Lab2 Report

Mohamed Elmaghraby Mohamed

55

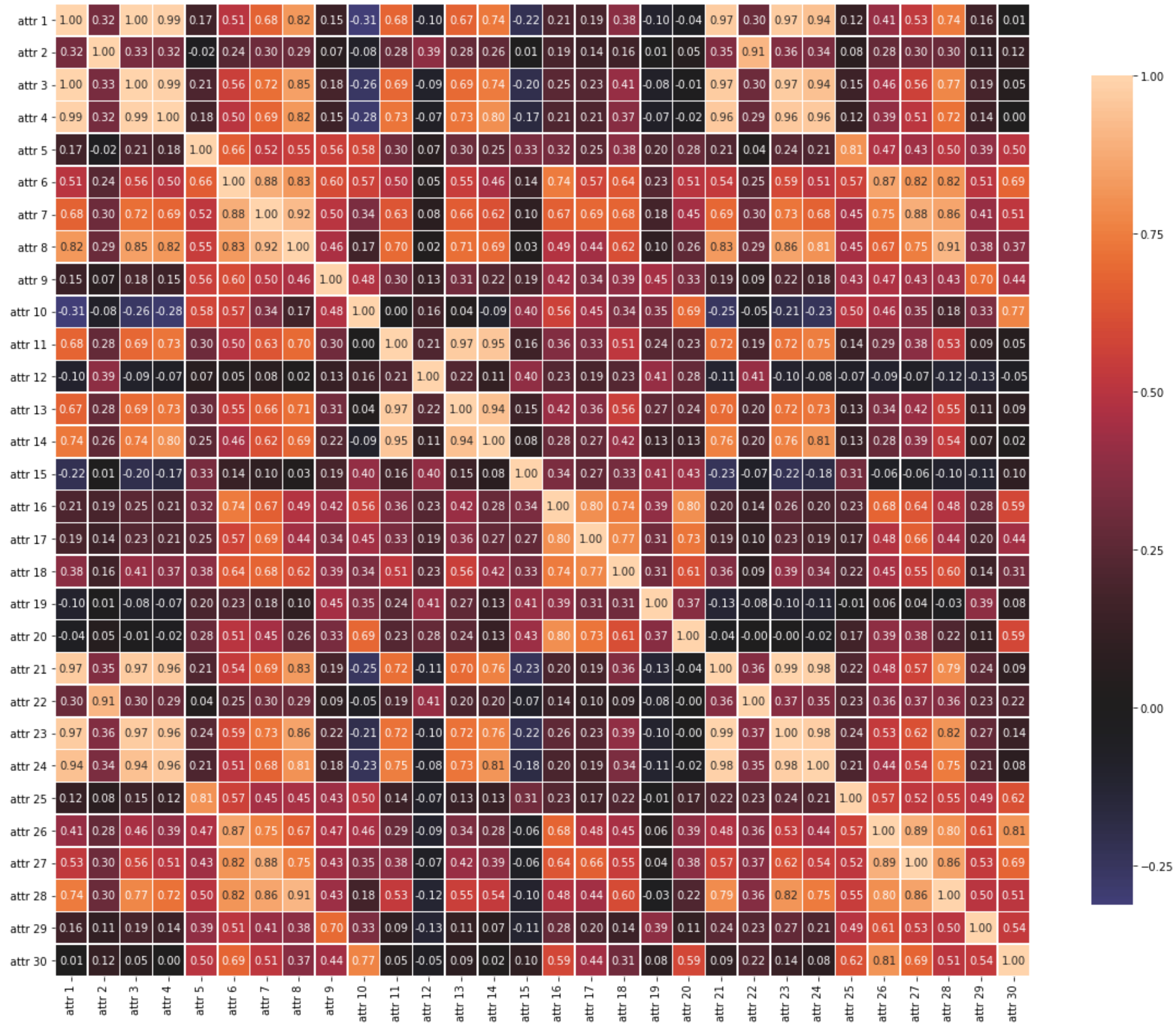
Prepare data:

- Prepare column names.
- Import breast-cancer-Wisconsin data.
- Merge both data sets.
- Explore data :
    - Number of instances = 569
    - number of attributes = 32
    - class names : ['M', 'B']
    - number of classes: 2

Data Exploration:

- **Plot Box plot for the data**
    - **Conclusion :**
        - most of the values of the attributes are around zero
        - attr 4 and attr 24: has bigger and spread values
        - attr 14: has many outliers
- **Correlation matrix**
    - Use `.corr(method = 'Pearson')` to get the Pearson's correlation between each 2 attributes.
    - **Conclusion**
        - some of the attributes have a high positive correlation
        - most of the attributes have a medium positive correlation
        - a small set of attributes has a low positive correlation

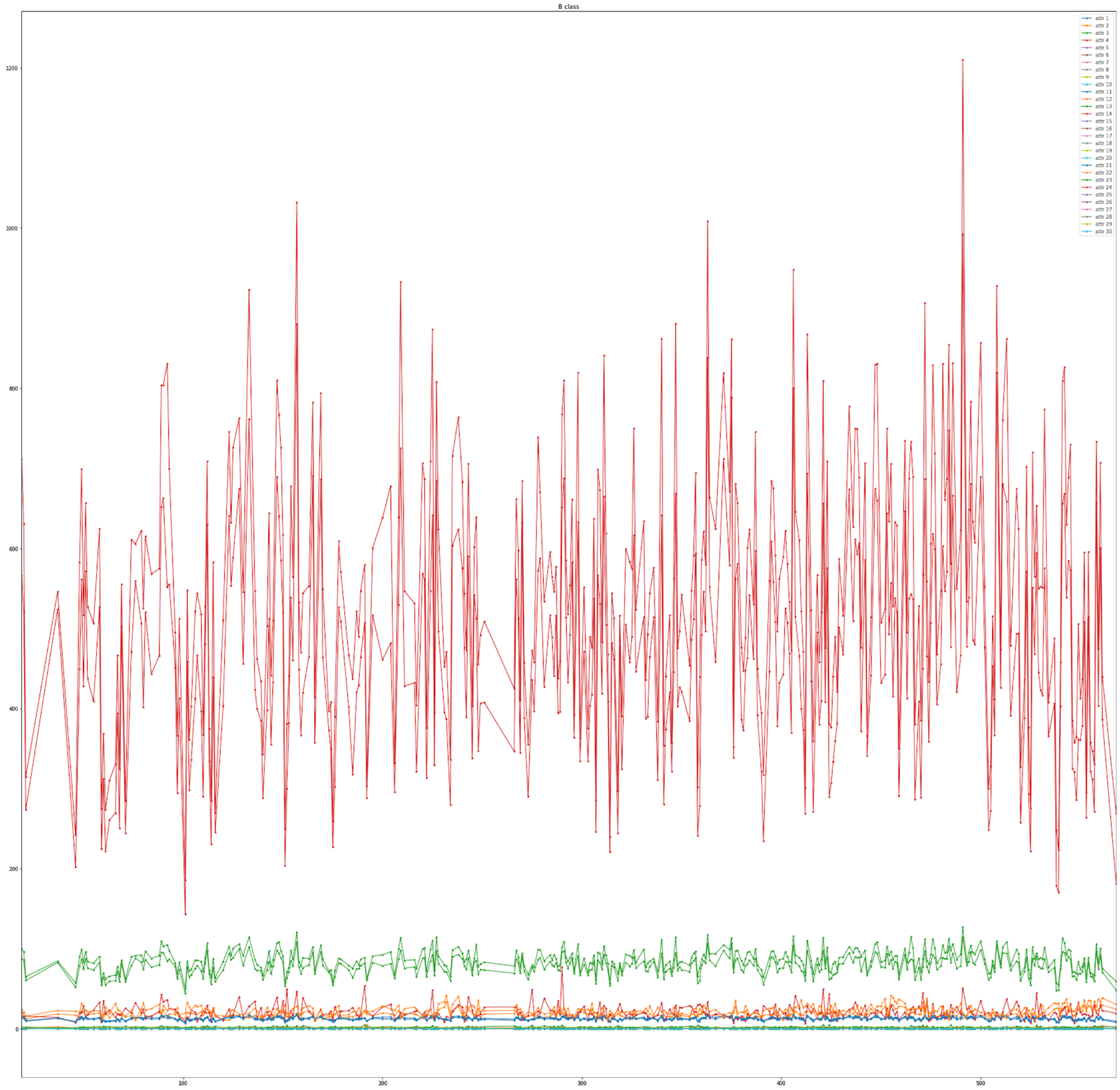
        - Examples :
            - attrs (21, 23, 24) vs attrs (1, 3, 4) has very strong correlation
            - attrs(25, 26, 27, 28, 29, 30) has very strong correlation
            - attrs (15, 16, ..., 20) vs attrs (1, 2 , 3 , 4) has very weak correlation
    - Draw the correlation using heatmap.



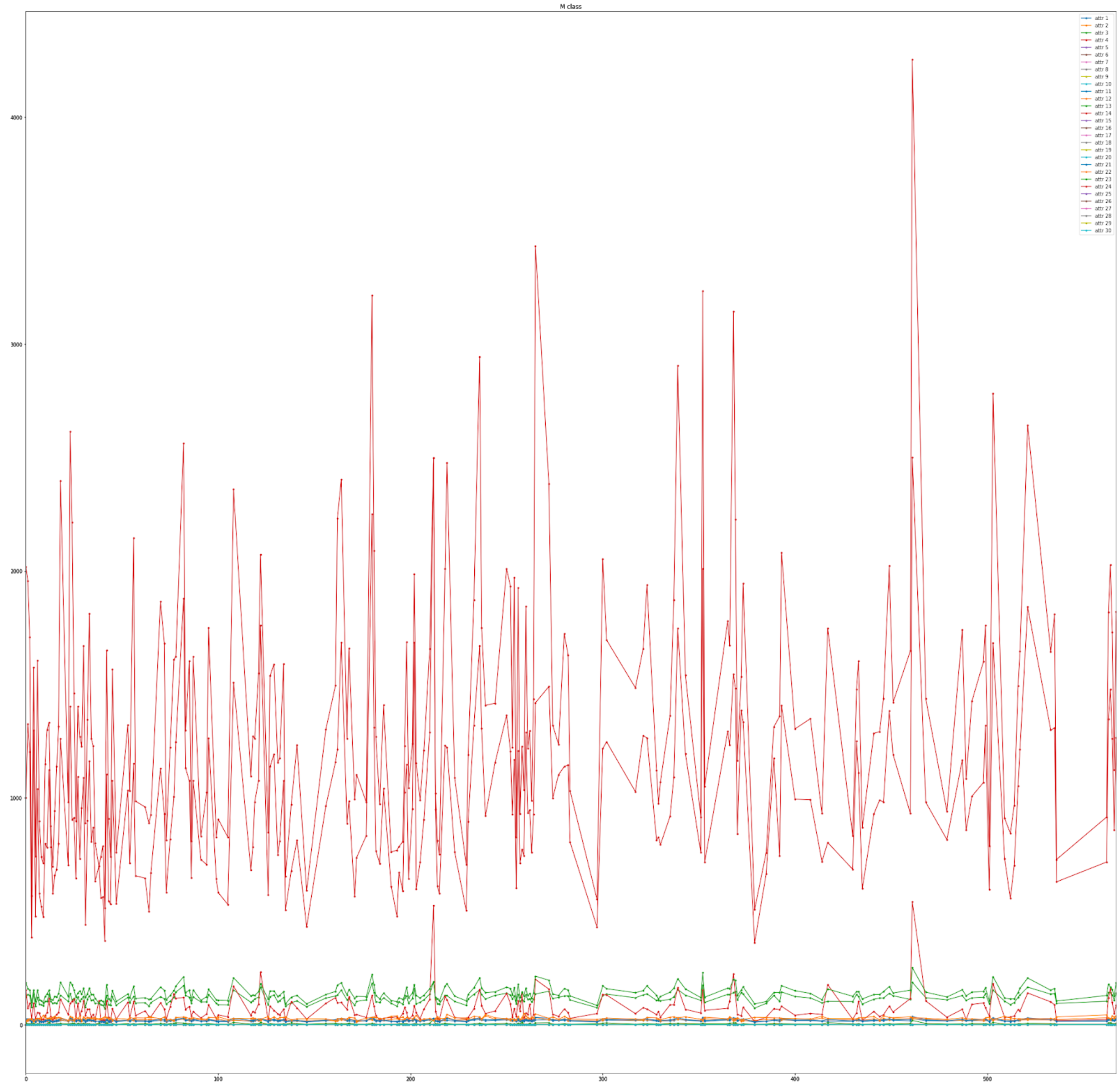
	attr 1	attr 2	attr 3	attr 4	attr 5	attr 6	attr 7	attr 8	attr 9	attr 10	attr 11	attr 12	attr 13	attr 14	attr 15	attr 16	attr 17	attr 18	attr 19	attr 20	attr 21	attr 22	attr 23	attr 24	attr 25	attr 26	attr 27	attr 28	attr 29	attr 30
attr 1	1.00	0.32	1.00	0.99	0.17	0.51	0.68	0.82	0.15	-0.31	0.68	-0.10	0.67	0.74	-0.22	0.21	0.19	0.38	-0.10	-0.04	0.97	0.30	0.97	0.94	0.12	0.41	0.53	0.74	0.16	0.01
attr 2	0.32	1.00	0.33	0.32	-0.02	0.24	0.30	0.29	0.07	-0.08	0.28	0.39	0.28	0.26	0.01	0.19	0.14	0.16	0.01	0.05	0.35	0.91	0.36	0.34	0.08	0.28	0.30	0.30	0.11	0.12
attr 3	1.00	0.33	1.00	0.99	0.21	0.56	0.72	0.85	0.18	-0.26	0.69	-0.09	0.69	0.74	-0.20	0.25	0.23	0.41	-0.08	-0.01	0.97	0.30	0.97	0.94	0.15	0.46	0.56	0.77	0.19	0.05
attr 4	0.99	0.32	0.99	1.00	0.18	0.50	0.69	0.82	0.15	-0.28	0.73	-0.07	0.73	0.80	-0.17	0.21	0.21	0.37	-0.07	-0.02	0.96	0.29	0.96	0.96	0.12	0.39	0.51	0.72	0.14	0.00
attr 5	0.17	-0.02	0.21	0.18	1.00	0.66	0.52	0.55	0.56	0.58	0.30	0.07	0.30	0.25	0.33	0.32	0.25	0.38	0.20	0.28	0.21	0.04	0.24	0.21	0.81	0.47	0.43	0.50	0.39	0.50
attr 6	0.51	0.24	0.56	0.50	0.66	1.00	0.88	0.83	0.60	0.57	0.50	0.05	0.55	0.46	0.14	0.74	0.57	0.64	0.23	0.51	0.54	0.25	0.59	0.51	0.57	0.87	0.82	0.82	0.51	0.69
attr 7	0.68	0.30	0.72	0.69	0.52	0.88	1.00	0.92	0.50	0.34	0.63	0.08	0.66	0.62	0.10	0.67	0.69	0.68	0.18	0.45	0.69	0.30	0.73	0.68	0.45	0.75	0.88	0.86	0.41	0.51
attr 8	0.82	0.29	0.85	0.82	0.55	0.83	0.92	1.00	0.46	0.17	0.70	0.02	0.71	0.69	0.03	0.49	0.44	0.62	0.10	0.26	0.83	0.29	0.86	0.81	0.45	0.67	0.75	0.91	0.38	0.37
attr 9	0.15	0.07	0.18	0.15	0.56	0.60	0.50	0.46	1.00	0.48	0.30	0.13	0.31	0.22	0.19	0.42	0.34	0.39	0.45	0.33	0.19	0.09	0.22	0.18	0.43	0.47	0.43	0.43	0.70	0.44
attr 10	-0.31	-0.08	-0.26	-0.28	0.58	0.57	0.34	0.17	0.48	1.00	0.00	0.16	0.04	-0.09	0.40	0.56	0.45	0.34	0.35	0.69	-0.25	-0.05	-0.21	-0.23	0.50	0.46	0.35	0.18	0.33	0.77
attr 11	0.68	0.28	0.69	0.73	0.30	0.50	0.63	0.70	0.30	0.00	1.00	0.21	0.97	0.95	0.16	0.36	0.33	0.51	0.24	0.23	0.72	0.19	0.72	0.75	0.14	0.29	0.38	0.53	0.09	0.05
attr 12	-0.10	0.39	-0.09	-0.07	0.07	0.05	0.08	0.02	0.13	0.16	0.21	1.00	0.22	0.11	0.40	0.23	0.19	0.23	0.41	0.28	-0.11	0.41	-0.10	-0.08	-0.07	-0.09	-0.07	-0.12	-0.13	-0.05
attr 13	0.67	0.28	0.69	0.73	0.30	0.55	0.66	0.71	0.31	0.04	0.97	0.22	1.00	0.94	0.15	0.42	0.36	0.56	0.27	0.24	0.70	0.20	0.72	0.73	0.13	0.34	0.42	0.55	0.11	0.09
attr 14	0.74	0.26	0.74	0.80	0.25	0.46	0.62	0.69	0.22	-0.09	0.95	0.11	0.94	1.00	0.08	0.28	0.27	0.42	0.13	0.13	0.76	0.20	0.76	0.81	0.13	0.28	0.39	0.54	0.07	0.02
attr 15	-0.22	0.01	-0.20	-0.17	0.33	0.14	0.10	0.03	0.19	0.40	0.16	0.40	0.15	0.08	1.00	0.34	0.27	0.33	0.41	0.43	-0.23	-0.07	-0.22	-0.18	0.31	-0.06	-0.06	-0.10	-0.11	0.10
attr 16	0.21	0.19	0.25	0.21	0.32	0.74	0.67	0.49	0.42	0.56	0.36	0.23	0.42	0.28	0.34	1.00	0.80	0.74	0.39	0.80	0.20	0.14	0.26	0.20	0.23	0.68	0.64	0.48	0.28	0.59
attr 17	0.19	0.14	0.23	0.21	0.25	0.57	0.69	0.44	0.34	0.45	0.33	0.19	0.36	0.27	0.27	0.80	1.00	0.77	0.31	0.73	0.19	0.10	0.23	0.19	0.17	0.48	0.66	0.44	0.20	0.44
attr 18	0.38	0.16	0.41	0.37	0.38	0.64	0.68	0.62	0.39	0.34	0.51	0.23	0.56	0.42	0.33	0.74	0.77	1.00	0.31	0.61	0.36	0.09	0.39	0.34	0.22	0.45	0.55	0.60	0.14	0.31
attr 19	-0.10	0.01	-0.08	-0.07	0.20	0.23	0.18	0.10	0.45	0.35	0.24	0.41	0.27	0.13	0.41	0.39	0.31	0.31	1.00	0.37	-0.13	-0.08	-0.10	-0.11	-0.01	0.06	0.04	-0.03	0.39	0.08
attr 20	-0.04	0.05	-0.01	-0.02	0.28	0.51	0.45	0.26	0.33	0.69	0.23	0.28	0.24	0.13	0.43	0.80	0.73	0.61	0.37	1.00	-0.04	-0.00	-0.00	-0.02	0.17	0.39	0.38	0.22	0.11	0.59
attr 21	0.97	0.35	0.97	0.96	0.21	0.54	0.69	0.83	0.19	-0.25	0.72	-0.11	0.70	0.76	-0.23	0.20	0.19	0.36	-0.13	-0.04	1.00	0.36	0.99	0.98	0.22	0.48	0.57	0.79	0.24	0.09
attr 22	0.30	0.91	0.30	0.29	0.04	0.25	0.30	0.29	0.09	-0.05	0.19	0.41	0.20	0.20	-0.07	0.14	0.10	0.09	-0.08	-0.00	0.36	1.00	0.37	0.35	0.23	0.36	0.37	0.36	0.23	0.22
attr 23	0.97	0.36	0.97	0.96	0.24	0.59	0.73	0.86	0.22	-0.21	0.72	-0.10	0.72	0.76	-0.22	0.26	0.23	0.39	-0.10	-0.00	0.99	0.37	1.00	0.98	0.24	0.53	0.62	0.82	0.27	0.14
attr 24	0.94	0.34	0.94	0.96	0.21	0.51	0.68	0.81	0.18	-0.23	0.75	-0.08	0.73	0.81	-0.18	0.20	0.19	0.34	-0.11	-0.02	0.98	0.35	0.98	1.00	0.21	0.44	0.54	0.75	0.21	0.08
attr 25	0.12	0.08	0.15	0.12	0.81	0.57	0.45	0.45	0.43	0.50	0.14	-0.07	0.13	0.13	0.31	0.23	0.17	0.22	-0.01	0.17	0.22	0.23	0.24	0.21	1.00	0.57	0.52	0.55	0.49	0.62
attr 26	0.41	0.28	0.46	0.39	0.47	0.87	0.75	0.67	0.47	0.46	0.29	-0.09	0.34	0.28	-0.06	0.68	0.48	0.45	0.06	0.39	0.48	0.36	0.53	0.44	0.57	1.00	0.89	0.80	0.61	0.81
attr 27	0.53	0.30	0.56	0.51	0.43	0.82	0.88	0.75	0.43	0.35	0.38	-0.07	0.42	0.39	-0.06	0.64	0.66	0.55	0.04	0.38	0.57	0.37	0.62	0.54	0.52	0.89	1.00	0.86	0.53	0.69
attr 28	0.74	0.30	0.77	0.72	0.50	0.82	0.86	0.91	0.43	0.18	0.53	-0.12	0.55	0.54	-0.10	0.48	0.44	0.60	-0.03	0.22	0.79	0.36	0.82	0.75	0.55	0.80	0.86	1.00	0.50	0.51
attr 29	0.16	0.11	0.19	0.14	0.39	0.51	0.41	0.38	0.70	0.33	0.09	-0.13	0.11	0.07	-0.11	0.28	0.20	0.14	0.39	0.11	0.24	0.23	0.27	0.21	0.49	0.61	0.53	0.50	1.00	0.54
attr 30	0.01	0.12	0.05	0.00	0.50	0.69	0.51	0.37	0.44	0.77	0.05	-0.05	0.09	0.02	0.10	0.59	0.44	0.31	0.08	0.59	0.09	0.22	0.14	0.08	0.62	0.81	0.69	0.51	0.54	1.00

- **Line plot**
  - I plotted a line plot for each class independently
  - **Conclusion**
    - by comparing both attr 3 and attr 23 seems to has a higher value in class 'B' than in class' (colored by green)
    - same for attrs (1, 2, 21, 22, 11, 12) (colored by blue and orange)

Class B



Class M

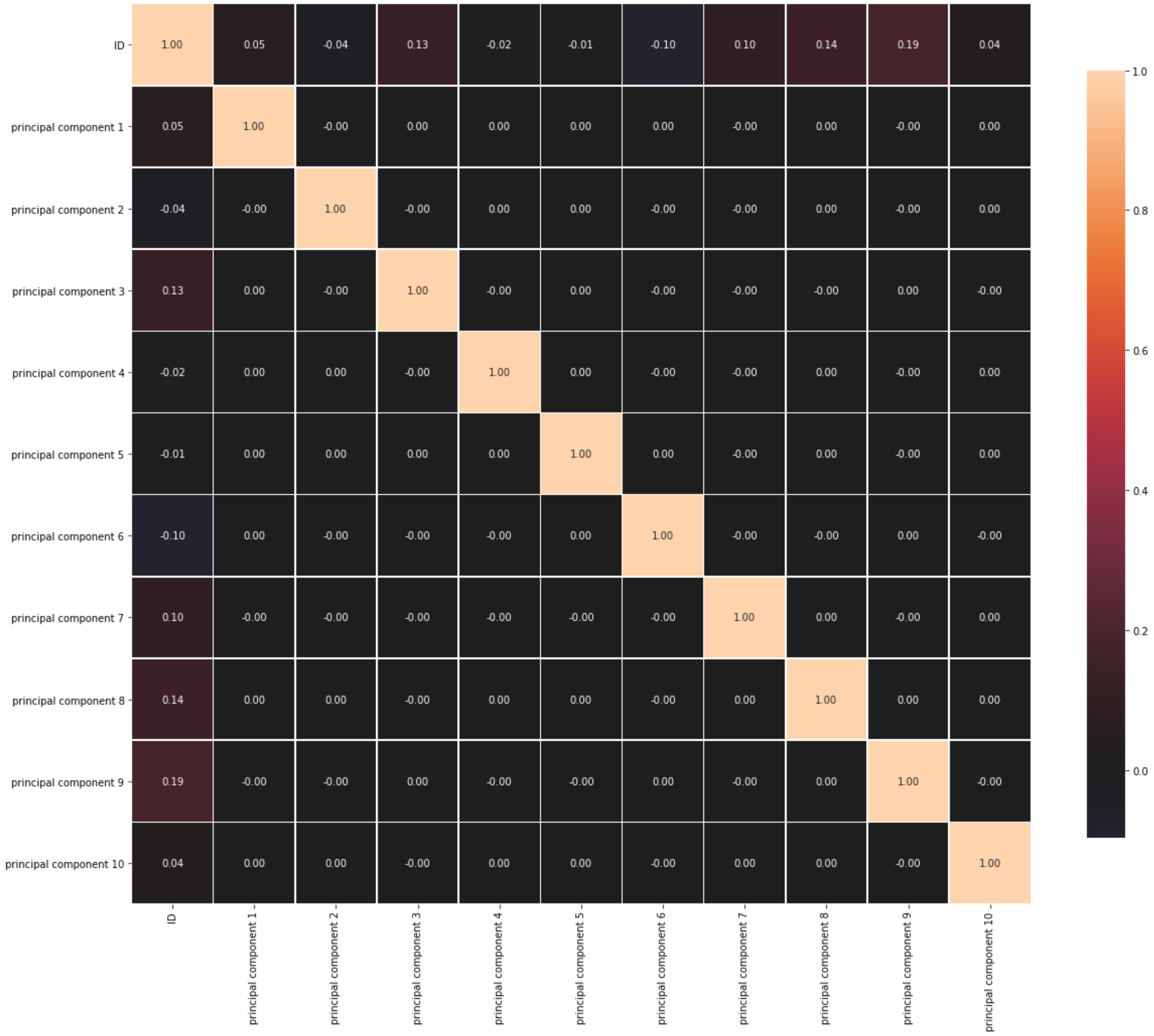

M class
attr 1
attr 2
attr 3
attr 4
attr 5
attr 6
attr 7
attr 8
attr 9
attr 10
attr 11
attr 12
attr 13
attr 14
attr 15
attr 16
attr 17
attr 18
attr 19
attr 20
attr 21
attr 22
attr 23
attr 24
attr 25
attr 26
attr 27
attr 28
attr 29
attr 30
4000
3000
2000
1000
0
0
100
200
300
400
500

Data Preprocessing:

- Split the data int train and test sets using StratifiedShuffleSplit
- **Normalization:**
    - **Z-score:**
        - Calculate z-score using `zscore(x) from scipy library`
        - **Conclusion**:
            - The result data spread on less scale, so values are more close to each other.
            - A value is exactly equal to the mean of all the values of the feature, it will be normalized to 0. If it is below the mean, it will be a negative number, and if it is above the mean it will be a positive number.

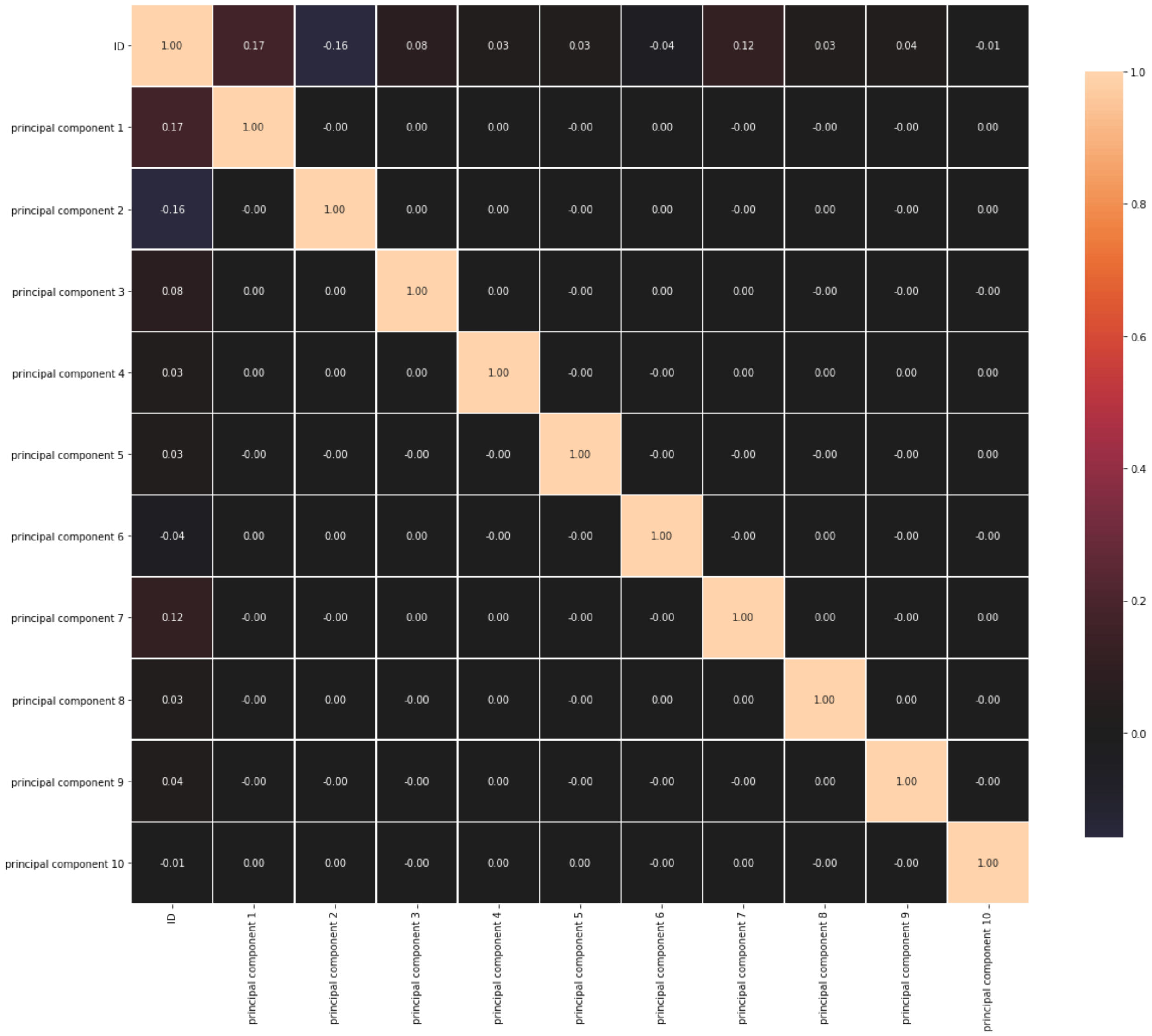
- **Dimensionality reduction**
    - Feature Projection:
        - Use `PCA from the scikit-learn` library seeking to sum of variance ratio we need of 0.95.
        - Number of PCA components = 10
        - Plot correlation matrix for result data.

Train Data



	ID	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10
ID	1.00	0.05	-0.04	0.13	-0.02	-0.01	-0.10	0.10	0.14	0.19	0.04
principal component 1	0.05	1.00	-0.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00
principal component 2	-0.04	-0.00	1.00	-0.00	0.00	0.00	-0.00	-0.00	0.00	-0.00	0.00
principal component 3	0.13	0.00	-0.00	1.00	-0.00	0.00	-0.00	-0.00	-0.00	0.00	-0.00
principal component 4	-0.02	0.00	0.00	-0.00	1.00	0.00	-0.00	-0.00	0.00	-0.00	0.00
principal component 5	-0.01	0.00	0.00	0.00	0.00	1.00	0.00	-0.00	0.00	-0.00	0.00
principal component 6	-0.10	0.00	-0.00	-0.00	-0.00	0.00	1.00	-0.00	-0.00	0.00	-0.00
principal component 7	0.10	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	1.00	0.00	-0.00	0.00
principal component 8	0.14	0.00	0.00	-0.00	0.00	0.00	-0.00	0.00	1.00	0.00	0.00
principal component 9	0.19	-0.00	-0.00	0.00	-0.00	-0.00	0.00	-0.00	0.00	1.00	-0.00
principal component 10	0.04	0.00	0.00	-0.00	0.00	0.00	-0.00	0.00	0.00	-0.00	1.00

Test Data



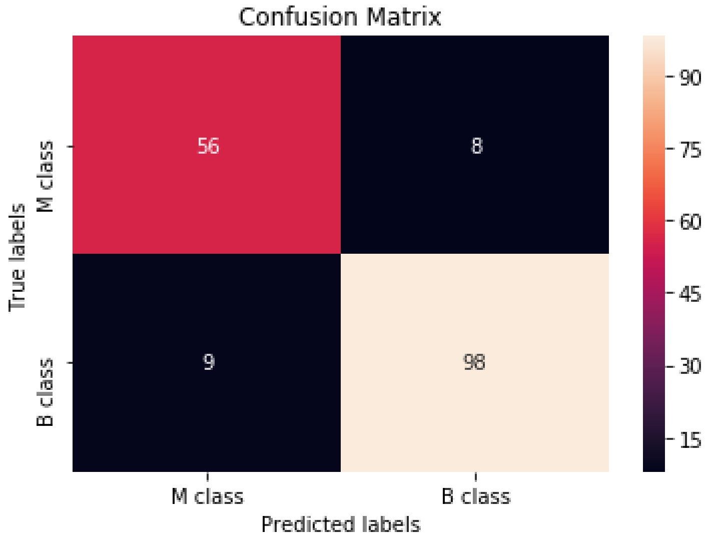
	ID	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10
ID	1.00	0.17	-0.16	0.08	0.03	0.03	-0.04	0.12	0.03	0.04	-0.01
principal component 1	0.17	1.00	-0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00	-0.00	0.00
principal component 2	-0.16	-0.00	1.00	0.00	0.00	-0.00	0.00	-0.00	0.00	-0.00	0.00
principal component 3	0.08	0.00	0.00	1.00	0.00	-0.00	0.00	0.00	-0.00	-0.00	-0.00
principal component 4	0.03	0.00	0.00	0.00	1.00	-0.00	-0.00	0.00	0.00	0.00	0.00
principal component 5	0.03	-0.00	-0.00	-0.00	-0.00	1.00	-0.00	-0.00	-0.00	-0.00	0.00
principal component 6	-0.04	0.00	0.00	0.00	-0.00	-0.00	1.00	-0.00	0.00	-0.00	-0.00
principal component 7	0.12	-0.00	-0.00	0.00	0.00	-0.00	-0.00	1.00	0.00	-0.00	0.00
principal component 8	0.03	-0.00	0.00	-0.00	0.00	-0.00	0.00	0.00	1.00	0.00	-0.00
principal component 9	0.04	-0.00	-0.00	-0.00	0.00	-0.00	-0.00	-0.00	0.00	1.00	-0.00
principal component 10	-0.01	0.00	0.00	-0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00	1.00

■ Conclusion

- Principal component analysis convert a set of observations of correlated variables into a set of values of linearly uncorrelated variables called principal components
- less than half of the attributes can describe the data with 0.95 percent.

Classification

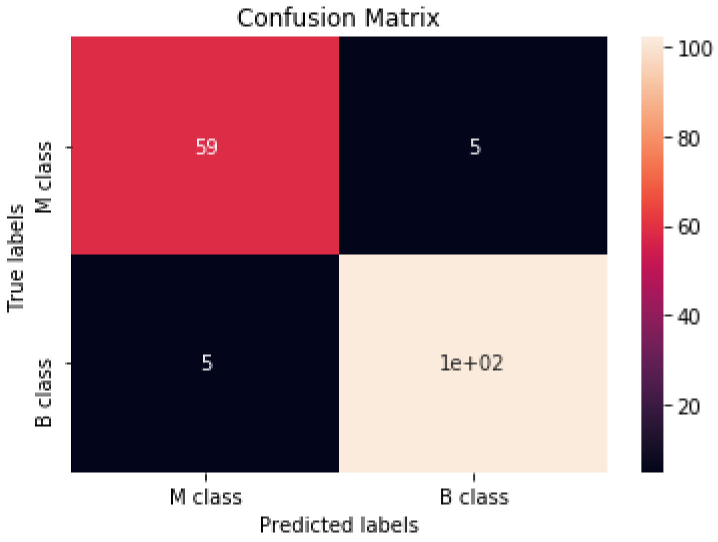
- ## Decision tree
  - Used GridSearchCV for tunning the depth parameter among the given values [5, 7, 10, 15, 20, 30] and get 5 as the best value.
  - Train the model with DecisionTreeClassifier from Sk-learn
  - Compute precision, recall, F-score using precision_recall_fscore_support from SK-learn
    - precision: 0.9193548387096774 for "M" class , 0.9357798165137615 for "B" class
    - recall: 0.890625 for "M" class , 0.9532710280373832 for "B" class
    - fscore: 0.9047619047619047 for "M" class , 0.9444444444444445 for "B" class
  - Compute Confusion matrix using confusion_matrix from SK-learn



Note

The same algorithm tested for depth = 15, 50 and less value for Evaluation matrices are obtained, so parameter tunning helped to get better accuracy.

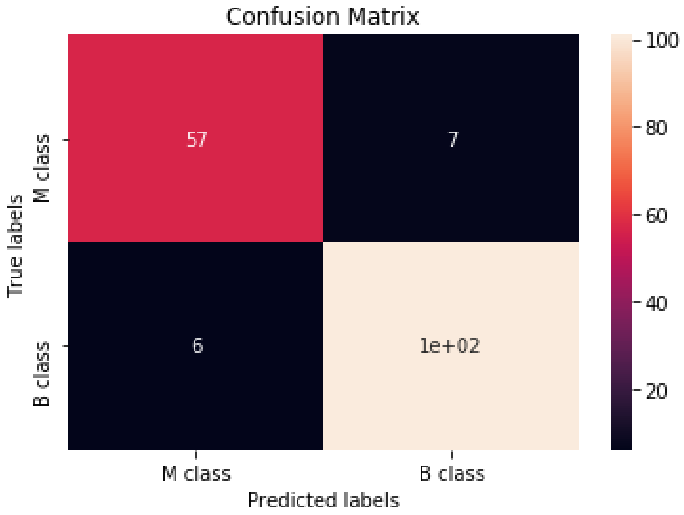
- **AdaBoost Classifier**
  - Used GridSearchCV for tunning
    - n_estimators in range [10, 50, 100, 200, 500]
    - learning_rate in range [0.25, 0.5, 1, 1.5, 2, 3]
    - Best value of learning_rate = 1.5
    - Best value of n_estimators = 200
  - Train the model with AdaBoostClassifier from Sk-learn
  - Compute precision, recall, F-score using precision_recall_fscore_support from SK-learn
    - precision: 0.9375 for "M" class , 0.9626168224299065 for "B" class
    - recall: 0.9375 for "M" class , 0.9626168224299065 for "B" class
    - fscore: 0.9375 for "M" class , 0.9626168224299065 for "B" class
  - Compute Confusion matrix using confusion_matrix from SK-learn



Note

The same algorithm tested for learning rate = 0.5, 2..5 and n_estimtors = 50, 300 and less value for Evaluation matrices are obtained, so parameter tunning helped to get better accuracy.

- **Random Forest Classifier**
  - Used GridSearchCV for tunning
    - depth in range [5, 10, 15, 20, 50]
    - n_estimators in range [10, 50, 100, 200, 500]
    - Best value of depth = 10
    - Best value of n_estimators = 50
  - Train the model with RandomForestClassifier from Sk-learn
  - Compute precision, recall, F-score using precision_recall_fscore_support from SK-learn
    - precision: 0.9016393442622951 for "M" class , 0.9181818181818182 for "B" class
    - recall: 0.859375 for "M" class , 0.9439252336448598 for "B" class
    - fscore: 0.88 for "M" class , 0.9308755760368663 for "B" class
  - Compute Confusion matrix using confusion_matrix from SK-learn



Note

The same algorithm tested for depth = 2, 20 and n_estimtors = 50, 200 and less value for Evaluation matrices are obtained, so parameter tunning helped to get better accuracy.

- **Conclusion**
    - All the models have very good values for the precession, recall, and F-score
    - The best model is the AdaBoost classifier as the number of wrongly classified samples is less = 9