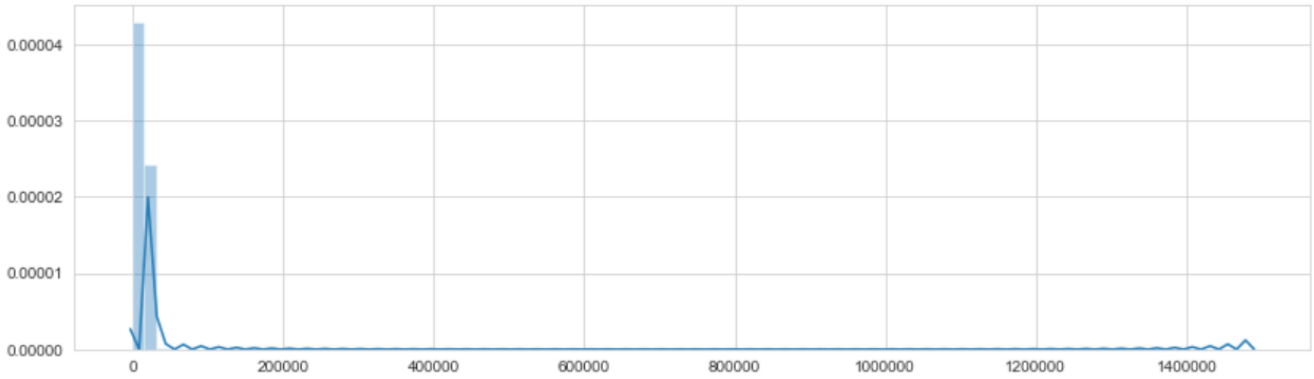
Data-Summary

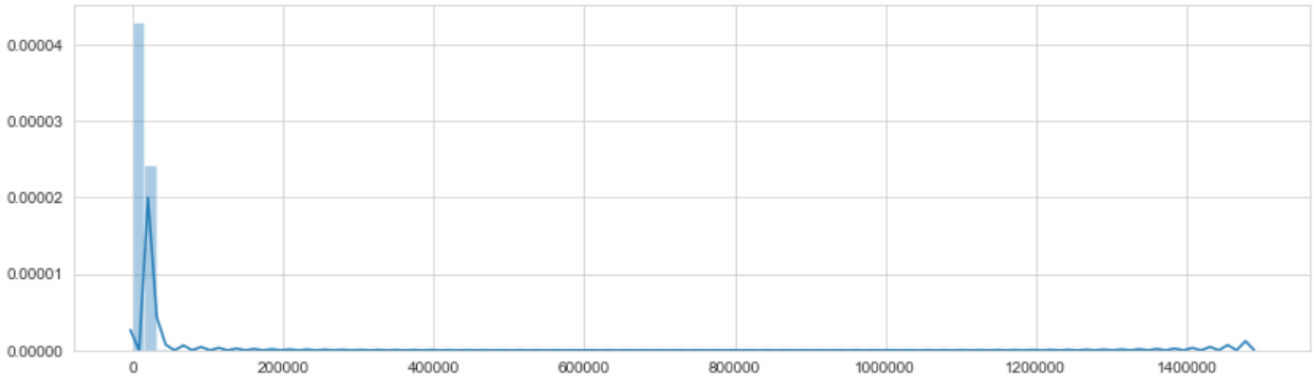
Initial Data Analysis

-We have a data set unlabeled with shape 66137 rows and 296 attributes.
-4 Float, 291 integer and 1 object attribute.
-3837 Duplicates.
-The data is imbalanced with the majority class C have almost the 70% and minority A 1,3%. Here we can describe our data like having 4 out of 5 minority classes with the second one after the majority occupy the 15% of the data.

Some first plot of the data:

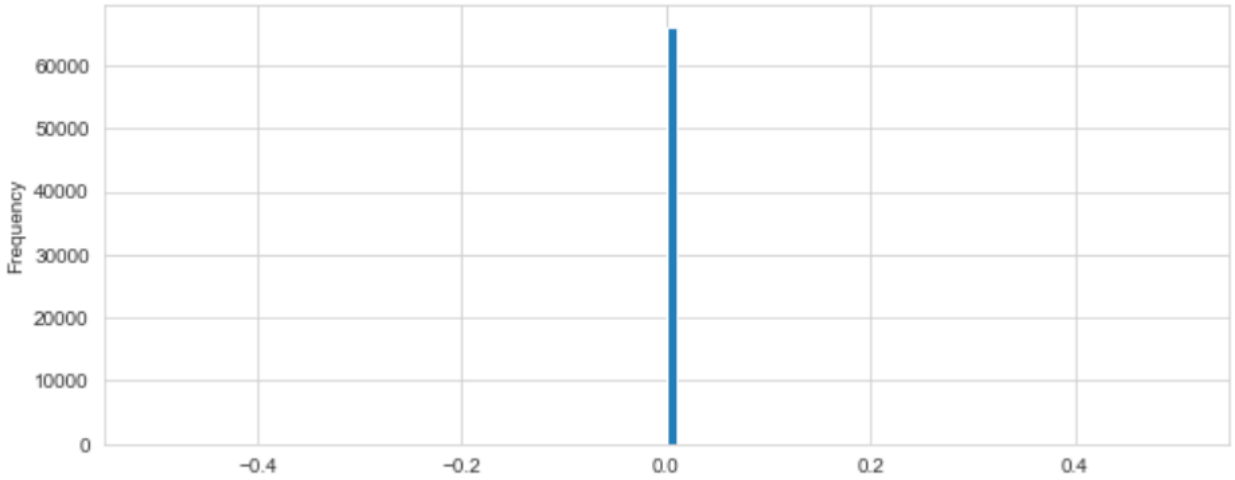


Sum distribution.

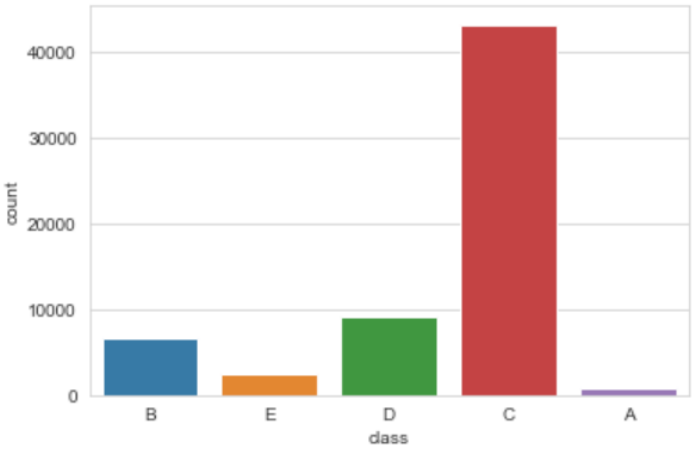


Mean distribution.

Median distribution

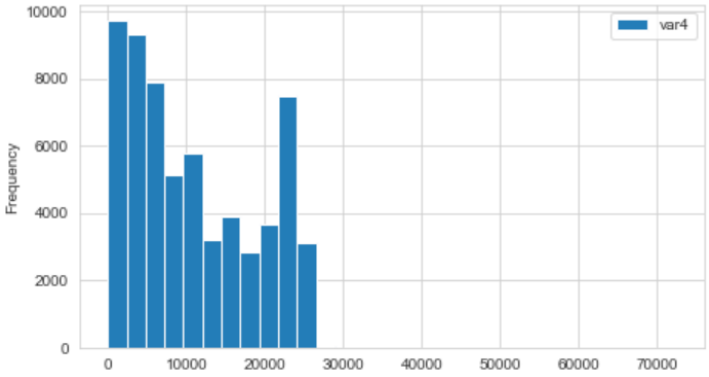


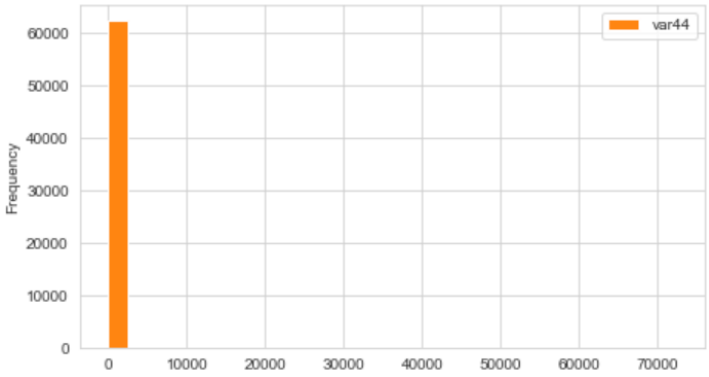
Distribution of the classes

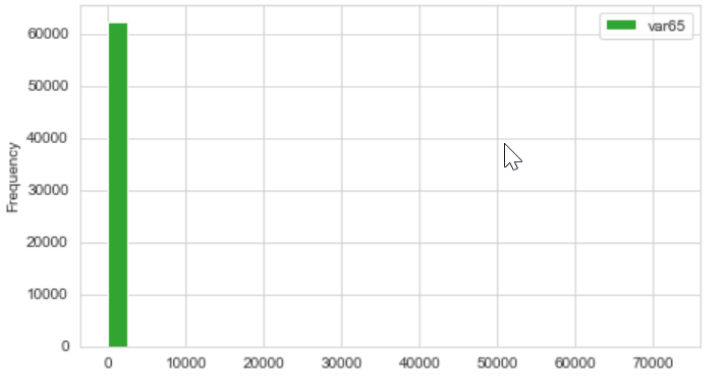


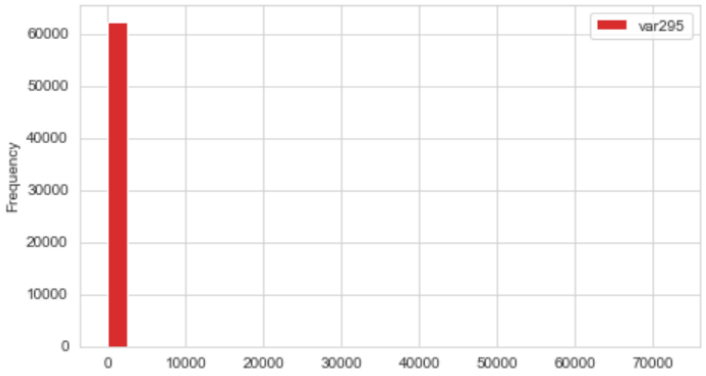
	class	counts
0	A	864
1	B	6547
2	C	43217
3	D	9179
4	E	2493

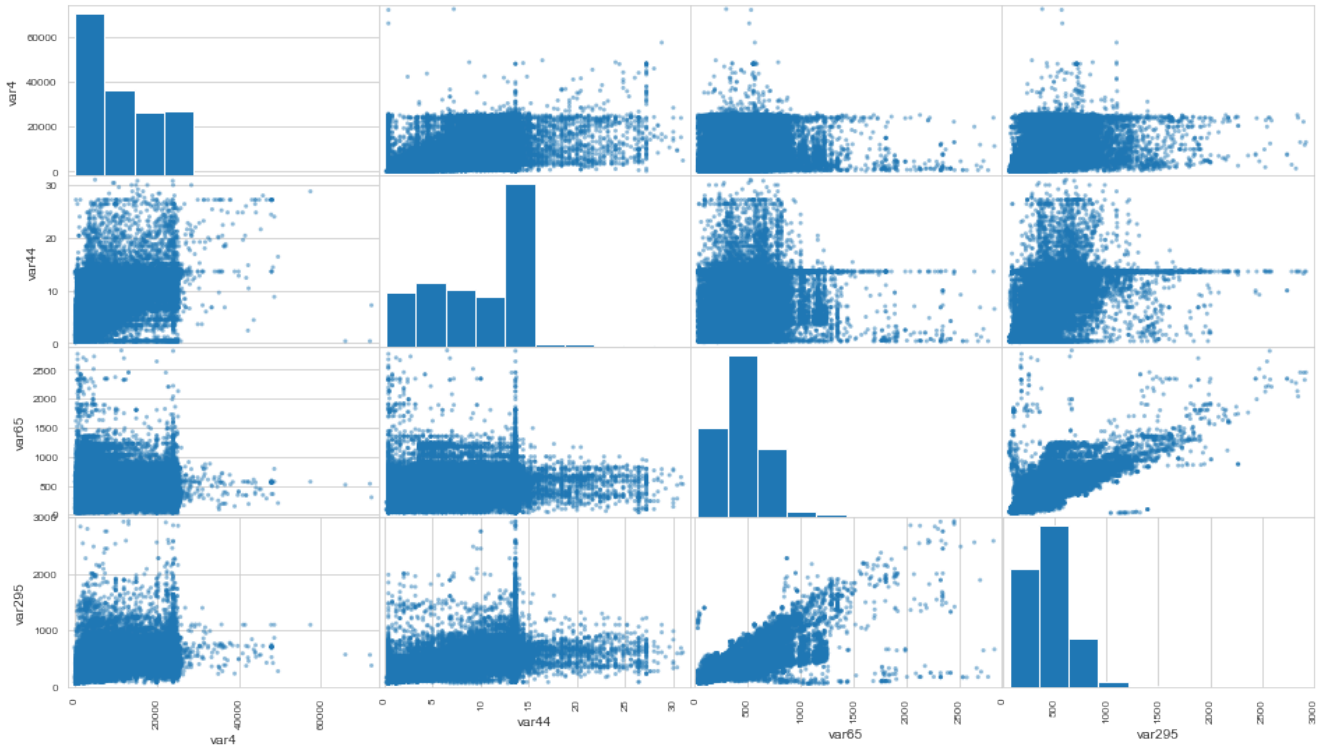
Low variation of the 4 float variables besides one.





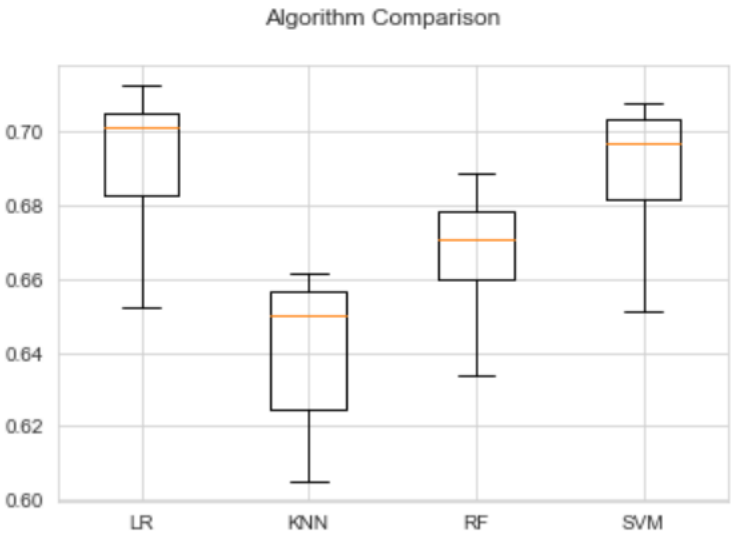






There is a correlation (0.743285) between the var 295 & 65.

-A first evaluation of several algorithms gave the following rank.



0	LR: 0.692500 (0.020456)
1	KNN: 0.640750 (0.020806)
2	RF: 0.666625 (0.016984)
3	SVM: 0.689500 (0.019358)

Here we are observing an overall performance measured with accuracy_score.
The above estimation of performance took place through cross validation. We will see that these results are not so representative with the following performance of our algorithms.

Logistic regression gave accuracy: 0.48 and Kappa score: 0.1673 ,for normalized data.
We have here to mention that there is a need to take into account a different performance metric than accuracy since our data are imbalanced , accuracy misleading to skew results.

Just for comparison the logistic regression with accuracy: 0.48 and Kappa score: 0.1673 had a classification report as:

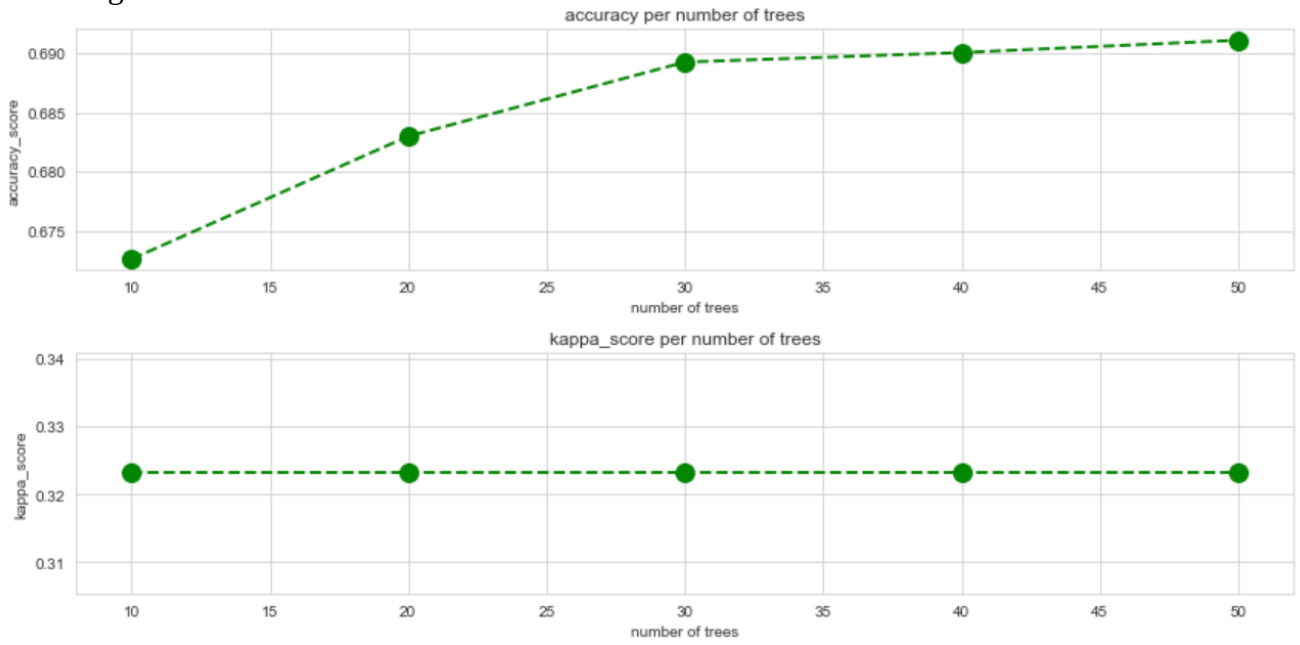
	precision	recall	f1-score	support
A	0.04	0.33	0.06	172
B	0.18	0.22	0.20	1284
C	0.85	0.59	0.70	8718
D	0.23	0.20	0.22	1821
E	0.09	0.31	0.13	465

Where random forest with accuracy: 0.69 and Kappa score: 0.10 :

	precision	recall	f1-score	support
A	0.01	0.17	0.02	12
B	0.06	0.28	0.10	280
C	0.95	0.72	0.82	11535
D	0.11	0.35	0.17	570
E	0.04	0.32	0.08	63

We observing that even with greater accuracy is unable to predict each class quite accurate. This is happening cause this performance metric give more weight to the majority class. But the logistic regression even with lower accuracy and higher kappa we see the precision of each class be much better.

Following we can see the results of random forest for several number of trees.With all data normalized.



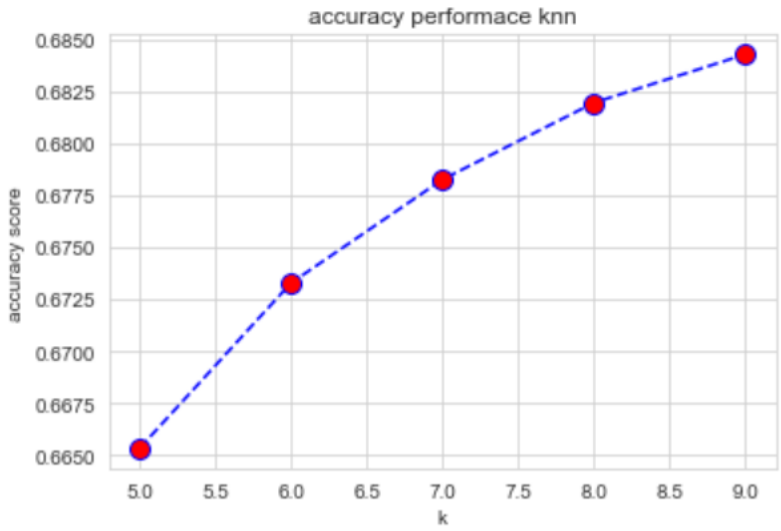
Above we observe that the accuracy is getting improved as we increasing the number of trees but the kappa remain stable. This means that nothing is changing in the distributed predictions for the minority classes.

Best performance: we took was with knn. Accuracy: 0.74 and Kappa score: 0.32 (fair performance)
For the Kappa:
(a value < 0 is indicating no agreement , 0–0.20 as slight, 0.21–0.40 as fair, 0.41–0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1 as almost perfect agreement.)
-Possibly with further tuning of knn we could get even better results. The under-sampling (with TomekLinks) didn't helped our task.
Follow we have the classification report of knn:

	precision	recall	f1-score	support
A	0.43	0.12	0.18	172
B	0.47	0.32	0.38	1284
C	0.77	0.95	0.85	8718
D	0.61	0.25	0.36	1821
E	0.62	0.12	0.20	465

Where we can see a much better balance between the minority classes A,B,D,E

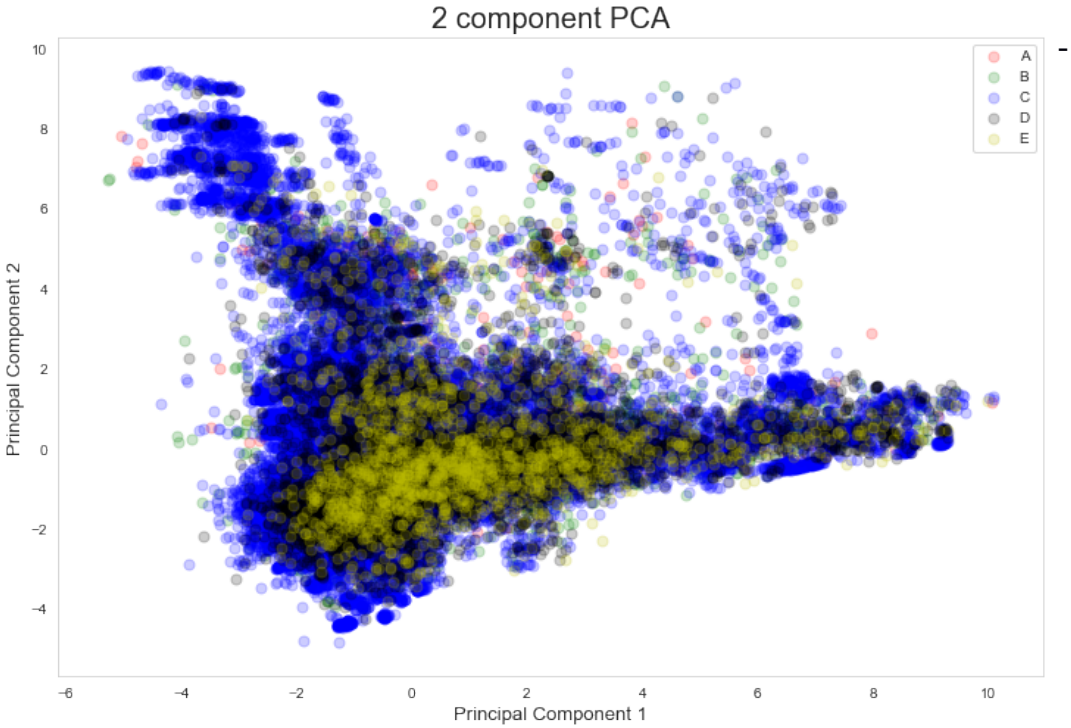
Mean Error plot VS Accuracy performance plot



Configuration of knn:
KneighborsClassifier :
algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=5, p=2,weights='uniform'.

PCA approach:

An overall appearance of our data in 2-D projection after applying pca.
But not even pca, or feature extraction with the importance variable through random forest or the subtraction of the outliers could help to overcome the first performance of knn. A simple algorithm compare to other but quite efficient for our task.



-