Data Mining

WALT DISNEY MOVIES

Vahid Ramezani, Sara Asadi | Data Mining | Feb 2022

Abstract

Objective: deploying data-mining algorithms on a data set of movies produced by Walt Disney, comparing effects of different methods and parameters and providing reliable results.

Methods and Materials: we studied data and features extracted, integrated and collected in a dataset, from movies produced by Walt Disney CO. The data provides us with some information about common known properties of a real-world movie. In this project, we took advantage of a wide variety of data mining, statistical analysis and machine learning algorithms to extract knowledge from raw data.

Results: Further, in this article, results of applying different data mining and machine learning algorithms are provided for comparison, often alongside intuitive tools.

Conclusion: All the algorithms discussed in this article including data pre-processing, statistical analysis, feature extraction, machine learning, data mining, etc. can be helpful hidden extracting knowledge from raw data.

INTRODUCTION

Data mining is playing an important role in computer science. Data mining and machine learning algorithms are frequently used to bring new invention to this world and to bring comfort to humanity. If looked deep enough, computer science, specifically data science usage and effects can be seen, not too far from our home.

We, and all of our data scientist friends, are trying to use different and suitable methods to provide new ways and new invention. In this article, we are going to give a report of our effort and endeavor to implement our theatrical knowledge for best use.

Further, you will find some steps we took to deploy different algorithms on the dataset and you can observe and compare the results.

Material and Methods

We studied data and features extracted, integrated and collected in a dataset, from movies produced by Walt Disney CO. The data provides us with some information about common known properties of a real-world movie. In this project, we took advantage of a wide variety of data mining, statistical analysis and machine learning algorithms to extract knowledge from raw data.

Python, as one of the most powerful programming languages and an inseparable tool for data science real-world deployment and data scientist, was out main coworker through this survey.

Data pre-processing algorithms such as different normalization methods, data cleaning methods, data integration, dimensional reduction, numerosity reduction and various machine learning models and algorithms like K-means, DB Scan, KNN, SVM, MLP, Bayes, Decision Tree, etc. have been used and mentioned in this report.

Results

Results are presented in different categories based on the order of steps taken through the project.

PHASE 1

Know Data

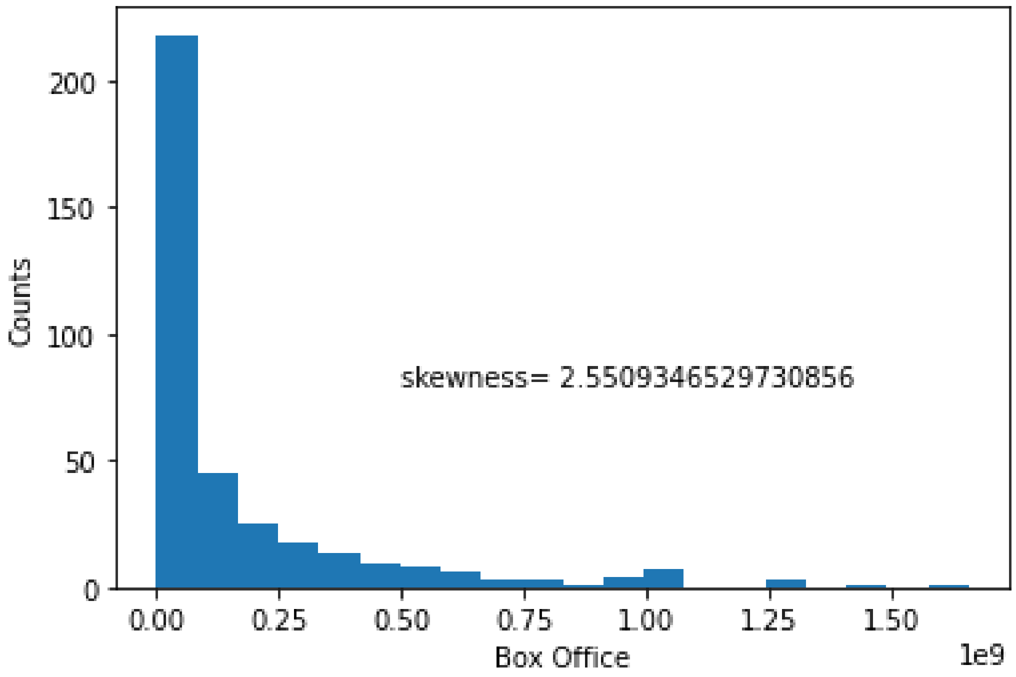
The dataset consist of 452 data tuples each having 26 attributes. Most of the attributes are nominal and are hardly suitable for data science algorithms. Some of the attributes have more than 50-60 percent missing data. There are some duplicate attributes.

Skewness

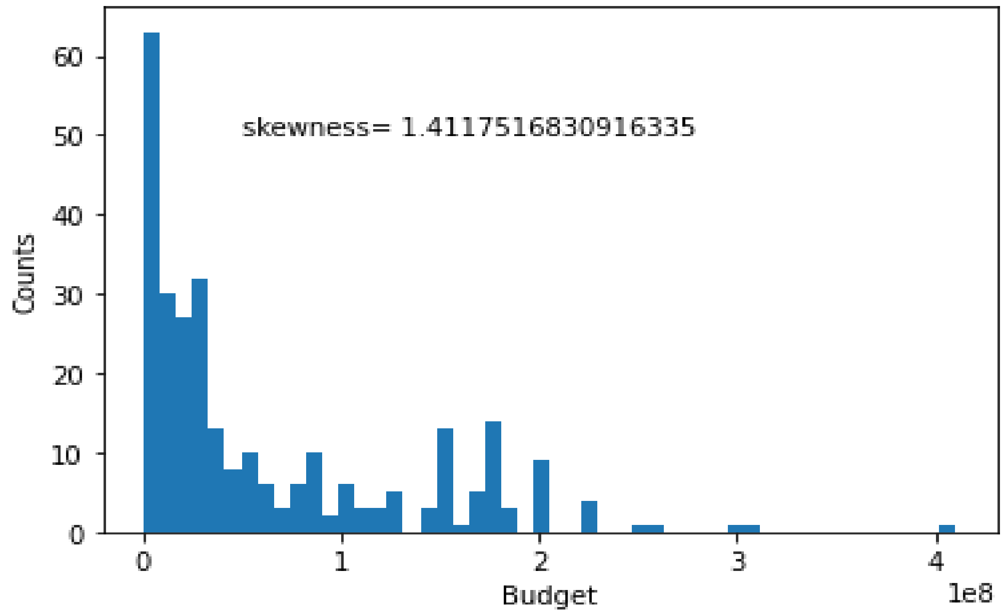
You can simply observe the skewness computed for different attributes here.

Results are as follows:

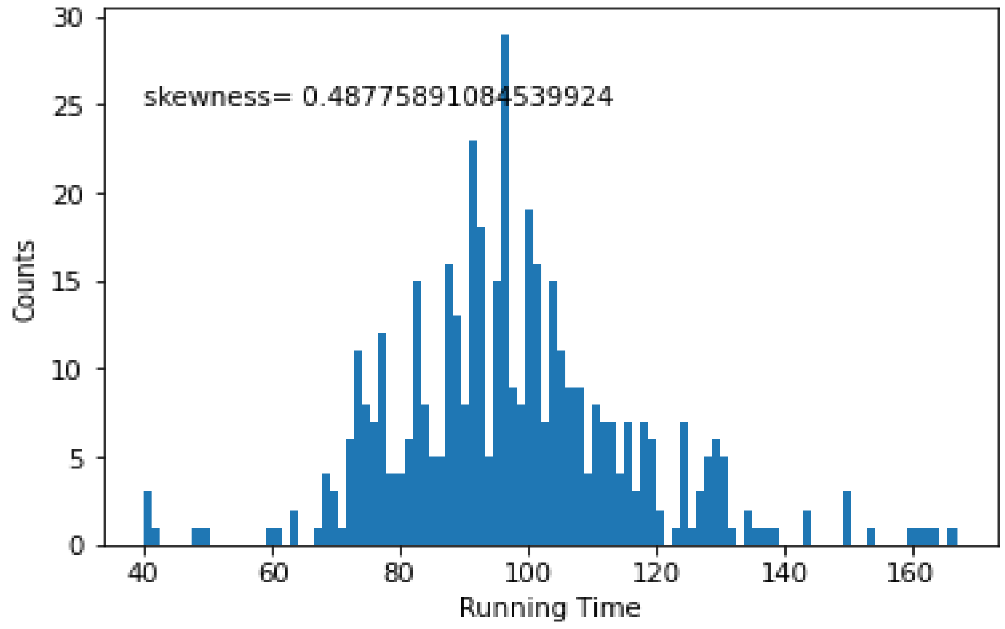
Box Office



Budget

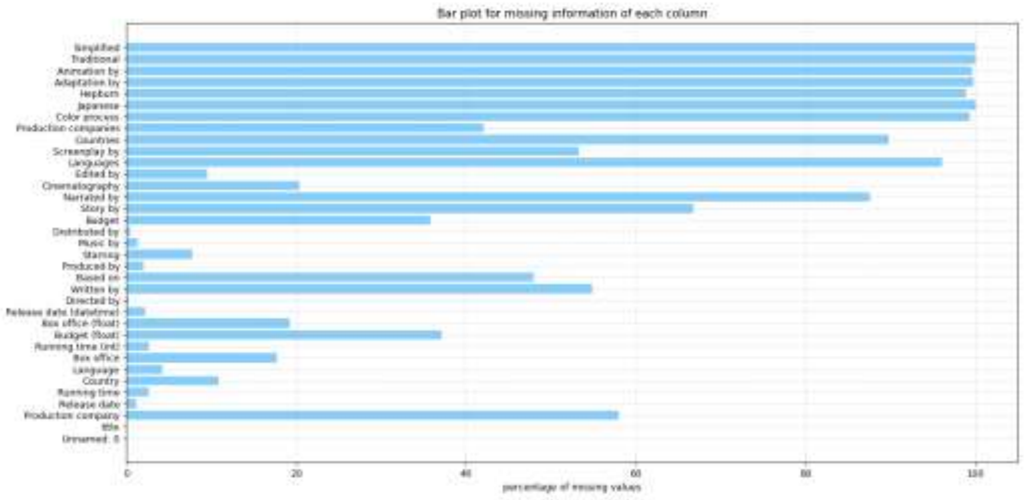


Running Time



Data Correctness and Completeness

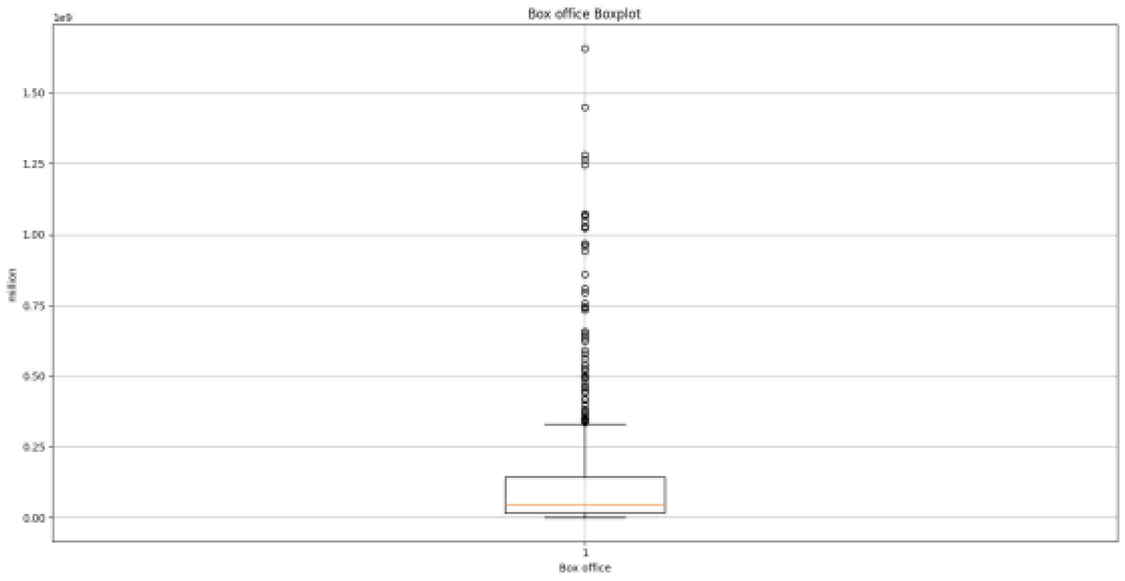
Using human senses and statistical factors, the correctness and completeness of data have been analyzed. Data cleaning step have been taken based on the results of the aforementioned analysis. You can see below the visual results of the analysis.

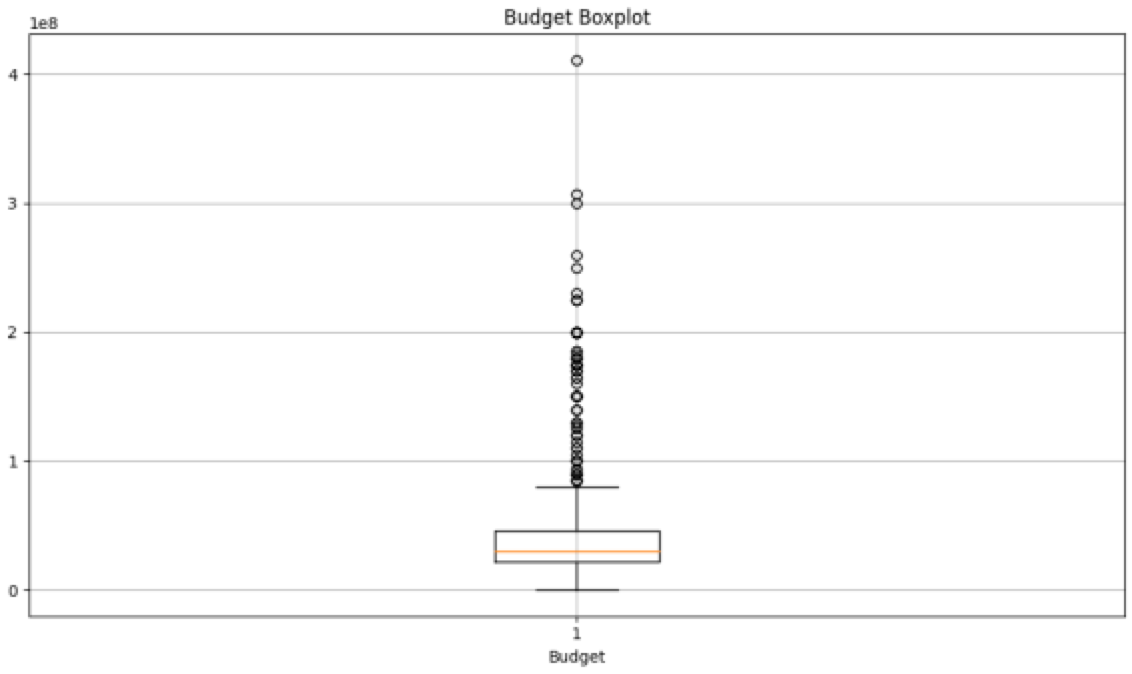


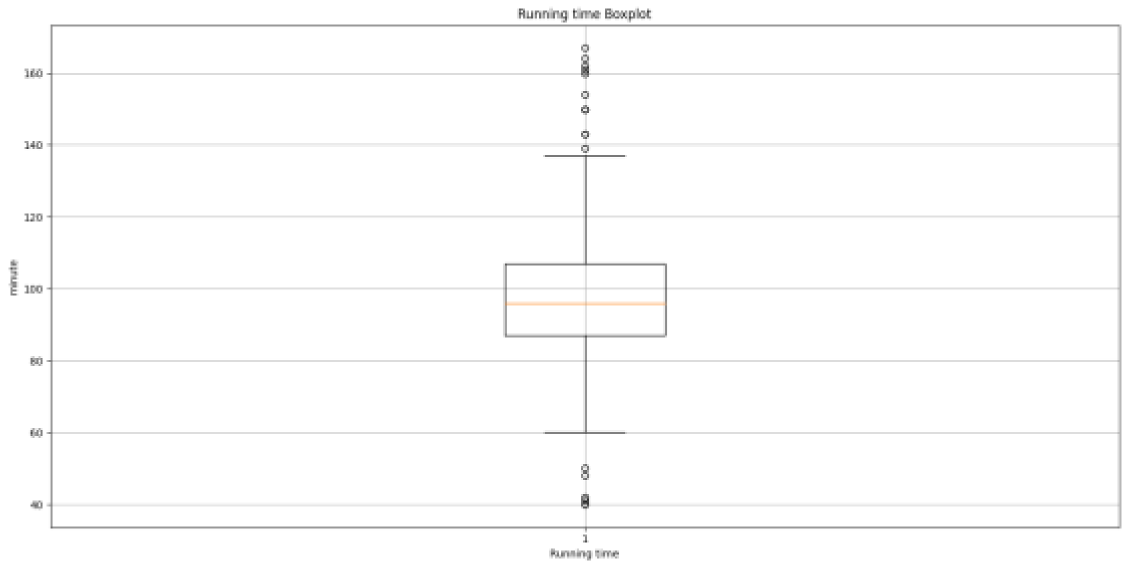
Outlier Detection

Using **box plot** tool, the valid range of values for data has been computed and data records above the upper quartile and below the lower quartile have been removed.

Box plots are shown below.





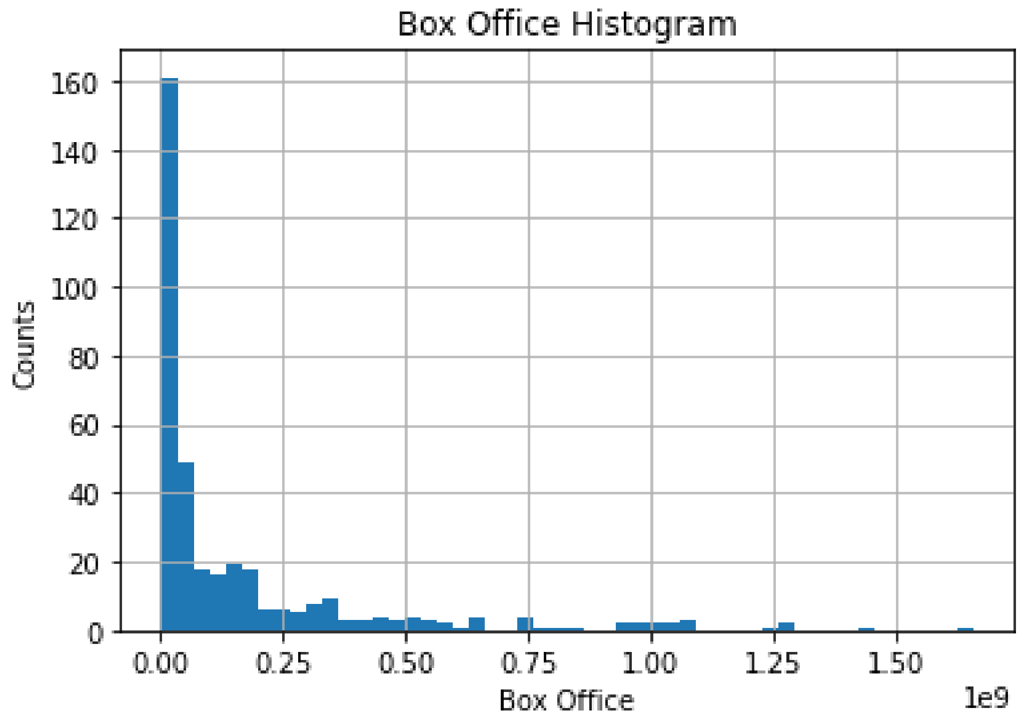


Histogram

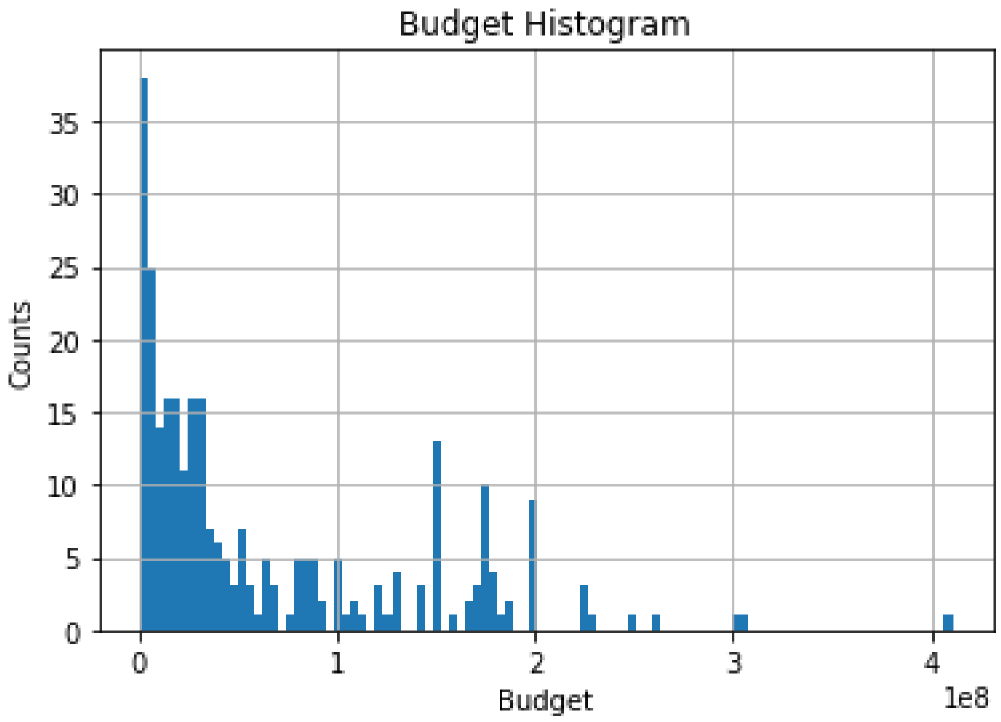
Below you can see histogram of some of the data attributes.

You can find the computation code here.

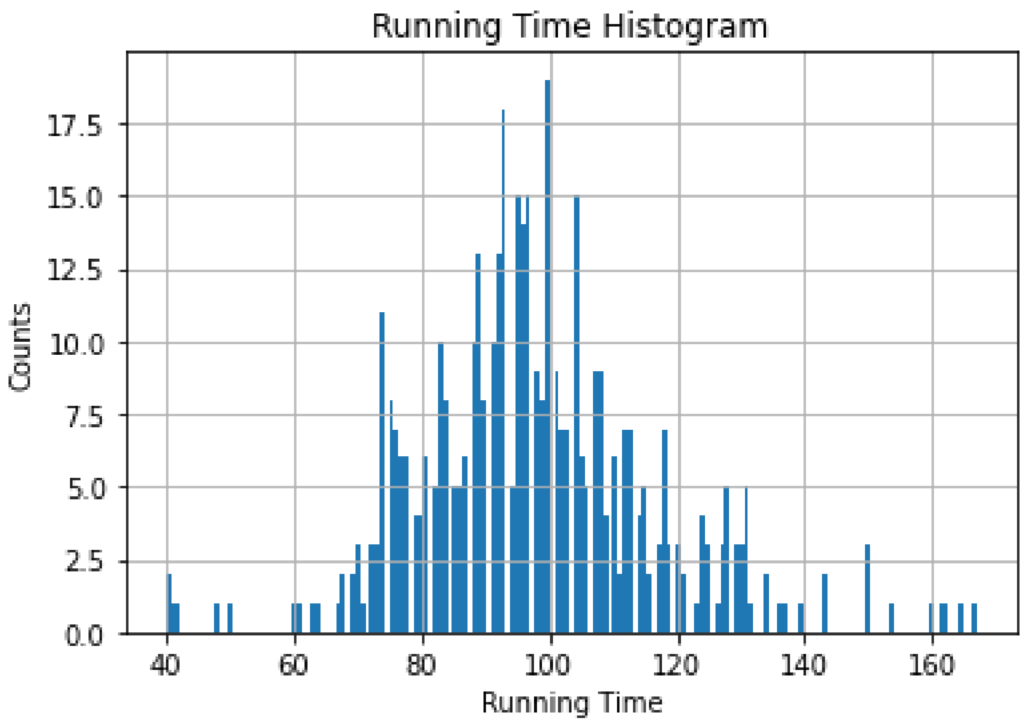
Box Office



Budget



Running Time



Dissimilarity Matrix

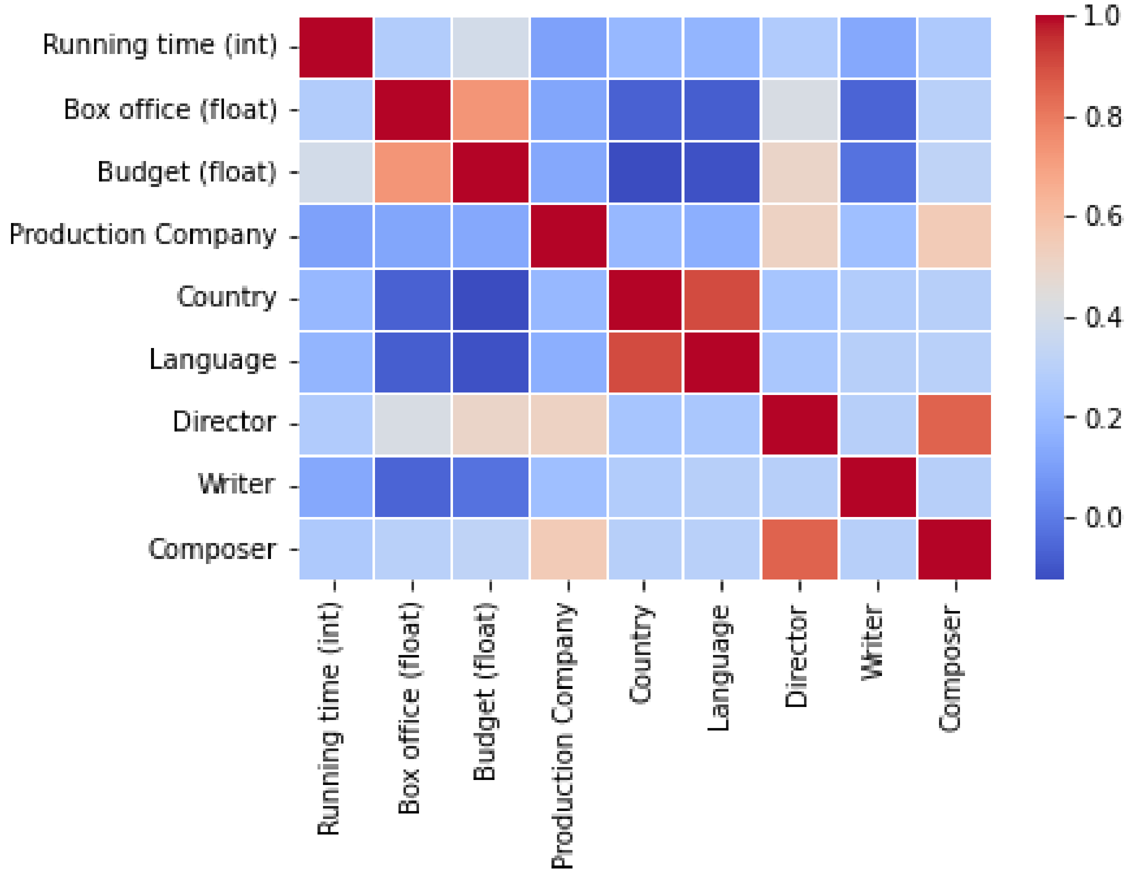
After removing unnecessary attributes, the dissimilarity matrix have been computed. You can see and follow the computation steps here:

Correlation

Correlation between attributes of dataset have been computed after some cleaning and normalization steps. Some unnecessary attributes have been ignored and some nominal attributes have been converted to numerical values.

The procedure and the results can be find here.

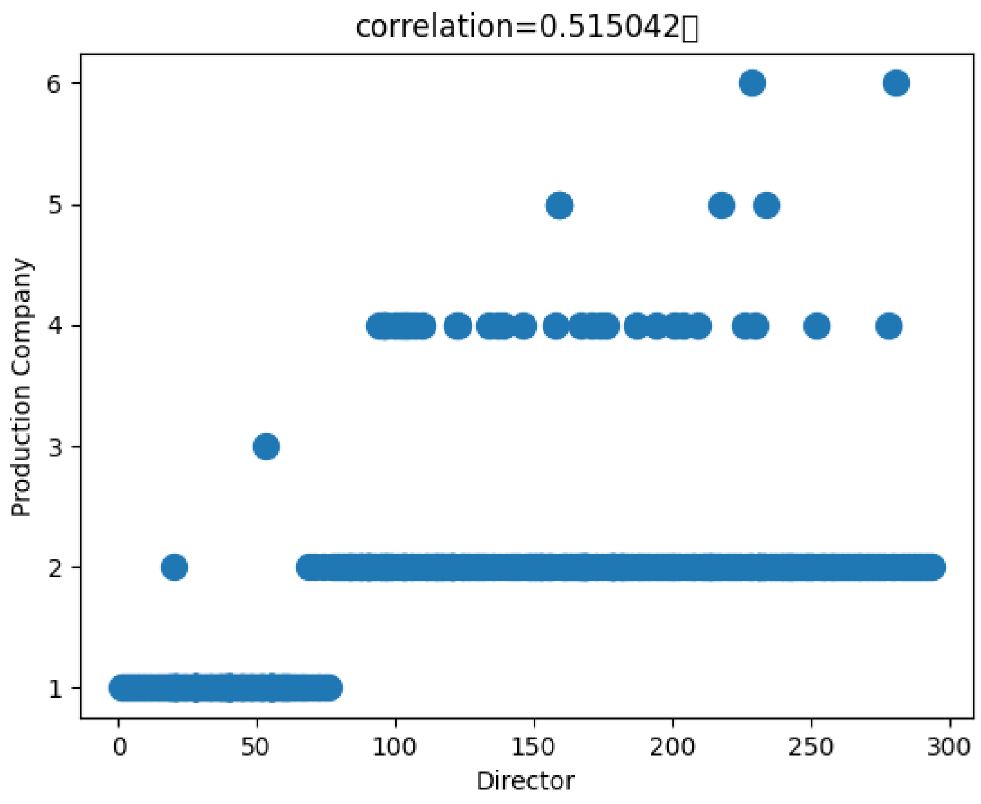
Correlation Heat map



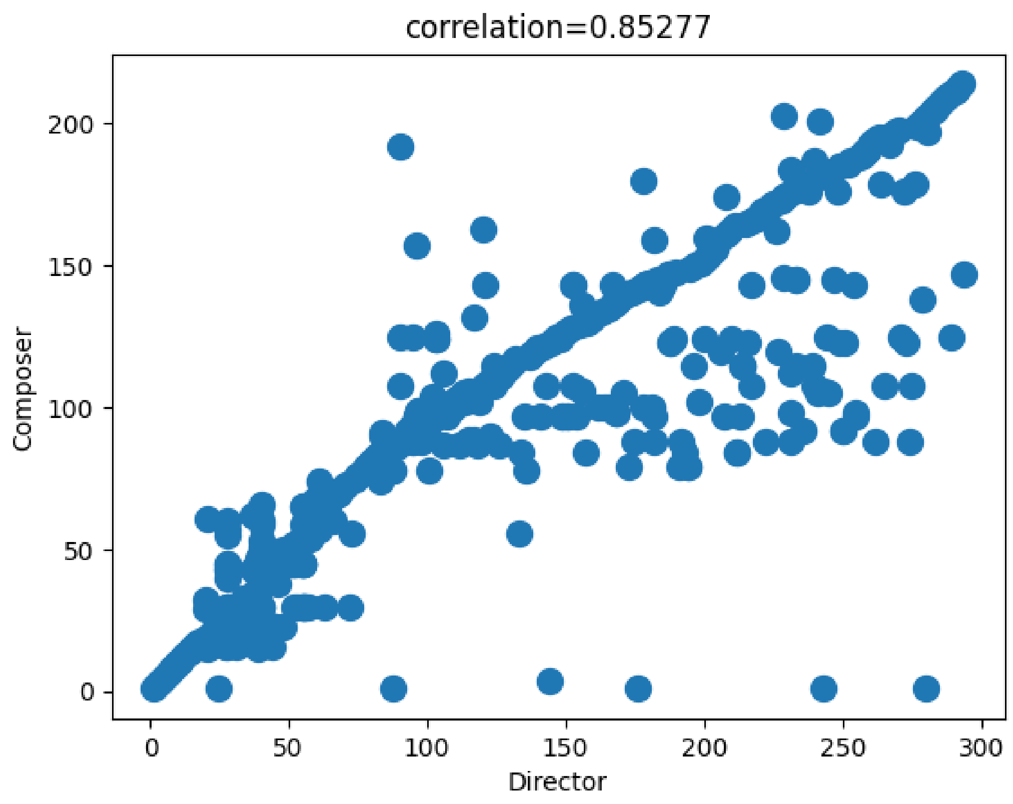
Scatter Plot

Scatter plot for correlated attributes are shown below.

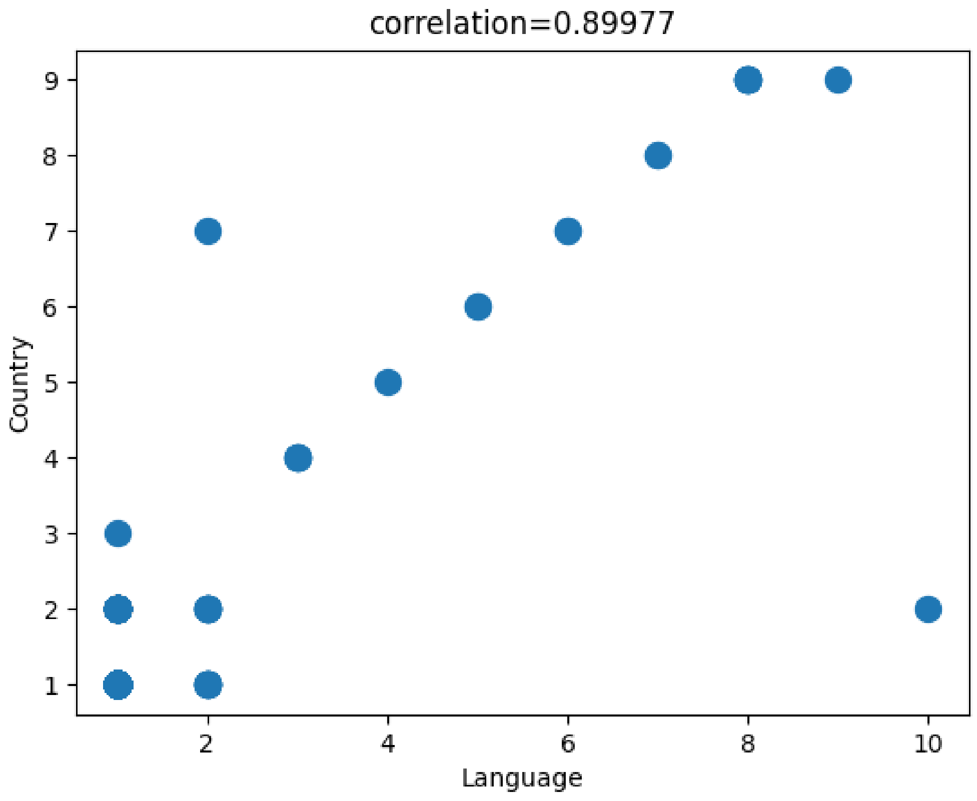
Company-Director Scatter Plot



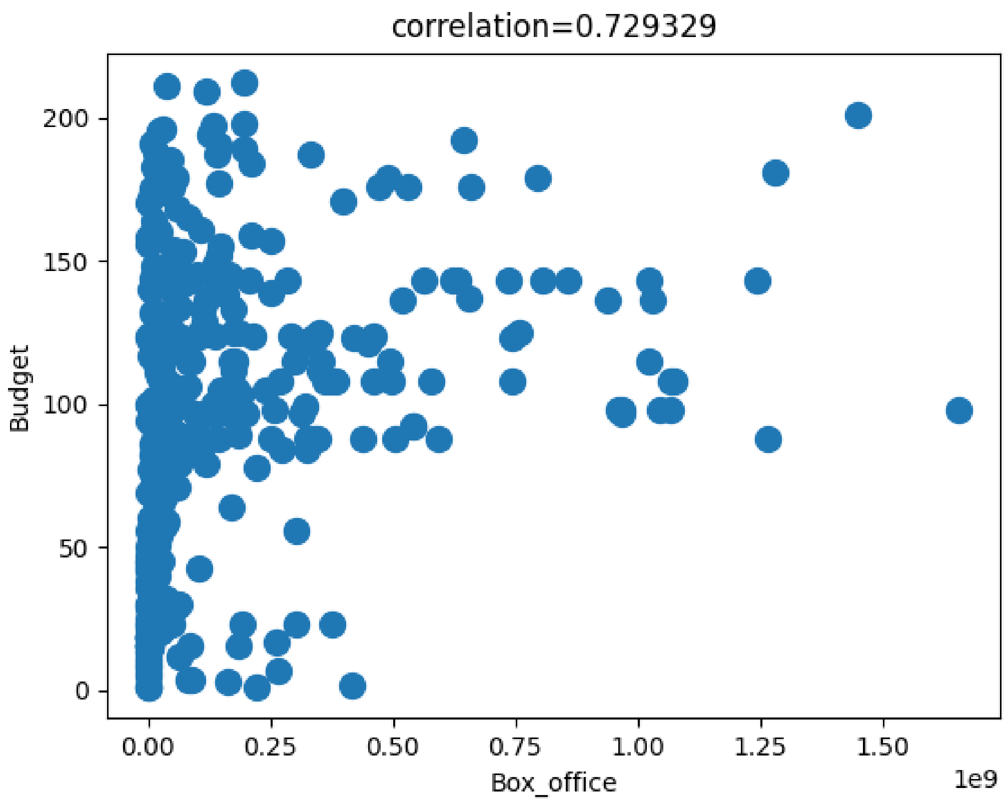
Director-Composer Scatter Plot



Language-Country Scatter Plot



Budget-Box Office Scatter Plot



Data Cleaning

Doing cleaning on the dataset with 35 attributes(columns), we reached a dataset with only 17 attributes(columns).

Columns containing more than 40% missing or invalid values have been dropped.

In addition, the remaining columns' missing values have been imputed automatically using attribute's median.

You can see the procedure and the results by executing cleaningDataset.py file.

Redundant Data

Redundant data records have been handled or removed during data cleaning process.

Dimensional Reduction

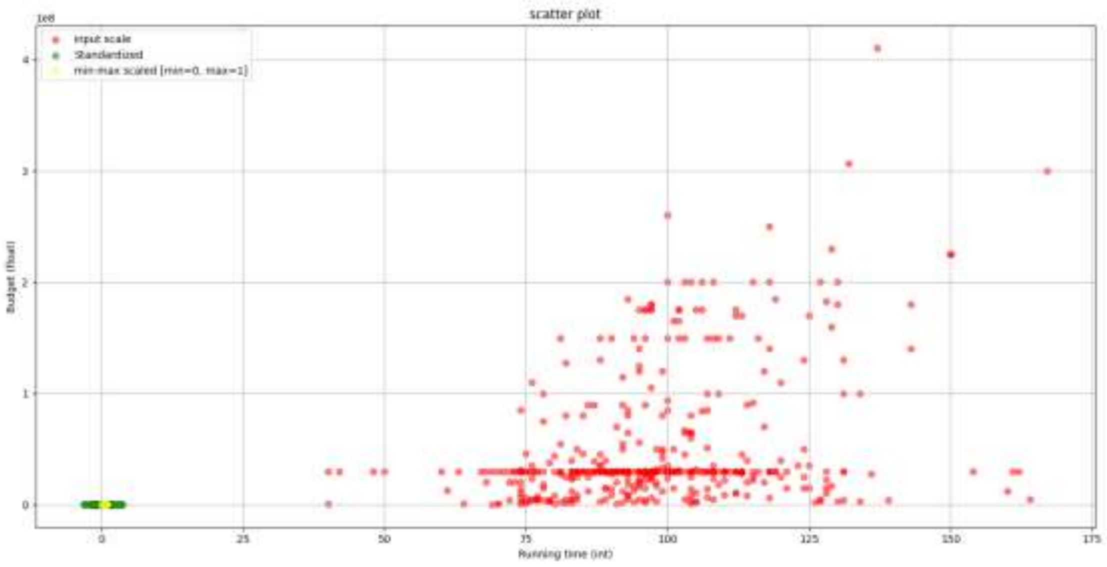
Some useful notes can be taken from regularly observing the dataset. There are some attributes with different labels but containing same data values, outwardly and inwardly. There also are some attributes with more than 40% of missing data or invalid data.

All these cases have been addressed while doing cleaning on the data set.

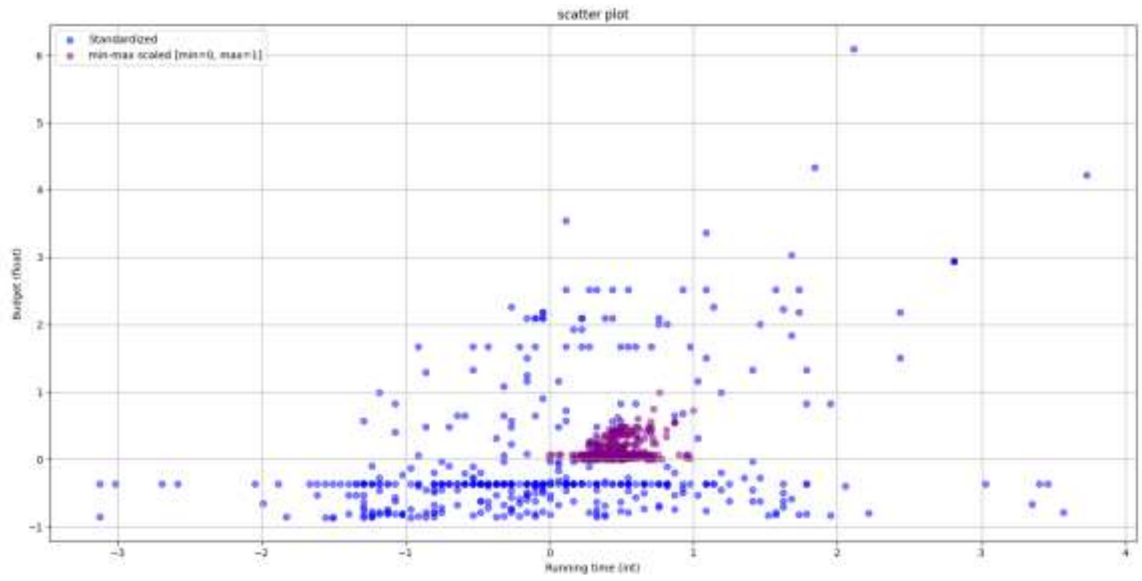
Normalization

Data have been normalized using min-max scaling which scaled data in range [0,1]. Alongside with the normalizing the dataset, z-score standardization have been also done on the dataset. You can see and compare an attribute before and after the procedure below.

Original and Normalized Data Scatter Plot



Normalized Data Scatter Plot



The results are accessible at min-max_normalization.py.

Numerosity Reduction

Knowing the concepts of the records of the datasets, it is simply wrong to perform a numerosity reduction on the data.

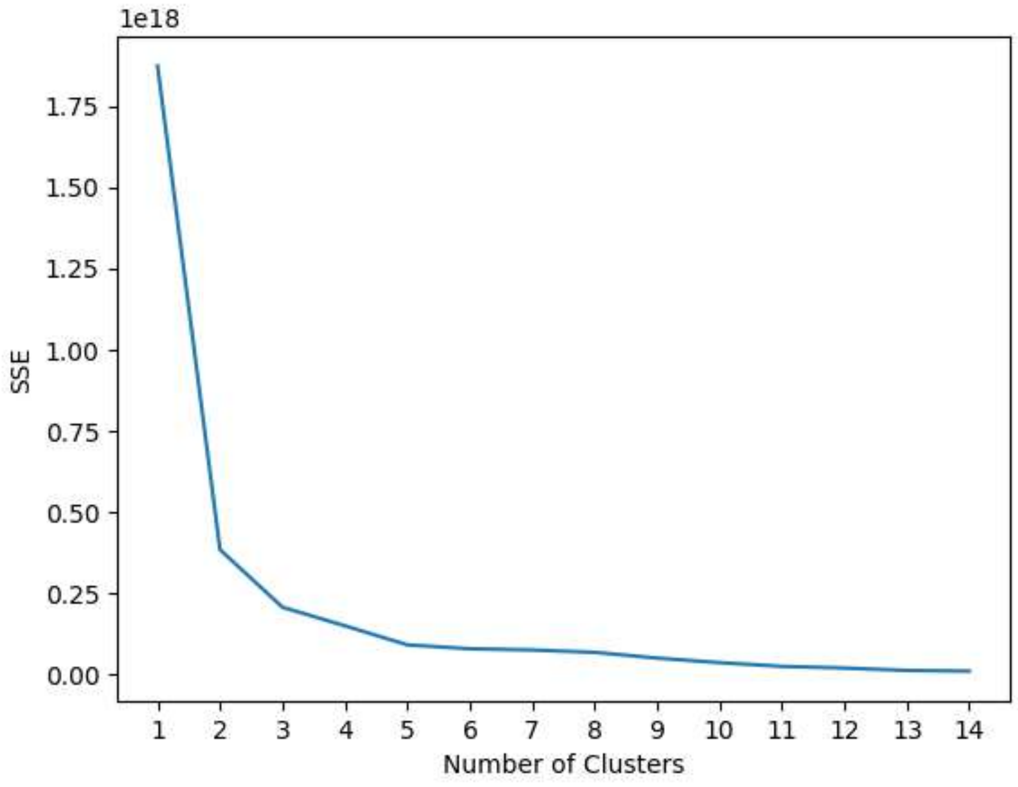
Each record contains data about a single movie produced by Disney Pictures and records are not normally related to on another. **However**, redundancy is not accepted and duplicate data records must be removed from the dataset. **Also** outlier records have been detected and removed. This problems have been solved during data cleaning and outlier detection processes.

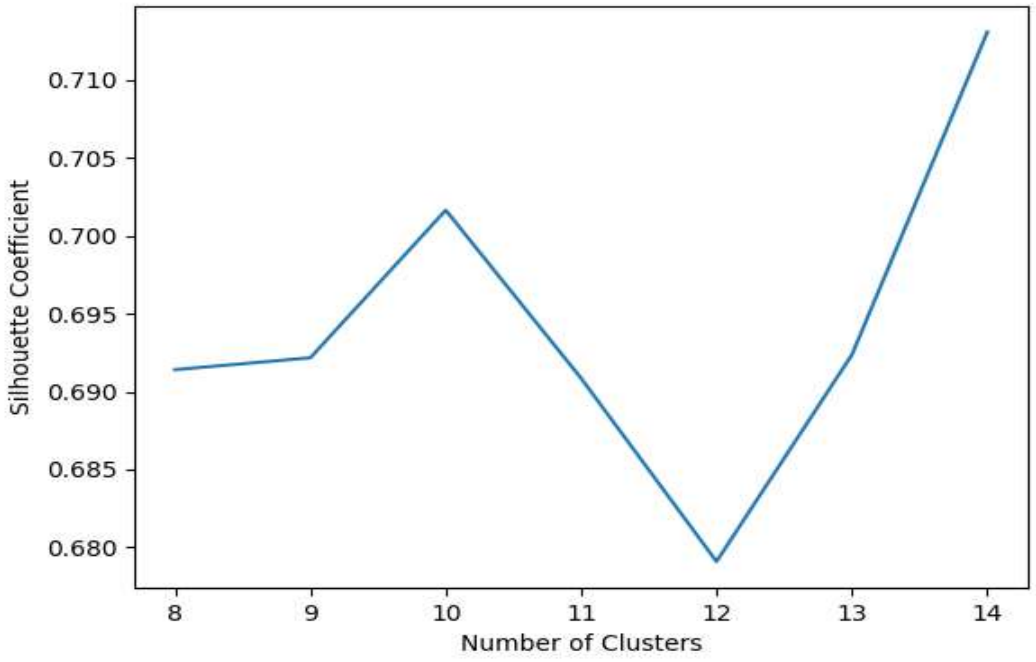
PHASE 2

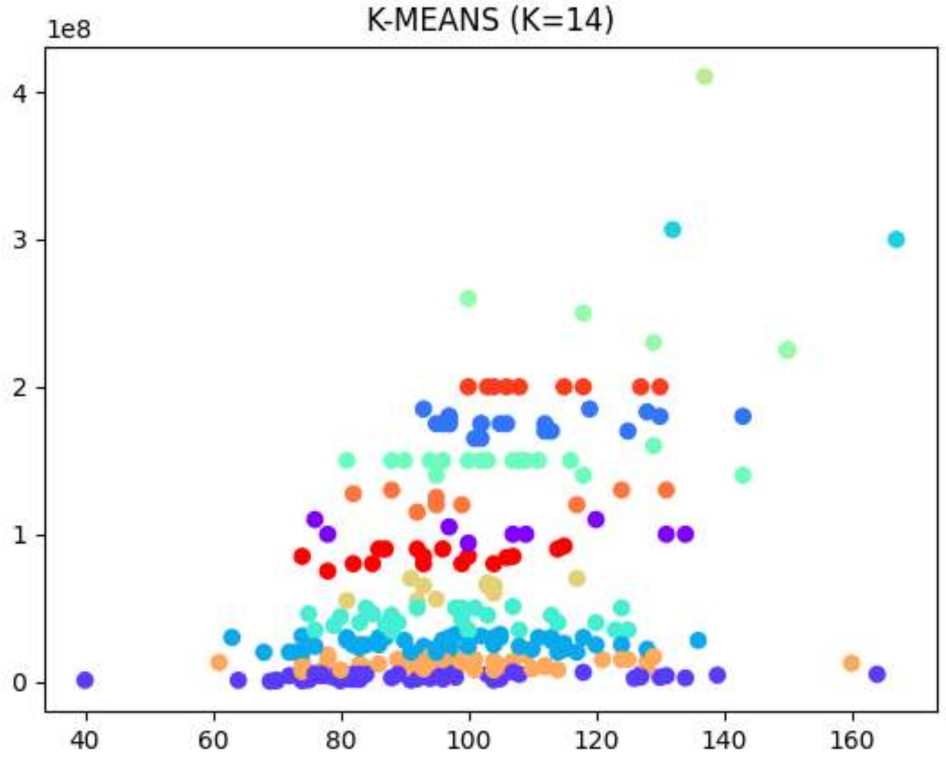
Clustering

K-Means

In K-means, we selected k from 1 to 15 and calculated SSE for all cases. According to the Elbow method and the Silhouette Coefficients method, the highest accuracy and the most suitable clustering for this dataset was 14.

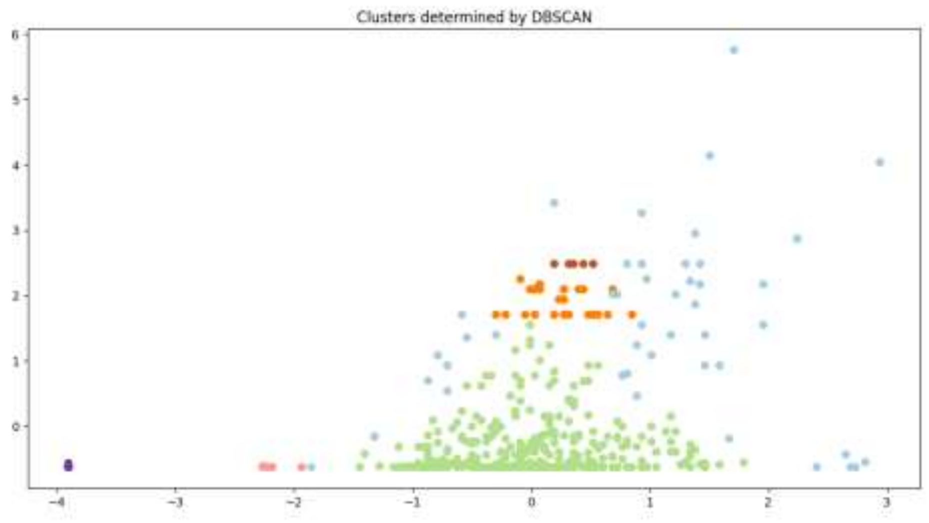



Silhouette Coefficient
0.680
0.685
0.690
0.695
0.700
0.705
0.710
8
9
10
11
12
13
14
Number of Clusters


K-MEANS (K=14)
1e8
0
1
2
3
4
40
60
80
100
120
140
160

DB Scan

As DBSCAN is Density-Based it works better on high density datasets but this dataset with these dimensions with k = 14 was more accurate, but when we got more than 14, it was over fit and an inaccurate and general answers obtained and in this case DB scan showed better ARI .



MLP

In this part, a multi-layer perceptron was trained and deployed for predictions on incoming data. Some of the parameters of this MLP is changed by set-test-set method. Final parameters have had the best results in tests and we decided to stick with those.

The amount of data tuples that we have and can use to train classification models are strictly limited so it's predictable that models tend to either over fit on the data set more often or don't reach an accepted accuracy.

```
MLPClassifier(hidden_layer_sizes=(256,128,64,16),solver='lbfgs',max_iter=3000)
```

Purity of the clf model: 91%

SVM

SVM Classifier had good accuracy and perform faster prediction compared to Naïve Bayes algorithm. It also used less memory .SVM is not suitable for large datasets because of its high training time and it takes more time in training compared to Naïve Bayes. However, as our dataset was not high dimensional, it worked with 96% accuracy.

KNN

KNN could find complex patterns but its output was more challenging to interpret than SVM. SVM was better than KNN to detect outliers. If training data was much larger than this one KNN was better than SVM. At all SVM provided significantly better classification accuracy and classification speed than KNN. SVM had good accuracy and perform faster prediction compared to Naïve Bayes algorithm. It also used less memory

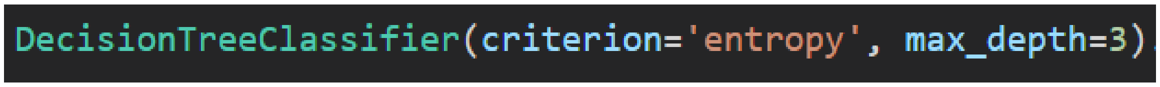
Bayes

Using python sckit-learn library, we applied Gaussian naïve Bayes on the data set. Shortage in data did not have visible effects on Bayes and the results are quite satisfying.

This algorithm succeeded to predict 56 out of 91 test data labels correctly. Its purity is around 61%. It gives us a relatively good accuracy while not over-fitted on the data set.

Decision Tree

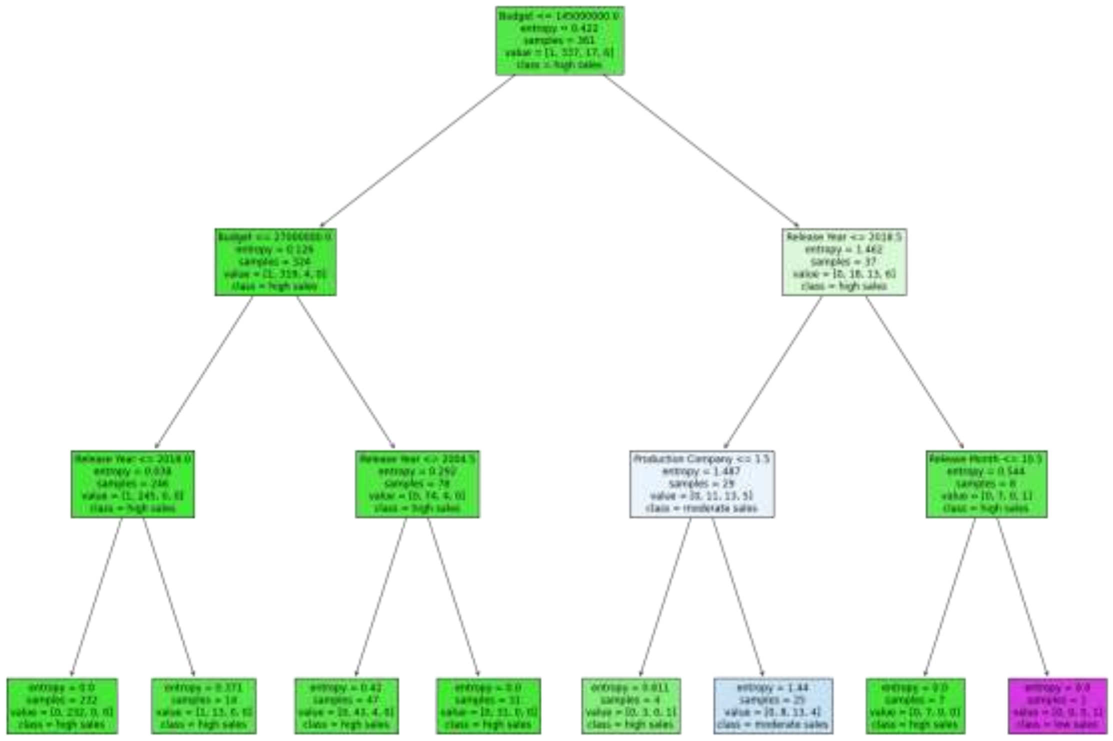
Everyone knows that DT is very strong and efficient classifiers. Here everything is the same. Good accuracy while can be easily prevented from being over fitted on the data by simply setting maximum allowed length of the tree.



```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

This setup predicted test data with 82% accuracy.

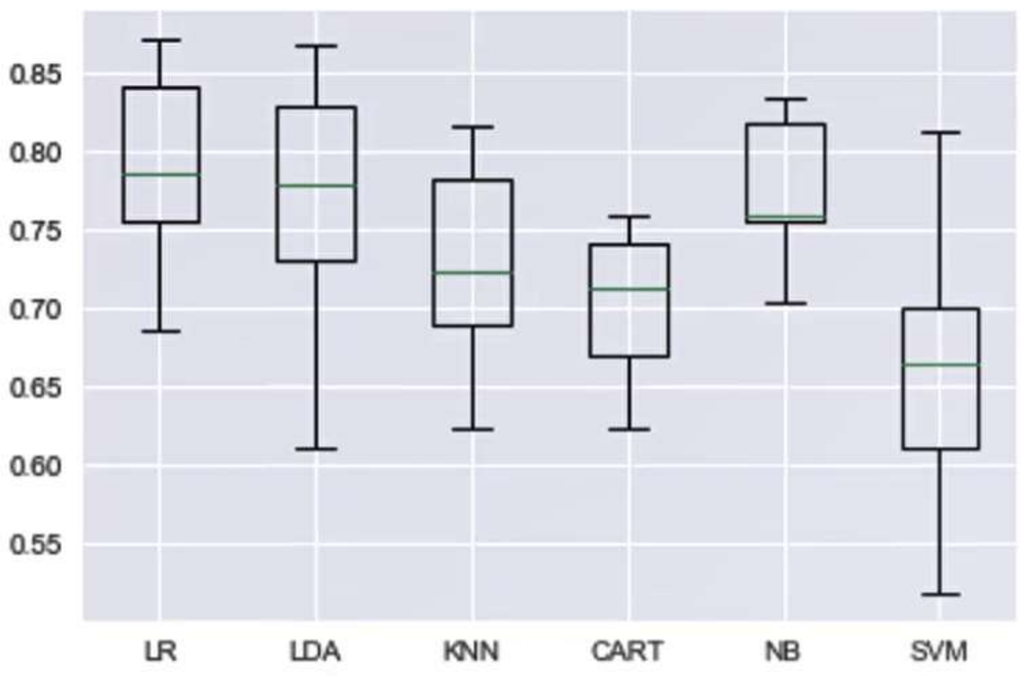
The created tree is also shown below.



Other algorithms

In addition, we compared other popular algorithms with SVM and KNN, which both were great algorithms in this dataset. In the division with a ratio of 30-70, which 30 was for testing, Logistic Regression was the first algorithm to compete in accuracy and speed and efficiency, then Linear Discriminant Analysis and then KNN.

You can see the boxplot of all six algorithms below:


0.85
0.80
0.75
0.70
0.65
0.60
0.55
LR
LDA
KNN
CART
NB
SVM