```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
sat=pd.read_csv('https://github.com/ybifoundation/Dataset/raw/main/SAT%20GPA.csv')
```

```
sat.head()
```

	SAT	GPA
0	1270	3.4
1	1220	4.0
2	1160	3.8
3	950	3.8
4	1070	4.0

```
sat.describe()
```

	SAT	GPA
count	1000.000000	1000.000000
mean	1033.290000	3.203700
std	142.873681	0.542541
min	530.000000	1.800000
25%	930.000000	2.800000
50%	1030.000000	3.200000
75%	1130.000000	3.700000
max	1440.000000	4.500000

```
sat.info()
```

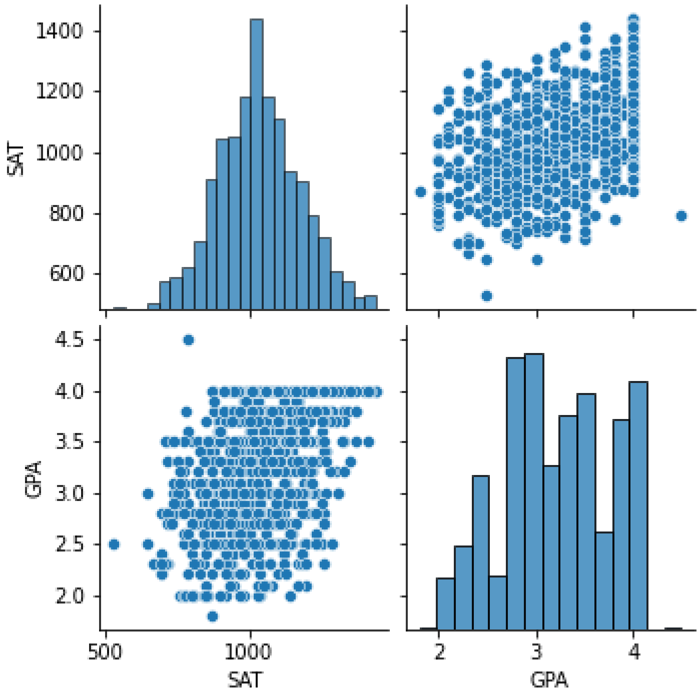
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   SAT     1000 non-null   int64
 1   GPA     1000 non-null   float64
dtypes: float64(1), int64(1)
memory usage: 15.8 KB
```

```
sat.corr()
```

	SAT	GPA
SAT	1.000000	0.429649
GPA	0.429649	1.000000

```
sns.pairplot(sat)
```

```
<seaborn.axisgrid.PairGrid at 0x7effccad2f90>
```



```
sat.columns
```

```
Index(['SAT', 'GPA'], dtype='object')
```

```
y=sat['SAT']
```

```
y.shape
```

```
(1000,)
```

```
x=sat[['GPA']]
```

```
x.shape
```

```
(1000, 1)
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,random_state=2529)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((700, 1), (300, 1), (700,), (300,))
```

```
x_train
```

	GPA
669	3.7
583	3.7
688	2.8
422	3.9
825	4.0
...	...
740	2.5
399	2.6
828	3.2
562	2.7
352	3.0

700 rows × 1 columns

```
from sklearn.linear_model import LinearRegression
```

```
reg=LinearRegression()
```

```
reg.fit(x_train,y_train)
```

```
LinearRegression()
```

```
reg.intercept_
```

```
673.2291896122774
```

```
reg.coef_
```

```
array([111.01584994])
```

```
y_pred=reg.predict(x_test)
```

```
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,r2_score
```

```
mean_absolute_error(y_test,y_pred)
```

```
     105.93877473699905
```

```
mean_absolute_percentage_error(y_test,y_pred)
```

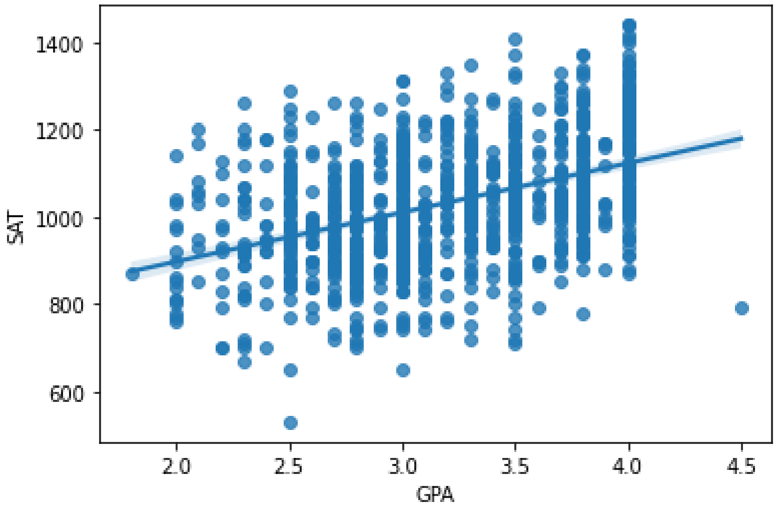
```
     0.10467104034918914
```

```
r2_score(y_test,y_pred)
```

```
     0.18785383761597474
```

```
sns.regplot(x='GPA',y='SAT',data=sat)
```

```
     <matplotlib.axes._subplots.AxesSubplot at 0x7effc794a8d0>
```



Multiple Regression

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
df=pd.read_csv('https://github.com/ybifoundation/Dataset/raw/main/Boston.csv')
```

```
df.head()
```

	CRIM	ZN	INDUS	CHAS	NX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    int64  
 4   NX       506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64  
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(12), int64(2)
memory usage: 55.5 KB
```

```
df.describe()
```

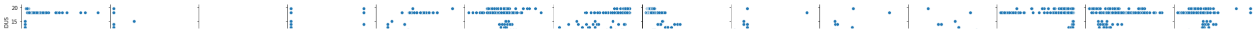
	CRIM	ZN	INDUS	CHAS	NX	RM	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.00
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.57
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.14
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.90
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.02
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.50
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.07
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.00

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7effc5916bd0>
```

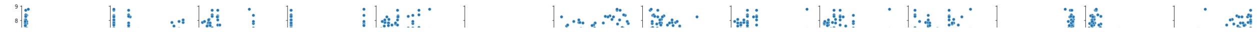
```
df.columns
```

```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'MEDV'],
      dtype='object')
```



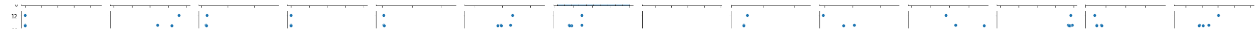
```
y=df['MEDV']
```

```
x=df[['CRIM', 'ZN', 'INDUS', 'CHAS', 'NX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT']]
```



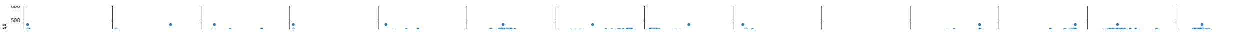
```
x.shape
```

```
(506, 13)
```

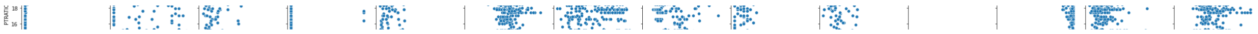


```
from sklearn.model_selection import train_test_split
```

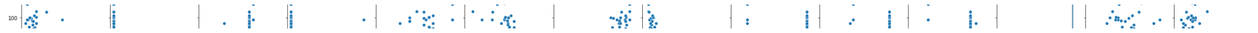
```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,random_state=2529)
```



```
from sklearn.preprocessing import StandardScaler
```



```
sc=StandardScaler
```



```
x_train
```

	CRIM	ZN	INDUS	CHAS	NX	RM	AGE	DIS	RAD	TAX	PTRATIO	
310	2.63548	0.0	9.90	0	0.544	4.973	37.8	2.5194	4	304.0	18.4	35
202	0.02177	82.5	2.03	0	0.415	7.610	15.7	6.2700	2	348.0	14.7	39
76	0.10153	0.0	12.83	0	0.437	6.279	74.5	4.0522	5	398.0	18.7	37
152	1.12658	0.0	19.58	1	0.871	5.012	88.0	1.6102	5	403.0	14.7	34
186	0.05602	0.0	2.46	0	0.488	7.831	53.6	3.1992	3	193.0	17.8	39
...	...	...	...	...	...	...	...	...	...	...	...	
228	0.29819	0.0	6.20	0	0.504	7.686	17.0	3.3751	8	307.0	17.4	37
399	9.91655	0.0	18.10	0	0.693	5.852	77.8	1.5004	24	666.0	20.2	33
316	0.31827	0.0	9.90	0	0.544	5.914	83.2	3.9986	4	304.0	18.4	39
50	0.08873	21.0	5.64	0	0.439	5.963	45.7	6.8147	4	243.0	16.8	39
352	0.07244	60.0	1.69	0	0.411	5.884	18.5	10.7103	4	411.0	18.3	39

354 rows × 13 columns

```
from sklearn.linear_model import LinearRegression
```

```
model=LinearRegression()
```

```
model.fit(x_train,y_train)
```

```
     LinearRegression()
```

```
model.intercept_
```

```
     34.21916368863014
```

```
model.coef_
```

```
     array([-1.29412069e-01,  3.65184937e-02,  1.54418944e-02,  2.35486887e+00,
            -2.04171489e+01,  4.41356565e+00,  4.61075512e-03, -1.58626723e+00,
             2.51478665e-01, -9.59591213e-03, -9.64169204e-01,  1.00972679e-02,
            -5.43198745e-01])
```

```
df.columns
```

```
     Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
            'PTRATIO', 'B', 'LSTAT', 'MEDV'],
           dtype='object')
```

```
y_pred=model.predict(x_test)
```

```
from sklearn.metrics import mean_absolute_percentage_error,r2_score
```

```
mean_absolute_percentage_error(y_test,y_pred)
```

```
     0.16355935882218034
```

```
r2_score(y_test,y_pred)
```

```
     0.6551914852365506
```

● ×