EDA Assignment - 1

February 20, 2024

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

- In order to predict the quality of wine, the dataset contains 12 crucial features. These features are listed below: #

1. Fixed acidity: This feature represents the concentration of non-volatile acids present in the wine. It plays a significant role in determining the overall taste and balance of the wine. #
2. Volatile acidity: This feature represents the concentration of acetic acid present in the wine. Excessive volatile acidity can make the wine taste and smell like vinegar, which is certainly undesirable. #
3. Citric acid: This weak organic acid is found in small amounts in wine, but it is important because it adds freshness and complexity to it. #
4. Residual sugar: This feature indicates how much sugar remains in the wine after fermentation. It is an essential characteristic as it impacts how sweet or dry a particular type of wine may be. #
5. Chlorides: The concentration of salts present in the wine is referred to as chlorides. High levels can indicate poor winemaking practices, while low levels are desirable for optimal taste. #
6. Free sulfur dioxide: The amount of sulfur dioxide added to preserve wines is known as free sulfur dioxide; this helps prevent oxidation and microbial spoilage. #
7. Total sulfur dioxide: Total sulfur dioxide refers to both free and bound amounts present within a given volume or batch - high levels could signify poor winemaking practices. #
8. Density: Density measures mass per unit volume; this property indicates alcohol content and sweetness level within a particular batch or bottle. #
9. pH: pH measures acidity or alkalinity levels within a given sample - this property influences color, stability, aroma, and flavor profile within different types of wines. #
10. Sulphates: Sulphates serve as preservatives that act as antioxidants; higher concentrations may suggest suboptimal winemaking techniques. #
11. Alcohol percentage by volume (ABV): Alcohol percentage by volume refers to how much alcohol is present in the wine - this property has a significant impact on the taste, aroma, and body of the wine. #
12. Quality (score between 0 and 10): This is a subjective measure of overall quality based on sensory evaluations - this is the target variable that we are trying to predict.

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.

- There are no missing values in the wine dataset. #
- However, advantages and disadvantages of different impuatation techniques are:
- There are several techniques available for imputing missing data in a dataset.

1. Mean Imputation
    - where missing values are replaced with the mean value of the feature.
    - Advantage
        - This technique is simple and easy to implement.
    - Disadvantage
        - The missing values are completely random and that the mean value is representative of the missing values, which may not always be true. #
2. Median Imputation
    - It is preffered while dealing with skewed data.
    - This method replaces missing values with the median value of the feature
    - Advantage
        - It is more robust to outliers compared to mean imputation.
    - Disadvantage
        - It assumes that the missing values are completely random. #
3. Regression Imputation
    - It involves using a regression model to predict missing values based on other features in the dataset.
    - Advantage
        - It can be more accurate than mean or median imputation as it takes into account relationships between features.
    - Disadvantage
        - This method assumes that the missing data is not biased and that the regression model used is correctly specified. #
4. Multiple imputation
    - It creates multiple datasets based on distributions of existing data and combines them to obtain more accurate estimates of missing values.
    - Advantage
        - This method accounts for uncertainty associated with imputing data.
    - Disadvantage
        - It is computationally intensive and impractical for large datasets.

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

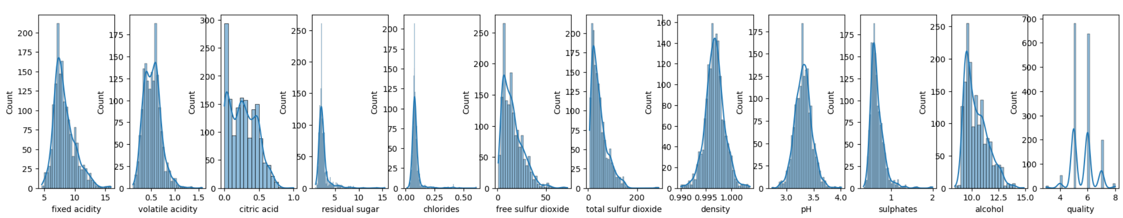
- The key factors that affect students' performance in exams are:
    - lunch
        * Standard lunch help students perform well in exams.
    - gender
        * Female student tend to perform well than male students.
    - race_ethnicity
        * Students of group A and group B tends to perform poorly in exam. #
- To analyze the above factors, I have used histogram visualisation technique.}

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?

- The process of feature engineering in the context of the student performance data set includes:
  - 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 #
- To transform the variables for the model includes:
  - Handling missing data
  - Encoding categorical variables:

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?

[10]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("winequality-red.csv")

l = df.columns.values
number_of_columns = 12
number_of_rows = int(len(l)-1/number_of_columns)

plt.figure(figsize=(2*number_of_columns, 5*number_of_rows))
for i in range(0, len(l)):
    plt.subplot(number_of_rows + 1, number_of_columns, i+1)
    sns.histplot(df[l[i]], kde=True)
```



- From the above visualization, the features exhibiting non-normality are:
  - Volatile Acidity
  - citric acid
  - residual sugar
  - chlorides
  - free sulfur dioxide
  - total sulfur dioxide

    - sulphates
    - alcohol #
- Transformations techniques that can be applied to these features to improve normality are :
  1. Log transformation
     - This is one of the most commonly used transformations for normalizing data.
     - It is particularly useful when the data is skewed to the right (i.e., positively skewed).
     - A log transformation can help to reduce the skewness of the data by compressing large values and expanding small values. #
  2. Square root transformation
     - This transformation is useful when the data is skewed to the right and the values are positive.
     - It can help to reduce the skewness and make the data more symmetric. #
  3. Box-Cox transformation
     - The Box-Cox transformation is useful when the data is skewed and the skewness cannot be corrected by a simple transformation.
     - It involves finding the best transformation parameter lambda that maximizes the normality of the data.
     - This is a family of transformations that includes both the log transformation and the square root transformation. #
  4. Reciprocal transformation
     - This transformation is useful when the data is skewed to the left (i.e., negatively skewed).
     - It can help to make the data more symmetric. #
  5. Exponential transformation
     - This transformation is useful when the data is skewed to the left and the values are positive.
     - It can help to make the data more symmetric and reduce the skewness.

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?

[15]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("winequality-red.csv")
df.head()
```

[15]:
```
   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
```

We will remove the 'Quality' of the wine as it is the target feature.

[16]:
```python
df.drop(columns=['quality'], inplace=True)
df.head()
```

[16]:
```
   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
0      9.4
1      9.8
2      9.8
3      9.8
4      9.4
```

Now we will scale the data

[17]:
```python
from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
df_scaled = pd.DataFrame(scalar.fit_transform(df), columns=df.columns)
df_scaled
```

[17]:
```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0         -0.528360          0.961877    -1.391472       -0.453218  -0.243707
1         -0.298547          1.967442    -1.391472        0.043416   0.223875
```

```
2         -0.298547          1.297065    -1.186070         -0.169427   0.096353
3          1.654856         -1.384443     1.484154         -0.453218  -0.264960
4         -0.528360          0.961877    -1.391472         -0.453218  -0.243707
…               …                 …            …          …        …
1594      -1.217796          0.403229    -0.980669         -0.382271   0.053845
1595      -1.390155          0.123905    -0.877968         -0.240375  -0.541259
1596      -1.160343         -0.099554    -0.723916         -0.169427  -0.243707
1597      -1.390155          0.654620    -0.775267         -0.382271  -0.264960
1598      -1.332702         -1.216849     1.021999          0.752894  -0.434990

      free sulfur dioxide  total sulfur dioxide   density        pH  \
0               -0.466193             -0.379133  0.558274  1.288643
1                0.872638              0.624363  0.028261 -0.719933
2               -0.083669              0.229047  0.134264 -0.331177
3                0.107592              0.411500  0.664277 -0.979104
4               -0.466193             -0.379133  0.558274  1.288643
…                     …                    …        …        …
1594             1.542054             -0.075043 -0.978765  0.899886
1595             2.211469              0.137820 -0.862162  1.353436
1596             1.255161             -0.196679 -0.533554  0.705508
1597             1.542054             -0.075043 -0.676657  1.677400
1598             0.203223             -0.135861 -0.666057  0.511130

      sulphates   alcohol
0     -0.579207 -0.960246
1      0.128950 -0.584777
2     -0.048089 -0.584777
3     -0.461180 -0.584777
4     -0.579207 -0.960246
…           …        …
1594  -0.461180  0.072294
1595   0.601055  0.729364
1596   0.542042  0.541630
1597   0.305990 -0.209308
1598   0.010924  0.541630

[1599 rows x 11 columns]
```

Now we are ready to apply for PCA.

[18]:
```
from sklearn.decomposition import PCA

pca = PCA()
df_pca = pd.DataFrame(pca.fit_transform(df_scaled))
df_pca
```

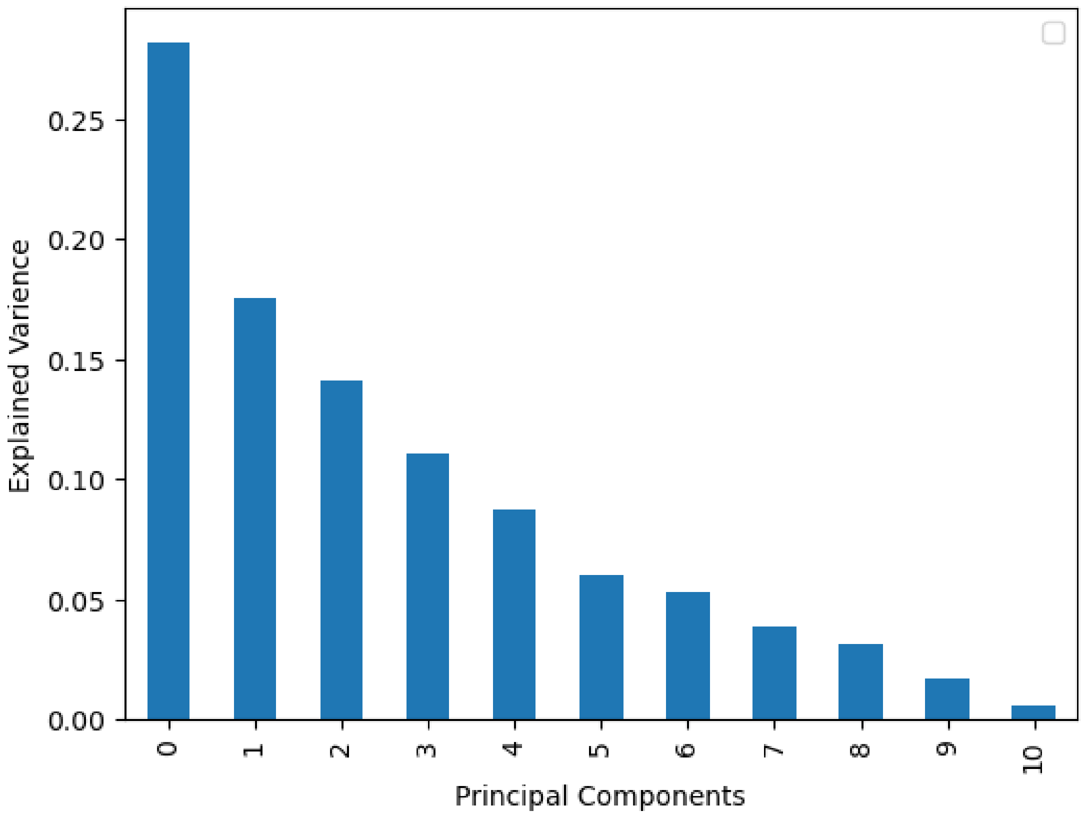
[18]:

```
             0         1         2         3         4         5         6  \
0    -1.619530  0.450950 -1.774454  0.043740  0.067014 -0.913921 -0.161043
1    -0.799170  1.856553 -0.911690  0.548066 -0.018392  0.929714 -1.009829
2    -0.748479  0.882039 -1.171394  0.411021 -0.043531  0.401473 -0.539553
3     2.357673 -0.269976  0.243489 -0.928450 -1.499149 -0.131017  0.344290
4    -1.619530  0.450950 -1.774454  0.043740  0.067014 -0.913921 -0.161043
…          …         …         …         …         …         …         …
1594 -2.150500  0.814286  0.617063  0.407687 -0.240936  0.054835  0.170812
1595 -2.214496  0.893101  1.807402  0.414003  0.119592 -0.674711 -0.607970
1596 -1.456129  0.311746  1.124239  0.491877  0.193716 -0.506410 -0.231082
1597 -2.270518  0.979791  0.627965  0.639770  0.067735 -0.860408 -0.321487
1598 -0.426975 -0.536690  1.628955 -0.391716  0.450482 -0.496154  1.189132

             7         8         9        10
0    -0.282258  0.005098 -0.267759  0.048630
1     0.762587 -0.520707  0.062833 -0.138142
2     0.597946 -0.086857 -0.187442 -0.118229
3    -0.455375  0.091577 -0.130393  0.316714
4    -0.282258  0.005098 -0.267759  0.048630
…          …         …         …         …
1594 -0.355866 -0.971524  0.356851 -0.053382
1595 -0.247640 -1.058135  0.478879 -0.241258
1596  0.079382 -0.808773  0.242248 -0.402910
1597 -0.468876 -0.612248  0.779404  0.040923
1598  0.042176  0.404309  0.779440 -0.449781

[1599 rows x 11 columns]
```

Now, we will look for variance for each of the PCA components

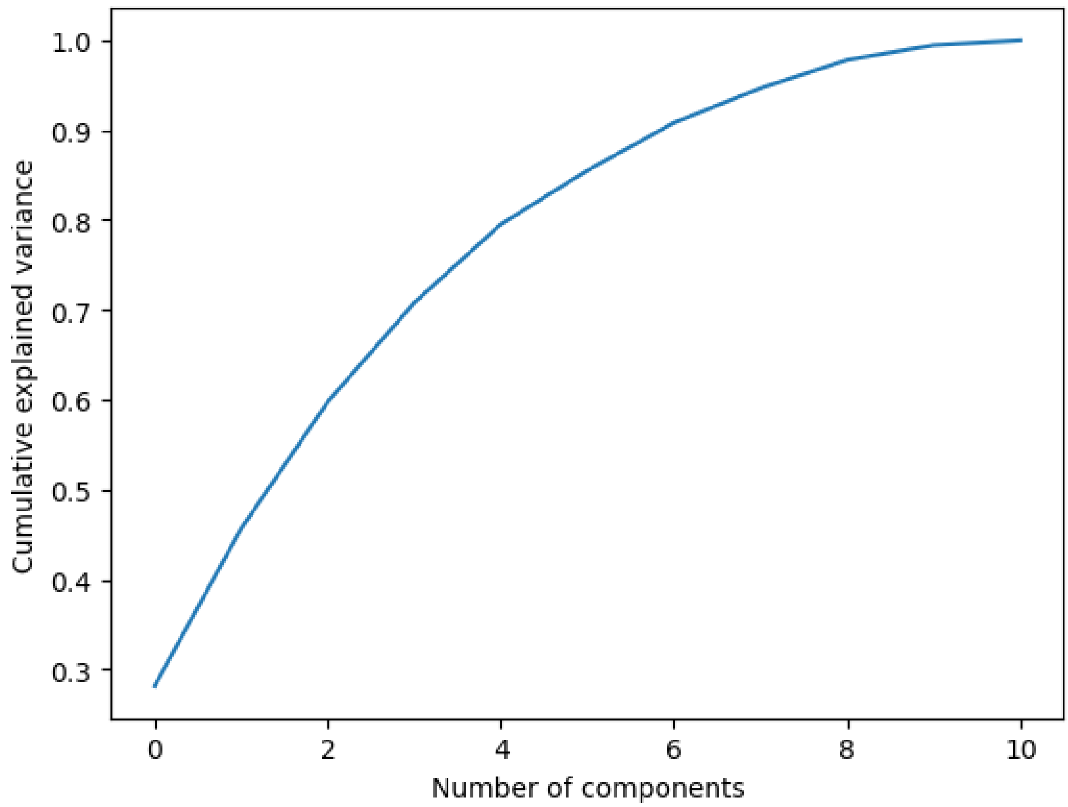
[26]:

```python
import matplotlib.pyplot as plt
pd.DataFrame(pca.explained_variance_ratio_).plot.bar()
plt.legend('')
plt.xlabel('Principal Components')
plt.ylabel('Explained Varience');
```



plot line graph of cumulative variance explained

[84]:
```
import numpy as np
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('Number of components')
plt.ylabel('Cumulative explained variance');
```



[85]:
```
pca_9 = PCA(.9)
pca_9.fit_transform(df_scaled)
print(
    "Minimum no of PCA components required to explain ~", round(
        pca_9.explained_variance_ratio_.sum()*100, 2), "of variance in the data␣
 ↪is",
    pca_9.n_components_,
    "components.")
```

```
Minimum no of PCA components required to explain ~ 90.83 of variance in the data
is 7 components.
```

- Minimum no of PCA components required to explain ~ 90.83 of variance in the data is 7 components.