Q3 + 1.5 * IQR))).any(axis=1)
hr_data_new_no_outliers = hr_data_new[~outliers]
hr_data_new_no_outliers.to_csv('cleaned_hr_data.csv', index=False)
```

Start coding or generate with AI.