Importing Libraries

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

Loading Dataset

In [3]:
```
pd.set_option('display.max_columns', None)
burnoutDF=pd.read_csv('burnoutPredData.csv')
burnoutDF
```

Out[3]:

	Employee ID	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate
0	fffe32003000360033003200	2008-09-30	Female	Service	No	2.0	3.0	3.8	0.16
1	fffe3700360033003500	2008-11-30	Male	Service	Yes	1.0	2.0	5.0	0.36
2	fffe31003300320037003900	2008-03-10	Female	Product	Yes	2.0	NaN	5.8	0.49
3	fffe32003400380032003900	2008-11-03	Male	Service	Yes	1.0	1.0	2.6	0.20
4	fffe31003900340031003600	2008-07-24	Female	Service	No	3.0	7.0	6.9	0.52
...	...	...	...	...	...	...	...	...	...
22745	fffe31003500370039003100	2008-12-30	Female	Service	No	1.0	3.0	NaN	0.41
22746	fffe33003000350031003800	2008-01-19	Female	Product	Yes	3.0	6.0	6.7	0.59
22747	fffe390032003000	2008-11-05	Male	Service	Yes	3.0	7.0	NaN	0.72
22748	fffe33003300320036003900	2008-01-10	Female	Service	No	2.0	5.0	5.9	0.52
22749	fffe3400350031003800	2008-01-06	Male	Product	No	3.0	6.0	7.8	0.61

22750 rows × 9 columns

In [4]:
```
#Convert into datetime datatype
burnoutDF["Date of Joining"] = pd.to_datetime(burnoutDF["Date of Joining"])
```

In [5]:
```
#Give the number of row and columns
burnoutDF.shape
```

Out[5]:
```
(22750, 9)
```

In [6]:
```
#General Information
burnoutDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22750 entries, 0 to 22749
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Employee ID           22750 non-null  object
 1   Date of Joining       22750 non-null  datetime64[ns]
 2   Gender                22750 non-null  object
 3   Company Type          22750 non-null  object
 4   WFH Setup Available   22750 non-null  object
 5   Designation           22750 non-null  float64
 6   Resource Allocation   21369 non-null  float64
 7   Mental Fatigue Score  20633 non-null  float64
 8   Burn Rate             21626 non-null  float64
dtypes: datetime64[ns](1), float64(4), object(4)
memory usage: 1.6+ MB
```

In [7]:
```
#Show top 5 rows
burnoutDF.head()
```

Out[7]:

	Employee ID	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate
0	fffe32003000360033003200	2008-09-30	Female	Service	No	2.0	3.0	3.8	0.16
1	fffe3700360033003500	2008-11-30	Male	Service	Yes	1.0	2.0	5.0	0.36
2	fffe31003300320037003900	2008-03-10	Female	Product	Yes	2.0	NaN	5.8	0.49
3	fffe32003400380032003900	2008-11-03	Male	Service	Yes	1.0	1.0	2.6	0.20
4	fffe31003900340031003600	2008-07-24	Female	Service	No	3.0	7.0	6.9	0.52

In [8]:
```
#Extract all the columns of the dataset
burnoutDF.columns
```

Out[8]:
```
Index(['Employee ID', 'Date of Joining', 'Gender', 'Company Type',
       'WFH Setup Available', 'Designation', 'Resource Allocation',
       'Mental Fatigue Score', 'Burn Rate'],
      dtype='object')
```

In [9]:
```
#check for null values
burnoutDF.isna().sum()
```

Out[9]:
```
Employee ID               0
Date of Joining           0
Gender                    0
Company Type              0
WFH Setup Available       0
Designation               0
Resource Allocation     1381
Mental Fatigue Score    2117
Burn Rate               1124
dtype: int64
```

In [10]:
```
#check the duplicate values
burnoutDF.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
#Calculate the sum, std, min, max and count of every attributes
burnoutDF.describe()
```

Out[11]:

	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate
count	22750.000000	21369.000000	20633.000000	21626.000000
mean	2.178725	4.481398	5.728188	0.452005
std	1.135145	2.047211	1.920839	0.198226
min	0.000000	1.000000	0.000000	0.000000
25%	1.000000	3.000000	4.600000	0.310000
50%	2.000000	4.000000	5.900000	0.450000
75%	3.000000	6.000000	7.100000	0.590000
max	5.000000	10.000000	10.000000	1.000000

In [12]:
```
#show the unique values
for i, col in enumerate(burnoutDF.columns):
    print(f"\n\n{burnoutDF[col].unique()}")
    print(f"\n{burnoutDF[col].value_counts()}\n\n")
```

```
['fffe32003000360033003200' 'fffe3700360033003500'
 'fffe31003300320037003900' ... 'fffe390032003000'
 'fffe33003300320036003900' 'fffe3400350031003800']

fffe32003000360033003200    1
fffe3600360035003500        1
fffe3800360034003400        1
fffe31003000310033003600    1
fffe31003400350031003700    1
                           ..
fffe33003400340032003400    1
fffe32003100370036003600    1
fffe31003900310035003800    1
fffe32003400320034003200    1
fffe3400350031003800        1
Name: Employee ID, Length: 22750, dtype: int64
```

```
['2008-09-30T00:00:00.000000000' '2008-11-30T00:00:00.000000000'
 '2008-03-10T00:00:00.000000000' '2008-11-03T00:00:00.000000000'
 '2008-07-24T00:00:00.000000000' '2008-11-26T00:00:00.000000000'
 '2008-01-02T00:00:00.000000000' '2008-10-31T00:00:00.000000000'
 '2008-12-27T00:00:00.000000000' '2008-03-09T00:00:00.000000000'
 '2008-03-16T00:00:00.000000000' '2008-05-12T00:00:00.000000000'
 '2008-01-20T00:00:00.000000000' '2008-02-23T00:00:00.000000000'
 '2008-05-14T00:00:00.000000000' '2008-02-03T00:00:00.000000000'
 '2008-03-17T00:00:00.000000000' '2008-03-28T00:00:00.000000000'
 '2008-05-29T00:00:00.000000000' '2008-06-27T00:00:00.000000000'
 '2008-08-31T00:00:00.000000000' '2008-01-15T00:00:00.000000000'
 '2008-05-04T00:00:00.000000000' '2008-11-17T00:00:00.000000000'
 '2008-09-14T00:00:00.000000000' '2008-10-09T00:00:00.000000000'
 '2008-10-11T00:00:00.000000000' '2008-09-18T00:00:00.000000000'
 '2008-09-16T00:00:00.000000000' '2008-12-16T00:00:00.000000000'
 '2008-05-03T00:00:00.000000000' '2008-08-04T00:00:00.000000000'
 '2008-07-31T00:00:00.000000000' '2008-06-17T00:00:00.000000000'
 '2008-04-28T00:00:00.000000000' '2008-10-30T00:00:00.000000000'
 '2008-02-27T00:00:00.000000000' '2008-06-22T00:00:00.000000000'
 '2008-02-18T00:00:00.000000000' '2008-06-24T00:00:00.000000000'
 '2008-12-08T00:00:00.000000000' '2008-08-05T00:00:00.000000000'
 '2008-04-11T00:00:00.000000000' '2008-03-26T00:00:00.000000000'
 '2008-08-09T00:00:00.000000000' '2008-08-28T00:00:00.000000000'
 '2008-03-21T00:00:00.000000000' '2008-07-22T00:00:00.000000000'
 '2008-05-20T00:00:00.000000000' '2008-01-23T00:00:00.000000000'
 '2008-09-10T00:00:00.000000000' '2008-05-26T00:00:00.000000000'
 '2008-12-22T00:00:00.000000000' '2008-04-08T00:00:00.000000000'
```

```
'2008-02-25T00:00:00.000000000' '2008-04-24T00:00:00.000000000'
'2008-01-08T00:00:00.000000000' '2008-11-20T00:00:00.000000000'
'2008-09-11T00:00:00.000000000' '2008-06-11T00:00:00.000000000'
'2008-02-28T00:00:00.000000000' '2008-08-20T00:00:00.000000000'
'2008-10-18T00:00:00.000000000' '2008-08-14T00:00:00.000000000'
'2008-07-17T00:00:00.000000000' '2008-07-05T00:00:00.000000000'
'2008-02-04T00:00:00.000000000' '2008-08-01T00:00:00.000000000'
'2008-05-01T00:00:00.000000000' '2008-05-21T00:00:00.000000000'
'2008-10-21T00:00:00.000000000' '2008-03-19T00:00:00.000000000'
'2008-09-27T00:00:00.000000000' '2008-03-12T00:00:00.000000000'
'2008-09-17T00:00:00.000000000' '2008-02-13T00:00:00.000000000'
'2008-09-19T00:00:00.000000000' '2008-07-03T00:00:00.000000000'
'2008-10-27T00:00:00.000000000' '2008-01-22T00:00:00.000000000'
'2008-04-15T00:00:00.000000000' '2008-10-26T00:00:00.000000000'
'2008-01-31T00:00:00.000000000' '2008-01-03T00:00:00.000000000'
'2008-03-13T00:00:00.000000000' '2008-03-27T00:00:00.000000000'
'2008-11-15T00:00:00.000000000' '2008-08-17T00:00:00.000000000'
'2008-08-08T00:00:00.000000000' '2008-06-28T00:00:00.000000000'
'2008-05-06T00:00:00.000000000' '2008-12-17T00:00:00.000000000'
'2008-09-08T00:00:00.000000000' '2008-07-04T00:00:00.000000000'
'2008-10-28T00:00:00.000000000' '2008-02-19T00:00:00.000000000'
'2008-02-11T00:00:00.000000000' '2008-03-02T00:00:00.000000000'
'2008-08-10T00:00:00.000000000' '2008-01-04T00:00:00.000000000'
'2008-10-12T00:00:00.000000000' '2008-11-14T00:00:00.000000000'
'2008-09-02T00:00:00.000000000' '2008-10-04T00:00:00.000000000'
'2008-05-31T00:00:00.000000000' '2008-03-03T00:00:00.000000000'
'2008-02-21T00:00:00.000000000' '2008-12-04T00:00:00.000000000'
'2008-09-05T00:00:00.000000000' '2008-02-24T00:00:00.000000000'
'2008-12-09T00:00:00.000000000' '2008-01-19T00:00:00.000000000'
'2008-01-26T00:00:00.000000000' '2008-05-10T00:00:00.000000000'
'2008-05-16T00:00:00.000000000' '2008-05-07T00:00:00.000000000'
'2008-10-16T00:00:00.000000000' '2008-07-09T00:00:00.000000000'
'2008-03-11T00:00:00.000000000' '2008-08-15T00:00:00.000000000'
'2008-08-25T00:00:00.000000000' '2008-12-14T00:00:00.000000000'
'2008-04-26T00:00:00.000000000' '2008-04-03T00:00:00.000000000'
'2008-12-19T00:00:00.000000000' '2008-08-13T00:00:00.000000000'
'2008-03-08T00:00:00.000000000' '2008-02-05T00:00:00.000000000'
'2008-02-17T00:00:00.000000000' '2008-04-16T00:00:00.000000000'
'2008-10-24T00:00:00.000000000' '2008-03-05T00:00:00.000000000'
'2008-09-25T00:00:00.000000000' '2008-03-01T00:00:00.000000000'
'2008-05-23T00:00:00.000000000' '2008-09-07T00:00:00.000000000'
'2008-03-23T00:00:00.000000000' '2008-01-25T00:00:00.000000000'
'2008-12-29T00:00:00.000000000' '2008-06-15T00:00:00.000000000'
'2008-10-03T00:00:00.000000000' '2008-01-17T00:00:00.000000000'
'2008-01-30T00:00:00.000000000' '2008-10-13T00:00:00.000000000'
'2008-02-08T00:00:00.000000000' '2008-11-25T00:00:00.000000000'
'2008-04-23T00:00:00.000000000' '2008-11-07T00:00:00.000000000'
'2008-06-20T00:00:00.000000000' '2008-12-23T00:00:00.000000000'
```

```
'2008-11-24T00:00:00.000000000' '2008-06-21T00:00:00.000000000'
'2008-11-29T00:00:00.000000000' '2008-08-11T00:00:00.000000000'
'2008-04-29T00:00:00.000000000' '2008-11-19T00:00:00.000000000'
'2008-12-25T00:00:00.000000000' '2008-02-14T00:00:00.000000000'
'2008-03-04T00:00:00.000000000' '2008-10-06T00:00:00.000000000'
'2008-08-16T00:00:00.000000000' '2008-10-29T00:00:00.000000000'
'2008-07-15T00:00:00.000000000' '2008-04-21T00:00:00.000000000'
'2008-09-01T00:00:00.000000000' '2008-01-06T00:00:00.000000000'
'2008-03-20T00:00:00.000000000' '2008-04-14T00:00:00.000000000'
'2008-02-16T00:00:00.000000000' '2008-10-10T00:00:00.000000000'
'2008-09-26T00:00:00.000000000' '2008-06-01T00:00:00.000000000'
'2008-07-11T00:00:00.000000000' '2008-07-23T00:00:00.000000000'
'2008-07-10T00:00:00.000000000' '2008-10-05T00:00:00.000000000'
'2008-03-14T00:00:00.000000000' '2008-06-14T00:00:00.000000000'
'2008-10-23T00:00:00.000000000' '2008-02-22T00:00:00.000000000'
'2008-05-19T00:00:00.000000000' '2008-09-20T00:00:00.000000000'
'2008-01-18T00:00:00.000000000' '2008-07-13T00:00:00.000000000'
'2008-11-04T00:00:00.000000000' '2008-12-05T00:00:00.000000000'
'2008-07-27T00:00:00.000000000' '2008-12-07T00:00:00.000000000'
'2008-06-04T00:00:00.000000000' '2008-09-09T00:00:00.000000000'
'2008-11-01T00:00:00.000000000' '2008-01-28T00:00:00.000000000'
'2008-04-04T00:00:00.000000000' '2008-07-06T00:00:00.000000000'
'2008-12-28T00:00:00.000000000' '2008-07-08T00:00:00.000000000'
'2008-01-21T00:00:00.000000000' '2008-10-19T00:00:00.000000000'
'2008-01-07T00:00:00.000000000' '2008-12-24T00:00:00.000000000'
'2008-06-09T00:00:00.000000000' '2008-09-13T00:00:00.000000000'
'2008-10-14T00:00:00.000000000' '2008-11-08T00:00:00.000000000'
'2008-12-26T00:00:00.000000000' '2008-05-08T00:00:00.000000000'
'2008-08-12T00:00:00.000000000' '2008-08-24T00:00:00.000000000'
'2008-09-21T00:00:00.000000000' '2008-11-10T00:00:00.000000000'
'2008-01-09T00:00:00.000000000' '2008-05-18T00:00:00.000000000'
'2008-10-08T00:00:00.000000000' '2008-09-22T00:00:00.000000000'
'2008-08-06T00:00:00.000000000' '2008-04-30T00:00:00.000000000'
'2008-12-20T00:00:00.000000000' '2008-04-13T00:00:00.000000000'
'2008-04-12T00:00:00.000000000' '2008-11-18T00:00:00.000000000'
'2008-02-15T00:00:00.000000000' '2008-06-07T00:00:00.000000000'
'2008-11-16T00:00:00.000000000' '2008-06-26T00:00:00.000000000'
'2008-05-11T00:00:00.000000000' '2008-09-03T00:00:00.000000000'
'2008-03-06T00:00:00.000000000' '2008-09-24T00:00:00.000000000'
'2008-04-01T00:00:00.000000000' '2008-05-25T00:00:00.000000000'
'2008-05-22T00:00:00.000000000' '2008-01-13T00:00:00.000000000'
'2008-06-06T00:00:00.000000000' '2008-01-16T00:00:00.000000000'
'2008-03-22T00:00:00.000000000' '2008-04-20T00:00:00.000000000'
'2008-02-02T00:00:00.000000000' '2008-10-01T00:00:00.000000000'
'2008-10-07T00:00:00.000000000' '2008-06-03T00:00:00.000000000'
'2008-11-12T00:00:00.000000000' '2008-08-26T00:00:00.000000000'
'2008-05-17T00:00:00.000000000' '2008-12-30T00:00:00.000000000'
'2008-06-19T00:00:00.000000000' '2008-11-22T00:00:00.000000000'
```

```
'2008-05-13T00:00:00.000000000' '2008-03-30T00:00:00.000000000'
'2008-06-16T00:00:00.000000000' '2008-04-27T00:00:00.000000000'
'2008-07-01T00:00:00.000000000' '2008-12-15T00:00:00.000000000'
'2008-09-06T00:00:00.000000000' '2008-04-19T00:00:00.000000000'
'2008-01-12T00:00:00.000000000' '2008-12-02T00:00:00.000000000'
'2008-01-24T00:00:00.000000000' '2008-07-02T00:00:00.000000000'
'2008-08-29T00:00:00.000000000' '2008-07-29T00:00:00.000000000'
'2008-06-29T00:00:00.000000000' '2008-01-11T00:00:00.000000000'
'2008-11-09T00:00:00.000000000' '2008-07-30T00:00:00.000000000'
'2008-08-23T00:00:00.000000000' '2008-06-05T00:00:00.000000000'
'2008-09-23T00:00:00.000000000' '2008-06-18T00:00:00.000000000'
'2008-01-14T00:00:00.000000000' '2008-12-06T00:00:00.000000000'
'2008-01-10T00:00:00.000000000' '2008-06-13T00:00:00.000000000'
'2008-07-18T00:00:00.000000000' '2008-07-28T00:00:00.000000000'
'2008-07-26T00:00:00.000000000' '2008-01-01T00:00:00.000000000'
'2008-08-27T00:00:00.000000000' '2008-08-30T00:00:00.000000000'
'2008-04-10T00:00:00.000000000' '2008-07-14T00:00:00.000000000'
'2008-09-28T00:00:00.000000000' '2008-04-02T00:00:00.000000000'
'2008-10-15T00:00:00.000000000' '2008-06-30T00:00:00.000000000'
'2008-03-07T00:00:00.000000000' '2008-10-22T00:00:00.000000000'
'2008-08-02T00:00:00.000000000' '2008-03-15T00:00:00.000000000'
'2008-03-18T00:00:00.000000000' '2008-05-28T00:00:00.000000000'
'2008-02-09T00:00:00.000000000' '2008-08-22T00:00:00.000000000'
'2008-11-02T00:00:00.000000000' '2008-04-22T00:00:00.000000000'
'2008-11-21T00:00:00.000000000' '2008-02-12T00:00:00.000000000'
'2008-02-07T00:00:00.000000000' '2008-07-19T00:00:00.000000000'
'2008-11-23T00:00:00.000000000' '2008-07-21T00:00:00.000000000'
'2008-08-21T00:00:00.000000000' '2008-11-11T00:00:00.000000000'
'2008-12-13T00:00:00.000000000' '2008-04-25T00:00:00.000000000'
'2008-11-05T00:00:00.000000000' '2008-08-19T00:00:00.000000000'
'2008-04-17T00:00:00.000000000' '2008-08-07T00:00:00.000000000'
'2008-12-31T00:00:00.000000000' '2008-05-27T00:00:00.000000000'
'2008-09-29T00:00:00.000000000' '2008-05-30T00:00:00.000000000'
'2008-12-18T00:00:00.000000000' '2008-02-20T00:00:00.000000000'
'2008-12-11T00:00:00.000000000' '2008-11-27T00:00:00.000000000'
'2008-07-20T00:00:00.000000000' '2008-11-28T00:00:00.000000000'
'2008-08-03T00:00:00.000000000' '2008-10-20T00:00:00.000000000'
'2008-07-07T00:00:00.000000000' '2008-06-08T00:00:00.000000000'
'2008-03-24T00:00:00.000000000' '2008-12-21T00:00:00.000000000'
'2008-04-09T00:00:00.000000000' '2008-05-05T00:00:00.000000000'
'2008-06-12T00:00:00.000000000' '2008-04-18T00:00:00.000000000'
'2008-01-27T00:00:00.000000000' '2008-10-17T00:00:00.000000000'
'2008-05-09T00:00:00.000000000' '2008-03-29T00:00:00.000000000'
'2008-09-12T00:00:00.000000000' '2008-07-25T00:00:00.000000000'
'2008-04-07T00:00:00.000000000' '2008-05-02T00:00:00.000000000'
'2008-06-02T00:00:00.000000000' '2008-10-02T00:00:00.000000000'
'2008-02-26T00:00:00.000000000' '2008-07-12T00:00:00.000000000'
'2008-02-06T00:00:00.000000000' '2008-06-23T00:00:00.000000000'
```

```
 '2008-11-06T00:00:00.000000000' '2008-07-16T00:00:00.000000000'
 '2008-06-25T00:00:00.000000000' '2008-01-29T00:00:00.000000000'
 '2008-02-29T00:00:00.000000000' '2008-03-25T00:00:00.000000000'
 '2008-08-18T00:00:00.000000000' '2008-04-05T00:00:00.000000000'
 '2008-05-15T00:00:00.000000000' '2008-12-12T00:00:00.000000000'
 '2008-10-25T00:00:00.000000000' '2008-04-06T00:00:00.000000000'
 '2008-11-13T00:00:00.000000000' '2008-09-04T00:00:00.000000000'
 '2008-05-24T00:00:00.000000000' '2008-06-10T00:00:00.000000000'
 '2008-03-31T00:00:00.000000000' '2008-12-01T00:00:00.000000000'
 '2008-01-05T00:00:00.000000000' '2008-09-15T00:00:00.000000000'
 '2008-12-10T00:00:00.000000000' '2008-02-10T00:00:00.000000000'
 '2008-12-03T00:00:00.000000000' '2008-02-01T00:00:00.000000000']

2008-01-06    86
2008-05-21    85
2008-02-04    82
2008-07-16    81
2008-07-13    80
              ..
2008-06-27    44
2008-07-06    44
2008-07-04    43
2008-12-24    43
2008-12-07    39
Name: Date of Joining, Length: 366, dtype: int64



['Female' 'Male']

Female    11908
Male      10842
Name: Gender, dtype: int64



['Service' 'Product']

Service    14833
Product     7917
Name: Company Type, dtype: int64



['No' 'Yes']
```

```
Yes    12290
No     10460
Name: WFH Setup Available, dtype: int64
```

```
[2. 1. 3. 0. 4. 5.]

2.0    7588
3.0    5985
1.0    4881
4.0    2391
0.0    1507
5.0     398
Name: Designation, dtype: int64
```

```
[ 3.  2. nan  1.  7.  4.  6.  5.  8. 10.  9.]

4.0     3893
5.0     3861
3.0     3192
6.0     2943
2.0     2075
7.0     1965
1.0     1791
8.0     1044
9.0      446
10.0     159
Name: Resource Allocation, dtype: int64
```

```
[ 3.8  5.   5.8  2.6  6.9  3.6  7.9  4.4  nan  5.3  1.8  4.7  5.9  6.7
  4.   7.6  6.3  7.7  6.6  7.4  3.9  3.   8.7  7.3  5.4  6.   7.5 10.
  6.4  5.1  5.6  6.1  3.1  8.   6.8  4.9  9.2  6.5  6.2  8.2  4.1  4.3
  0.8  2.9  2.   9.1  0.   5.7  8.3  5.5  7.   3.3  7.8  7.2  5.2  8.9
  4.5  8.1  8.6  9.5  3.5  4.8  2.4  3.7  1.   8.8  9.3  4.6  9.9  0.5
  2.8  9.   3.4  4.2  1.6  2.7  1.3  3.2  8.4  7.1  9.4  2.1  9.7  2.5
  1.9  1.7  9.6  0.7  0.2  1.2  8.5  9.8  2.2  1.1  0.9  2.3  0.4  1.4
  1.5  0.6  0.3  0.1]

6.0    470
```

```
5.8    464
5.9    458
6.1    457
6.3    454
      ...
0.5     24
0.2     23
0.4     19
0.1     17
0.3     13
Name: Mental Fatigue Score, Length: 101, dtype: int64
```

```
[0.16 0.36 0.49 0.2  0.52 0.29 0.62 0.33 0.56 0.67 0.5  0.12 0.4  0.51
 0.32 0.39 0.59 0.22 0.68 0.57 0.47 0.46 0.61 0.91 0.44 0.6  0.45 0.19
 0.31 0.81 0.42 0.53  nan 0.94 0.37 0.65 0.38 0.15 0.26 0.28 0.71 0.8
 0.63 0.79 0.72 0.34 0.27 0.66 0.04 0.05 0.11 0.41 0.76 0.43 0.85 0.35
 0.   0.55 0.48 0.7  0.18 0.23 0.25 0.75 0.1  0.73 0.58 0.88 0.77 0.3
 0.06 0.03 0.69 0.24 0.74 0.86 0.92 0.78 0.21 0.98 0.02 0.82 0.93 0.83
 0.87 0.64 0.54 0.17 1.   0.08 0.09 0.14 0.13 0.07 0.84 0.99 0.01 0.97
 0.95 0.9  0.96 0.89]

0.47    475
0.43    444
0.41    434
0.45    431
0.50    428
       ...
0.98     18
0.97     17
0.95     17
0.96     13
0.99      8
Name: Burn Rate, Length: 101, dtype: int64
```

In [13]:
```
#Drop irrelevant column
burnoutDF=burnoutDF.drop(['Employee ID'],axis=1)
```

In [14]:
```
#Check the skewness of the attributes
intFloatburnoutDF=burnoutDF.select_dtypes([np.int, np.float])
for i, col in enumerate(intFloatburnoutDF.columns):
    if (intFloatburnoutDF[col].skew() >= 0.1):
        print("\n",col, "feature is Positively Skewed and value is: ", intFloatburnoutDF[col].skew())
    elif (intFloatburnoutDF[col].skew() <= -0.1):
```

```
        print("\n",col, "feature is Negatively Skewed and value is: ", intFloatburnoutDF[col].skew())
    else:
        print("\n",col, "feature is Normally Distributed and value is: ", intFloatburnoutDF[col].skew())
```

```
Designation feature is Normally Distributed and value is:  0.09242138478903683

Resource Allocation feature is Positively Skewed and value is:  0.20457273454318103

Mental Fatigue Score feature is Negatively Skewed and value is:  -0.4308950578815428

Burn Rate feature is Normally Distributed and value is:  0.045737370909640515
```

In [15]:
```
#Replace the null values with mean
burnoutDF['Resource Allocation'].fillna(burnoutDF['Resource Allocation'].mean(),inplace=True)
burnoutDF['Mental Fatigue Score'].fillna(burnoutDF['Mental Fatigue Score'].mean(),inplace=True)
burnoutDF['Burn Rate'].fillna(burnoutDF['Burn Rate'].mean(),inplace=True)
```

In [16]:
```
#check for null values
burnoutDF.isna().sum()
```

Out[16]:
```
Date of Joining         0
Gender                  0
Company Type            0
WFH Setup Available     0
Designation             0
Resource Allocation     0
Mental Fatigue Score    0
Burn Rate               0
dtype: int64
```

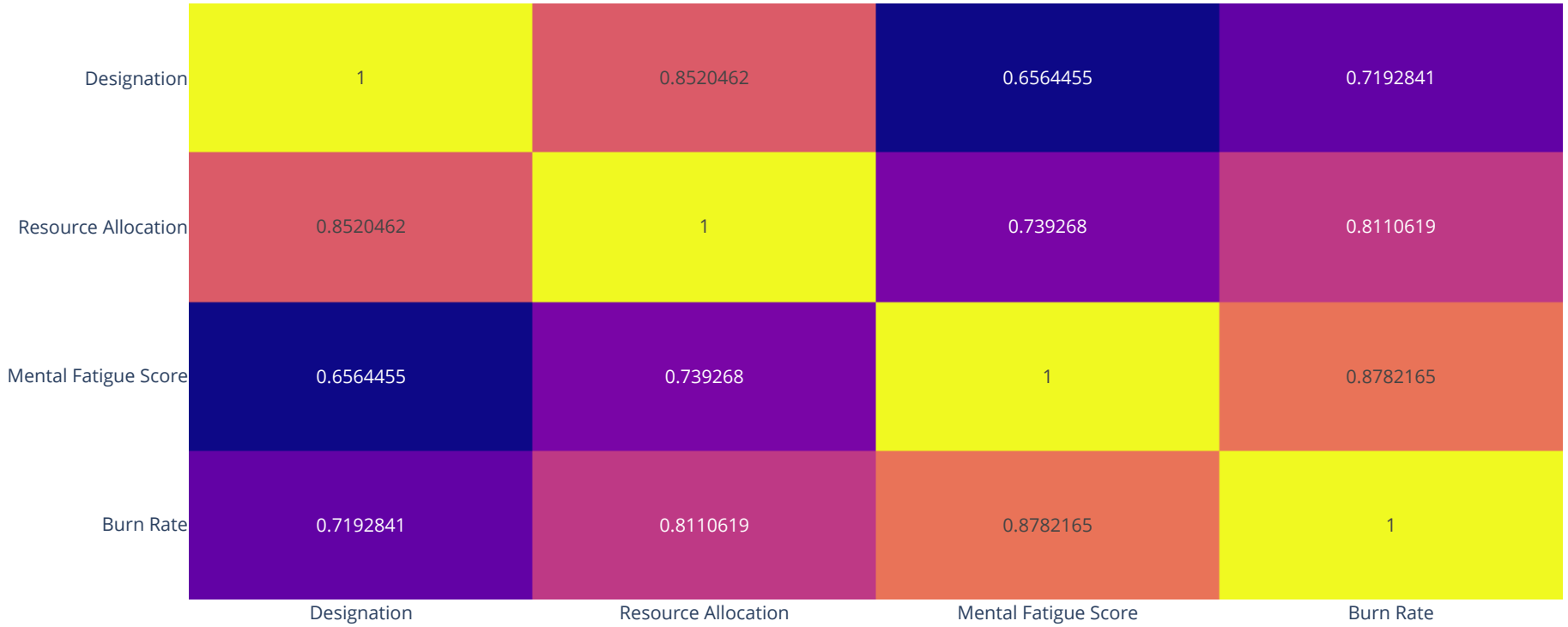
In [17]:
```
#show the correlation
burnoutDF.corr()
```

Out[17]:

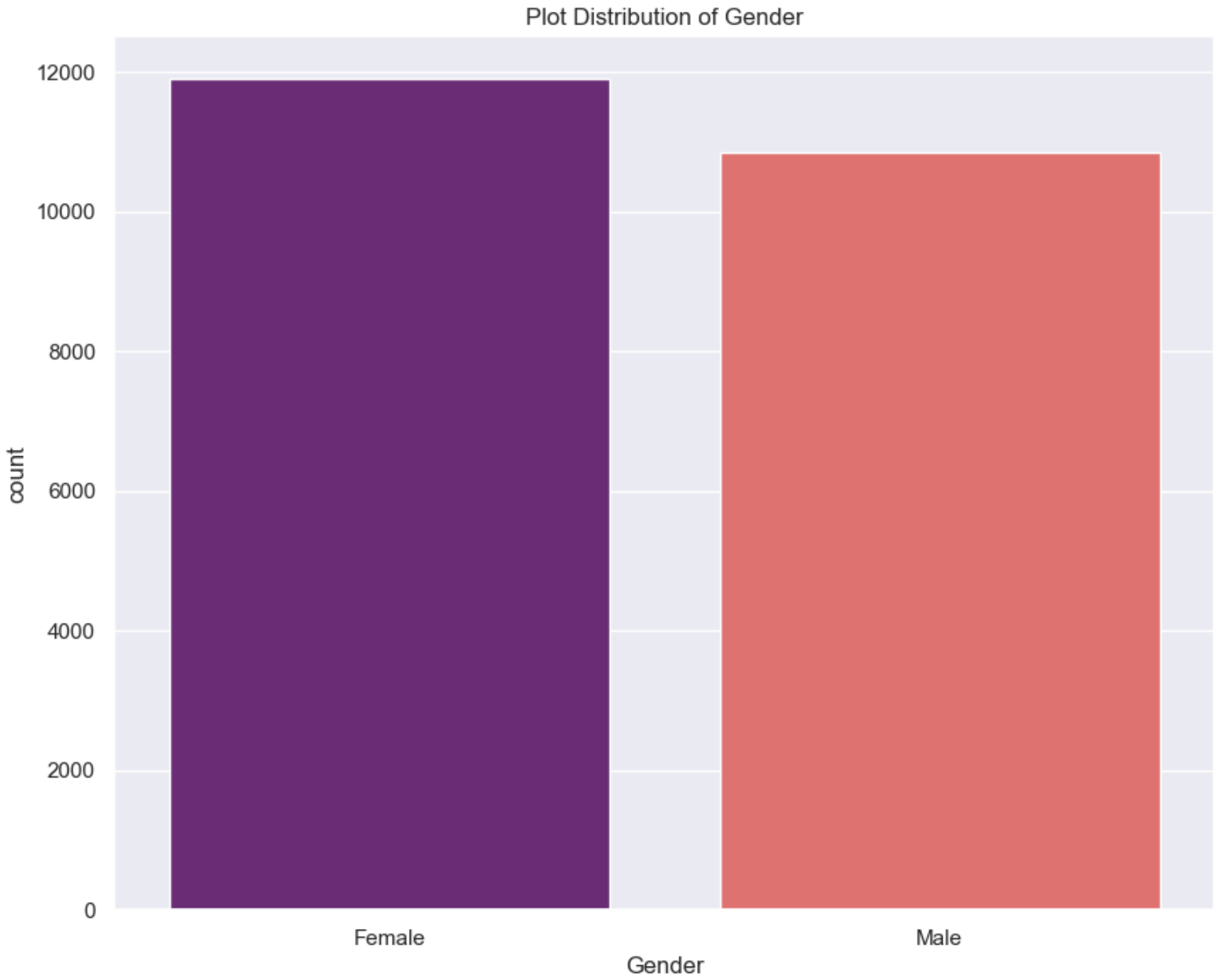
	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate
Designation	1.000000	0.852046	0.656445	0.719284
Resource Allocation	0.852046	1.000000	0.739268	0.811062
Mental Fatigue Score	0.656445	0.739268	1.000000	0.878217
Burn Rate	0.719284	0.811062	0.878217	1.000000

Data Visualization

In [18]:
```
#plotting Heat Map to check correlation
Corr=burnoutDF.corr()
sns.set(rc={'figure.figsize':(14,12)})
fig=px.imshow(Corr, text_auto=True, aspect="auto")
fig.show()
```

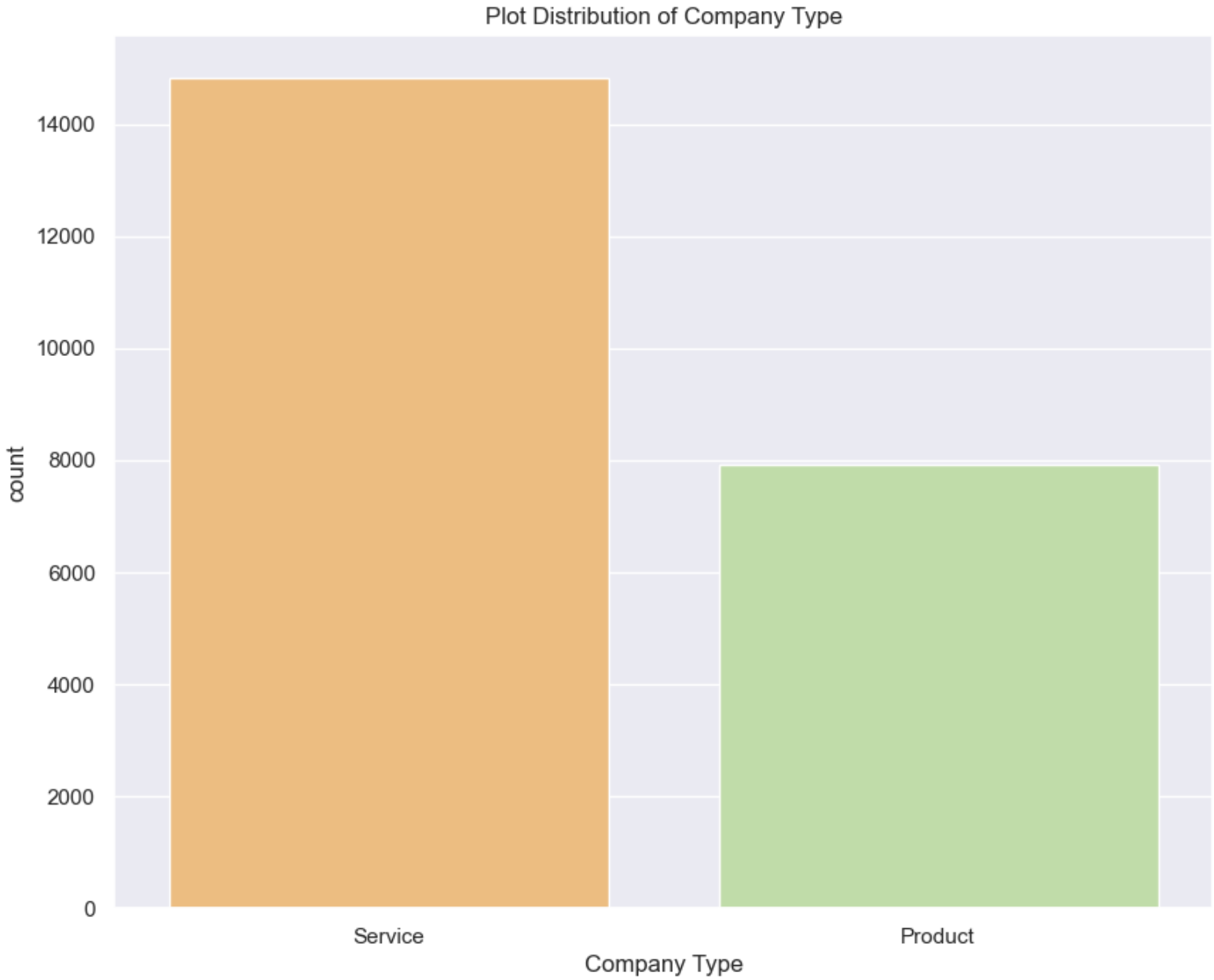


In [19]:
```
#count plot distribution of "Gender"
plt.figure(figsize=(10,8))
sns.countplot(x="Gender", data=burnoutDF, palette="magma")
plt.title("Plot Distribution of Gender")
plt.show()
```

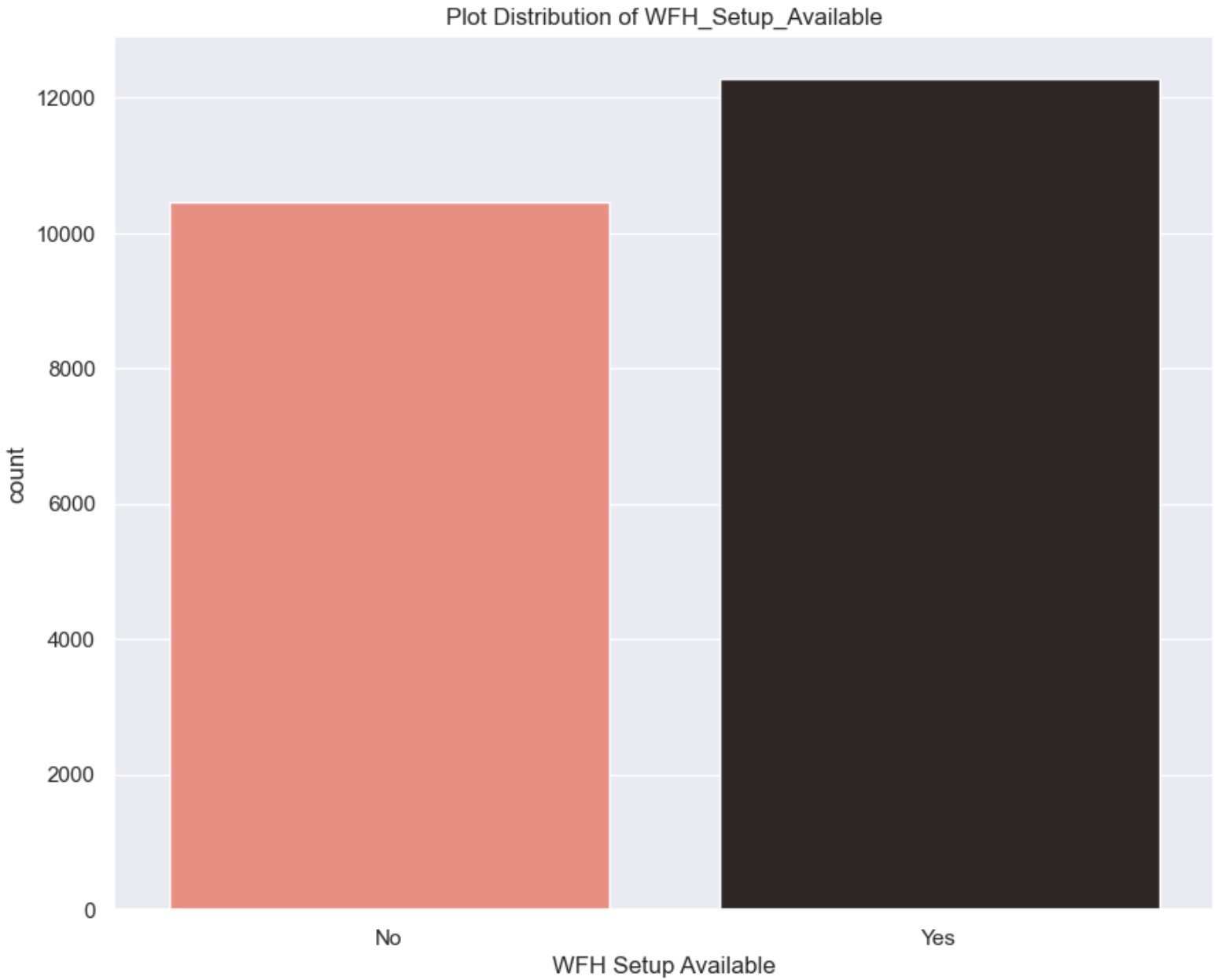


In [20]:
```
#Count plot distribution of "Company Type"
plt.figure(figsize=(10,8))
sns.countplot(x="Company Type", data=burnoutDF, palette="Spectral")
```

```
plt.title("Plot Distribution of Company Type")
plt.show()
```

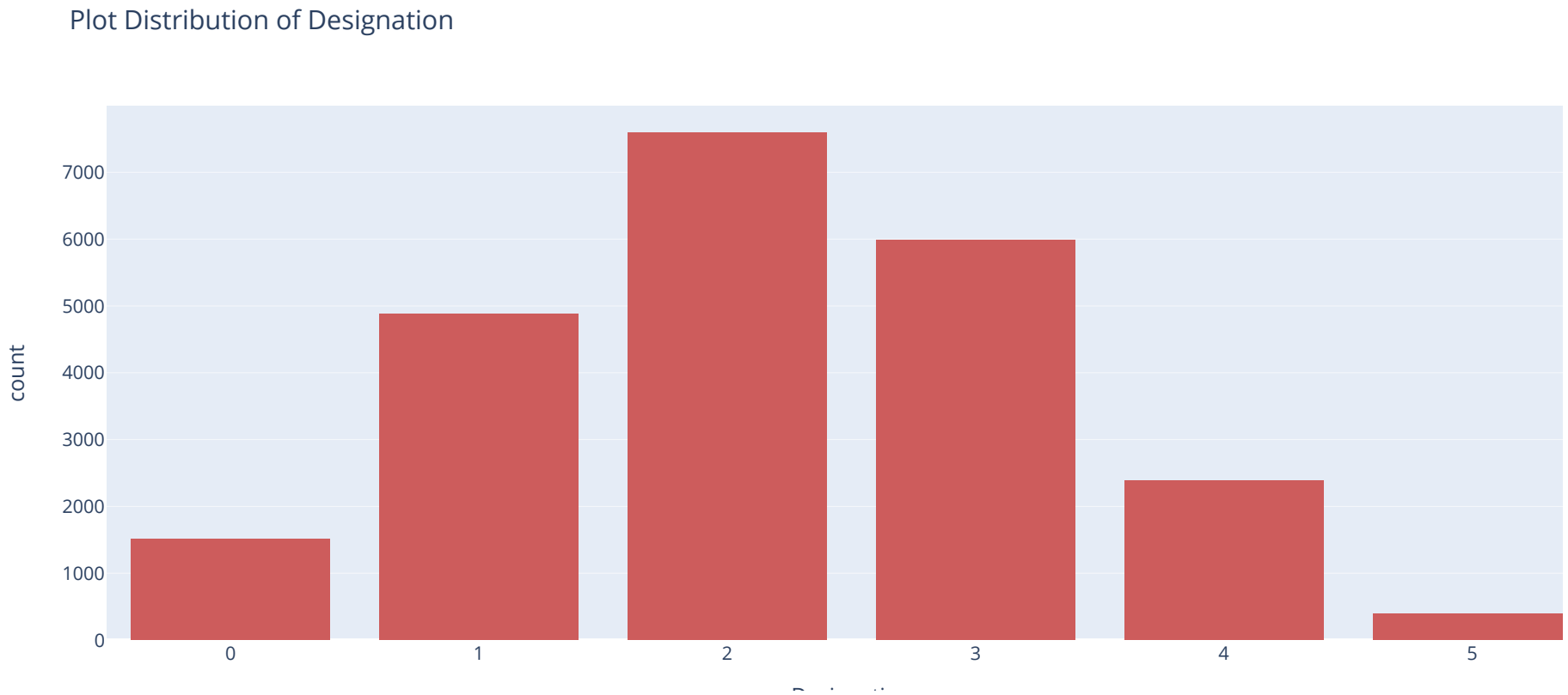


In [21]:
```
#Count plot distribution of "WFH Setup Available"
plt.figure(figsize=(10, 8))
sns.countplot(x="WFH Setup Available", data=burnoutDF, palette="dark:salmon_r")
plt.title("Plot Distribution of WFH_Setup_Available")
plt.show()
```

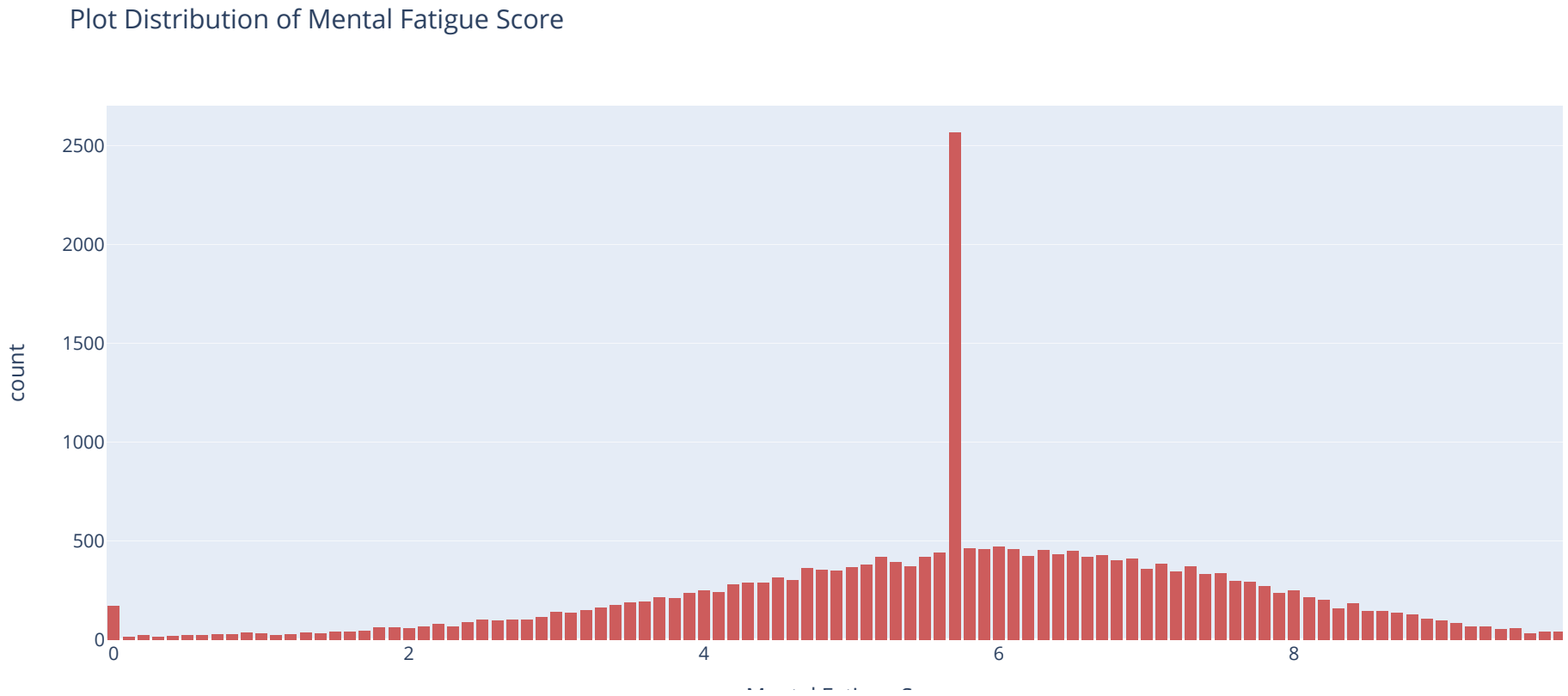


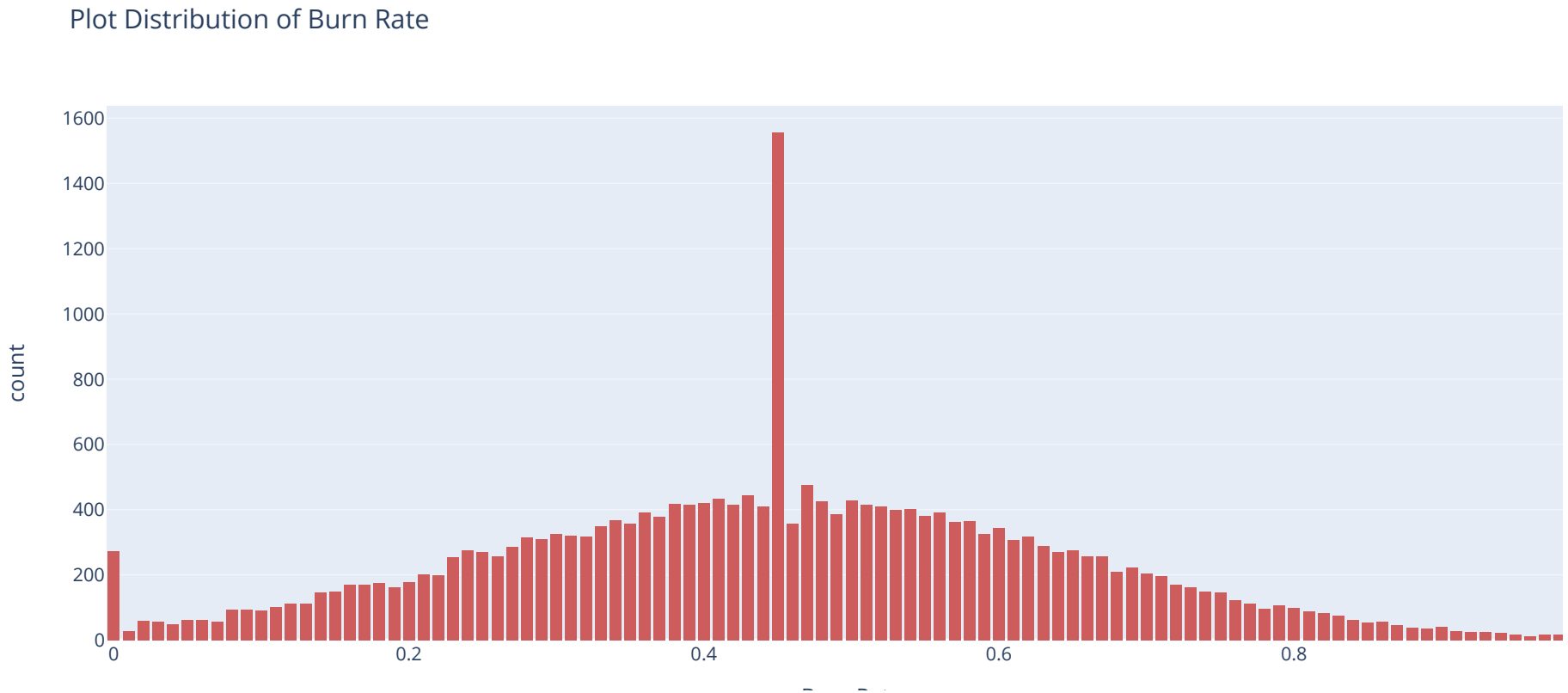
In [22]:
```
#Count-Plot Distribution of attributes with the help of Histogram
import plotly.express as px

burn_st = burnoutDF.loc[:, 'Date of Joining':'Burn Rate']
```

```
burn_st = burn_st.select_dtypes([int, float])

for col in burn_st.columns:
    fig = px.histogram(burn_st, x=col, title="Plot Distribution of " + col, color_discrete_sequence=['indianred'])
    fig.update_layout(bargap=0.2)
    fig.show()
```

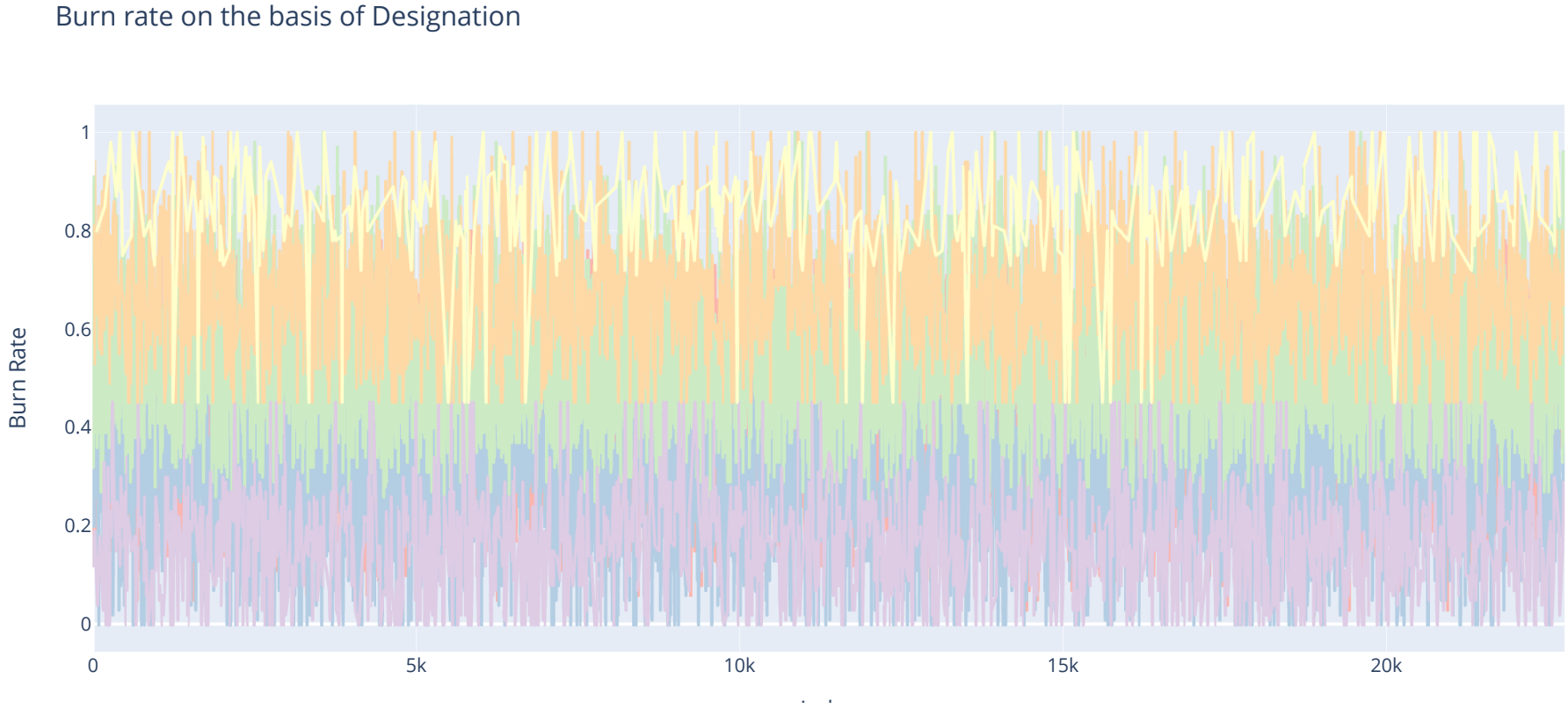



Plot Distribution of Resource Allocation
count
4000
3500
3000
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7
8
9

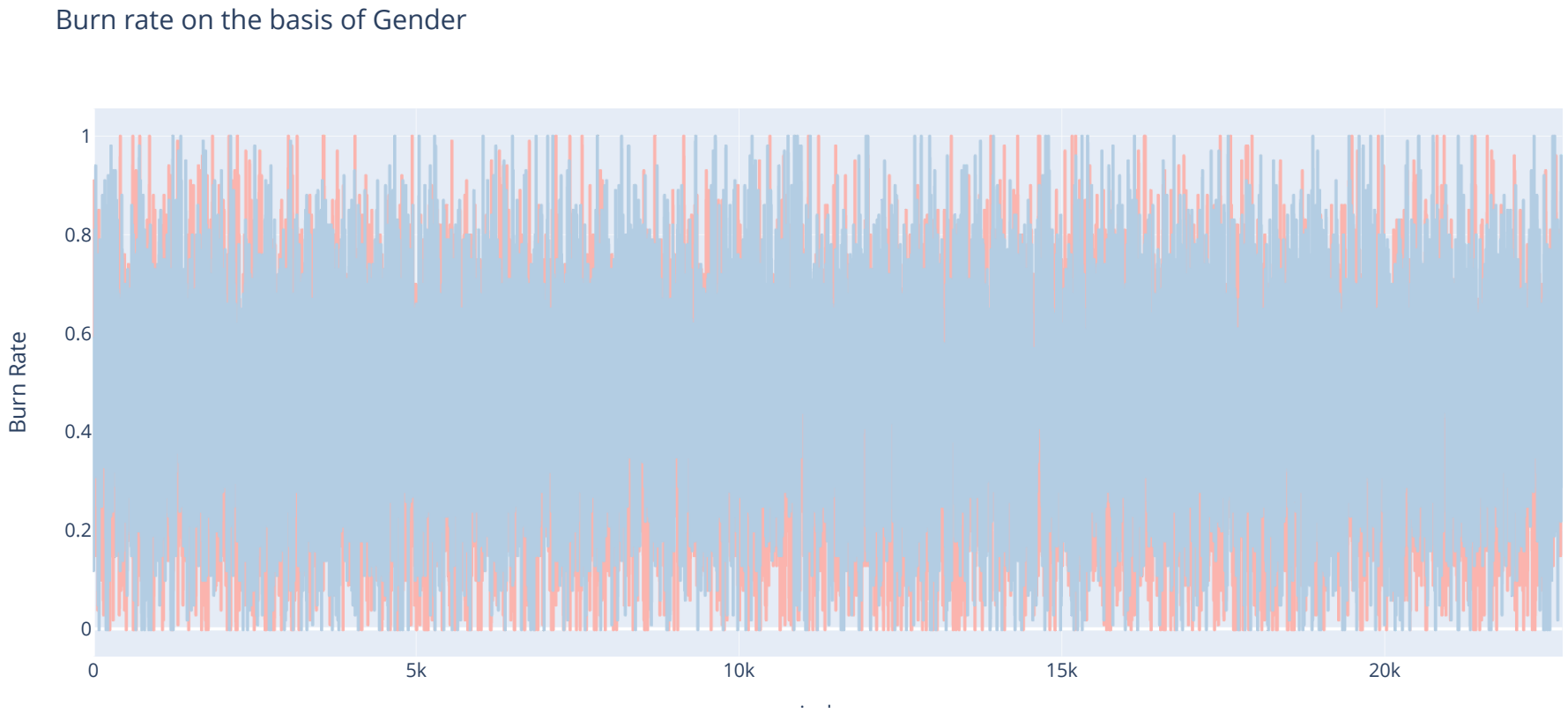

Plot Distribution of Mental Fatigue Score
count
2500
2000
1500
1000
500
0
0
2
4
6
8



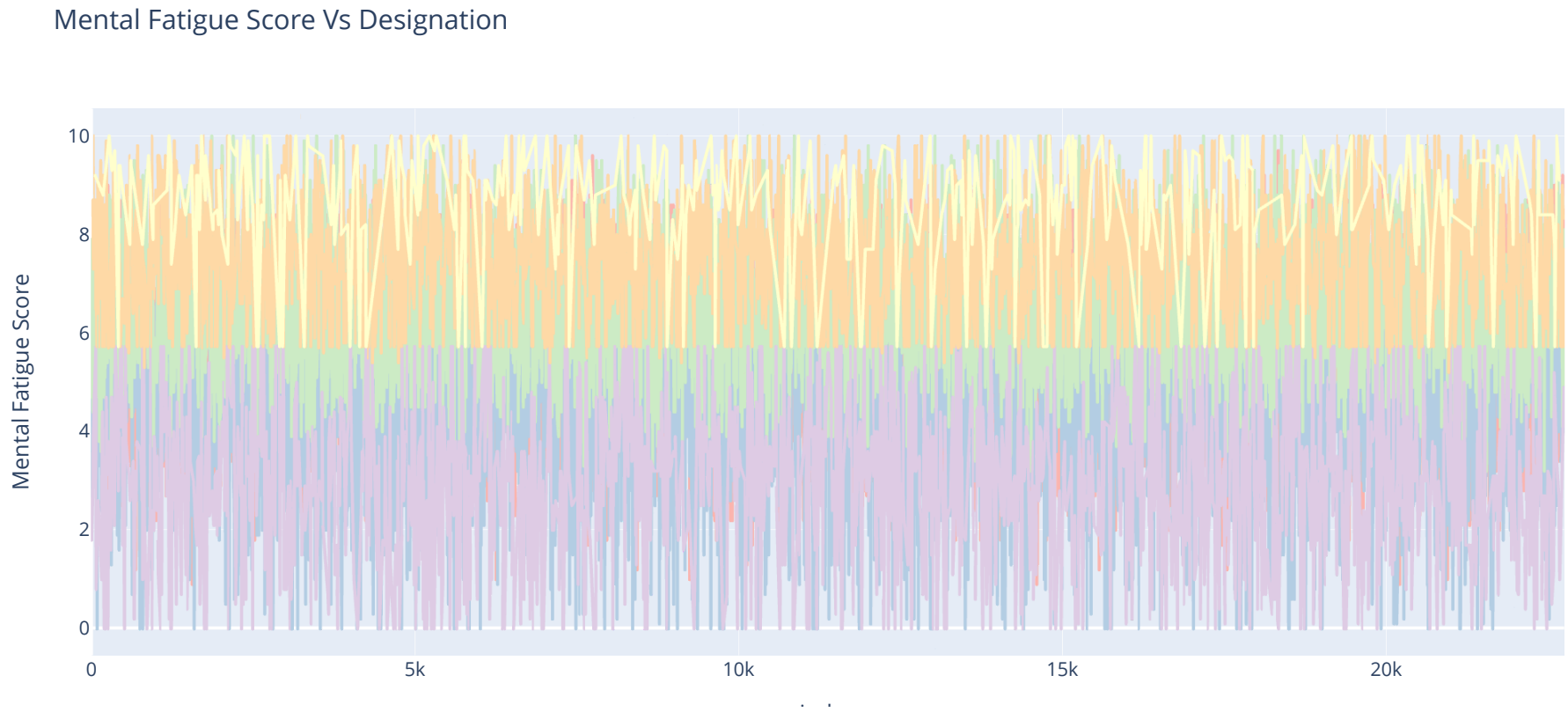
In [23]:
```
#Plot distribution of Burn Rate on the basis of Designation
fig = px.line(burnoutDF, y="Burn Rate", color="Designation", title="Burn rate on the basis of Designation",color_discrete_sequence =px.colors.qua
fig.update_layout(bargap=0.1)
fig.show()
```



In [24]:
```
#Plot distribution of Burn rate on the basis of gender
fig = px.line(burnoutDF, y="Burn Rate", color="Gender", title="Burn rate on the basis of Gender",color_discrete_sequence =px.colors.qualitative.
fig.update_layout(bargap=0.2)
fig.show()
```

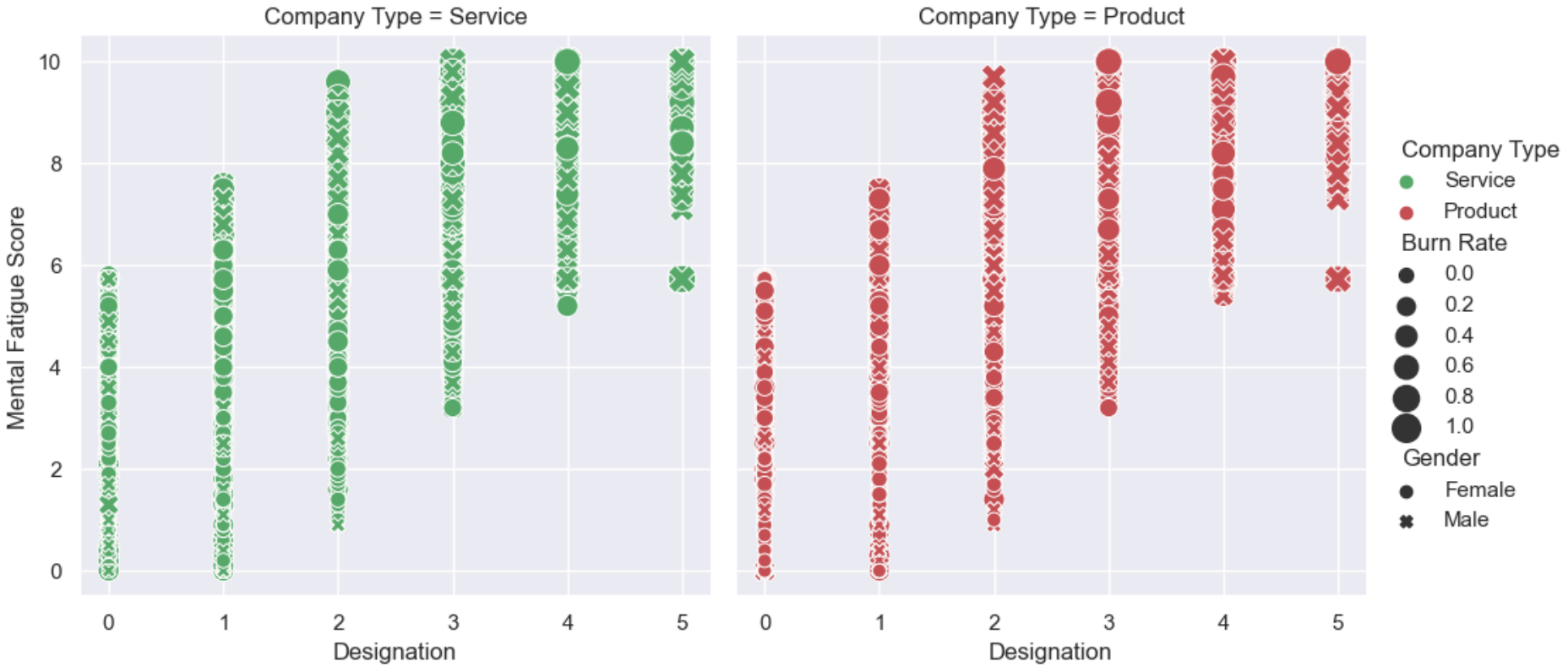


In [25]:
```
#Plot distribution of Mental Fatigue Score on the basis of Designation
fig = px.line(burnoutDF, y="Mental Fatigue Score", color="Designation", title="Mental Fatigue Score Vs Designation",color_discrete_sequence =px.c
fig.update_layout(bargap=0.2)
fig.show()
```



In [26]:
```
#Plot Distribution of "Designation Vs Mental Fatigue Score" as per company type, burn rate and gender
sns.relplot(
    data=burnoutDF, x="Designation", y="Mental Fatigue Score", col="Company Type",
    hue="Company Type", size="Burn Rate", style="Gender",
    palette=["g", "r"], sizes=(50, 200)
)
```

Out[26]:
```
<seaborn.axisgrid.FacetGrid at 0x2adb1388dc0>
```



Label Encoding

In [27]:
```
#Label Encoding and Assign in new variable
from sklearn import preprocessing
Lable_encode = preprocessing.LabelEncoder()
```

In [28]:
```
#Assign in new variable
from sklearn.preprocessing import LabelEncoder

Label_encode = LabelEncoder()

burnoutDF['GenderLabel'] = Label_encode.fit_transform(burnoutDF['Gender'].values)
burnoutDF['Company_TypeLabel'] = Label_encode.fit_transform(burnoutDF['Company Type'].values)
burnoutDF['WFH_Setup_AvailableLabel'] = Label_encode.fit_transform(burnoutDF['WFH Setup Available'].values)
```

In [29]:
```
#Check Assigned Values
gn = burnoutDF.groupby('Gender')
```

```
gn = gn['GenderLabel']
gn.first()
```

Out[29]:
```
Gender
Female    0
Male      1
Name: GenderLabel, dtype: int32
```

In [30]:
```
#Check Assigned Values
gn = burnoutDF.groupby('Company Type')
gn = gn['Company_TypeLabel']
gn.first()
```

Out[30]:
```
Company Type
Product    0
Service    1
Name: Company_TypeLabel, dtype: int32
```

In [31]:
```
#Check Assigned Values
wsa = burnoutDF.groupby('WFH_Setup_AvailableLabel')
wsa = wsa['WFH_Setup_AvailableLabel']
wsa.first()
```

Out[31]:
```
WFH_Setup_AvailableLabel
0    0
1    1
Name: WFH_Setup_AvailableLabel, dtype: int32
```

In [32]:
```
#Show last 10 rows
burnoutDF.tail(10)
```

Out[32]:

	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate	GenderLabel	Company_TypeLabel	WFH_Setup_AvailableLabel
22740	2008-09-05	Female	Product	No	3.0	6.0	7.300000	0.550000	0	0	0
22741	2008-01-07	Male	Product	No	2.0	5.0	6.000000	0.452005	1	0	0
22742	2008-07-28	Male	Product	No	3.0	5.0	8.100000	0.690000	1	0	0
22743	2008-12-15	Female	Product	Yes	1.0	3.0	6.000000	0.480000	0	0	1
22744	2008-05-27	Male	Product	No	3.0	7.0	6.200000	0.540000	1	0	0
22745	2008-12-30	Female	Service	No	1.0	3.0	5.728188	0.410000	0	1	0
22746	2008-01-19	Female	Product	Yes	3.0	6.0	6.700000	0.590000	0	0	1
22747	2008-11-05	Male	Service	Yes	3.0	7.0	5.728188	0.720000	1	1	1
22748	2008-01-10	Female	Service	No	2.0	5.0	5.900000	0.520000	0	1	0
22749	2008-01-06	Male	Product	No	3.0	6.0	7.800000	0.610000	1	0	0

Feature Selection

In [33]:
```
#Feature Selection
Columns=['Designation', 'Resource Allocation', 'Mental Fatigue Score',
        'GenderLabel', 'Company_TypeLabel', 'WFH_Setup_AvailableLabel']
x=burnoutDF[Columns]
y=burnoutDF['Burn Rate']
```

In [34]:
```
print(x)
```

```
       Designation  Resource Allocation  Mental Fatigue Score  GenderLabel  \
0              2.0             3.000000              3.800000            0
1              1.0             2.000000              5.000000            1
2              2.0             4.481398              5.800000            0
3              1.0             1.000000              2.600000            1
4              3.0             7.000000              6.900000            0
...            ...                  ...                   ...          ...
22745          1.0             3.000000              5.728188            0
22746          3.0             6.000000              6.700000            0
22747          3.0             7.000000              5.728188            1
22748          2.0             5.000000              5.900000            0
22749          3.0             6.000000              7.800000            1

       Company_TypeLabel  WFH_Setup_AvailableLabel
0                      1                         0
1                      1                         1
2                      0                         1
3                      1                         1
4                      1                         0
...                  ...                       ...
22745                  1                         0
22746                  0                         1
22747                  1                         1
22748                  1                         0
22749                  0                         0

[22750 rows x 6 columns]
```

In [35]:
```
print(y)
```

```
0        0.16
1        0.36
2        0.49
3        0.20
4        0.52
         ...
22745    0.41
22746    0.59
22747    0.72
22748    0.52
22749    0.61
Name: Burn Rate, Length: 22750, dtype: float64
```

Implementing PCA- Principal Component Analysis

In [36]:
```
#Principle Component Analysis
import numpy as np
```

```
from sklearn.decomposition import PCA

# Generating random data
np.random.seed(42)
X = np.random.rand(22750, 4)

pca = PCA(0.95)
X_pca = pca.fit_transform(X)

print("PCA shape of X is:", X_pca.shape, "and original shape is:", X.shape)
print("% of importance of selected features is:", pca.explained_variance_ratio_)
print("The number of features selected through PCA is:", pca.n_components_)
```

```
PCA shape of X is: (22750, 4) and original shape is: (22750, 4)
% of importance of selected features is: [0.2538984  0.25185043 0.24895039 0.24530078]
The number of features selected through PCA is: 4
```

Data Splitting

In [37]:
```
#Data Splitting in train and test
from sklearn.model_selection import train_test_split
X_train_pca, X_test, Y_train, Y_test = train_test_split(X_pca,y, test_size = 0.25, random_state=10)
```

In [38]:
```
#Print the shape of splitted data
print(X_train_pca.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

```
(17062, 4) (5688, 4) (17062,) (5688,)
```

Model Implementation

Random Forest Regressor

In [39]:
```
from sklearn.metrics import r2_score
```

In [40]:
```
#Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor()
rf_model.fit(X_train_pca, Y_train)

train_pred_rf = rf_model.predict(X_train_pca)
train_r2 = r2_score(Y_train, train_pred_rf)
```

```
test_pred_rf = rf_model.predict(X_test)
test_r2 = r2_score(Y_test, test_pred_rf)

#Accuracy Score
print("Accuracy score of train data: "+str(round(100*train_r2, 4))+" %")
print("Accuracy score of test data: "+str(round(100*test_r2, 4))+" %")
```

```
Accuracy score of train data: 85.1066 %
Accuracy score of test data: -5.2416 %
```

AdaBoost Regressor

In [41]:
```
#AdaBoost Regressor
from sklearn.ensemble import AdaBoostRegressor
abr_model = AdaBoostRegressor()
abr_model.fit(X_train_pca, Y_train)

train_pred_adaboost = abr_model.predict(X_train_pca)
train_r2 = r2_score(Y_train, train_pred_adaboost)
test_pred_adaboost = abr_model.predict(X_test)
test_r2 = r2_score(Y_test, test_pred_adaboost)

#Accuracy Score
print("Accuracy score of train data: "+str(round(100*train_r2, 4))+" %")
print("Accuracy score of test data: "+str(round(100*test_r2, 4))+" %")
```

```
Accuracy score of train data: 0.1034 %
Accuracy score of test data: -0.1033 %
```

In [ ]:
```
```