In [1]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
```

In [ ]:
```
```

In [2]:
```
df=pd.read_csv('student.csv')
df
```

Out[2]:

	Unnamed: 0	Id	Student_Age	Sex	High_School_Type	Scholarship	Additional_Wo
0	0	5001	21	Male	Other	50%	Y
1	1	5002	20	Male	Other	50%	Y
2	2	5003	21	Male	State	50%	
3	3	5004	18	Female	Private	50%	Y
4	4	5005	22	Male	Private	50%	
...	...	...	...	...	...	...	
140	140	5141	22	Female	State	50%	Y
141	141	5142	18	Female	State	75%	
142	142	5143	18	Female	Private	75%	
143	143	5144	22	Female	State	75%	Y
144	144	5145	18	Female	Private	100%	

145 rows × 16 columns

In [3]:
```
df.head(5)
```

Out[3]:

	Unnamed: 0	Id	Student_Age	Sex	High_School_Type	Scholarship	Additional_Work
0	0	5001	21	Male	Other	50%	Yes
1	1	5002	20	Male	Other	50%	Yes
2	2	5003	21	Male	State	50%	No
3	3	5004	18	Female	Private	50%	Yes
4	4	5005	22	Male	Private	50%	No

In [4]:
```
df.tail(5)
```

Out[4]:

	Unnamed: 0	Id	Student_Age	Sex	High_School_Type	Scholarship	Additional_Wo
140	140	5141	22	Female	State	50%	Y
141	141	5142	18	Female	State	75%	
142	142	5143	18	Female	Private	75%	
143	143	5144	22	Female	State	75%	Y
144	144	5145	18	Female	Private	100%	

In [5]:
```
df.describe()
```

Out[5]:

	Unnamed: 0	Id	Student_Age	Weekly_Study_Hours
count	145.000000	145.000000	145.000000	145.000000
mean	72.000000	5073.000000	19.682759	2.331034
std	42.001984	42.001984	1.992010	4.249273
min	0.000000	5001.000000	18.000000	0.000000
25%	36.000000	5037.000000	18.000000	0.000000
50%	72.000000	5073.000000	19.000000	0.000000
75%	108.000000	5109.000000	21.000000	2.000000
max	144.000000	5145.000000	26.000000	12.000000

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145 entries, 0 to 144
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Unnamed: 0          145 non-null    int64 
 1   Id                  145 non-null    int64 
 2   Student_Age         145 non-null    int64 
 3   Sex                 145 non-null    object
 4   High_School_Type    145 non-null    object
 5   Scholarship         144 non-null    object
 6   Additional_Work     145 non-null    object
 7   Sports_activity     145 non-null    object
 8   Transportation      145 non-null    object
 9   Weekly_Study_Hours  145 non-null    int64 
 10  Attendance          145 non-null    object
 11  Reading             145 non-null    object
 12  Notes               145 non-null    object
 13  Listening_in_Class  145 non-null    object
 14  Project_work        145 non-null    object
 15  Grade               145 non-null    object
dtypes: int64(4), object(12)
memory usage: 18.3+ KB
```

In [7]:
```
df.shape
```

Out[7]:
```
(145, 16)
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Unnamed: 0          0
Id                  0
Student_Age         0
Sex                 0
High_School_Type    0
Scholarship         1
Additional_Work     0
Sports_activity     0
Transportation      0
Weekly_Study_Hours  0
Attendance          0
Reading             0
Notes               0
Listening_in_Class  0
Project_work        0
Grade               0
dtype: int64
```

In [9]:
```
df.groupby('Additional_Work').sum()
```

Out[9]:

Additional_Work	Unnamed: 0	Id	Student_Age	
No	6714	486810	1868	MaleMaleMaleMaleFemaleFemaleFemaleFen
Yes	3726	248775	986	MaleMaleFemaleFemaleFemaleMaleMaleMal

In [10]:
```
df.groupby('Attendance').sum()
```

Out[10]:

Attendance	Unnamed: 0	Id	Student_Age	
3	112	5113	20	
Always	6602	496700	1934	MaleMaleFemaleMaleMaleMaleFemaleFemaleFem
Never	1485	106506	424	MaleFemaleMaleMaleMaleMaleMaleMaleMal
Sometimes	2241	127266	476	FemaleFemaleFemaleMaleMaleMaleFemaleMaleF

In [11]:
```
df=df.drop(['Unnamed: 0','Id','Student_Age','Scholarship','Transportation
```

In [12]:
```
df
```

Out[12]:

	High_School_Type	Additional_Work	Weekly_Study_Hours	Attendance	Reading	Liste
0	Other	Yes	0	Always	Yes	
1	Other	Yes	0	Always	Yes	
2	State	No	2	Never	No	
3	Private	Yes	2	Always	No	
4	Private	No	12	Always	Yes	
...	...	...	...	...	...	
140	State	Yes	0	Always	No	
141	State	No	0	Never	No	
142	Private	No	0	Always	Yes	
143	State	Yes	12	Sometimes	No	
144	Private	No	12	Always	Yes	

145 rows × 8 columns

In [13]:
```
Q1 = df['Weekly_Study_Hours'].quantile(0.25)
Q3 = df['Weekly_Study_Hours'].quantile(0.75)
IQR = Q3 - Q1
df = df[(df['Weekly_Study_Hours'] >= Q1 - 1.5 * IQR) & (df['Weekly_Study_H
```

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 120 entries, 0 to 142
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   High_School_Type    120 non-null    object
 1   Additional_Work     120 non-null    object
 2   Weekly_Study_Hours  120 non-null    int64 
 3   Attendance          120 non-null    object
 4   Reading             120 non-null    object
 5   Listening_in_Class  120 non-null    object
 6   Project_work        120 non-null    object
 7   Grade               120 non-null    object
dtypes: int64(1), object(7)
memory usage: 8.4+ KB
```

In [15]:
```
df['Grade'] = df['Grade'].map({'AA': 1, 'BA': 0.875, 'BB': 0.750, 'CB': 0
df['Attendance']=df['Attendance'].map({'Always': 1.5, 'Sometimes': 1.0, 'N
df['Project_work']=df['Project_work'].map({'Yes':1,'No':0})
df['Reading']=df['Reading'].map({'Yes':1,'No':0})

df['Listening_in_Class']=df['Listening_in_Class'].map({'Yes':1,'No':0})
df['Additional_Work']=df['Additional_Work'].map({'Yes':1,'No':0})
```

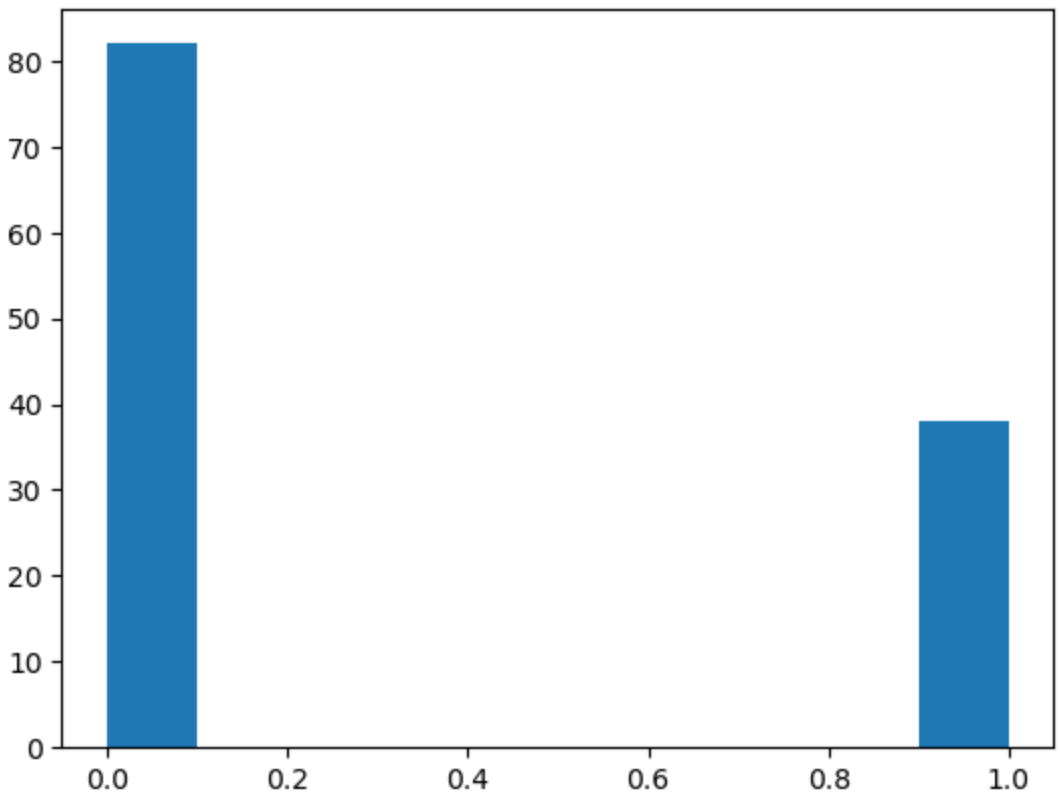
In [16]:
```
df=pd.get_dummies(df,dtype=int)
```

In [17]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 120 entries, 0 to 142
Data columns (total 10 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Additional_Work          120 non-null    int64
 1   Weekly_Study_Hours       120 non-null    int64
 2   Attendance               120 non-null    float64
 3   Reading                  120 non-null    int64
 4   Listening_in_Class       120 non-null    int64
 5   Project_work             120 non-null    int64
 6   Grade                    120 non-null    float64
 7   High_School_Type_Other   120 non-null    int32
 8   High_School_Type_Private  120 non-null    int32
 9   High_School_Type_State   120 non-null    int32
dtypes: float64(2), int32(3), int64(5)
memory usage: 8.9 KB
```

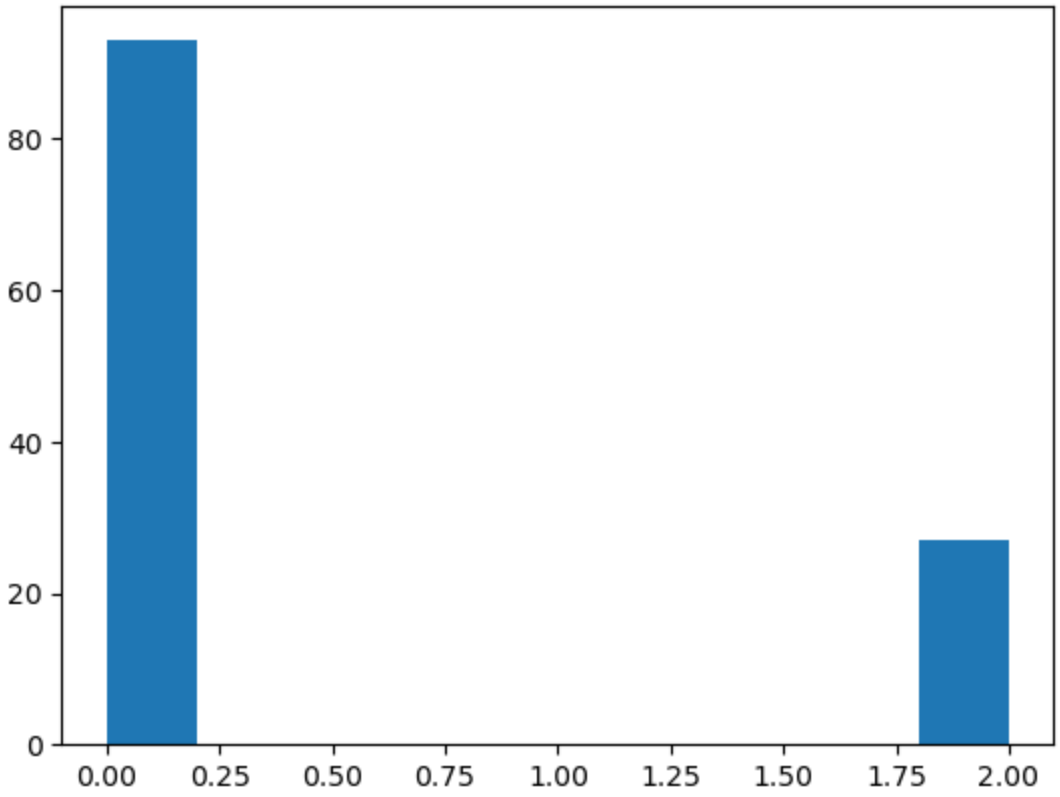
In [18]:
```
import matplotlib.pyplot as plt
plt.hist(df['Additional_Work'])
```

Out[18]:
```
(array([82.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0., 38.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```



In [19]:
```
import matplotlib.pyplot as plt
plt.hist(df['Weekly_Study_Hours'])
```

Out[19]:
```
(array([93.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0., 27.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <BarContainer object of 10 artists>)
```



In [20]:
```
cor_mat=df.corr()
```

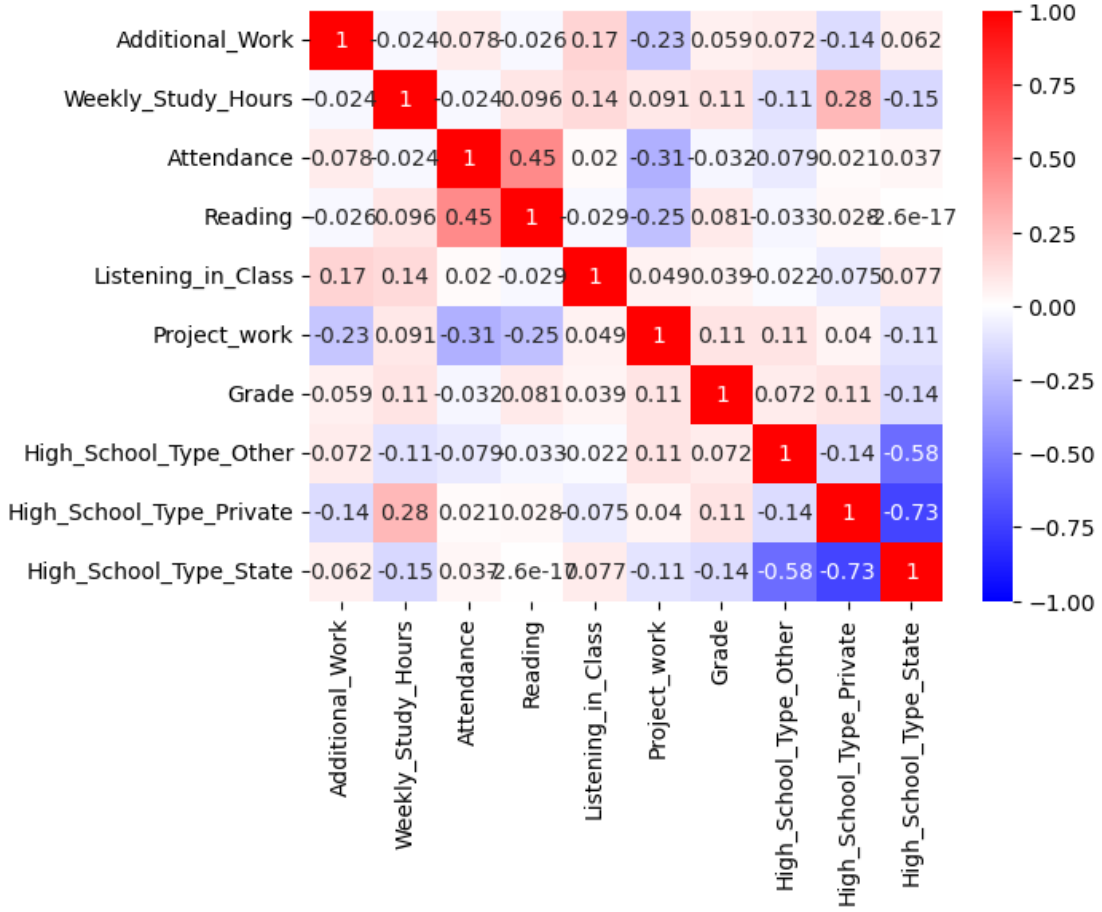
In [21]:
```
cor_mat
```

Out[21]:

	Additional_Work	Weekly_Study_Hours	Attendance	Reading
Additional_Work	1.000000	-0.023595	0.077880	-2.633298e-02
Weekly_Study_Hours	-0.023595	1.000000	-0.024481	9.600307e-02
Attendance	0.077880	-0.024481	1.000000	4.543984e-01
Reading	-0.026333	0.096003	0.454398	1.000000e+00
Listening_in_Class	0.169967	0.144005	0.020132	-2.901786e-02
Project_work	-0.226350	0.090813	-0.312465	-2.494785e-01
Grade	0.059440	0.105675	-0.031898	8.130353e-02
High_School_Type_Other	0.071657	-0.113085	-0.078912	-3.340766e-02
High_School_Type_Private	-0.135460	0.276648	0.021095	2.806804e-02
High_School_Type_State	0.062057	-0.149782	0.037276	-2.569667e-17

In [22]:
```
import seaborn as sns
sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidth=0,cmap='bwr')
```

Out[22]: <Axes: >



In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 120 entries, 0 to 142
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Additional_Work           120 non-null    int64
 1   Weekly_Study_Hours        120 non-null    int64
 2   Attendance                120 non-null    float64
 3   Reading                   120 non-null    int64
 4   Listening_in_Class        120 non-null    int64
 5   Project_work              120 non-null    int64
 6   Grade                     120 non-null    float64
 7   High_School_Type_Other    120 non-null    int32
 8   High_School_Type_Private  120 non-null    int32
 9   High_School_Type_State    120 non-null    int32
dtypes: float64(2), int32(3), int64(5)
memory usage: 8.9 KB
```

In [24]:
```
df.isnull().sum()
```

Out[24]:
```
Additional_Work             0
Weekly_Study_Hours          0
Attendance                  0
Reading                     0
Listening_in_Class          0
Project_work                0
Grade                       0
High_School_Type_Other      0
High_School_Type_Private    0
High_School_Type_State      0
dtype: int64
```

Linear Regression

In [25]:
```
y=df['Grade']
y
```

Out[25]:
```
0      1.000
1      1.000
2      1.000
3      1.000
5      0.875
       ...
137    0.875
139    0.000
140    0.500
141    0.500
142    1.000
Name: Grade, Length: 120, dtype: float64
```

In [26]:
```
x=df.drop('Grade',axis=1)
x
```

Out[26]:

	Additional_Work	Weekly_Study_Hours	Attendance	Reading	Listening_in_Class	Proje
0	1	0	1.5	1	0	
1	1	0	1.5	1	1	
2	0	2	0.5	0	0	
3	1	2	1.5	0	0	
5	0	2	1.5	1	1	
...	...	...	...	...	...	
137	0	0	1.5	1	0	
139	1	2	1.0	0	1	
140	1	0	1.5	0	0	
141	0	0	0.5	0	1	
142	0	0	1.5	1	0	

120 rows × 9 columns

In [27]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33,randor
x_test.head(5)
```

Out[27]:

	Additional_Work	Weekly_Study_Hours	Attendance	Reading	Listening_in_Class	Projec
56	0	0	1.5	1	1	
61	0	2	1.5	0	1	
5	0	2	1.5	1	1	
71	0	0	1.5	0	1	
33	1	0	1.5	1	0	

In [28]:
```
x_train.shape
```

Out[28]: (80, 9)

In [29]:
```
y_train.shape
```

Out[29]: (80,)

In [30]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression() #creating object of LinearRegression
reg.fit(x_train,y_train)
LinearRegression()
#training and fitting
```

Out[30]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [31]:
```
ypred = reg.predict(x_test)
ypred
```

Out[31]:
```
array([0.59790319, 0.72551973, 0.67346423, 0.69561043, 0.57117139,
       0.75487754, 0.59503864, 0.4938383 , 0.72472669, 0.65796333,
       0.44818655, 0.65796333, 0.51111124, 0.51111124, 0.76226702,
       0.62627252, 0.75622498, 0.73180332, 0.59503864, 0.59503864,
       0.5082467 , 0.57117139, 0.79779651, 0.55676299, 0.68787264,
       0.55676299, 0.55071147, 0.6463011 , 0.51111124, 0.61817057,
       0.5082467 , 0.49944901, 0.84378709, 0.59790319, 0.58063024,
       0.57251882, 0.58063024, 0.55676299, 0.63750342, 0.55676299])
```

In [ ]:
```
X_pred=reg.predict()
```

In [32]:
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[32]:
```
-0.1998852439459411
```

In [33]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[33]:
```
0.08341780323878081
```

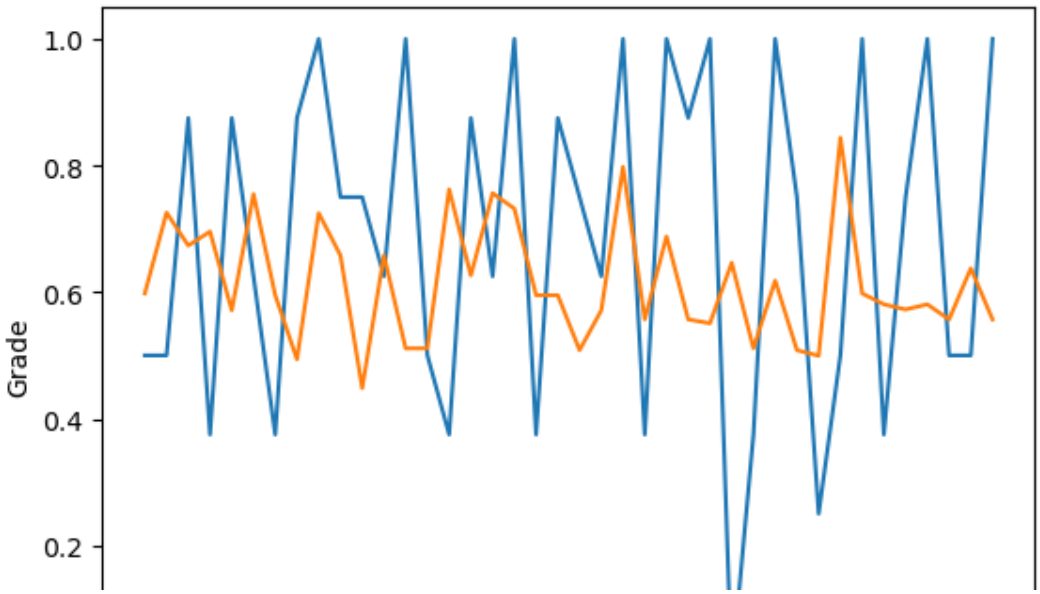
In [34]:
```
Results=pd.DataFrame(columns=['Grade','Predicated'])
Results['Grade']=y_test
Results['Predicated']=ypred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(15)
```

Out[34]:

	index	Grade	Predicated	Id
0	56	0.500	0.597903	0
1	61	0.500	0.725520	1
2	5	0.875	0.673464	2
3	71	0.375	0.695610	3
4	33	0.875	0.571171	4
5	80	0.625	0.754878	5
6	91	0.375	0.595039	6
7	15	0.875	0.493838	7
8	50	1.000	0.724727	8
9	129	0.750	0.657963	9
10	25	0.750	0.448187	10
11	78	0.625	0.657963	11
12	16	1.000	0.511111	12
13	46	0.500	0.511111	13
14	108	0.375	0.762267	14

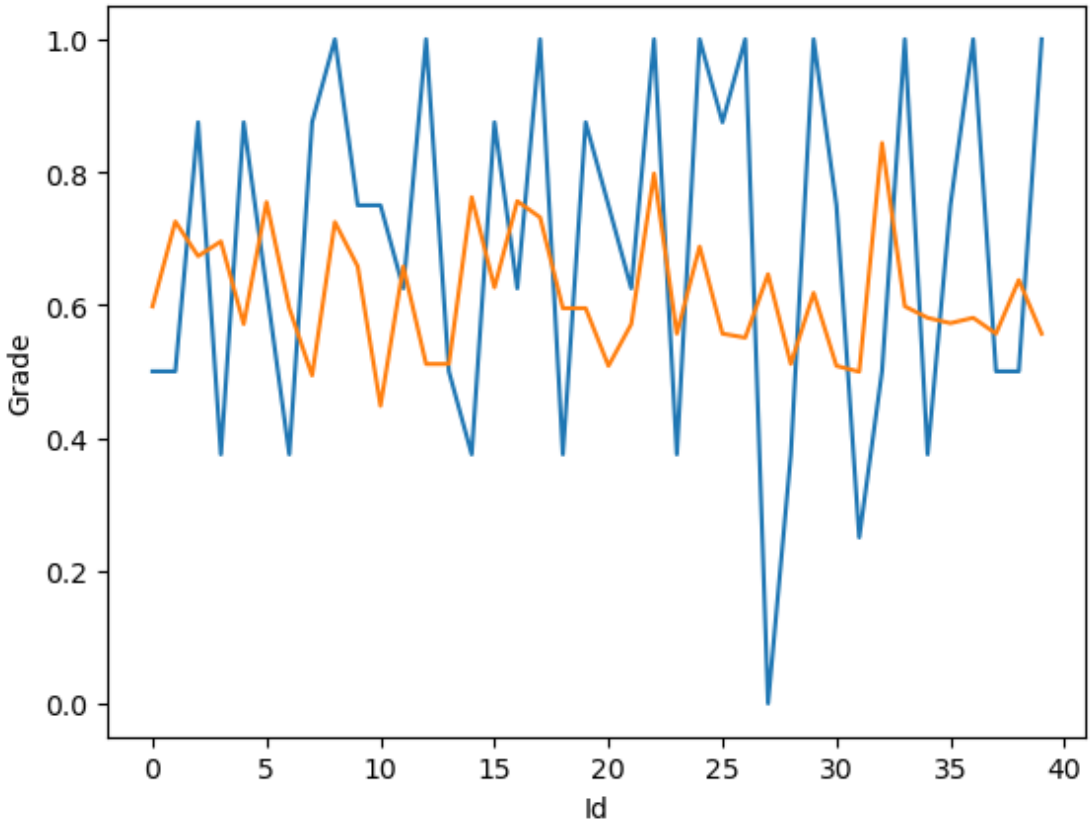
In [35]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='Grade',data=Results.head(50))
sns.lineplot(x='Id',y='Predicated',data=Results.head(50))
plt.plot()
```

Out[35]: []



In [36]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='Grade',data=Results.tail(50))
sns.lineplot(x='Id',y='Predicated',data=Results.tail(50))
plt.plot()
```

Out[36]: []



Type *Markdown* and LaTeX: α^2