24th March Assignment

April 4, 2023

1 Assignment 47

[25]:
```
import pandas as pd
```

[26]:
```
df=pd.read_csv('winequality-red.csv',delimiter=';')
```

[27]:
```
df.head()
```

[27]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Q1. What are the key features of the wine quality data set? Discuss the importance of each feature in predicting the quality of wine.

Ans. The wine quality dataset is a collection of various chemical properties of wines, along with their quality ratings by human tasters. The dataset has two versions: one for red wine, and one for white wine. Here are the key features of the dataset:

1. Fixed acidity: This feature measures the amount of non-volatile acids present in the wine, which can affect its taste and overall quality.
2. Volatile acidity: This feature measures the amount of volatile acids present in the wine, which can contribute to its sourness or vinegary taste. High levels of volatile acidity can indicate that the wine is spoiled.
3. Citric acid: This feature measures the amount of citric acid present in the wine, which can contribute to its freshness and fruity flavor.
4. Residual sugar: This feature measures the amount of sugar remaining in the wine after fermentation, which can affect its sweetness and overall taste.
5. Chlorides: This feature measures the amount of salt in the wine, which can contribute to its flavor and texture.
6. Free sulfur dioxide: This feature measures the amount of sulfur dioxide gas dissolved in the wine, which is used as a preservative and can affect its aroma and taste.
7. Total sulfur dioxide: This feature measures the total amount of sulfur dioxide present in the wine, which is a combination of the free and bound sulfur dioxide. High levels of total sulfur dioxide can indicate that the wine has been over-sulfited.
8. Density: This feature measures the density of the wine, which can indicate its alcohol content and overall body.
9. pH: This feature measures the acidity or basicity of the wine, which can affect its taste and stability.
10. Sulphates: This feature measures the amount of sulfur compounds present in the wine, which can affect its aroma and taste.
11. Alcohol: This feature measures the percentage of alcohol present in the wine, which can affect its flavor and body.
12. Quality: This feature is the target variable, and represents the quality rating of the wine on a scale from 0 to 10, as determined by human tasters.

Each feature plays an important role in predicting the quality of wine, as they provide information about various chemical and sensory characteristics of the wine. For example, high levels of volatile acidity can indicate that the wine is spoiled and has a lower quality, while higher alcohol content can indicate a fuller body and richer flavor. By analyzing the different features and their relationships with the quality rating, it is possible to develop a model that can accurately predict the quality of wine based on its chemical composition.

Q2. How did you handle missing data in the wine quality data set during the feature engineering process? Discuss the advantages and disadvantages of different imputation techniques.

[28]:
```
# Check for Missing Values:

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

[28]:
```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Observation As we checked for Missing Values in Red-Wine Dataset, we conclude that there is no any missing value present in Red-Wine Dataset.

But, there are several techniques that can be used to handle missing data, including:

1. Deletion: This involves removing any rows or columns that contain missing data. This can be useful if the missing data is a small percentage of the overall dataset and the remaining data is still sufficient to build a good model. However, deletion can result in loss of valuable information and reduce the sample size, which may affect the accuracy of the model.
2. Mean/Median/Mode Imputation: This involves filling in missing values with the mean, median, or mode of the corresponding feature. This technique is simple and easy to implement, and can help preserve the overall distribution of the data. However, it can lead to biased or inaccurate results, particularly if the missing data is not missing at random or if the feature has a skewed distribution.
3. Regression Imputation: This involves using regression analysis to predict missing values based on other features in the dataset. This technique can be more accurate than mean/median/mode imputation, particularly if there are strong relationships between the missing feature and other features in the dataset. However, it can be computationally intensive and may not work well if there are too many missing values or if the relationships between features are weak.
4. K-Nearest Neighbors (KNN) Imputation: This involves using the values of the nearest neighbors to predict the missing values. This technique can be effective if there are strong correlations between neighboring observations and can preserve the overall distribution of the data. However, it can be computationally intensive and may not work well if there are too many missing values or if the dataset has a large number of features.

The advantages and disadvantages of different imputation techniques:

Deletion:

Advantages:

- Simple and easy to implement.
- Does not require any assumptions about the missing data.
- Can be useful if the missing data is a small percentage of the overall dataset. ##### Disadvantages:
- Can result in loss of valuable information.
- Can reduce the sample size and affect the accuracy of the model.
- May introduce bias if the missing data is not missing at random. #### Mean/Median/Mode Imputation: ##### Advantages:
- Simple and easy to implement.
- Can help preserve the overall distribution of the data.
- Can work well if the missing data is missing at random and the feature has a roughly symmetric distribution. ##### Disadvantages:
- Can be inaccurate if the missing data is not missing at random or if the feature has a skewed distribution.
- Can introduce bias if the missing data is related to other variables in the dataset. #### Regression Imputation: ##### Advantages:
- Can be more accurate than mean/median/mode imputation.
- Can work well if there are strong relationships between the missing feature and other features in the dataset.
- Can preserve the overall distribution of the data. ##### Disadvantages:
- Can be computationally intensive, particularly for large datasets.
- May not work well if there are too many missing values or if the relationships between features are weak.
- Can introduce bias if the model used for regression is misspecified. #### K-Nearest Neighbors (KNN) Imputation: ##### Advantages:
- Can be effective if there are strong correlations between neighboring observations.
- Can preserve the overall distribution of the data.
- Can work well if there are many missing values or if the dataset has a large number of features. ##### Disadvantages:
- Can be computationally intensive, particularly for large datasets.
- Can introduce bias if the nearest neighbors are not representative of the overall dataset.
- Can be sensitive to the choice of the number of neighbors.

Q3. What are the key factors that affect students' performance in exams? How would you go about analyzing these factors using statistical techniques?

Q4. Describe the process of feature engineering in the context of the student performance data set. How did you select and transform the variables for your model?

solution for 3-4 is below

[29]:
```
df1 = pd.read_csv('StudentsPerformance.csv')

df1.head()
```

[29]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

[30]:
```
# check for Missing Value:

df1.isnull().sum()
```

[30]:
```
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
```

[31]:
```
# Check for Duplicate Values:

df1.duplicated().sum()
```

[31]:
```
0
```

Observation : There is no any Missing & Duplicate Values in Student Performance Dataset.

[32]:
```
# Check for Unique Values:

df1.nunique()
```

[32]:
```
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
```

[33]:
```
# Segrregate Numerical & Categorical Features:

Numerical_Features = [feature for feature in df1.columns if df1[feature].dtype!
 ↪='O']
Categorical_Feature = [feature for feature in df1.columns if df1[feature].
 ↪dtype=='O']

print(Numerical_Features)
print(Categorical_Feature)
```

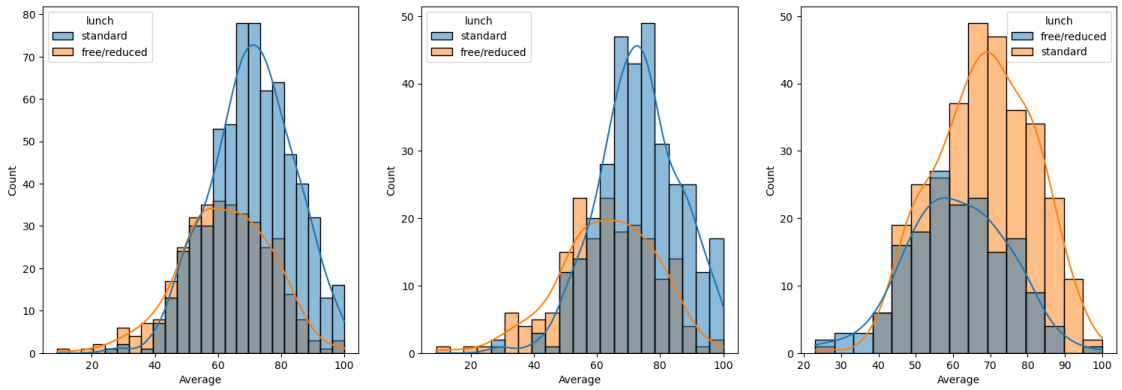
```
['math score', 'reading score', 'writing score']
['gender', 'race/ethnicity', 'parental level of education', 'lunch', 'test
preparation course']
```

[34]:
```
# Aggregate the Total Score with Mean:

df1['Total_Score'] = ( df1['math score'] + df1['reading score'] + df1['writing␣
 ↪score'] )
df1['Average'] = df1['Total_Score'] / 3
```

[35]:
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)

# Average:
sns.histplot(data=df1,x='Average',kde=True,hue='lunch')
plt.subplot(142)

# Female:
sns.histplot(data=df1[df1.gender=='female'],x='Average',kde=True,hue='lunch')
plt.subplot(143)

# Male:
sns.histplot(data=df1[df1.gender=='male'],x='Average',kde=True,hue='lunch')
```

```
plt.show()
```

```
/tmp/ipykernel_109/464007475.py:6: MatplotlibDeprecationWarning: Auto-removal of
overlapping axes is deprecated since 3.6 and will be removed two minor releases
later; explicitly call ax.remove() as needed.
  plt.subplot(141)
/tmp/ipykernel_109/464007475.py:10: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(142)
/tmp/ipykernel_109/464007475.py:14: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(143)
```



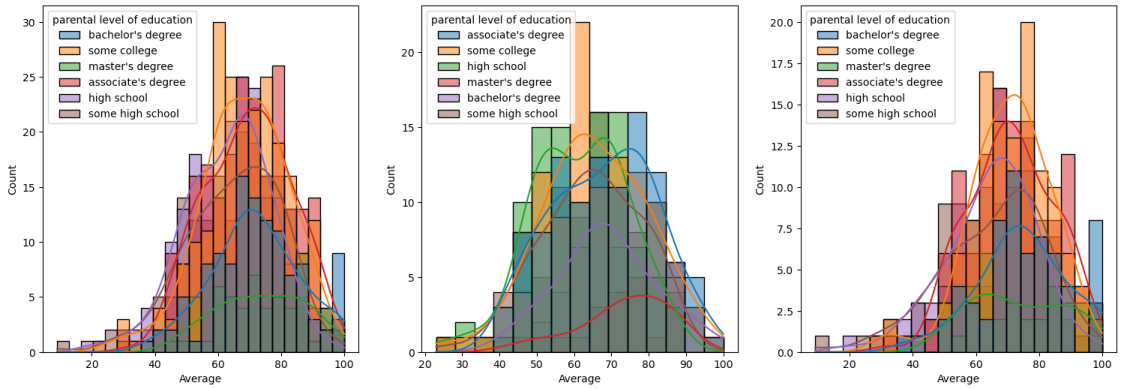
Insights

- Eating a standard lunch is positively correlated with better performance on exams for students.
- Both male and female students who eat a standard lunch tend to perform better on exams.

[36]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)

ax = sns.histplot(data=df1,x='Average',kde=True,hue='parental level of␣
 ↪education')
plt.subplot(142)

ax = sns.histplot(data=df1[df1.
 ↪gender=='male'],x='Average',kde=True,hue='parental level of education')
plt.subplot(143)
```

```
ax = sns.histplot(data=df1[df1.
  ↪gender=='female'],x='Average',kde=True,hue='parental level of education')
plt.show()
```

```
/tmp/ipykernel_109/3953374720.py:2: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(141)
/tmp/ipykernel_109/3953374720.py:5: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(142)
/tmp/ipykernel_109/3953374720.py:8: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(143)
```



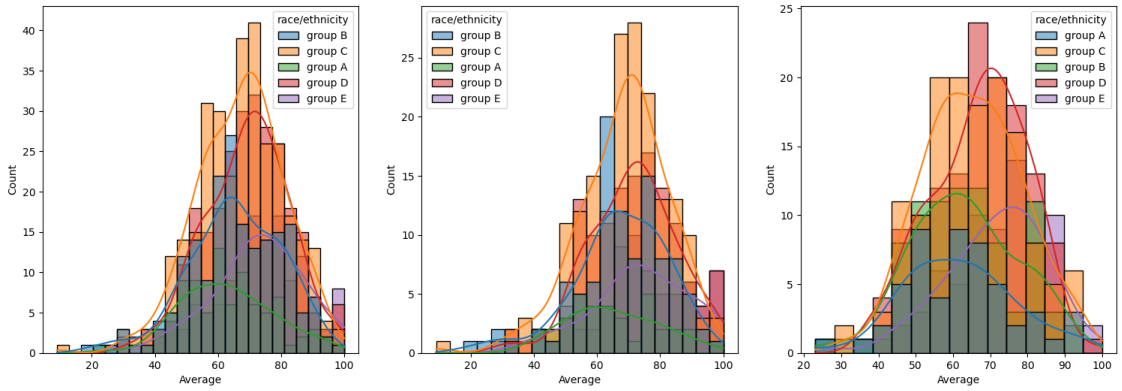
Observation:

- The first plot indicates that parent's education does not have a significant impact on student performance in exams.
- The second plot shows that male students whose parents have associate's or master's degrees tend to perform better.
- The third plot suggests that parent's education does not have a significant effect on the performance of female students.

[37]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)

ax =sns.histplot(data=df1,x='Average',kde=True,hue='race/ethnicity')
plt.subplot(142)
```

```
ax =sns.histplot(data=df1[df1.gender=='female'],x='Average',kde=True,hue='race/
 ↪ethnicity')
plt.subplot(143)

ax =sns.histplot(data=df1[df1.gender=='male'],x='Average',kde=True,hue='race/
 ↪ethnicity')
plt.show()
```

```
/tmp/ipykernel_109/4017723633.py:2: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(141)
/tmp/ipykernel_109/4017723633.py:5: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(142)
/tmp/ipykernel_109/4017723633.py:8: MatplotlibDeprecationWarning: Auto-removal
of overlapping axes is deprecated since 3.6 and will be removed two minor
releases later; explicitly call ax.remove() as needed.
  plt.subplot(143)
```



Observation :

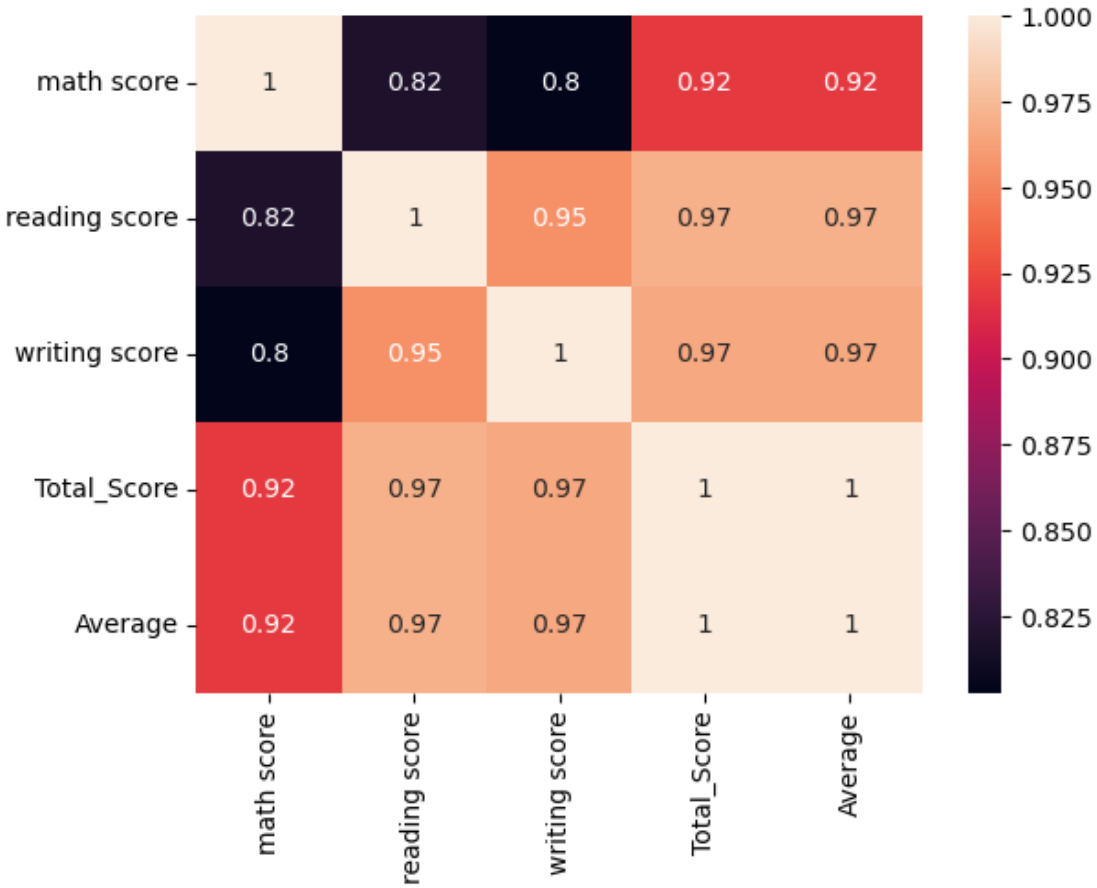
- Both groups A and B exhibit a tendency towards poor performance on exams.
- Both male and female students in groups A and B exhibit a tendency towards poor performance on exams.

[38]:
```
# HeatMap:

sns.heatmap(df1.corr(),annot=True)
```

```
/tmp/ipykernel_109/1226669325.py:3: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
```

```
to silence this warning.
  sns.heatmap(df1.corr(),annot=True)
```

[38]: <AxesSubplot: >



Q5. Load the wine quality data set and perform exploratory data analysis (EDA) to identify the distribution of each feature. Which feature(s) exhibit non-normality, and what transformations could be applied to these features to improve normality?

[39]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('winequality-red.csv',delimiter=';')
df.head()
```

[39]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	\
0	7.4	0.70	0.00	1.9	0.076	
1	7.8	0.88	0.00	2.6	0.098	

```
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
```

[41]:
```
import numpy as np
```

[42]:
```
# Log transform the "volatile acidity" feature
df["volatile acidity"] = np.log(df["volatile acidity"])

df.head()
```

[42]:
```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4         -0.356675         0.00             1.9      0.076
1            7.8         -0.127833         0.00             2.6      0.098
2            7.8         -0.274437         0.04             2.3      0.092
3           11.2         -1.272966         0.56             1.9      0.075
4            7.4         -0.356675         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
```

Q6. Using the wine quality data set, perform principal component analysis (PCA) to reduce the number of features. What is the minimum number of principal components required to explain 90% of the variance in the data?

[43]:
```
# Importing the necessary libraries
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Load the wine quality dataset
df = pd.read_csv('winequality-red.csv',delimiter=';')

# Standardize the data
scaler = StandardScaler()
X = scaler.fit_transform(df.drop('quality', axis=1))

# Perform PCA
pca = PCA()
pca.fit(X)

# Determine the number of principal components required to explain 90% of the␣
 ↪variance
total_variance = sum(pca.explained_variance_)
variance_threshold = 0.9
explained_variance = 0
num_components = 0

for variance in pca.explained_variance_ratio_:
    explained_variance += variance
    num_components += 1
    if explained_variance >= variance_threshold:
        break

print(f"Minimum number of principal components required to explain␣
 ↪{variance_threshold * 100}% of the variance: {num_components}")
```

```
Minimum number of principal components required to explain 90.0% of the
variance: 7
```