Comparison of Classification and Clustering Algorithms on Different Missingness, Imputation and Noise Techniques

IS509 – Introduction to Data Science Term Project Report

Mert Mecit
Information Systems Department
Middle East Technical University
Ankara, Turkey
Email: mmecit@metu.edu.tr

Abstract— **In this paper, I tried to compare several classification and clustering algorithms. Applied data consists of one target variable and five explanatory variables and has continuous, ordinal and discrete values. First, information for dataset are introduced, then based on correlation values, different types of missing values are created following different missingness patterns. Those missing values are handled by different methods. Finally, some erroneous values are created. In analysis part, classification algorithms, support vector machines and logistic regression are applied on the databases created above and their performances are compared. Also, PCA applied and without PCA performances of classification algorithms are compared. For clustering algorithms, k-means clustering and agglomerative clustering are compared with the original dataset.**

Keywords-classification; clustering; MCAR; MAR; MNAR; imputation; PCA; missingness; SVM; logistic regression; k-means; agglomerative

I. INTRODUCTION

Classification and clustering algorithms are widely used in machine learning literature and business environments. Classification is generally used in supervised learning environment, where target values are known for the variables in dataset. Classification is done to put an individual to a group based on its characteristics. On the other hand, clustering is generally used in unsupervised learning environments, where target values are not known for the variables in the dataset. Clustering is done to group individuals based on their characteristics.

Both classification and clustering algorithms have different purposes from customer relationship management, financial portfolio management, spam filters, sports and so on. As there are many classification and clustering algorithms, generally which one is the best depends on the situation.

In this paper, I will try to compare different classification algorithms, namely support vector machine and logistic regression and different clustering algorithms, namely k-means clustering and agglomerative clustering algorithms with the given dataset. I will apply different classification algorithms to different types of data, which will be explained in later sections.

First, I will explain my dataset. Then, I will generate different datasets with different kinds of missingness mechanisms, imputation techniques and noise methods. Finally, I will apply classification and clustering methods, I will also look at PCA effect on classification. I will look at the results of those different classification and clustering methods and comment on them.

II. DATASET

Dataset I will be using for this paper and their descriptive statistics are given as follows:

Statistics	**Variable**					
	X1	***X2***	***X3***	***X4***	***X5***	***Y***
Count	210	210	210	210	210	210
Mean	2.02	1.97	3.00	79.91	98.29	1.35
Std	0.79	0.54	0.93	11.01	22.44	1.12
Min	0.00	1.00	1.00	58.40	33.80	0.00
25%	1.25	2.00	2.00	70.82	81.33	0.00
50%	2.00	2.00	3.00	78.28	98.48	1.00
75%	3.00	2.00	4.00	89.27	115.91	2.00
max	4.00	3.00	4.00	113.87	141.82	3.00

Figure 1. Descriptive Statistics

X1 is a discrete variable ranging from 0 to 4. This variable has a normal distribution, Most of the values are 2 (in the middle), considerable amount of 1 and 3 are here and very low amount of observations are on the edge values, 0 and 4.

X2 is ordinal, can take low medium and high. I have converted them into numerical forms, 1 indicating low and 3 indicating high. Most of our data consist of entries with medium x2 value.

X3 is also ordinal, can take A, B, C and D values. I have converted them into numerical forms, 1 indicating A and 4 indicating D. Most common entry in our dataset is D and A is least common one.

X4 is a continuous value ranging from 58 to 114. This variable is relatively right-skewed and have many observations around 90.

X5 is also continuous ranging from 34 to 142. This variable is not as skewed as X4 and there are a lot of observations lower and higher than mean.

Y is our target variable and is nominal. It can take integer values between 0 and 3. It is relatively evenly distributed around those numbers.

III. Correlation Analysis

In this part, I will conduct a correlation analysis to determine which of the attributes can be informative about our target class. This analysis is done because while inserting missingness or erroneous values into our dataset, I will be using that variable.

Correlation table is given as follows:

Variable	**Variable**					
	X1	***X2***	***X3***	***X4***	***X5***	***Y***
X1	1.00	0.11	0.03	-0.89	-0.35	0.61
X2	0.11	1.00	0.02	0.05	-0.61	0.23
X3	0.03	0.02	1.00	-0.04	0.33	0.00
X4	-0.89	0.05	-0.04	1.00	0.24	-0.60
X5	-0.35	-0.61	0.33	0.24	1.00	-0.37
Y	0.61	0.23	0.00	-0.60	-0.37	1.00

Figure 2. Correlation Matrix

As we can see, x1 variable has the highest correlation with our target variable. However, x4 almost have the same correlation with our target variable. They are very close. Also, as correlation is heavily dependent on mean value and for some of our variables mean is not meaningful, I will use adjusted R-squared between each of variables and target value.

Variable	**Variable**
	Y
X1	0.73
X2	0.50
X3	0.54
X4	0.61
X5	0.49

Figure 3. Adj-R^2 Matrix

X1 variable have both highest correlation and adjusted R^2 value with our target variable. So, we can conclude that X1 variable is the most likely to be informative about our target class and this variable will be used to generate missing data and introduce noisy data.

IV. Missing & Noisy Dataset Creation

In this section, I will explain how I created new datasets with missing and noisy values that will be used in classification analysis. All of the missingness and noise will be on X1 variable.

A. *Introducing Missing Values*

Missingness can be 3 type depending on:

a) Whether missingness of the variable is dependent on other variable

b) Whether missingness of the variable is dependent on the value itself.

If both have no reply, remaining data is said to be missing completely at random (MCAR). If (a) holds and (b) does not hold, data is said to be missing at random (MAR) and if both holds, data is said to be missing not at random (MNAR).

In our analysis, I will introduce every type of the missingness.

First, I will introduce MCAR. I have deleted 20% of the X1 values and 40% of the X1 values totally at a random manner. To check whether assumption (a) and (b) holds, I used different methods. To check assumption (a), I have created missingness map, if value is null it takes 1 and if it has a value, it takes 0. Then, I have conducted correlation analysis with other variables in our dataset. As the highest correlation is lower than 0.12 for 20% missing data and 0.06 for 40% missing data, I can conclude that there is no correlation between any other variable and missingness of our 20% and 40% missing datasets. To check assumption (b), I used t-test methodology. I looked for whether there is a difference between means of my original dataset and missing-values dataset. This test has following hypotheses:

$$H_0: \mu_1 = \mu_2$$
$$H_1: \mu_1 \neq \mu_2$$

When I applied t-test to both of them, both 20% missing data and 40% missing data have p values of higher than 0.5. So, we cannot reject null hypothesis of no difference between means and conclude that missingness of x1 value is not dependent on its value.

As both of our assumptions hold, we have created 2 MCAR dataset here, namely mcar20 and mcar40. Now, we can create a MAR dataset by relating missingness of x1 to any other variable but not to itself.

To create a MAR, I looked at correlation matrix in figure 2. As we can see, x1 variable has highest correlation with x4. So, I think creating MAR with a condition on X4 would be easy. I created random indexes from X4 variable where X4 is lower than 78 (close to its mean). Then, I selected 42 of them (as I want to create 20% missing data). I deleted X1 values in those indexes. This should be the MAR but we need to conduct proper tests.

To test assumption (a), I created a missingness map and looked at this variable's correlation with other variables. This time, x1 has more than 0.44 correlation with x4. This indicates a moderate to high correlation between missingness of x1 and value of x4. This assumption holds for MAR. To test assumption (b), I did a similar analysis, conducted a t-test. This time, t-test have a value over 0.05, so we still fail to reject null hypothesis.

To be MAR, assumption (a) should hold but assumption (b) should not hold. This is our case, so we can conclude this dataset is in MAR with 20% missingness and named it mar20. Finally, I will create MNAR dataset with 20% missingness of X1. Here, missingness of X1 should be related to its value. I created random indexes from X1 variable where X1 is either 0 or 1. Then, I selected 42 of them (as I want to create 20% missing data). I deleted X1 values in those indexes. This should be the MNAR but we need to conduct proper tests.

Here, testing assumption (b) is enough as if it holds, data is guaranteed to be in MNAR. To test assumption (b), I did a similar analysis above, conducted a t-test. This time, t-test have a value lower than 0.0002, so we can reject null hypothesis and find a difference between means.

We can conclude this new dataset is in MNAR with 20% missingness and named it mnar20.

To sum up, we have created mcar20, mcar40, mar20 and mnar20 datasets with different missingness methods and tested underlying assumptions.

B. *Introducing Error*

In this part, I will create two new datasets by introducing noisy data in X1.

As noise, I added '5' values into X1 by randomly selecting 42 (for 20% error) and 84 (for 40% error). So, I created two new datasets and named them as err20 and err40 to use them later in classification analysis.

Now, as we have created different missingness and erroneous datasets, we can conduct different imputation mechanisms on them.

V. Imputation

In this section, I will conduct different imputation methods on different missing data introduced in part IV. I will use feature deleting, row deleting, imputing a constant value, and imputing by regression methods on 4 datasets that I have created with missing data (mcar20, mcar40, mar20, mnar20).

A. *Feature Deleting*

One of the simplest method to deal with missing values is to delete the column entirely. This method has advantage of being easy to implement and if the missing values are on the same column (this is our case), we may not lose many information. However, one significant disadvantage is we can loss huge amounts of information.

In our datasets, when we delete x1 columns, we will have same amount of rows (210) with one less columns (5). All of datasets will have (210,5) shape. New created datasets are named as mcar20df, mcar40df, mar20df and mnar20df.

B. *Row Deleting*

Row deleting is another simple method to deal with missing values. Whenever there is a missing value in a row, we delete the row entirely. Advantage of this method is its easy to use but it has a disadvantage of losing information because by deleting row, we also delete information on hand.

In our datasets, when we delete rows that have a missing x1 value, all of the datasets have same amount of columns. However, this time mcar20, mar20 and mnar20 have 168 rows and mcar40 have 126 rows. New created datasets are named as mcar20rl, mcar40rl, mar40rl and mnar40rl.

C. *Imputing a Constant Value*

Imputing a constant value to missing cells is another simple method to dealing with missing values. Generally, this constant is chosen as mean, median or mode. Advantage of this method is similar to the previous two ones, its simple. Also, with this approach, we do not lose much information. Disadvantage of this method is it reduces variability and it weakens the correlation.

In our dataset, as x1 is the missing variable we have and it's a discrete variable, I prefer to use median as our substitution constant. For each of the dataset, I have calculated median of non-missing x1 values and used that value to impute every missing x1 value. I have created four new datasets, namely mcar20m, mcar40m, mar20m, mnar20m and all of those newly created datasets have shape of (210, 6), they do not lose information.

D. *Imputing by Regression*

Regression based imputation is one of the non-simple imputation models. Here, we create a regression equation from non-missing values to predict missing values. Advantage of this method is it uses available information. Disadvantages are it is not very easy to use this method (although programming environment can do it easily), like imputing mean or median, it reduces variability.

In our dataset, the methodology I chose is MICE imputation. It is a multiple iterative imputation method. In my model, I have used 10 different models to predict the missing data. Then, those 10 predictions are combined to come up with a single data for missing value.

When I applied MICE to each of the dataset (mcar20, mcar40, mar20 and mnar20) I came up with new non-missing datasets and named them as mcar20reg, mcar40reg, mar20reg and mnar20reg. All of those new datasets have shape of (210, 6).

Now, we have all the datasets that we created. Figure 4 briefly summarizes all the datasets that we have.

Dataset	**Characteristics**	
	Description	***Shape***
data	Original dataset	(210,6)
mcar20	%20 of X1 values deleted by MCAR	(210,6)
mcar40	%40 of X1 values deleted by MCAR	(210,6)
mar20	%20 of X1 values deleted by MAR	(210,6)
mnar20	%20 of X1 values deleted by MNAR	(210,6)
mcar20df	Feature deletion on mcar20	(210,5)
mcar40df	Feature deletion on mcar40	(210,5)
mar20df	Feature deletion on mar20	(210,5)
mnar20df	Feature deletion on mnar20	(210,5)
mcar20rl	Row deletion on mcar20	(168,6)
mcar40rl	Row deletion on mcar40	(168,6)
mar20rl	Row deletion on mar20	(168,6)
mnar20rl	Row deletion on mnar20	(126,6)
mcar20m	Median imputation on mcar20	(210,6)
mcar40m	Median imputation on mcar40	(210,6)
mar20m	Median imputation on mar20	(210,6)
mnar20m	Median imputation on mnar20	(210,6)
mcar20reg	Regression imputation on mcar20	(210,6)
mcar40reg	Regression imputation on mcar40	(210,6)
mar20reg	Regression imputation on mar20	(210,6)
mnar20reg	Regression imputation on mnar20	(210,6)
err20	%20 of X1 values are noisy	(210,6)
err40	%40 of X1 values are noisy	(210,6)

Figure 4. Dataset Summary

VI. Classification Analysis

Our dataset are prepared, so we can continue with our analysis now. First analysis that we will be conducting is classification analysis. Classification is simply classifying an individual based on its characteristics. Detecting fraudulent activities in banking, e-mail spam detection, churn rate prediction in marketing are among areas that utilizes classification analysis methods a lot.

As it is widely used, many kinds of classification algorithms have emerged. Logistic regression, decision trees, naïve bayes, support vector machine, stochastic gradient descent and random forest are some of the most popular ones. In this paper, I will use logistic regression and support vector machine.

I will apply support vector machine and logistic regression algorithms to 19 datasets that are listed in the figure 4 above (no missing data ones). First, I will start with support vector machine, then I will look at logistic regression and finally I will compare those two classification algorithms. As there are 5 dimensions, unfortunately visualization techniques will not be applied.

A. *Support Vector Machine*

In this algorithm, main purpose is to find decision boundary between two classes that are furthest from any point in the training data. It is one of the simplest classification methods.

To conduct support vector machine analysis (or classification analysis in general), I need to divide my data into training an test data. I have used 20% data for testing and 80% for training purposes. I wanted to use high amounts of training data to improve the fit and I did not want to use more than %90+ to overcome any kind of overfitting in my datasets.

After dividing 19 datasets into 4 sets (x_train, x_test, y_train and y_test), I can apply support vector machine now. I used Python’s sklearn library to conduct SVM analysis. For all my 23 datasets, I have used linear kernel. I used it because I have many number of features in my dataset and also it is simply the most used one. I chose C as 2 and I wanted a low C to make decision surface smooth. I chose gamma as auto.

I have prepared my model and trained it. Then, I used my model on test data and came up with precision and recall values for each of the dataset. I will be showing weighted average of them as it is based on the frequencies of the target variable.

Dataset	**SVM**		**Dataset**	**SVM**	
	Precision	**Recall**		**Precision**	**Recall**
data	**0.75**	**0.74**	mcar40m	**0.74**	**0.71**
mcar20df	**0.83**	**0.79**	mar20m	**0.77**	**0.76**
mcar40df	**0.83**	**0.79**	mnar20m	**0.78**	**0.76**
mar20df	**0.83**	**0.79**	mcar20reg	**0.72**	**0.71**
mnar20df	**0.83**	**0.79**	mcar40reg	**0.72**	**0.71**
mcar20rl	**0.63**	**0.62**	mar20reg	**0.80**	**0.79**
mcar40rl	**0.76**	**0.65**	mnar20reg	**0.76**	**0.74**
mar20rl	**0.71**	**0.71**	err20	**0.84**	**0.81**
mnar20rl	**0.71**	**0.71**	err40	**0.81**	**0.79**
mcar20m	**0.75**	**0.74**			

Figure 5. SVM Summary

SVM precision and recall values are given above. Although they are informative, I believe they are mostly beneficial for comparison purposes. So, a detailed comment will be made on comments part.

B. *Logistic Regression*

Logistic regression is another classification method that is widely utilized. Contrary to SVM which utilizes geometry, logistic regression is more based on statistical approaches. However, just like SVM, logistic regression have a wide area of use. Detecting spam emails, detecting fraudulent transactions and tumors as malignant or benign are among some uses of logistic regression.

I will be using my training and test dataset that have 19 different sets. For my logistic regression analysis, I used liblinear solver as it supports both L1 and L2 regularization. In Python, regularization is applied by default. Random state is used for shuffling purposes. Also, C is chosen to be 2 to be consistent with SVM.

I have prepared my model and trained it. Then, I used my model on test data and came up with precision and recall values for each of the dataset. I will be showing weighted average of them as it is based on the frequencies of the target variable.

Dataset	SVM		Dataset	SVM	
	Precision	**Recall**		**Precision**	**Recall**
data	**0.83**	**0.79**	mcar40m	**0.77**	**0.74**
mcar20df	**0.73**	**0.71**	mar20m	**0.78**	**0.74**
mcar40df	**0.73**	**0.71**	mnar20m	**0.83**	**0.81**
mar20df	**0.73**	**0.71**	mcar20reg	**0.83**	**0.79**
mnar20df	**0.73**	**0.71**	mcar40reg	**0.83**	**0.79**
mcar20rl	**0.64**	**0.62**	mar20reg	**0.83**	**0.79**
mcar40rl	**0.66**	**0.46**	mnar20reg	**0.81**	**0.79**
mar20rl	**0.64**	**0.56**	err20	**0.76**	**0.71**
mnar20rl	**0.78**	**0.76**	err40	**0.75**	**0.69**
mcar20m	**0.81**	**0.76**			

Figure 6. Logistic Regression Summary

Logistic regression precision and recall values are given above. Although they are informative, I believe they are mostly beneficial for comparison purposes. So, a detailed comment will be made on comments part.

C. Comparison

How well SVM and logistic regression algorithms are doing are given in precision and recall values in their corresponding parts above. Now, I will briefly compare them both in terms of precision and recall values. First, I will present overall averages in the table below, then I will compare each dataset individually.

Variable	Average
SVM_precision	0.77
Log_precision	0.76
SVM_recall	0.74
Log_recall	0.72

Figure 7. SVM vs Logistic Regression

Table above shows the averages. By precision, we mean percent of instances that is labeled are actually positive. By recall, we mean % of positive instances labeled as positive. So, in this sense, SVM outperforms on average logistic regression both in precision and recall.

However, when we look at each dataset individually (or at least by imputation method), we see that logistic regression does better job only in original dataset and regression based imputation ones. Other than those, SVM outperforms the logistic regression.

D. Introducing PCA

Principal component analysis (PCA) is a method of dimensionality reduction. It may help us to reduce noise but it generally is very beneficial for visualization purposes. It basically transform our original complex data to a smaller space.

In my analysis, I want to look at whether PCA transformation will increase my classification accuracy or not. I will apply PCA transformation only to my original dataset.

To conduct PCA, first I need to determine how much component I need. Methodology I used is to select ones with high explained variance ratios. Explained variance ratios are given in the table below.

Component	Explained Variance
1	0.81
2	0.18
3	0.001

Figure 8. PCA Explained Variance, Classification

As we can see 2 principal components are very good for explaining variance, they add up to over 99% of the variance, so I will choose 2 principal components.

I will apply SVM and logistic regression on my newly created dataset. The way I apply those 2 methodologies are the same with the ones I applied before, I used same parameters to make them comparable.

Results are as follows:

Variable	Precision	Recall
SVM_with_PCA	0.77	0.74
SVM_without_PCA	0.75	0.74
Log_with_PCA	0.76	0.74
Log_without_PCA	0.83	0.79

Figure 9. PCA vs Without PCA

As we can see from precision and recall values, our performance on SVM slightly increased but our performance on logistic regression is decreased. However, as this is only to one of the datasets, we cannot generalize and claim that PCA is not useful in classification purposes. This original dataset was one of the datasets that our both models are performing relatively good.

This concludes my analysis on classification part. In this part, I have compared SVM and logistic regression and for my dataset and different imputation methods, SVM did slightly better job than logistic regression. Then, I introduced PCA methodology and tested whether my performance on both algorithms increased on first dataset.

Although performance of SVM increased slightly, logistic regression performance decreased.

VII. Cluster Analysis

Second analysis that we will be conducting is cluster analysis. Clustering is simply grouping individuals based on their characteristics. Discovering distinct groups of customers, recognizing data pattern and spam filtering are among areas that utilizes classification analysis methods a lot.

As it is widely used, many kinds of clustering algorithms have emerged. Agglomerative clustering, BIRCH, DBSCAN, K-means, Mini batch K-means, OPTICS, K-medoids are among most popular clustering algorithms. In this paper, I will be using agglomerative clustering and K-means clustering and compare their performances on my original dataset.

I will apply k-means clustering and agglomerative clustering algorithms to my original dataset. First, I will start with k-means clustering, then I will look at agglomerative clustering and finally I will compare those two clustering algorithms. To show them visually, I will apply PCA to them as well.

A. K-Means Clustering

In this algorithm, main purpose is to find K clusters in database by an iterative approach. Algorithm continuously iterate itself until convergence criterion is satisfied. It is one of the simplest clustering methods.

To conduct k-means clustering analysis (or classification analysis in general), I need to standardize my data. I used StandardScaler and fit_transform functions of Python to standardize my data. Then, I also applied PCA transformation to be able to visualize my clustering. When I apply PCA to my scaled data, explained variances are as follows:

Component	**Explained Variance**
1	0.45
2	0.25
3	0.17

Figure 10. PCA Explained Variance, Clustering

Again, I can use 2 because it explains high amount of variance and also is good for visualization purposes.

Now, I have standardized data. So, I can proceed to find the optimal cluster number for my k-means clustering. From k=2 to k=10, clusters are formed as follows:

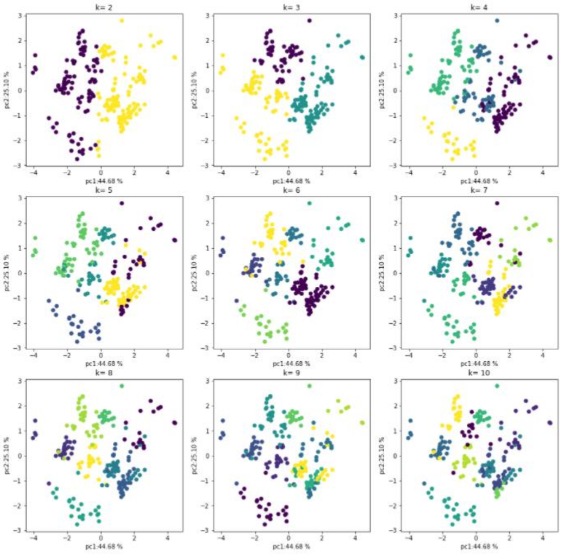

Figure 11. K-means Clustering Clusters

There is a grouped part below. I believe the good clustering must catch that part. From my point of view, k=4 results in best clustering and it prevents overfitting. Now, we can have a look at average silhouette approach to determine best k value:

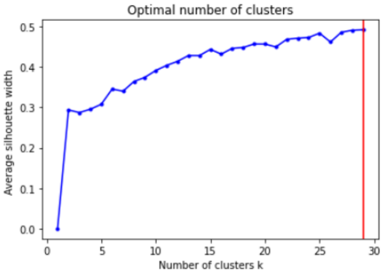

Figure 12. K-means, Average Silhouette

Even with max k=30, this algorithm suggests us to use 30 clusters. This causes a huge overfitting problem in our data, so I would rather using 4 clusters for this purpose.

B. Agglomerative Clustering

This is a type of hierarchical clustering algorithm. Main purpose is to find clusters in database by continuously merging clusters that were individual in the beginning.

As I have standardized my data, I can proceed to find the optimal cluster number for my agglomerative clustering. From k=2 to k=10, clusters are formed as follows:

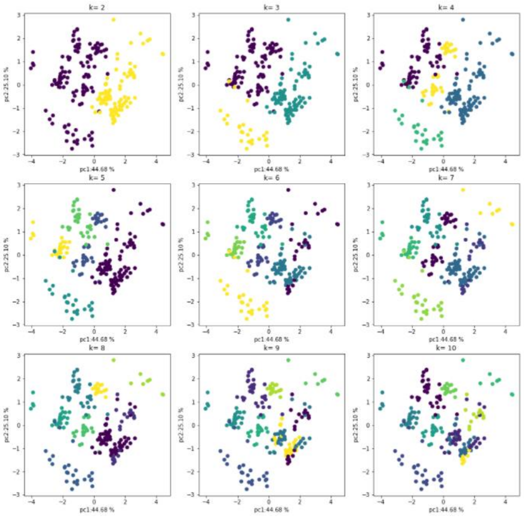

Figure 13. Agglomerative Clustering Clusters

There is a grouped part below. I believe the good clustering must catch that part. Also, here when k=7, we can capture the little cluster above (yellow dots). So, I believe although k=3 and k=4 are good, 7 looks the best clustering algorithm here. Now, we can have a look at average silhouette approach to determine best k value:

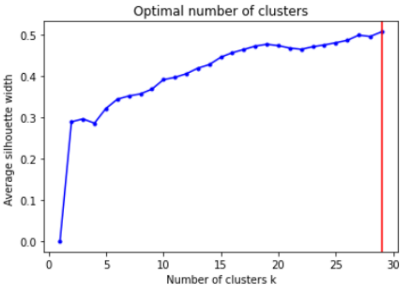

Figure 14. Agglomerative Clustering, Average Silhouette

Even with max k=30, this algorithm suggests us to use 30 clusters. This causes a huge overfitting problem in our data, so I would rather using 7 clusters for this purpose.

C. *Comparison*

When we compare those two clustering algorithms, we can see results are somewhat similar but there are some differences. For example, middle part in k=3 differs a lot from k-means to agglomerative clustering. However, although this data is very dense, cluster algorithms are not doing very good here. Maybe a more advanced clustering algorithm such as DBSCAN may work better here.

VIII. Conclusion

To conclude, in this paper I have tried to compare same classification and clustering algorithms on a data that has different missingness and error types and missing values are filled or removed with different methods.

Dataset consist of continuous, discrete and ordinal values. To create missing and erroneous data, x1 variable, which has the highest correlation with target valuable is chosen.

Then, 4 new datasets are created by transforming our dataset into MCAR (x2), MAR and MNAR forms. Also, 2 new datasets are created to introduce errors on variable x1. Then, 4 different types of missingness handling methods are used on 4 missing data dataset, column deletion, row deletion, median imputation, and regression imputation.

Then, we proceed to our analysis. First, classification analysis is conducted by comparing 2 algorithms, support vector machine and logistic regression. In terms of precision and recall, on average, support vector machine was performing better classification than logistic regression. However, for our original dataset and regression imputed datasets, logistic regression was performing better. Then, I introduced PCA into our original dataset and compared PCA applied classification and our normal classification. For SVM, performance has improved but for logistic regression, performance has worsened.

Finally, we looked at different clustering algorithm performance on my original dataset. After standardizing my data, I looked at k-means clustering and agglomerative clustering. When I did average silhouette analysis, both was suggesting a k which is very high (>30). So, I visualized them and tried to find a suitable k manually. Heuristically, a k=4 for k-means and a k=7 for agglomerative clustering seemed to work well.