In [4]:
```
import pandas as pd
from warnings import filterwarnings
filterwarnings("ignore")
```

In [5]:
```
A=pd.read_csv("C:/Users/ASUS/Downloads/WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

In [6]:
```
A.head()
```

Out[6]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber
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

In [7]:
```
A.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
```

In [8]:
```
A.isna().sum()
```

Out[8]:
```
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
```

In [12]:
```
A.describe()
```

Out[12]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000

8 rows × 26 columns

In [15]:
```
Y=A[["MonthlyRate"]]
X=A.drop(["MonthlyRate"],axis=1)
```

In [16]:
```
Y
```

Out[16]:

	MonthlyRate
0	19479
1	24907
2	2396
3	23159
4	16632
...	...
1465	12290
1466	21457
1467	5174
1468	13243
1469	10228

1470 rows × 1 columns

In [17]:
```
X
```

Out[17]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNum
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2

1470 rows × 34 columns

Exploratory Data Analysis

In [20]:
```
cat=[]
con=[]
for i in A.columns:
    if(A[i].dtypes=="object"):
        cat.append(i)
    else:
        con.append(i)
```

In [21]:
```
cat
```

Out[21]:
```
['Attrition',
 'BusinessTravel',
 'Department',
 'EducationField',
 'Gender',
 'JobRole',
 'MaritalStatus',
 'Over18',
 'OverTime']
```

In [22]:
```
con
```

Out[22]:
```
['Age',
 'DailyRate',
 'DistanceFromHome',
 'Education',
 'EmployeeCount',
 'EmployeeNumber',
 'EnvironmentSatisfaction',
 'HourlyRate',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
 'MonthlyIncome',
 'MonthlyRate',
 'NumCompaniesWorked',
 'PercentSalaryHike',
 'PerformanceRating',
 'RelationshipSatisfaction',
 'StandardHours',
 'StockOptionLevel',
 'TotalWorkingYears',
 'TrainingTimesLastYear',
 'WorkLifeBalance',
 'YearsAtCompany',
 'YearsInCurrentRole',
 'YearsSinceLastPromotion',
 'YearsWithCurrManager']
```

In [23]:
```
def ANOVA(A,cat,con):
    from pandas import DataFrame
    from statsmodels.formula.api import ols
    rel = con + " ~ " + cat
    model = ols(rel,A).fit()
    from statsmodels.stats.anova import anova_lm
```

```
    anova_results = anova_lm(model)
    Q = DataFrame(anova_results)
    a = Q['PR(>F)'][cat]
    return round(a,3)
```

In [24]:
```
imp_con_cols = []
for i in cat:
    print("--------MonthlyRate vs",i,"----------")
    x = ANOVA(A,i,"MonthlyRate")
    print(x)
    if(x < 0.05):
        imp_con_cols.append(i)
```

```
--------MonthlyRate vs Attrition ----------
0.561
--------MonthlyRate vs BusinessTravel ----------
0.838
--------MonthlyRate vs Department ----------
0.57
--------MonthlyRate vs EducationField ----------
0.753
--------MonthlyRate vs Gender ----------
0.112
--------MonthlyRate vs JobRole ----------
0.779
--------MonthlyRate vs MaritalStatus ----------
0.31
--------MonthlyRate vs Over18 ----------
nan
--------MonthlyRate vs OverTime ----------
0.412
```

In [25]:
```
imp_con_cols.append("Gender")
```

In [27]:
```
from PM8 import preprocessing
Anew=preprocessing(A)
```

In [28]:
```
Anew.corr()
```

Out[28]:

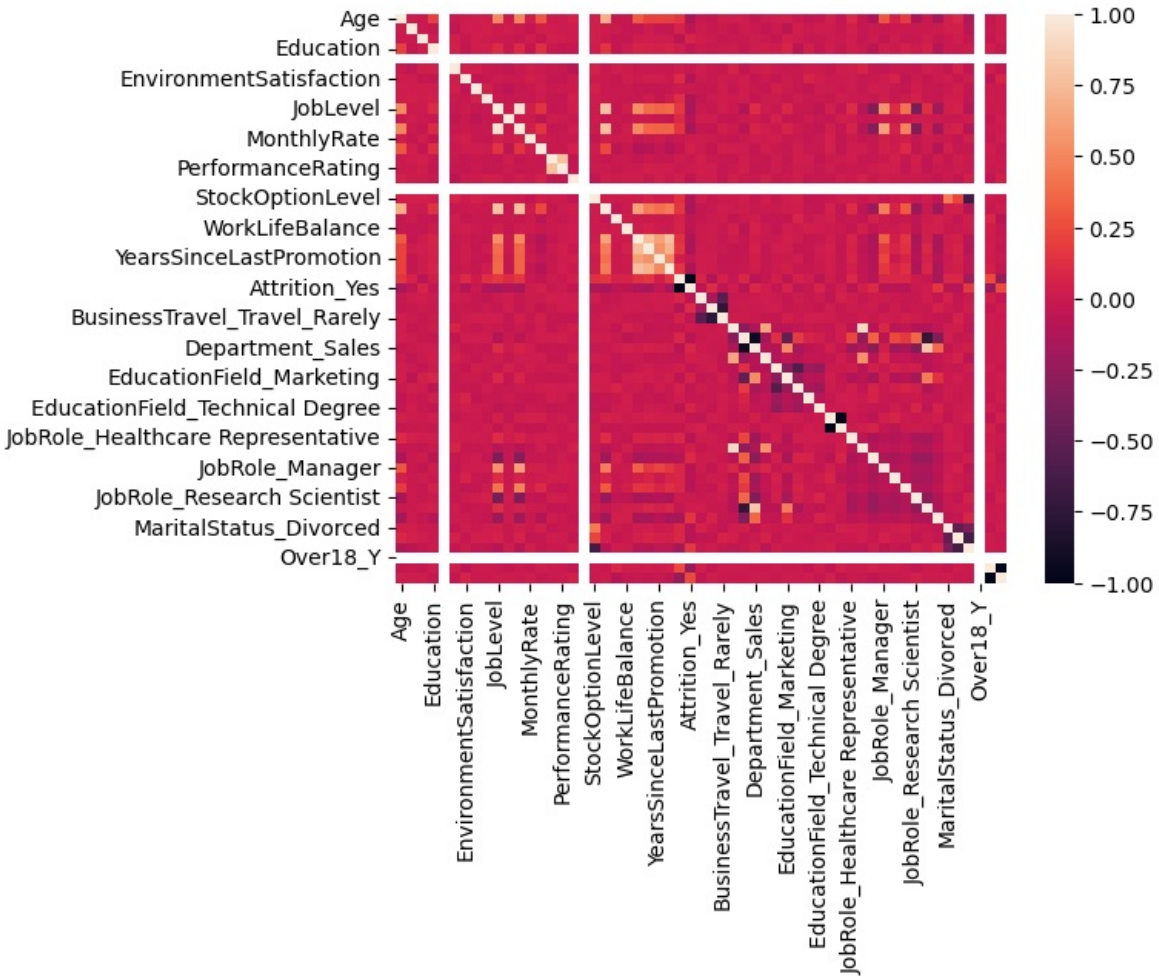
	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfa
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.010145	0.01
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.050990	0.01
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	0.032916	-0.01
Education	0.208034	-0.016806	0.021042	1.000000	NaN	0.042070	-0.02
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN	
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	1.000000	0.01
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	0.017621	1.00
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	0.035179	-0.04
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	-0.006888	-0.00
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	-0.018519	0.00
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.046247	-0.00
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	-0.014829	-0.00
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	0.012648	0.03
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	-0.001251	0.01
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	-0.012944	-0.03
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	-0.020359	-0.02
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	-0.069861	0.00
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	0.062227	0.00
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	-0.014365	-0.00
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	0.023603	-0.01
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	0.010309	0.02
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	-0.011240	0.00
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	-0.008416	0.01
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	-0.009019	0.01
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	-0.009197	-0.00
Attrition_No	0.159205	0.056652	-0.077924	0.031373	NaN	0.010577	0.10
Attrition_Yes	-0.159205	-0.056652	0.077924	-0.031373	NaN	-0.010577	-0.10
BusinessTravel_Non-Travel	-0.011215	0.012096	0.023605	0.004524	NaN	0.022272	0.00

BusinessTravel_Travel_Frequently	-0.024743	-0.011776	0.005081	-0.008292	NaN	-0.007980	-0.01
BusinessTravel_Travel_Rarely	0.028791	0.002078	-0.020116	0.004126	NaN	-0.007976	0.00
Department_Human Resources	0.020523	-0.026726	-0.012901	0.011435	NaN	0.063431	-0.00
Department_Research & Development	0.017883	0.014871	-0.008117	-0.018604	NaN	-0.041923	0.02
Department_Sales	-0.027549	-0.003616	0.014085	0.014215	NaN	0.015441	-0.02
EducationField_Human Resources	0.001696	-0.043144	-0.002624	0.026479	NaN	0.035345	-0.00
EducationField_Life Sciences	0.016824	0.004028	-0.024499	0.013184	NaN	-0.000609	-0.02
EducationField_Marketing	0.038162	-0.064449	0.039294	0.072405	NaN	-0.014487	0.00
EducationField_Medical	-0.006354	0.034202	0.013486	-0.072335	NaN	-0.008689	-0.02
EducationField_Other	-0.041466	-0.003893	-0.007969	0.038043	NaN	0.010432	0.06
EducationField_Technical Degree	-0.027604	0.030869	-0.014802	-0.026742	NaN	0.005938	0.02
Gender_Female	0.036311	0.011716	0.001851	0.016547	NaN	-0.022556	-0.00
Gender_Male	-0.036311	-0.011716	-0.001851	-0.016547	NaN	0.022556	0.00
JobRole_Healthcare Representative	0.098825	0.040141	0.022916	0.024270	NaN	0.025945	0.01
JobRole_Human Resources	-0.029856	-0.021156	-0.024089	-0.005295	NaN	0.067287	-0.02
JobRole_Laboratory Technician	-0.143176	-0.006728	0.012369	-0.063566	NaN	-0.019722	-0.00
JobRole_Manager	0.294248	-0.013224	-0.039190	0.028453	NaN	-0.035058	0.01
JobRole_Manufacturing Director	0.049726	-0.005302	0.011848	-0.005290	NaN	-0.014350	0.05
JobRole_Research Director	0.185891	-0.000021	-0.022351	0.049694	NaN	-0.013983	-0.04
JobRole_Research Scientist	-0.146518	-0.002624	-0.010986	0.000709	NaN	-0.017686	0.00
JobRole_Sales Executive	-0.002001	-0.000513	0.030761	0.053398	NaN	0.023263	-0.02
JobRole_Sales Representative	-0.175785	0.005375	-0.015994	-0.091465	NaN	0.006255	0.00
MaritalStatus_Divorced	0.033120	0.037080	-0.005440	-0.002439	NaN	-0.025149	0.01
MaritalStatus_Married	0.083919	0.040035	0.030232	-0.001865	NaN	0.053933	-0.02
MaritalStatus_Single	-0.119185	-0.075835	-0.027445	0.004168	NaN	-0.035189	0.00
Over18_Y	NaN	NaN	NaN	NaN	NaN	NaN	
OverTime_No	-0.028062	-0.009135	-0.025514	0.020322	NaN	0.024037	-0.07
OverTime_Yes	0.028062	0.009135	0.025514	-0.020322	NaN	-0.024037	0.07

57 rows × 57 columns

In [30]:
```
import seaborn as sb
sb.heatmap(Anew.corr())
```

Out[30]:
```
<Axes: >
```



In [31]:
```
Anew.corr()["MonthlyRate"].sort_values()
```

Out[31]:
```
Gender_Male                          -0.041482
YearsWithCurrManager                 -0.036746
EducationField_Other                 -0.035606
MaritalStatus_Married                -0.034689
StockOptionLevel                     -0.034323
DailyRate                            -0.032182
JobRole_Human Resources              -0.027470
JobRole_Research Scientist           -0.027008
Education                            -0.026084
Department_Human Resources           -0.024390
YearsAtCompany                       -0.023655
OverTime_No                          -0.021431
JobInvolvement                       -0.016322
JobRole_Laboratory Technician        -0.016056
HourlyRate                           -0.015297
Attrition_No                         -0.015170
YearsInCurrentRole                   -0.012815
EducationField_Marketing             -0.011559
BusinessTravel_Travel_Rarely         -0.010484
PerformanceRating                    -0.009811
PercentSalaryHike                    -0.006429
Department_Research & Development    -0.005453
EducationField_Technical Degree      -0.004535
RelationshipSatisfaction             -0.004085
EducationField_Medical               -0.001723
JobRole_Sales Representative         -0.001200
MaritalStatus_Divorced               -0.000227
BusinessTravel_Travel_Frequently      0.000344
JobSatisfaction                       0.000644
TrainingTimesLastYear                 0.001467
YearsSinceLastPromotion               0.001567
JobRole_Healthcare Representative     0.003829
JobRole_Manufacturing Director        0.007711
WorkLifeBalance                       0.007963
EducationField_Human Resources        0.009567
JobRole_Sales Executive               0.011854
EmployeeNumber                        0.012648
Attrition_Yes                         0.015170
BusinessTravel_Non-Travel             0.015279
Department_Sales                      0.016388
NumCompaniesWorked                    0.017521
OverTime_Yes                          0.021431
EducationField_Life Sciences          0.025545
JobRole_Research Director             0.025875
TotalWorkingYears                     0.026442
DistanceFromHome                      0.027473
Age                                   0.028051
JobRole_Manager                       0.031717
MonthlyIncome                         0.034814
MaritalStatus_Single                  0.037260
EnvironmentSatisfaction               0.037600
JobLevel                              0.039563
Gender_Female                         0.041482
MonthlyRate                           1.000000
EmployeeCount                              NaN
StandardHours                              NaN
Over18_Y                                   NaN
Name: MonthlyRate, dtype: float64
```

In [32]:
```
X=A[["JobLevel"]]
```

Split the data in training and testing set

In [37]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(X,Y,test_size=0.2,random_state=21)
```

Create regression models:

1. SLR
2. MLR
3. DTR
4. RFR | ABR

In [38]:
```
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.metrics import mean_absolute_error,explained_variance_score
```

In [39]:
```
def model(mobj):
    model = mobj.fit(xtrain,ytrain)
    pred_tr = model.predict(xtrain)
    tr_acc = mean_absolute_error(ytrain,pred_tr)
    pred_ts = model.predict(xtest)
    ts_acc = mean_absolute_error(ytest,pred_ts)
```

```
    exp_var = explained_variance_score(ytest,pred_ts)
    return round(tr_acc,3),round(ts_acc,3),exp_var,mobj
```

In [40]:
```
lm = LinearRegression()
model(lm)
```

Out[40]:
```
(6164.147, 6265.515, 0.004169958769105198, LinearRegression())
```

In [41]:
```
from PM8 import preprocessing
X = A.drop(labels=["MonthlyRate"],axis=1)
Xnew = preprocessing(X)

from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(Xnew,Y,test_size=0.2,random_state=21)
```

In [42]:
```
Q = []
for i in range(2,20):
    dtr = DecisionTreeRegressor(random_state=21,max_depth=i)
    Q.append(model(dtr))
```

In [43]:
```
Q
```

Out[43]:
```
[(6074.345,
  6332.767,
  -0.04170664128709212,
  DecisionTreeRegressor(max_depth=2, random_state=21)),
 (6000.14,
  6345.39,
  -0.06116753684483922,
  DecisionTreeRegressor(max_depth=3, random_state=21)),
 (5861.491,
  6313.796,
  -0.04838971242604395,
  DecisionTreeRegressor(max_depth=4, random_state=21)),
 (5688.867,
  6486.016,
  -0.14061616285458056,
  DecisionTreeRegressor(max_depth=5, random_state=21)),
 (5407.625,
  6630.185,
  -0.2299109615808883,
  DecisionTreeRegressor(max_depth=6, random_state=21)),
 (5008.309,
  6951.234,
  -0.37175540275813,
  DecisionTreeRegressor(max_depth=7, random_state=21)),
 (4497.233,
  7221.869,
  -0.4816401663643539,
  DecisionTreeRegressor(max_depth=8, random_state=21)),
 (4039.262,
  7371.881,
  -0.5471797019623046,
  DecisionTreeRegressor(max_depth=9, random_state=21)),
 (3437.394,
  7517.979,
  -0.6647589643117675,
  DecisionTreeRegressor(max_depth=10, random_state=21)),
 (2848.33,
  7749.763,
  -0.7885594561498659,
  DecisionTreeRegressor(max_depth=11, random_state=21)),
 (2332.73,
  7740.942,
  -0.8179995488627705,
  DecisionTreeRegressor(max_depth=12, random_state=21)),
 (1816.009,
  8265.631,
  -1.0061188291071117,
  DecisionTreeRegressor(max_depth=13, random_state=21)),
 (1357.462,
  8338.422,
  -1.066983967490426,
  DecisionTreeRegressor(max_depth=14, random_state=21)),
 (1007.61,
  8204.346,
  -1.0284664718562766,
  DecisionTreeRegressor(max_depth=15, random_state=21)),
 (703.974,
  8366.163,
  -1.0867643130987519,
  DecisionTreeRegressor(max_depth=16, random_state=21)),
 (481.782,
  8319.248,
  -1.093226906005413,
  DecisionTreeRegressor(max_depth=17, random_state=21)),
 (295.735,
  8590.245,
  -1.152023628017227,
  DecisionTreeRegressor(max_depth=18, random_state=21)),
 (182.673,
  8319.57,
  -1.0965203092219529,
  DecisionTreeRegressor(max_depth=19, random_state=21))]
```

In [51]:
```
Q = []
for i in range(2,20):
    rfr = RandomForestRegressor(random_state=21,n_estimators=i)
    Q.append(model(rfr))
```

In [52]:
```
Q
```

Out[52]:
```
[(2894.49,
  7542.997,
  -0.5588149695369911,
  RandomForestRegressor(n_estimators=2, random_state=21)),
 (2820.181,
  6885.969,
  -0.28294089547226275,
  RandomForestRegressor(n_estimators=3, random_state=21)),
 (2724.226,
  6747.918,
  -0.2609229818943155,
  RandomForestRegressor(n_estimators=4, random_state=21)),
 (2653.244,
  6822.812,
  -0.2444534419179456,
  RandomForestRegressor(n_estimators=5, random_state=21)),
 (2659.111,
  6602.476,
  -0.20489978061707426,
  RandomForestRegressor(n_estimators=6, random_state=21)),
 (2610.651,
  6437.011,
  -0.12896581405391183,
  RandomForestRegressor(n_estimators=7, random_state=21)),
 (2566.612,
  6440.987,
  -0.12446642754624349,
  RandomForestRegressor(n_estimators=8, random_state=21)),
 (2525.04,
  6373.667,
  -0.09228826227679576,
  RandomForestRegressor(n_estimators=9, random_state=21)),
 (2511.313,
  6408.03,
  -0.09054138037616277,
  RandomForestRegressor(n_estimators=10, random_state=21)),
 (2485.92,
  6365.343,
  -0.07183002770381908,
  RandomForestRegressor(n_estimators=11, random_state=21)),
 (2490.037,
  6373.082,
  -0.07538376758478615,
  RandomForestRegressor(n_estimators=12, random_state=21)),
 (2477.057,
  6357.532,
  -0.06543521775252015,
  RandomForestRegressor(n_estimators=13, random_state=21)),
 (2448.62,
  6337.16,
  -0.05372552349449755,
  RandomForestRegressor(n_estimators=14, random_state=21)),
 (2442.794,
  6362.48,
  -0.0506703322694364,
  RandomForestRegressor(n_estimators=15, random_state=21)),
 (2437.101,
  6380.812,
  -0.0537991431447864,
  RandomForestRegressor(n_estimators=16, random_state=21)),
 (2427.986,
  6394.657,
  -0.05503663600102371,
  RandomForestRegressor(n_estimators=17, random_state=21)),
 (2421.508,
  6396.134,
  -0.05602250181141866,
  RandomForestRegressor(n_estimators=18, random_state=21)),
 (2413.536,
  6359.905,
  -0.04833805717227313,
  RandomForestRegressor(n_estimators=19, random_state=21))]
```

In [53]:
```
Q = []
for i in range(20,40):
    abr = AdaBoostRegressor(DecisionTreeRegressor(max_depth=3, random_state=21),n_estimators=i)
    Q.append(model(abr))
```

In [54]:
```
Q
```

Out[54]:
```
[(6035.407,
  6266.269,
  6.87926868913058e-05,
  AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                    n_estimators=20)),
 (6039.574,
  6278.251,
  -0.009534962737091313,
```

```
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=21)),
(6071.283,
 6239.264,
 0.006266835114127622,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=22)),
(5944.774,
 6244.689,
 -0.0013698917446380854,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=23)),
(6078.267,
 6237.018,
 0.005304377607059041,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=24)),
(6050.259,
 6271.97,
 0.0008426502345316722,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=25)),
(6035.32,
 6308.904,
 -0.006150452497419634,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=26)),
(6033.12,
 6265.646,
 0.006645640683752285,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=27)),
(6077.372,
 6274.081,
 0.0015392579599716738,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=28)),
(6042.585,
 6287.108,
 -0.004467559028518586,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=29)),
(6040.475,
 6302.352,
 -0.00017294581171811707,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=30)),
(6041.496,
 6300.948,
 -0.016903679758276002,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=31)),
(6021.334,
 6287.519,
 -0.011889674640965131,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=32)),
(6083.17,
 6275.301,
 -0.006458336484943761,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=33)),
(6051.161,
 6269.861,
 -0.0016997576251194246,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=34)),
(6047.599,
 6233.327,
 0.014656401870239866,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=35)),
(6057.915,
 6264.337,
 0.005025555850770802,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=36)),
(6066.114,
 6287.018,
 -0.003121125045016049,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=37)),
(6091.36,
 6291.215,
 -0.011383680177151119,
 AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                   n_estimators=38)),
(6009.154,
 6222.081,
```

```
0.003630254785532583,
AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=3, random_state=21),
                  n_estimators=39))]
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js