

HSE Level Student GPA

Import Libraries

In [2]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

Load DataSets

In [3]:

```
df=pd.read_csv("D:\\datasets\\train.csv")
```

Preprocessing

In [4]:

```
df.head()
```

Out[4]:

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	Ict	Total	Gpa
0	3	84.0	60	53	61	64	49	50	71	492	5.00
1	14	57.0	60	61	58	55	65	48	69	473	4.83
2	15	57.0	62	60	62	48	56	58	69	472	4.83
3	23	79.0	50	64	52	62	55	39	64	465	4.75
4	29	74.0	57	65	62	58	51	46	58	471	4.67

In [5]:

```
df.shape
```

Out[5]:

```
(126, 11)
```

In [6]:

```
df.dtypes
```

Out[6]:

```
Rank          int64
Bangla        float64
English        int64
Botany         int64
Zoology        int64
Physics        int64
Chemistry      int64
Math           int64
Ict            int64
Total          int64
Gpa            float64
dtype: object
```

In [7]:

```
df.isnull().head()
```

Out[7]:

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	Ict	Total	Gpa
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False

In [8]:

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

Out[8]:

```
Rank          0
Bangla         1
English        0
Botany         0
Zoology        0
Physics        0
Chemistry      0
Math           0
Ict            0
Total          0
Gpa            0
dtype: int64
```

In [9]:

```
df.describe()
```

Out[9]:

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	
count	126.000000	125.000000	126.000000	126.000000	126.000000	126.000000	126.000000	12
mean	243.976190	62.720000	56.746032	49.404762	54.373016	49.785714	50.396825	3
std	138.299065	9.915449	6.212164	7.793514	5.662133	7.578503	8.173204	
min	3.000000	35.000000	40.000000	31.000000	39.000000	31.000000	30.000000	1
25%	132.250000	56.000000	53.000000	44.000000	51.000000	45.000000	45.000000	2
50%	243.500000	62.000000	57.000000	50.500000	55.000000	51.000000	50.000000	3
75%	360.750000	70.000000	61.000000	55.000000	59.000000	56.000000	55.750000	3
max	520.000000	84.000000	72.000000	65.000000	65.000000	65.000000	71.000000	5

In [10]:

```
numeric=df.select_dtypes(include=[np.number])
numeric.head()
```

Out[10]:

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	Ict	Total	Gpa
0	3	84.0	60	53	61	64	49	50	71	492	5.00
1	14	57.0	60	61	58	55	65	48	69	473	4.83
2	15	57.0	62	60	62	48	56	58	69	472	4.83
3	23	79.0	50	64	52	62	55	39	64	465	4.75
4	29	74.0	57	65	62	58	51	46	58	471	4.67

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126 entries, 0 to 125
Data columns (total 11 columns):
Rank          126 non-null int64
Bangla        125 non-null float64
English       126 non-null int64
Botany        126 non-null int64
Zoology       126 non-null int64
Physics       126 non-null int64
Chemistry     126 non-null int64
Math          126 non-null int64
Ict           126 non-null int64
Total         126 non-null int64
Gpa           126 non-null float64
dtypes: float64(2), int64(9)
memory usage: 10.9 KB
```

In [12]:

df.tail()

Out[12]:

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	Ict	Total	Gpa
121	487	53.0	41	48	49	44	39	36	50	360	3.07
122	494	35.0	40	36	45	55	53	35	52	351	3.00
123	500	41.0	50	42	47	37	40	13	57	327	2.93
124	510	65.0	42	32	45	37	34	19	55	329	2.71
125	520	48.0	41	32	45	35	33	15	56	305	2.50

In [13]:

df.corr()

Out[13]:

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Mat
Rank	1.000000	-0.285798	-0.379579	-0.661717	-0.518789	-0.624258	-0.596382	-0.51944
Bangla	-0.285798	1.000000	0.050572	0.256126	0.055782	0.071915	-0.164281	-0.03817
English	-0.379579	0.050572	1.000000	0.203402	0.251762	0.095863	0.256308	0.06797
Botany	-0.661717	0.256126	0.203402	1.000000	0.498184	0.332922	0.264845	0.16485
Zoology	-0.518789	0.055782	0.251762	0.498184	1.000000	0.220006	0.274058	0.13816
Physics	-0.624258	0.071915	0.095863	0.332922	0.220006	1.000000	0.408225	0.31918
Chemistry	-0.596382	-0.164281	0.256308	0.264845	0.274058	0.408225	1.000000	0.26818
Math	-0.519448	-0.038173	0.067972	0.164851	0.138162	0.319186	0.268186	1.00000
Ict	-0.518344	0.149516	0.280337	0.322206	0.121924	0.227275	0.233696	0.16740
Total	-0.944456	0.383463	0.462419	0.699696	0.544367	0.622142	0.577815	0.49894
Gpa	-0.981024	0.309310	0.397711	0.668485	0.522812	0.624042	0.591072	0.55599

Data Exploration

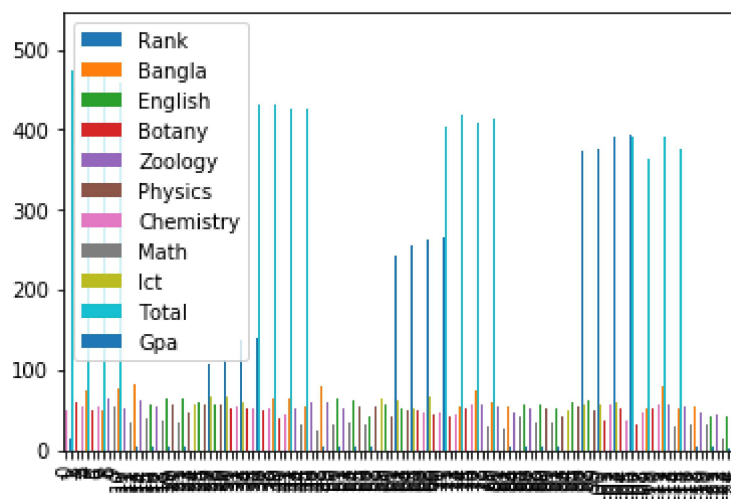
In [14]:

```
plt.figure(figsize=(20,10))  
corr=df.corr()  
df.plot(kind="bar")
```

Out[14]:

<matplotlib.axes._subplots.AxesSubplot at 0x1d1a3395c50>

<Figure size 1440x720 with 0 Axes>



In [15]:

```
df['Rank'].value_counts()
```

Out[15]:

255	1
348	1
457	1
70	1
71	1
72	1
73	1
330	1
331	1
332	1
333	1
88	1
89	1
90	1
347	1
349	1
56	1
350	1
100	1
360	1
361	1
362	1
363	1
108	1
109	1
110	1
111	1
373	1
374	1
375	1
..	
188	1
443	1
389	1
390	1
391	1
136	1
137	1
138	1
139	1
140	1
141	1
444	1
155	1
412	1
413	1
160	1
161	1
162	1
74	1
164	1
165	1
166	1
329	1
105	1

163	1
433	1
434	1
104	1
420	1
256	1

Name: Rank, Length: 126, dtype: int64

In [16]:

```
data=df
data
```

Out[16]:

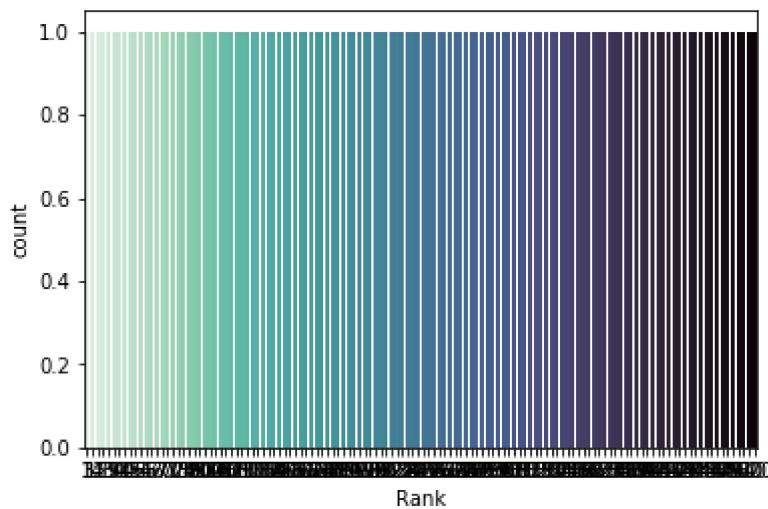
	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	Ict	Total	Gpa
0	3	84.0	60	53	61	64	49	50	71	492	5.00
1	14	57.0	60	61	58	55	65	48	69	473	4.83
2	15	57.0	62	60	62	48	56	58	69	472	4.83
3	23	79.0	50	64	52	62	55	39	64	465	4.75
4	29	74.0	57	65	62	58	51	46	58	471	4.67
5	30	77.0	63	49	56	45	62	47	68	467	4.67
6	31	80.0	54	61	63	57	55	29	61	460	4.67
7	40	50.0	61	62	60	63	63	39	68	466	4.64
8	45	66.0	53	57	64	58	49	46	61	454	4.58
9	46	61.0	62	55	63	53	47	54	58	453	4.58
10	54	76.0	54	59	62	51	50	39	68	459	4.50
11	55	60.0	55	51	51	56	70	51	64	458	4.50
12	56	64.0	65	58	60	56	55	34	65	457	4.50
13	57	83.0	55	59	53	62	41	35	67	455	4.50
14	70	72.0	55	49	63	60	59	37	59	454	4.42
15	71	63.0	67	55	62	59	50	39	56	451	4.42
16	72	53.0	56	49	54	65	68	35	70	450	4.42
17	73	50.0	58	52	54	55	71	44	65	449	4.42
18	74	75.0	45	60	56	64	51	36	62	449	4.42
19	88	56.0	65	52	56	54	63	34	67	447	4.33
20	89	63.0	60	58	53	57	66	33	54	444	4.33
21	90	73.0	55	51	59	48	60	34	61	441	4.33
22	100	62.0	65	56	60	48	52	54	64	461	4.29
23	104	61.0	67	48	58	48	50	42	72	446	4.25
24	105	70.0	60	52	51	51	55	48	58	445	4.25
25	106	72.0	60	51	54	56	50	35	67	445	4.25
26	107	58.0	47	56	59	57	57	45	64	443	4.25
27	108	56.0	65	53	57	56	56	32	68	443	4.25
28	109	68.0	58	47	59	57	67	39	47	442	4.25
29	110	78.0	60	58	53	57	42	33	60	441	4.25
...
96	363	59.0	50	36	40	55	42	56	48	386	3.58
97	373	66.0	65	42	58	51	47	23	57	409	3.57
98	374	62.0	62	60	45	37	52	26	56	400	3.57

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	Ict	Total	Gpa
99	375	75.0	52	49	44	49	40	30	60	399	3.57
100	376	70.0	51	44	47	54	46	30	56	398	3.57
101	388	69.0	57	36	50	48	39	38	61	398	3.50
102	389	71.0	56	34	48	48	58	28	55	398	3.50
103	390	71.0	57	44	48	43	44	31	60	398	3.50
104	391	NaN	58	51	55	34	53	22	53	397	3.50
105	392	73.0	55	58	48	44	37	20	60	395	3.50
106	393	67.0	57	42	51	39	42	29	65	392	3.50
107	412	55.0	58	31	45	39	46	44	60	378	3.43
108	413	51.0	51	40	64	41	46	27	47	367	3.43
109	417	53.0	55	38	39	45	37	42	55	364	3.42
110	418	70.0	57	51	58	36	48	29	48	397	3.36
111	419	62.0	61	37	50	49	58	32	47	396	3.36
112	420	79.0	58	43	53	32	45	21	59	390	3.36
113	433	48.0	53	37	58	50	43	44	46	379	3.33
114	434	58.0	52	49	59	40	40	29	48	375	3.33
115	443	53.0	52	35	52	55	49	23	58	377	3.29
116	444	65.0	56	51	55	45	37	18	49	376	3.29
117	457	77.0	50	40	45	38	35	31	56	372	3.21
118	458	55.0	45	47	52	32	45	32	64	372	3.21
119	459	68.0	53	38	48	42	41	29	52	371	3.21
120	474	40.0	60	39	55	51	30	31	53	359	3.17
121	487	53.0	41	48	49	44	39	36	50	360	3.07
122	494	35.0	40	36	45	55	53	35	52	351	3.00
123	500	41.0	50	42	47	37	40	13	57	327	2.93
124	510	65.0	42	32	45	37	34	19	55	329	2.71
125	520	48.0	41	32	45	35	33	15	56	305	2.50

126 rows × 11 columns

In [17]:

```
sns.countplot(x="Rank",data= data, palette="mako_r")  
plt.xlabel("Rank")  
plt.show()
```



In [18]:

```
countmax=len(data[data.Gpa >=4 ])  
countmin=len(data[data.Gpa <=3 ])
```

In [19]:

```
print("percentage of max score Gpa:{:.2f}%".format((countmax/(len(data.Gpa))*100)))  
print("percentage of min score Gpa:{:.2f}%".format((countmin/(len(data.Gpa))*100)))
```

```
percentage of max score Gpa:46.03%  
percentage of min score Gpa:3.17%
```

In [20]:

```
data.groupby("Gpa").mean()
```

Out[20]:

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	
Gpa									
2.50	520.0	48.000000	41.000000	32.000000	45.000000	35.000000	33.000000	15.000000	56.00
2.71	510.0	65.000000	42.000000	32.000000	45.000000	37.000000	34.000000	19.000000	55.00
2.93	500.0	41.000000	50.000000	42.000000	47.000000	37.000000	40.000000	13.000000	57.00
3.00	494.0	35.000000	40.000000	36.000000	45.000000	55.000000	53.000000	35.000000	52.00
3.07	487.0	53.000000	41.000000	48.000000	49.000000	44.000000	39.000000	36.000000	50.00
3.17	474.0	40.000000	60.000000	39.000000	55.000000	51.000000	30.000000	31.000000	53.00
3.21	458.0	66.666667	49.333333	41.666667	48.333333	37.333333	40.333333	30.666667	57.33
3.29	443.5	59.000000	54.000000	43.000000	53.500000	50.000000	43.000000	20.500000	53.50
3.33	433.5	53.000000	52.500000	43.000000	58.500000	45.000000	41.500000	36.500000	47.00
3.36	419.0	70.333333	58.666667	43.666667	53.666667	39.000000	50.333333	27.333333	51.33
3.42	417.0	53.000000	55.000000	38.000000	39.000000	45.000000	37.000000	42.000000	55.00
3.43	412.5	53.000000	54.500000	35.500000	54.500000	40.000000	46.000000	35.500000	53.50
3.50	390.5	70.200000	56.666667	44.166667	50.000000	42.666667	45.500000	28.000000	59.00
3.57	374.5	68.250000	57.500000	48.750000	48.500000	47.750000	46.250000	27.250000	57.25
3.58	361.5	62.000000	55.250000	43.750000	49.000000	46.250000	43.750000	37.000000	51.50
3.64	348.5	62.000000	56.500000	43.750000	55.500000	51.750000	46.750000	32.000000	59.00
3.67	331.0	57.400000	53.400000	44.400000	51.400000	46.200000	49.000000	36.000000	55.00
3.71	310.0	63.600000	55.400000	51.600000	55.600000	44.000000	48.400000	29.400000	62.60
3.75	294.5	62.250000	58.500000	47.250000	50.500000	48.500000	49.000000	32.000000	61.00
3.79	285.0	62.666667	57.333333	47.666667	53.333333	52.666667	50.000000	35.333333	62.66
3.83	263.5	57.333333	58.500000	47.000000	52.833333	46.166667	49.833333	33.166667	62.33
3.86	255.5	56.500000	59.000000	51.500000	61.500000	47.000000	54.500000	20.000000	58.50
3.92	243.0	50.333333	52.000000	46.000000	50.666667	51.000000	61.333333	39.000000	56.00
3.93	228.0	64.333333	62.000000	56.666667	58.666667	54.333333	50.333333	28.666667	64.66
4.00	198.5	65.600000	56.300000	57.000000	56.400000	52.500000	50.600000	32.900000	59.50
4.07	188.5	63.000000	59.000000	53.000000	56.500000	51.000000	55.500000	32.000000	60.00
4.08	163.0	64.285714	60.285714	48.571429	54.857143	51.000000	51.714286	36.285714	59.85
4.14	155.0	65.000000	53.000000	61.000000	65.000000	54.000000	51.000000	31.000000	63.00
4.17	138.5	60.000000	60.000000	52.833333	56.500000	51.166667	54.166667	38.333333	59.83
4.21	131.0	64.000000	67.000000	52.000000	56.000000	60.000000	62.000000	28.000000	64.00
4.25	107.5	68.125000	59.625000	50.250000	55.625000	54.750000	53.750000	38.250000	62.75
4.29	100.0	62.000000	65.000000	56.000000	60.000000	48.000000	52.000000	54.000000	64.00
4.33	89.0	64.000000	60.000000	53.666667	56.000000	53.000000	63.000000	33.666667	60.66
4.42	72.0	62.600000	56.200000	53.000000	57.800000	60.600000	59.800000	38.200000	62.40

	Rank	Bangla	English	Botany	Zoology	Physics	Chemistry	Math	
Gpa									
4.50	55.5	70.750000	57.250000	56.750000	56.500000	56.250000	54.000000	39.750000	66.00
4.58	45.5	63.500000	57.500000	56.000000	63.500000	55.500000	48.000000	50.000000	59.50
4.64	40.0	50.000000	61.000000	62.000000	60.000000	63.000000	63.000000	39.000000	68.00
4.67	30.0	77.000000	58.000000	58.333333	60.333333	53.333333	56.000000	40.666667	62.33
4.75	23.0	79.000000	50.000000	64.000000	52.000000	62.000000	55.000000	39.000000	64.00
4.83	14.5	57.000000	61.000000	60.500000	60.000000	51.500000	60.500000	53.000000	69.00
5.00	3.0	84.000000	60.000000	53.000000	61.000000	64.000000	49.000000	50.000000	71.00

importing Sklearn

In [21]:

```
from sklearn.model_selection import train_test_split
train,test=train_test_split(df,test_size=0.2,random_state=0)
```

Train and Test

In [22]:

```
x_train=train[['Rank','Physics','Chemistry','Zoology']]
y_train=train.Gpa
x_test=test[['Rank','Physics','Chemistry','Zoology']]
y_test=test.Gpa
```

In [23]:

```
x_train
```

Out[23]:

	Rank	Physics	Chemistry	Zoology
116	444	45	37	55
13	57	62	41	53
33	137	54	48	56
75	293	51	37	51
26	107	57	57	59
120	474	51	30	55
62	243	52	57	49
60	229	57	52	62
125	520	35	33	45
78	296	56	57	56
3	23	62	55	52
66	261	40	54	56
6	31	57	55	63
59	228	52	57	58
104	391	34	53	55
95	362	45	40	57
109	417	45	37	39
51	197	53	42	59
123	500	37	40	47
68	263	43	45	51
27	108	56	56	57
18	74	64	51	56
119	459	42	41	48
11	55	56	70	51
90	348	51	48	52
73	285	49	45	58
108	413	41	46	64
89	347	57	51	63
97	373	51	47	58
1	14	55	65	58
...
118	458	32	45	52
72	284	53	52	42
77	295	35	46	45
25	106	56	50	54

	Rank	Physics	Chemistry	Zoology
37	141	44	51	60
81	310	47	49	55
110	418	36	48	58
46	188	47	50	60
113	433	50	43	58
39	160	49	49	58
102	389	48	58	48
65	256	45	52	61
58	227	54	42	56
12	56	56	55	60
111	419	49	58	50
88	333	44	46	53
70	265	51	46	52
87	332	53	57	51
36	140	46	54	59
114	434	40	40	59
21	90	48	60	59
83	312	40	47	63
9	46	53	47	63
103	390	43	44	48
121	487	44	39	49
67	262	45	48	48
64	255	49	57	62
117	457	38	35	45
47	189	55	61	53
44	165	53	59	56

100 rows × 4 columns

In [24]:

x_test

Out[24]:

	Rank	Physics	Chemistry	Zoology
8	45	58	49	64
50	196	54	59	60
43	164	49	47	51
24	105	51	55	51
63	244	58	62	49
99	375	49	40	44
54	200	49	60	52
91	349	53	45	50
7	40	63	63	60
112	420	32	45	53
10	54	51	50	62
71	266	42	51	62
98	374	37	52	45
48	194	58	43	58
86	331	45	45	50
2	15	48	56	62
100	376	54	46	47
84	329	58	52	47
22	100	48	52	60
45	166	43	58	54
56	202	49	43	56
92	350	46	43	57
94	361	44	43	53
61	242	43	65	54
16	72	65	68	54
30	111	56	53	54

In [25]:

```
y_train
```

Out[25]:

116	3.29
13	4.50
33	4.17
75	3.75
26	4.25
120	3.17
62	3.92
60	3.93
125	2.50
78	3.75
3	4.75
66	3.83
6	4.67
59	3.93
104	3.50
95	3.58
109	3.42
51	4.00
123	2.93
68	3.83
27	4.25
18	4.42
119	3.21
11	4.50
90	3.64
73	3.79
108	3.43
89	3.64
97	3.57
1	4.83
	...
118	3.21
72	3.79
77	3.75
25	4.25
37	4.17
81	3.71
110	3.36
46	4.07
113	3.33
39	4.08
102	3.50
65	3.86
58	3.93
12	4.50
111	3.36
88	3.67
70	3.83
87	3.67
36	4.17
114	3.33
21	4.33
83	3.71
9	4.58
103	3.50


```
121    3.07
67     3.83
64     3.86
117    3.21
47     4.07
44     4.08
```

Name: Gpa, Length: 100, dtype: float64

In [26]:

```
y_test
```

Out[26]:

```
8      4.58
50     4.00
43     4.08
24     4.25
63     3.92
99     3.57
54     4.00
91     3.64
7      4.64
112    3.36
10     4.50
71     3.83
98     3.57
48     4.00
86     3.67
2      4.83
100    3.57
84     3.67
22     4.29
45     4.08
56     4.00
92     3.64
94     3.58
61     3.92
16     4.42
30     4.25
```

Name: Gpa, dtype: float64

In [27]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(100, 4)
(26, 4)
(100,)
(26,)
```

SGDRegressor

In [37]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import SGDRegressor
Scaler=StandardScaler()
x_train_std=Scaler.fit_transform(x_train.astype(float))
x_test_std=Scaler.fit_transform(x_test.astype(float))
```

Create instance and fit the data

In [38]:

```
sgdr=SGDRegressor(penalty="l2",alpha=0.15,max_iter=100)
sgdr.fit(x_train_std,y_train)
```

Out[38]:

```
SGDRegressor(alpha=0.15, average=False, epsilon=0.1, eta0=0.01,
             fit_intercept=True, l1_ratio=0.15, learning_rate='invscaling',
             loss='squared_loss', max_iter=100, n_iter=None, penalty='l2',
             power_t=0.25, random_state=None, shuffle=True, tol=None, verbose=0,
             warm_start=False)
```

predict the data

In [39]:

```
y_pred=sgdr.predict(x_test_std)
y_pred
```

Out[39]:

```
array([4.44702792, 4.04110505, 3.95819347, 4.17213033, 3.87798847,
       3.30963105, 3.93969655, 3.47924845, 4.54244193, 3.15823697,
       4.36203271, 3.73899761, 3.29527259, 3.97555999, 3.46790153,
       4.47534869, 3.39819494, 3.58439284, 4.2138401 , 3.98963983,
       3.87310287, 3.46606212, 3.39251913, 3.82562393, 4.45469312,
       4.20465817])
```

In [40]:

```
print(y_test[0:5])
```

```
8      4.58
50     4.00
43     4.08
24     4.25
63     3.92
Name: Gpa, dtype: float64
```

In [41]:

```
print(y_pred[0:5])
```

```
[4.44702792 4.04110505 3.95819347 4.17213033 3.87798847]
```

MSE Values

In [42]:

```
from sklearn import metrics
print("Mean Absolute Error:%0.3f" %metrics.mean_absolute_error(y_test,y_pred))
print("Mean Squared Error:%0.3f" %metrics.mean_squared_error(y_test,y_pred))
print("Root Mean Squared Error:%0.3f" %np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

Mean Absolute Error:0.130

Mean Squared Error:0.023

Root Mean Squared Error:0.152

In [43]:

```
print("Variance:%0.3f" %metrics.r2_score(y_test,y_pred))
```

Variance:0.839

True and Predict

In [44]:

```
print("true value:",y_test.values[0:10])
print("Predicted value:",y_pred[0:10])
```

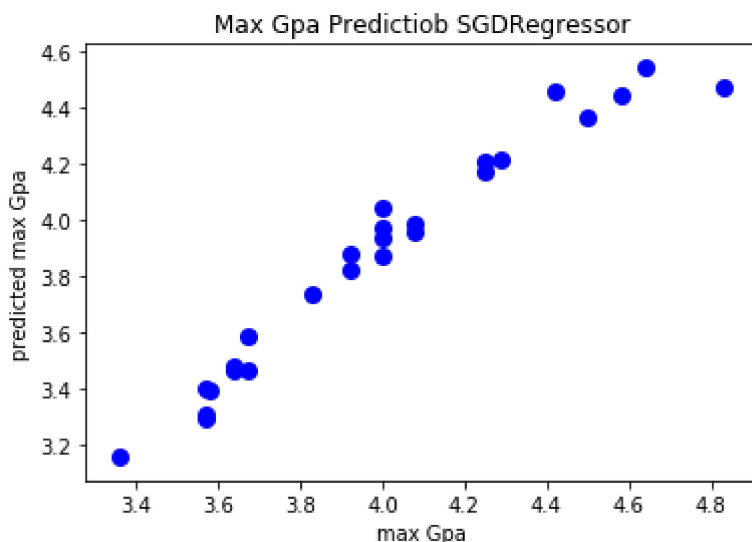
true value: [4.58 4. 4.08 4.25 3.92 3.57 4. 3.64 4.64 3.36]

Predicted value: [4.44702792 4.04110505 3.95819347 4.17213033 3.87798847 3.30963105

3.93969655 3.47924845 4.54244193 3.15823697]

In [45]:

```
plt.scatter(y_test,y_pred,color="blue", linewidth=3)
plt.xlabel('max Gpa')
plt.ylabel("predicted max Gpa")
plt.title("Max Gpa Prediction SGDRegressor")
plt.show()
```



In []: