

Laxman Dutt Degala

Vishnu Gude

Ayyappa

Akhileshwar

 | KDM – CS560 | March 10, 2014

CAAM for Smart Energy

Increment 1

**Please find below our increments (Increment 1):**

1. How Application works (Deeper dive)
2. Understanding the Data & Decision on fields (Attributes) to be used which impact the Target
3. Decision on tool sets for effective Data Mining
4. Expected Patterns
5. Future Plan
6. **How Application works**

Application is mash-up application where the users are provided with:

1. Energy Consumed and Charges per month, when given the appliances details and Number of people in the family etc.
2. Once the user understands the average electricity charges per month in his community, he/she can set an over-limit value. Based on the daily usage, we would take the existing data (for every monthly cycle) and estimate the electric charges for the month. If the estimate charges crosses the over-limit value that was earlier set, he/she would be warned.
3. **Understanding the data & Decision on attributes to be used which impact the Target.**

We could find huge chunk of data when looked at the thesis presented on CAAM for Smart Energy. In the .csv file, there are 1075 columns in total.

Attached the layout file along with this document.

As we cannot use all the columns for predicting the energy consumed, we will have to come up with approx. 20 attributes which have maximum effect on deciding the *Amount to be Paid per month*, which is the target in our model.

We could manually select a set of 50 attributes which directly/indirectly affect the monthly electricity bill. Now, we are running a set of tools in iteration to figure out the top 20-25 attributes which hugely impact the target (electricity bill, in our case). Information about the tools is provided in the following paragraphs.

1. **Decision on tool sets to be used for efficient data mining:**

As the data is huge, we would need an intelligent and efficient method for data mining. We have started thinking about Mahout, then moved to ‘**R**’, which is a well-developed, simple, effective programming language, with lots of inbuilt tools for data analysis and provides ease with its statistical data facilities.

It’s enhanced with **RStudio** installation, which is a powerful IDE for ‘**R**’.

R supports Hadoop in the form of ‘**RHadoop**’. R is inbuilt in the cloudera image on which we are currently working. We just need to enter ‘R’ in terminal to get the basic information about ‘R’ environment that’s built.

Please find attached the installation guidelines for RHadoop and RStudio. Can be neatly formatted and updated in near future so that it’s useful for others as well.

We then came across **RATTLE**, which is built on top of ‘R’ to build a predictive model.

When it comes to further refining the attributes from the above mentioned 50 attributes, we can use RATTLE’s feature EXPLORE>Box Plot to see how an input attribute impacts the target.

**Install R and Rattle.**

Please use the below stated commands in R environment to install Rattle.

> install.packages("RGtk2")   
> install.packages("rattle", dependencies=TRUE)

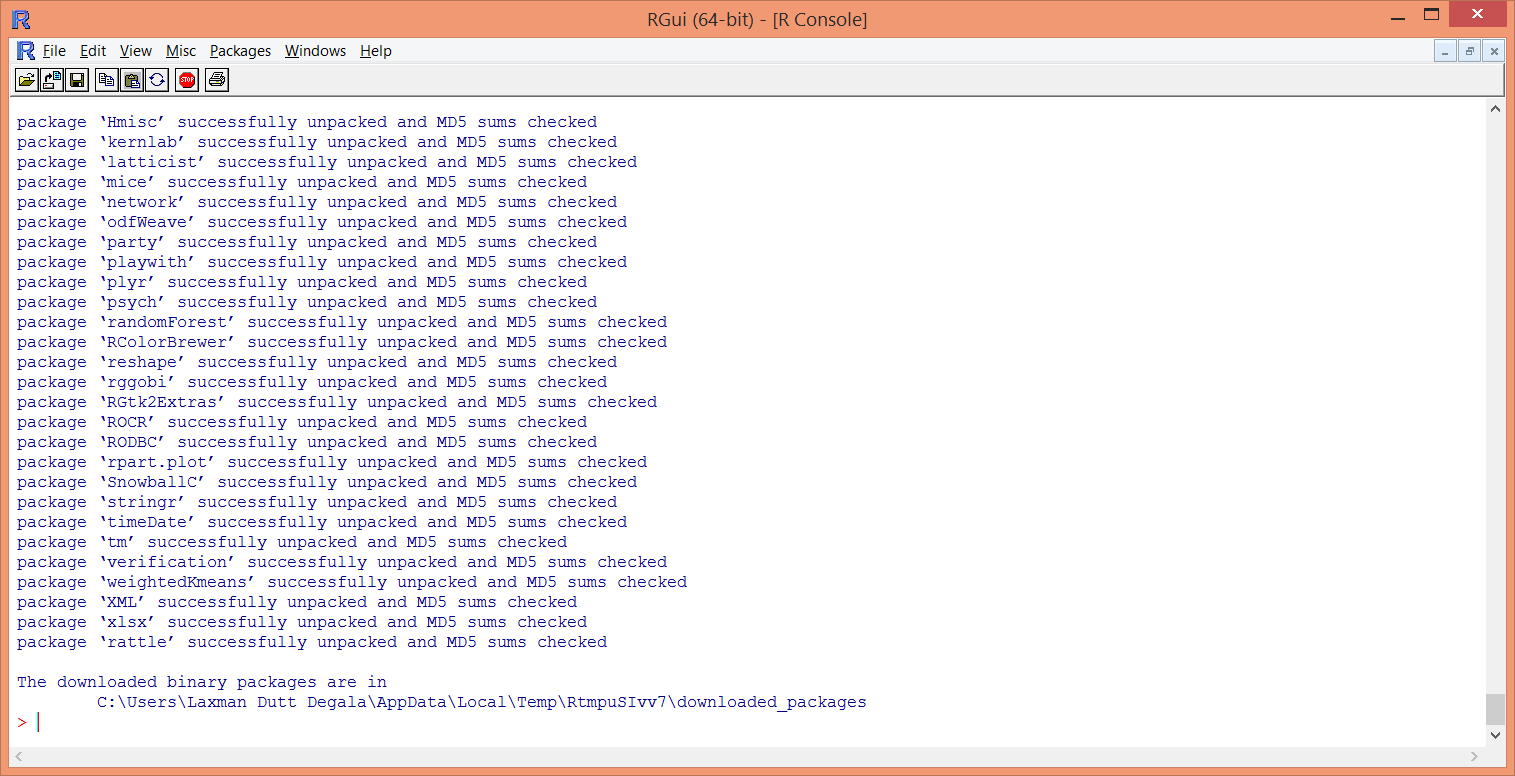
Commands in R to kick-start Rattle:

1. Library(rattle)
2. Rattle() – This will pop up a new GUI

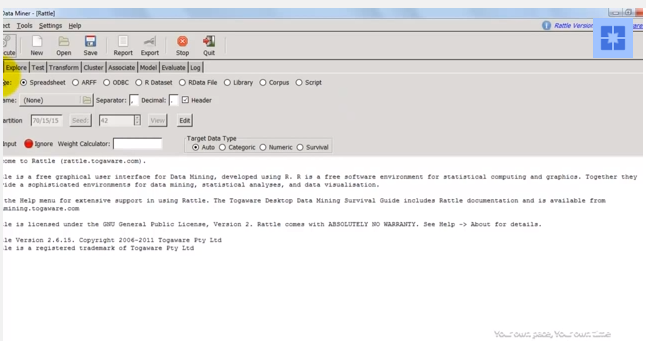
Will create a more clear and updated document on this to share with CS560

group.

R GUI:



Rattle GUI:

****

**4. Expected Patterns**

Below snapshots are taken for 3 different attributes, from a data which has a binary target (Yes or No).

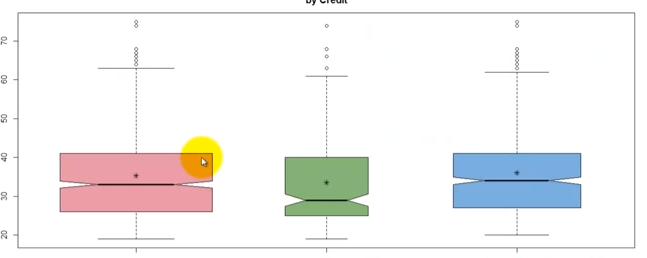
Box Plot image for an input attribute to see the impact.

1st figure shows for all the users.

2nd – Default users (No)

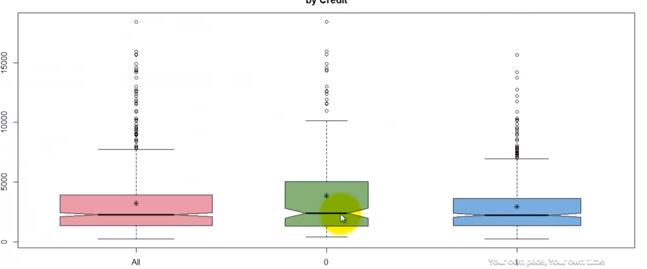
3rd – Non-default users (Yes)

Looking at the difference of Median (Lower box) and Variance (Upper box) for all users/Default/Non-default users we can figure out which attribute has high impact on target.

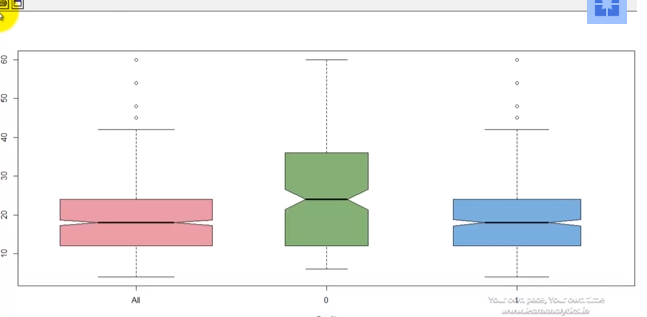


If the difference of Median is very minute or small, we can ignore the

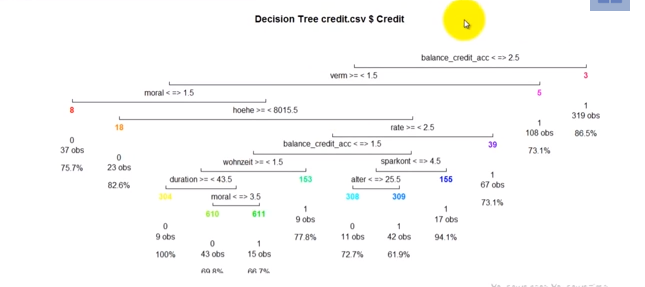
attribute as its impact on the target is very less.



Below stated snapshot shows both median and variance for defaulters are high.



A Decision Tree Model created in Rattle is stated below:



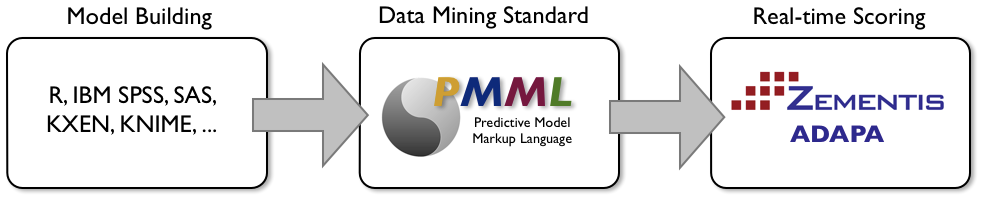
Apart from these features, we can use the same tool to

1. Explore the data
2. Divide the data to Test and Train
3. Modify the data (If required)
4. Cluster the data
5. Create the Model

**5.** **Future Plan:**

The predictive model that’s created using the above stated processes in Rattle, is fed to the Data Mining Standard (PMML) which is then deployed to the real-time scoring engine (ADAPA).

ADAPA executes Predictive models represented in PMML format.



Apart from the data that we have (From Surveys), we could get the real time data (Smart Meters) from **Tracebase.org.** Currently analyzing the data to see if it can be appended to the existing data. If required, we might end up using this real time data.