

CKD Classification harnessing ML algorithms SMO-SVM : A WEKA Approach**Dr. Rajesh S. Walse¹**

¹ Assistant Professor (Computer Science),
Department of Dairy Business Management,
College of Dairy Technology, Warud (Pusad),
M.A.F.S.U, Nagpur, Maharashtra State, India
rajeshwalse@gmail.com

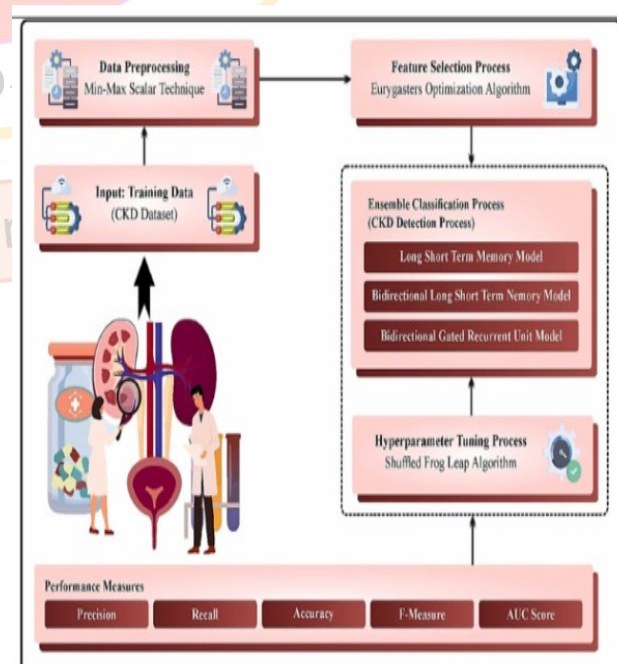
Abstract

In this chapter, we're using a way to classify and group data to compare information about people with chronic kidney disease. It's a big problem in India because more and more people are getting CKD, often because of what they eat and other health issues. The Indian Journal of Nephrology and others [12] have noted that CKD cases have jumped up a lot in the last decade. So, with the current COVID-19 pandemic and looking ahead, research that can predict CKD might really help doctors and the healthcare industry, rather than focusing on CKD patients based on other health factors. This could help slow down the rise of CKD, prevent more kidney damage, and allow for better treatments using these advanced machine learning ideas, potentially boosting survival rates. The sad reality is that many CKD patients are only found when their symptoms are quite advanced, meaning their serum creatinine levels are already very high. We're using SMO-SVM, a supervised method, along with unsupervised Hierarchical Clustering here. Our proposed model is tested on data about chronic kidney disease that includes 25 features and 400 patient records. This often ties into unsupervised, hierarchical clustering that uses Euclidean and Manhattan calculations. For the supervised SMO-SVM part in WEKA, we found the best predictions after three tests. We used a logistic and kernel calibrator with a polynomial function, and employed cross-validation with folds ranging from 4 to 15.

Keywords: Data Mining, Classification, Clustering, SMO, SVM, Calibrator, Logistic, Kernel, Polynomial, Euclidean, Manhattan, WEKA, CKD.

1.Introduction

It seems like more and more people in India are developing chronic kidney disease lately, often due to their diet and other health problems. In fact, the number of CKD patients has shot up dramatically over the past decade, as noted in the Indian Journal of Nephrology and other sources. Because of this, research that helps doctors and the medical field predict who might develop CKD or not, based on their existing health conditions, will be really important down the line. This kind of work could help slow the rise of CKD and prevent further kidney damage.



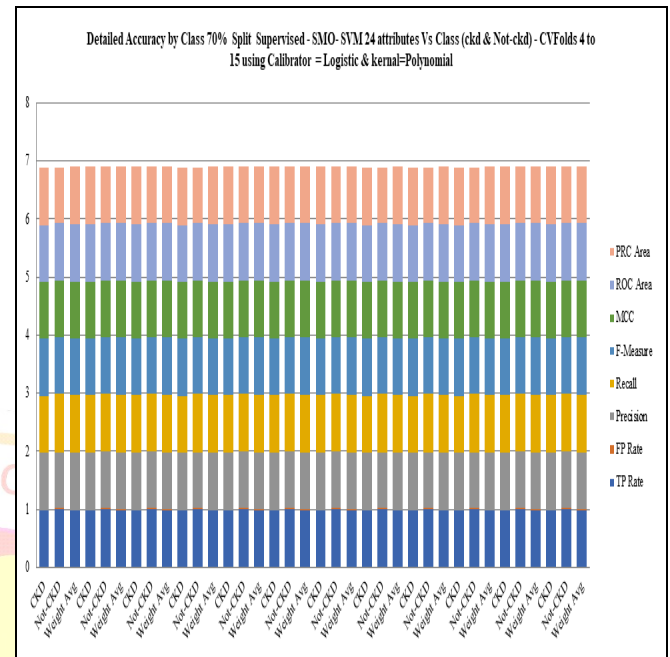


Figure: 25 Detailed accuracy by class Full attribute Vs Class



1. Test Mode: split 70% , 2. Total Number of Instances=400

Sr. No.	Particulars	4	5	6	7	8	9	10	11	12	13	14	15
1	Time taken to build model	0.03	0.03	0.03	0.03	0.02	0.02	0.03	0.03	0.03	0.02	0.03	0.03
2	Correctly Classified Instances	98.50	98.75	98.75	98.50	98.75	98.75	98.75	98.50	98.50	98.50	98.75	98.75
3	Incorrectly Classified Instances	1.50	1.25	1.25	1.50	1.25	1.25	1.25	1.50	1.50	1.50	1.25	1.25
4	Kappa statistic	0.9683	0.9735	0.9735	0.9683	0.9735	0.9735	0.9735	0.9683	0.9683	0.9683	0.9735	0.9735
5	Mean absolute error	0.015	0.0125	0.0125	0.015	0.0125	0.0125	0.0125	0.015	0.015	0.015	0.0125	0.0125
6	Root mean squared error	0.1225	0.1118	0.1118	0.1225	0.1118	0.1118	0.1118	0.1225	0.1225	0.1225	0.1118	0.1118
7	Relative absolute error	3.1985	2.6656	2.6656	3.1986	2.6655	2.6656	2.6657	3.1987	3.1987	3.1986	2.6656	2.6657
8	Root relative squared error	25.2971	23.094	23.0938	25.297	23.0929	23.093	23.094	25.2968	25.2968	25.2964	23.0929	23.0937

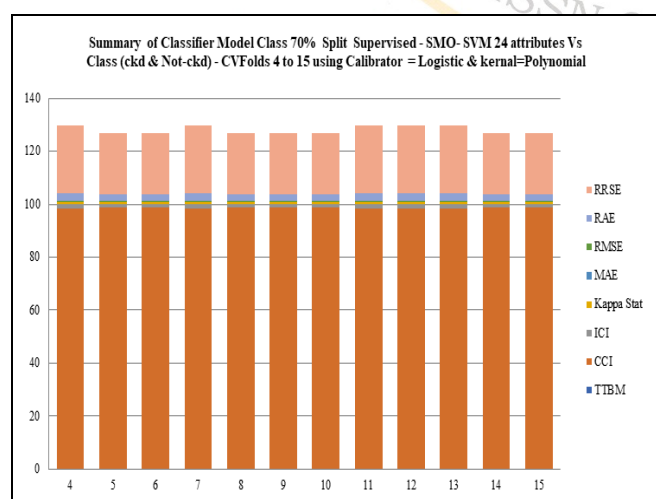
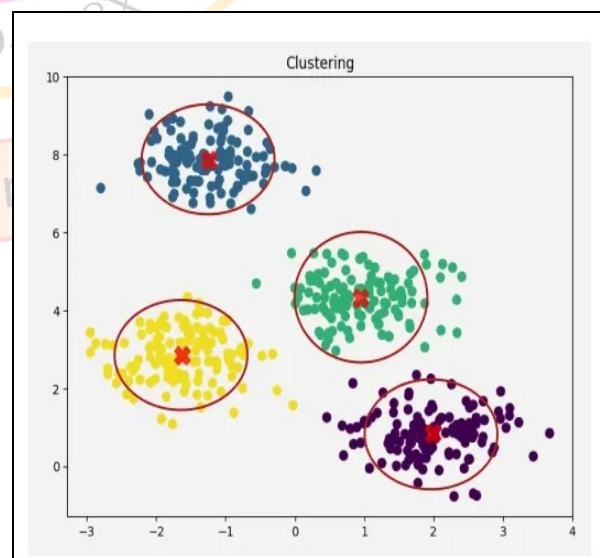
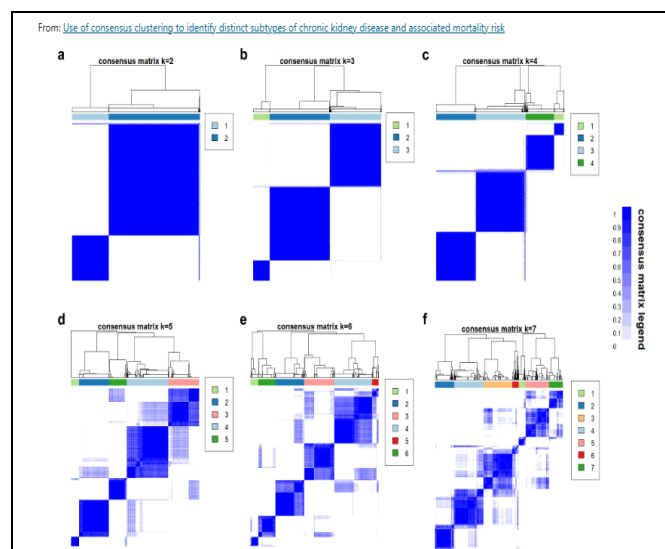


Figure: 26 Summary of Classifier Model with Full attribute Vs Class





Confusion matrix

The above result for all Cross Validation Folds from CVF = 4 to CVF=15 is the same

Table 9: Predicted values of Confusion matrix full attributes Vs Class (ckd-not-ckd)

Sr. No.	CVF	Predicted (a)	Predicted (b)	< Classified as
1	4	244	6	a = ckd
		0	150	b = not-ckd
2	5	245	5	a = ckd
		0	150	b = not-ckd
3	6	245	5	a = ckd
		0	150	b = not-ckd
4	7	244	6	a = ckd
		0	150	b = not-ckd
5	8	245	5	a = ckd
		0	150	b = not-ckd
6	9	245	5	a = ckd
		0	150	b = not-ckd
7	10	245	5	a = ckd
		0	150	b = not-ckd
8	11	244	6	a = ckd
		0	150	b = not-ckd
9	12	244	6	a = ckd
		0	150	b = not-ckd
10	13	244	6	a = ckd
		0	150	b = not-ckd
11	14	245	5	a = ckd
		0	150	b = not-ckd
12	15	245	5	a = ckd
		0	150	b = not-ckd

Figure: 27 Summary of Classifier Model with Full attribute Vs Class

Unsupervised – Clustering – Hierarchical – Euclidean & Manhattan Distance Function using No of Clusters 2,3,4& 5 test results: Full Attributes

Step 1: Using Euclidean Distance function No of Clusters 2,3,4& 5

Figure: 28 Preprocess Full attribute Vs Class

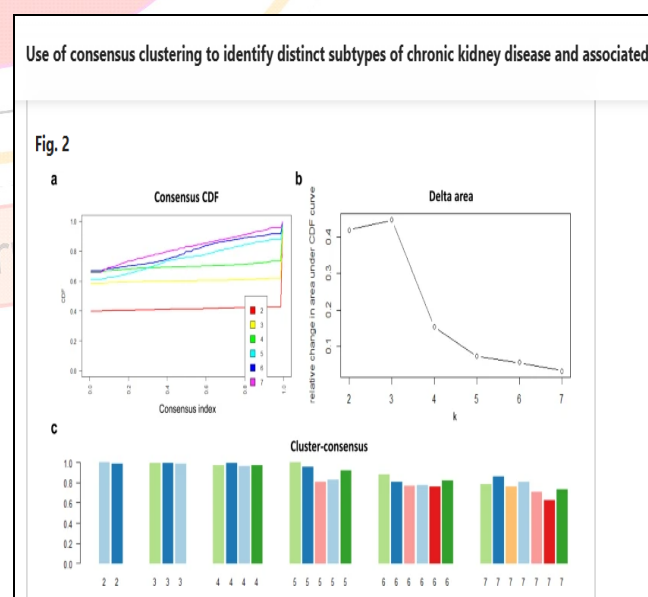


Figure: 29 Hierarchical clustering class with Euclidean function of Full attribute Vs Class with no of cluster 2,3,4& 5

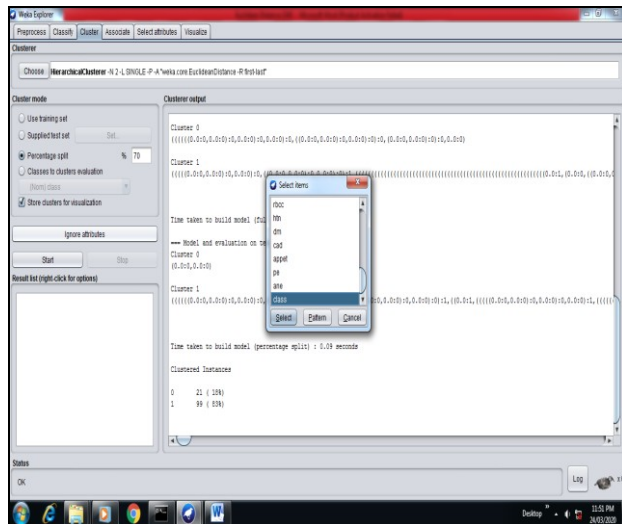


Figure: 30No of 02 cluster of hierarchical clustering class clustering output no of cluster 2,3,4& 5

STEP 1:

(Euclidean Distance) = 2 Cluster = Full Attributes

==== Run information =====

Scheme: weka.clusterers.HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

Relation: Chronic_Kidney_Disease_(RS Walse)-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: 400

Attributes: 25

- age
- bp
- sg
- al
- su
- rbc
- pc
- pcc
- ba
- bgr
- bu
- sc
- sod
- pot
- hemo
- pcv
- wbcc
- rbcc
- htn
- dm
- cad

- appet
- pe
- ane

Ignored:

class

Test mode: Classes to clusters evaluation on training data

==== Clustering model (full training set) =====

Cluster 0

Time taken to build model (full training data) : 0.73 seconds

==== Model and evaluation on training set =====

Clustered Instances

0	399 (100%)
1	1 (0%)

Class attribute: class

Classes to Clusters:

0	1	<-- assigned to cluster
249	1	ckd
150	0	notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Incorrectly clustered instances : 151.0 37.75 %

STEP 2:

(Euclidean Distance) = 3 Cluster = Full Attributes

Time taken to build model (full training data) : 0.73 seconds

==== Model and evaluation on training set =====

Clustered Instances

0	398 (100%)
1	1 (0%)
2	1 (0%)

Class attribute: class

Classes to Clusters:

0	1	2	<-- assigned to cluster
248	1	1	ckd
150	0	0	notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Cluster 2 <-- No class

Incorrectly clustered instances : 152.0 38 %

STEP 3:**(Euclidean Distance) = 4 Cluster = Full Attributes**

Time taken to build model (full training data) : 0.7 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 397 (99%)

1 1 (0%)

2 1 (0%)

3 1 (0%)

Class attribute: class

Classes to Clusters:

0 1 2 3 <-- assigned to cluster

247 1 1 1 | ckd

150 0 0 0 | notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Cluster 2 <-- No class

Cluster 3 <-- No class

Incorrectly clustered instances : 153.0 38.25 %

STEP 4:**(Euclidean Distance) = 5 Cluster = Full Attributes**

Time taken to build model (full training data) : 0.69 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 396 (99%)

1 1 (0%)

2 1 (0%)

3 1 (0%)

4 1 (0%)

Class attribute: class

Classes to Clusters:

0 1 2 3 4 <-- assigned to cluster

246 1 1 1 1 | ckd

150 0 0 0 0 | notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Cluster 2 <-- No class

Cluster 3 <-- No class

Cluster 4 <-- No class

Incorrectly clustered instances : 154.0 38.5 %

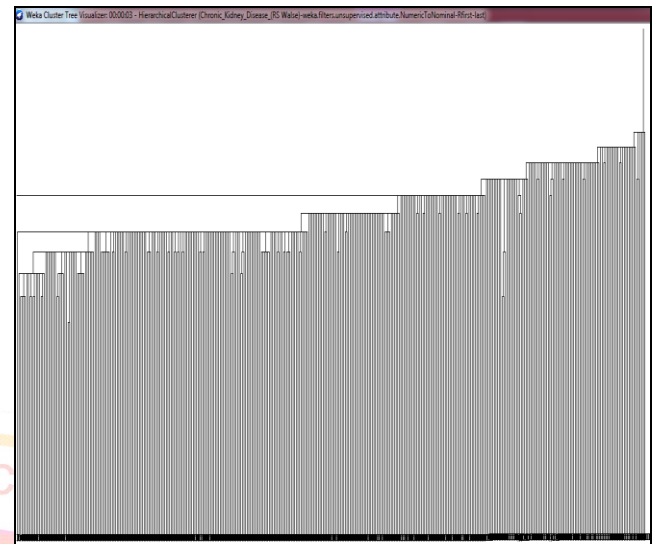


Figure: 31 Hierarchical clustering with Manhattan distance Function of Dendrogram

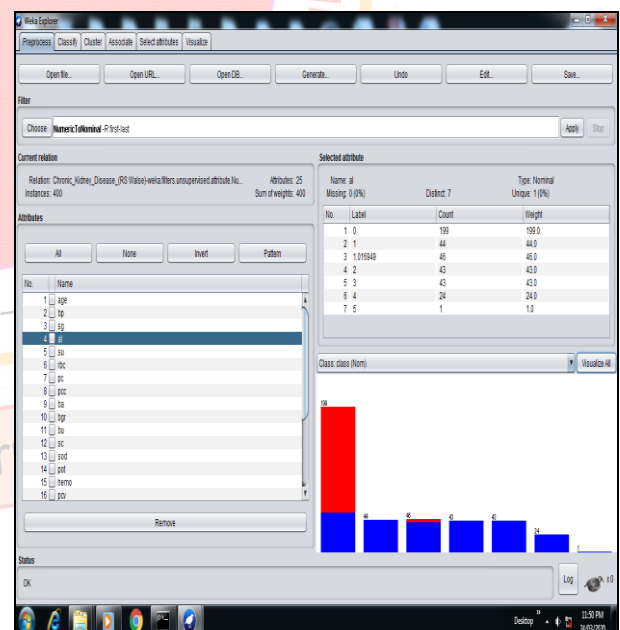
Figure 3: Performance accuracy by Class and Confusion matrix by decision tree.**Step 2: Using Manhattan Distance function No of Clusters 2,3,4& 5**

Figure: 32 Preprocess data set with full attributes for Manhattan Distance function

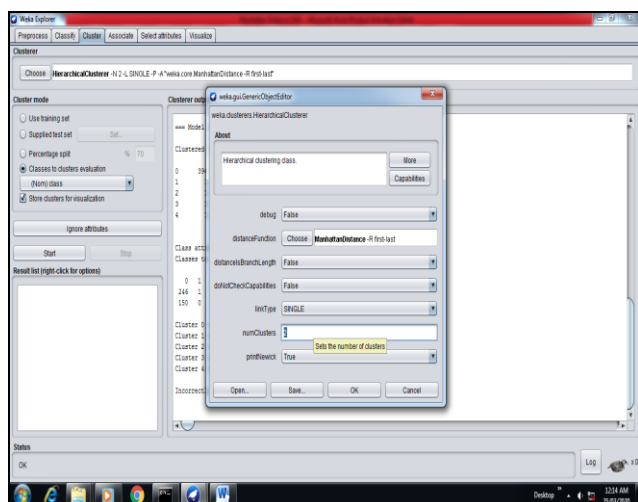


Figure: 33 Hierarchical Clustering class with Manhattan Distance function for no of cluster 2,3,4& 5

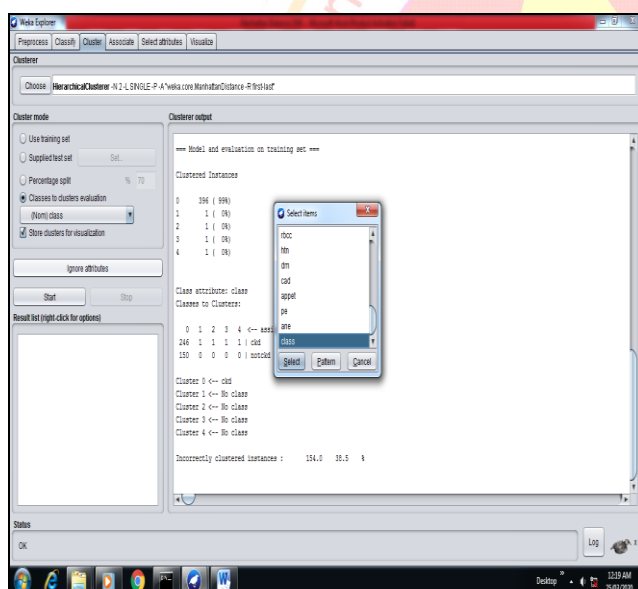


Figure: 34 Result of Hierarchical Clustering class with Manhattan Distance function for no of cluster 2,3,4& 5

STEP 1:

(Manhattan Distance) = 2 Cluster = Full Attributes

=== Run information ===

Scheme: weka.clusterers.HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.ManhattanDistance -R first-last"

Relation: Chronic_Kidney_Disease_(RS_Walse)-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: 400

Attributes: 25

Time taken to build model (full training data) : 0.72 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 399 (100%)

1 1 (0%)

Class attribute: class

Classes to Clusters:

0 1 <-- assigned to cluster

249 1 | ckd

150 0 | notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Incorrectly clustered instances : 151.0 37.75 %

STEP 2:

(Manhattan Distance) = 3 Cluster = Full Attributes

Time taken to build model (full training data) : 0.71 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 398 (100%)

1 1 (0%)

2 1 (0%)

Class attribute: class

Classes to Clusters:

0 1 2 <-- assigned to cluster

248 1 1 | ckd

150 0 0 | notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Cluster 2 <-- No class

Incorrectly clustered instances : 152.0 38 %

STEP 3:

(Manhattan Distance) = 4 Cluster = Full Attributes

Time taken to build model (full training data) : 0.73 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 397 (99%)

1 1 (0%)

2 1 (0%)
3 1 (0%)

Class attribute: class

Classes to Clusters:

0 1 2 3 <-- assigned to cluster
247 1 1 1 | ckd
150 0 0 0 | notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Cluster 2 <-- No class

Cluster 3 <-- No class

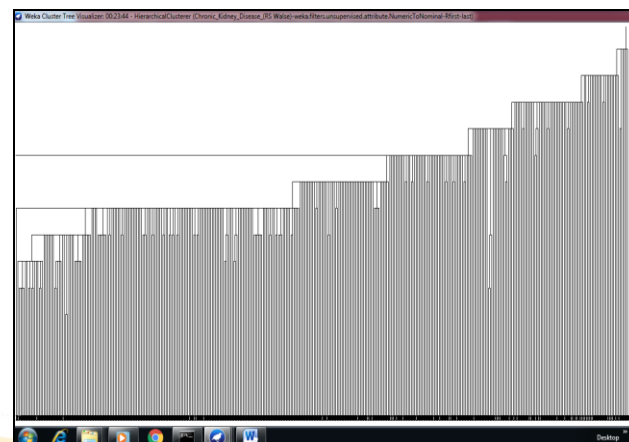


Figure: 35 Result of Hierarchical Clustering class with Dendrogram

STEP 4:

(Manhattan Distance) = 5 Cluster = Full Attributes

== Model and evaluation on training set ==

Clustered Instances

0 396 (99%)
1 1 (0%)
2 1 (0%)
3 1 (0%)
4 1 (0%)

Class attribute: class

Classes to Clusters:

0 1 2 3 4 <-- assigned to cluster
246 1 1 1 1 | ckd
150 0 0 0 0 | notckd

Cluster 0 <-- ckd

Cluster 1 <-- No class

Cluster 2 <-- No class

Cluster 3 <-- No class

Cluster 4 <-- No class

Incorrectly clustered instances : 154.0 38.5 %

5.Results and discussion

{a} Comparative statement of Supervised SMO-SVM- Classifier with one, two & full attributes with 70% split using Calibrator: Logistic and Kernel= Polynomial function

The research, the kidney dataset processed with different attributes (25), which contains 400 rows, i.e., instances and 25 attributes, means columns. The researcher has select every attributes to displays type of attributes, the type means Nominal, how many missing values present in the data set for each attribute viz. instances, how many distinct values are present in the dataset, distinct means different values, if we select attribute is shown Nom- in front of attribute-nom means nominal type. The numeric data gives summary of the overall data in the form of descriptive statistics. Also, if data is qualitative then it treated as an attribute class and it shows in the form of label count and its weight in the form of true/false or yes/no.

Table 10: Correctness by Class values

	No of Attributes	Class	TPR	FPR	Precision	Recall	F-Measure	MC	ROC Area	PRC Area
*CVF=4	01 (ht nVs Cl ass)	CKD	0.588	0.000	1.000	0.588	0.741	0.590	0.794	0.845
		No t-CKD	1.000	0.412	0.593	1.000	0.744	0.590	0.794	0.593

		We igh t Av g	0. 74 3	0. 15 5	0. 84 7	0. 7 4 3	0.7 42	0 . 5 9 0	0.7 94	0.7 51
** C V F=4	02 (ht n, ap pet Vs Cl ass)	CK D	0. 73 7	0. 00 0	1. 00 0	0. 7 3 7	0.8 48	0 . 7 1 2	0.8 68	0.9 04
		No t- CK D	1. 00 0	0. 26 3	0. 68 8	1. 0 0 0	0.8 15	0 . 1 7 2	0.8 68	0.6 88
		We igh t Av g	0. 83 3	0. 09 6	0. 88 6	0. 8 3 3	0.8 36	0 . 1 7 2	0.8 68	0.8 24
*** C V F=5	24 (F ull Vs Cl ass)	CK D	0. 98 0	0. 00 0	1. 00 0	0. 9 8 0	0.9 90	0 . 9 7 4	0.9 90	0.9 92
		No t- CK D	1. 00 0	0. 02 0	0. 96 8	1. 0 0 0	0.9 84	0 . 9 7 4	0.9 90	0.9 68
		We igh t Av g	0. 98 8	0. 00 8	0. 98 8	0. 9 8 8	0.9 88	0 . 9 7 4	0.9 90	0.9 83

* Found max and same accuracy for 5 to 15 CVF

** Found max and same accuracy for 5 to 15 CVF

*** 5,6,8,9,10,14,15 found max and same accuracy CVF

Table 10 shows the correctness by Class values Supervised- classify – SMO- SVM- with attributes 2,3 & Full Attributes –Applying Cross validation fold from 4 to 15 with 70% split, and applying the **Calibrator**: Logistic and **Kernel**: Polynomial function and it shows the comparative study of the result and found the accurate prediction value, therefore research perfectly state that the increasing no of cross validation folds with respective increasing attributes the accuracy of the correctness of the class values increases shows in the form of ROC Area CVF=4 ROC Area = 0.794 using one attribute , CVF=4 ROC Area = 0.868 using two attributes and CVF=5 ROC Area=0.990. Hence we

found best and Novel Predictive Module with the help of SMO- SVM- Classifier Class Module on the basis of Calibrator and Kernel function.

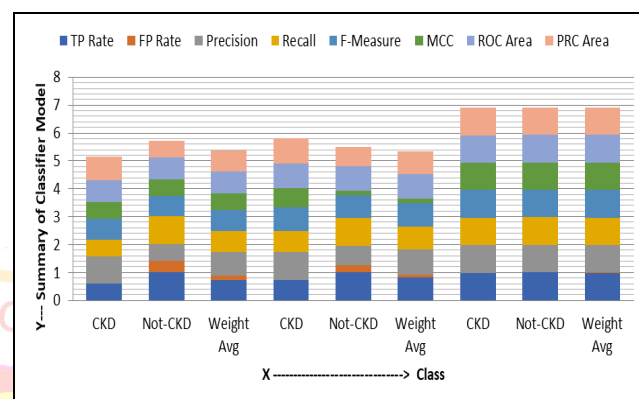


Figure: 36 Comparative study of Correctness by class values of SMO-SVM- Classifier with 1,2 and full attributes with 70% and CVF from 4 to 15.

Table 11: Summary of Classifier model (Train set data)

Sr. No	Particulars	01 (htn Vs Class)	02 (htn, appet Vs Class)	24 (Full Vs Class)
		CV Folds	CV Folds	CV Folds
		4 Found max and same accuracy for 5 to 15 CVF	4 Found max and same accuracy for 5 to 15 CVF	5,6,8,9,10,14,15 found max and same accuracy CVF
1	Test mode: 70% train, 30% test	70.0% train	70.0% train	70.0% train
2	Correctly Classified Instances	74.25	83.33	98.75
3	Incorrectly Classified Instances	25.75	16.66	1.25
4	Kappa statistic	0.517	0.6725	0.9735
5	Mean absolute error	0.2575	0.1667	0.0125
6	Root mean squared error	0.5074	0.4082	0.1118

7	Relative absolute error	54.90	35.6241	2.6656
8	Root relative squared error	104.81	84.6877	23.094
9	Total Number of Instances	400	400	400

(CVF= Cross Validation Folds)

Table 11 shows the Summary of classifier model with train set data, of SMO- classifier by applying one, two and full 25 attributes to test the result and getting some of the useful and accurate result, so as to use our research to further researcher for further prediction of data and effectively useful of the SMO – SVM classifier algorithm, further table 11 shows the how accuracy will be increased on the basis of different parameters like Cross Validation Folds just testing from 4 to 15, accuracy of the result is higher as compared to the earlier result in the form of Correctly Classifier Instances is 74.25 % , 83.33% and 98.75% respectively. Therefore also found the best Novel Summary of Classified module with highest accuracy of CCI.

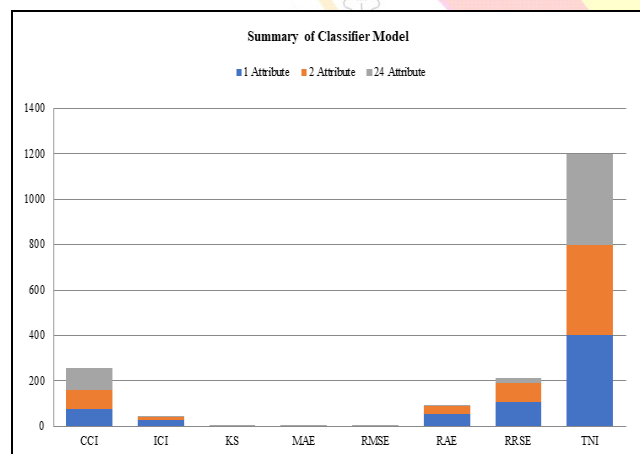


Figure: 37Comparative study of Summary of Classifier Module of SMO-SVM- Classifier with 1,2 and full attributes with 70% and CVF from 4 to 15.

Table 12:Comparative statement of Confusion Matrix SMO-SVM- Classifier with one, two & full attributes with 70% split using Calibrator: Logistic and Kernel= Polynomial function

Sr . N o.	No of Attributes	CVF	Predicted (a)	Predicted (b)	< - Classif ied as
1	One Attribute	4-15	147	103	a = ckd
			0	150	b = not-ckd
2	Two Attributes	4	147	103	a = ckd
			0	150	b = not-ckd
		5-15	172	78	a = ckd
			0	150	b = not-ckd
3	Full Attributes	4,7,11,12,13	244	6	a = ckd
			0	150	b = not-ckd
		5,6,8,9,10,14,15	245	5	a = ckd
			0	150	b = not-ckd

(CVF = Cross Validation Folds)

Table 12 shows the Comparative statement of Confusion Matrix SMO-SVM- Classifier with one, two & full attributes with 70% split using Calibrator: Logistic and Kernel= Polynomial function, it plot the threshold Curve and Cost benefit curve for class ckd and Not-ckd in the form of % population of confusion matrix and cost matrix. Confusion matrix of predicted a & b also cost matrix a & b with the random and gain value. The researcher also find the research output is using full length of attributes it creates minimum Confusion matrix with accurate predicted class value with minimum no of iterations of clusters.

{b} Comparative statement of Un-Supervised Hierarchical Clustering algorithm using Euclidean and Manhattan distance function for the no of cluster 2,3,4 & 5.

Comparative statement of the Hierarchical clustering algorithm using Euclidean nad Manhattan distance function result with accuracy.

❖ Euclidean Distance:

=== Run information ===

Scheme: weka.clusterers.HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

Relation:Chronic_Kidney_Disease_(RS Walse)-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

❖ Manhattan Distance:

=== Run information ===

Scheme:weka.clusterers.HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.ManhattanDistance -R first-last"

Relation:Chronic_Kidney_Disease_(RS Walse)-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

=== Clustering model (full training set) ===

Table 13:Comparative statement of Unsupervised Hierarchical Clustering algorithm using Euclidean and Manhattan Distance Function of full attributes with 70% split. (Cluster 2,3,4& 5)

Sr. No.	No. of Cluster	Euclidean Distance		Manhattan Distance	
		Clustered Instances	Classes to Clusters	Clustered Instances	Classes to Clusters
1	2	0 399(100%) 1 1(0%)	249 1 ckd 150 0 notckd	0 399(100%) 1 1(0%)	249 1 ckd 150 0 notckd
2	3	0 398 (100%) 1 1 (0%) 2 1 (0%)	248 1 ckd 150 0 0 notckd	0 398 (100%) 1 1 (0%) 2 1 (0%)	248 1 ckd 150 0 0 notckd
3	4	0 397 (99%) 1 1 (0%) 2 1 (0%) 3 1 (0%)	247 1 1 ckd 150 0 0 0 notckd	0 397 (99%) 1 1 (0%) 2 1 (0%) 3 1 (0%)	247 1 1 ckd 150 0 0 0 notckd
4	5	0 396 (99%) 1 1 (0%)	246 1 1 ckd 150 0	0 396 (99%) 1 1 (0%)	246 1 1 ckd 150 0

2	1 (0%)	0 0	2	1 (0%)	0 0
3	1 (0%)	notckd	3	1 (0%)	notckd
4	1 (0%)		4	1 (0%)	

Table 13 describes the researcher also compare the result by using Unsupervised Hierarchical Clustering algorithm of Chronic Kidney Disease data with the support of Euclidean and Manhattan Distance clustering function with the help of 70% Split by setting the properties No of Clusters from 2,3,4, & 5 of Euclidean and Manhattan distance function. And finally research found the best and accurate prediction is the, Increasing the Number of Clusters of full attributes of Chronic Kidney Disease data accuracy of creating no of clusters as well as classes to Clusters increases. One more main comparative results also found, both the Euclidean and Manhattan distance function output result is also same for No of clusters 2,3,4& 5.

6. Conclusion

The results showed that the Novel discovered module, which combines both classification and clustering, turned out to be the most accurate classifier. It achieved an accuracy of 98.75%, meaning it correctly classified instances. This was better than previous results where only one or two attributes were used. We also found that using a Calibrator with Logistic and Kernel functions, specifically a Polynomial function, with cross-validation folds of 5, 6, 8, 9, 10, 14, and 15, yielded the same results in the confusion matrix.

We determined the best predictive model for Supervised-SMO in WEKA by conducting three tests. These tests used a logistic calibrator and a polynomial kernel function, with cross-validation folds ranging from 4 to 15 (meaning we trained on some data and tested on the rest) with a 70% split. The final finding was that increasing the number of attributes led to improvements in accuracy for correctly classified instances (CCI), the ROC Area value, and the confusion matrix values.

Similarly, our research confirmed the predictions for the Unsupervised Hierarchical Clustering algorithm. When we used all the

attributes, we found that increasing the number of clusters from 2, 3, 4, and 5, using both Euclidean and Manhattan functions, resulted in better and more accurate clusters. The method we used clearly outlines the practical steps involved. For further research, some of the attributes were measured RBC count, HP, Diabetes Mellitus, CAD, Appetite, Pedal Edema, Anemia, etc. Now looking ahead, this research aims to equip doctors and the medical field with a tool to identify whether a patient has Chronic Kidney Disease (CKD) or not, based on their existing health indicators. The goal is to slow down the rise in CKD cases and prevent further kidney damage. Data mining is proving to be quite useful in forecasting future kidney issues. In this study, we've examined three algorithms. Specifically, we've explored chronic kidney disease prediction using the SMO-SVM classifier algorithm, a machine learning technique, with Logistic as the calibrator and a Polynomial function for the kernel. Ultimately, our research intends to analyze kidney disease, or its potential development in the future. This will help current kidney patients understand the causes of their condition and allow those without kidney disease to assess their future risk, potentially saving them the need for extra tests and expenses.

References

1. Arasu, S. Dilli, and R. Thirumalaiselvi. "A novel imputation method for effective prediction of coronary Kidney disease." In 2017 2nd International Conference on Computing and Communications Technologies (ICCCT), pp. 127-136. IEEE, 2017.
2. Ariff, M. H., I. Ismarani, and N. Shamsuddin. "RFID based systematic livestock health management system." In 2014 IEEE Conference on Systems, Process and Control (ICSPC 2014), pp. 111-116. IEEE, 2014.
3. Bharara, Sanyam, A. SaiSabitha, and AbhayBansal. "A review on knowledge extraction for business operations using data mining." In 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence, pp. 512-518. IEEE, 2017.
4. Chuan, Zou, Tang Ying, Bai Li, ZengYuqun, and Lu Fuhua. "Application of clustering analysis to explore syndrome evolution law of peritoneal dialysis patients." In 2013 IEEE International Conference on Bioinformatics and Biomedicine, pp. 23-26. IEEE, 2013.
5. Due ThanhAnhLuong, Dept of the computer. Sci&Engi, uni, at Buffalo Buffalo, NY, USAA K- Means Approach to Clustering disease Progressions IEEE Keywords, Sep 2017
6. Güllüoğlu, SabriSerkan. "Segmenting customers with data mining techniques." In 2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC), pp. 154-159. IEEE, 2015.
7. Jinyin, Chen, et al. "A novel cluster center fast determination clustering algorithm." Applied Soft Computing 57 (2017): 539-555.
8. Khanna, Umesh. "The economics of dialysis in India." Indian journal of nephrology 19, no. 1 (2009): 1.
9. Kunwar, Veenita, et al. "Chronic Kidney Disease analysis using data mining classification techniques." 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence). IEEE, 2016.
10. Narander Kumar, SabitaKLhatri, Department of Computer, 3rd IEEE International Conference on Computational Intelligence and Communication Technology (IEEE-CICT 2017) "Implementing WEKA for medical data classification and early disease prediction 978-1-50, 2017
11. Uboltham, Issariya, NakornthipPrompoon, and Wirichada Pan-Ngum. "AKIHelper: Acute kidney injury diagnostic tool using KDIGO guideline approach." In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1-6. IEEE, 2016.
12. Khanna, Umesh. "The economics of dialysis in India." Indian journal of nephrology 19, no. 1 (2009): 1.
13. Varma, P. P. "Prevalence of chronic kidney disease in India-Where are we heading?." Indian journal of nephrology 25, no. 3 (2015): 133.
14. https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease