ASSIGNMENT

INTELLIGENT DATA ANALYSIS

REPORT

BY
ANAGHA RAMADAS MULLOTH
STUDENT ID: 2583584

Table of Contents

THE DATA

The dataset is taken from Kaggle. The dataset is related to red variants of Spanish wines and describes several popularity and description metrics that affect their price and quality.

Link to the dataset: https://www.kaggle.com/datasets/fedesoriano/spanish-wine-quality-dataset/data

1. **Content**
   Number of instances (rows): 7500
   Number of features (columns): 11

2. **Attribute Information**

   Screenshot from Jupyter notebook

```
: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7500 entries, 0 to 7499
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   winery       7500 non-null   object 
 1   wine         7500 non-null   object 
 2   year         7498 non-null   object 
 3   rating       7500 non-null   float64
 4   num_reviews  7500 non-null   int64  
 5   country      7500 non-null   object 
 6   region       7500 non-null   object 
 7   price        7500 non-null   float64
 8   type         6955 non-null   object 
 9   body         6331 non-null   float64
 10  acidity      6331 non-null   float64
dtypes: float64(4), int64(1), object(6)
memory usage: 644.7+ KB
```

Attribute Name	Attribute datatype	Attribute Description
winery	object	Name of the Winery
wine	object	Name of the Wine

year	object	Year in which the grapes were harvested
rating	Float64	Average rating given to the wine by the users from 1-5
num_reviews	Int64	Number of users who reviewed the wine
country	object	Country of origin (Spain)
region	object	Region of the wine
price	Float64	Price in Euros
type	object	Wine variety
body	Float64	Score from 1-5 that is defined as the richness and weight of the wine in your mouth
acidity	Float64	Score from 1-5 that is defined as the wine's pucker or tartness which makes a wine refreshing and tasty.

DATA PREPROCESSING

1. **Label Encoding:**

   Label Encoding is a technique used in data preprocessing to convert categorical data into numerical representations. In label encoding, each unique category or label is assigned an integer value.
   An alternative for label encoding is one-hot encoding. However, one-hot encoding would create many dummy variables due to high number of categories for each attribute as shown below. Hence, we use label coding for attributes with categorical values in our dataset.

```
Number of unique categories in winery: 480
Number of unique categories in wine: 847
Number of unique categories in country: 1
Number of unique categories in region: 76
Number of unique categories in type: 21
```

2. **Convert Object data type to numeric data type**

Principal Component Analysis (PCA) operates on numerical data. PCA relies on mathematical operations such as covariance matrices, eigen value decomposition, and linear transformations, which are meaningful only for numerical data. Hence, we convert attributes with object data type in our dataset to numeric data type.

```
print(df.dtypes)
```

```
winery           int64
wine             int64
year           float64
rating         float64
num_reviews      int64
country          int64
region           int64
price          float64
type             int64
body           float64
acidity        float64
```

3. **Handling Missing Values - Imputation with Mode**

Handling missing values is an important step in data preprocessing. Here, we replace missing values in our dataset with the mode of the respective columns. This process is known as imputation. Imputation can also be performed using mean or median of the respective columns too. Imputation using mean is mostly preferred for numerical data and imputation using median is suitable for variables with a natural order. However, since our dataset contains label encoded values of categorical data with no natural order, mode is the more meaningful solution as it provides the numerical value of the most frequent category.

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

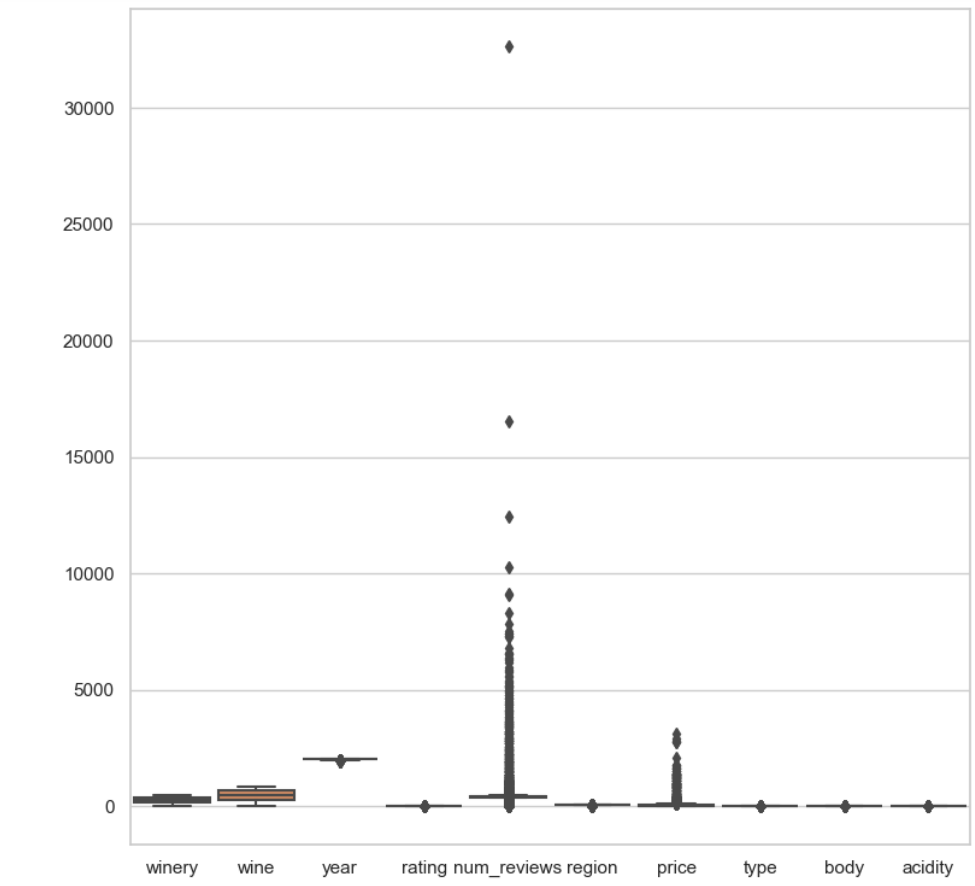
```
winery         0
wine           0
year           0
rating         0
num_reviews    0
country        0
region         0
price          0
type           0
body           0
acidity        0
dtype: int64
```

4. **Dropping Redundant Attributes**

The 'country' attribute in our dataset contains a single value as the entire data regarding wines belong to the Country of Spain. Hence, we drop this column to avoid redundancy.
The final preprocessed data is given below

```
df.head()
```

	winery	wine	year	rating	num_reviews	region	price	type	body	acidity
0	422	759	2013.0	4.9	58	69	995.00	19	5.0	3.0
1	33	819	2018.0	4.9	31	74	313.50	18	4.0	2.0
2	447	778	2009.0	4.8	1793	57	324.95	11	5.0	3.0
3	447	778	1999.0	4.8	1705	57	692.96	11	5.0	3.0
4	447	778	1996.0	4.8	1309	57	778.06	11	5.0	3.0



5. **Removing outliers**

Outliers are data points that deviate significantly from the rest of the data and can have substantial impact on the results of statistical analysis and machine learning models. The presence of outliers can affect the performance and influence the results of predictive models. Removing outliers is a common preprocessing step in data analysis. The decision to remove outliers from our dataset is made after careful analysis of the data in order to avoid removal of relevant data points and valuable information.

We analysis the Descriptive statistics of our columns which contain the following information:

a. Count: number of non-null values in the column
b. Mean: average value of the column
c. Standard Deviation (std): The measure of the amount of variation or dispersion of the values in the column.
d. Min: smallest value in the column
e. 25% or Q1: The value below which 25% of the data falls.
f. 50% or Q2: The middle value of the dataset
g. 75% or Q3: The value below which 75% of the data falls.
h. Max: largest value in the column.

```
Descriptive Statistics:
            winery         wine         year       rating   num_reviews  \
count  7500.000000  7500.000000  7500.000000  7500.000000   7500.000000
mean    258.368667   467.920267  2013.554000     4.254933    451.109067
std     128.547692   231.513538     6.811104     0.118029    723.001856
min       0.000000     0.000000  1910.000000     4.200000     25.000000
25%     158.000000   260.000000  2011.000000     4.200000    389.000000
50%     285.000000   496.000000  2015.000000     4.200000    404.000000
75%     373.000000   666.000000  2017.000000     4.200000    415.000000
max     479.000000   846.000000  2021.000000     4.900000  32624.000000

            region        price         type         body      acidity
count  7500.000000  7500.000000  7500.000000  7500.000000  7500.000000
mean     50.197733    60.095822    11.859867     4.133733     2.954933
std      17.308585   150.356676     4.351899     0.539031     0.228858
min       0.000000     4.990000     0.000000     2.000000     1.000000
25%      53.000000    18.900000    10.000000     4.000000     3.000000
50%      57.000000    28.530000    12.000000     4.000000     3.000000
75%      59.000000    51.350000    12.000000     4.000000     3.000000
max      75.000000  3119.080000    21.000000     5.000000     3.000000
```

Based on the Descriptive Statistics information, the attributes are selected from which outliers need to be removed such that it improves the accuracy of the summary statistics by reflecting the spread of the majority of the data. Outliers are removed from the following attributes using the technique of Winsorizing.

a. winery
b. wine
c. num_reviews
d. price

Winsorizing is a statistical technique used to handle outliers in a dataset by replacing extreme values with values closer to the center of the distribution. The process involves setting a threshold, and any data point beyond that threshold are replaced with the

nearest datapoint within the threshold. This technique reduces the impact of extreme values on statistical analyses and models while retaining majority of the data.

The descriptive statistics after removing outliers is given below:

```
Descriptive Statistics:
              winery         wine         year       rating  num_reviews  \
count  7500.000000  7500.000000  7500.000000  7500.000000  7500.000000
mean    258.495867   468.796133  2013.554000     4.254933   368.088133
std     127.247946   226.341232     6.811104     0.118029   144.493514
min      33.000000    92.000000  1910.000000     4.200000    45.000000
25%     158.000000   260.000000  2011.000000     4.200000   389.000000
50%     285.000000   496.000000  2015.000000     4.200000   404.000000
75%     373.000000   666.000000  2017.000000     4.200000   415.000000
max     453.000000   790.000000  2021.000000     4.900000   693.000000

              region        price         type         body      acidity
count  7500.000000  7500.000000  7500.000000  7500.000000  7500.000000
mean     50.197733    42.512718    11.859867     4.133733     2.954933
std      17.308585    34.222967     4.351899     0.539031     0.228858
min       0.000000    11.950000     0.000000     2.000000     1.000000
25%      53.000000    18.900000    10.000000     4.000000     3.000000
50%      57.000000    28.530000    12.000000     4.000000     3.000000
75%      59.000000    51.350000    12.000000     4.000000     3.000000
max      75.000000   150.000000    21.000000     5.000000     3.000000
```

FEATURES USED FOR LABELLING THE DATAPOINTS

The price and rating attributes will be used for labelling the projected datapoints in the dataset.

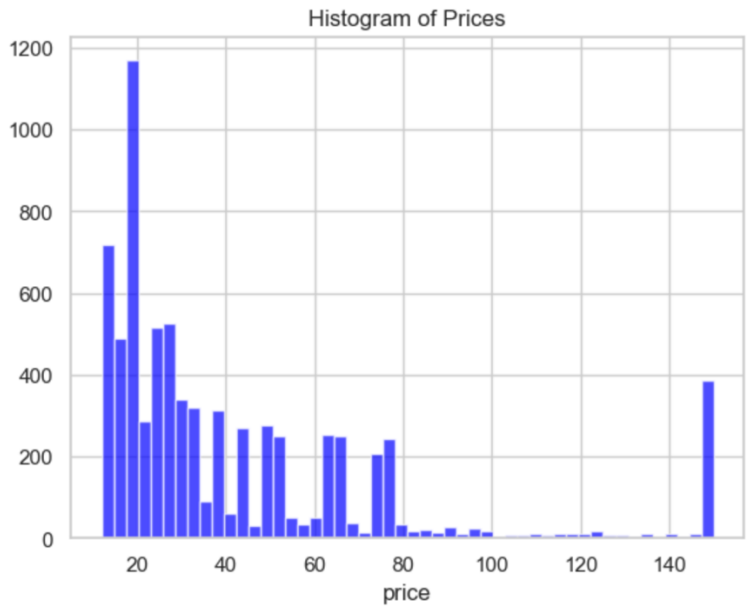
Questions asked about the data

- What dimension can the data be reduced to while retaining 70% of the total variance (i.e., the Cumulative variance threshold is set to 70%)
- How much variance is explained by each Principal Component
- What features contribute the most to each Principal Component

DESIGNING THE LABELLING SCHEMES

Price feature as the label

The following is the Histogram representing the Price distribution in the dataset.

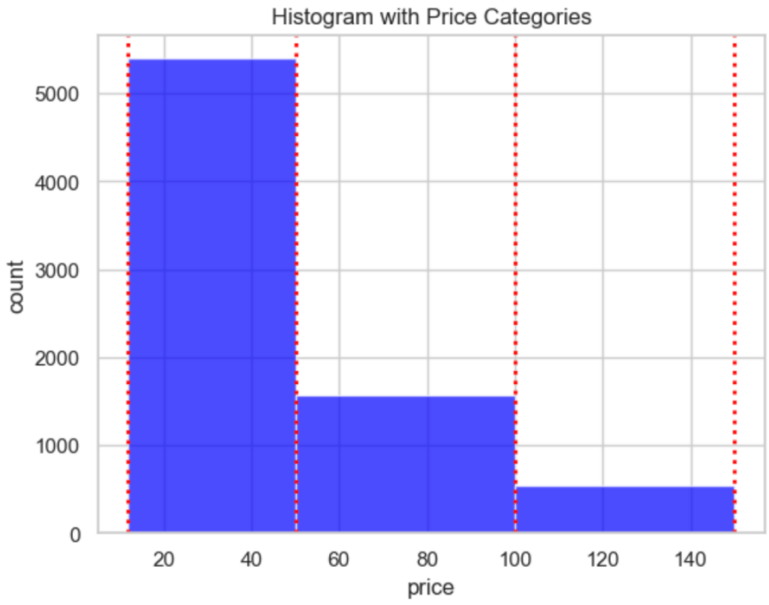


From the descriptive statistics earlier, the prices of the wines vary between 11.95 euros and 150 euros.

From this histogram, we see that most of the wines prices fall in the range upto 100 euros. Hence, splitting the data points into three categorical labels based on prices seems apt for the dataset.

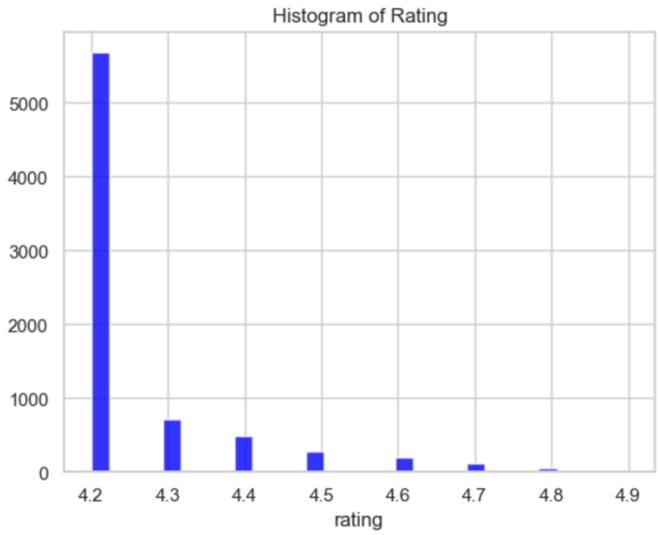
The prices are split into three categorical labels ‘Cheap’, ‘Medium’ and ‘Expensive’ in the ranges [11.95, 50], [50, 100] and, [100, 150] respectively.

Given below is the Histogram with the dotted red lines depicting the three price categories:



Rating feature as the label

The following is the Histogram representing the Rating distribution in the dataset.

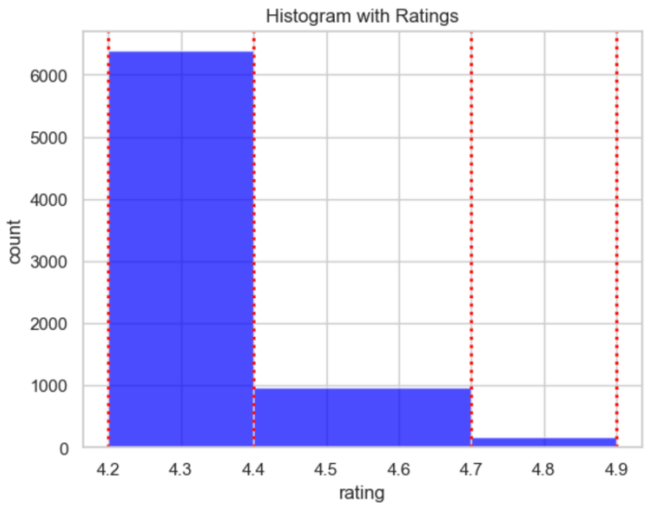


From the descriptive statistics earlier, the ratings of the wines vary between 4.2 and 4.9.

From this histogram, we see that most of the wines ratings fall in the range upto 4.7. Hence, splitting the data points into three categorical labels based on ratings also seems apt for the dataset.

The ratings are split into three categorical labels ‘Poor, ‘Good and ‘Very Good in the ranges [4.2, 4.4], [4.4, 4.7] and, [4.7, 4.9] respectively.

Given below is the Histogram with the dotted red lines depicting the three rating categories:



DIMENSIONALITY REDUCTION – PRINCIPAL COMPONENT ANALYSIS

Price feature as the label

Before performing dimensionality reduction we remove the feature used for labelling ('price')

The steps involved in PCA analysis are:

Standardize the Data – this process involves subtracting the mean and dividing by the standard deviation for each feature. Standardization ensures that all features contribute equally to PCA by standardizing the units or scales of measurement of all the features.

```
scaler = StandardScaler()
standardized_df = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)
print(standardized_df)
```

```
        winery      wine      year    rating  num_reviews    region      type  \
0     1.285011  1.282238 -0.081343  5.465686    -2.146178  1.086370  1.640803
1    -1.772216  1.419208  0.652801  5.465686    -2.236153  1.375263  1.411003
2     1.481491  1.366187 -0.668659  4.618381     2.248776  0.393026 -0.197597
3     1.481491  1.366187 -2.136947  4.618381     2.248776  0.393026 -0.197597
4     1.481491  1.366187 -2.577434  4.618381     2.248776  0.393026 -0.197597
...        ...       ...       ...       ...          ...       ...       ...
7495 -0.781958  0.663661  0.359143 -0.465453     0.165499  0.508583  0.032203
7496 -0.789817 -0.127233  0.652801 -0.465453     0.151656  0.161911 -0.657198
7497  0.593402 -0.736972  0.505972 -0.465453     0.151656 -1.513670 -0.427398
7498  0.451936 -0.560236 -0.375001 -0.465453     0.144735  0.393026 -0.197597
7499  1.159264  0.562038  0.359143 -0.465453     0.137814  0.393026 -0.197597

          body   acidity
0     1.607189  0.196933
1    -0.248116 -4.172880
2     1.607189  0.196933
3     1.607189  0.196933
4     1.607189  0.196933
...        ...       ...
7495 -0.248116  0.196933
7496 -0.248116  0.196933
7497 -0.248116  0.196933
7498  1.607189  0.196933
7499  1.607189  0.196933

[7500 rows x 9 columns]
```

Calculate the Covariance Matrix – The covariance matrix gives insights into how the features in the dataset vary together.

```
covariance_matrix = standardized_df.cov()
print(covariance_matrix)
```

```
                winery      wine      year    rating  num_reviews    region  \
winery        1.000133 -0.156249 -0.164581 -0.017580     0.019156  0.060096
wine         -0.156249  1.000133  0.003286  0.035521     0.077322  0.153108
year         -0.164581  0.003286  1.000133 -0.297269     0.156859 -0.133129
rating       -0.017580  0.035521 -0.297269  1.000133    -0.450754  0.044447
num_reviews   0.019156  0.077322  0.156859 -0.450754     1.000133  0.016135
region        0.060096  0.153108 -0.133129  0.044447     0.016135  1.000133
type         -0.123894  0.156407  0.137814 -0.026709     0.011584  0.271471
body          0.088714 -0.041725 -0.115361  0.166659     0.016649  0.314553
acidity       0.222874 -0.151821  0.154372 -0.098909     0.092672 -0.153030

                 type      body   acidity
winery      -0.123894  0.088714  0.222874
wine         0.156407 -0.041725 -0.151821
year         0.137814 -0.115361  0.154372
rating      -0.026709  0.166659 -0.098909
num_reviews  0.011584  0.016649  0.092672
region       0.271471  0.314553 -0.153030
type         1.000133  0.262357 -0.184174
body         0.262357  1.000133 -0.026810
acidity     -0.184174 -0.026810  1.000133
```

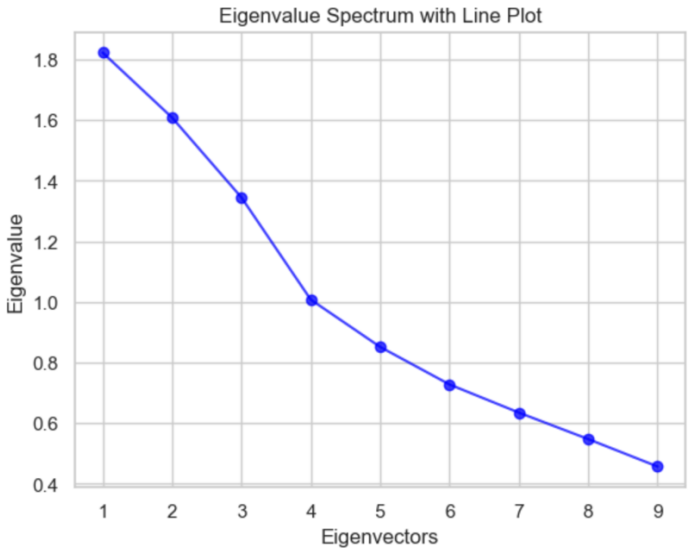
Calculate the Eigenvectors and Eigenvalues

– Eigen vectors represent the directions of maximum variance, and eigen values indicate the magnitude of variance along each eigenvector.

Sort Eigenvectors by Eigenvalues

– Sort the eigenvectors in descending order based on their corresponding eigen values. The eigenvectors with higher eigenvalues capture more variance in the data.

Given below is the Eigen value spectrum plot depicting the plot of eigenvalues for each eigenvector in the descending order of eigenvalues

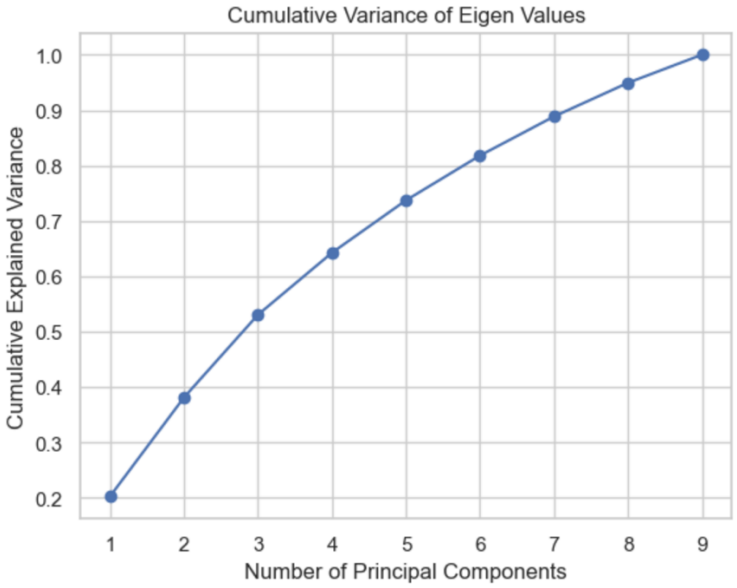


Explained Variance

– Calculate the explained variance, which represents the proportion of the total variance explained by each principal component.

```
Explained Variance
 [0.20248349 0.38115827 0.53056531 0.64249943 0.73710128 0.81793443
 0.88845029 0.94925164 1.        ]
```

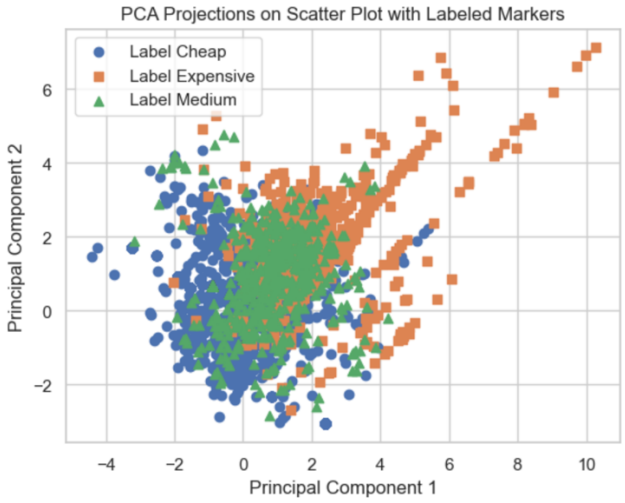
Given below is the plot of cumulative explained variance for each of the principal components.



Select the Principal Components (PCs) – Choose the top k eigen vectors corresponding to k highest eigen values such that the principal components retain at least 70% of total variance in the original data. This will determine the number of dimensions to which the data will be reduced. According to the cumulative variance plot given above the top 5 principal components need to be chosen to capture more than 70% of the cumulative explained variance. This means that we can preserve 70% of the relevant information from original dataset by reducing the dimension from 9 to 5. We can also reduce the dimension to 4 at the cost of reducing the amount of relevant information kept. The total variance gets reduced to approximately 60% by choosing k=4 for our principal component analysis.

Project the data onto Principal Components –

Project the data onto the selected principal components to obtain the new feature space. For demonstrative purposes the plot of the projection of datapoints on PC1 and PC2 (the first 2 principal components / dominant eigenvectors) is given below.



From the above plot we can see that not much relation can be made out of the 3 labelled categories. This is because from the earlier Cumulative Variance of Eigen Vectors plot it is evident that the first 2 principal components only retain approximately 40% of the total variance of the original information.

Interesting aspects regarding the Data based on Eigenvalue and Eigenvector Analysis of the Covariance Matrix

The entries in each eigen vector correspond to weights or coefficients of the original features in the principal components. The larger the magnitude of an entry, the more the influence the corresponding original feature has on that principal component. The sign of the weight indicates whether the feature is positively or negatively correlated to that principal component. This means that as a feature with positive weight increases, the principal component score also increases while when a feature having negative weight increases, the principal component score decreases.

The weights corresponding to each feature in the top 5 principal components (that preserve 70% of total variance) are given below:

Features	PC1	PC2	PC3	PC4	PC5
Winery	-0.054720	0.272800	-0.592292	0.210252	-0.154737
Wine	0.183529	-0.324772	0.272094	0.456989	-0.687926
Year	-0.344537	-0.369021	0.110995	-0.529411	-0.212220
Rating	0.438537	0.410851	0.166498	-0.168020	-0.270862
Num_reviews	-0.310334	-0.425650	-0.276774	0.369169	0.152180
Region	0.421843	-0.267216	-0.324263	0.159223	-0.047660
Type	0.310820	-0.469383	-0.056420	-0.371954	-0.025699
Body	0.387972	-0.115190	-0.465834	-0.277635	0.060001
acidity	-0.361818	0.170165	-0.364937	-0.247748	-0.595583

The features rating, region, body, type and wine (ordered in descending order of their influence) are positively correlated to PC1 while the features acidity, year, num_reviews and, winery (ordered in descending order of their influence) are negatively correlated to PC1.

Similarly we can see the correlation of each feature on principal components 2, 3, 4 and 5.

Rating feature as the label

Before performing dimensionality reduction we remove the feature used for labelling (‘rating)

The steps involved in PCA analysis are:

Standardize the Data – This step is similar to the one we performed while using price as the label feature.

```
scaler = StandardScaler()
standardized_rating_d_x = pd.DataFrame(scaler.fit_transform(rating_dataset_X), columns=rating_dataset_X.columns)

print(standardized_rating_d_x)
```

```
        winery      wine      year  num_reviews    region     price      type  \
0     1.285011  1.282238 -0.081343    -2.146178  1.086370  3.141003  1.640803
1    -1.772216  1.419208  0.652801    -2.236153  1.375263  3.141003  1.411003
2     1.481491  1.366187 -0.668659     2.248776  0.393026  3.141003 -0.197597
3     1.481491  1.366187 -2.136947     2.248776  0.393026  3.141003 -0.197597
4     1.481491  1.366187 -2.577434     2.248776  0.393026  3.141003 -0.197597
...        ...       ...       ...          ...       ...       ...       ...
7495 -0.781958  0.663661  0.359143     0.165499  0.508583 -0.658453  0.032203
7496 -0.789817 -0.127233  0.652801     0.151656  0.161911 -0.752548 -0.657198
7497  0.593402 -0.736972  0.505972     0.151656 -1.513670 -0.527830 -0.427398
7498  0.451936 -0.560236 -0.375001     0.144735  0.393026  0.642514 -0.197597
7499  1.159264  0.562038  0.359143     0.137814  0.393026 -0.318016 -0.197597

          body   acidity
0     1.607189  0.196933
1    -0.248116 -4.172880
2     1.607189  0.196933
3     1.607189  0.196933
4     1.607189  0.196933
...        ...       ...
7495 -0.248116  0.196933
7496 -0.248116  0.196933
7497 -0.248116  0.196933
7498  1.607189  0.196933
7499  1.607189  0.196933

[7500 rows x 9 columns]
```

Calculate the Covariance Matrix – The covariance matrix gives insights into how the features in the dataset vary together

```
covariance_matrix_d_x = standardized_rating_d_x.cov()
print(covariance_matrix_d_x)
```

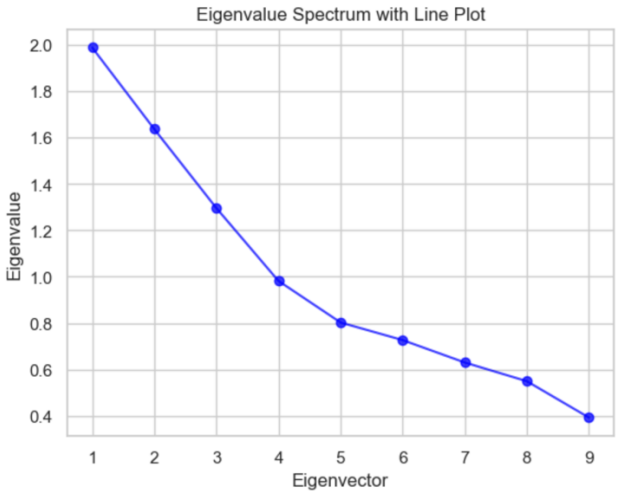
```
                winery      wine      year  num_reviews    region     price  \
winery       1.000133 -0.156249 -0.164581     0.019156  0.060096  0.062602
wine        -0.156249  1.000133  0.003286     0.077322  0.153108 -0.047403
year        -0.164581  0.003286  1.000133     0.156859 -0.133129 -0.541723
num_reviews  0.019156  0.077322  0.156859     1.000133  0.016135 -0.323570
region       0.060096  0.153108 -0.133129     0.016135  1.000133  0.202962
price        0.062602 -0.047403 -0.541723    -0.323570  0.202962  1.000133
type        -0.123894  0.156407  0.137814     0.011584  0.271471 -0.049272
body         0.088714 -0.041725 -0.115361     0.016649  0.314553  0.248761
acidity      0.222874 -0.151821  0.154372     0.092672 -0.153030 -0.149681

                type      body   acidity
winery      -0.123894  0.088714  0.222874
wine         0.156407 -0.041725 -0.151821
year         0.137814 -0.115361  0.154372
num_reviews  0.011584  0.016649  0.092672
region       0.271471  0.314553 -0.153030
price       -0.049272  0.248761 -0.149681
type         1.000133  0.262357 -0.184174
body         0.262357  1.000133 -0.026810
acidity     -0.184174 -0.026810  1.000133
```

Calculate the Eigenvectors and Eigenvalues – This step is similar to the one we performed while using price as the label feature.

Sort Eigenvectors by Eigenvalues – This step is similar to the one we performed while using price as the label feature.

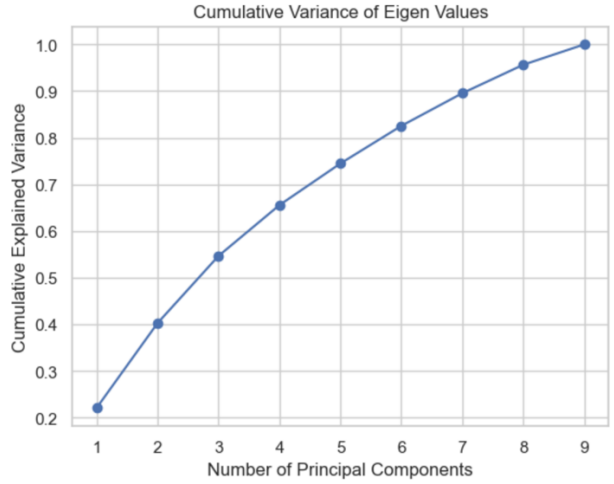
Given below is the Eigen value spectrum plot depicting the plot of eigenvalues for each eigenvector in the descending order of eigenvalues.



Explained Variance

```
Explained variance
 [0.22087556 0.40266638 0.54661272 0.65565814 0.74478997 0.82547505
 0.8954393  0.9564317  1.        ]
```

Given below is the plot of cumulative explained variance for each of the principal components.

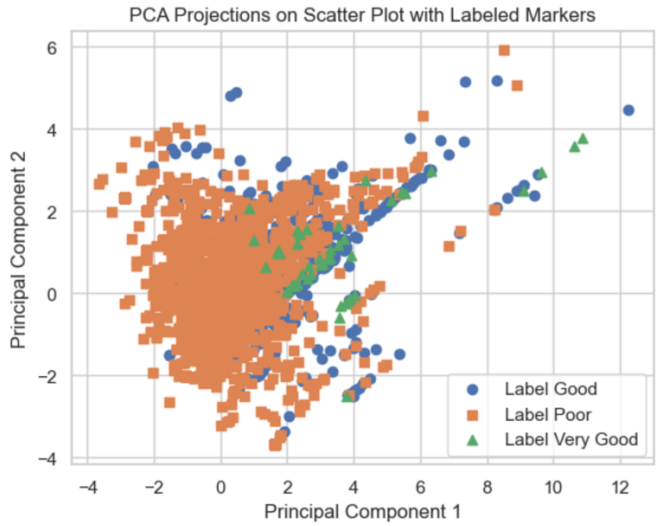


Select the Principal Components (PCs) - This step is like the one we performed while using price as the label feature.

According to the cumulative variance plot given above the top 5 principal components need to be chosen to capture more than 70% of the cumulative explained variance. This means that we can preserve 70% of the relevant information from original dataset by reducing the dimension from 9 to 5. We can also reduce the dimension to 4 at the cost of reducing the amount of relevant information kept. The total variance gets reduced to approximately 65% by choosing k=4 for our principal component analysis.

Project the data onto Principal Components –

Project the data onto the selected principal components to obtain the new feature space. For demonstrative purposes the plot of the projection of datapoints on PC1 and PC2 (the first 2 principal components / dominant eigenvectors) is given below.



Like the price label, from the above plot we can see that not much relation can be made out of the 3 labelled categories. This is because from the earlier Cumulative Variance of Eigen Vectors plot it is evident that the first 2 principal components only retain approximately 40% of the total variance of the original information.

Interesting aspects regarding the Data based on Eigenvalue and Eigenvector Analysis of the Covariance Matrix

The weights corresponding to each feature in the top 5 principal components (that preserve 70% of total variance) are given below.

Features	PC1	PC2	PC3	PC4	PC5
Winery	0.076768	0.352535	-0.517196	0.234140	-0.288215
Wine	0.050651	-0.401444	0.219310	0.599674	-0.509326
Year	-0.475509	-0.309495	-0.116545	-0.350909	-0.170444
Num_reviews	-0.263288	-0.219948	-0.359023	0.518174	0.609178
Region	0.389442	-0.311685	-0.289540	0.153163	-0.130464
Price	0.555833	0.247766	0.121337	-0.008470	0.028953
Type	0.157172	-0.545125	-0.159643	-0.332415	-0.159128
Body	0.368923	-0.168551	-0.468241	-0.242920	0.196430
acidity	-0.273293	0.293753	-0.445690	-0.029803	-0.419051

The features price, region, body, type, winery and wine (ordered in descending order of their influence) are positively correlated to PC1 while the features year, acidity, and num_reviews (ordered in descending order of their influence) are negatively correlated to PC1.

Similarly, we can see the correlation of each feature on principal components 2, 3, 4 and 5.

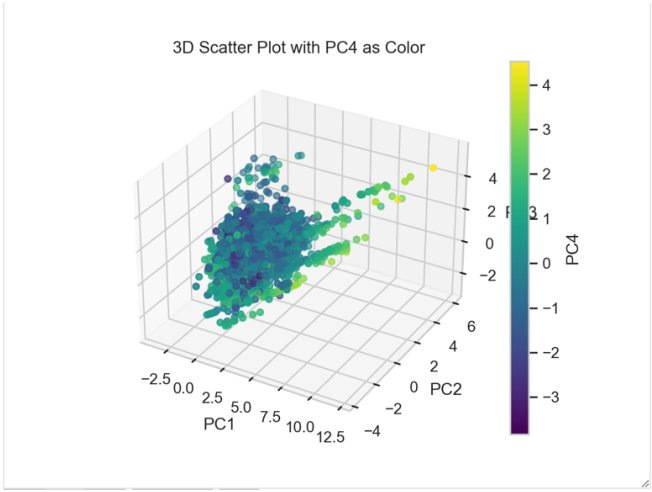
VISUALIZE THE RESULTS

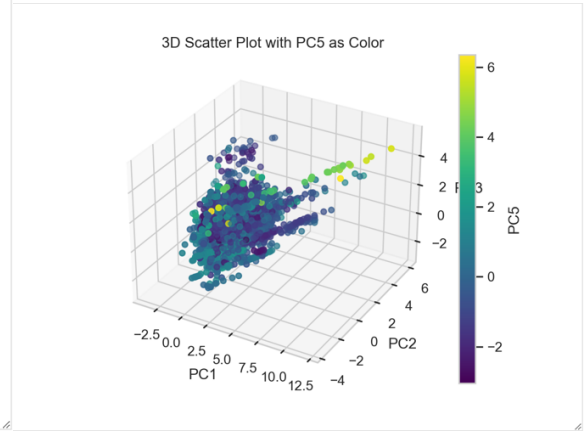
Price feature as the label

Visualizing five principal components directly on 2D or 3D graph can be challenging due to the limitation of visualizing in a lower-dimensional space. Hence, we use a combination of techniques to provide insight into the behavior of the principal components.

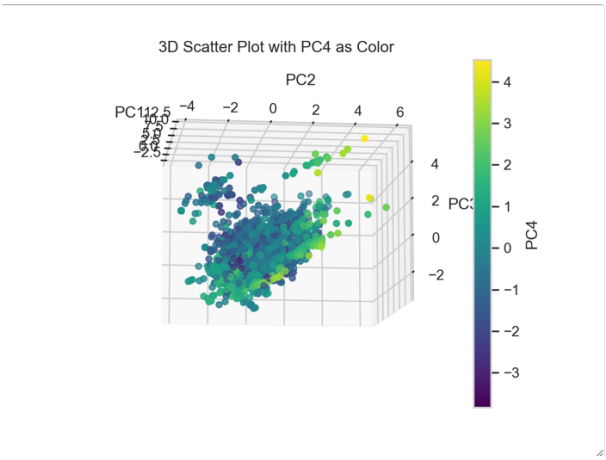
1. **3D Scatter Plot**

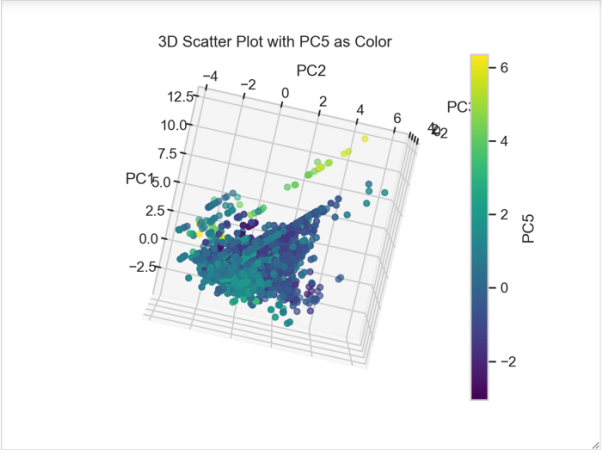
   Two 3D Scatter plots are created showing the first 3 principal components on the x, y, and z axes. The fourth and fifth components are represented using colour markers which then adds an additional dimension to the plot.





These plots can then be rotated using an interactive environment like jupyter notebook to view the generated plot in different angles. This would then help us gain a comprehensive understanding of the data structure. Given below are the screenshots of the same plots as given above in a different angle after rotation.

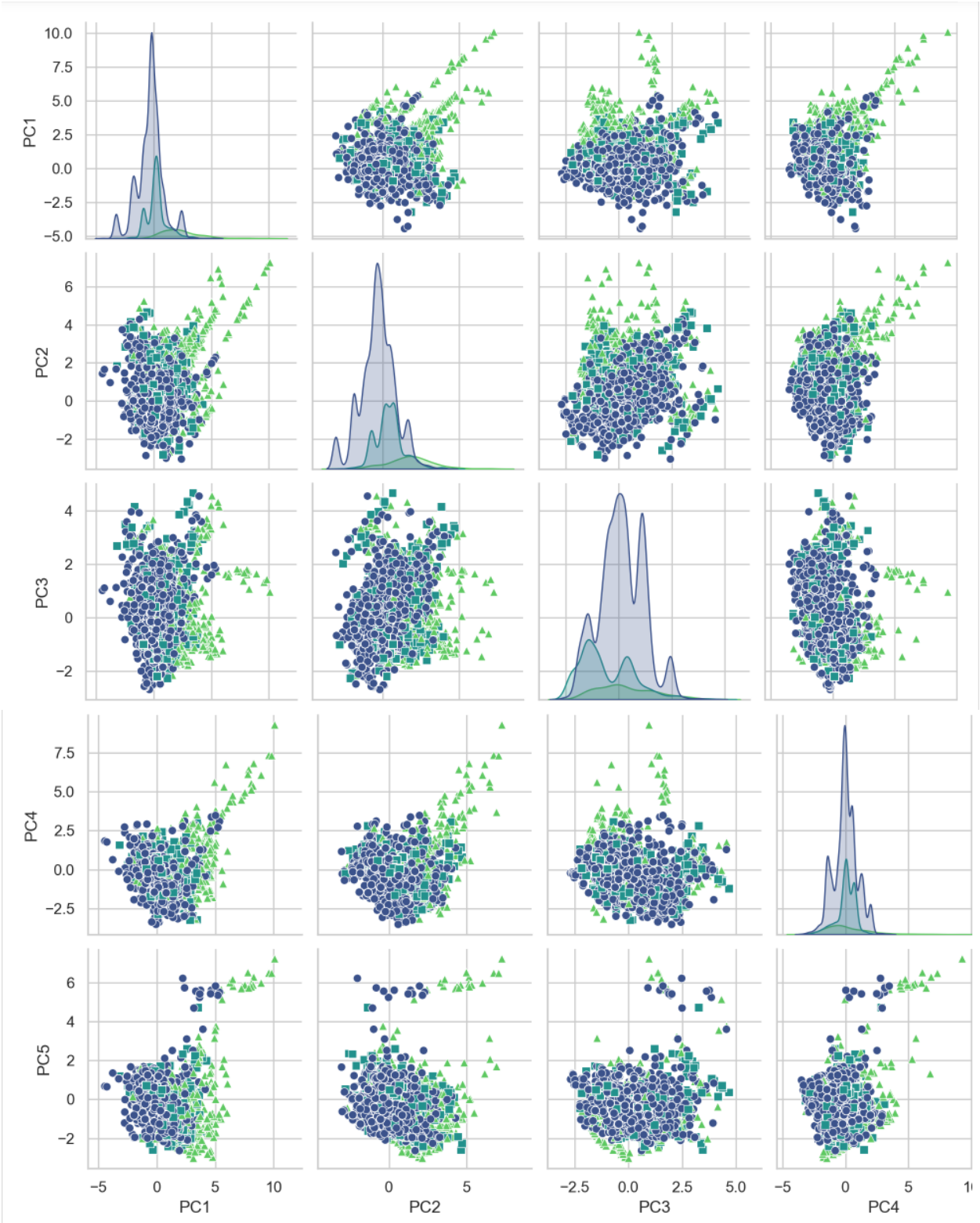




The second angle helps us distinguish the different colored data points better.

2D Scatter Plots (Pairs Plot)

Creating multiple 2D scatter plots can help visualize the relationship between 2 principal components at a time. Given below are the 2D scatter plots for 5 principal components. In the below graph the markers = ['o', 's', '^'] in blue, dark green and yellowish-green represent the 3 categories of prices ‘Cheap’, ‘Medium’ and ‘Expensive’.


PC1
PC2
PC3
PC4
PC5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
6
4
2
0
−2
−5
0
5
10
−2.5
0.0
2.5
5.0

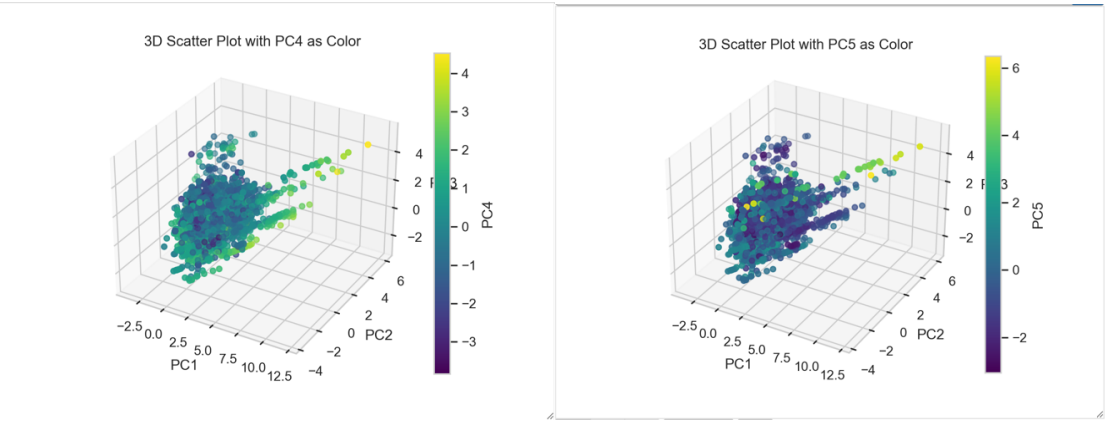
Observation

- In the pair plots, we can see a clear grouping of clusters for the ‘Cheap’ and ‘Medium’ classes indicated by the colors blue and dark-green.
- Few yellowish-green markers seem to be deviating a bit from the main clusters suggesting they might be outliers.

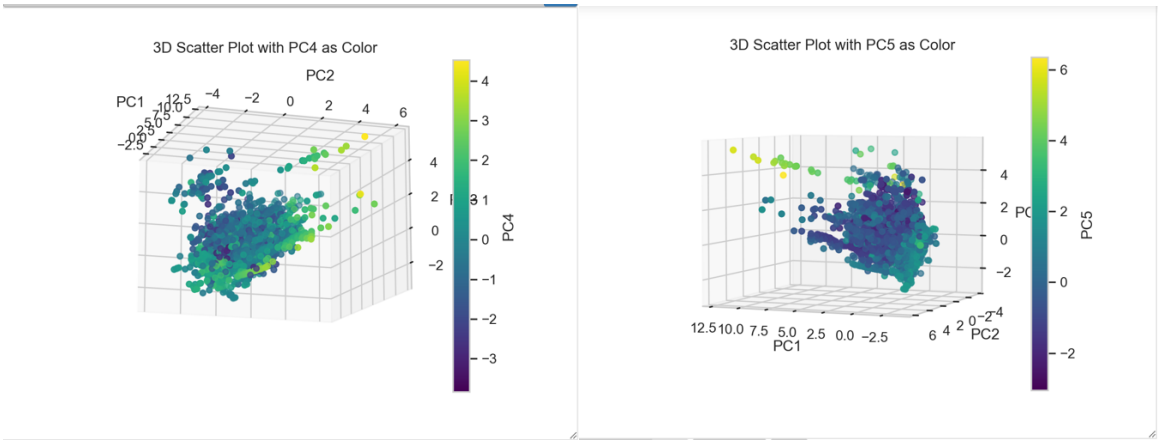
Rating feature as the label

1. 3D Scatter Plot

Given below is the 3D scatter plot of the principle components of the data when rating is taken as the target label



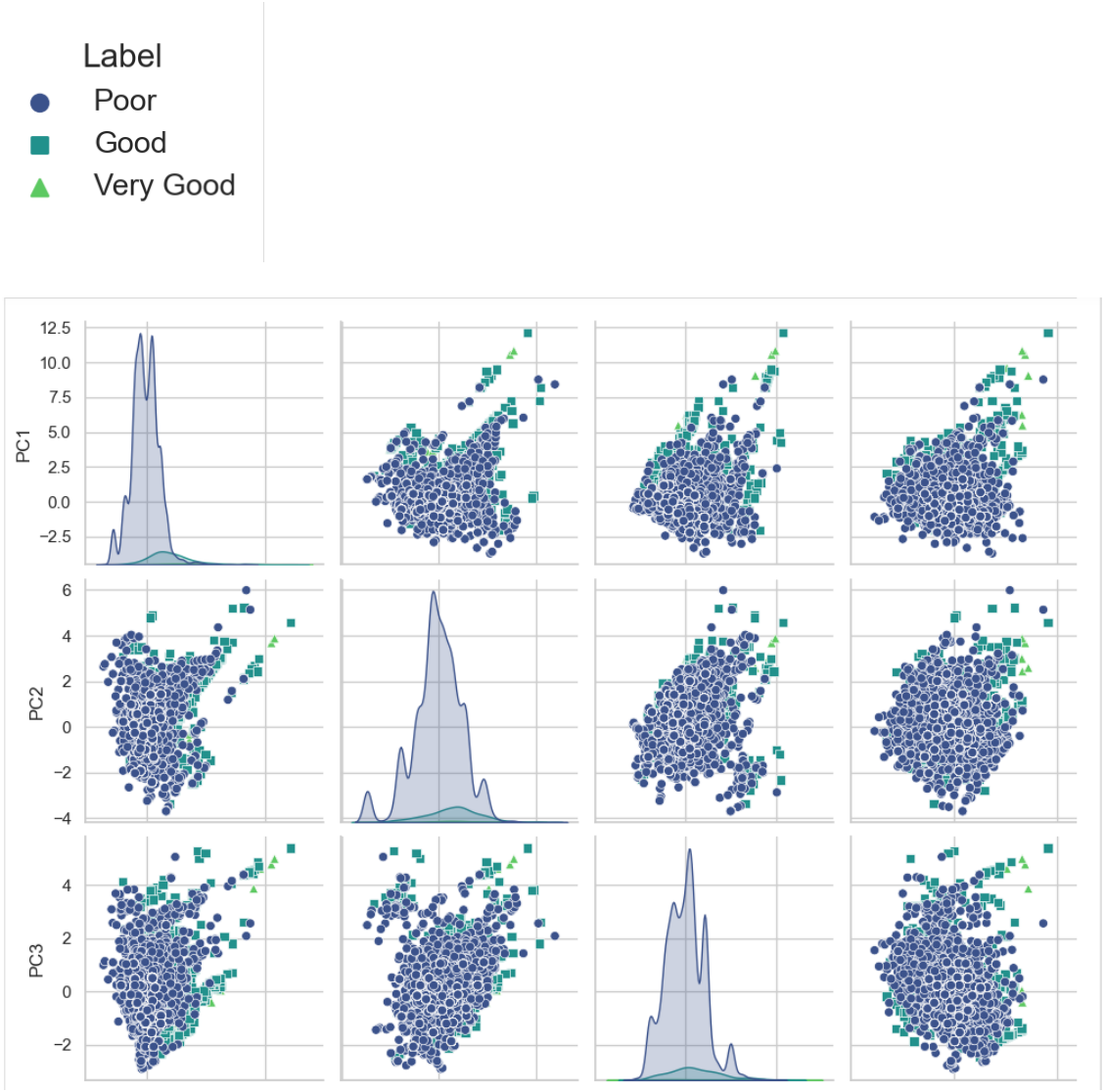
Given below are the screenshots of the same plots as given above in a different angle after rotation.

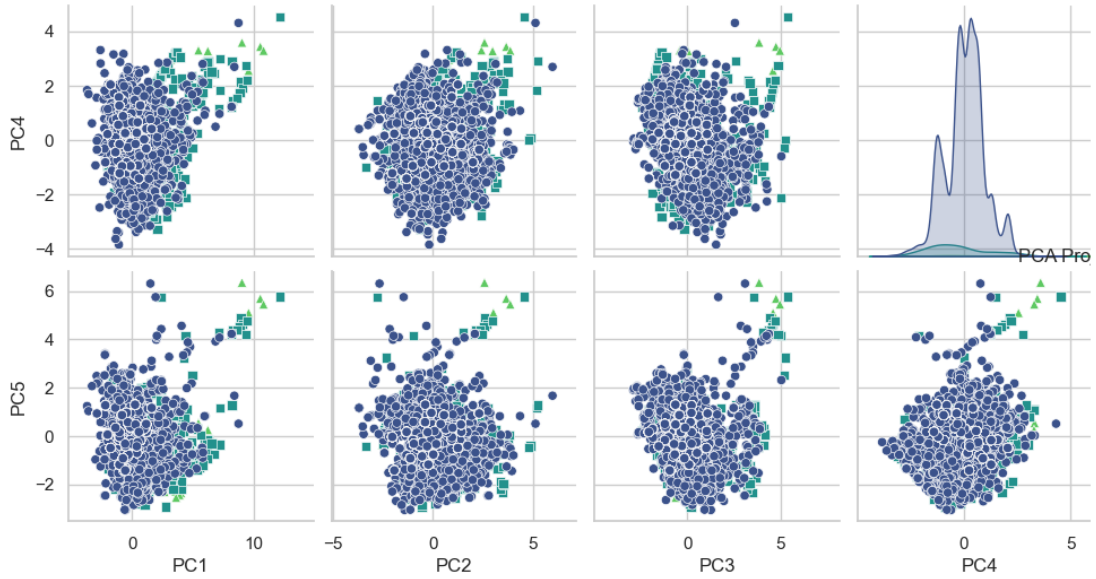


The second angle helps us distinguish the different colored data points better.

2. 2D Scatter Plots (Pairs Plot)

Given below are the 2D scatter plots for 5 principal components of data when rating feature is taken as the target label. In the below graph the markers = ['o', 's', '^'] in blue, dark green and yellowish green represent the 3 categories of ratings ‘Poor, ‘Good and ‘Very Good’.





Observation

- In the pair plots, we can see a clear clustering of ‘Poor’ and ‘Good’ rating labels indicated by the blue and dark-green markers.
- Few dark-green and yellowish-green markers seem to be deviating a bit from the main clusters suggesting they might be outliers.