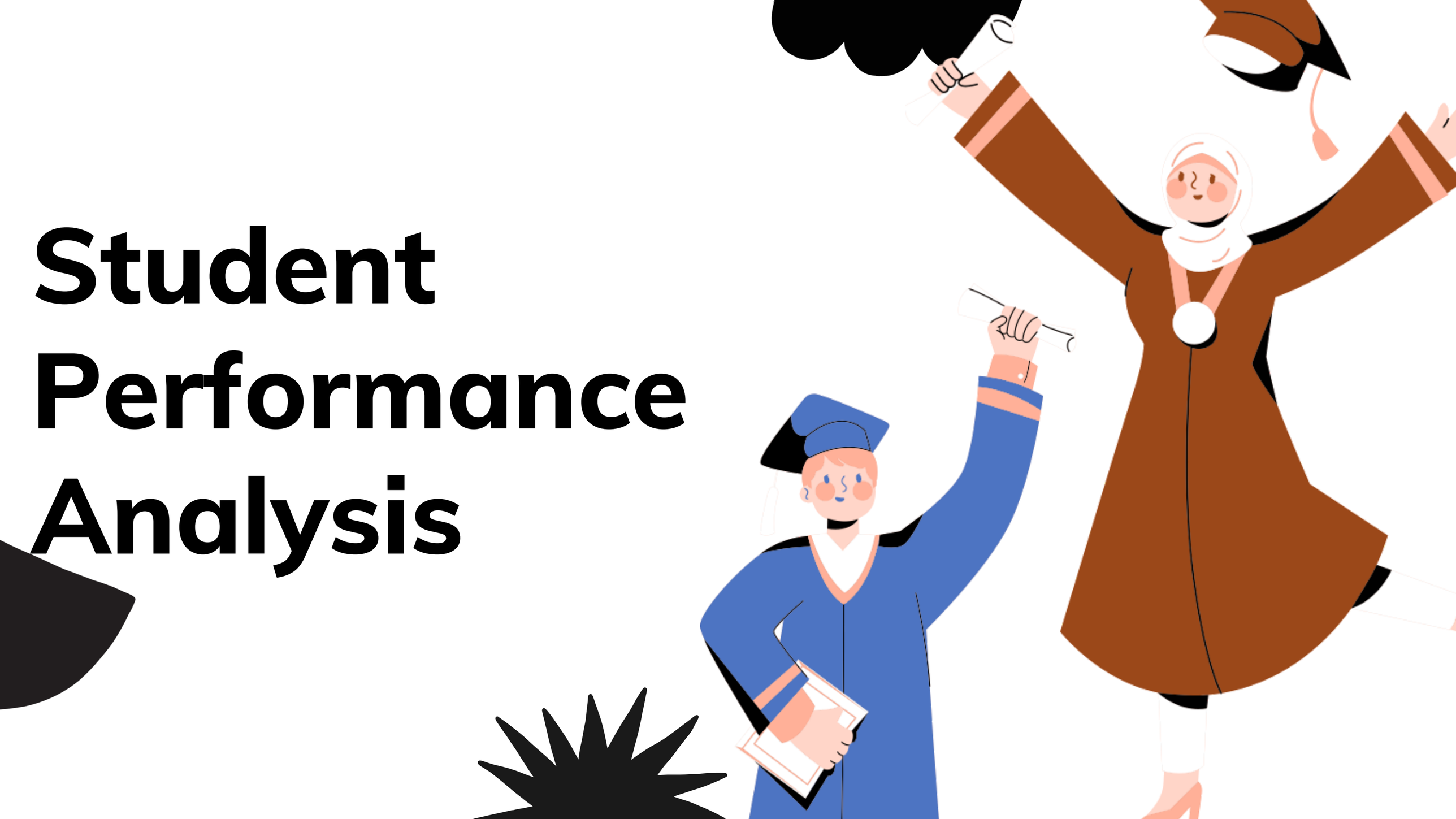
Student Performance Analysis



About

Hello, I'm Charish

PGP-24 (Hyderabad)

LinkdIn - https://www.linkedin.com/in/charish-bhimarasetty-12870616a/

PROJECT BACKGROUND

Problem Statement

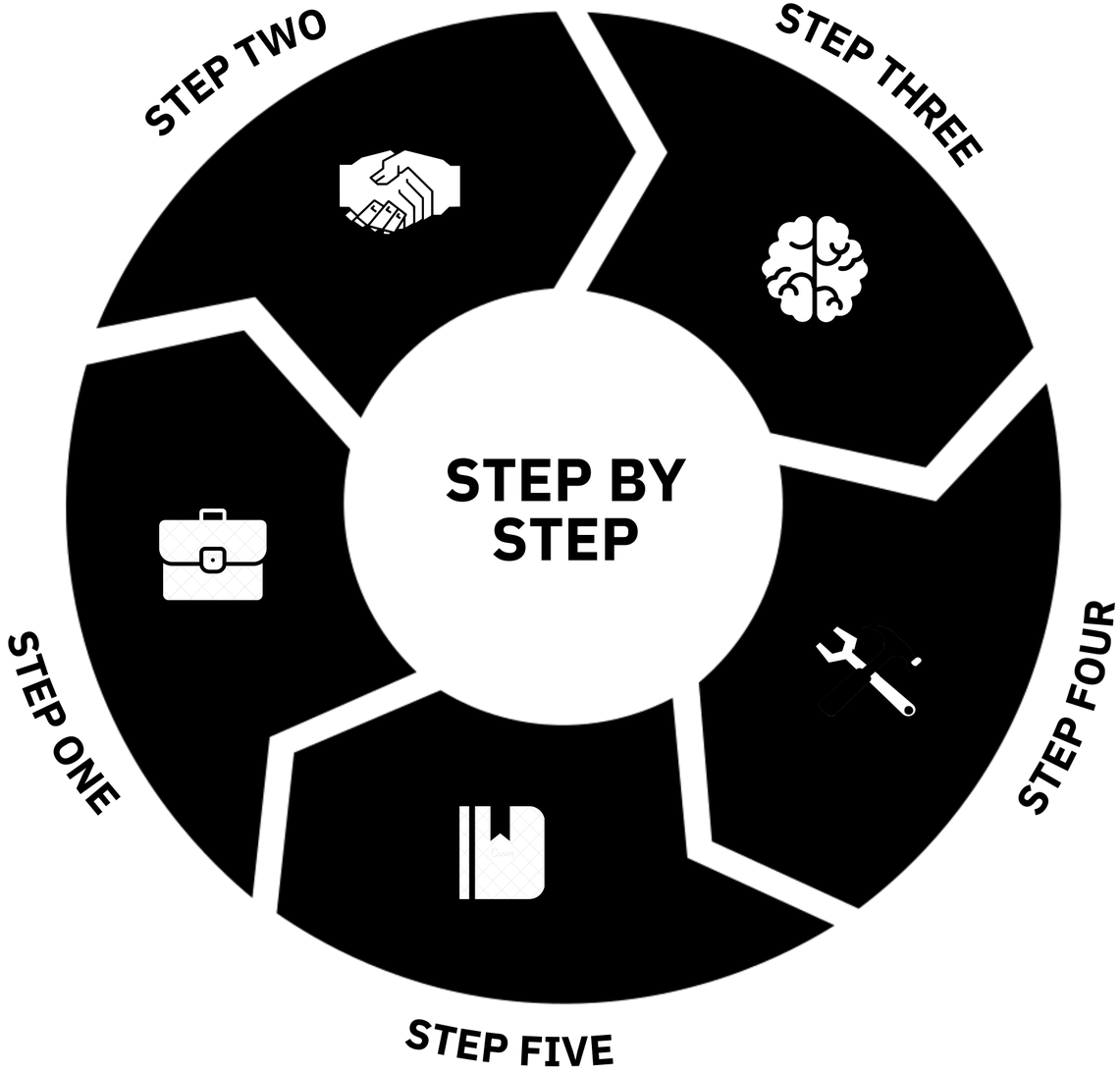
- Success in school is essential. Academically successful adults are more likely to have job opportunities. Also, those who succeed intellectually are less likely to commit crimes.
- Not every student comes from a similar background. Some kids' families provide them with unwavering support, while other students' families are discordant. Success in the classroom can be impacted by a variety of factors. To that purpose, based on the student's background, I will develop a machine learning model to forecast the worth of both math and Portuguese courses. I sourced the datasets I used from UCI.

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). In [Cortez and Silva, 2008], the two datasets were modeled under binary/five-level classification and regression tasks. Important note: the target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. It is more difficult to predict G3 without G2 and G1, but such prediction is much more useful

Data Overview

STEP BY STEP

- **STEP ONE**
Dataset
- **STEP TWO**
Exploratory Data Analysis
- **STEP THREE**
Data Preprocess-sing
- **STEP FOUR**
Modelling And Evaluation
- **STEP FIVE**
Business Insight and Recommendations



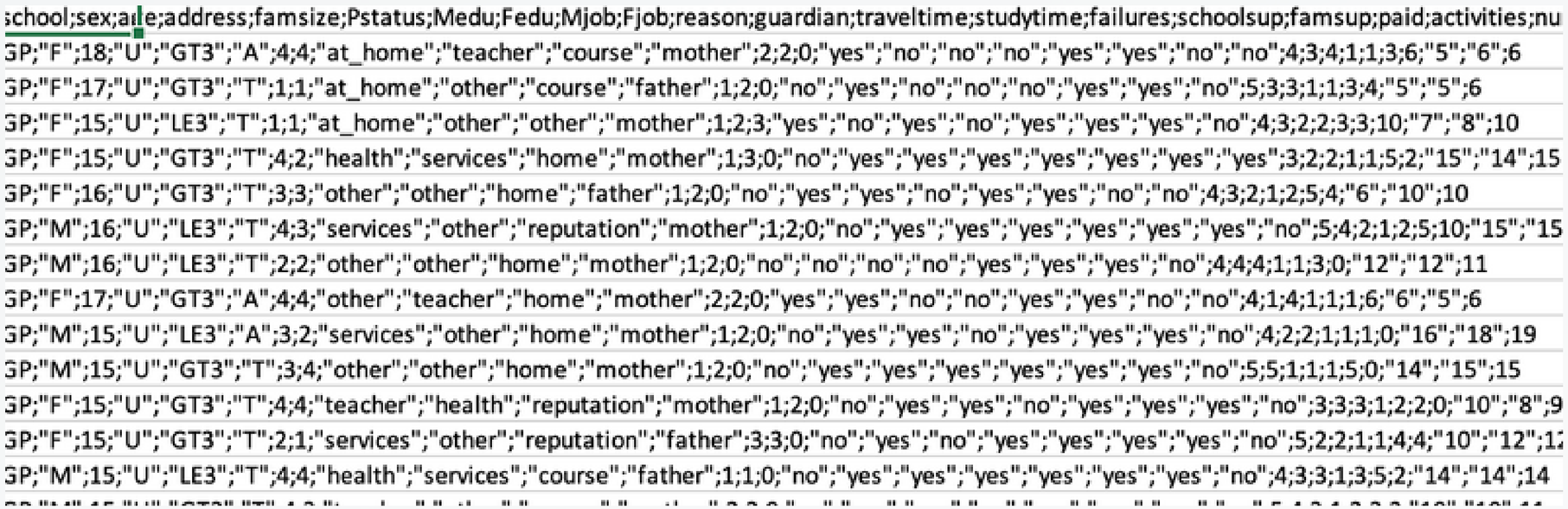
STEP 2

Data Set

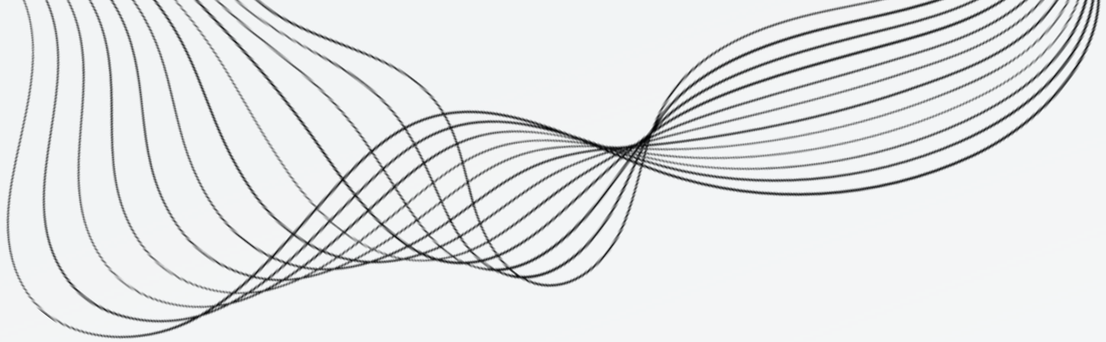
Firstly, the data was in an unorganised format, used the 'DELIMITER' function to organise the data and change the file format from .csv to .excel.

Created a new Total Grade Column by take average of all the 3 grade columns

```
#Creating The Total Grade Column
data['total grade'] = (data['G1']+data['G2']+data['G3'])/3
data = data.drop(['G1','G2','G3'],axis=1)
```



school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;studytime;failures;schoolsup;famsup;paid;activities;nu
GP;"F";18;"U";"GT3";"A";4;4;"at_home";"teacher";"course";"mother";2;2;0;"yes";"no";"no";"no";"yes";"yes";"no";"no";4;3;4;1;1;3;6;"5";"6";6
GP;"F";17;"U";"GT3";"T";1;1;"at_home";"other";"course";"father";1;2;0;"no";"yes";"no";"no";"no";"yes";"yes";"no";5;3;3;1;1;3;4;"5";"5";6
GP;"F";15;"U";"LE3";"T";1;1;"at_home";"other";"other";"mother";1;2;3;"yes";"no";"yes";"no";"yes";"yes";"yes";"no";4;3;2;2;3;3;10;"7";"8";10
GP;"F";15;"U";"GT3";"T";4;2;"health";"services";"home";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"yes";3;2;2;1;1;5;2;"15";"14";15
GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"home";"father";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"no";"no";4;3;2;1;2;5;4;"6";"10";10
GP;"M";16;"U";"LE3";"T";4;3;"services";"other";"reputation";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;4;2;1;2;5;10;"15";"15
GP;"M";16;"U";"LE3";"T";2;2;"other";"other";"home";"mother";1;2;0;"no";"no";"no";"no";"yes";"yes";"yes";"no";4;4;4;1;1;3;0;"12";"12";11
GP;"F";17;"U";"GT3";"A";4;4;"other";"teacher";"home";"mother";2;2;0;"yes";"yes";"no";"no";"yes";"yes";"no";"no";4;1;4;1;1;1;6;"6";"5";6
GP;"M";15;"U";"LE3";"A";3;2;"services";"other";"home";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";4;2;2;1;1;1;0;"16";"18";19
GP;"M";15;"U";"GT3";"T";3;4;"other";"other";"home";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;5;1;1;1;5;0;"14";"15";15
GP;"F";15;"U";"GT3";"T";4;4;"teacher";"health";"reputation";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";3;3;3;1;2;2;0;"10";"8";9
GP;"F";15;"U";"GT3";"T";2;1;"services";"other";"reputation";"father";3;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;2;2;1;1;4;4;"10";"12";1
GP;"M";15;"U";"LE3";"T";4;4;"health";"services";"course";"father";1;1;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";4;3;3;1;3;5;2;"14";"14";14



PREPROCESSING DATA

STAGE 2 IS ANOTHER NEXT STEP THAT WE DID MANIPULATION ON DATA BEFORE IT IS USED IN ORDER TO BUILD THE MODEL.

WHAT WE HAVE DONE ON THIS STAGE:

HANDLE MISSING VALUES

HANDLE DUPLICATED DATA

HANDLE OUTLIERS

FEATURE TRANSFORMATION

FEATURE ENCODING

HANDLE CLASS IMBALANCE

FEATURE SELECTION

FEATURE EXTRACTION

Data Info

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1044 entries, 0 to 648
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   school      1044 non-null   object
 1   sex         1044 non-null   object
 2   age         1044 non-null   int64 
 3   address     1044 non-null   object
 4   famsize     1044 non-null   object
 5   Pstatus     1044 non-null   object
 6   Medu        1044 non-null   int64 
 7   Fedu        1044 non-null   int64 
 8   Mjob        1044 non-null   object
 9   Fjob        1044 non-null   object
 10  reason      1044 non-null   object
 11  guardian    1044 non-null   object
 12  traveltime  1044 non-null   int64 
 13  studytime   1044 non-null   int64 
 14  failures    1044 non-null   int64 
 15  schoolsup   1044 non-null   object
 16  famsup      1044 non-null   object
 17  paid        1044 non-null   object
 18  activities  1044 non-null   object
 19  nursery     1044 non-null   object
 20  higher      1044 non-null   object
 21  internet    1044 non-null   object
 22  romantic    1044 non-null   object
 23  famrel      1044 non-null   int64 
 24  freetime    1044 non-null   int64 
 25  goout       1044 non-null   int64 
 26  Dalc        1044 non-null   int64 
 27  Walc        1044 non-null   int64 
 28  health      1044 non-null   int64 
 29  absences    1044 non-null   int64 
 30  G1          1044 non-null   int64 
 31  G2          1044 non-null   int64 
 32  G3          1044 non-null   int64 
dtypes: int64(16), object(17)
memory usage: 277.3+ KB
```

DATA SET

Missing Values:

No Missing Values

Shape:

```
# getting the shape of the data
data.shape
```

```
(1044, 33)
```

Duplicate Values:

```
duplicated=data.duplicated().sum()
if duplicated:
    print('Duplicated rows in dataset are {}'.format(duplicated))
else:
    print('Dataset contains no duplicate values')
```

```
Duplicated rows in dataset are 2
```

```
duplicated=data[data.duplicated(keep=False)]
duplicated.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	
30	GP	M	15	U	GT3	T	4	4	health	services	...	yes	no	5	4	2	3	4	5	0	1
71	GP	M	15	U	GT3	T	4	2	other	other	...	yes	no	3	3	3	1	1	3	0	1
30	GP	M	15	U	GT3	T	4	4	health	services	...	yes	no	5	4	2	3	4	5	0	1
71	GP	M	15	U	GT3	T	4	2	other	other	...	yes	no	3	3	3	1	1	3	0	1

4 rows × 31 columns

```
data=data.drop_duplicates(keep=False)
```

STEP 3

EDA

EXPLORATORY DATA ANALYSIS

DESCRIPTIVE STATISTICS

Dtypes:

Initially, there are 14 numeric and 19 object columns
they are in binary from change them to object columns by using .astype()

Correlation:

We can observe that the target variable and 8 characteristics highly correlate. Some of them correlate positively, while others negatively. For instance, the failures feature has the biggest negative association. It makes sense that a student who has received terrible grades in the past could continue to do so in the future.

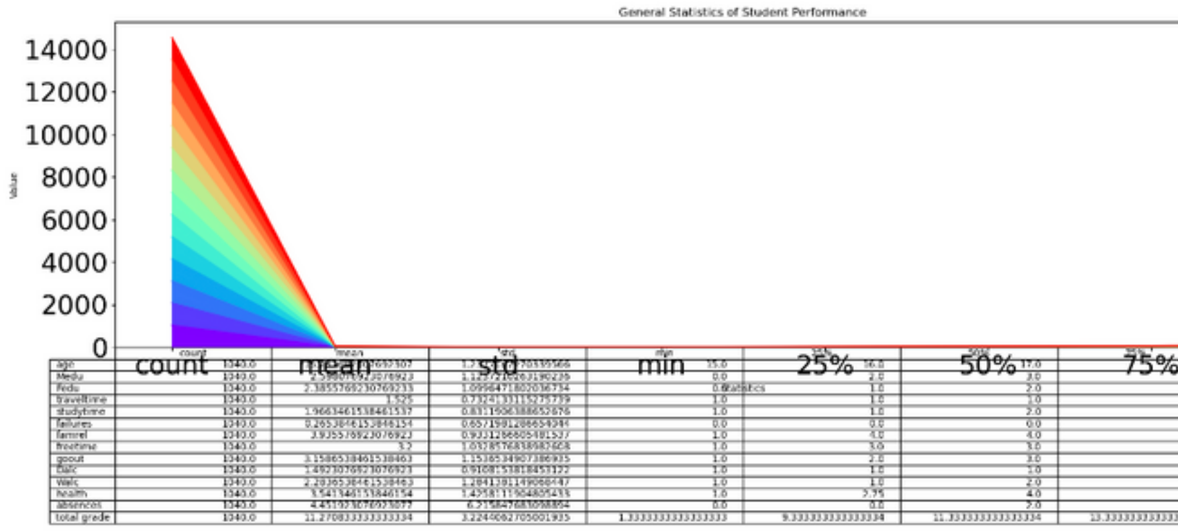
Also, a student's final grade decreases the more alcohol they consume over the weekend or during the week. Freetime and traveltime factors also have a negative association. The student's performance declines as more free time is available to him or her.

Descriptive Statistics

```
# describing the dataset
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	1044.0	16.726054	1.239975	15.000000	16.000000	17.000000	18.000000	22.000000
Medu	1044.0	2.603448	1.124907	0.000000	2.000000	3.000000	4.000000	4.000000
Fedu	1044.0	2.387931	1.099938	0.000000	1.000000	2.000000	3.000000	4.000000
traveltime	1044.0	1.522989	0.731727	1.000000	1.000000	1.000000	2.000000	4.000000
studytime	1044.0	1.970307	0.834353	1.000000	1.000000	2.000000	2.000000	4.000000
failures	1044.0	0.264368	0.656142	0.000000	0.000000	0.000000	0.000000	3.000000
famrel	1044.0	3.935824	0.933401	1.000000	4.000000	4.000000	5.000000	5.000000
freetime	1044.0	3.201149	1.031507	1.000000	3.000000	3.000000	4.000000	5.000000
goout	1044.0	3.156130	1.152575	1.000000	2.000000	3.000000	4.000000	5.000000
Dalc	1044.0	1.494253	0.911714	1.000000	1.000000	1.000000	2.000000	5.000000
Walc	1044.0	2.284483	1.285105	1.000000	1.000000	2.000000	3.000000	5.000000
health	1044.0	3.543103	1.424703	1.000000	3.000000	4.000000	5.000000	5.000000
absences	1044.0	4.434866	6.210017	0.000000	0.000000	2.000000	6.000000	75.000000
total grade	1044.0	11.267241	3.218805	1.333333	9.333333	11.333333	13.333333	19.333333

```
data.describe().plot(kind = "area",fontsize=27, figsize = (23,6), table = True,colormap="rainbow")
plt.xlabel('Statistics',)
plt.ylabel('Value')
plt.title("General Statistics of Student Performance")
plt.show()
```



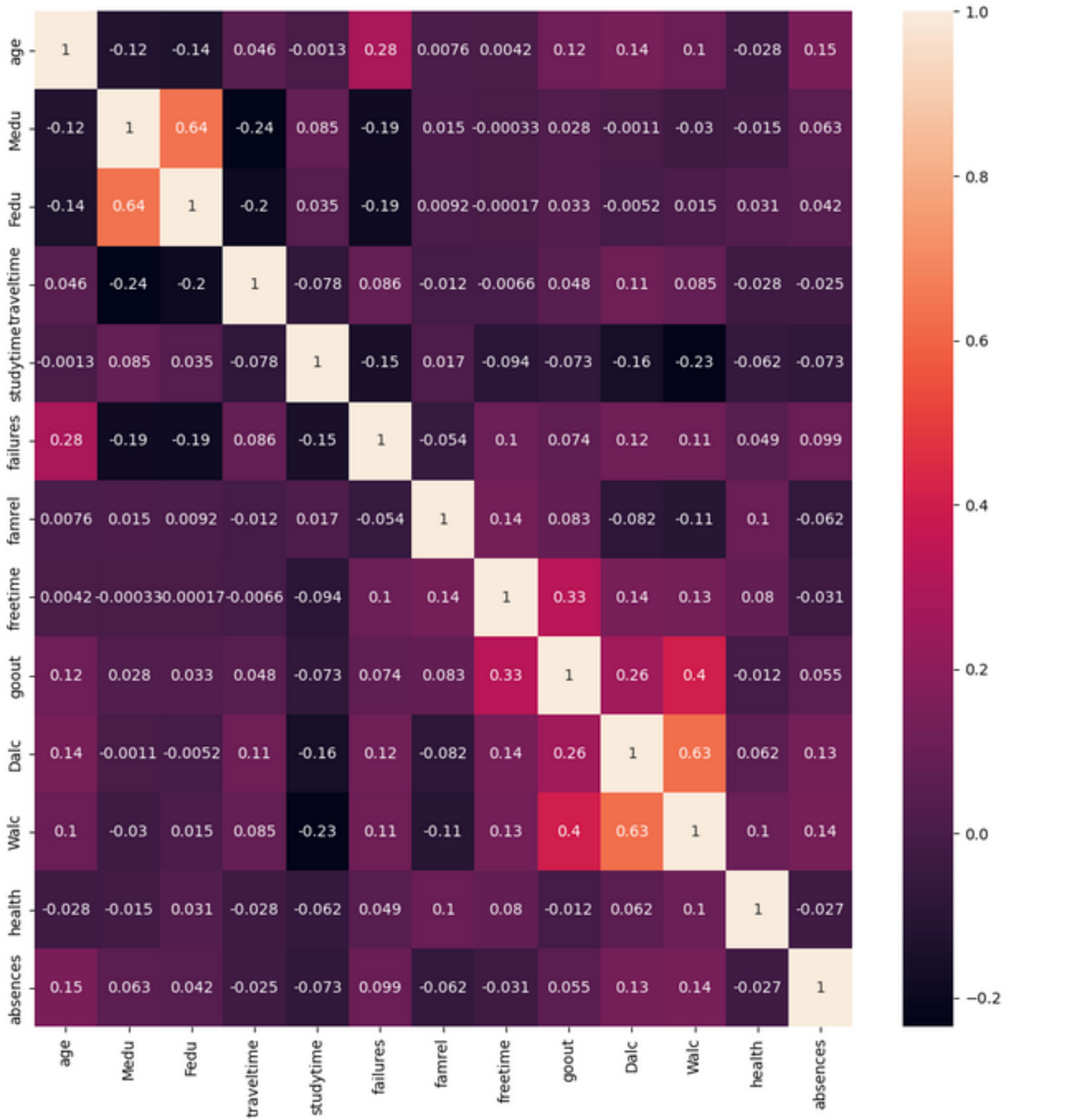
Moreover, more-traveled students receive worse grades.

Studytime, Medu, and Fedu are three factors that are positively connected with the target variables. It goes without saying that your study performance will improve with increased study time. It's interesting to note that parents' educational attainment and their children's performance are highly correlated.

You can examine correlation between the variables as well as the target variable and other factors on the heatmap. For example, there is a substantial association between father's and mother's education, the amount of time the student goes out and the alcohol intake, number of failures and age of the student, etc.

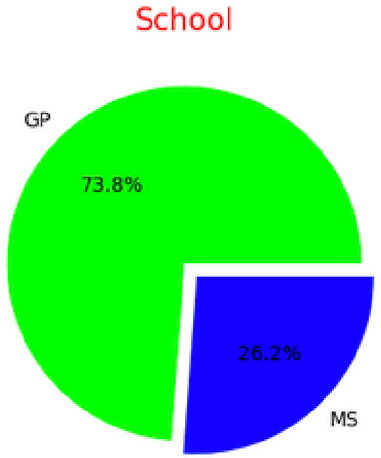
```
plt.figure(figsize=(12,12))
sns.heatmap(data.drop('total grade',axis=1).corr(),annot=True)
plt.show()
```

```
/var/folders/m_/nmwrqc3j5f9dbpczh0mf3wz80000gn/T/ipykernel_85569/632302315.py:2: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid column
s or specify the value of numeric_only to silence this warning.
  sns.heatmap(data.drop('total grade',axis=1).corr(),annot=True)
```



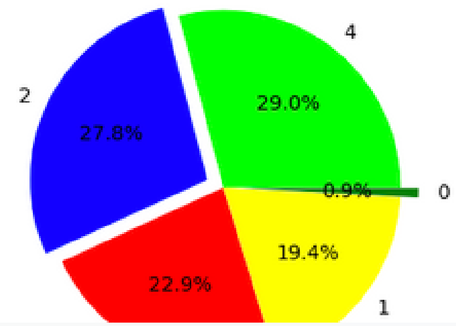
Jnivariate Analysis

```
olors = ['lime','blue']
xplode = [0,0.1,]
lt.figure(figsize = (4,4))
lt.pie(data['school'].value_counts().values, explode = explode, labels = data['school'].value_counts().index, color
lt.title('School',color='Red',fontsize=15)
lt.show()
```



We can see 74% of the students belong to GP' - Gabriel Pereira school and rest 27% belong to MS' - Mousinho da Silveira school

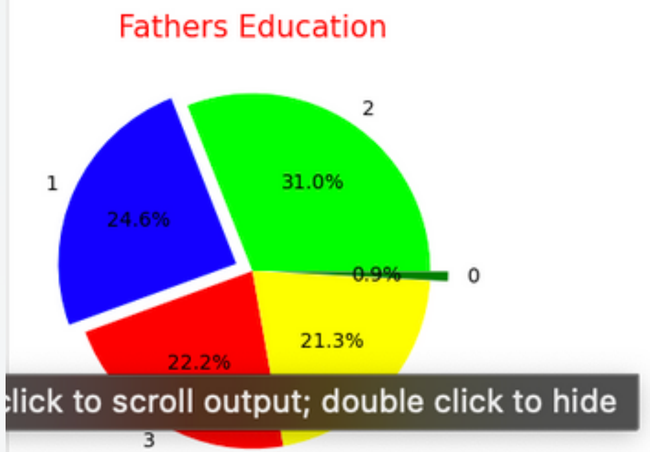
```
olors = ['lime','blue','red','yellow','green']
xplode = [0,0.1,0,0,0.1]
lt.figure(figsize = (4,4))
lt.pie(data['Medu'].value_counts().values, explode = explode, labels = data['Medu'].value_counts().index, colors =
lt.title('Mothers Education',color='Red',fontsize=15)
lt.show()

Mother's education (numeric: 0 - none, 1 - primary education (4th grade),
2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)
```



We can see most of the students mothers have received higher education 29% of the lot

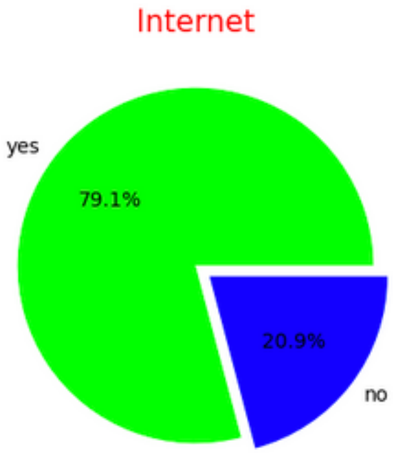
We can see most of the students Fathers have received only 5th -9th Grade Education

```
'red','yellow','green']
]
4))
ue_counts().values, explode = explode, labels = data['Fedu'].value_counts().index, colors = colors, autopct ='%1.1f%%')
tion',color='Red',fontsize=15)


eric: 0 - none, 1 - primary education (4th grade),
 3 â€“ secondary education or 4 â€“ higher education)
```



Most of the Students has got internet connection at their place it sums to 79%

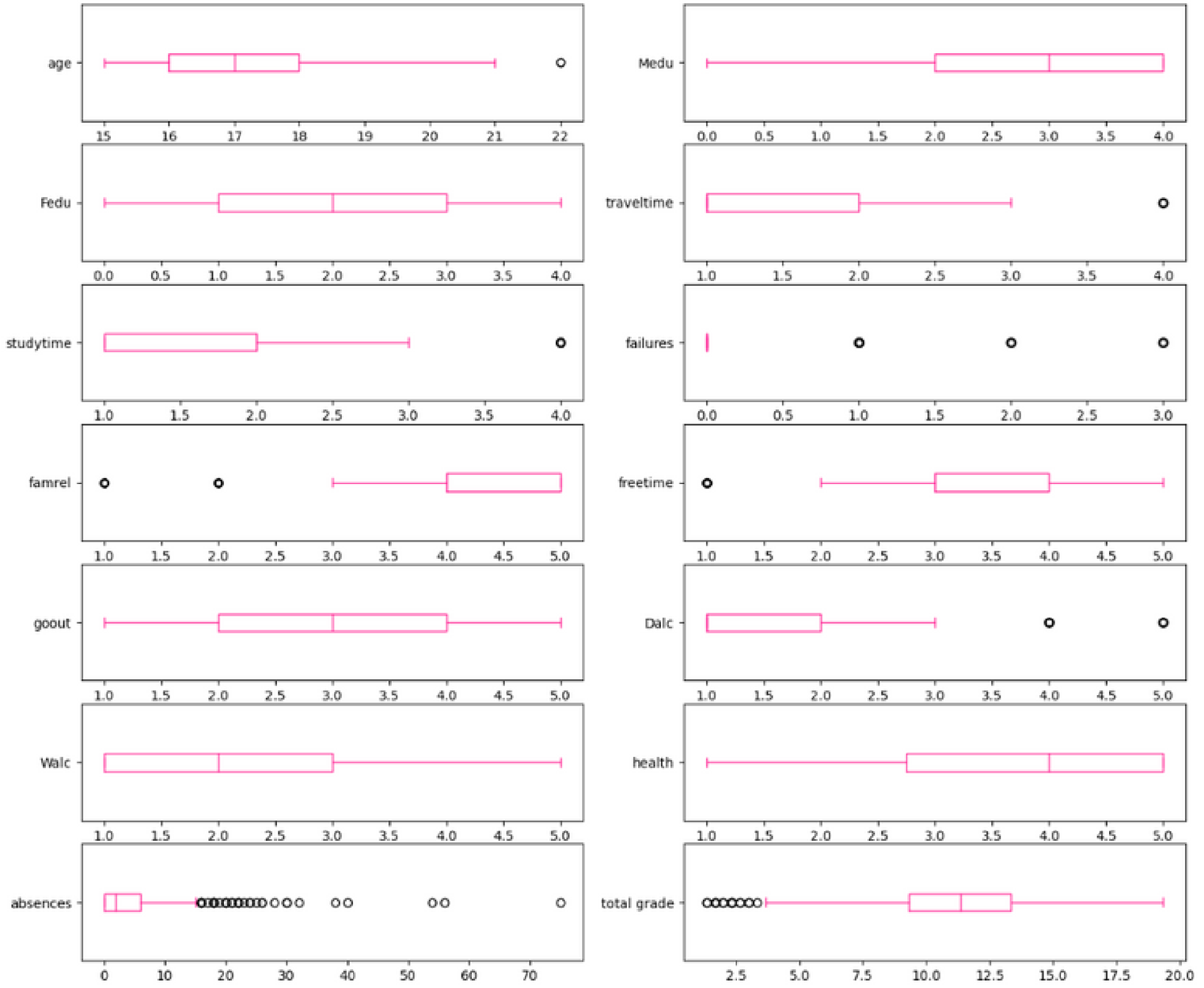
```
colors = ['lime','blue']
explode = [0,0.1]
plt.figure(figsize = (4,4))
plt.pie(data['internet'].value_counts().values, explode = explode, labels = data['internet'].value_counts().index, colo
plt.title('Internet',color='Red',fontsize=15)
plt.show()
```



We can see there are not many outliers, I have used Boxplot to identify the outliers and skewness in the variables.

Outliers in absences, total grade free time and failures columns.

```
Group_Box_Plot=data.plot(kind='box', subplots=True, layout=(8,2),
sharex=False,sharey=False,figsize=(15, 15),
color='deeppink',vert=False);
```



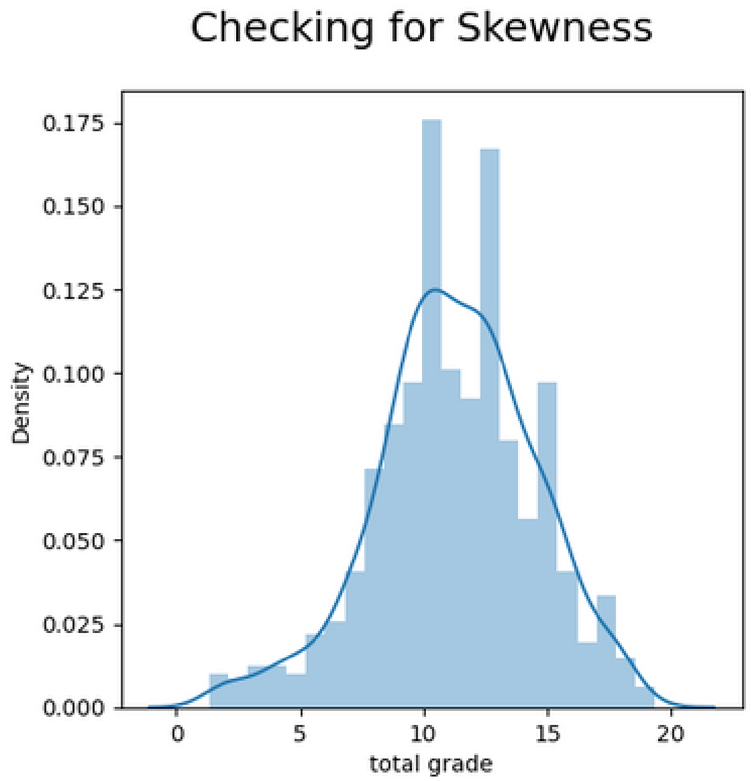
Checking the Skewness in the target variable:

Skewness - ' -0.2929 ' .
More weight to the right tail of distribution.

```
print( 'Skewness for data : ', data['total grade'].skew())
plt.figure(figsize=(5,5))
sns.distplot(data['total grade'])

plt.suptitle('Checking for Skewness', fontsize = 18)
plt.show()

#skewness < 0 : more weight in the right tail of the distribution
```

```
Skewness for data :  -0.2929853185750368
```

```
/var/folders/m_/nmwrqc3j5f9dbpczh0mf3wz80000gn/T/ipykernel_85569/1793286044.py:3: UserWarnin

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['total grade'])
```



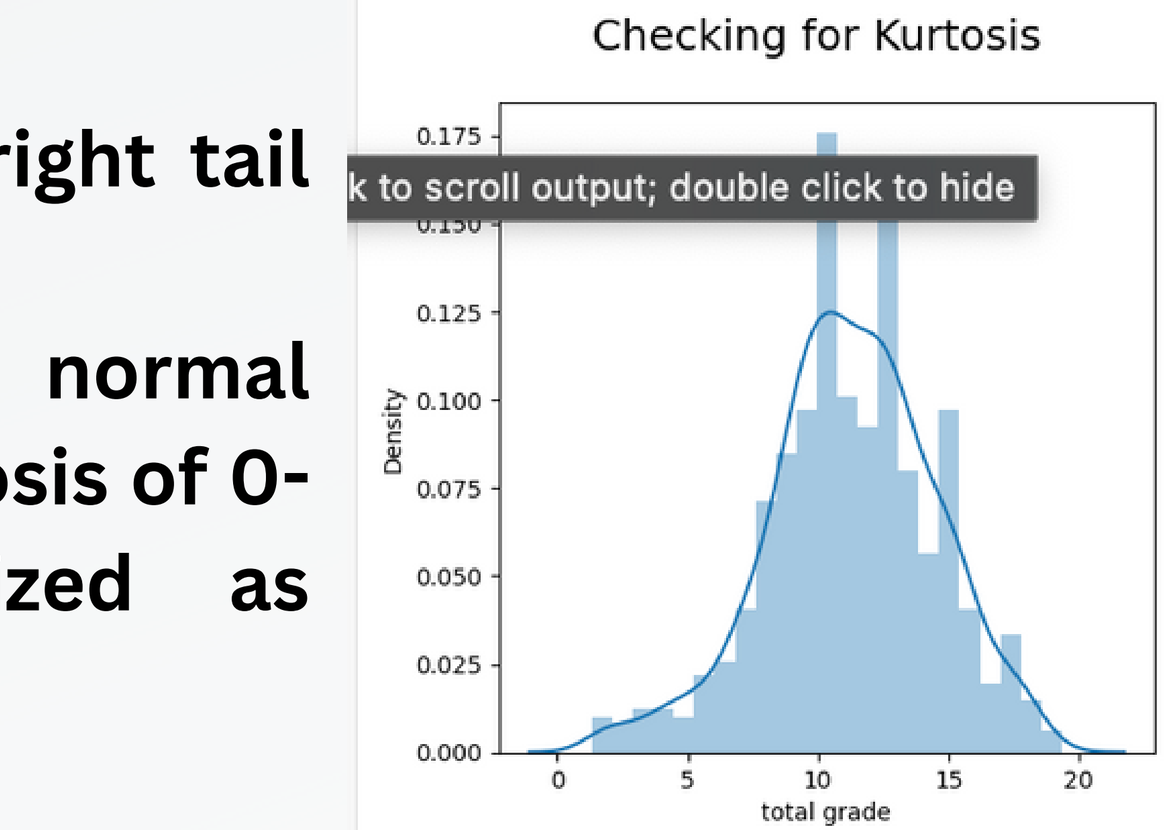
Checking the Kurtosis in the target variable: 'Total Grade'

Kurtosis - ' 0.232 ' .
More weight to the right tail of distribution.
A standard normal distribution has kurtosis of 0-3 and is recognized as mesokurtic.

```
print( 'Kurtosis for data : ', data['total grade'].kurt())
plt.figure(figsize=(5,5))
sns.distplot(data['total grade'])

plt.suptitle('Checking for Kurtosis', fontsize = 18)
plt.show()
```

```
Kurtosis for data :  0.23163700722991942
```

```
/var/folders/m_/nmwrqc3j5f9dbpczh0mf3wz80000gn/T/ipykernel_85569/2086897471.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['total grade'])
```



School - We can see GP school students scoring more compared to the MS school students.

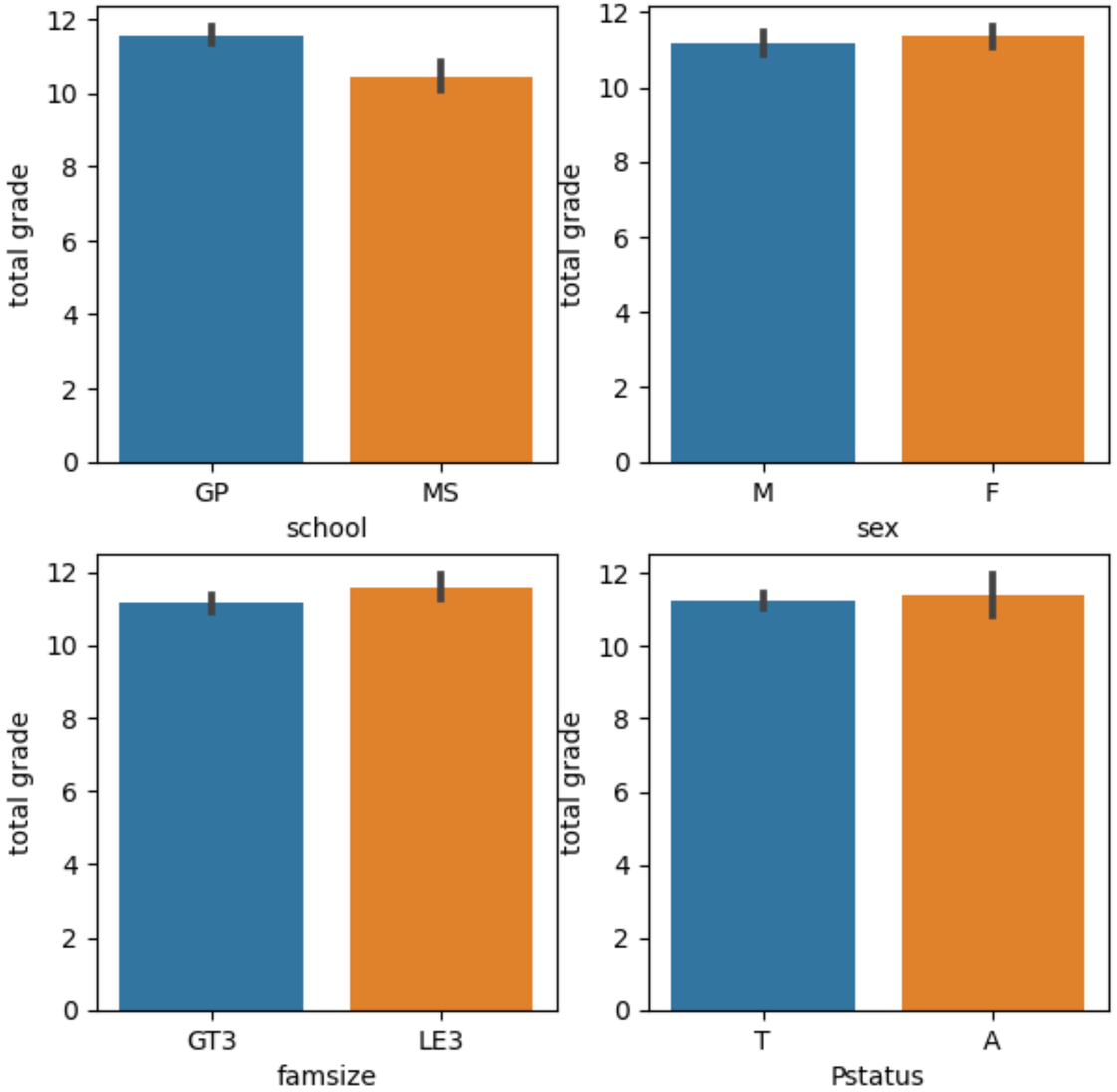
Sex - Margin is so close but we can see female students scoring more compared to male students.

Family size - LE3' - less or equal to 3 has scored more than the 'GT3' - greater than 3 students.

PStatus -
parent's cohabitation status (binary: 'T' - living together or 'A' - apart
Seems almost same

```
#Visualize categorical variables with numerical variables and give conclusions
fig, axarr  = plt.subplots(2,2,figsize=(7,7))
sns.barplot(x='school', y='total grade', data=data, order=['GP','MS'], ax=axarr[0,0])
sns.barplot(x='sex', y='total grade', data=data, order=['M','F'], ax=axarr[0,1])
sns.barplot(x='famsize', y='total grade', data=data, order=['GT3','LE3'], ax=axarr[1,0])
sns.barplot(x='Pstatus', y='total grade', data=data, order=['T','A'], ax=axarr[1,1])
```

```
<AxesSubplot: xlabel='Pstatus', ylabel='total grade'>
```



Reason - Reputed school students have scored more compared to the schools which are closer to home and have course preference.

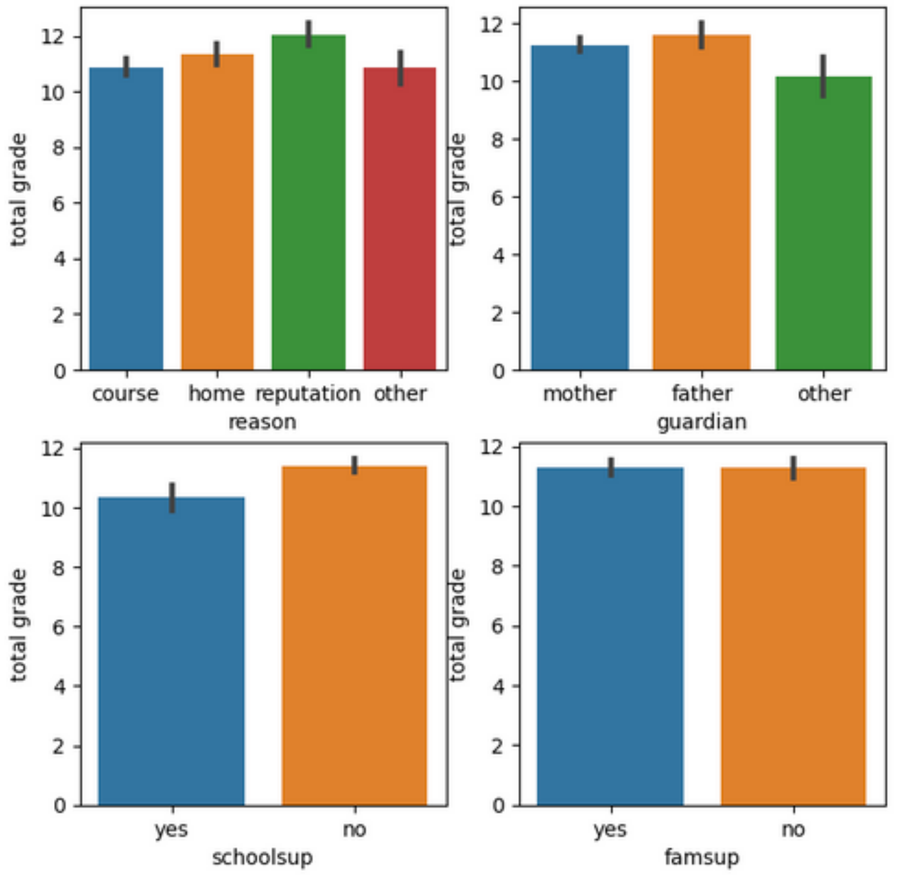
Gaurdian - Students who has got parents have scored more than others, we can see parents having an effect on a child's grade.

Schoolsupport - Students have score more marks without school support.

Family support - No much difference in the grade with this variable.

```
fig, axarr  = plt.subplots(2,2,figsize=(7,7))
sns.barplot(x='reason', y='total grade', data=data, order=['course','home','reputation','other'], ax=axarr[0,0])
sns.barplot(x='guardian', y='total grade', data=data, order=['mother','father','other'], ax=axarr[0,1])
sns.barplot(x='schoolsup', y='total grade', data=data, order=['yes','no'], ax=axarr[1,0])
sns.barplot(x='famsup', y='total grade', data=data, order=['yes','no'], ax=axarr[1,1])
```

```
<AxesSubplot: xlabel='famsup', ylabel='total grade'>
```



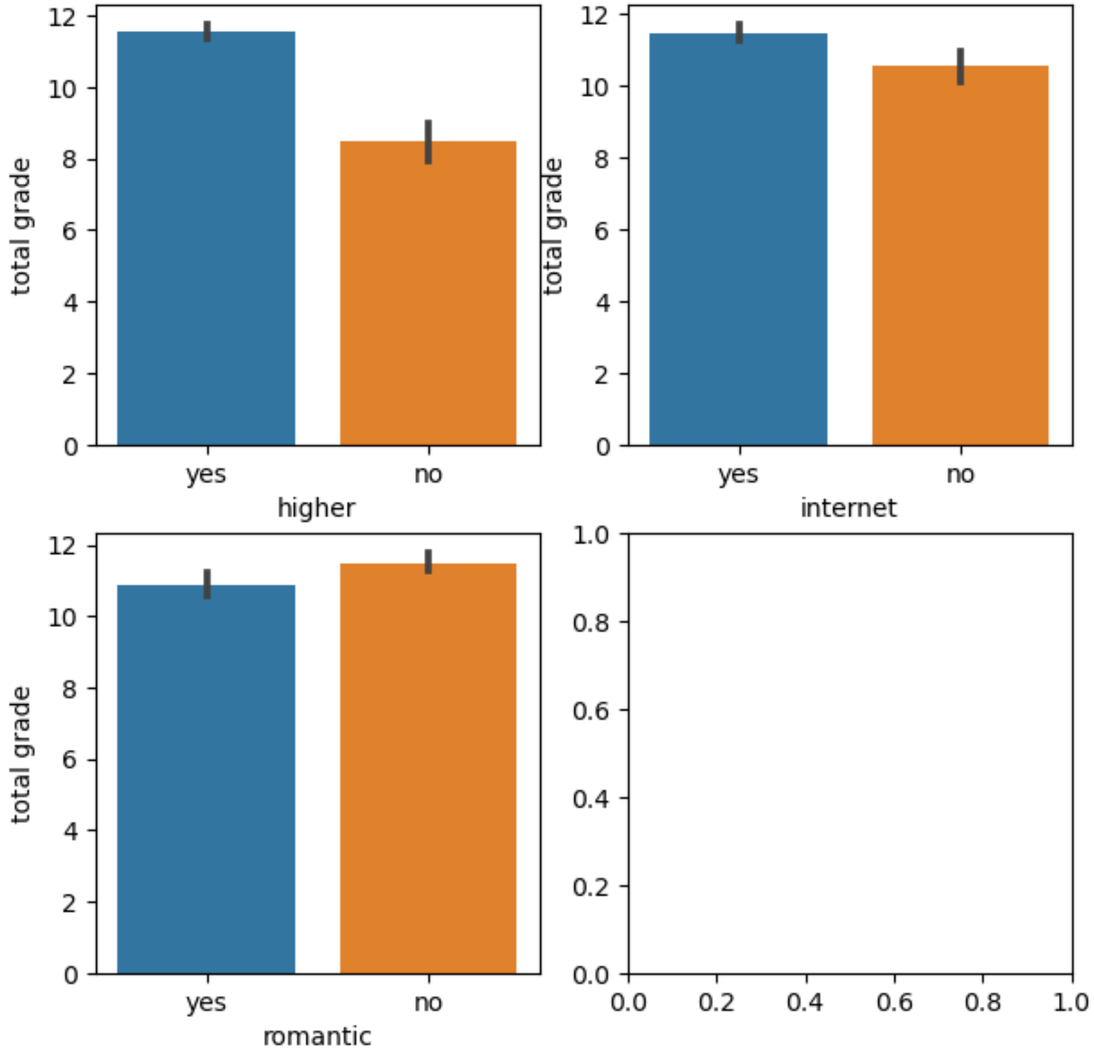
Higher - Students who have an opinion to take higher education have scored more than those who don't want to study higher.

Internet - Students with an internet connection have scored more than students without an internet connection.

Romantic - No relationships have made students score higher than students in relationships

```
fig, axarr  = plt.subplots(2,2,figsize=(7,7))
sns.barplot(x='higher', y='total grade', data=data, order=['yes','no'], ax=axarr[0,0])
sns.barplot(x='internet', y='total grade', data=data, order=['yes','no'], ax=axarr[0,1])
sns.barplot(x='romantic', y='total grade', data=data, order=['yes','no'], ax=axarr[1,0])
```

```
<AxesSubplot: xlabel='romantic', ylabel='total grade'>
```

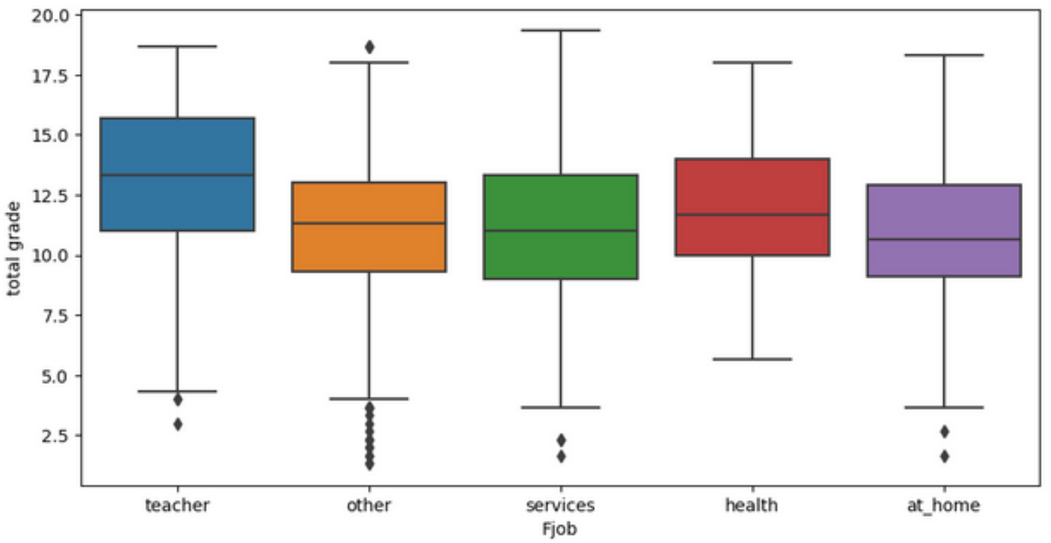


The boxplot makes it possible to observe the median value and the low and high quartiles of the data. The remaining distribution is depicted by the whiskers. Points outside of the whiskers signify outliers.

One of the fascinating conclusions you can get from the graphs above is that a student is more likely to receive a high final grade if their parent or mother is a teacher.

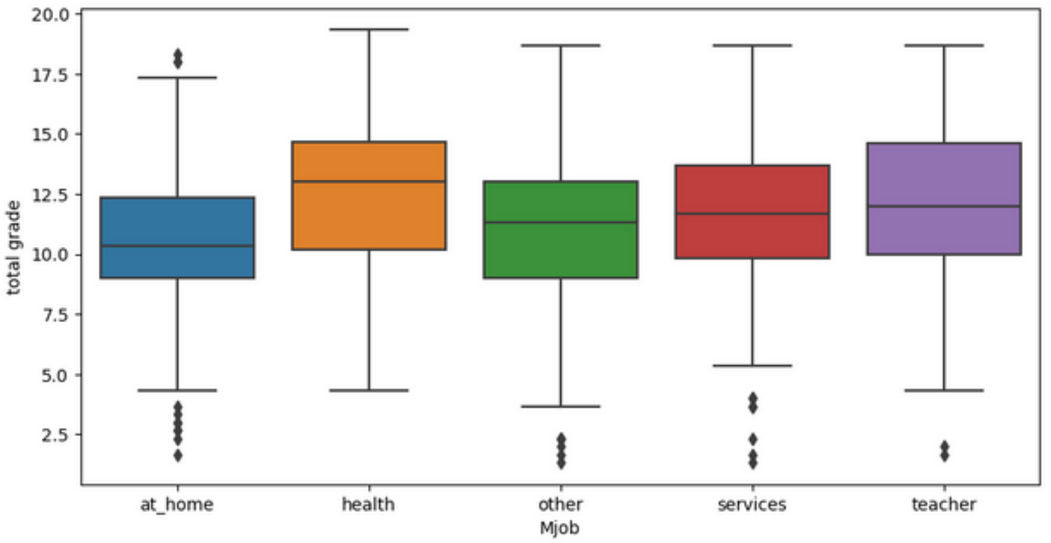
```
#Variables based on fathers and Mother job:

plt.figure(figsize=(10,5))
sns.boxplot(x=data['Fjob'],y=data['total grade'])
```

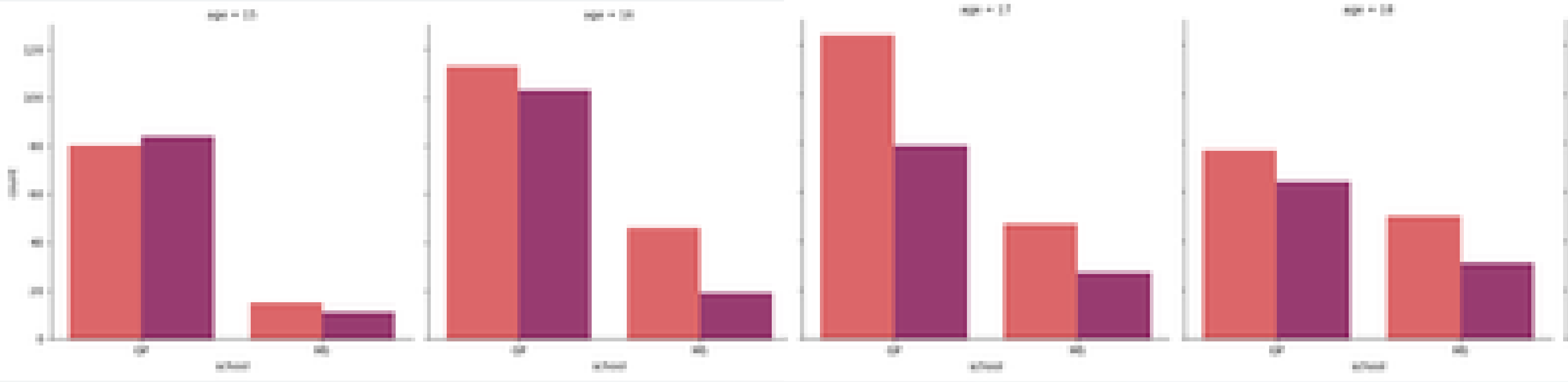
```
<AxesSubplot: xlabel='Fjob', ylabel='total grade'>
```

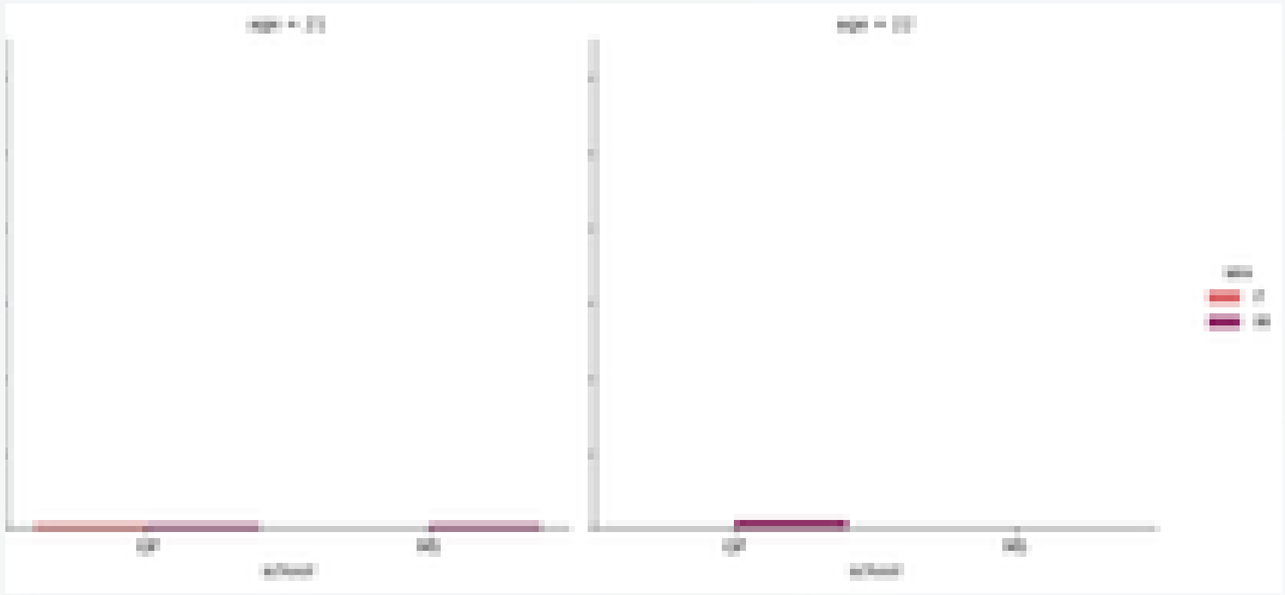



```
plt.figure(figsize=(10,5))
sns.boxplot(x=data['Mjob'],y=data['total grade'])
```

```
<AxesSubplot: xlabel='Mjob', ylabel='total grade'>
```







Used Cat- plot for determining the age, sex and school variables.

We see that most students belong to the gender female with the age bracket = 17 years

Plotted Histogram to study the range of values for various columns under particular circumstances

#age - student's age (numeric: from 15 to 22); Most of the students belong to the age bracket 15-16 years

#Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ #secondary education or 4 â€“ higher education) ; Mothers highly educated.

#Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ #secondary education or 4 â€“ higher education) : Fathers are just educated 5th -9th grade

#traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour): Most of the students just stay 15 mins away from the school.

#studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
Most of the students study for just 2-5 hours.

#famrel - the quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
Family relations of the just are close to excellent.

#freetime - free time after school (numeric: from 1 - very low to 5 - very high).
The Freetime of the students is moderate.

#goout - going out with friends (numeric: from 1 - very low to 5 - very high).
Students moderately go out.

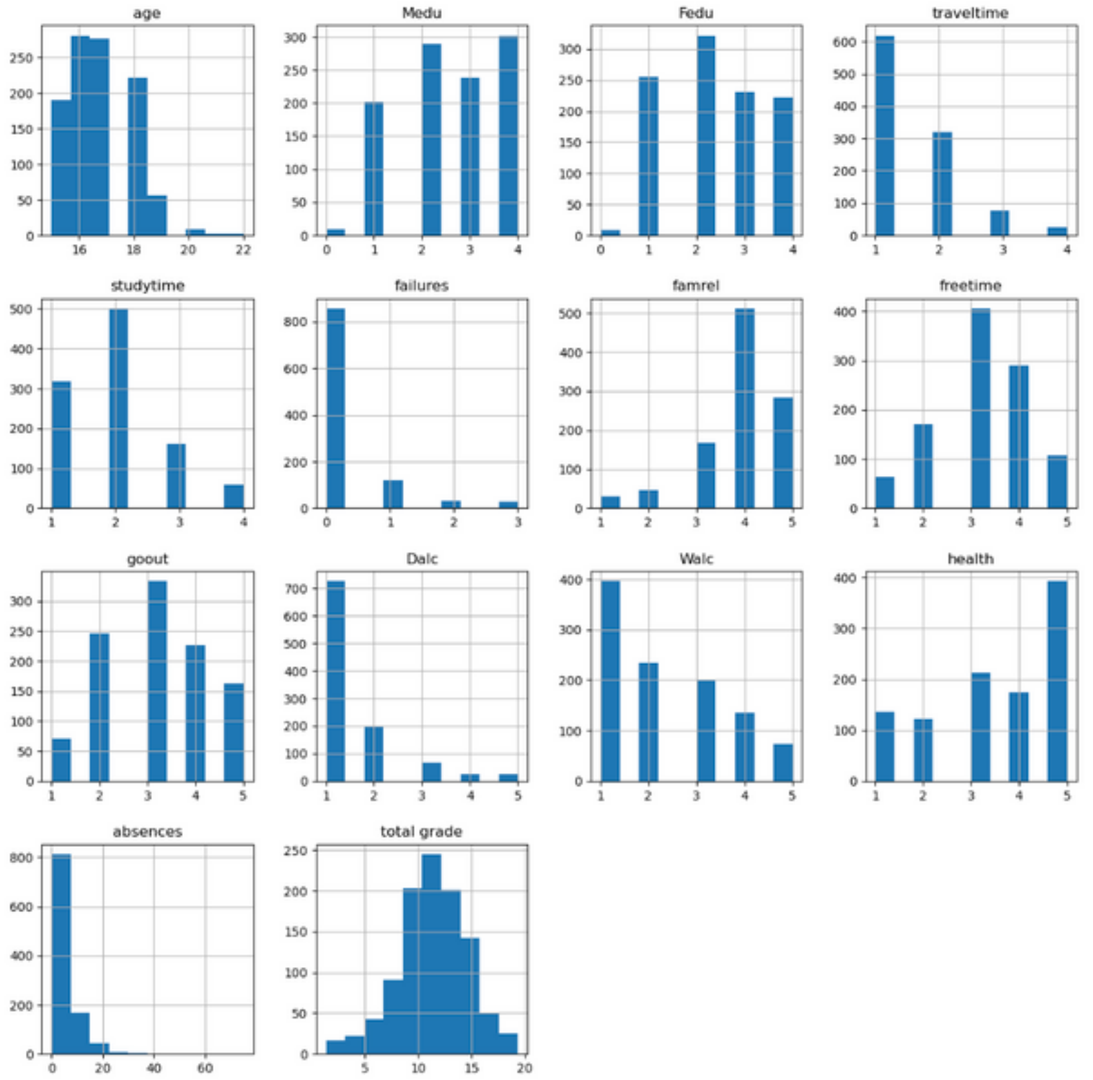
#Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high).
Most of the students won't consume alcohol.

#Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high).
Weekend alcohol consumption of the students is also low

#health - current health status (numeric: from 1 - very bad to 5 - very good).
Most of the students health condition is good

```
data.hist(figsize=(15,15))
```

```
array([[<AxesSubplot: title={'center': 'age'}>,
        <AxesSubplot: title={'center': 'Medu'}>,
        <AxesSubplot: title={'center': 'Fedu'}>,
        <AxesSubplot: title={'center': 'traveltime'}>],
       [<AxesSubplot: title={'center': 'studytime'}>,
        <AxesSubplot: title={'center': 'failures'}>,
        <AxesSubplot: title={'center': 'famrel'}>,
        <AxesSubplot: title={'center': 'freetime'}>],
       [<AxesSubplot: title={'center': 'goout'}>,
        <AxesSubplot: title={'center': 'Dalc'}>,
        <AxesSubplot: title={'center': 'Walc'}>,
        <AxesSubplot: title={'center': 'health'}>],
       [<AxesSubplot: title={'center': 'absences'}>,
        <AxesSubplot: title={'center': 'total grade'}>, <AxesSubplot: >,
        <AxesSubplot: >]], dtype=object)
```



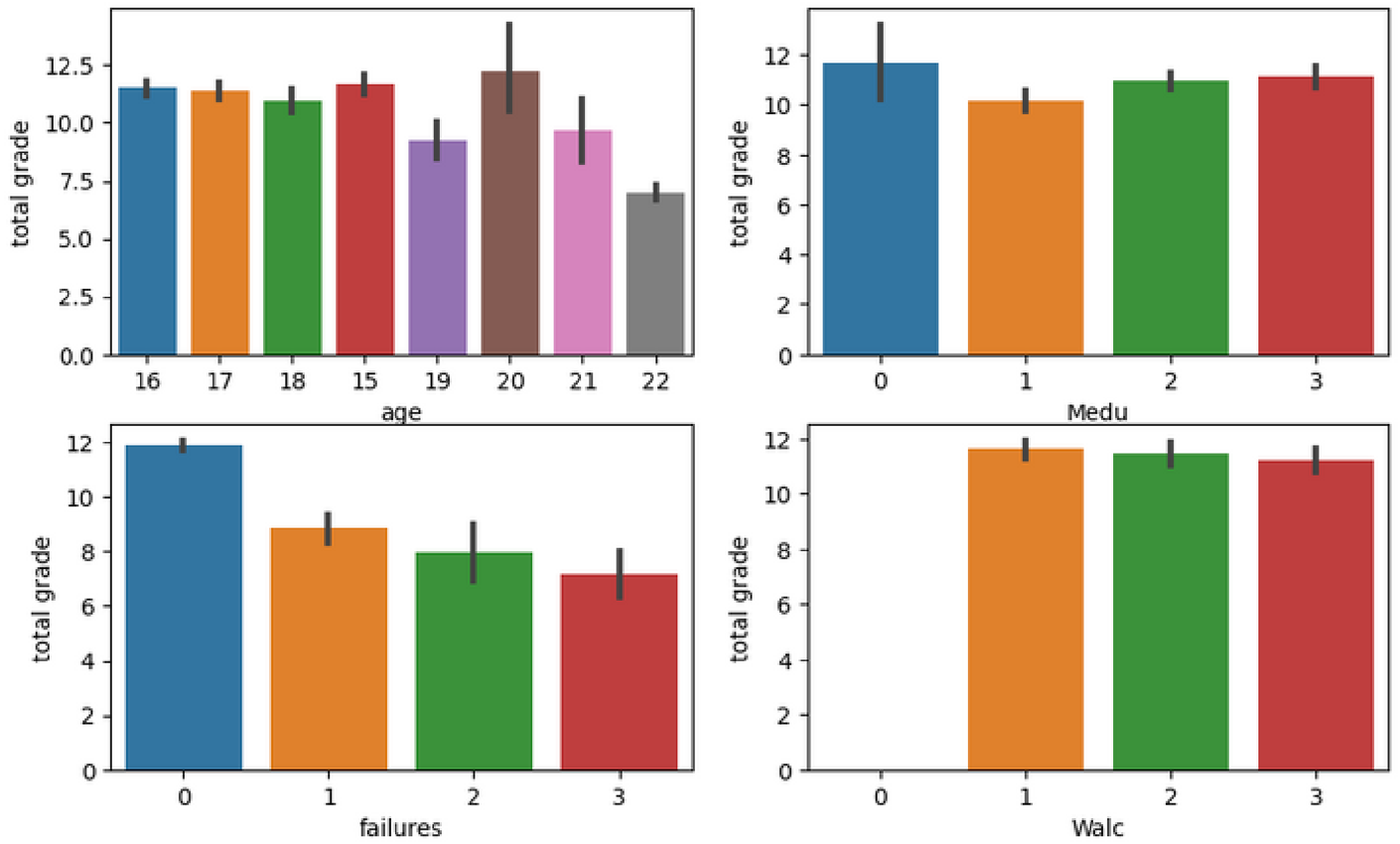
Age - Students of the age group 20 have scored well compared to the other age groups.

Failures - Obviously students with less failure rate have scored well.

Mothers' Education - Students with no mothers' education scored equally well and more than students with mothers' educated.

```
fig, axarr  = plt.subplots(2,2,figsize=(10,6))
sns.barplot(x='age', y='total grade', data=data, order=[16,17,18,15,19,20,21,22], ax=axarr[0,0])
sns.barplot(x='failures', y='total grade', data=data, order=[0,1,2,3], ax=axarr[1,0])
sns.barplot(x='Medu', y='total grade', data=data, order=[0,1,2,3], ax=axarr[0,1])
sns.barplot(x='Walc', y='total grade', data=data, order=[0,1,2,3], ax=axarr[1,1])
```

```
<AxesSubplot: xlabel='Walc', ylabel='total grade'>
```



Go out - People who moderately go outscored more as compared to others.

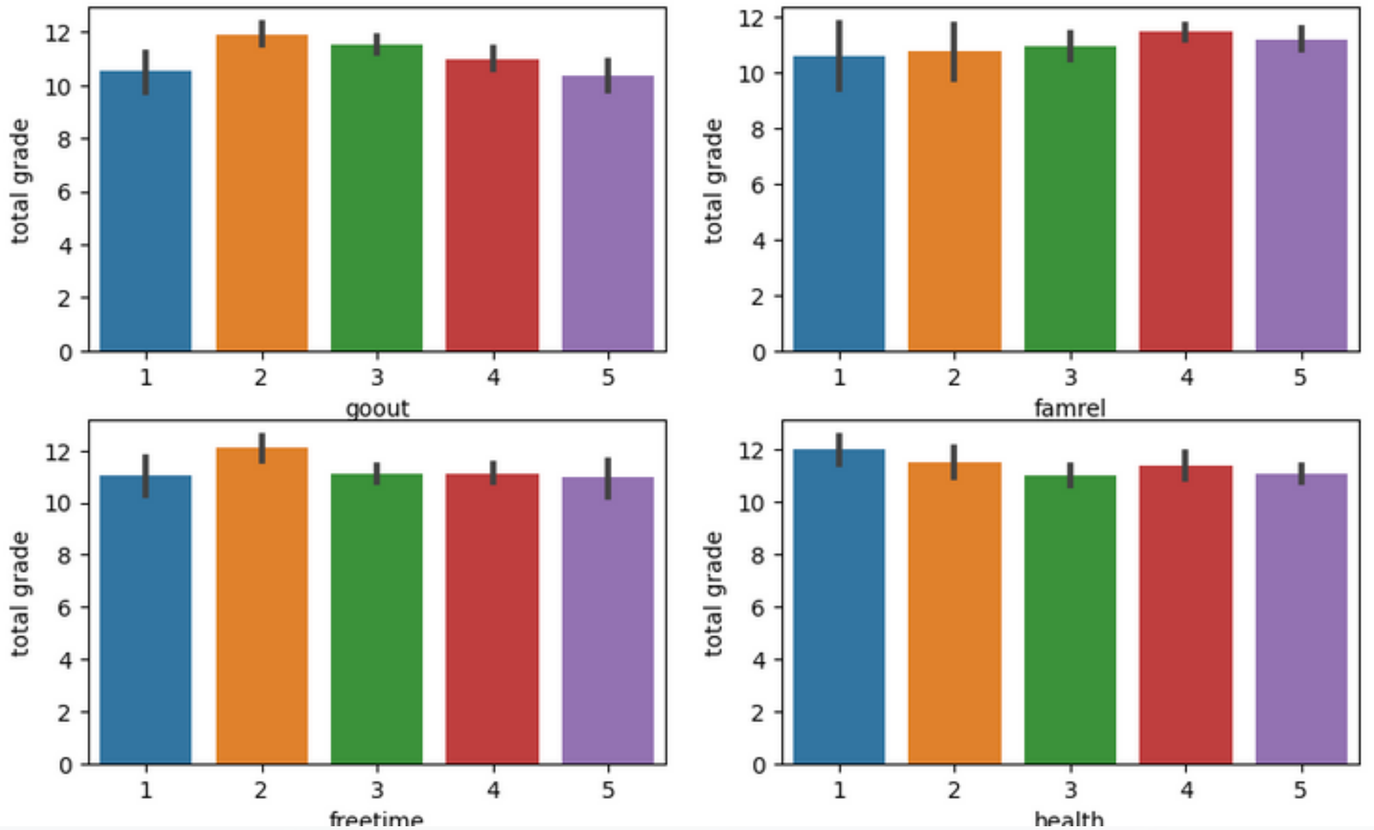
Famrelations - Students whose family relations are close to high scored more as compared to the others.

Freetime - Students who have got almost more free time scored higher.

Health - Students whose health conditions are poor scored higher as compared to others.

```
fig, axarr  = plt.subplots(2,2,figsize=(10,6))
sns.barplot(x='goout', y='total grade', data=data, order=[1,2,3,4,5], ax=axarr[0,0])
sns.barplot(x='freetime', y='total grade', data=data, order=[1,2,3,4,5], ax=axarr[1,0])
sns.barplot(x='famrel', y='total grade', data=data, order=[1,2,3,4,5], ax=axarr[0,1])
sns.barplot(x='health', y='total grade', data=data, order=[1,2,3,4,5], ax=axarr[1,1])
```

```
<AxesSubplot: xlabel='health', ylabel='total grade'>
```



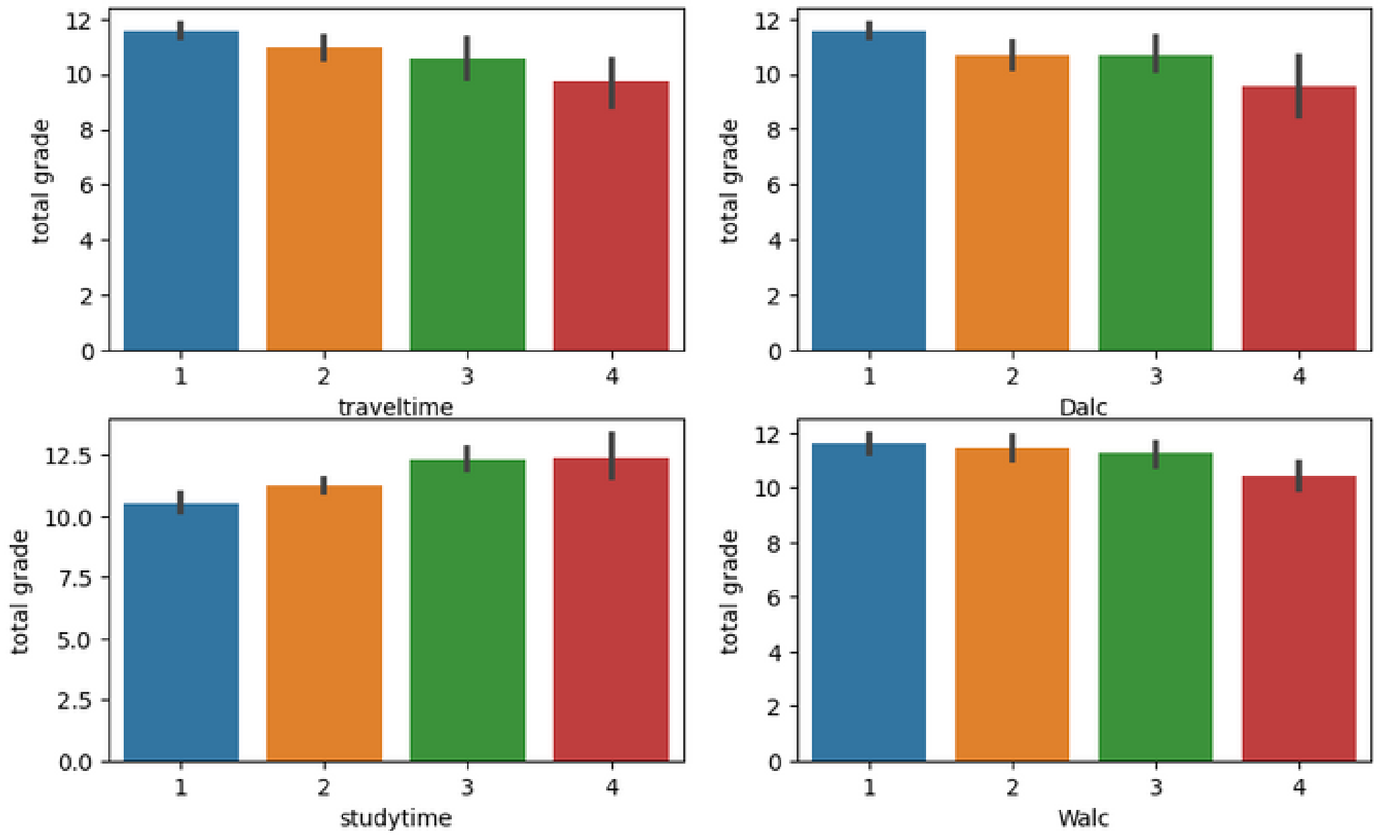
Travel Time - Students whose travel time is less scored more as they could get some time to rest and study compared to others.

Study time - Students who study time is more scored more as compared to others

Students who has consumed less alcohol has scored more compared to others.

```
fig, axarr  = plt.subplots(2,2,figsize=(10,6))
sns.barplot(x='traveltime', y='total grade', data=data, order=[1,2,3,4], ax=axarr[0,0])
sns.barplot(x='studytime', y='total grade', data=data, order=[1,2,3,4], ax=axarr[1,0])
sns.barplot(x='Dalc', y='total grade', data=data, order=[1,2,3,4], ax=axarr[0,1])
sns.barplot(x='Walc', y='total grade', data=data, order=[1,2,3,4], ax=axarr[1,1])
```

```
<AxesSubplot: xlabel='Walc', ylabel='total grade'>
```



Have Label-encoded all object columns and used StandardScaler for all numeric columns

```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
object_encode=objectcols.apply(le.fit_transform)
```

```
object_encode=pd.DataFrame(object_encode)
```

```
object_encode.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	nursery	higher	internet	romantic	famrel	freetime	goout	Dalc	Walc	health
0	0	0	3	1	0	0	4	4	0	4	...	1	1	0	0	3	2	3	0	0	2
1	0	0	2	1	0	1	1	1	0	2	...	0	1	1	0	4	2	2	0	0	2
2	0	0	0	1	1	1	1	1	0	2	...	1	1	1	0	3	2	1	1	2	2
3	0	0	0	1	0	1	4	2	1	3	...	1	1	1	1	2	1	1	0	0	4
4	0	0	1	1	0	1	3	3	2	2	...	1	1	0	0	3	2	1	0	1	4

5 rows × 29 columns

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

```
scaler=StandardScaler()
```

```
numeric_standard=scaler.fit_transform(numericcols.drop('total grade',axis=1))
```

```
numeric_standard=pd.DataFrame(numeric_standard,columns=['absences'])
```

```
numeric_standard.head()
```

	absences
0	0.249173
1	-0.072740
2	0.892999
3	-0.394653
4	-0.072740

Separated Features and Labels to 'X' and y

As the target variable data is almost symmetric, so I have not used any 'Logorithmic' or 'Square' transformations .

```
#Preparing Training, Testing, And Validating Dataset
from sklearn.model_selection import train_test_split
X_train_full, X_test, y_train_full, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train_full, y_train_full, test_size=0.2, random_state=42)
```

Separated Features and Labels to 'X' and y

Have split the data to Train and Test data with 80-20 percent ratio.

Separating Features and Labels

```
#Separating Features and Labels
X = data_new
y = data['total grade']
```

```
print(X.shape)
print(y.shape)
```

```
(1040, 32)
(1040,)
```

```
y.dtype
```

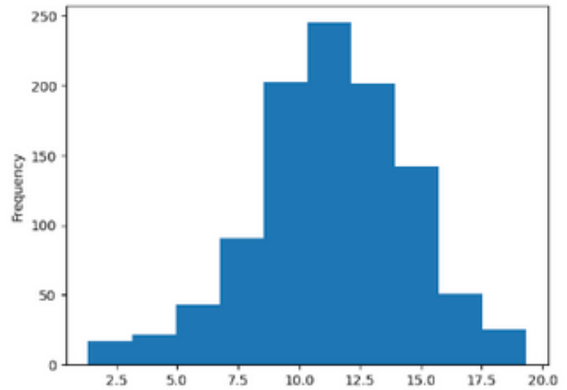
```
dtype('float64')
```

```
y.head()
```

```
0     5.666667
1     5.333333
2     8.333333
3    14.666667
4     8.666667
Name: total grade, dtype: float64
```

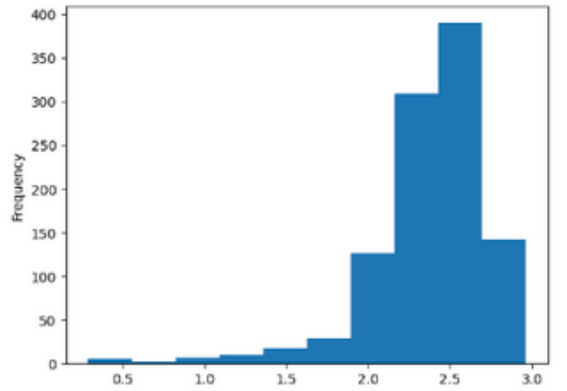
```
y.plot(kind='hist')
```

```
<AxesSubplot: ylabel='Frequency'>
```




```
np.log(y).plot(kind='hist')
```

```
<AxesSubplot: ylabel='Frequency'>
```



MODELING AND EVALUATION

THIS IS A STEP WHERE WE TESTED OUR DATA TRAIN TO MACHINE LEARNING MODEL AND EVALUATED IT. WE TESTED IT TO 5 DIFFERENT MODELS:

LOGISTIC REGRESSION

DECISION TREE REGRESSOR

RANDOM FOREST REGRESSOR

LASSO

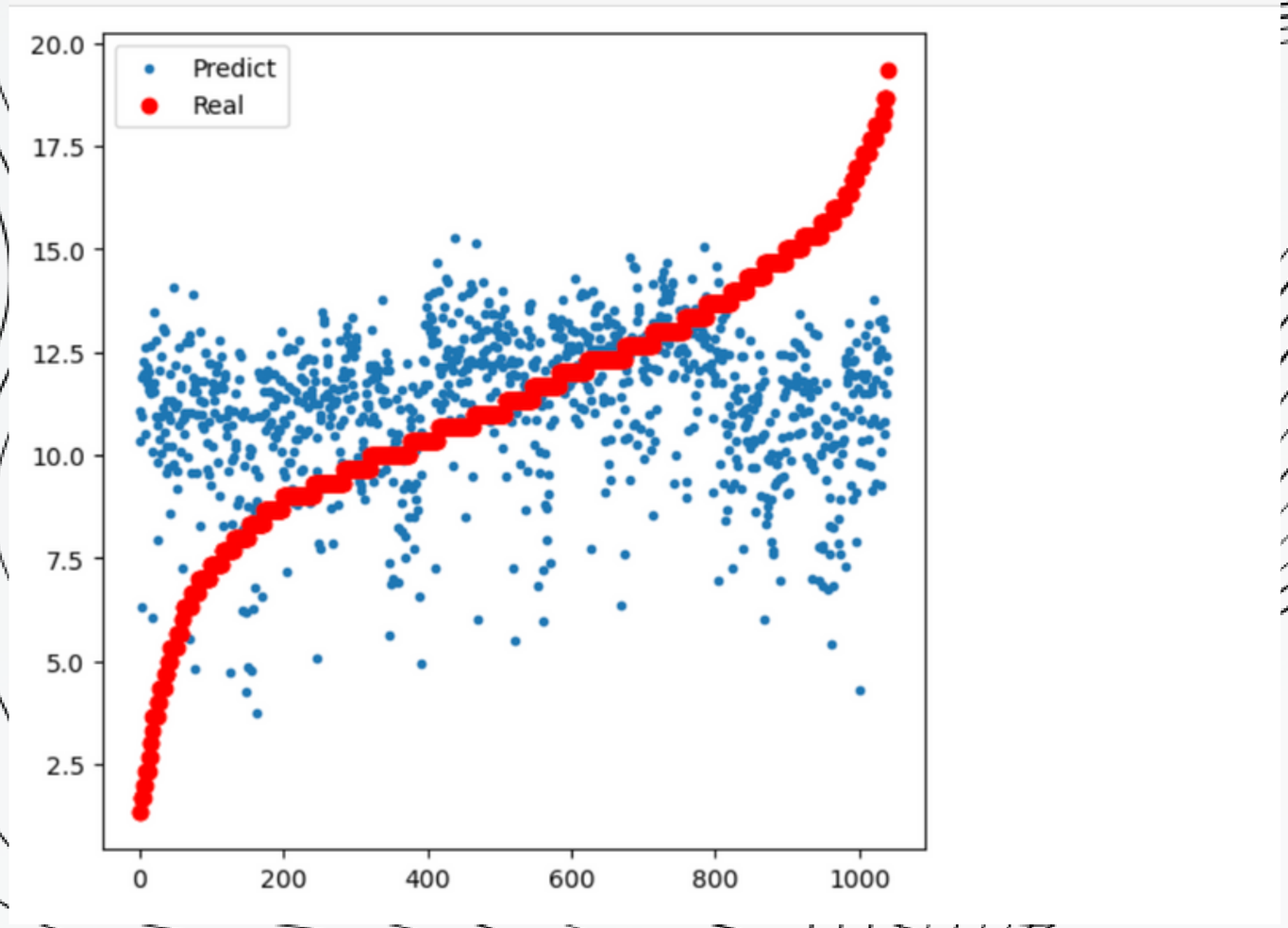
RIDGE

THE GOAL OF ITS ACTION IS THAT AT THE CONCLUSION OF THIS STAGE, WE WILL NOT ONLY BE ABLE TO IDENTIFY THE BEST MODEL BUT ALSO THE BEST PREPROCESSING METHOD FOR THE DATASET. WE DID IT TO OBTAIN A BETTER OUTCOME BECAUSE, AS WE ALL KNOW, DATA SCIENCE IS AN EXPERIMENTAL FIELD. WHAT WE HAVE ACCOMPLISHED SO FAR: SPLITTING AND PREPROCESSING DATA (DATA TRAIN AND DATA TEST) TESTING FEATURE ENGINEERING MODELS TUNING HYPERPARAMETERS AND CHOOSING FEATURES MODEL OPTIONAL MOST INFLUENTIAL/INFLUENCE ON MODEL OUTPUT IS EVALUATION.

From this evaluation, the Machine Learning model using the Ridge method has the smallest RMSE.

Machine Learning Model	Accuracy	Mean Squared Error	Mean Absolute Error	Root mean Squared Error
Linear Regressor	0.300	5.585	1.856	2.363
Decision Tree Regressor	0.666	9.550	2.381	3.090
Random Forest Regressor	0.911	5.040	1.710	5.040
Lasso	0.040	7.914	2.300	2.810
Ridge	0.300	5.578	1.854	2.362

```
#Visualize The Machine Learning Model

fig = plt.figure(figsize=(6,6))
data = data.sort_values(by=['total grade'])
X = data_new
y = data['total grade']
plt.scatter(range(X.shape[0]), model_ridge.predict(X), marker='.', label='Predict')
plt.scatter(range(X.shape[0]), y, color='red', label='Real')
plt.legend(loc='best', prop={'size': 10})
plt.show()
```



CONCLUSION

BY SEEING THE MODEL ACCURACY AND OTHER METRICS OF THE MODEL, IT IS CLEARLY OBSERVED THAT THE RIDGE MODEL HAS PERFORMED WELL THAN OTHER MODELS.