EDA_Student_Performance_Indicator

February 17, 2024

0.1 EDA Student Performance Indicator

0.1.1 1) Problem statement

- This project understands how the student's performance (test scores) is affected by other variables such as Gender, Ethnicity, Parental level of education, Lunch and Test preparation course.

0.1.2 2) Data Collection

- Dataset Source - https://www.kaggle.com/datasets/spscientist/students-performance-in-exams?datasetId=74977
- The data consists of 8 column and 1000 rows.

0.1.3 3) Dataset Information

- gender : sex of students -> (Male/female)
- race/ethnicity : ethnicity of students -> (Group A, B,C, D,E)
- parental level of education : parents' final education ->(bachelor's degree,some college,master's degree,associate's degree,high school)
- lunch : having lunch before test (standard or free/reduced)
- test preparation course : complete or not complete before test
- math score
- reading score
- writing score

[2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

[3]:
```
# Read the dataset
df = pd.read_csv("stud.csv")
df.head()
```

[3]:
```
  gender race_ethnicity parental_level_of_education         lunch  \
0  female        group B           bachelor's degree      standard
1  female        group C                some college      standard
2  female        group B             master's degree      standard
3    male        group A          associate's degree  free/reduced
4    male        group C                some college      standard

  test_preparation_course  math_score  reading_score  writing_score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[4]:
```
df.shape
```

[4]:
```
(1000, 8)
```

0.1.4 3. Data Checks to perform

- Check Missing values
- Check Duplicates
- Check data type
- Check the number of unique values of each column
- Check statistics of data set
- Check various categories present in the different categorical column

[5]:
```
## check missing values
df.isnull().sum()
```

[5]:
```
gender                         0
race_ethnicity                 0
parental_level_of_education    0
lunch                          0
test_preparation_course        0
math_score                     0
reading_score                  0
writing_score                  0
dtype: int64
```

0.2 Insights or Observation

There are no missing values

[6]:
```
df.isna().sum()
```

[6]:
```
gender                         0
race_ethnicity                 0
parental_level_of_education    0
lunch                          0
test_preparation_course        0
math_score                     0
reading_score                  0
writing_score                  0
dtype: int64
```

[7]:
```
## Check Duplicates
df.duplicated().sum()
```

[7]: 0

0.3 There are no duplicate values in the dataset

0.4 check datatypes

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   race_ethnicity               1000 non-null   object
 2   parental_level_of_education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test_preparation_course      1000 non-null   object
 5   math_score                   1000 non-null   int64
 6   reading_score                1000 non-null   int64
 7   writing_score                1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

[9]:
```
## 3.1 Check the number of unique values of each column
df.nunique()
```

[9]:
```
gender                          2
race_ethnicity                  5
parental_level_of_education     6
lunch                           2
test_preparation_course         2
math_score                     81
reading_score                  72
```

```
writing_score                   77
dtype: int64
```

[10]:
```python
## Check the statistics of dataset
df.describe()
```

[10]:
```
       math_score  reading_score  writing_score
count  1000.00000    1000.000000    1000.000000
mean     66.08900      69.169000      68.054000
std      15.16308      14.600192      15.195657
min       0.00000      17.000000      10.000000
25%      57.00000      59.000000      57.750000
50%      66.00000      70.000000      69.000000
75%      77.00000      79.000000      79.000000
max     100.00000     100.000000     100.000000
```

0.5 Insights or Observation

- From the above description of numerical data,all means are very close to each other- between 66 and 69
- All the standard deviation are also close- between 14.6- 15.19
- While there is a minimum of 0 for maths,other are having 17 and 10 value

[12]:
```python
## explore more info about the data
df.head()
```

[12]:
```
   gender race_ethnicity parental_level_of_education         lunch  \
0  female        group B           bachelor's degree      standard
1  female        group C                some college      standard
2  female        group B             master's degree      standard
3    male        group A          associate's degree  free/reduced
4    male        group C                some college      standard

  test_preparation_course  math_score  reading_score  writing_score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

[20]:
```python
numerical_features=[feature for feature in df.columns if df[feature].dtype !=␣
↪'O']
categorical_features=[feature for feature in df.columns if df[feature].dtype ==␣
↪'O']
```

[21]:
```python
numerical_features
```

[21]: ['math_score', 'reading_score', 'writing_score']

[22]:
```
categorical_features
```

[22]:
```
['gender',
 'race_ethnicity',
 'parental_level_of_education',
 'lunch',
 'test_preparation_course']
```

[23]:
```
df['gender'].value_counts()
```

[23]:
```
female    518
male      482
Name: gender, dtype: int64
```

[24]:
```
df['race_ethnicity'].value_counts()
```

[24]:
```
group C    319
group D    262
group B    190
group E    140
group A     89
Name: race_ethnicity, dtype: int64
```

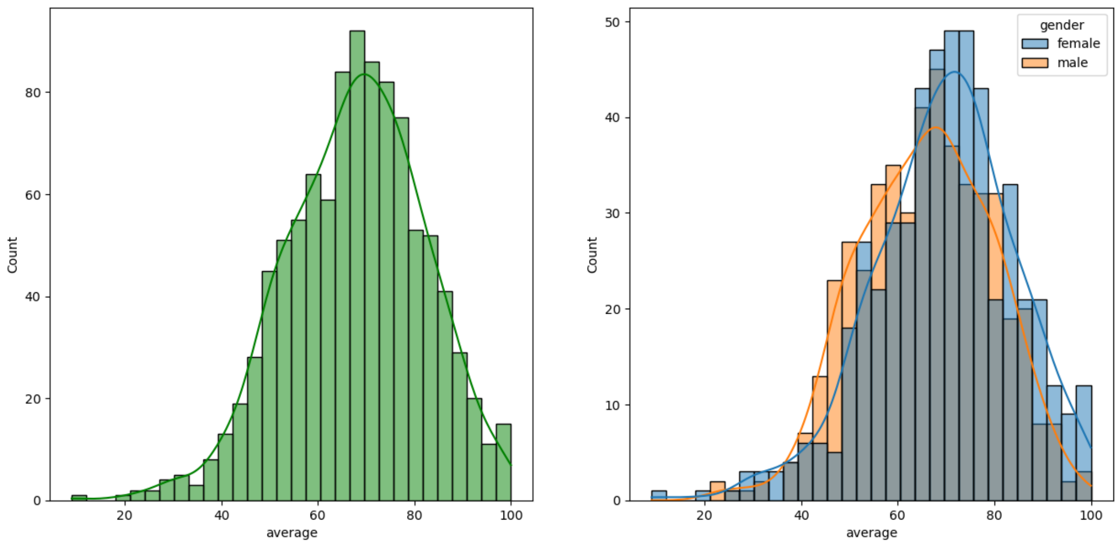
[26]:
```
## Aggregate the total score with mean
df['total_score']=(df['math_score']+df['reading_score']+df['writing_score'])
df['average']=df['total_score']/3
df.head()
```

[26]:
```
   gender race_ethnicity parental_level_of_education         lunch  \
0  female        group B           bachelor's degree      standard
1  female        group C                some college      standard
2  female        group B             master's degree      standard
3    male        group A          associate's degree  free/reduced
4    male        group C                some college      standard

  test_preparation_course  math_score  reading_score  writing_score  \
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75

   total_score    average
0          218  72.666667
1          247  82.333333
```

```
2           278  92.666667
3           148  49.333333
4           229  76.333333
```

[33]:
```
### Explore More Visualisation
fig,axis=plt.subplots(1,2,figsize=(15,7))
plt.subplot(121) # we are ploting in 1st row 2nd column and ploting 1st diagram
sns.histplot(data=df,x='average',bins=30,kde=True,color='g')
plt.subplot(122)
sns.histplot(data=df,x='average',bins=30,kde=True,hue='gender')
```

[33]:
```
<AxesSubplot: xlabel='average', ylabel='Count'>
```

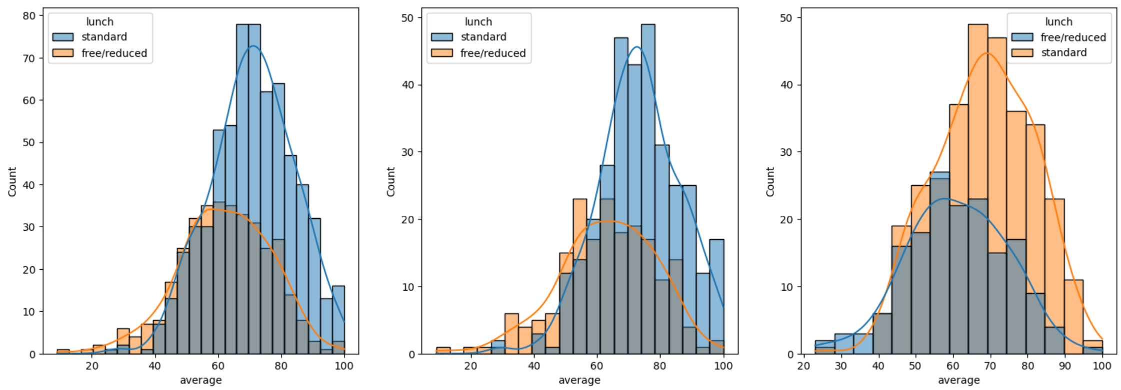


0.6 Insights

- Female student tend to perform well than male students

[38]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='lunch')
plt.subplot(142)
sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='lunch')
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='lunch')
```

[38]:
```
<AxesSubplot: xlabel='average', ylabel='Count'>
```



0.7 Insights

- Standard Lunch help students perform well in exams
- Standard lunch helps perform well in exams be it a male of female

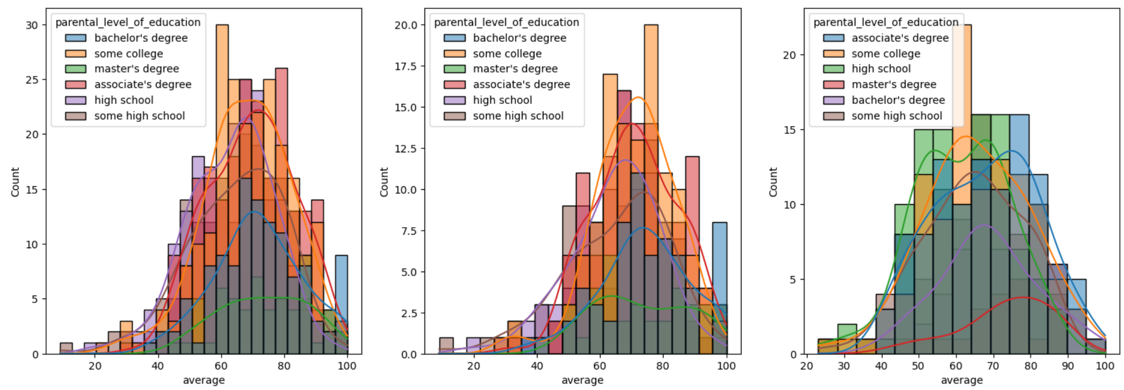
[39]:
```
df.head()
```

[39]:
```
   gender race_ethnicity parental_level_of_education         lunch  \
0  female        group B           bachelor's degree      standard
1  female        group C                some college      standard
2  female        group B             master's degree      standard
3    male        group A          associate's degree  free/reduced
4    male        group C                some college      standard

  test_preparation_course  math_score  reading_score  writing_score  \
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75

   total_score    average
0          218  72.666667
1          247  82.333333
2          278  92.666667
3          148  49.333333
4          229  76.333333
```

[40]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='parental_level_of_education')
plt.subplot(142)
```

```
sns.histplot(data=df[df.
 ↪gender=='female'],x='average',kde=True,hue='parental_level_of_education')
plt.subplot(143)
sns.histplot(data=df[df.
 ↪gender=='male'],x='average',kde=True,hue='parental_level_of_education')
```

[40]: <AxesSubplot: xlabel='average', ylabel='Count'>



Insights

- In general parent's education don't help student perform well in exam.
- 3rd plot shows that parent's whose education is of associate's degree or master's degree their male child tend to perform well in exam
- 2nd plot we can see there is no effect of parent's education on female students.

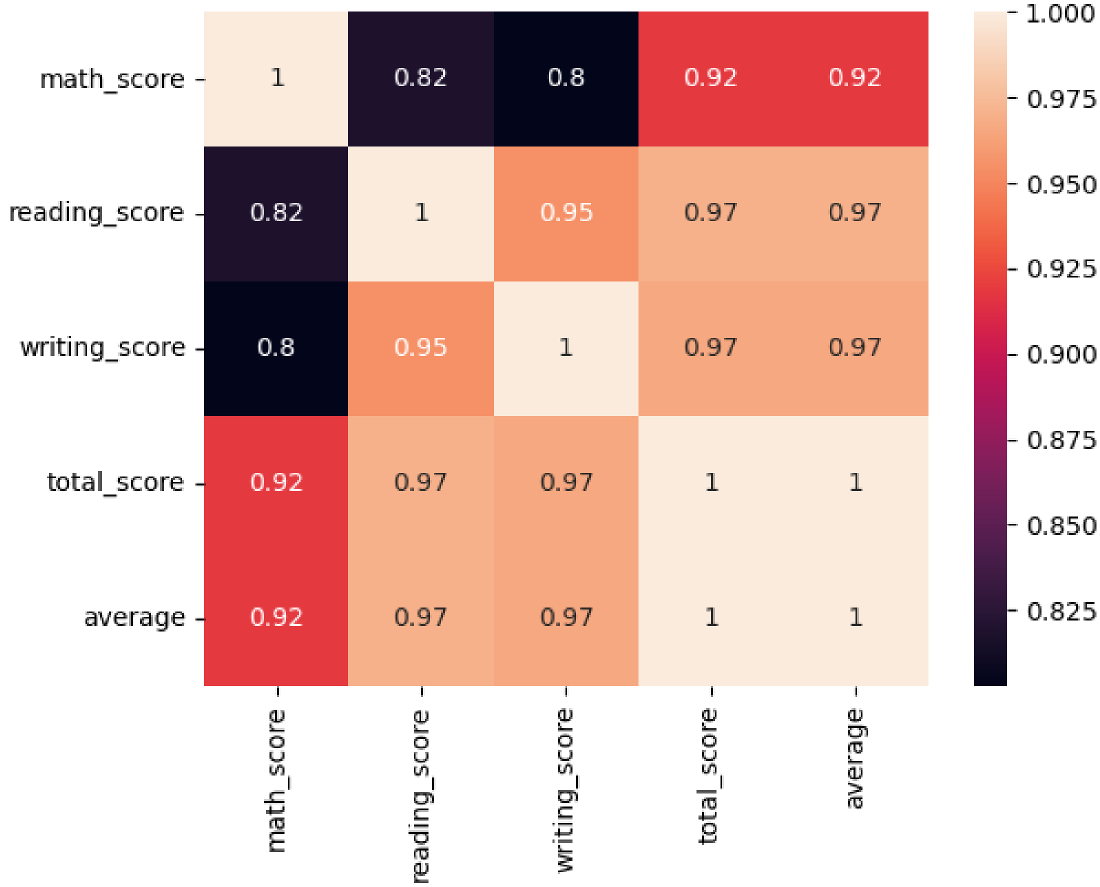
plt.subplots(1,3,figsize=(25,6)) plt.subplot(141) ax =sns.histplot(data=df,x=‘average’,kde=True,hue=‘race_ethni
plt.subplot(142) ax =sns.histplot(data=df[df.gender==‘female’],x=‘average’,kde=True,hue=‘race_ethnicity’)
plt.subplot(143) ax =sns.histplot(data=df[df.gender==‘male’],x=‘average’,kde=True,hue=‘race_ethnicity’)
plt.show()

Insights

- Students of group A and group B tends to perform poorly in exam.
- Students of group A and group B tends to perform poorly in exam irrespective of whether they are male or female
- Students of group E tends to perform good in exam irrespective of whether they are male or female

[45]:
```
sns.heatmap(df.corr(),annot=True)
```

[45]: <AxesSubplot: >



math_score
reading_score
writing_score
total_score
average
1
0.82
0.8
0.92
0.92
0.82
1
0.95
0.97
0.97
0.8
0.95
1
0.97
0.97
0.92
0.97
0.97
1
1
0.92
0.97
0.97
1
1
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825


```
[ ]:
```