DIMENSIONALITY REDUCTION OF HIGH DIMENSIONAL DATA

SUMMER INTERNSHIP PROJECT SUMMARY

Done by

BARATHKUMAR V (2016103015)
MARIAM JAMILAH (2016107026)
SOWMYA S (2016103067)

COLLEGE OF ENGINEERING GUINDY

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2018

1. PROBLEM STATEMENT

High Dimensionality means the number of attributes is greater than the number of observations in a dataset. Such data is very difficult to handle and hence it is important to reduce the dimensionality of the dataset. The main problem with the data collected is that it is dirty - contains missing values and noise in it. These anomalies make the data inconsistent, harder to process and give us inaccurate results upon processing. In order to avoid this, data preprocessing is performed. The data that results from the preprocessing stage is complete and consistent. This data is then subjected to further processing such as feature selection to get a reduced feature subset. This subset with reduced number of features gives accurate results while reducing the run time. The aim of the project is to reduce the number of features in the given dataset, required to predict the class labels and at the same time achieve an accurate result using feature selection methods.

2. PROBLEM DESCRIPTION

The project was done to test if the accuracy of the test data remained high after applying the various feature selection algorithms to the training data.

- A model dataset on Dorothea drug testing was used for testing the module. The dataset was downloaded from UCI Machine Learning Repository.
- R Studio was used as the module to train and tests the data model for sampling the algorithms.
- The downloaded train data was uploaded to the R studio and the preprocessing was carried out on the dirty data.
- The module is first supplied with training data and the accuracy of this data is tested using a Decision Tree classifier algorithm, C4.5.

- This classifier algorithm is available in the package RWeka, which is an additional package installed in R studio.
- The training data is then processed further by applying a feature selection algorithm to it.
- The test data obtained is tested for accuracy. The results are noted and recorded for further analysis.

3. WORKFLOW OF PROJECT

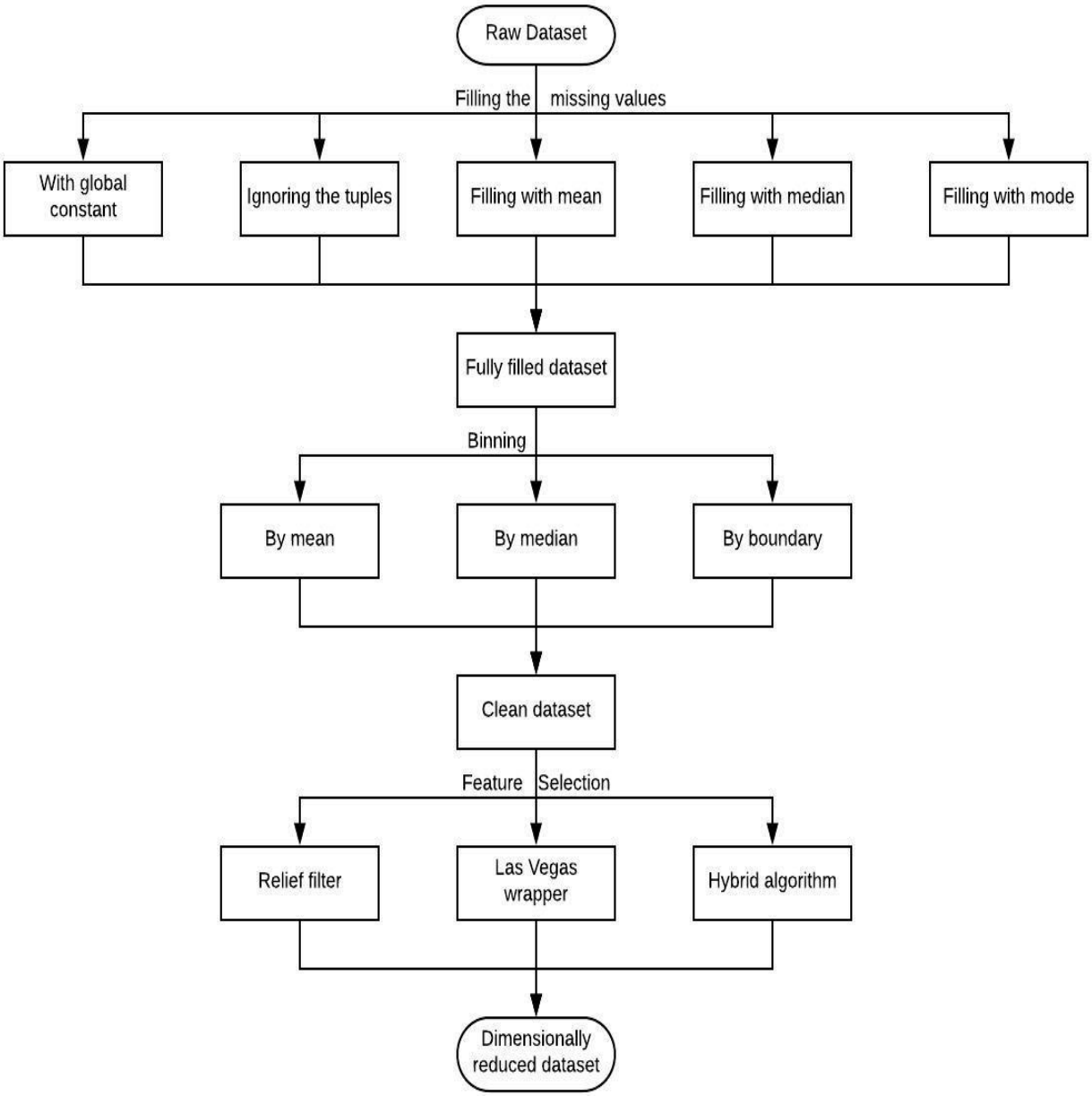

Fig. 1 Workflow diagram of the project

4. IMPLEMENTATION AND TOOLS

TOOLS USED:

*Rstudio and R console

-Package imported: RWeka

*Java Oracle

IMPLEMENTATION:

Data preprocessing:

The availability of enormous amount of data in the real world has led to the creation of databases and datasets containing dirty data - meaning incomplete, noisy and inconsistent data

- Incomplete: lacking attribute values or certain attributes of interest
- Noisy and inconsistent: containing errors, discrepancies or outliers

Therefore, it is extremely important to have clean data of high quality as poor quality data will not generate efficient and accurate results when data mining techniques are applied. Thus it is important to improve the efficiency and quality of data and this can be achieved by data preprocessing. Data preprocessing involves the cleaning of data. Data cleaning techniques involve the filling of missing values and removal of noisy and inconsistent data.

Missing values maybe due to

- equipment malfunctioning
- inconsistent with other recorded data and thus deleted
- data not entered due to misunderstanding
- certain data may not be considered important at the time of entry
- not register history or changes of the data

In case of any missing value, we can

- manually enter the value
- assign a constant value
- eliminate rows with missing values
- fill the missing values with the mean of the column
- fill the missing values with the median of the column
- fill the missing values with the mode of the column

Noisy and inconsistent data maybe due to

- faulty data collection instruments
- data entry problems data entry problems
- data transmission problems
- technology limitation
- inconsistency in naming convention

In case of noisy and inconsistent data, we perform binning. In binning, the data is sorted and partioned into equi-width bins and then the noisy data is smoothened by

- bin mean
- bin median
- bin boundary

Feature Selection:

I.FILTER METHOD:

Input : O, the set of all objects; C, the set of all conditional features; its, the no of iterations; Ɛ, weight threshold value.

Output : R, the feature subset

(1) R <- { }

(2) for each Ws, Ws <- 0

(3) for each i = 1…its

(4) choose an object x in O randomly

(5) calculate x's nearHit (nH) and nearMiss (nM)

(6) for each j =1…|C|

(7) $W_j = W_j + |x^{(i)} – nM^{(i)}(x)| - |x^{(i)} – nH^{(i)}(x)|$

(8) for each j =1…|C|

(9) if $W_j >= \text{Ɛ}$; R <- RU{j}

(10) return R

Fig 2. Flow of the Relief Filter Algorithm

Filter method selects the feature subset on the basis of the intrinsic characters of the data, independent of the machine learning algorithm. Among that RELIEF is considered one of the most successful algorithms used for assessing the quality of features due to its simplicity and effectiveness. The key idea of RELIEF is to iteratively estimate the feature weights according to their ability to discriminate between neighbouring patterns. In this algorithm, a random object is selected from the dataset and the nearest neighbouring sample with the same class label (nearHit) and different class label (nearMiss) are identified. The nearest neighbouring sample of an object means the object with the most number of features having same value. The weight of the each feature is updated by $W_j = W_j + |x^{(i)} – nM^{(i)}(x)| - |x^{(i)} – nH^{(i)}(x)|$. After running the for loop for the number of iterations, the weight threshold value is calculated by taking the mean of weight of all the features. Then select the features which is having the weight greater than the threshold value.

II. WRAPPER METHOD:

Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model is used to evaluate a combination of features and assign a score based on model accuracy. As wrapper methods train a new model for each subset, they are very computationally intensive, but usually provide the best performing feature set for that particular type of model. At times, Filter methods have been used as a preprocessing step for wrapper methods, allowing a wrapper to be used on larger problems.

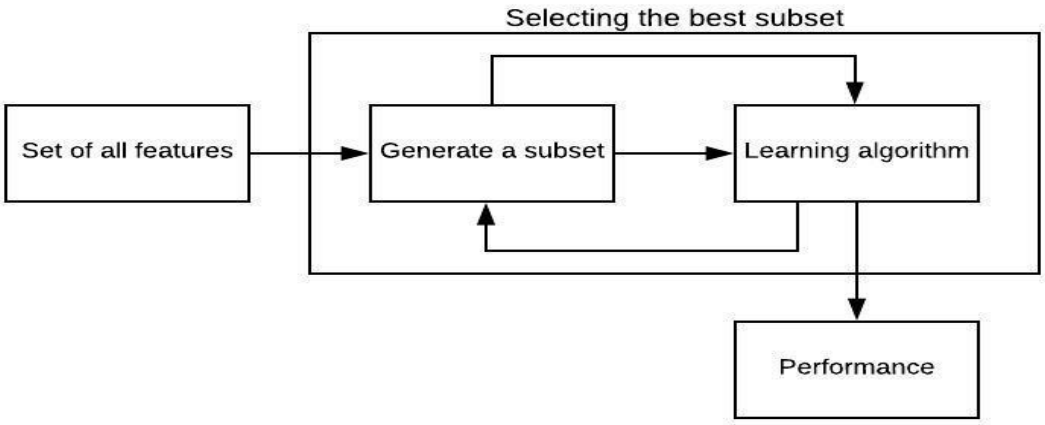

Fig. 3 Flow of the Las Vegas Wrapper Algorithm

LAS VEGAS WRAPPER ALGORITHM:

INPUT:

C - the set of conditional features;

K - update threshold

ε - error threshold

ALGORITHM:

(1) R←C; k = 0

(2) **while** ε not updated for K times

(3) T←randomFeatureSubset()

(4) εt = learn(T)

(5) **if** (εt < ε) **or** (εt == ε **and** |T| < |R|)

(6) **output** T

(7) k = 0; ε = εt; R←T

(8) k = k+1

(9) ε = learn(R); **return** R

OUTPUT:

R - the feature subset

EXPLANATION:

The Las Vegas Wrapper algorithm uses random subset creation that guarantees that given enough time, the optimal solution will be found. It produces intermediate solutions while working towards better ones that result in a lower classification error.

In this algorithm, initially the full set of conditional features is taken as the best subset. The algorithm then generates a random feature subset and evaluates the error threshold εt using a inductive learning algorithm, here C4.5 is used. It compares εt with ε,

- if εt < ε or εt == ε and |T| < |R|, then εt becomes the new ε, T becomes the new R and k becomes 0 and then the algorithm continues to generate random subsets.
- if εt > ε or εt == ε and |T| > |R|, the algorithm continues to generate random subsets until K times.

This algorithm requires two threshold values to be supplied: ε, the classification error threshold and the value K, used to determine when to exit the algorithm due to there being no recent updates to the best subset encountered so far.

III. HYBRID METHOD:

EXPLANATION:

The Hybrid Algorithm is defined as a combination of both Iterative Relief Filter method and Las Vegas wrapper method. The complete and consistent dataset is considered and the Iterative filter method algorithm is applied to it. With the feature subset obtained thus, we apply the Las Vegas wrapper method to it to get a further reduced feature subset. Thus with the now doubly reduced feature set, we test its accuracy by applying the C4.5 classification algorithm and the results are recorded.

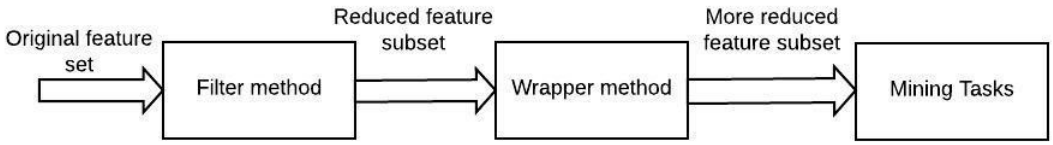

Fig. 4 Flow of the Hybrid Algorithm

CLASSIFICATION:

The C4.5 is a learning algorithm used to generate a decision tree, it is often referred to as a statistical classifier. It is implemented in R using the RWeka package. This package uses J48 which is an open source Java implementation of C4.5 and it requires the class labels to be in the string type for its successful running. C4.5 is used due to its relatively fast induction time which is a critical factor in wrapper based algorithms.

5. ANALYSIS

ITERATIVE RELIEF FILTER ALGORITHM

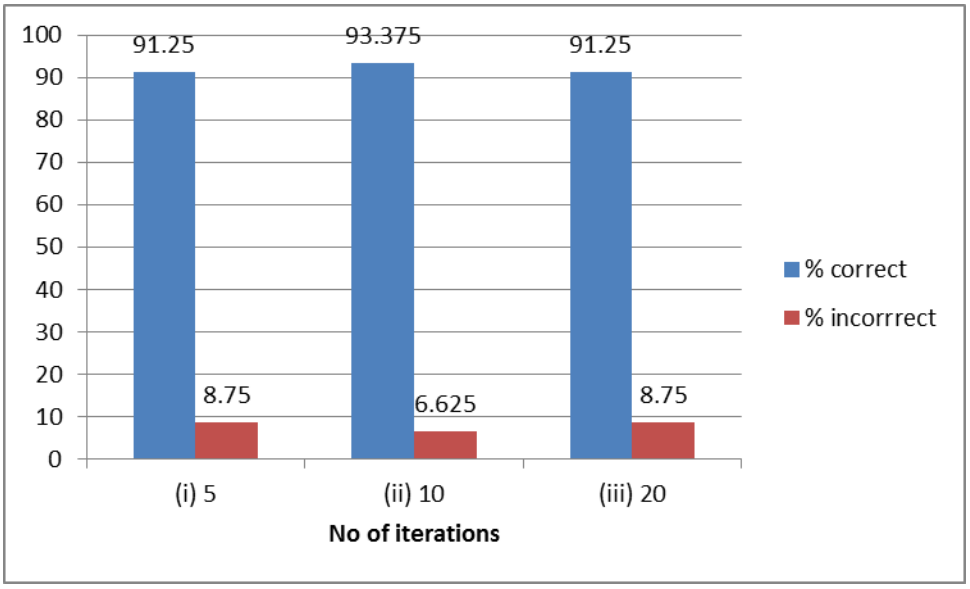

Fig.5 Iterative Relief Filter Algorithm for varying "its"

From the fig. 5 - (i) we infer that, after applying the Relief Filter algorithm where the number of iterations is 5, the number of attributes have reduced from 6061 to 852 and the percentage of correctly classified attributes thus obtained is 91.25%.Therefore with 852 attributes we can obtain an accuracy of 91.25%. Hence dimensionality reduction is achieved.

From the fig. 5 - (ii) we infer that, after applying the Relief Filter algorithm where the number of iterations is 10, the number of attributes have reduced from 6061 to 1112 and the percentage of correctly classified attributes thus obtained is 93.375%. Therefore with 1112 attributes we can obtain an accuracy of 93.375%. Hence dimensionality reduction is achieved.

From the fig. 5 - (iii) we infer that, after applying the Relief Filter algorithm where the number of iterations is 20, the number of attributes have reduced from 6061

to 1436 and the percentage of correctly classified attributes thus obtained is 91.25%. Therefore with 1436 attributes we can obtain an accuracy of 91.25%. Hence dimensionality reduction is achieved.

LAS VEGAS WRAPPER ALGORITHM

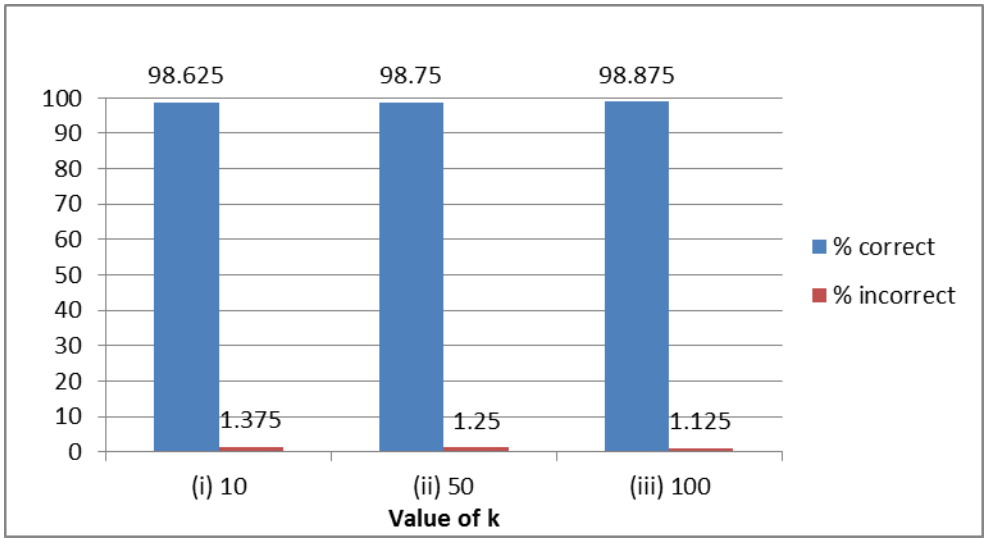

Fig. 6: Las Vegas Wrapper Algorithm for varying "k"

From the fig. 6 - (i) we infer that, after applying the Las Vegas Wrapper algorithm where K=10, the percentage of correctly classified attributes has increased from 94.25% to 98.625% and the number of attributes have reduced from 6061 to 3034. Therefore with 3034 attributes we can obtain an accuracy of 98.625%which is higher than the accuracy of the clean dataset. Hence dimensionality reduction is achieved.

From the fig. 6 - (ii) we infer that, after applying the Las Vegas Wrapper algorithm where K=50, the percentage of correctly classified attributes has increased from 94.25% to 98.75% and the number of attributes have reduced from 6061 to 4349. Therefore with 4349 attributes we can obtain an accuracy of 98.75% which is higher

than the accuracy of the clean dataset. Hence dimensionality reduction is achieved.

From the fig. 6 - (iii) we infer that, after applying the Las Vegas Wrapper algorithm where K=100, the percentage of correctly classified attributes has increased from 94.25% to 98.875% and the number of attributes have reduced from 6061 to 3427.Therefore with 3427 attributes we can obtain an accuracy of 98.875% which is higher than the accuracy of the clean dataset. Hence dimensionality reduction is achieved.

HYBRID ALGORITHM

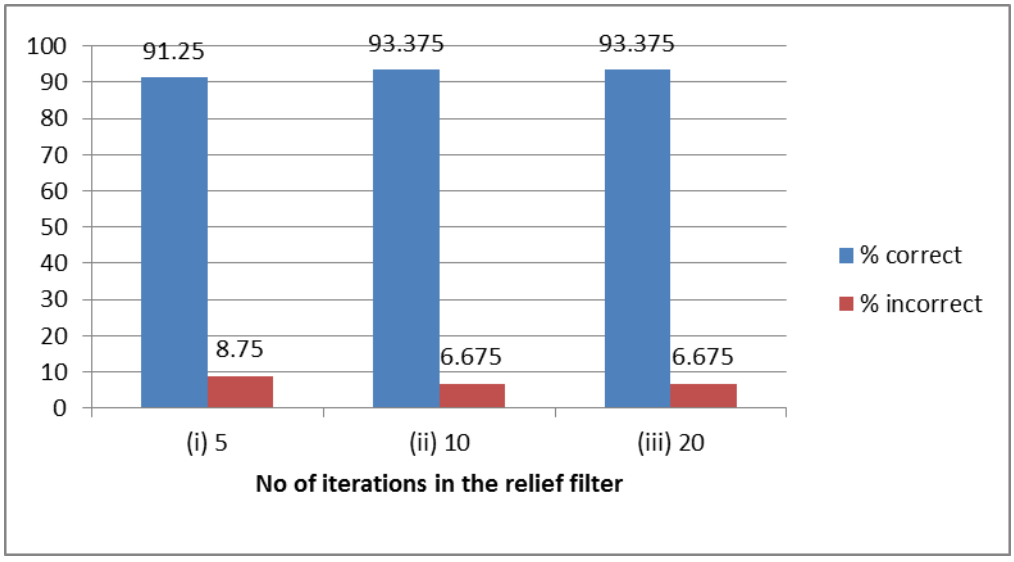

Fig 7: Hybrid Algorithm for k=10, and varying "its"

From the fig. 7 - (i) we infer that, after applying the Hybrid algorithm where the number of iterations is 5 and K= 5, the number of attributes have reduced from 6061 to 132 and the percentage of correctly classified attributes thus obtained is 91.25%. Therefore with 132 attributes we can obtain an accuracy of 91.25%. Hence dimensionality reduction is achieved.

From the fig. 7 - (ii) we infer that, after applying the Hybrid algorithm where the number of iterations is 10 and K= 5, the number of attributes have reduced from 6061 to 1102 and the percentage of correctly classified attributes thus obtained is 93.375%.Therefore with 1102 attributes we can obtain an accuracy of 93.375%. Hence dimensionality reduction is achieved.

From the fig. 7 - (iii) we infer that, after applying the Hybrid algorithm where the number of iterations is 20 and K= 5, the number of attributes have reduced from 6061 to 1436 and the percentage of correctly classified attributes thus obtained is 95.375%. Therefore with 1436 attributes we can obtain an accuracy of 95.375%. Hence dimensionality reduction is achieved.

6. CONCLUSION

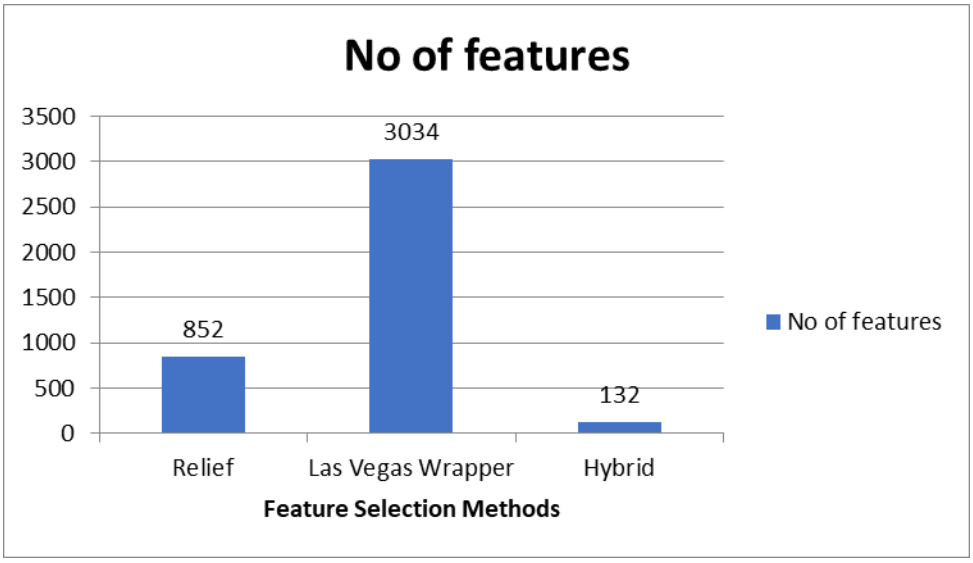

Fig 8: Comparing no. of reduced features obtained after applying the algorithms

From the fig,8 we conclude that, the algorithm which is best suited for feature selection is Hybrid. It first executes the Relief Filter algorithm and generates a subset and this subset is then subjected to the Las Vegas Wrapper algorithm which finally delievers the reduced feature subset.

In the first stage of the Hybrid algorithm, Relief Filter algorithm assigns weights to all the attributes and then filters them to generate a subset whereas in the second stage, the Las Vegas Wrapper algorithm selects random features from the subset generated in the first stage and forms a subset which is then evalvated to result in the final reduced feature subset.

Thus by varying the number of iterations in Relief Filter and the update threshold value in Las Vegas Wrapper, we can obtain a subset with as less as 136 features resulting in an accuracy of 91.25%. Thus it is the combination of the Relief Filter and the Las Vegas Wrapper algorithms in the Hybrid algorithm that results in a complete and thorough analysis of all the attributes yielding the best result.

7. APPENDIX I

SCREENSHOTS OF RESULTS

ITERATIVE RELIEF ALGORITHM

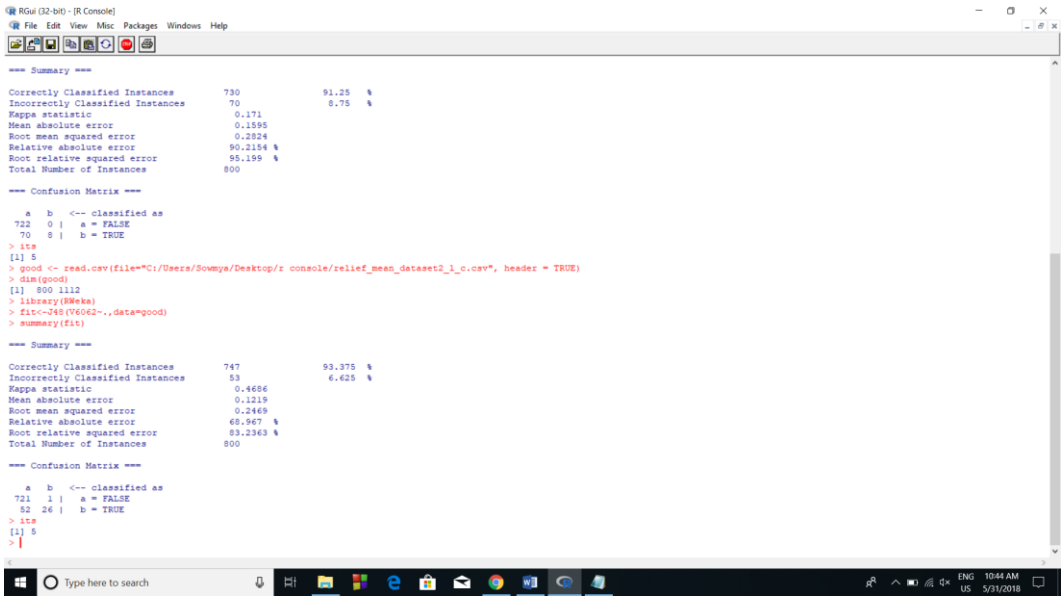

Fig. 9 Relief filter algorithm when its=5

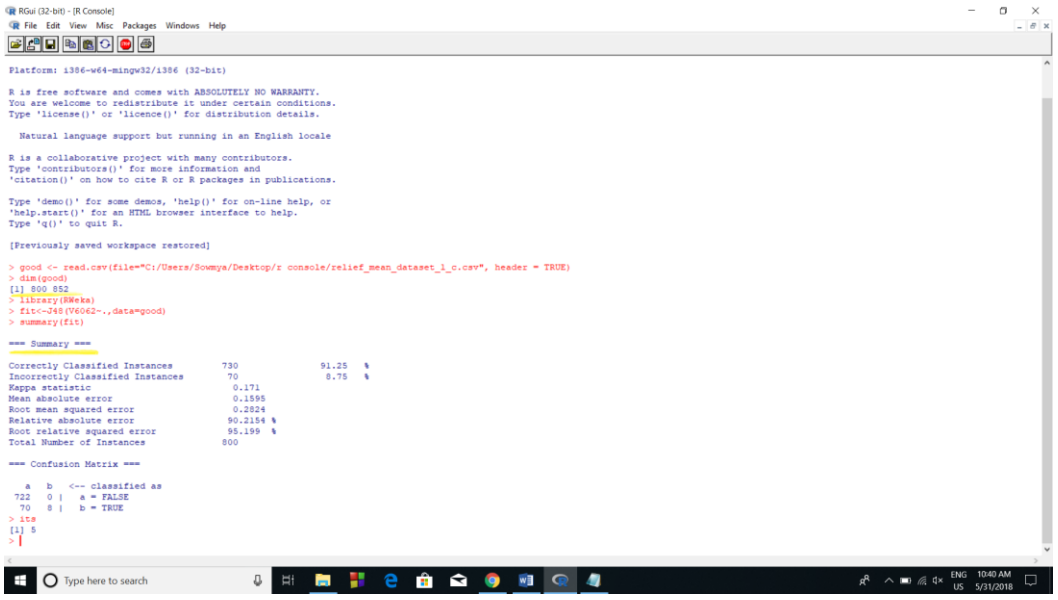

Fig. 10 Relief filter algorithm when its=10

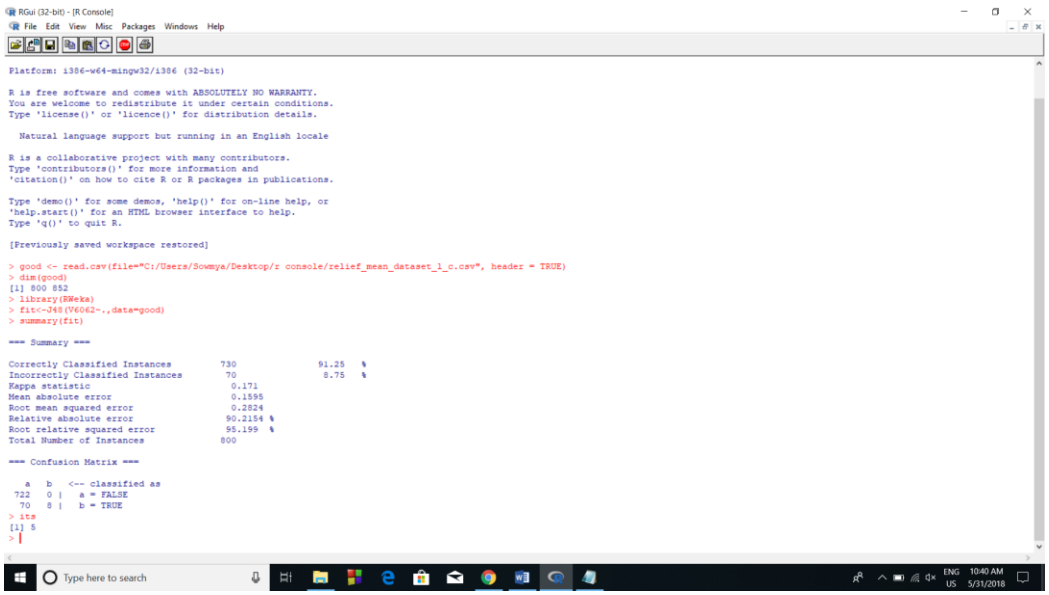

Fig. 11 Relief filter algorithm when its=20

LAS VEGAS WRAPPER ALGORITHM

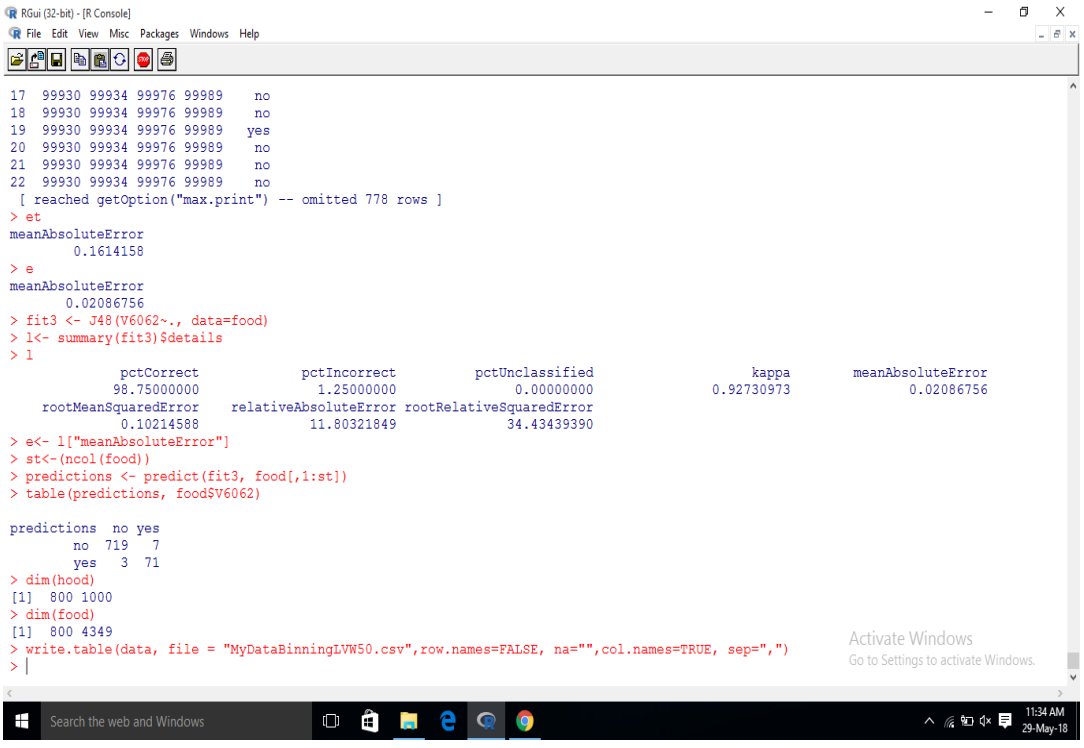

Fig. 12 Las Vegas Wrapper Algorithm when k=10

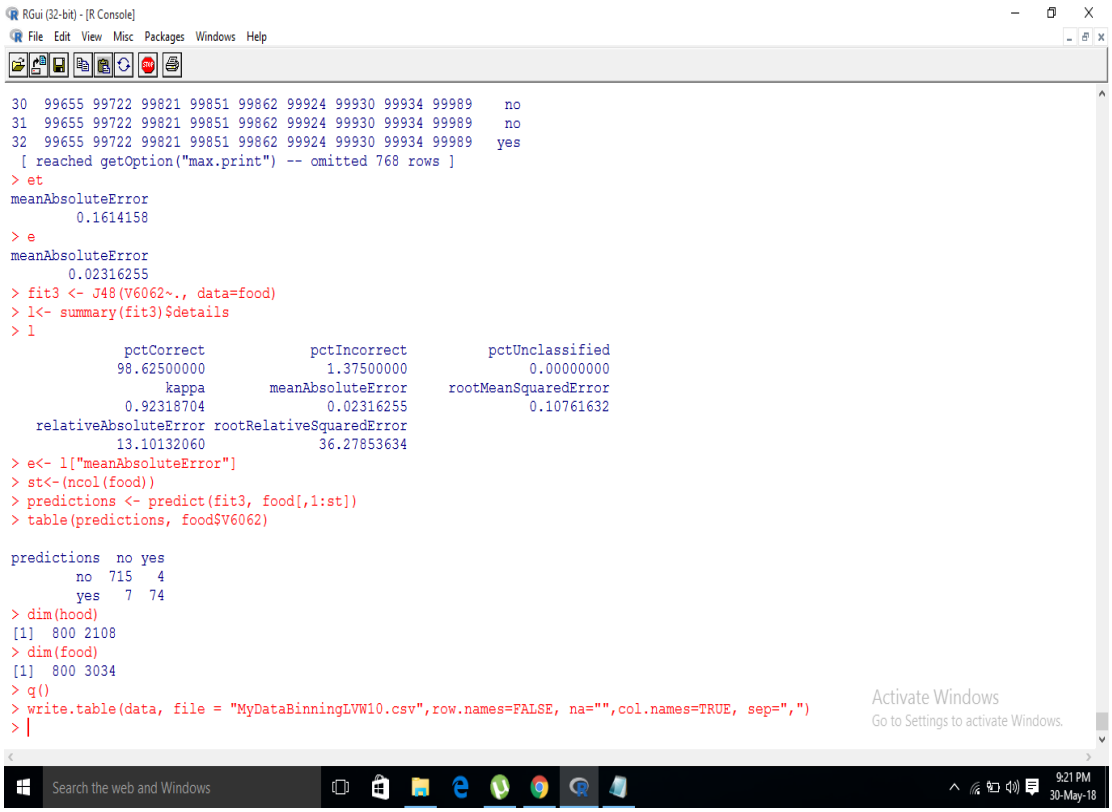

Fig. 13 Las Vegas Wrapper Algorithm when k=50

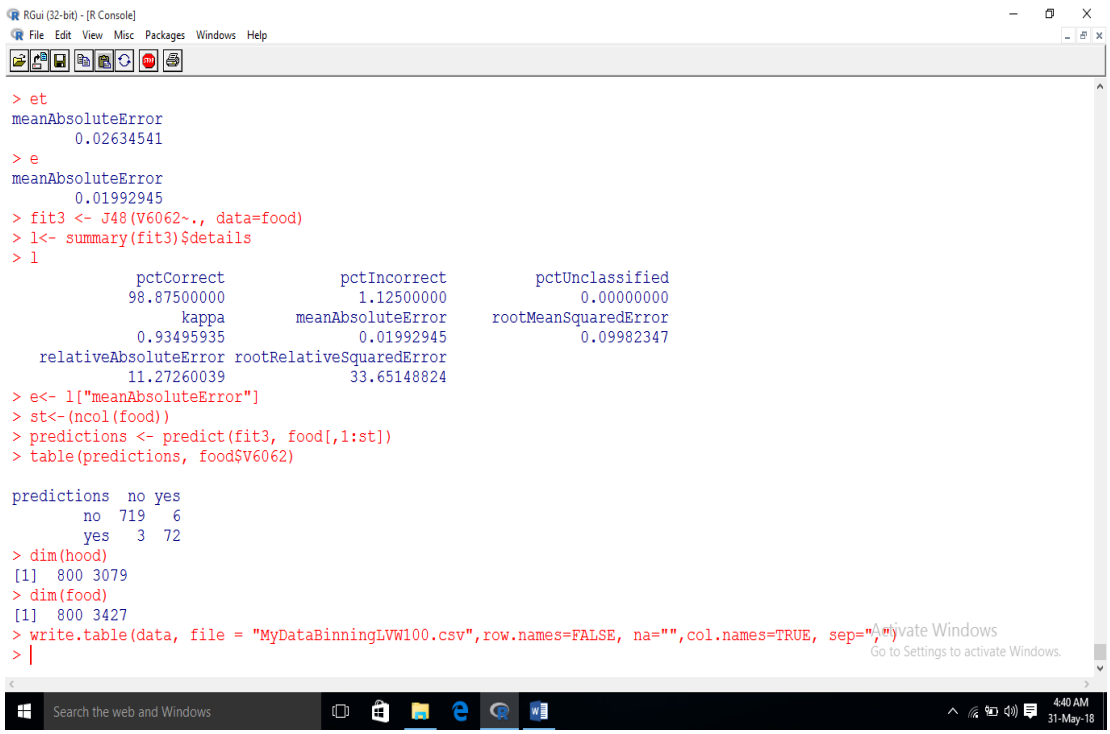

Fig. 14 Las Vegas Wrapper Algorithm when k=100

HYBRID ALGORITHM

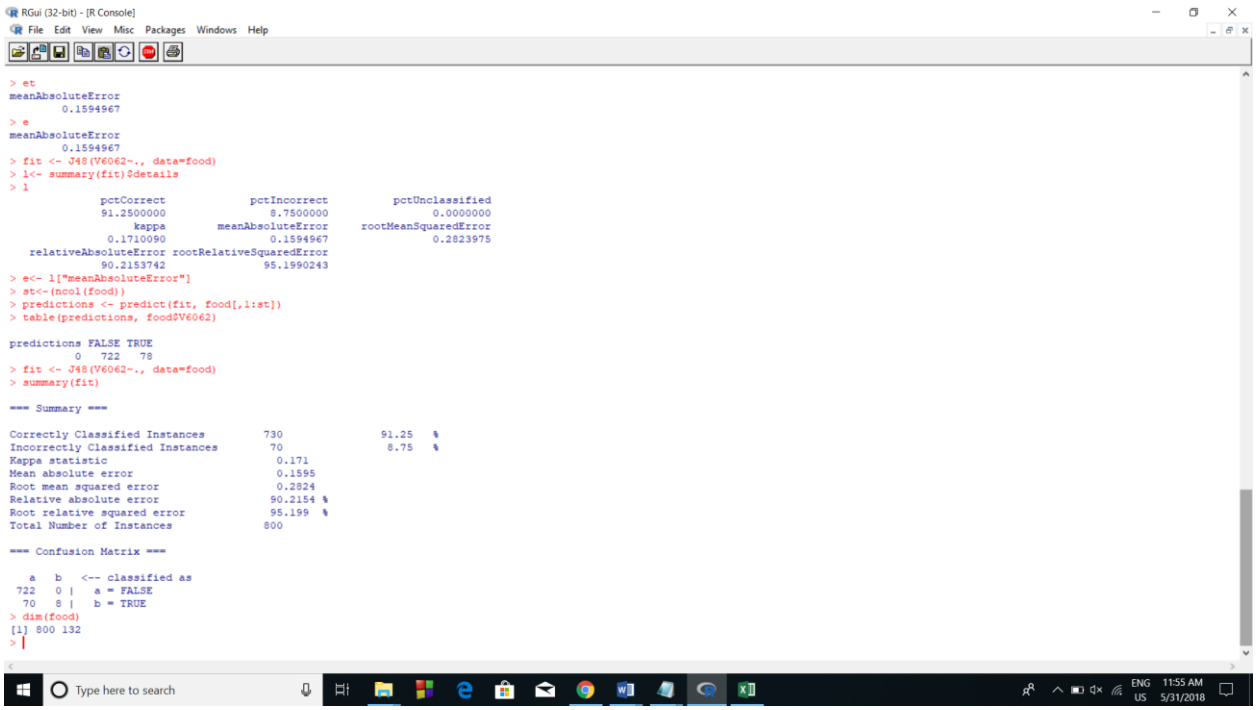

Fig. 15 Hybrid Algorithm when k=10, its =5

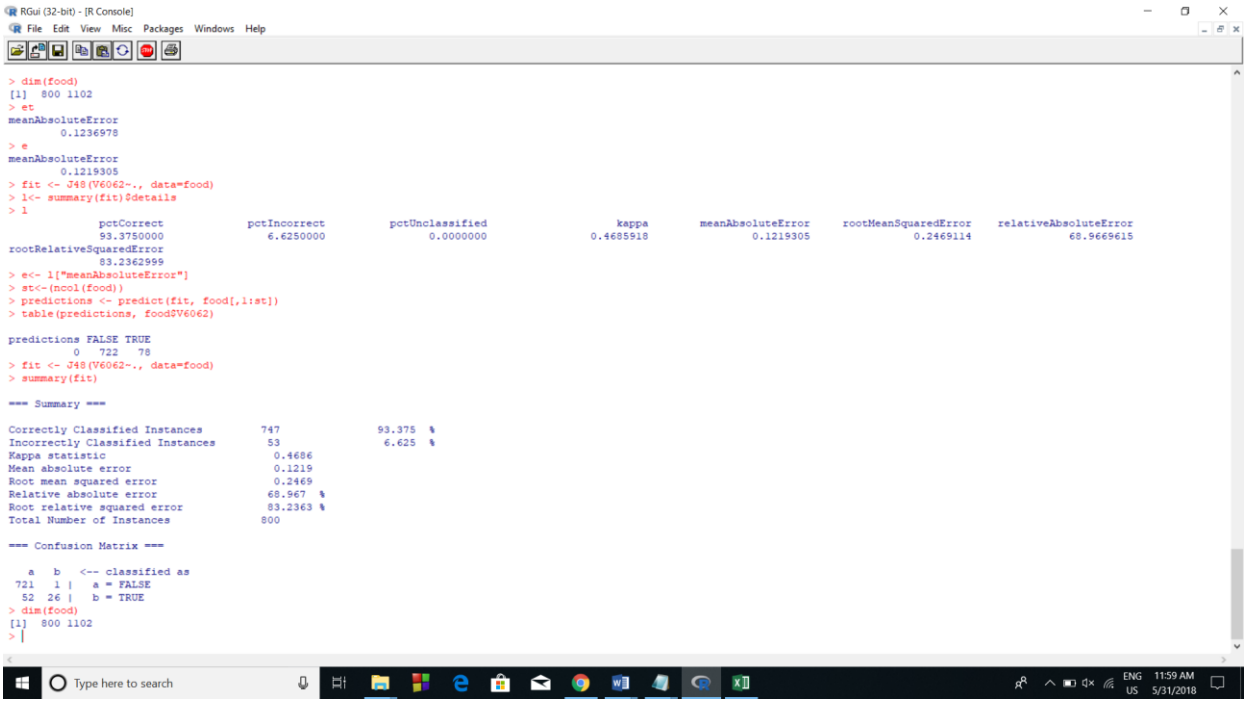

Fig. 16 Hybrid Algorithm when k=10, its =10

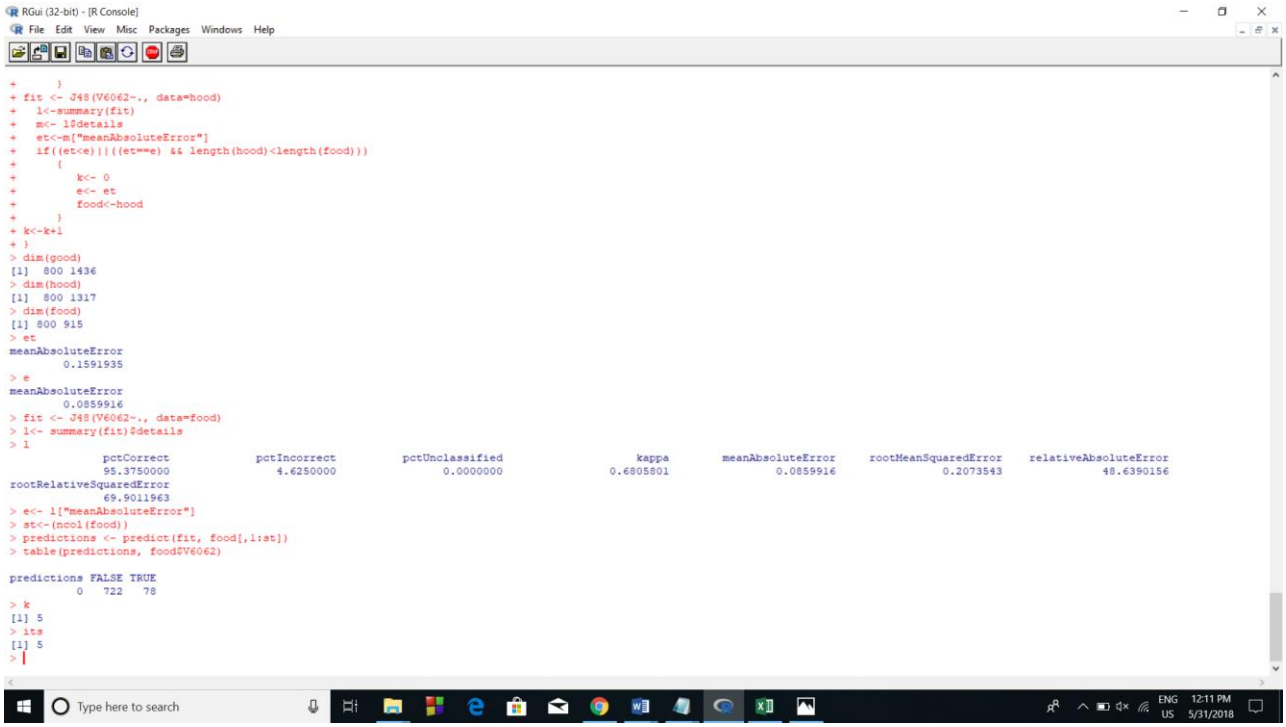

Fig. 17 Hybrid Algorithm when k=10, its =20