**FINAL PROJECT**

Course Name: Introduction to Data Science

Exam Type: Multi-class Classification

Yavuz Selim Gügen

Ertuğrul Çoban

**Part I Project Summary**

**Write a brief summary of the dataset**: Our dataset contains 7 type of glass, one of them doesn’t appears in data set. The classes of glass are based on their physical and chemical values.

Study aims to classify these types according to their physical and chemical values.

|  |  |
| --- | --- |
| **1** | Processed float building windows |
| **2** | Processed non-float building windows |
| **3** | Processed float vehicle windows |
| **4** | Processed non-float vehicle windows |
| **5** | Containers |
| **6** | Tableware |
| **7** | Headlamps |

**Web site address of the datasets:** <https://archive.ics.uci.edu/ml/datasets/Glass+Identification>

**Part II Dataset Description**

**What is it about:** From USA Forensic Science Service; 6 types of glass; defined in terms of their oxide content (i.e. Na, Fe, K, etc)

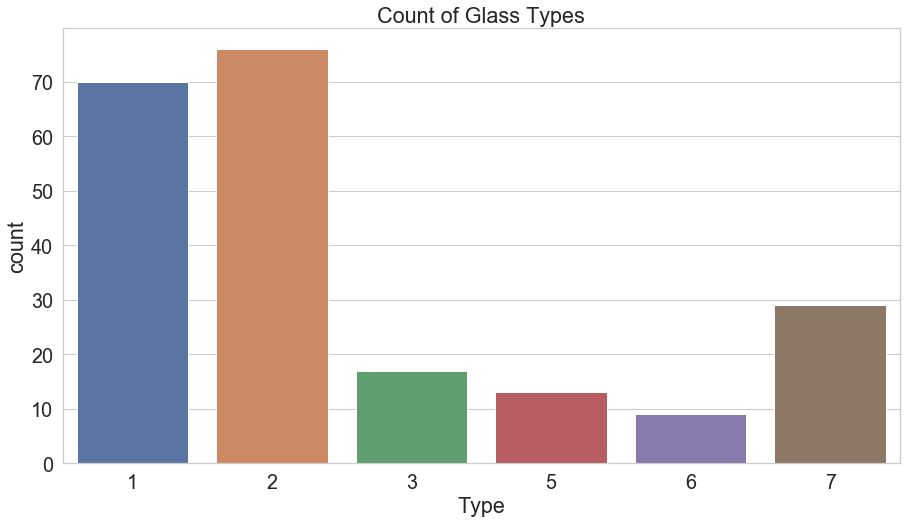
**How many samples are there:** 214

**How many variables are there in your dataset:** 10

**What are the types of each of those variables:** Integer

**Part III Basic Descriptive Statistics**

|  | **RI** | **Na** | **Mg** | **Al** | **Si** | **K** | **Ca** | **Ba** | **Fe** | **Type** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 214.000000 | 214.000000 | 214.000000 | 214.000000 | 214.000000 | 214.000000 | 214.000000 | 214.000000 | 214.000000 | 214.000000 |
| **mean** | 1.518365 | 13.407850 | 2.684533 | 1.444907 | 72.650935 | 0.497056 | 8.956963 | 0.175047 | 0.057009 | 2.780374 |
| **std** | 0.003037 | 0.816604 | 1.442408 | 0.499270 | 0.774546 | 0.652192 | 1.423153 | 0.497219 | 0.097439 | 2.103739 |
| **min** | 1.511150 | 10.730000 | 0.000000 | 0.290000 | 69.810000 | 0.000000 | 5.430000 | 0.000000 | 0.000000 | 1.000000 |
| **25%** | 1.516523 | 12.907500 | 2.115000 | 1.190000 | 72.280000 | 0.122500 | 8.240000 | 0.000000 | 0.000000 | 1.000000 |
| **50%** | 1.517680 | 13.300000 | 3.480000 | 1.360000 | 72.790000 | 0.555000 | 8.600000 | 0.000000 | 0.000000 | 2.000000 |
| **75%** | 1.519157 | 13.825000 | 3.600000 | 1.630000 | 73.087500 | 0.610000 | 9.172500 | 0.000000 | 0.100000 | 3.000000 |
| **max** | 1.533930 | 17.380000 | 4.490000 | 3.500000 | 75.410000 | 6.210000 | 16.190000 | 3.150000 | 0.510000 | 7.000000 |

****

RangeIndex: 214 entries, 0 to 213 Data columns (total 10 columns):

|  |  |  |
| --- | --- | --- |
| RI 214 | Non-null | float64 |
| Na 214 | Non-null | float64 |
| Mg 214 | Non-null | float64 |
| Al 214 | Non-null | float64 |
| Si 214 | Non-null | float64 |
| K 214 | Non-null | float64 |
| Ca 214 | Non-null | float64 |
| Ba 214 | Non-null | float64 |
| Fe 214 | Non-null | float64 |
| Type 214 | Non-null | float64 |

For outliers:

**from** **scipy** **import** stats

glass\_df = glass\_df[(np.abs(stats.zscore(glass\_df)) < 3).all(axis=1)]

**Part IV Data Management Processes**

Our data comes from one csv file. There is no merge process.

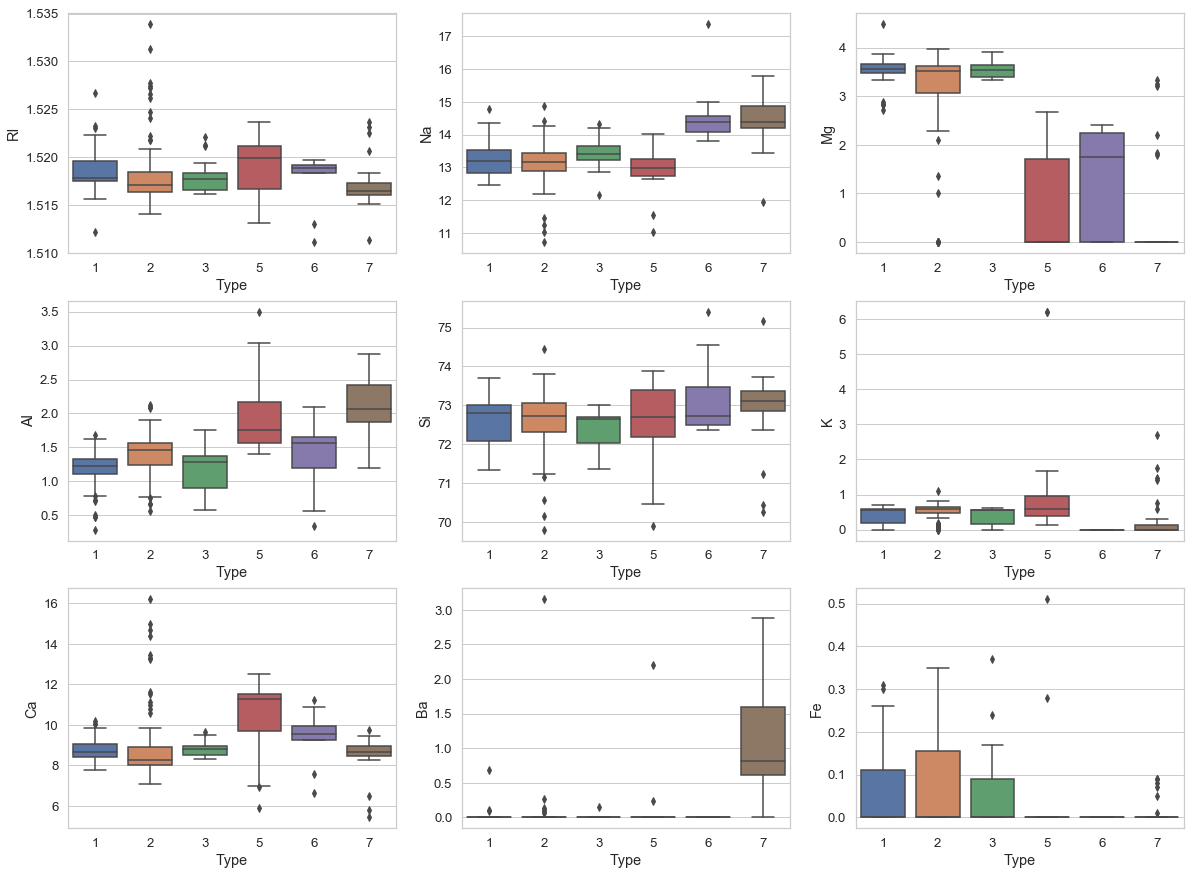
Missing values are converted to 0.

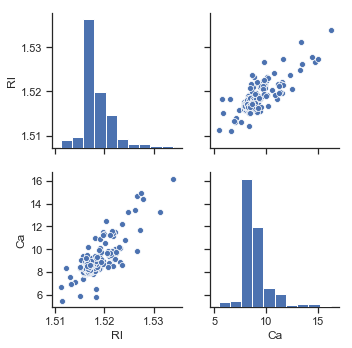
Outliers eliminated thanks to Z-score test.

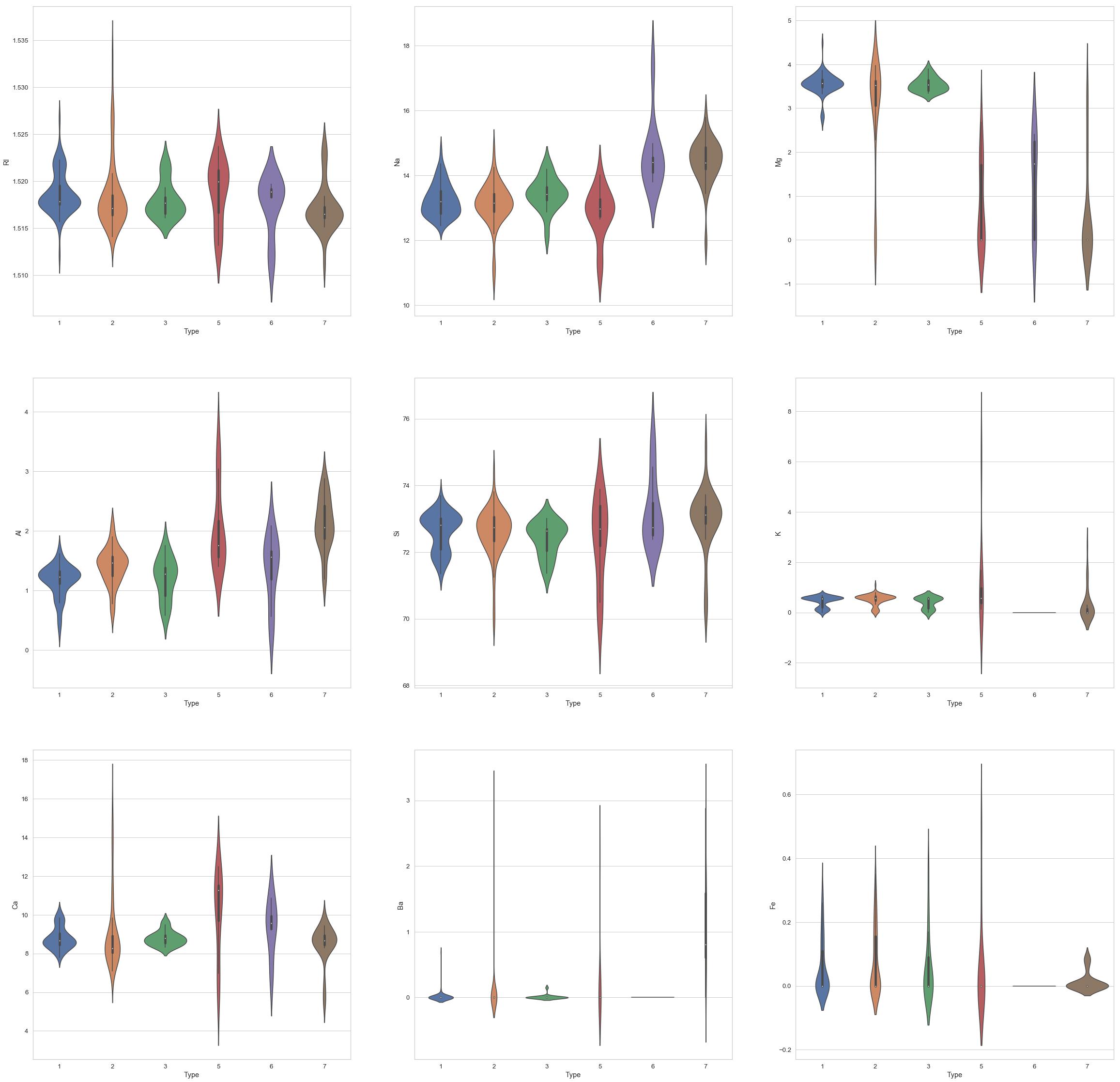
Data transformation is not required. Because all column got integer type values.

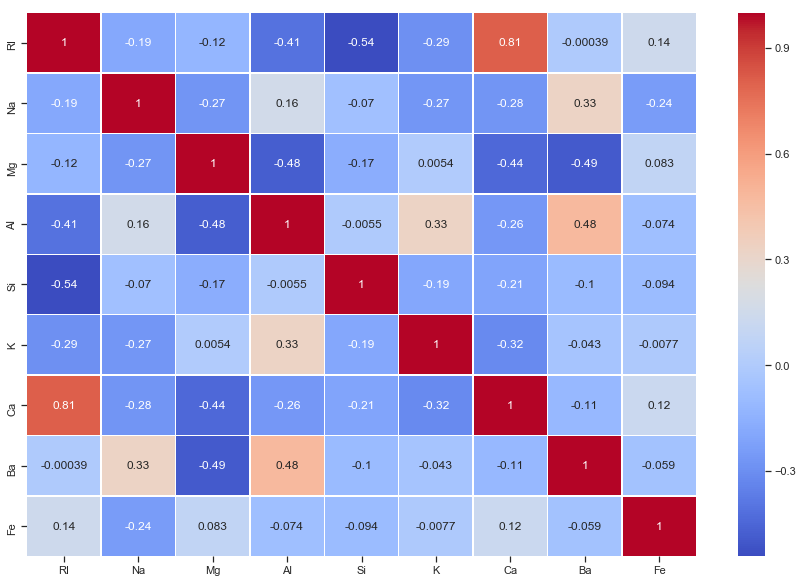
And there is no need to scale any feature except type column which holds class names.

**Part V Data Visualization**









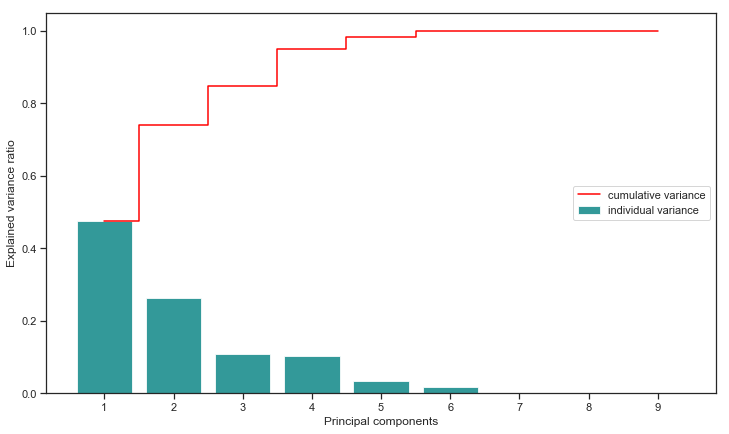
**Part VI Machine Learning (ML) Implementation**

**Which type of ML method have you used?**

Decision tree classifier, Random forest classifier, SVC and Voting classifier which has first 3 classifier tested. Random forest classifier has better results according to accuracy.

|  |  |
| --- | --- |
| DecisionTreeClassifier | 0.6307692307692307 |
| RandomForestClassifier | 0.7692307692307693 |
| SVC | 0.7538461538461538 |
| VotingClassifier | 0.6461538461538462 |

**Feature Selection using Principal Component Analysis**



We choose 5 pc because cumulative variance reaches maximum value at 5.

**Fine Tuning**

clf = RandomForestClassifier( random\_state = 0)

param\_grid = {

"n\_estimators": [7,10,12],

"max\_depth":[3,5,7],

"min\_samples\_split":[3,5,7],

"min\_samples\_leaf":[2,5,10],

"max\_features" : ['sqrt', 'auto']

}

grid = GridSearchCV(clf, param\_grid, scoring='accuracy')

After cross-validation best parameter values are below:

'max\_depth': 5,

'max\_features': 'sqrt',

'min\_samples\_leaf': 2,

'min\_samples\_split': 5,

'n\_estimators': 10