```
!pip install pycaret
```

```
Requirement already satisfied: pandas<2.0.0,>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (1.5.3)
Requirement already satisfied: jinja2>=1.2 in /usr/local/lib/python3.10/dist-packages (from pycaret) (3.1.2)
Collecting scipy~=1.10.1 (from pycaret)
  Downloading scipy-1.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 41.2 MB/s eta 0:00:00
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (1.3.2)
Requirement already satisfied: scikit-learn<1.3.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (1.2.2)
Collecting pyod>=1.0.8 (from pycaret)
  Downloading pyod-1.1.1.tar.gz (159 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 159.4/159.4 kB 18.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: imbalanced-learn>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from pycaret) (0.10.1)
Collecting category-encoders>=2.4.0 (from pycaret)
  Downloading category_encoders-2.6.3-py2.py3-none-any.whl (81 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 81.9/81.9 kB 9.7 MB/s eta 0:00:00
Requirement already satisfied: lightgbm>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (4.1.0)
Requirement already satisfied: numba>=0.55.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (0.56.4)
Requirement already satisfied: requests>=2.27.1 in /usr/local/lib/python3.10/dist-packages (from pycaret) (2.31.0)
Requirement already satisfied: psutil>=5.9.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (5.9.5)
Requirement already satisfied: markupsafe>=2.0.1 in /usr/local/lib/python3.10/dist-packages (from pycaret) (2.1.3)
Requirement already satisfied: importlib-metadata>=4.12.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (6.8.0)
Requirement already satisfied: nbformat>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (5.9.2)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from pycaret) (2.2.1)
Collecting deprecation>=2.1.0 (from pycaret)
  Downloading deprecation-2.1.0-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from pycaret) (3.4.1)
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (3.7.1)
Collecting scikit-plot>=0.3.7 (from pycaret)
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: yellowbrick>=1.4 in /usr/local/lib/python3.10/dist-packages (from pycaret) (1.5)
Requirement already satisfied: plotly>=5.0.0 in /usr/local/lib/python3.10/dist-packages (from pycaret) (5.15.0)
Collecting kaleido>=0.2.1 (from pycaret)
  Downloading kaleido-0.2.1-py2.py3-none-manylinux1_x86_64.whl (79.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.9/79.9 MB 8.9 MB/s eta 0:00:00
Collecting schemdraw==0.15 (from pycaret)
  Downloading schemdraw-0.15-py3-none-any.whl (106 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 106.8/106.8 kB 9.3 MB/s eta 0:00:00
Collecting plotly-resampler>=0.8.3.1 (from pycaret)
  Downloading plotly_resampler-0.9.1-py3-none-any.whl (73 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 73.4/73.4 kB 7.8 MB/s eta 0:00:00
Requirement already satisfied: statsmodels>=0.12.1 in /usr/local/lib/python3.10/dist-packages (from pycaret) (0.14.0)
Collecting sktime!=0.17.1,!=0.17.2,!=0.18.0,<0.22.0,>=0.16.1 (from pycaret)
  Downloading sktime-0.21.1-py3-none-any.whl (17.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 51.7 MB/s eta 0:00:00
Collecting tbats>=1.1.3 (from pycaret)
  Downloading tbats-1.1.3-py3-none-any.whl (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.0/44.0 kB 5.6 MB/s eta 0:00:00
Collecting pmdarima!=1.8.1,<3.0.0,>=1.8.0 (from pycaret)
  Downloading pmdarima-2.0.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl (2.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 83.5 MB/s eta 0:00:00
Collecting wurlitzer (from pycaret)
  Downloading wurlitzer-3.0.3-py3-none-any.whl (7.3 kB)
Requirement already satisfied: patsy>=0.5.1 in /usr/local/lib/python3.10/dist-packages (from category-encoders>=2.4.0->pycaret) (0.5.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from deprecation>=2.1.0->pycaret) (23.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn>=0.8.1->pycaret) (3.2.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata>=4.12.0->pycaret) (3.17.0)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages (from ipython>=5.5.0->pycaret) (67.7.2)
```

```
import pandas as pd
```

```
data=pd.read_csv('data.csv')
data
```

	Age	Attrition	BusinessTravel	DistanceFromHome	Gender	HourlyRate	JobSatisfaction	MaritalStatus	MonthlyIncome	Over18	OverTime	TotalWorkingYears
0	41	No	Travel_Rarely	1	Female	94	4	Single	5993	Y	Yes	8
1	49	No	Travel_Frequently	8	Male	61	2	Married	5130	Y	No	10
2	37	Yes	Travel_Rarely	2	Male	92	3	Single	2090	Y	Yes	7
3	33	No	Travel_Frequently	3	Female	56	3	Married	2909	Y	Yes	8
4	27	No	Travel_Rarely	2	Male	40	2	Married	3468	Y	No	6
...	...	...	...	...	...	...	...	...	...	...	...	...
295	42	No	Travel_Frequently	26	Female	77	2	Married	13525	Y	No	23
296	18	No	Travel_Rarely	3	Male	54	3	Single	1420	Y	No	0
297	35	No	Travel_Rarely	16	Male	96	2	Married	8020	Y	No	12
298	36	No	Travel_Frequently	18	Male	81	4	Married	3688	Y	No	4
299	51	No	Travel_Rarely	2	Male	84	2	Divorced	5482	Y	No	13

300 rows × 12 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Age                300 non-null    int64 
 1   Attrition          300 non-null    object
 2   BusinessTravel     300 non-null    object
 3   DistanceFromHome   300 non-null    int64 
 4   Gender             300 non-null    object
 5   HourlyRate         300 non-null    int64 
 6   JobSatisfaction    300 non-null    int64 
 7   MaritalStatus      300 non-null    object
 8   MonthlyIncome      300 non-null    int64 
 9   Over18             300 non-null    object
 10  OverTime           300 non-null    object
 11  TotalWorkingYears  300 non-null    int64 
dtypes: int64(6), object(6)
memory usage: 28.2+ KB
```

```
data.drop('HourlyRate', axis=1, inplace=True)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Age                300 non-null    int64 
 1   Attrition          300 non-null    object
 2   BusinessTravel     300 non-null    object
 3   DistanceFromHome   300 non-null    int64 
 4   Gender             300 non-null    object
 5   JobSatisfaction    300 non-null    int64 
 6   MaritalStatus      300 non-null    object
 7   MonthlyIncome      300 non-null    int64 
 8   Over18             300 non-null    object
 9   OverTime           300 non-null    object
 10  TotalWorkingYears  300 non-null    int64 
```

```
dtypes: int64(5), object(6)
memory usage: 25.9+ KB
```

```
data['Attrition'].unique()
```

```
array(['No', 'Yes'], dtype=object)
```

```
data['BusinessTravel'].unique()
```

```
array(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel'], dtype=object)
```

```
data['Gender'].unique()
```

```
array(['Female', 'Male'], dtype=object)
```

```
data['MaritalStatus'].unique()
```

```
array(['Single', 'Married', 'Divorced'], dtype=object)
```

```
data['Over18'].unique()
```

```
array(['Y'], dtype=object)
```

```
data['OverTime'].unique()
```

```
array(['Yes', 'No'], dtype=object)
```

```
data['Age'].value_counts()
```

```
32    17
37    16
30    16
36    16
35    16
38    14
34    14
31    12
29    12
33    11
45    10
51    10
41     9
27     9
50     8
40     8
26     8
46     8
28     7
42     7
59     6
22     5
55     5
44     5
25     5
43     5
53     5
19     4
24     4
39     4
49     4
56     3
54     3
58     3
52     3
21     2
57     1
47     1
23     1
20     1
48     1
18     1
Name: Age, dtype: int64
```

```
data['DistanceFromHome'].value_counts()
```

```
1     49
2     39
6     20
3     20
9     19
5     16
7     14
4     14
8     12
23    10
20     7
26     7
18     7
11     7
29     6
10     6
16     6
24     5
21     5
19     5
14     5
15     4
27     3
25     3
12     3
22     3
28     2
17     2
13     1
Name: DistanceFromHome, dtype: int64
```

```
data['JobSatisfaction'].value_counts()
```

```
4    104
3     85
2     57
1     54
Name: JobSatisfaction, dtype: int64
```

```
data['MonthlyIncome'].value_counts()
```

```
5993    2
3038    2
2293    2
3072    2
2911    2
       ..
2926    1
2956    1
2073    1
2042    1
5482    1
Name: MonthlyIncome, Length: 293, dtype: int64
```

```
data['TotalWorkingYears'].value_counts()
```

```
10    37
6     25
5     21
7     20
9     19
8     17
1     15
12    15
17    14
16    12
4     11
13     9
21     8
2      8
22     7
23     7
11     6
3      5
20     5
28     4
14     4
15     4
19     4
0      3
24     3
25     3
18     2
37     2
31     1
29     1
38     1
30     1
40     1
26     1
36     1
34     1
32     1
33     1
Name: TotalWorkingYears, dtype: int64
```

```
import matplotlib.pyplot as plt
import seaborn as sns
f,ax = plt.subplots(figsize=(8,8))
sns.distplot(data['JobSatisfaction'])
plt.xlim([0,6])
```

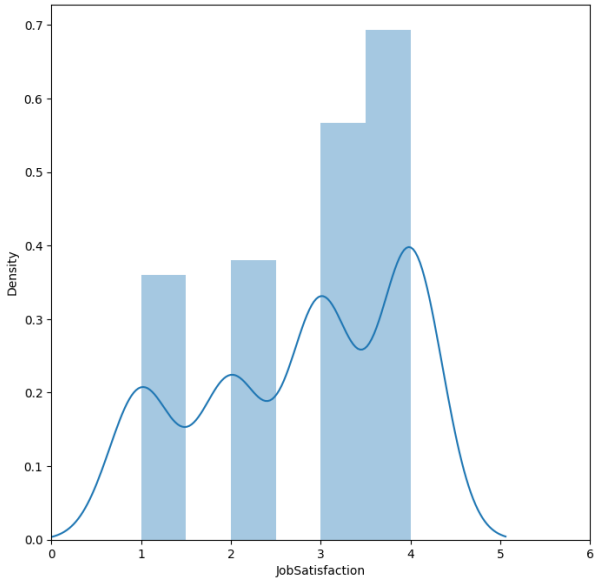
```
<ipython-input-46-86672421bd1a>:4: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['JobSatisfaction'])
(0.0, 6.0)
```

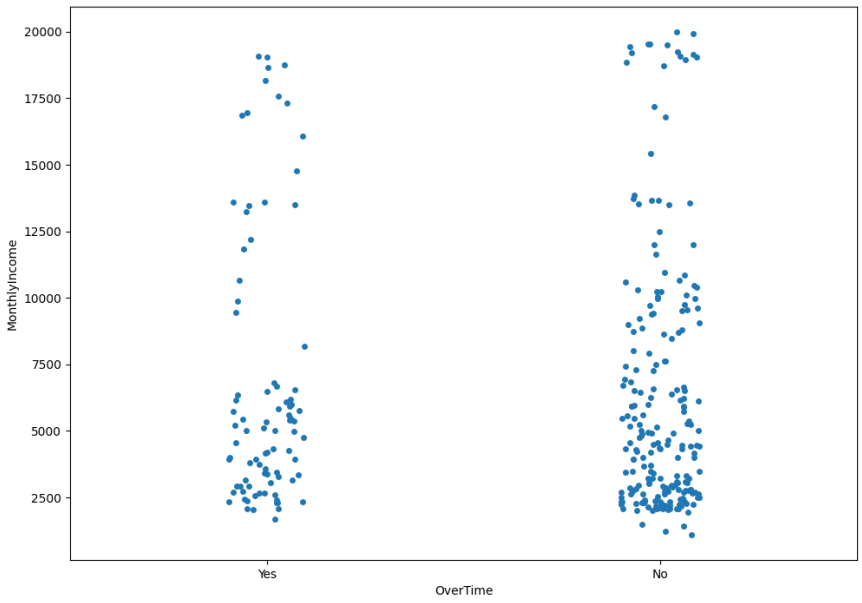



```
var = 'OverTime'
fig, ax = plt.subplots()
fig.set_size_inches(11.7, 8.27)
sns.swarmplot(x = var, y ='MonthlyIncome', data = data)
```

```
<Axes: xlabel='OverTime', ylabel='MonthlyIncome'>
```


```
var = 'OverTime'
fig, ax = plt.subplots()
fig.set_size_inches(11.7, 8.27)
sns.stripplot(x = var, y ='MonthlyIncome', data = data)
```

```
<Axes: xlabel='OverTime', ylabel='MonthlyIncome'>
```



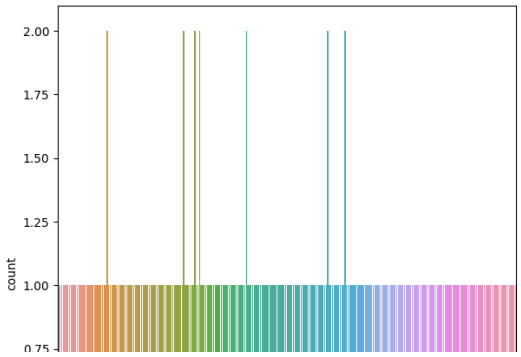

```
var = "OverTime"
plt.figure(figsize=(20, 10))
sns.catplot(x=var, kind="count", palette="ch:.25", height=8, aspect=2, data=data);
plt.xticks(rotation=360);
```

```
<Figure size 2000x1000 with 0 Axes>
```

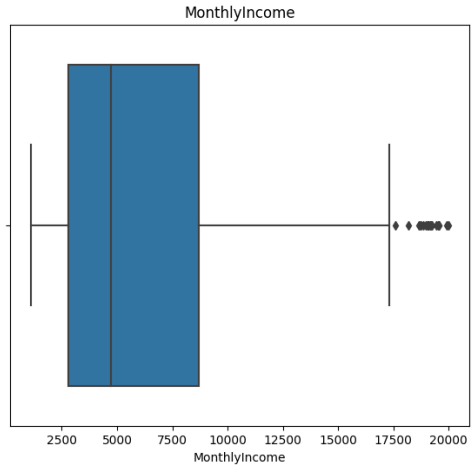

```
#missing values
data.isna().sum()
```

```
MonthlyIncome    0
OverTime         0
dtype: int64
```

```
fig = plt.figure(figsize=[15,8])
ax1 = fig.add_subplot(121)
sns.countplot(x="MonthlyIncome", data=data)
plt.show()
```

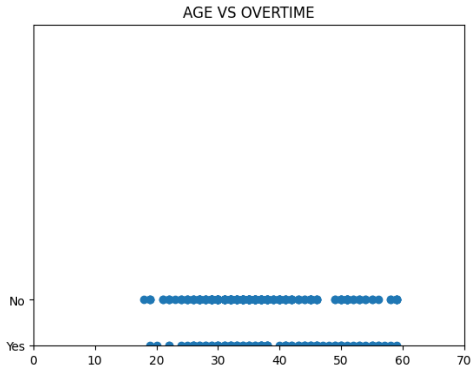



```
fig = plt.figure(figsize=[15,6])
ax1 = fig.add_subplot(121)
ax1.title.set_text('MonthlyIncome')
sns.boxplot(x='MonthlyIncome', data=data)
plt.show()
```




```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

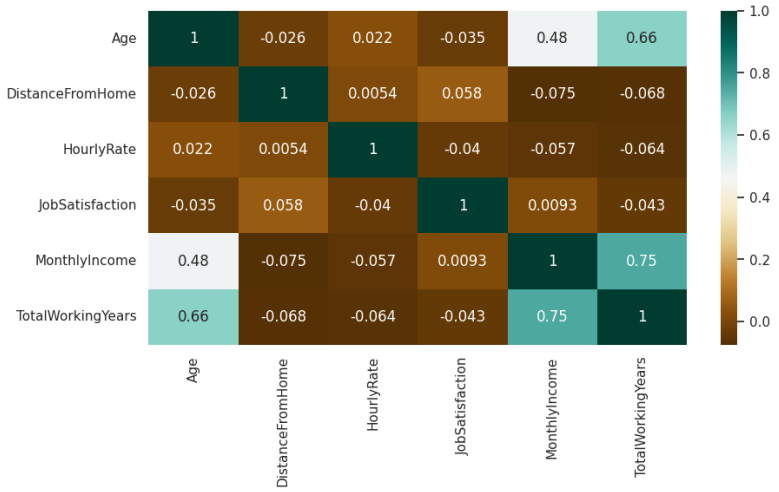
```
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
import pandas as pd
X=pd.read_csv("data.csv")
x1 = X['Age'].values
x2 = X['OverTime'].values
X = np.array(list(zip(x1, x2))).reshape(len(x1), 2)
plt.plot()
plt.xlim([0,70])
plt.ylim([0,7])
plt.title('AGE VS OVERTIME')
plt.scatter(x1, x2)
plt.show()
```




```
import pandas as pd
import numpy as np
import seaborn as sns                       #visualisation
import matplotlib.pyplot as plt             #visualisation
%matplotlib inline
sns.set(color_codes=True)
df = pd.read_csv("data.csv")
duplicate_rows_df = df[df.duplicated()]
print("number of duplicate rows: ", duplicate_rows_df.shape)
df.count()
df = df.drop_duplicates()
df.head(5)
print(df.count())
print(df.isnull().sum())
df = df.dropna()    # Dropping the missing values.
print(df.count())
print(df.isnull().sum())   # After dropping the values
plt.figure(figsize=(10,5))
c= df.corr()
sns.heatmap(c,cmap="BrBG",annot=True)
fig, ax = plt.subplots(figsize=(10,6))
ax.scatter(df['OverTime'], df['MonthlyIncome'])
plt.xticks(rotation=360);
ax.set_xlabel('OverTime')
```

```
ax.set_ylabel('MonthlyIncome')
plt.xticks(rotation=360);
plt.show()
```

```
    number of duplicate rows:  (0, 12)
    Age                  300
    Attrition            300
    BusinessTravel       300
    DistanceFromHome     300
    Gender               300
    HourlyRate           300
    JobSatisfaction      300
    MaritalStatus        300
    MonthlyIncome        300
    Over18               300
    OverTime             300
    TotalWorkingYears    300
    dtype: int64
    Age                  0
    Attrition            0
    BusinessTravel       0
    DistanceFromHome     0
    Gender               0
    HourlyRate           0
    JobSatisfaction      0
    MaritalStatus        0
    MonthlyIncome        0
    Over18               0
    OverTime             0
    TotalWorkingYears    0
    dtype: int64
    Age                  300
    Attrition            300
    BusinessTravel       300
    DistanceFromHome     300
    Gender               300
    HourlyRate           300
    JobSatisfaction      300
    MaritalStatus        300
    MonthlyIncome        300
    Over18               300
    OverTime             300
    TotalWorkingYears    300
    dtype: int64
    Age                  0
    Attrition            0
    BusinessTravel       0
    DistanceFromHome     0
    Gender               0
    HourlyRate           0
    JobSatisfaction      0
    MaritalStatus        0
    MonthlyIncome        0
    Over18               0
    OverTime             0
    TotalWorkingYears    0
    dtype: int64
    <ipython-input-100-f60b8dac0808>:19: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the v
      c= df.corr()
```

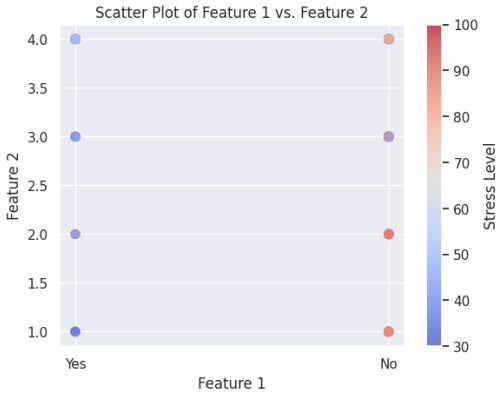



```
import matplotlib.pyplot as plt

# Assuming you have two numerical features, 'Feature1' and 'Feature2'
feature1 = df['OverTime']
feature2 = df['JobSatisfaction']

# Plot a scatter plot
plt.scatter(feature1, feature2, c=df['HourlyRate'], cmap='coolwarm', s=50, alpha=0.7)
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Scatter Plot of Feature 1 vs. Feature 2')
```

```
plt.colorbar(label='Stress Level')
plt.show()
```




```
plt.hist(df['HourlyRate'], bins=10, color='skyblue', alpha=0.7)
plt.title('Distribution of Stress Levels')
plt.xlabel('Stress Level')
plt.ylabel('Frequency')
plt.show()
```