Matheus Vinicius Ferreira Figueiredo Teixeira

301236904

# Exercise 1

## Research and investigate the LIBSVM format, in your analysis report define the format and show an example with explanation.

The LIBSVM format is a text-based data format designed for representing labeled feature vectors, primarily used in support vector machines and other machine learning applications. Each line in a LIBSVM file represents a single instance and follows this structure:

<label> <index1>:<value1> <index2>:<value2> ... <indexN>:<valueN>

No empty lines are allowed in the middle. The indexes must be positive integers in ascending order, but 0 values can be omitted (sparce representation). The feature pairs are space-separated. And, for binary classification, normally +1 and -1 are used.

Consider the following example:

+1 1:0.708 2:1.0 4:0.5

-1 1:0.583 3:1.0 4:0.237

+1 2:0.33 3:0.8 4:0.4

For the first line the label is +1, for the first index the value is 0.708, and the index 3 is omitted, meaning its value is 0.

## Carry out some basic investigation: count the number of records, count the number of columns print the inferred schema and explain what each column contains and **record the results in your analysis report.**

As it can be seen, there are **100 records** and only **2 columns**. The first column is a column of doubles that holds the label information. The second column is a vector column that holds the features.

root

|-- label: double (nullable = true)

|-- features: vector (nullable = true)

A screenshot of a computer

Description automatically generated

## Printout the following:

**Also note the results in your written response.**

### Name of input column

features

### Name of output column

indexedFeatures\_matheus

### # of features

692

### Map of categories

{645: {0.0: 0}, 69: {0.0: 0}, 365: {0.0: 0}, 138: {0.0: 0}, 479: {0.0: 0}, 333: {0.0: 0}, 249: {0.0: 0}, 0: {0.0: 0}, 666: {0.0: 0, 10.0: 1}, 88: {0.0: 0}, 170: {0.0: 0}, 115: {0.0: 0}, 276: {0.0: 0, 3.0: 1, 153.0: 2, 252.0: 3}, 308: {0.0: 0}, 5: {0.0: 0}, 449: {0.0: 0}, 120: {0.0: 0, 253.0: 1}, 614: {0.0: 0, 140.0: 1}, 677: {0.0: 0}, 202: {0.0: 0, 13.0: 1, 44.0: 2, 87.0: 3}, 10: {0.0: 0}, 56: {0.0: 0}, 533: {0.0: 0}, 142: {0.0: 0}, 340: {0.0: 0}, 670: {0.0: 0}, 174: {0.0: 0, 175.0: 1}, 42: {0.0: 0}, 417: {0.0: 0}, 24: {0.0: 0}, 37: {0.0: 0}, 25: {0.0: 0}, 257: {0.0: 0, 73.0: 1, 120.0: 2}, 389: {0.0: 0}, 52: {0.0: 0}, 14: {0.0: 0}, 504: {0.0: 0}, 110: {0.0: 0}, 587: {0.0: 0}, 619: {0.0: 0}, 196: {0.0: 0}, 559: {0.0: 0}, 638: {0.0: 0, 1.0: 1, 29.0: 2, 137.0: 3}, 20: {0.0: 0}, 421: {0.0: 0}, 46: {0.0: 0}, 93: {0.0: 0}, 284: {0.0: 0}, 228: {0.0: 0}, 448: {0.0: 0}, 57: {0.0: 0}, 78: {0.0: 0}, 29: {0.0: 0}, 475: {0.0: 0}, 164: {0.0: 0, 14.0: 1}, 591: {0.0: 0}, 646: {0.0: 0}, 253: {0.0: 0}, 106: {0.0: 0}, 121: {0.0: 0, 63.0: 1, 132.0: 2}, 84: {0.0: 0}, 147: {0.0: 0, 241.0: 1}, 280: {0.0: 0}, 61: {0.0: 0}, 221: {0.0: 0}, 396: {0.0: 0, 19.0: 1}, 89: {0.0: 0}, 133: {0.0: 0, 9.0: 1, 18.0: 2, 52.0: 3}, 116: {0.0: 0}, 1: {0.0: 0}, 507: {0.0: 0}, 312: {0.0: 0}, 74: {0.0: 0}, 307: {0.0: 0}, 452: {0.0: 0, 24.0: 1, 29.0: 2}, 6: {0.0: 0}, 248: {0.0: 0, 13.0: 1, 250.0: 2}, 60: {0.0: 0}, 117: {0.0: 0}, 678: {0.0: 0, 37.0: 1, 40.0: 2}, 529: {0.0: 0}, 85: {0.0: 0}, 201: {0.0: 0}, 220: {0.0: 0, 250.0: 1}, 366: {0.0: 0}, 534: {0.0: 0}, 102: {0.0: 0, 5.0: 1, 72.0: 2}, 334: {0.0: 0}, 28: {0.0: 0}, 38: {0.0: 0}, 561: {0.0: 0}, 392: {0.0: 0}, 70: {0.0: 0}, 424: {0.0: 0, 5.0: 1, 29.0: 2}, 192: {0.0: 0, 146.0: 1}, 21: {0.0: 0}, 137: {0.0: 0}, 165: {0.0: 0}, 33: {0.0: 0}, 92: {0.0: 0}, 229: {0.0: 0, 23.0: 1}, 252: {0.0: 0}, 197: {0.0: 0}, 361: {0.0: 0}, 65: {0.0: 0}, 97: {0.0: 0, 64.0: 1, 121.0: 2}, 665: {0.0: 0, 25.0: 1, 71.0: 2, 173.0: 3}, 224: {0.0: 0}, 615: {0.0: 0}, 9: {0.0: 0}, 53: {0.0: 0}, 169: {0.0: 0}, 141: {0.0: 0}, 420: {0.0: 0}, 109: {0.0: 0}, 256: {0.0: 0}, 225: {0.0: 0}, 339: {0.0: 0}, 77: {0.0: 0}, 193: {0.0: 0}, 669: {0.0: 0}, 476: {0.0: 0}, 642: {0.0: 0}, 590: {0.0: 0}, 679: {0.0: 0, 239.0: 1, 251.0: 2}, 96: {0.0: 0, 247.0: 1}, 393: {0.0: 0}, 647: {0.0: 0}, 173: {0.0: 0}, 13: {0.0: 0}, 41: {0.0: 0}, 503: {0.0: 0}, 134: {0.0: 0}, 73: {0.0: 0}, 105: {0.0: 0}, 2: {0.0: 0}, 311: {0.0: 0}, 558: {0.0: 0}, 674: {0.0: 0}, 530: {0.0: 0}, 586: {0.0: 0}, 618: {0.0: 0}, 166: {0.0: 0}, 32: {0.0: 0}, 34: {0.0: 0}, 148: {0.0: 0, 71.0: 1, 251.0: 2}, 45: {0.0: 0}, 279: {0.0: 0}, 64: {0.0: 0}, 17: {0.0: 0}, 584: {0.0: 0}, 562: {0.0: 0}, 423: {0.0: 0}, 191: {0.0: 0, 250.0: 1}, 22: {0.0: 0}, 44: {0.0: 0}, 59: {0.0: 0}, 118: {0.0: 0}, 281: {0.0: 0}, 27: {0.0: 0}, 641: {0.0: 0}, 71: {0.0: 0}, 391: {0.0: 0}, 12: {0.0: 0}, 445: {0.0: 0}, 54: {0.0: 0}, 611: {0.0: 0, 19.0: 1, 20.0: 2, 29.0: 3}, 144: {0.0: 0}, 49: {0.0: 0}, 335: {0.0: 0}, 86: {0.0: 0}, 672: {0.0: 0}, 172: {0.0: 0}, 113: {0.0: 0}, 219: {0.0: 0, 18.0: 1, 20.0: 2, 250.0: 3}, 419: {0.0: 0}, 81: {0.0: 0}, 362: {0.0: 0}, 451: {0.0: 0}, 76: {0.0: 0}, 7: {0.0: 0}, 39: {0.0: 0}, 649: {0.0: 0, 83.0: 1}, 98: {0.0: 0, 70.0: 1, 191.0: 2}, 616: {0.0: 0}, 477: {0.0: 0}, 367: {0.0: 0}, 535: {0.0: 0}, 103: {0.0: 0}, 140: {0.0: 0}, 621: {0.0: 0, 82.0: 1, 236.0: 2}, 91: {0.0: 0}, 66: {0.0: 0}, 251: {0.0: 0}, 668: {0.0: 0}, 198: {0.0: 0}, 108: {0.0: 0}, 278: {0.0: 0}, 223: {0.0: 0}, 394: {0.0: 0}, 306: {0.0: 0}, 135: {0.0: 0}, 563: {0.0: 0}, 226: {0.0: 0}, 3: {0.0: 0}, 505: {0.0: 0}, 80: {0.0: 0}, 167: {0.0: 0}, 35: {0.0: 0}, 473: {0.0: 0}, 675: {0.0: 0}, 589: {0.0: 0}, 531: {0.0: 0}, 255: {0.0: 0}, 648: {0.0: 0}, 112: {0.0: 0}, 617: {0.0: 0}, 194: {0.0: 0}, 145: {0.0: 0}, 48: {0.0: 0}, 557: {0.0: 0}, 63: {0.0: 0}, 640: {0.0: 0}, 18: {0.0: 0}, 282: {0.0: 0}, 95: {0.0: 0, 56.0: 1}, 310: {0.0: 0}, 50: {0.0: 0}, 67: {0.0: 0}, 199: {0.0: 0}, 673: {0.0: 0}, 16: {0.0: 0}, 585: {0.0: 0}, 502: {0.0: 0}, 338: {0.0: 0}, 643: {0.0: 0}, 31: {0.0: 0}, 336: {0.0: 0}, 613: {0.0: 0}, 11: {0.0: 0}, 72: {0.0: 0}, 446: {0.0: 0}, 612: {0.0: 0}, 143: {0.0: 0}, 43: {0.0: 0}, 250: {0.0: 0}, 450: {0.0: 0}, 99: {0.0: 0, 70.0: 1, 166.0: 2, 255.0: 3}, 363: {0.0: 0}, 87: {0.0: 0}, 671: {0.0: 0}, 104: {0.0: 0}, 368: {0.0: 0}, 588: {0.0: 0}, 40: {0.0: 0}, 26: {0.0: 0}, 390: {0.0: 0}, 55: {0.0: 0}, 114: {0.0: 0}, 171: {0.0: 0}, 139: {0.0: 0}, 418: {0.0: 0}, 23: {0.0: 0}, 8: {0.0: 0}, 75: {0.0: 0}, 119: {0.0: 0, 85.0: 1}, 58: {0.0: 0}, 667: {0.0: 0}, 478: {0.0: 0}, 82: {0.0: 0}, 620: {0.0: 0, 62.0: 1}, 447: {0.0: 0}, 36: {0.0: 0}, 168: {0.0: 0}, 146: {0.0: 0, 82.0: 1}, 30: {0.0: 0}, 51: {0.0: 0}, 19: {0.0: 0}, 422: {0.0: 0}, 564: {0.0: 0, 9.0: 1, 20.0: 2, 73.0: 3}, 305: {0.0: 0}, 107: {0.0: 0}, 4: {0.0: 0}, 136: {0.0: 0}, 506: {0.0: 0}, 79: {0.0: 0}, 195: {0.0: 0}, 474: {0.0: 0}, 532: {0.0: 0}, 94: {0.0: 0}, 283: {0.0: 0}, 395: {0.0: 0}, 644: {0.0: 0}, 47: {0.0: 0}, 15: {0.0: 0}, 163: {0.0: 0, 85.0: 1}, 200: {0.0: 0}, 68: {0.0: 0}, 62: {0.0: 0}, 277: {0.0: 0}, 691: {0.0: 0, 36.0: 1, 73.0: 2}, 501: {0.0: 0}, 90: {0.0: 0}, 111: {0.0: 0}, 254: {0.0: 0}, 227: {0.0: 0}, 337: {0.0: 0}, 83: {0.0: 0}, 309: {0.0: 0}, 560: {0.0: 0}, 639: {0.0: 0}, 676: {0.0: 0}, 222: {0.0: 0}, 592: {0.0: 0, 73.0: 1}, 364: {0.0: 0}}

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Print the schema of the predictions and **note the results into your analysis report.**

root

|-- label: double (nullable = true)

|-- features: vector (nullable = true)

|-- indexedLabel\_matheus: double (nullable = false)

|-- indexedFeatures\_matheus: vector (nullable = true)

|-- rawPrediction: vector (nullable = true)

|-- probability: vector (nullable = true)

|-- prediction: double (nullable = false)

A screenshot of a computer

Description automatically generated

## Print the accuracy of your model and the test error and **note the results in your analysis report.**

**Accuracy:** 0.9487179487179487

**Test Error:** 0.05128205128205132

A screenshot of a computer

Description automatically generated

## Show the first 10 predictions with the actual labels and features **take a screenshot and add it to your analysis report.**

A screenshot of a computer

Description automatically generated

# Exercise 2

## Using spark high level api functions (i.e. not pandas), carry out some initial investigation and **record the results in your analysis,** at minimum provide the following:

### Printout the names of columns

['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol', 'quality']

### Printout the types of each column

[('fixed acidity', 'double'), ('volatile acidity', 'double'), ('citric acid', 'double'), ('residual sugar', 'double'), ('chlorides', 'double'), ('free sulfur dioxide', 'double'), ('total sulfur dioxide', 'double'), ('density', 'double'), ('pH', 'double'), ('sulphates', 'double'), ('alcohol', 'double'), ('quality', 'int')]

### Printout the basic statistics mean, median, the four quartiles

+-------+------------------+-------------------+-------------------+------------------+--------------------+-------------------+

|summary| fixed acidity| volatile acidity| citric acid| residual sugar| chlorides|free sulfur dioxide|

+-------+------------------+-------------------+-------------------+------------------+--------------------+-------------------+

| count| 1599| 1599| 1599| 1599| 1599| 1599|

| mean| 8.319637273295838| 0.5278205128205131| 0.2709756097560964|2.5388055034396517| 0.08746654158849257| 15.874921826141339|

| stddev|1.7410963181276948|0.17905970415353525|0.19480113740531824| 1.40992805950728|0.047065302010090085| 10.46015696980971|

| 25%| 7.1| 0.39| 0.09| 1.9| 0.07| 7.0|

| 50%| 7.9| 0.52| 0.26| 2.2| 0.079| 14.0|

| 75%| 9.2| 0.64| 0.42| 2.6| 0.09| 21.0|

+-------+------------------+-------------------+-------------------+------------------+--------------------+-------------------+

+-------+--------------------+--------------------+-------------------+------------------+------------------+------------------+

|summary|total sulfur dioxide| density| pH| sulphates| alcohol| quality|

+-------+--------------------+--------------------+-------------------+------------------+------------------+------------------+

| count| 1599| 1599| 1599| 1599| 1599| 1599|

| mean| 46.46779237023139| 0.9967466791744831| 3.311113195747343|0.6581488430268921|10.422983114446502|5.6360225140712945|

| stddev| 32.89532447829907|0.001887333953842...|0.15438646490354271|0.1695069795901101|1.0656675818473935|0.8075694397347051|

| 25%| 22.0| 0.9956| 3.21| 0.55| 9.5| 5|

| 50%| 38.0| 0.99675| 3.31| 0.62| 10.2| 6|

| 75%| 62.0| 0.99784| 3.4| 0.73| 11.1| 6|

+-------+--------------------+--------------------+-------------------+------------------+------------------+------------------+

### Printout the minimum, maximum value for each column

+-------+------------------+-------------------+-------------------+------------------+--------------------+-------------------+

|summary| fixed acidity| volatile acidity| citric acid| residual sugar| chlorides|free sulfur dioxide|

+-------+------------------+-------------------+-------------------+------------------+--------------------+-------------------+

| min| 4.6| 0.12| 0.0| 0.9| 0.012| 1.0|

| max| 15.9| 1.58| 1.0| 15.5| 0.611| 72.0|

+-------+------------------+-------------------+-------------------+------------------+--------------------+-------------------+

+-------+--------------------+--------------------+-------------------+------------------+------------------+------------------+

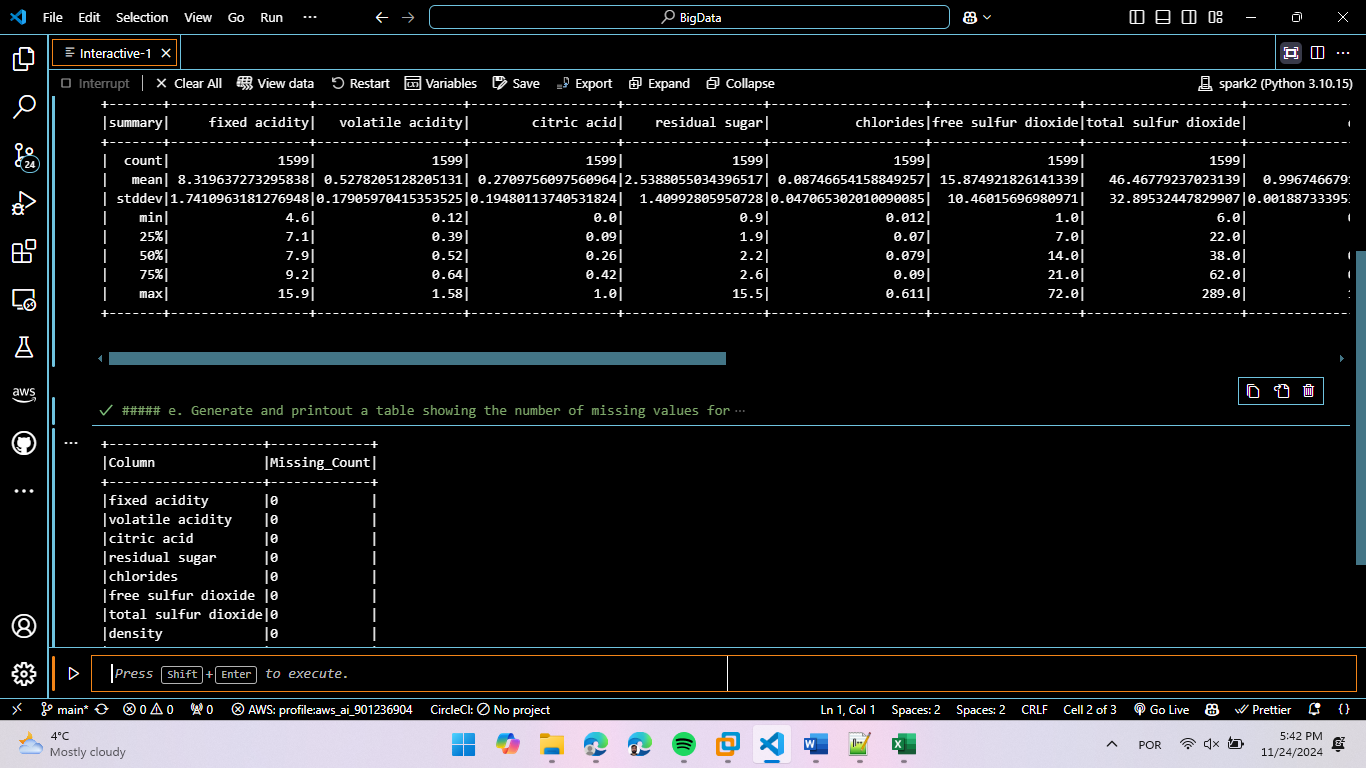
|summary|total sulfur dioxide| density| pH| sulphates| alcohol| quality|

+-------+--------------------+--------------------+-------------------+------------------+------------------+------------------+

| min| 6.0| 0.99007| 2.74| 0.33| 8.4| 3|

| max| 289.0| 1.00369| 4.01| 2.0| 14.9| 8|

+-------+--------------------+--------------------+-------------------+------------------+------------------+------------------+



A screenshot of a computer

Description automatically generated

### Generate and printout a table showing the number of missing values for each column. (*Hint: use isnan, when, count, col)*

A screenshot of a computer

Description automatically generated+--------------------+-------------+

|Column |Missing\_Count|

+--------------------+-------------+

|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 |

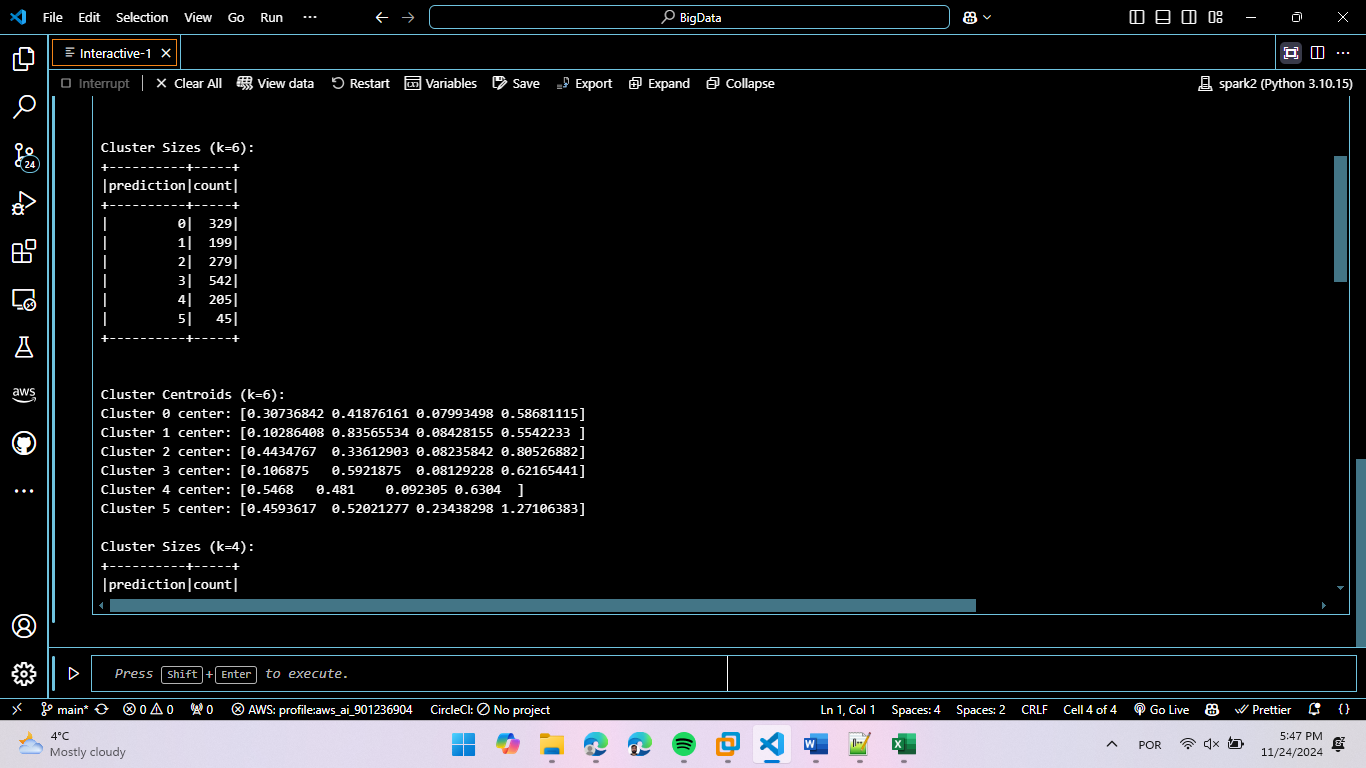
|sulphates |0 |

|alcohol |0 |

|quality |0 |

+--------------------+-------------+

## Print the cluster sizes and the cluster centroids, **record the results in your analysis report and write some conclusions.**



Cluster Sizes (k=6):

+----------+-----+

|prediction|count|

+----------+-----+

| 0| 329|

| 1| 199|

| 2| 279|

| 3| 542|

| 4| 205|

| 5| 45|

+----------+-----+

Cluster Centroids (k=6):

Cluster 0 center: [0.30736842 0.41876161 0.07993498 0.58681115]

Cluster 1 center: [0.10286408 0.83565534 0.08428155 0.5542233 ]

Cluster 2 center: [0.4434767 0.33612903 0.08235842 0.80526882]

Cluster 3 center: [0.106875 0.5921875 0.08129228 0.62165441]

Cluster 4 center: [0.5468 0.481 0.092305 0.6304 ]

Cluster 5 center: [0.4593617 0.52021277 0.23438298 1.27106383]

From the counts we can see that cluster 3 has 542 wines, and the other clusters have irregular number of wines with cluster 5 only holding 45 wines.

To derive meaningful conclusions about wine quality patterns, I conducted a cluster analysis using k=6 clusters. The analysis examined both the raw counts and percentage distribution of wines across different quality levels within each cluster. The findings are presented in the following tables

Analysis for k=6 clusters:

Table 1: Counts of wine per quality in each cluster

+------------------+---+---+---+---+---+---+

|prediction\_quality| 3| 4| 5| 6| 7| 8|

+------------------+---+---+---+---+---+---+

| 0| 0| 9|140|145| 31| 4|

| 1| 7| 22|107| 57| 5| 1|

| 2| 1| 2| 52|124| 93| 7|

| 3| 0| 14|273|218| 35| 2|

| 4| 2| 4| 85| 79| 31| 4|

| 5| 0| 2| 24| 15| 4| 0|

+------------------+---+---+---+---+---+---+

Table 2: Percentage of wine per quality in each cluster

+------------------+---+---+---+---+---+---+

|prediction\_quality| 3| 4| 5| 6| 7| 8|

+------------------+---+---+---+---+---+---+

| 0| 0%|16%|20%|22%|15%|22%|

| 1|70%|41%|15%| 8%| 2%| 5%|

| 2|10%| 3%| 7%|19%|46%|38%|

| 3| 0%|26%|40%|34%|17%|11%|

| 4|20%| 7%|12%|12%|15%|22%|

| 5| 0%| 3%| 3%| 2%| 2%| 0%|

+------------------+---+---+---+---+---+---+

It is important to note that the dataset exhibits significant class imbalance. The majority of wines are concentrated in the middle quality ranges, particularly qualities 5 and 6, while extreme qualities - especially 3 and 8 - have very few examples. This imbalance should be considered when interpreting the clustering results, as it may influence the model's ability to identify patterns in the underrepresented quality levels.

Analysis of the percentage distribution reveals clear patterns in how wines of different qualities cluster together. The highest quality wines, those rated 8, show a strong presence in cluster 2, with 38% of these wines falling into this group. Following this pattern through the quality levels, we observe that the majority of quality 7 wines also appear in cluster 2 (46%), quality 6 and 5 wines predominantly appear in cluster 3 (34% and 40% respectively), and quality 4 and 3 wines are most concentrated in cluster 1 (41% and 70% respectively).

This clustering pattern suggests that wines of similar quality levels share common characteristics. Specifically, wines rated 7 and 8 exhibit very similar properties, as evidenced by their concentration in cluster 2. A similar relationship exists between wines rated 5 and 6, which predominantly group together in cluster 3. The lower-quality wines, rated 3 and 4, also display similar characteristics, clustering together in cluster 1.

It's noteworthy that clusters 0, 4, and particularly cluster 5, contain relatively few examples across all quality levels. This distribution suggests that these clusters might be capturing outliers or wines with unique characteristics that don't align with the main quality-based patterns.

Based on these observations, it appears that reducing the number of clusters from 6 to 4 could provide a more efficient and meaningful categorization. This reduction would align with the natural groupings we've observed: one cluster for high-quality wines (7-8), another for medium-quality wines (5-6), a third for lower-quality wines (3-4), and a fourth cluster to capture wines that don't fit into these main quality-based patterns. This simplified clustering structure would maintain the key quality-based distinctions while potentially providing a more robust and interpretable model.

The analysis demonstrates that while wine quality exists on a continuous scale, there are distinct characteristic groupings that align with quality levels, suggesting that certain combinations of wine attributes are consistently associated with specific quality ranges.

## Repeat steps 8&9 but set the number of k to 4.

Cluster Sizes (k=4):

+----------+-----+

|prediction|count|

+----------+-----+

| 0| 545|

| 1| 464|

| 2| 521|

| 3| 69|

+----------+-----+

Cluster Centroids (k=4):

Cluster 0 center: [0.47537615 0.38266055 0.08455963 0.69636697]

Cluster 1 center: [0.07926724 0.72628233 0.08409914 0.58349138]

Cluster 2 center: [0.20543186 0.50873321 0.08009021 0.61445298]

Cluster 3 center: [0.44057971 0.48391304 0.18876812 1.18826087]

Now we can see a much better distribution among the clusters. Clusters 0, 1, and 2 have similar amounts of wines, while only cluster 3 have a very lower number of exemplars, which is consistent with our previous conclusions

We proceeded with a reduced number of clusters (k=4) to evaluate if this would provide a more efficient categorization of wines based on their quality levels. The results are presented in the following tables:

Analysis for k=4 clusters:

Table 1: Counts of wine per quality in each cluster:

+------------------+---+---+---+---+---+---+

|prediction\_quality| 3| 4| 5| 6| 7| 8|

+------------------+---+---+---+---+---+---+

| 0| 3| 9|149|240|132| 12|

| 1| 7| 31|250|152| 23| 1|

| 2| 0| 10|248|225| 34| 4|

| 3| 0| 3| 34| 21| 10| 1|

+------------------+---+---+---+---+---+---+

Table 2: Percentage of wine per quality in each cluster:

+------------------+---+---+---+---+---+---+

|prediction\_quality| 3| 4| 5| 6| 7| 8|

+------------------+---+---+---+---+---+---+

| 0|30%|16%|21%|37%|66%|66%|

| 1|70%|58%|36%|23%|11%| 5%|

| 2| 0%|18%|36%|35%|17%|22%|

| 3| 0%| 5%| 4%| 3%| 5%| 5%|

+------------------+---+---+---+---+---+---+

The results from the 4-cluster analysis reveal some interesting patterns that support our earlier hypothesis. Cluster 0 has emerged as the dominant cluster for high-quality wines, capturing 66% of both quality 7 and 8 wines. This strongly supports our earlier observation about the similarity between wines of these two quality levels.

Cluster 1 shows a clear affinity for lower-quality wines, containing 70% of quality 3 wines and 58% of quality 4 wines. This pattern gradually decreases as wine quality increases, dropping to only 5% for quality 8 wines, creating a clear gradient that suggests this cluster captures characteristics associated with lower-quality wines.

Cluster 2 appears to be most associated with medium-quality wines, containing similar proportions of quality 5 and 6 wines (36% and 35% respectively). This cluster shows lower representation at both the higher and lower ends of the quality spectrum, suggesting it captures characteristics typical of middle-range wines.

Cluster 3 has consistently low representation across all quality levels (3-5% for most qualities), suggesting it might be capturing outliers or wines with unique characteristics that don't align with the typical patterns for their quality level.

This 4-cluster solution appears to have successfully captured the main quality-based groupings we anticipated: high-quality wines (predominantly in cluster 0), lower-quality wines (cluster 1), and medium-quality wines (cluster 2), with an additional cluster (3) for outliers. The clearer separation between quality levels in this clustering solution, particularly for high-quality wines, suggests that reducing the number of clusters to 4 has indeed provided a more efficient and interpretable model for understanding the relationship between wine characteristics and quality ratings.