In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
df=pd.read_csv("https://raw.githubusercontent.com/dsrscientist/dataset4/main/medical_cost_insurance.csv")
```

In [4]:

```
df
```

Out[4]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [5]:

```
df.head()
```

Out[5]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [6]:

```
df.tail()
```

Out[6]:

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

In [7]:

```
df.shape
```

Out[7]:

```
(1338, 7)
```

In [8]:

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [10]:

```
df.describe()
```

Out[10]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [11]:

```
#checking unique value for Categorical data
df["sex"].unique()
```

Out[11]:

```
array(['female', 'male'], dtype=object)
```

In [12]:
```
df["sex"]=df["sex"].map({"female":0,"male":1})
```

In [13]:
```
df.head()
```

Out[13]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	yes	southwest	16884.92400
1	18	1	33.770	1	no	southeast	1725.55230
2	28	1	33.000	3	no	southeast	4449.46200
3	33	1	22.705	0	no	northwest	21984.47061
4	32	1	28.880	0	no	northwest	3866.85520

In [14]:
```
df["smoker"]=df["smoker"].map({"yes":1,"no":0})
```

In [15]:
```
df.head()
```

Out[15]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	southwest	16884.92400
1	18	1	33.770	1	0	southeast	1725.55230
2	28	1	33.000	3	0	southeast	4449.46200
3	33	1	22.705	0	0	northwest	21984.47061
4	32	1	28.880	0	0	northwest	3866.85520

In [16]:
```
df["region"].unique()
```

Out[16]:
```
array(['southwest', 'southeast', 'northwest', 'northeast'], dtype=object)
```

In [17]:
```
df["region"]=df["region"].map({"southwest":1,"southeast":2,"northwest":3,"northeast":4})
```

In [18]:
```
df.head()
```

Out[18]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	1	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	3	21984.47061
4	32	1	28.880	0	0	3	3866.85520

In [19]:
```
df.columns
```

Out[19]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'],
dtype='object')
```

In [21]:
```
X=df.drop(["charges"],axis=1)
```

In [22]:
```
y=df["charges"]
```

In [23]:

```
y
```

Out[23]:

```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
           ...
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

In [24]:

```
#train test split
from sklearn.model_selection import train_test_split
```

In [29]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_sta
te=42)
```

In [ ]:

In [27]:

```
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

In [32]:

```
lr=LinearRegression()
lr.fit(X_train,y_train)
svm=SVR()
svm.fit(X_train,y_train)
rf=RandomForestRegressor()
rf.fit(X_train,y_train)
gr=GradientBoostingRegressor()
gr.fit(X_train,y_train)
```

Out[32]:

```
GradientBoostingRegressor()
```

In [33]:

```
#prediction
y_pred1=lr.predict(X_test)
y_pred2=svm.predict(X_test)
y_pred3=rf.predict(X_test)
y_pred4=gr.predict(X_test)
```

In [36]:

```
df1=pd.DataFrame({"Actual":y_test,"Lr":y_pred1,"svm":y_pred2,'rf':y_pred3,"
gr":y_pred4})
```

In [37]:

```
df1
```

Out[37]:

	Actual	Lr	svm	rf	gr
764	9095.06825	8924.407244	9548.261584	11439.923610	11001.128629
887	5272.17580	7116.295018	9492.515425	5414.810675	5840.174656
890	29330.98315	36909.013521	9648.758701	28347.848156	28001.980112
1293	9301.89355	9507.874691	9555.044136	10330.994254	9745.291602
259	33750.29180	27013.350008	9420.421978	34560.282466	33639.100981
...	...	...	...	...	...
109	47055.53210	39116.968669	9648.902852	46843.755068	45431.423211
575	12222.89830	11814.555568	9625.431547	12783.760343	12465.025294
535	6067.12675	7638.107736	9504.168517	6337.923312	6974.336525
543	63770.42801	40959.081722	9605.004594	46785.988248	47862.047791
846	9872.70100	12258.228529	9590.987268	9742.272343	10289.655388

268 rows × 5 columns

In [38]:

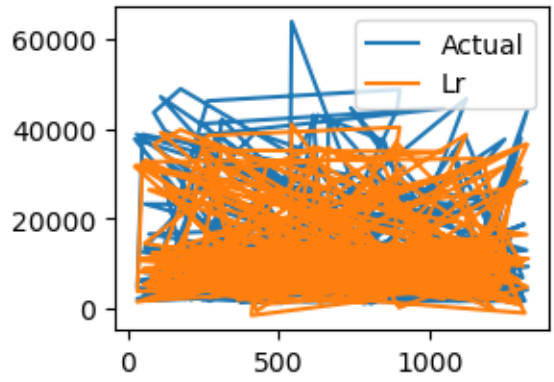
```
#Comparing actual vs prediction
import matplotlib.pyplot as plt
```

In [40]:

```
plt.subplot(221)
plt.plot(df1["Actual"],label="Actual")
plt.plot(df1["Lr"],label="Lr")
plt.legend()
```

Out[40]:

```
<matplotlib.legend.Legend at 0x222db477250>
```

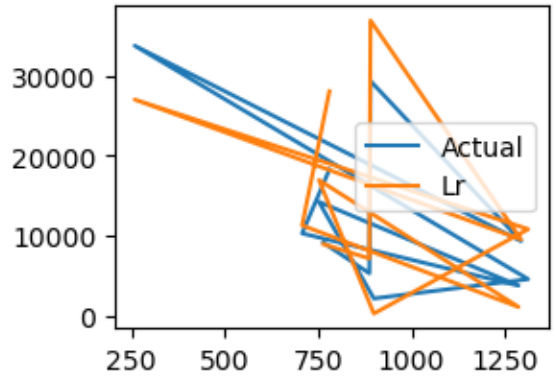


In [41]:

```
plt.subplot(221)
plt.plot(df1["Actual"].iloc[0:11],label="Actual")
plt.plot(df1["Lr"].iloc[0:11],label="Lr")
plt.legend()
```

Out[41]:

```
<matplotlib.legend.Legend at 0x222db3e3be0>
```

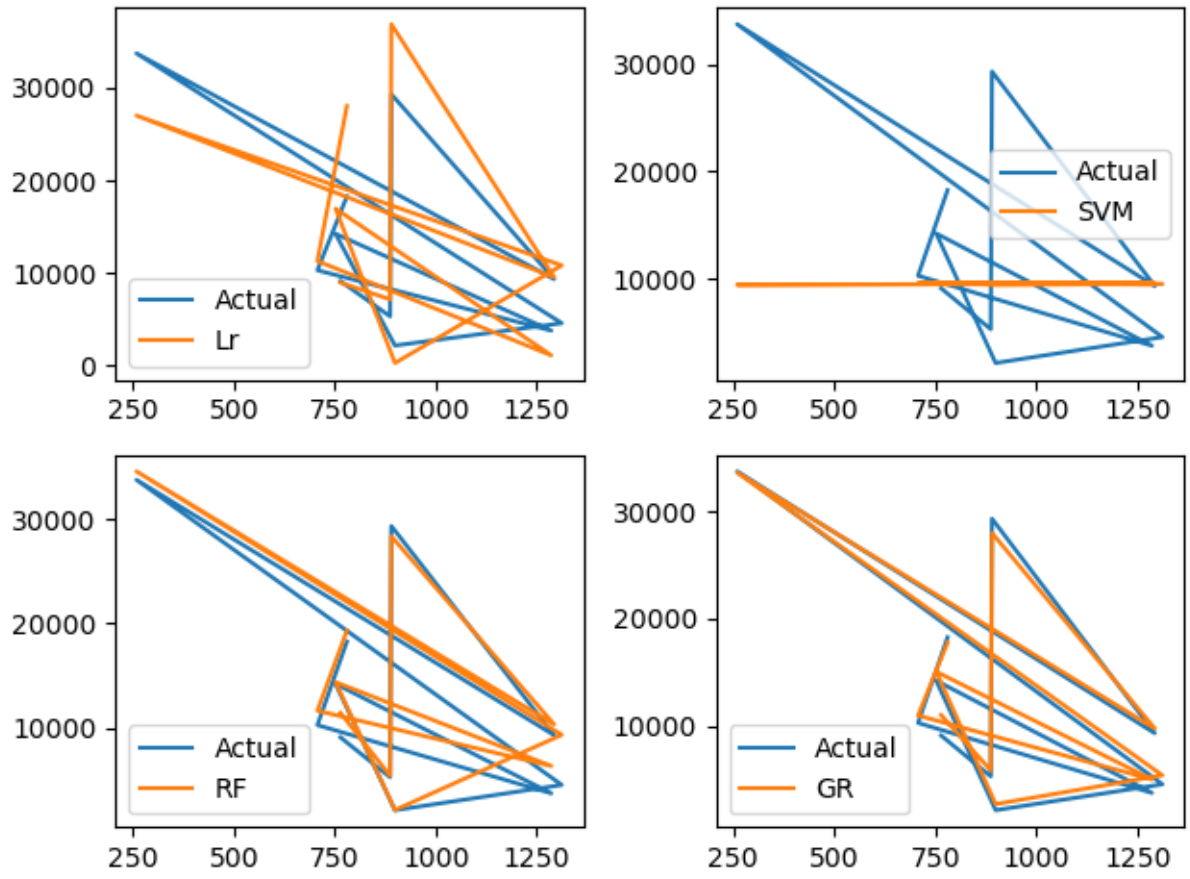


In [44]:

```
plt.subplot(221)
plt.plot(df1["Actual"].iloc[0:11],label="Actual")
plt.plot(df1["Lr"].iloc[0:11],label="Lr")
plt.legend()

plt.subplot(222)
plt.plot(df1["Actual"].iloc[0:11],label="Actual")
plt.plot(df1["svm"].iloc[0:11],label="SVM")
plt.legend()

plt.subplot(223)
plt.plot(df1["Actual"].iloc[0:11],label="Actual")
plt.plot(df1["rf"].iloc[0:11],label="RF")
plt.legend()

plt.subplot(224)
plt.plot(df1["Actual"].iloc[0:11],label="Actual")
plt.plot(df1["gr"].iloc[0:11],label="GR")

plt.tight_layout()

plt.legend()
```

Out[44]:

```
<matplotlib.legend.Legend at 0x222db9f79d0>
```



In [45]:

```
#Evaluating alogorithms
from sklearn import metrics
```

In [46]:

```
score1=metrics.r2_score(y_test,y_pred1)
score2=metrics.r2_score(y_test,y_pred2)
score3=metrics.r2_score(y_test,y_pred3)
score4=metrics.r2_score(y_test,y_pred4)
```

In [47]:

```
print(score1,score2,score3,score4)
```

```
0.7833463107364539 -0.07229762787861826 0.8638934007572328
0.8779726251291786
```

In [48]:

```
s1=metrics.mean_absolute_error(y_test,y_pred1)
s2=metrics.mean_absolute_error(y_test,y_pred2)
s3=metrics.mean_absolute_error(y_test,y_pred3)
s4=metrics.mean_absolute_error(y_test,y_pred4)
```

In [49]:

```
print(s1,s2,s3,s4)
```

```
4186.508898366434 8592.428727899724 2500.6330625641485 2447.9515580545844
```

In [50]:

```
#predict charges
data={"age":40,
      "sex":1,
      "bmi":40.30,
      "Children":4,
```

```
     "smoker":1,
     "region":2}

df2=pd.DataFrame(data,index=[0])
df2
```

Out[50]:

	age	sex	bmi	Children	smoker	region
0	40	1	40.3	4	1	2

In [52]:

```
#charges prediction basis prediction 4
new_pred=gr.predict(df2)
print(new_pred)
```

```
[43013.23345491]
```

In [53]:

```
gr=GradientBoostingRegressor()
gr.fit(X,y)
```

Out[53]:

```
GradientBoostingRegressor()
```

In [54]:

```
import joblib
```

In [55]:

```
joblib.dump(gr,'model_joblib_gr')
```

Out[55]:

```
['model_joblib_gr']
```

In [56]:

```
model=joblib.load("model_joblib_gr")
```

In [57]:

```
model.predict(df2)
```

Out[57]:

```
array([42148.361888])
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: