University of Pisa

COMPUTER SCIENCE DEPARTMENT

Academic Year 2017/2018

Data Mining II

Final Report

Federica Trevisan - 568019

Contents

1 Time series

1.1 IBM dataset

The dataset *IBM stock* represents the trend of the stock values of the american company in the last 57 years, from 1962 until 2018. It has been obtained from Yahoo!Finance service. The objective of this first task is to split the time series into years, and study their similarities, also using clustering.

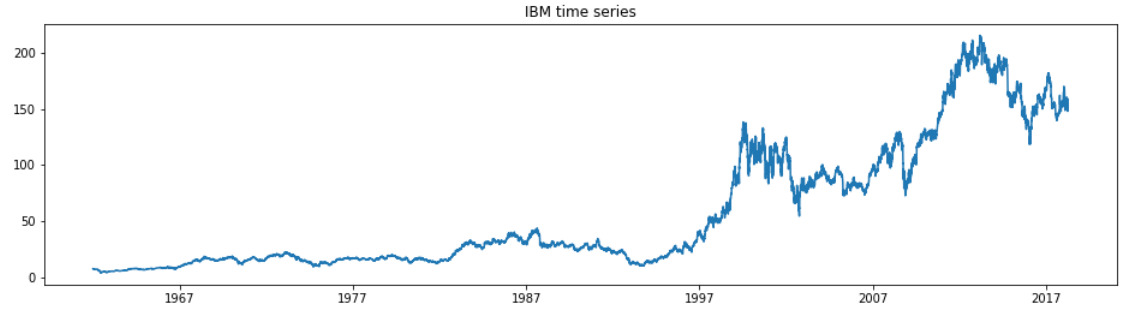


Figure 1: IBM dataset

Each record contains data describing the stock values and has 6 attributes:

- *Open* The price at the beginning of a trading day on the stock market;
- *High* means the highest price in a given period of time;
- *Low* is the lowest price in a given period of time;
- *Close* The price at the end of a trading day on the stock market;
- *Adj* stands for *Adjusted*. Using the closing price as a starting point, it takes into account factors such as dividends, stock splits and new stock offerings.
- *Volume* Is the amount of shares bought/sold of a stock in a given period of time.

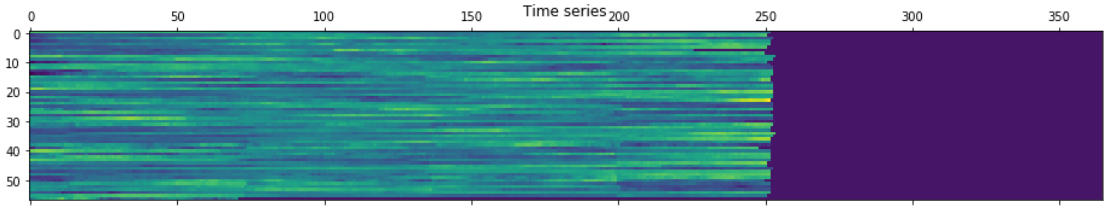


Figure 2: Time series heatmap

1.2 Data preparation

Before proceeding with the analysis, some changes have been done to the dataset:

- The time series have been splitted into years, with 57 series resulting;
- Since there are data only the first 70 days of the year 2018, it hasn't been considered for the analysis;
- From the time series the value equal to -3 has been removed;
- In order to get better results in the clustering and in the similarity study, the values have been normalized with *offset translation*, *amplitude scaling* and *smoothing*.

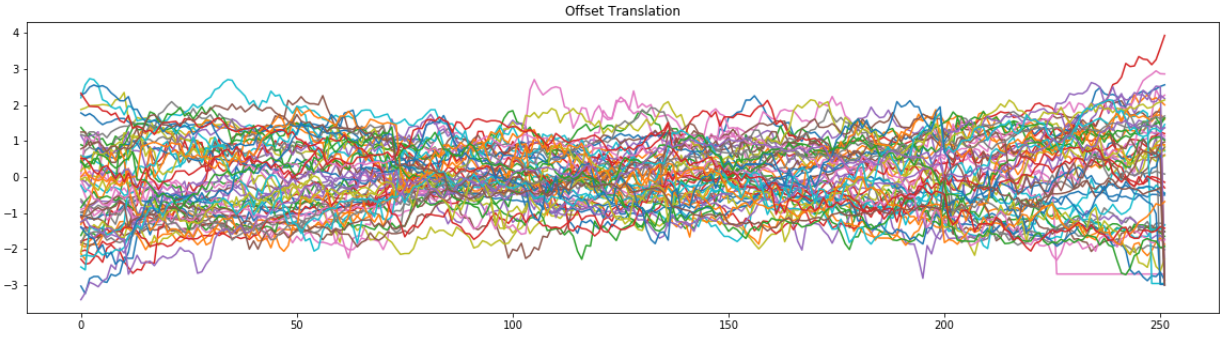


Figure 3: Offset Translation

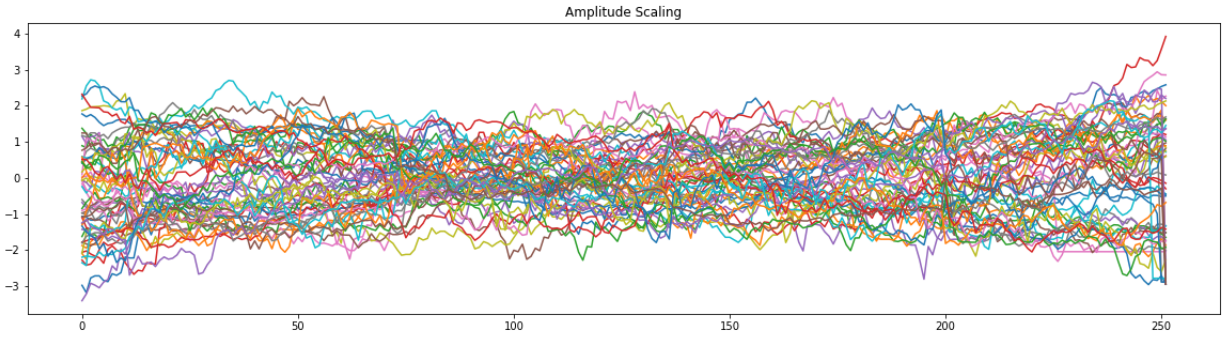


Figure 4: Amplitude scaling

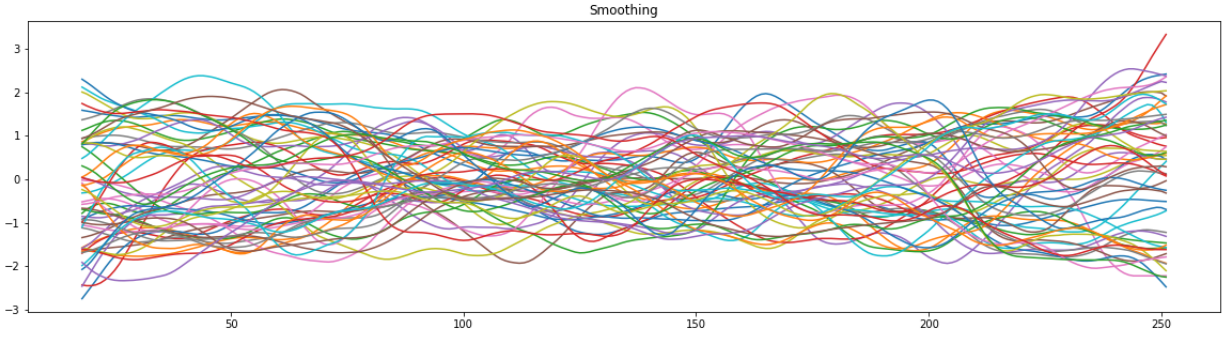
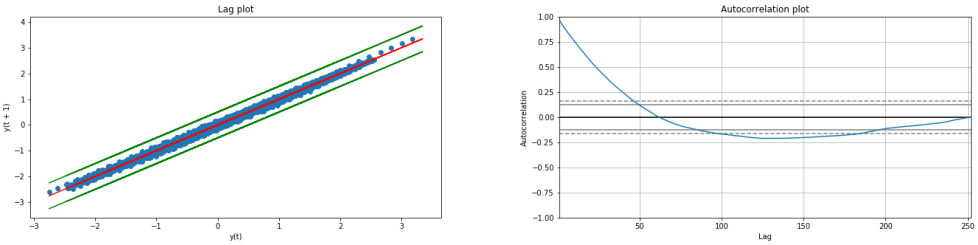


Figure 5: Smoothing

From the autocorrelation analysis, it is possible to notice that the dataset doesn't have randomly generated values, From the autocorrelation plot it is visible that there are many positive values of autocorrelation.



(a) Lag plot

(b) Autocorrelation plot

Figure 6: Lag and autocorrelation plot

1.3 Analysis with Dynamic Time Warping

In order to study the similarity of the time series the clustering algorithm DBSCAN has been chosen.
The density based algorithm has been applied on two different distance measures: Dynamic Time Warping (DTW) and Eucledian distance.
The results of the clusters found with the two different distance measures have been compared. In this paragraph the clusters found with DTW are analyzed.

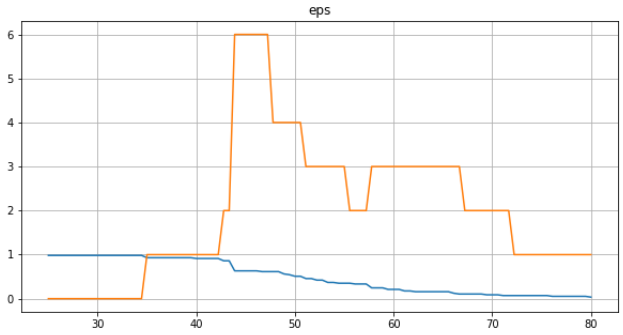


Figure 7: Dynamic Time Warping

In order to apply the DBSCAN algorithm, it is important to identify the right value of ε based on the value of K.
After a few trials, the value chosen for ε is equal to 48. It has been chosen by taking into account the resulting curve in Figure 7, the highest number of clusters can be found in the interval of ε between 40 and 50.
The value chosen for K is equal to 2, the algorithm has been computed using the DTW distance and six main clusters have been identified.

Fixing **eps = 48 e minPts = 2**, 6 clusters have been obtained:

- Cluster 0 -> ['1962', '1981', '2002']
- Cluster 1 -> ['1963', '1965', '1967', '1969', '1982', '1983', '1994', '1995', '1997', '1998', '2009']
- Cluster 2 -> ['1964', '1987', '2008']
- Cluster 3 -> ['1970', '2005']
- Cluster 4 -> ['1973', '1974', '1979', '1986']
- Cluster 5 -> ['2013', '2015']

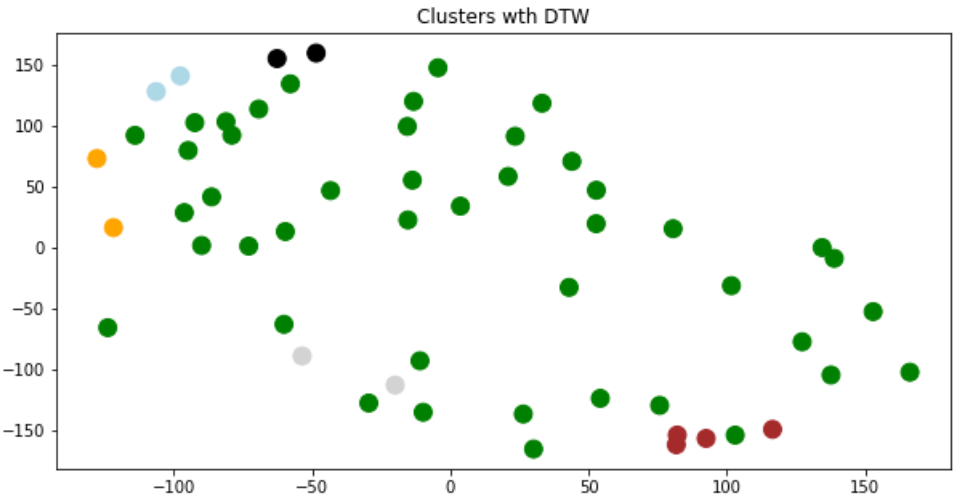


Figure 8: Scatterplot clusters DTW

As it emerges from the plot 8, all clusters are relatively dense and well separated; the years that are not inside a cluster are noise points.
From the following plots it can be noticed, that the clusters' time series tend to have the same trend; in some cases there are picks.

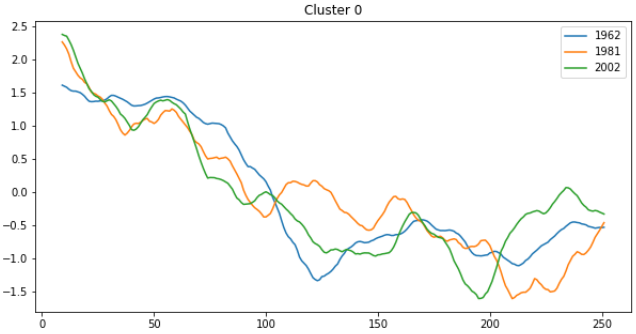


Figure 9: In the Cluster 0 the years 1962, 1981, 2002 have high value in the first months, that slightly decreases with a small increase at the last days of the years.

Figure 10: In the Cluster 1 there's a general linear increasing trend.

Figure 11: The years 1964, 1987, 2008, in the Cluster 2, after an excellent beginning, are characterized by a rapid decrease.

Figure 12: In the Cluster 3, the years 1970 and 2005 have a slight decrease followed by a slow increase.

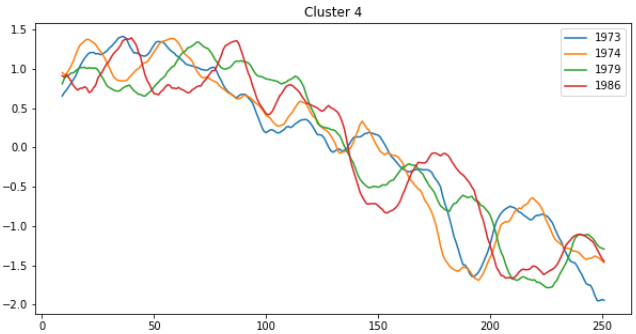


Figure 13: The years 1973, 1974, 1979 and 1986 of the Cluster 4 start at the beginning with high values, with a progressive decrease.

Figure 14: The years 2013 and 2015 have a similar trend with the cluster 4. The main difference is that in this case there are more pronounced picks.

Figure 15: An opposite situation emerges from the Clusters 6 with the years 1983 and 1995, they start with low values at the first months that increase gradually. Also in this case there are a few picks.

-

1.4 Analysis with Eucledian distance

The DBSCAN algorithm has been also applied using eucledian distance.

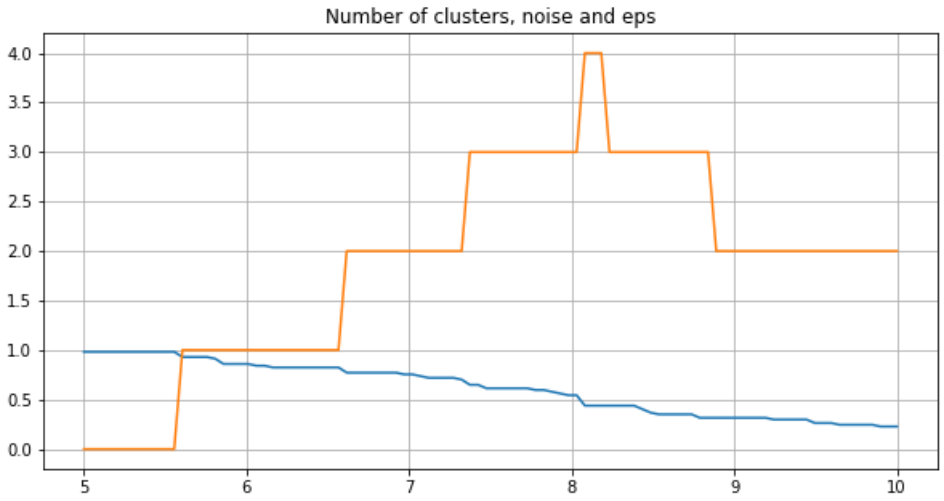


Figure 16: Epsilon eucledian distance

Fixing **eps=8.2 minPts=2** the number of clusters found is 5:

- Cluster 0 –> ['1962', '1991', '2002', '2017']
- Cluster 1 –> ['1964', '2015']
- Cluster 2 –> ['1965', '1967', '1969', '1982', '1983', '1994', '1995', '1996', '1997', '1998', '2003', '2006', '2007', '2009', '2010', '2011', '2016']
- Cluster 3 –> ['1973', '1974', '1979', '1981', '1986', '2013']
- Cluster 4 –> ['1987', '1992', '2008', '2014']

In the following plot the resulting clusters are mapped. Differently from DTW, a big cluster emerges at the top left corner in orange, the cluster number 2, and other small clusters at the bottom.
The small clusters don't seem to be dense and well separated, there's also a good presence of noise.

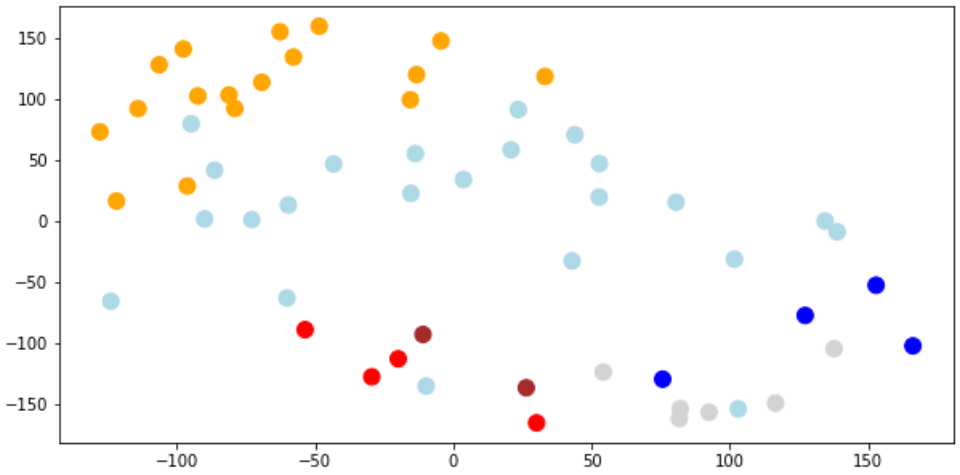


Figure 17: Scatterplot clusters eucledian distance

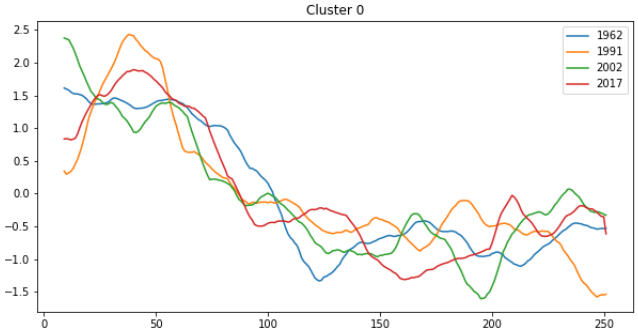


Figure 18: Cluster 0, Years 1962, 1991, 2002, 2017, is similar to cluster 0 found with DTW.

Some clusters found with Eucledian Distance don't have a high level of internal cohesion. In conclusion, even if DTW has a higher complexity in terms of computation, it seems to give better performances. The DTW method has given better results than eucledian distance, comparing the time series in a more accurate way.

Figure 19: Cluster 1, 1964, 2015

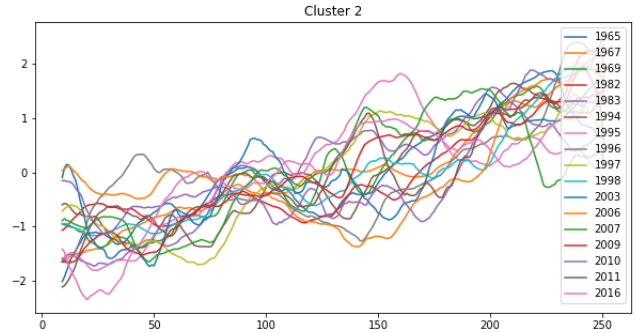


Figure 20: Cluster 2, is similar to cluster 1 found with DTW, with the general increasing trend.

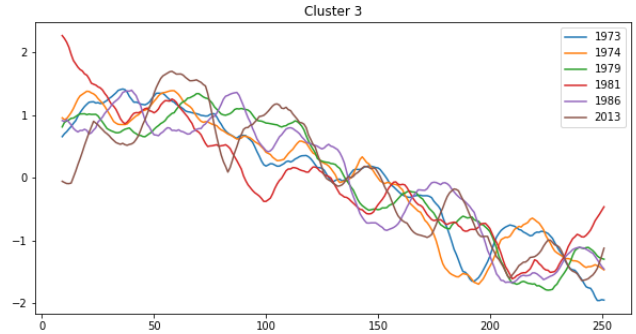


Figure 21: The Cluster 3 1973, 1974, 1979, 1981, 1986, 2013 is similar to the cluster 4 found with DTW, with high values slowly decreasing.

Figure 22: Cluster 4, 1987, 1992, 2008, 2014 is similar to the cluster 2 found with DTW.

2 Sequential patterns

The objective of this task is to discover patterns over the IBM stock value time series in order to find Motifs-like patterns: frequent contiguous subsequences of length at least 4 days.
In order to complete this task, the data have been splitted into monthly time series, preprocessed and then the algorithm SPAM, Sequential Pattern Mining, has been applied.

2.1 Preprocessing

In the preprocessing phase, the dataset has been splitted into **monthly time series**. The resulting 676 time series, corresponding to every month from 1962 to 2018 have been normalized with Z-normalization and have been discretized using the SAX algorithm, that transforms a sequence of rational numbers into a sequence of letters. Its source code can be found at this link: `https://github.com/seninp/saxpy`.
Before applying SAX, from the monthly time series have been removed the last 9 values, corresponding to the last 9 days of the month because all taking values equal to **-3**. The resulting dataset is composed from monthly times series of 22 values on average, like the working days in a month.
So the algorithm **SAX**, Symbolic Aggregate approXimation, can be applied, and it consists of two steps:

1. It transforms the original time-series into the PAA representation. PAA stands for Piecewise Aggregate Approximation, its purpose is to reduce the dimensionality of the dataset, dividing the original time-series into M equally sized frames and secondly computing the mean values for each frame.

2. It converts the PAA data into a string. The main idea is that the time series values take follow the shape of a gaussian distribution; the symbols have been chosen in a way that they represent different intervals.

For this analysis it has been chosen an alphabet of 7 values (a, b, c, d, e, f, g), in this way the dataset is splitted in 7 bins, each one corresponding to one letter.

2.2 SPAM Algorithm

In order to discover the motifs int the dataset, the tool SPMF has been used; an open-source data mining library written in Java, specialized in pattern mining `http://www.philippe-fournier-viger.com/spmf/index.php`.
It has been chosen to apply the algorithm SPAM, Sequential Pattern Mining, with the parameters:

- min pattern length equal to 4, in order to obtain subsequences of at least 4 days;
- max gap fixed to 1, in order to obtain continuous subsequences.

The patterns that have been discovered tuning different values of minsup are:

Patterns	**Support**
baaa	113
ffff	106
fffg	102
ffgg	153
fggg	184
gfff	107
ggff	121
gggf	147
gggg	172

Table 1: Minimum support equal to 15%

Patterns	**Support**
ffgg	153
fggg	184
gggf	147
gggg	172

Table 2: Minimum support equal to 20%

Patterns	**Support**
fggg	184
gggg	172

Table 3: Minimum support equal to 25%

Changing the values of minimum support, no subsequences of length 5 or more ave been found. The most frequent patterns (fggg and gggg) consist in a subsequence of length 4 and items with high values, identified by the letters f and g.

3 Alternative classification methods

3.1 Data Understanding

The dataset used for the next two tasks is the Abalone dataset, downloaded from the UCI Machine Learning Repository at this link: `https://archive.ics.uci.edu/ml/datasets/Abalone`.
It contains various features of abalones, a type of gastropod shellfish, known by its colorful inside shell.
The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope, a time-consuming task.
The main purpose of this task is to use alternative classification methods to predict the age of the abalone.
The abalone dataset comes from an original (non-machine-learning) study and it is composed of:

- Number of Instances: 4177
- Missing Attribute Values: None

The following table (Table 1) describes each attribute:

Name	**Data Type**	**Measurement Unit**	**Description**
Sex	nominal	-	M, F, and I (infant)
Length	continuous	mm	longest shell measurement
Diameter	continuous	mm	perpendicular to length
Height	continuous	mm	with meat in shell
Whole weight	continuous	grams	whole abalone
Shucked weight	continuous	grams	weight of meat
Viscera weight	continuous	grams	gut weight (after bleeding)
Shell weight	continuous	grams	after being dried
Rings	integer	–	+1.5 gives the age in years

Table 4: Dataset attributes

The meaning of the attributes is now briefly introduced:

- *sex* is the categorical variable that indicates the sex of the abalone. M stands for male, F for female, I for infant;
- *length* is the continuous numerical variable, it shows the longest shell measurement in millimeters;
- *diameter* is the continuous numerical variable, is measures the diameter of the shell perpendicular to the length in millimeters;
- *height* is the continuous numerical variable, it corresponds to the height of the abalone in millimeters, with meat in shell;
- *whole_weight* is the continuous numerical variable, it indicates the whole weight of the abalone in grams;
- *shucked_weight* is the continuous numerical variable, that measures the weight of the meat in grams;
- *viscera_weight* is the continuous numerical variable, that measures the weight of the meat after bleeding in grams;
- *shell_weight* is the continuous numerical variable, that measures the weight of the shell after being dried in grams;
- *rings* is the variable that +1.5 gives the age in years of the abalone. It takes values between 1 and 29. This attribute is the target of this classification task.

A correlation matrix (Fig. 1) has been computed, and it shows overall a strong correlation (index higher than 0.77) between the attribute pairs. The strongest correlation emerges between *length* and *diameter*, and between *weight_whole* and *weight_shucked*, *weight_viscera* and *weight_shell* respectively. This is a normal situation that describes that, growing the length and the diameter, grows also the weight of the abalone. The lowest correlation is seen between *weight_shucked* and *height*.

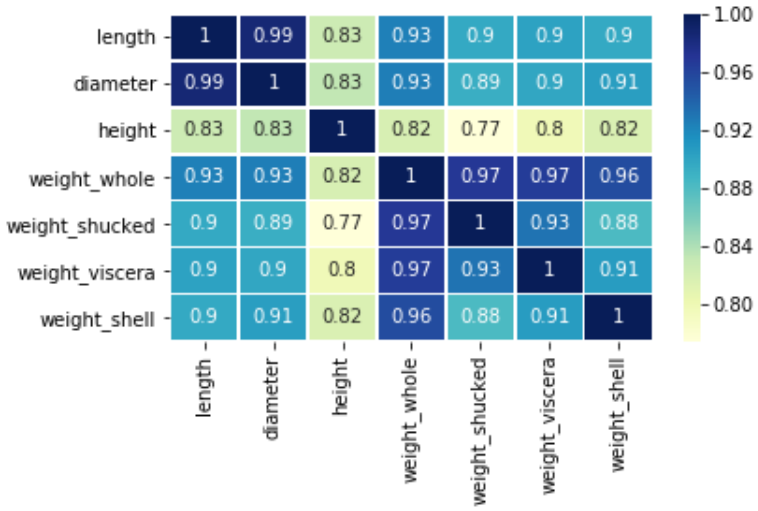
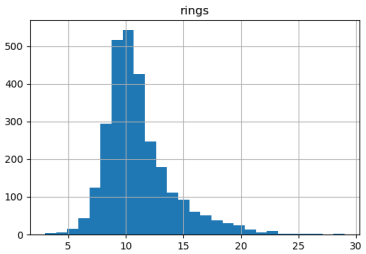


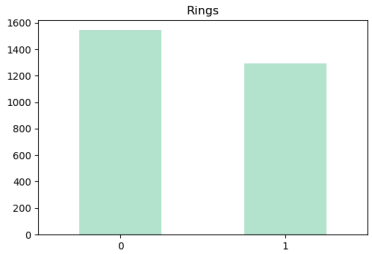
Figure 23: Correlation Matrix

3.2 Data preparation

In order to apply the classification models, some changes have been done to the dataset.
The "Infant" abalone observations have been discarded, in this way the attribute *sex* can take values Male or Female decreasing the datasets records from 4177 to 2835.
Discarding the Infant abalones the minimum value of the attribute *rings* is 3, and not 1 as it was at the beginning, leveraging the minimum age.
All attributes except for *sex* have been considered for the classification task.
Sex is a categorical attributes, and it makes no sense to assign numerical values to it because it's not incremental, so it has been scrapped for the analysis.
The target attribute *rings* has been discretized in order to achieve the classification task. The distribution can be approximated to a gaussian; the two bins have been obtained dividing the dataset in two balanced classes: rings less than 11 - labelled as 0, rings equal o more than 11 - labelled as 1.



(a) Rings histogram



(b) Rings discretized

Figure 24: Rings attribute, distribution and discretization

3.3 Classification models

The objective of this task is to try at least 3 different classification methods on the abalone dataset, using the discretized number of rings as class, and evaluating them with cross-validation.
Different alternative classification models have been trained, tested and validated with Python *sklearn* library; here they are explained in detail.

- **Decision Tree**
  Decision Trees are a supervised learning method used for classification. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It's simple to understand and to interpret.
  After a few trial, the maximum depth of the tree has been fixed to 5 levels, in this way the tree is readable and risk of overfitting is low. The split criteria used is Gini, even if there are no main differences between Gini and Entropy.

- **Random Forest**
  Random forest is an *ensemble* learning method for classification, that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. In other words, random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.
  In this case the number of estimators chosen is 50, also in this case there are no substantial differences between entropy and Gini splitting criteria.

- **K-Nearest Neighbors**
  This model classifies an object by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.
  The optimal value of K has been fixed after a few trials: k=13 is the value that allows to obtain the best results.

- **Neural Network**
  This model is inspired by the way biological neural networks in the human brain process information. It's an assembly of interconnected nodes and weighted links. The output node sums up each of its input value according to the weights of its links.
  In this model the number of hidden layers has been fixed to 5 and to 2 hidden units.

- **Gaussian Naive Bayes**
  The Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions.
  Default parameters have been used.

Model	**AUC**
Decision Tree	0.67
Random Forest	0.71
K-NN	0.67
Neural Network	0.72
Naive Bayes	0.58

Table 5: Area Under the Curve with cross validation

In order to evaluate the results with **cross validation** the dataset has been partitioned into 10 equal-sized folds: one is used as test set, while the remaining nine are used as training set.
The process is repeated 10 times, using each fold exactly once as test set, and the resulting 10 accuracy values are then averaged to get a single accuracy value.
The ROC (Receiver Operating Characteristic) curve of the models considered is provided here (Fig. 3).

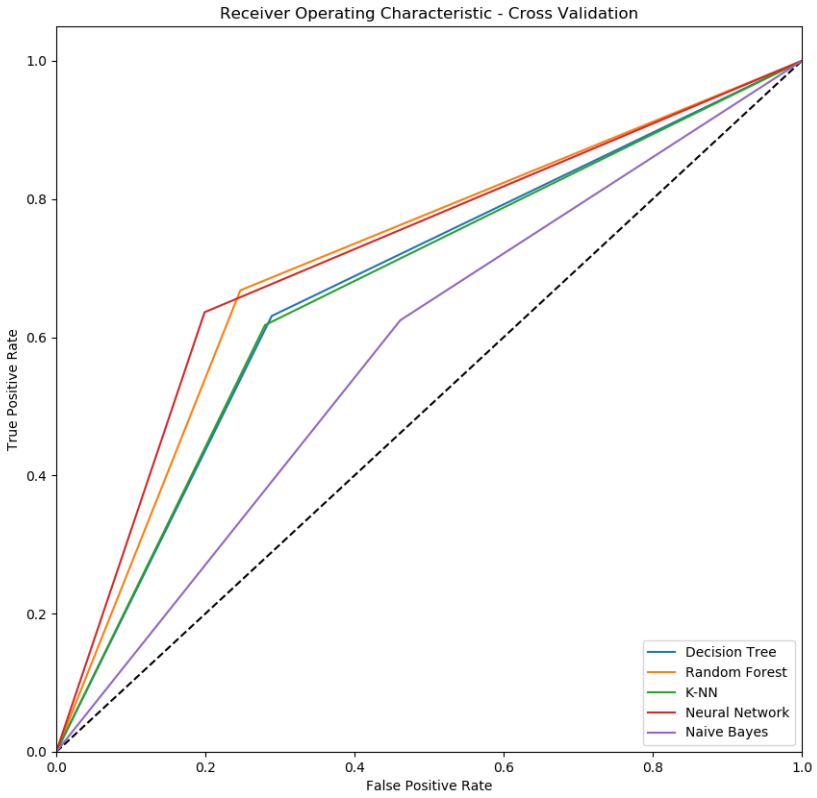


Figure 25: Model comparison

The closer to the top left corner of the plot the curve is, the higher the AUC (Area Under Curve) is, and so are the general performances.
The AUC of the Neural Network is 0.72 and for the Random Forest is equal to 0.71, which is relatively high, since the maximum possible value is 1 in the case of a perfect classifier.
The dotted line in the plot represents a "random guessing" classifier and its AUC is equal to 0.5.
Plotting together the ROC curves of the classification models it's easier to compare the performances.
From the ROC curves analysis, the best results are given from the Neural Network and Random Forest, with almost equal precision; while a good performance is given also from the K-NN algorithm and Decision Tree.
The worst results are given from the Naive Bayes classifier.
In conclusion, none of the previous models can solve with high accuracy the binary classification problem. They are not able to replace the manual counting of the number of rings through a microscope for determining the age of abalone.
Maybe the problem could have been seen as a non-binary classification problem, in order to give better results.

4 Outliers detection

The objective of this section is to identify the top 1% outliers from the complete Abalone dataset. Considering the complete dataset, without discarding the Infant Abalones, the records are 4177. This means that the number of outliers that need to be identified is more or less 42.
The methods that have been adopted for this analysis are DBscan, Local Outlier Factor and Convex Hull, implemented with *sklean* Python library.
The attribute *Sex* has been discarded in this analysis.

4.1 DBscan

DBscan is a distance-based approach that judges a point based on the distance to its neighbors. The basic assuption is that normal data objects have a dense neighborhood, while outliers are far apart from their neighbors, having a less dense neighborhood.
Given a radius ε, and a percentage π, a point p is considered an outlier if at most π percent of all other points have a distance to p less than ε.
In this case, the parameters chosen are:

- minimum points = 5 (absolute value of π)
- ε = 0,33

The epsilon plot (Fig. 4) has made it possible to identify the right value of ε, which is 0.33, by looking at the coordinates corresponding to the elbow.

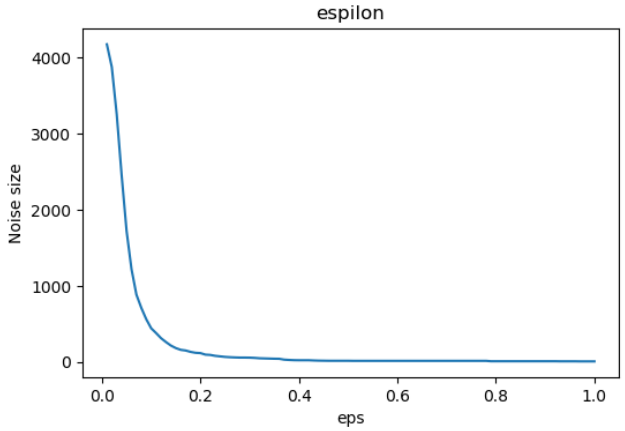


Figure 26: Noise and epsilon

The points that has been classified as outliers are the following:
[129, 163, 165, 166, 167, 232, 236, 270, 275, 294, 313, 314, 351, 355, 358, 433, 480, 501, 673, 674, 675, 678, 719, 811, 882, 891, 1039, 1209, 1216, 1426, 1427, 1428, 1763, 2051, 2090, 2108, 2161, 2201, 2209, 2305, 2334, 2335, 2436, 2627, 2811, 3007, 3149, 3280, 3427]

4.2 Local Outlier Factor

This example presents the Local Outlier Factor (LOF) estimator. The LOF algorithm is an unsupervised outlier detection method which computes the local density deviation of a given data point with respect to its neighbors. It considers as outlier samples that have a substantially lower density than their neighbors.
The parameter number of neighbors considered, n_neighbors = 20 appears to work well in general.
The points that has been classified as outliers with LOF are the following:
[236, 237, 238, 306, 480, 520, 526, 719, 720, 860, 894, 1039, 1044, 1054, 1210, 1216, 1228, 1264, 1426, 1429, 1755, 2051, 2115, 2123, 2127, 2371, 2619, 2627, 2641, 2711, 2728, 2790, 2811, 2975, 3086, 3472, 3521, 3716, 3718, 3801, 3814, 3996]

4.3 Convex Hull

This depth-based approach assumes that extreme points are outliers. The basic assumption is that outliers are located at the border of the data space, while normal objects are in the center of the data space.
In this way, the data objects are organized in convex hull layers, and outliers are objects on outer layers.

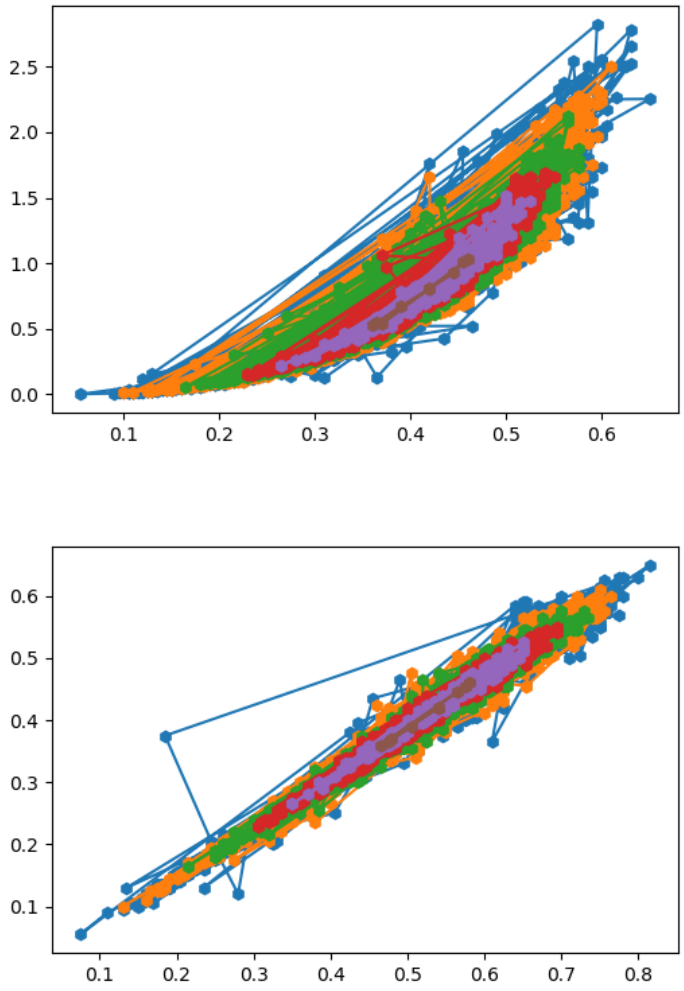
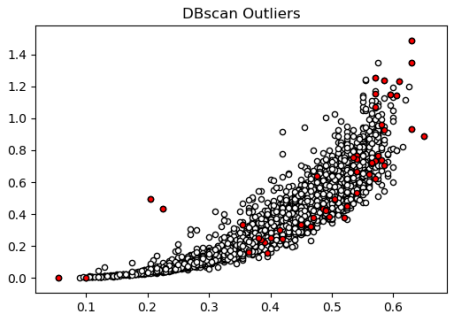


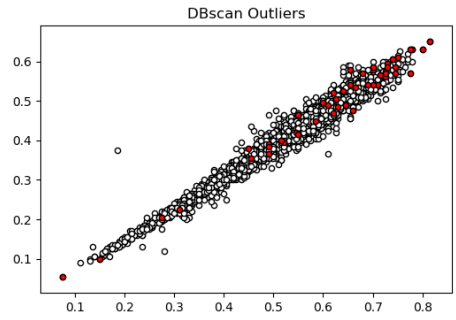
Figure 27: Convex Hull

4.4 Conclusion

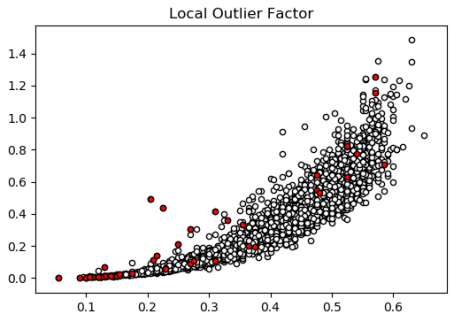
After applying the 3 different approaches for the outlier detection, it emerges that with DBSCAN and LOF it is possible to find 9 points in common:
[236, 480, 719, 1039, 1216, 1426, 2051, 2627, 2811].



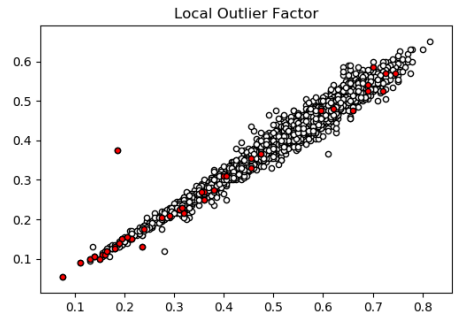
(a) DBSCAN



(b) DBSCAN



(a) LOF



(b) LOF

Figure 28: Outliers comparison