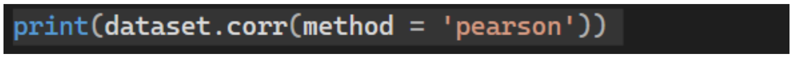
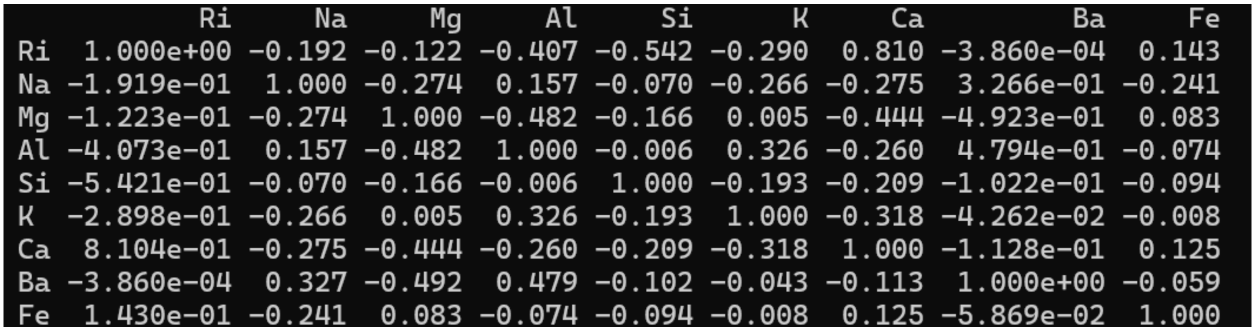
Part 2 Analysis

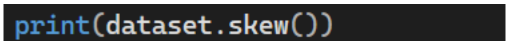
Listing 4: Pairwise Pearson Correlation, Skew for Each Attribute, Univariate Density Plot, Correlation Matrix Plot (correlation matrix must be normalized).



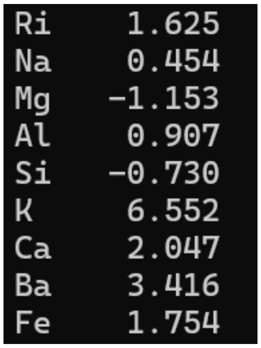
```
print(dataset.corr(method = 'pearson'))
```



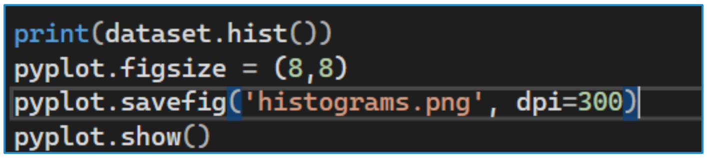
```
            Ri     Na     Mg     Al     Si      K     Ca         Ba     Fe
Ri  1.000e+00 -0.192 -0.122 -0.407 -0.542 -0.290  0.810 -3.860e-04  0.143
Na -1.919e-01  1.000 -0.274  0.157 -0.070 -0.266 -0.275  3.266e-01 -0.241
Mg -1.223e-01 -0.274  1.000 -0.482 -0.166  0.005 -0.444 -4.923e-01  0.083
Al -4.073e-01  0.157 -0.482  1.000 -0.006  0.326 -0.260  4.794e-01 -0.074
Si -5.421e-01 -0.070 -0.166 -0.006  1.000 -0.193 -0.209 -1.022e-01 -0.094
K  -2.898e-01 -0.266  0.005  0.326 -0.193  1.000 -0.318 -4.262e-02 -0.008
Ca  8.104e-01 -0.275 -0.444 -0.260 -0.209 -0.318  1.000 -1.128e-01  0.125
Ba -3.860e-04  0.327 -0.492  0.479 -0.102 -0.043 -0.113  1.000e+00 -0.059
Fe  1.430e-01 -0.241  0.083 -0.074 -0.094 -0.008  0.125 -5.869e-02  1.000
```



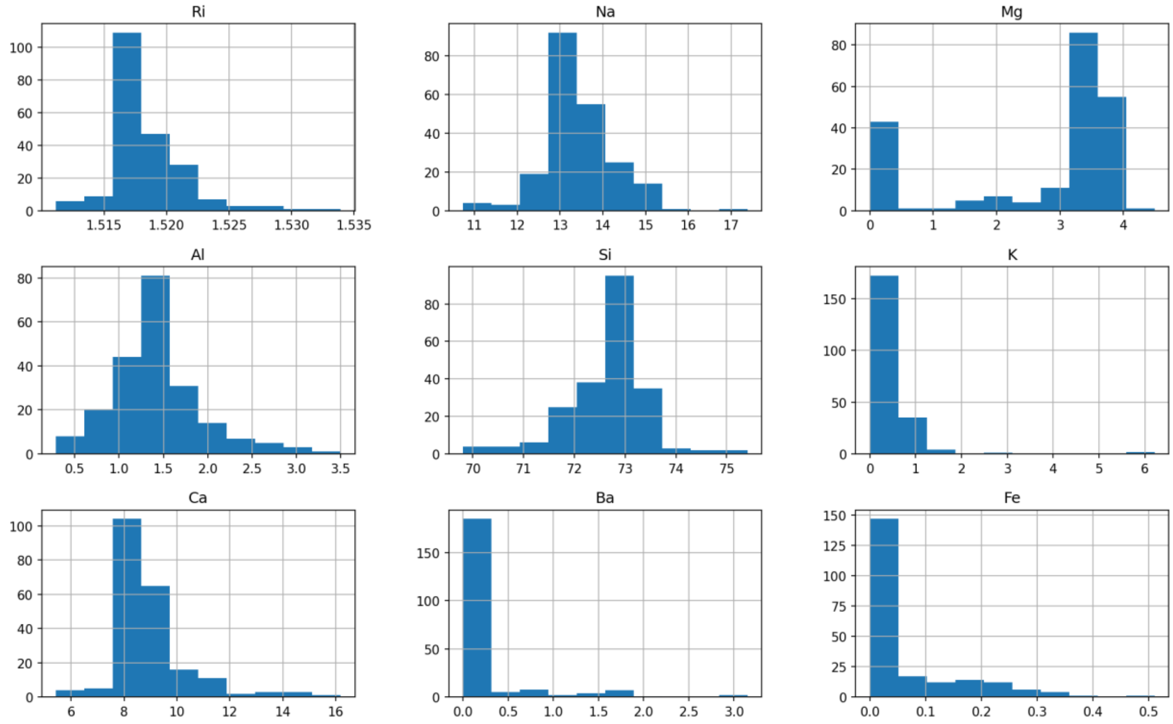
```
print(dataset.skew())
```

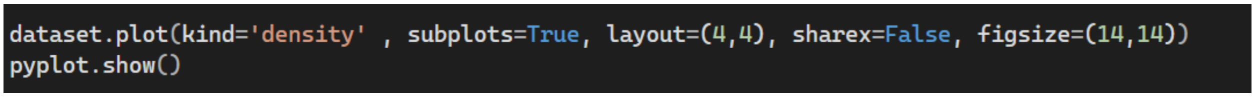


```
Ri     1.625
Na     0.454
Mg    -1.153
Al     0.907
Si    -0.730
K      6.552
Ca     2.047
Ba     3.416
Fe     1.754
```

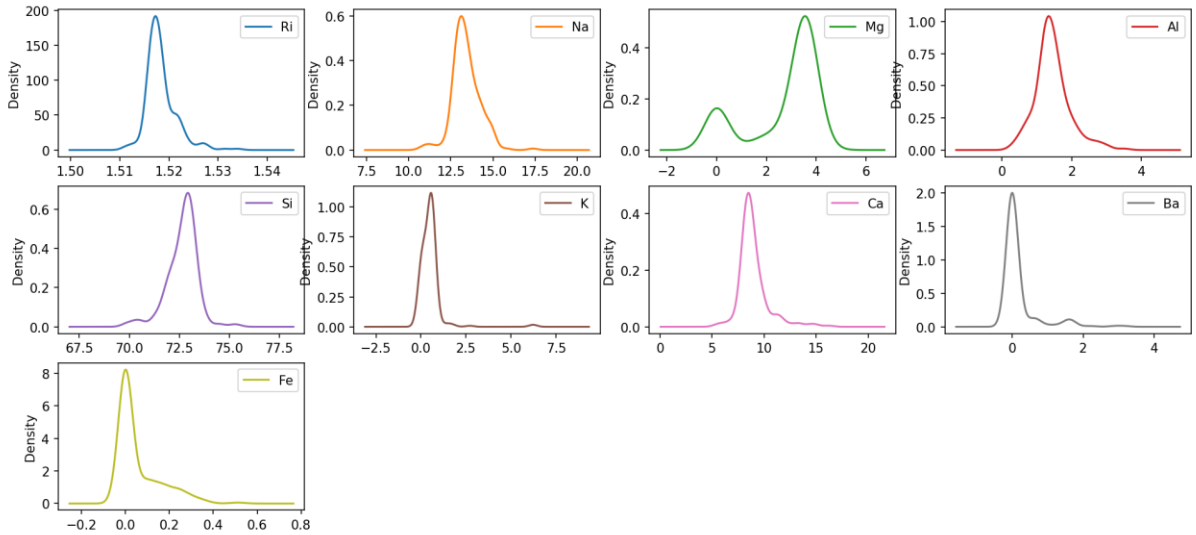


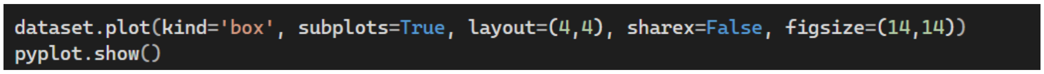
```
print(dataset.hist())
pyplot.figsize = (8,8)
pyplot.savefig('histograms.png', dpi=300)
pyplot.show()
```



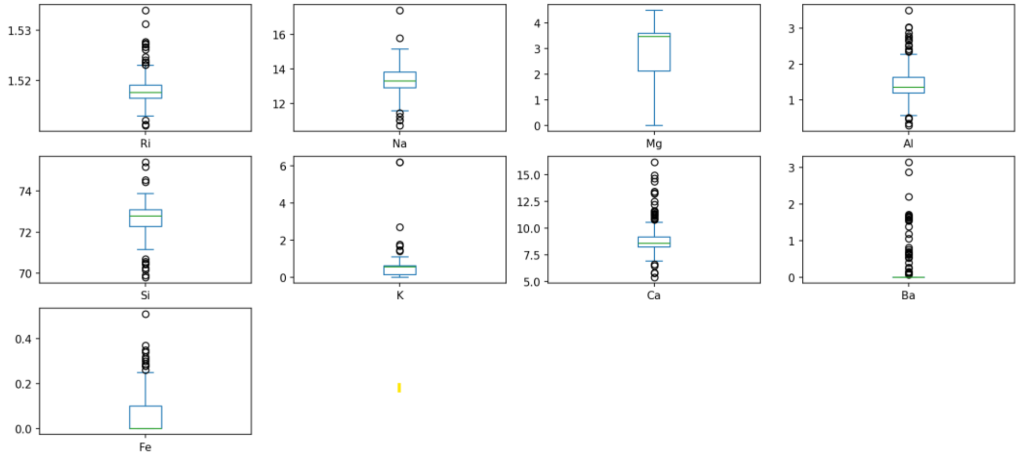


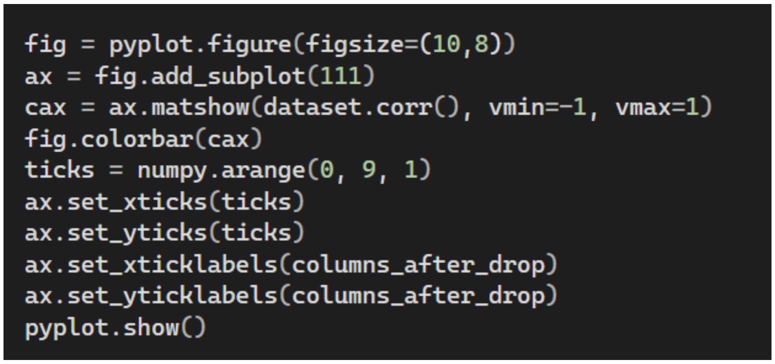
```
dataset.plot(kind='density' , subplots=True, layout=(4,4), sharex=False, figsize=(14,14))
pyplot.show()
```



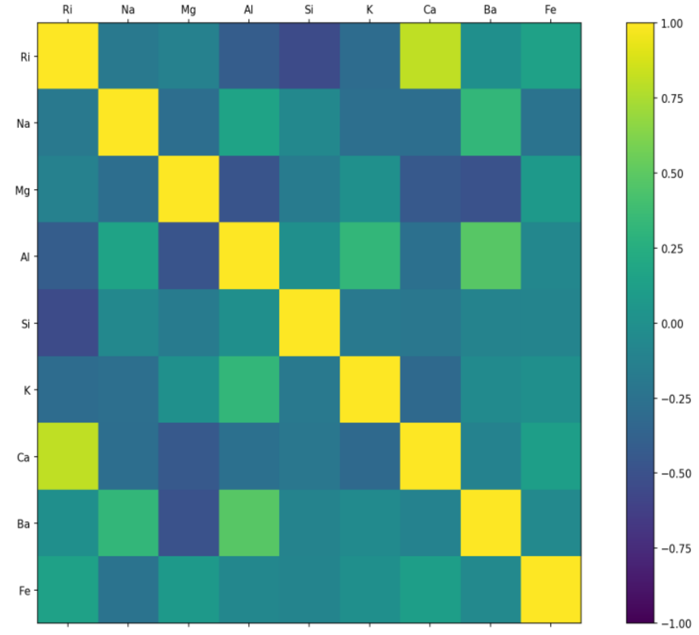


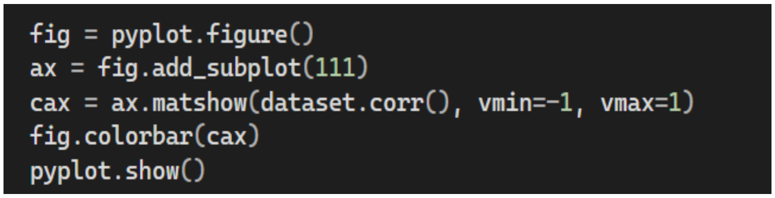
```
dataset.plot(kind='box', subplots=True, layout=(4,4), sharex=False, figsize=(14,14))
pyplot.show()
```



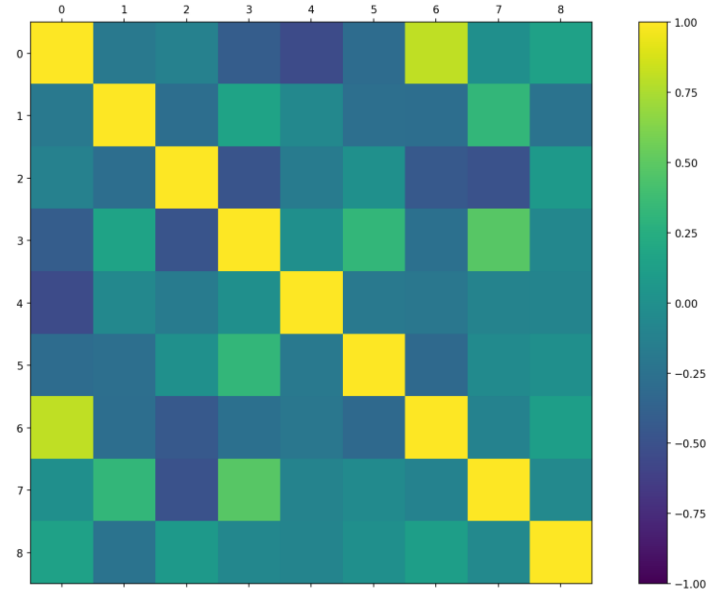


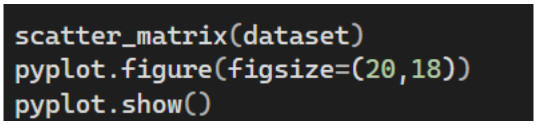
```
fig = pyplot.figure(figsize=(10,8))
ax = fig.add_subplot(111)
cax = ax.matshow(dataset.corr(), vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = numpy.arange(0, 9, 1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(columns_after_drop)
ax.set_yticklabels(columns_after_drop)
pyplot.show()
```



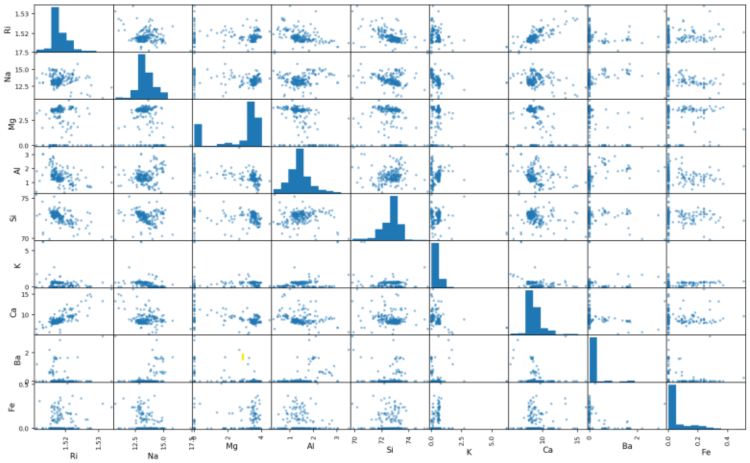


```
fig = pyplot.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(dataset.corr(), vmin=-1, vmax=1)
fig.colorbar(cax)
pyplot.show()
```





```
scatter_matrix(dataset)
pyplot.figure(figsize=(20,18))
pyplot.show()
```



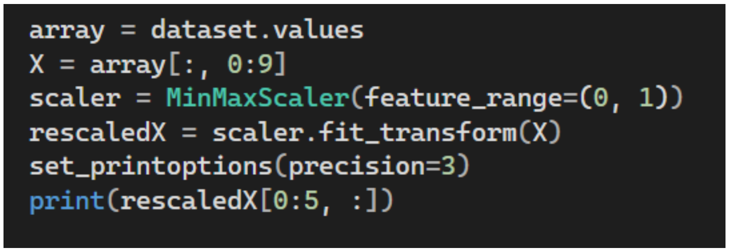
Our analysis of the chemical properties of the glass samples revealed some interesting patterns. First, through the Pearson correlation, we found that Calcium (Ca) and Rhodium (Rh) have a strong positive correlation, suggesting that higher calcium levels tend to increase rhodium presence in the glass. On the flip side, Aluminum (Al) has a negative correlation with Rhodium, meaning that more aluminum seems to reduce rhodium levels. Potassium (K) and Sodium (Na) also show negative correlations with Rhodium, which might point to interactions that are worth exploring in future research. Meanwhile, elements like Barium (Ba) and Iron (Fe) seem to have little influence on Rhodium, showing minimal correlations.

Looking at the skewness of the data, Potassium stands out with a high skewness value of 6.552, suggesting that some samples have particularly high concentrations of Potassium. Barium and Calcium also show positive skewness, meaning a few samples have much higher levels than the rest. Sodium and Aluminum, on the other hand, have distributions that are more balanced, though still slightly skewed, while Magnesium has a negative skew, suggesting most samples have low Magnesium concentrations.

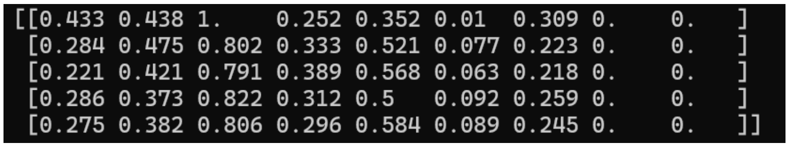
When visualizing these distributions with histograms, we confirmed the skewed nature of Potassium, Barium, and Calcium. Sodium and Aluminum appeared closer to normal distributions but still slightly off, while Silicon and Iron showed signs of a bimodal distribution, which could indicate different types of glass compositions. We also spotted some notable outliers in the data for Potassium and Barium, which could be related to special types of glass with unique properties.

The correlation matrix helped us see the bigger picture in terms of how the chemical elements relate to each other. For example, Calcium and the refractive index showed a strong positive relationship, while Silicon and the refractive index had almost no correlation, meaning they don’t really influence each other. The scatter matrix highlighted similar trends, with some clear outliers, particularly between Barium and Iron. The scatter plots also showed some variables, like Calcium and the refractive index, following a positive trend, while others, like Silicon and Iron, were much more random, confirming what we saw in the correlation heatmap.

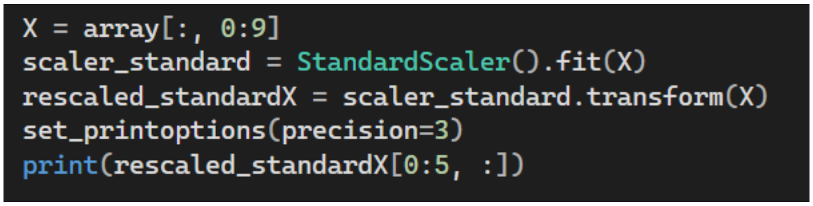
Listing 5: Rescaling Data, Standardize Data, Normalize Data, Binarization.



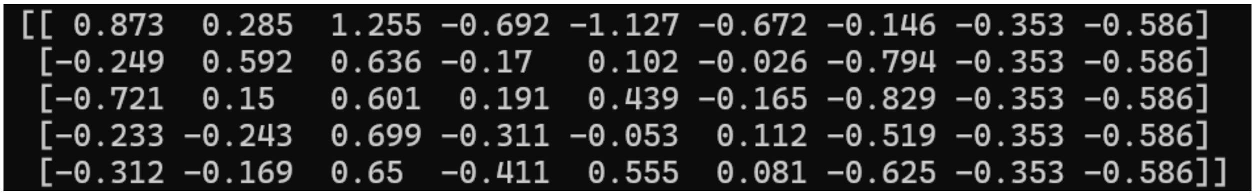
```
array = dataset.values
X = array[:, 0:9]
scaler = MinMaxScaler(feature_range=(0, 1))
rescaledX = scaler.fit_transform(X)
set_printoptions(precision=3)
print(rescaledX[0:5, :])
```



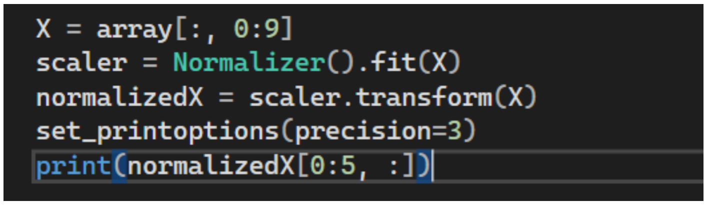
```
[[0.433 0.438 1.    0.252 0.352 0.01  0.309 0.    0.    ]
 [0.284 0.475 0.802 0.333 0.521 0.077 0.223 0.    0.    ]
 [0.221 0.421 0.791 0.389 0.568 0.063 0.218 0.    0.    ]
 [0.286 0.373 0.822 0.312 0.5   0.092 0.259 0.    0.    ]
 [0.275 0.382 0.806 0.296 0.584 0.089 0.245 0.    0.    ]]
```



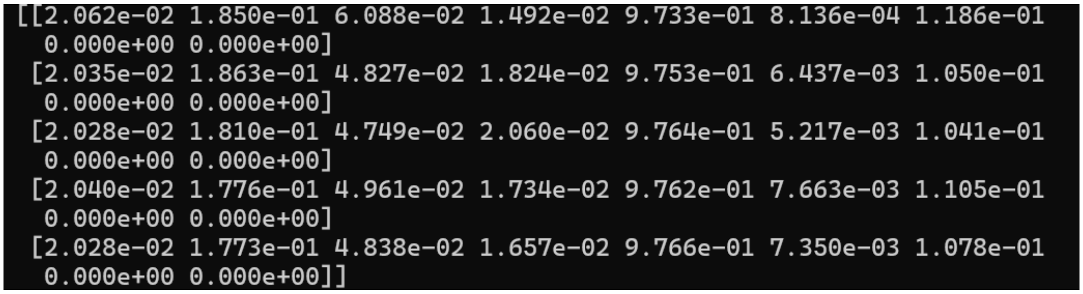
```
X = array[:, 0:9]
scaler_standard = StandardScaler().fit(X)
rescaled_standardX = scaler_standard.transform(X)
set_printoptions(precision=3)
print(rescaled_standardX[0:5, :])
```



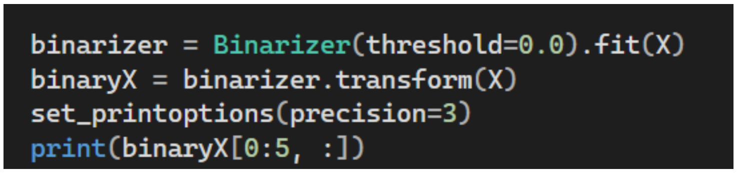
```
[[ 0.873  0.285  1.255 -0.692 -1.127 -0.672 -0.146 -0.353 -0.586]
 [-0.249  0.592  0.636 -0.17   0.102 -0.026 -0.794 -0.353 -0.586]
 [-0.721  0.15   0.601  0.191  0.439 -0.165 -0.829 -0.353 -0.586]
 [-0.233 -0.243  0.699 -0.311 -0.053  0.112 -0.519 -0.353 -0.586]
 [-0.312 -0.169  0.65  -0.411  0.555  0.081 -0.625 -0.353 -0.586]]
```



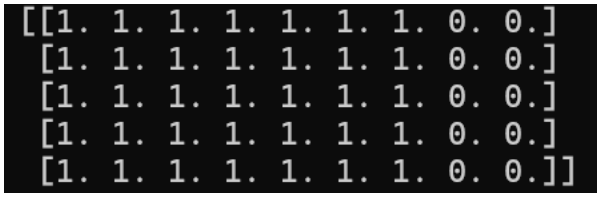
```
X = array[:, 0:9]
scaler = Normalizer().fit(X)
normalizedX = scaler.transform(X)
set_printoptions(precision=3)
print(normalizedX[0:5, :])
```



```
[[2.062e-02 1.850e-01 6.088e-02 1.492e-02 9.733e-01 8.136e-04 1.186e-01
  0.000e+00 0.000e+00]
 [2.035e-02 1.863e-01 4.827e-02 1.824e-02 9.753e-01 6.437e-03 1.050e-01
  0.000e+00 0.000e+00]
 [2.028e-02 1.810e-01 4.749e-02 2.060e-02 9.764e-01 5.217e-03 1.041e-01
  0.000e+00 0.000e+00]
 [2.040e-02 1.776e-01 4.961e-02 1.734e-02 9.762e-01 7.663e-03 1.105e-01
  0.000e+00 0.000e+00]
 [2.028e-02 1.773e-01 4.838e-02 1.657e-02 9.766e-01 7.350e-03 1.078e-01
  0.000e+00 0.000e+00]]
```



```
binarizer = Binarizer(threshold=0.0).fit(X)
binaryX = binarizer.transform(X)
set_printoptions(precision=3)
print(binaryX[0:5, :])
```



```
[[1. 1. 1. 1. 1. 1. 1. 0. 0.]
 [1. 1. 1. 1. 1. 1. 1. 0. 0.]
 [1. 1. 1. 1. 1. 1. 1. 0. 0.]
 [1. 1. 1. 1. 1. 1. 1. 0. 0.]
 [1. 1. 1. 1. 1. 1. 1. 0. 0.]]
```

In this step of the project, we focused on transforming the data to improve analysis. First, we used the MinMaxScaler to rescale the features between 0 and 1. This allowed us to see which features had the most variability. Elements like Sodium (Na), Magnesium (Mg), and Aluminum (Al) showed a lot of variation, which suggests they might hold more information and be worth focusing on. On the other hand, elements like Barium (Ba) and Iron (Fe) had very little variance, meaning they might not contribute as much to the overall trends in the data. Notably, Magnesium had values close to its maximum, indicating its significance in this dataset and potential for revealing interesting patterns if analyzed further.

Next, we standardized the data using the StandardScaler, which adjusts the features to have a mean of 0 and a standard deviation of 1. This step helps us better understand how each feature compares to the average. For instance, in one sample, a high value of 0.0873 for the refractive index suggests a significantly higher value compared to other samples. In contrast, negative values for Aluminum indicate that some samples have lower-than-average concentrations. These variations are important because they can influence properties like the viscosity and melting temperature of the glass. For example, elements like Magnesium and Aluminum typically improve glass durability.

After standardization, we moved on to normalizing the data, which ensures that the values of each feature are on a consistent scale. One key finding was that Potassium (K) had notably higher normalized values than the other elements, indicating its prominence in these glass samples. Other elements, like Magnesium, while present in moderate amounts, appear to play an essential role in the glass structure. Sodium (Na) and Silicon (Si), on the other hand, had lower levels, possibly suggesting they are less critical in this specific glass formulation. The absence of Calcium (Ca) stood out, hinting that this glass might be designed to be clear or free from certain impurities.

Finally, we applied a Binarizer to categorize the samples based on the presence or absence of specific chemicals. This simplified approach allowed us to get a quick overview of which elements are present across the samples. Any value greater than 0 was converted to 1, while values below 0 were converted to 0. This binary transformation is useful for identifying patterns quickly, such as which chemicals are consistently present in the glass compositions.