Glass Identification

for

Advanced Multivariate Statistics

by

Marticola No: 942091

Amanpreet singh

Università degli Studi di Milano

TABLE OF CONTENTS

Chapter 1

Glass Identification

1.1 Introduction

The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence.

1.1.1 Dataset:

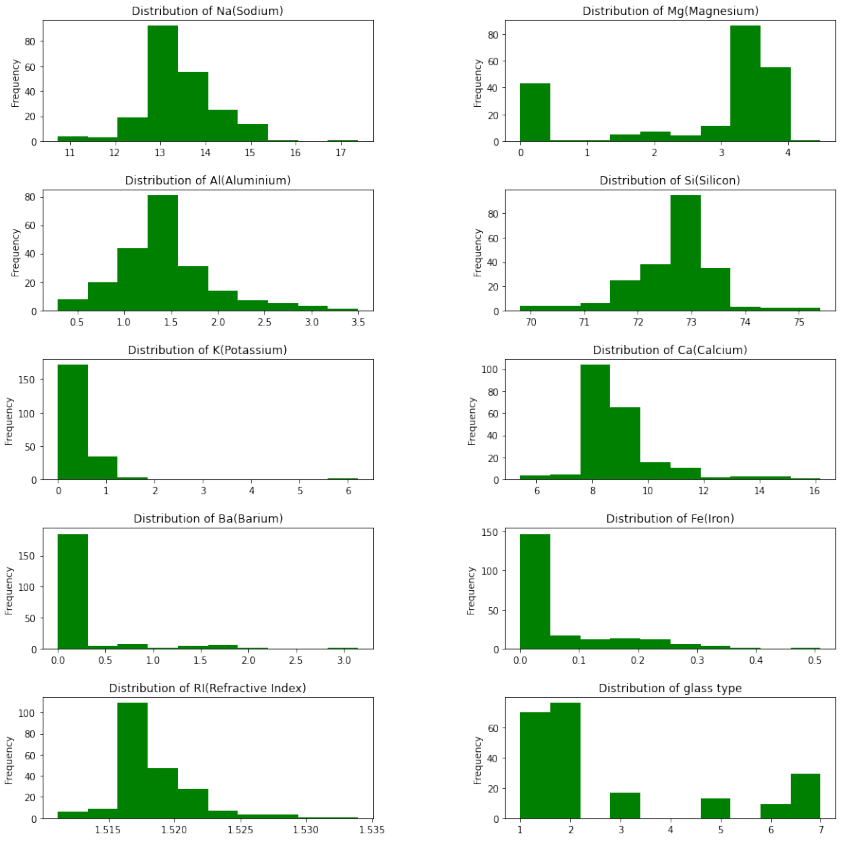
Variables
RI: refractive index
Na: Sodium
Mg: Magnesium
Al: Aluminum
Si: Silicon
K: Potassium
Ca: Calcium
Ba: Barium
Fe: Iron

Type of glass: Labels
1 building_windows_float_processed
2 building_windows_non_float_processed
3 vehicle_windows_float_processed
4 vehicle_windows_non_float_processed (none in this database)
5 containers
6 tableware
7 headlamps

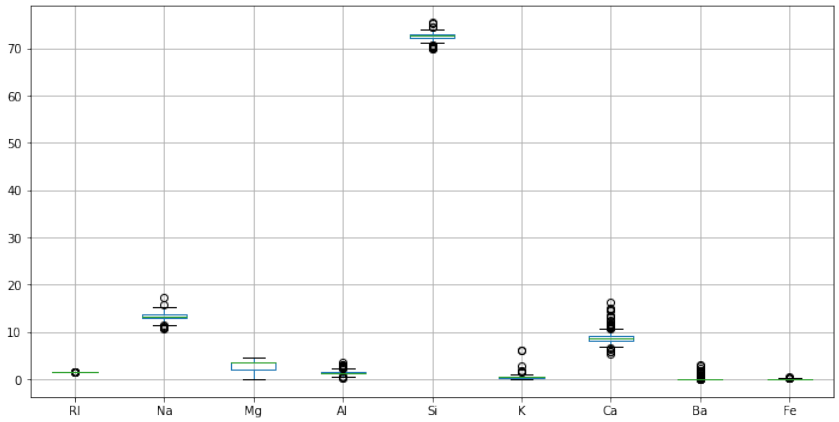
1.2 Exploratory Data Analysis

Distribution of Variables

None of the features are normally distributed and some have outliers

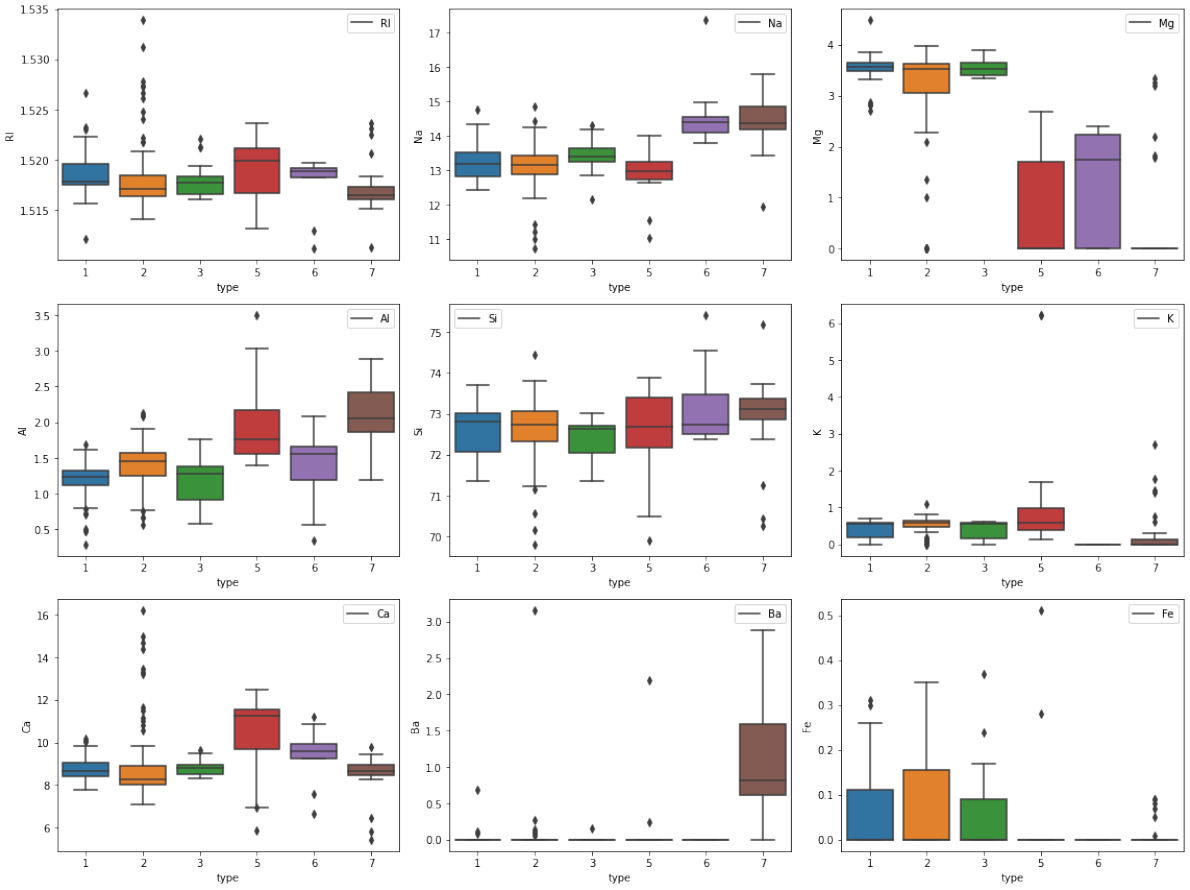


Silicon is the main component of Glass making more than 70% of composition
Combined Silicon, Sodium and Calcium make up around 90%
Iron is the least important component



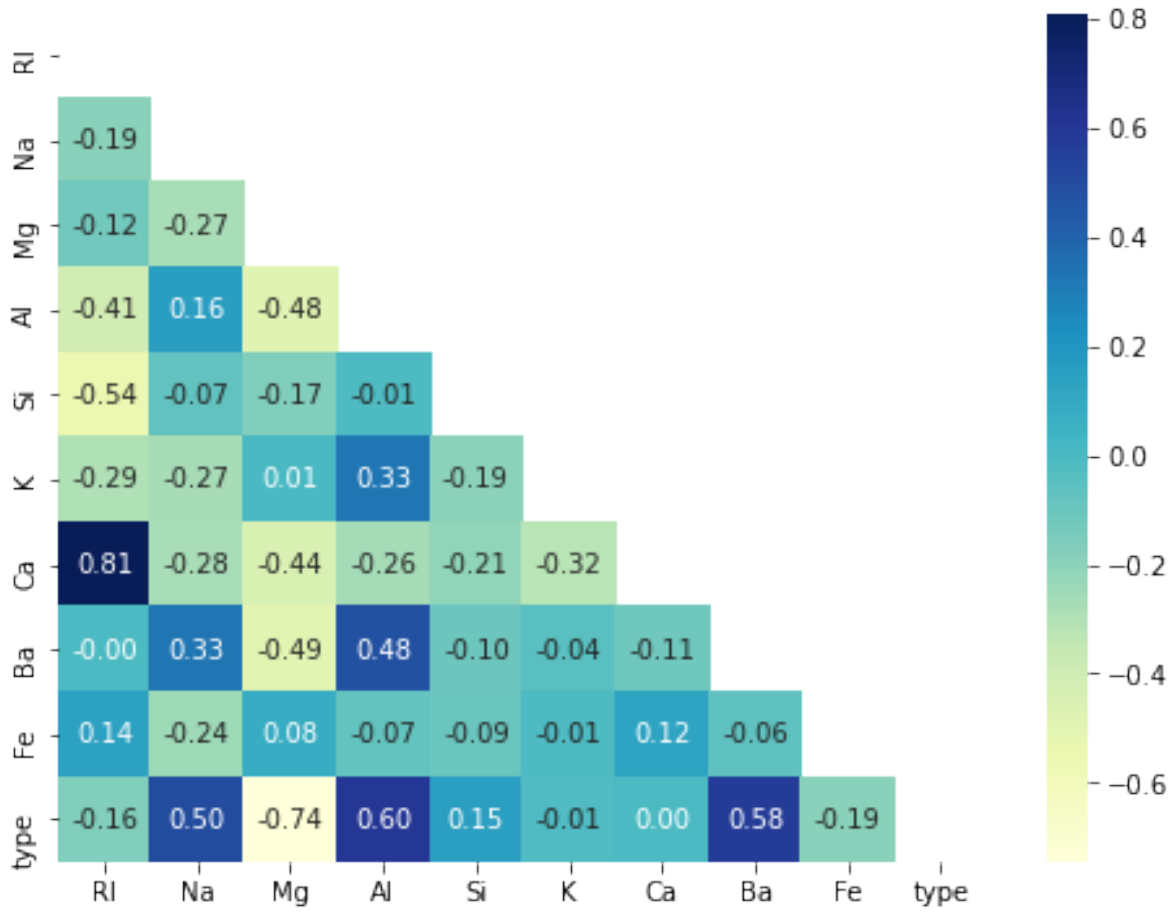
Distribution of A Variable in different Labels

Refractive index lies between 1.51 and 1.54
Type 6 and 7 have higher Na %
Type 1,2 and 3 have higher Mg %
Type 5 and 7 have higher Al %
Si % is similar in all types
Type 6 has no K composition
Type 5 and 6 have higher Ca composition
Ba is mostly used in Type 7
Fe is used in Type 1,2 and 3



Correlation

K and Ca have no correlation with labels (type), which means for some type it maybe high for some low causing cancelling effect,so need to do feature engineering ,for these two and do data smoothing



Feature Engineering

To improve K and Ca,I created Boolean for the rows where Ca is extremely high (more then 9) and K is extremely high (more then 0.7) and also where K is extremely low (less then 0.4)

```
data.groupby("type")["Ca"].mean()
```

```
type
1     8.797286
2     9.073684
3     8.782941
5    10.123846
6     9.356667
7     8.491379
Name: Ca, dtype: float64
```

```
[21] data.groupby("type")["K"].mean()
```

```
type
1    0.447429
2    0.521053
3    0.406471
5    1.470000
6    0.000000
7    0.325172
Name: K, dtype: float64
```

MANOVA Checking if combined effect of the new variables engineered has an effect on labels type or not with MANOVA

```
from statsmodels.multivariate.manova import MANOVA
maov=MANOVA.from_formula('K_morethandot7 + K_lessthandot4 + Ca_morethan9 ~ type', data=data)
print(maov.mv_test())
```

```
                  Multivariate linear model
============================================================

------------------------------------------------------------
       Intercept        Value  Num DF  Den DF  F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.9148 3.0000 210.0000  6.5184 0.0003
         Pillai's trace 0.0852 3.0000 210.0000  6.5184 0.0003
 Hotelling-Lawley trace 0.0931 3.0000 210.0000  6.5184 0.0003
    Roy's greatest root 0.0931 3.0000 210.0000  6.5184 0.0003
------------------------------------------------------------

------------------------------------------------------------
          type          Value  Num DF  Den DF  F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.7105 3.0000 210.0000 28.5233 0.0000
         Pillai's trace 0.2895 3.0000 210.0000 28.5233 0.0000
 Hotelling-Lawley trace 0.4075 3.0000 210.0000 28.5233 0.0000
    Roy's greatest root 0.4075 3.0000 210.0000 28.5233 0.0000
============================================================
```

The Wilks Lambda for type ,its Pvalue is quite significant,so we can say that mean values of new variables,combined do not have an cancelling effect and are different of types

ANOVA

Now to go further and check significance for each variable singularly, In order to check the relationship between variables and Label for new and old variables ,I performed Anova Test

H0 = There is no relationship between variable and Labels

H1 = There is relationship between variable and Labels

Checking P value for Anova Analysis

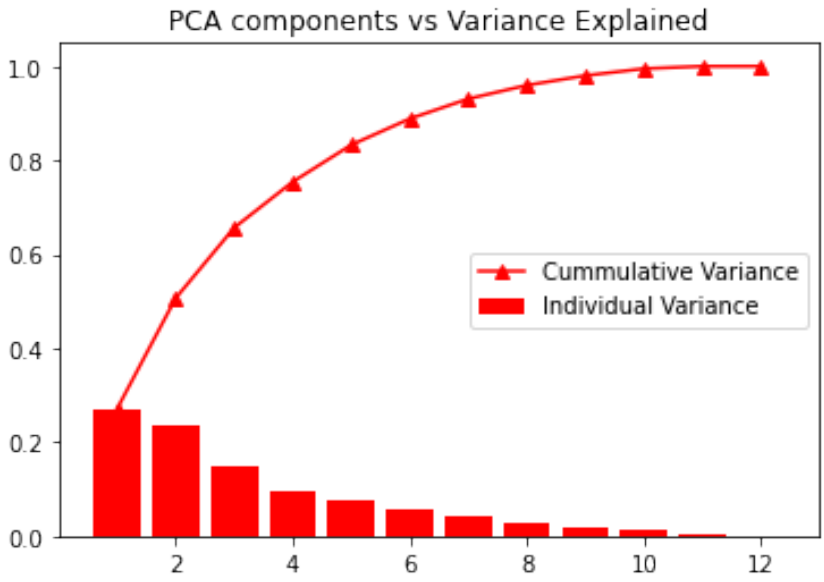
```
Pval for RI: 0.01617845580599427
Pval for Na: 4.061873356971206e-15
Pval for Mg: 3.8829946163472014e-39
Pval for Al: 3.26080946946565e-22
Pval for Si: 0.0266199101047075
Pval for K: 0.8837426923094087
Pval for Ca: 0.9889510387030452
Pval for Ba: 3.038430172779663e-20
Pval for Fe: 0.0057293003513817185
Pval for Ca_morethan9: 0.07403949750207843
Pval for K_morethandot7: 0.0001065753630029827
Pval for K_lessthandot4: 2.377474211882503e-09
```

K and Ca are not significant, but the new variables created from Feature Engineering are significant.

1.3 Methodology and Results

1.3.1

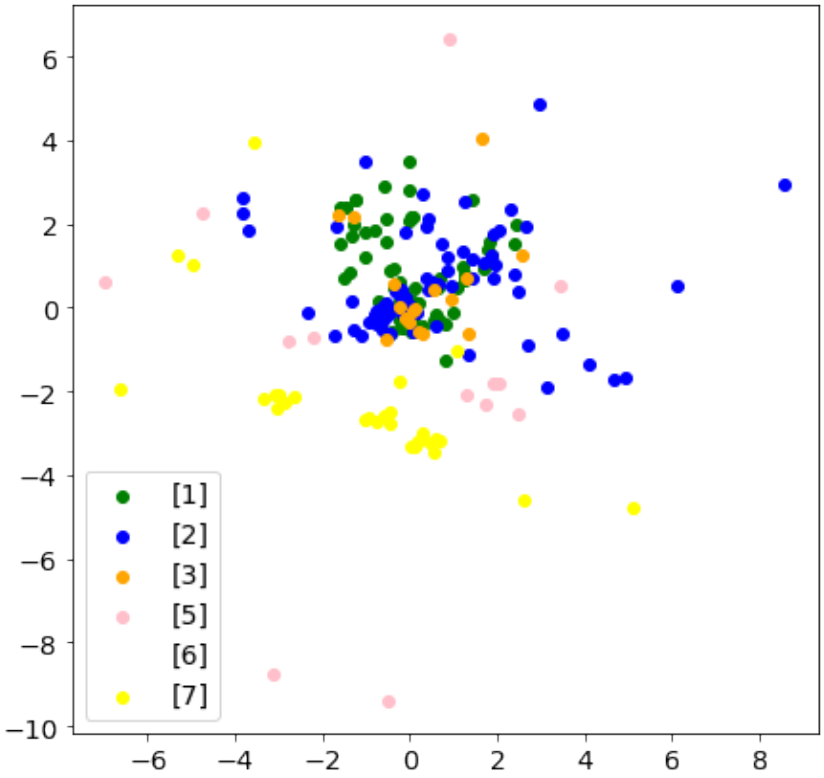
PCA It is a method of summarizing data by reducing dimensions.Many of information in features could be redundant. If so, we should be able to summarize features with fewer characteristics.PCA looks for properties that show as much variation across data as possible.PCA looks for properties that allow to reconstruct the original data characteristics as well as possible.



8 components explain 90 percent of the data.

1.3.2 Multi Dimensional Scaling

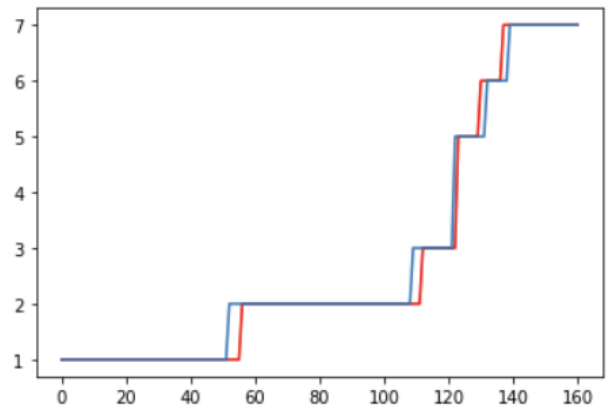
The purpose of multidimensional scaling (MDS) is to provide a visual representation of the pattern of proximities (i.e., similarities or distances) among a set of objects. MDS will help in visualising different type of glasses and how they are similar or dissimilar.



1.3.3 K Neighbours Classifier

K Nearest Neighbors is a Classification Algorithm based on K means . Just as with every classification algorithm is important that the algorithm doesn't "remember" the answers and that the answer it gets can be generalized to all the population, not just learned from the database. The traininig process is needed in KNN because , we want to avoid it from learning-the-database,. In a regression the training is for finding the optimum parameters, here too, but the idea of "parameters" slightly changes: Here the parameters are the selected points (individuals) , the k, separating in train/test allows the model to be more general when the test data is unknown. KNN calculates distance only in the training set. When the optimum parameters are found in the training set, the results in the test set are better, more general.
Optimum K value found=3 ,lines are not too smooth and have good accuracy score. Which means that we will check 3 nearest data points near the label to be classified and calculate distances between them

```
plt.plot(sorted(y_pre),color="red")
plt.plot(sorted(ytest))
plt.show()
print("ACCURACY :",acc,"when neighbors:",3)
```




```
ACCURACY : 0.8944099378881988 when neighbors: 3
```