Technical Specification

[Retail Analytics – Churn Prediction]

Author: [Swarna Gupta, Vipul Gupta]

Date: [Mar 10, 2016]

Version: [Draft]

# Document Control

**Document location**

| Location |
| --- |
| <http://172.25.164.110:9000/OEG-MC/RFM-ANALYSIS> |

**Author**

| Position | Name | Contact no |
| --- | --- | --- |
| SE  SA | Swarna Gupta  Vipul Gupta | 8148606736  9999155296 |

**Revision history**

| Version | Issue date | Author/editor | Description/Summary of changes |
| --- | --- | --- | --- |
| DRAFT | Feb 25, 2016 | Swarna Gupta | Draft version of LTV Module |
|  |  |  |  |

**Reviewed by**

| Version | Issue date | Name | Position | Review date |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Approvals**

Approval refers to the approver’s acceptance of the content and overall intention of this document, including acceptance of any commitments described in order to successfully deliver the initiative. The approver, where relevant, also confirms that this document complies with relevant strategies, policies and regulatory requirements.

| Version | Issue date | Name | Position | Approval date |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

# Table of Contents

Table of Contents

[1 INTRODUCTION 4](#_Toc446348691)

[1.1 Objectives 4](#_Toc446348692)

[1.2 Scope 4](#_Toc446348693)

[2 Functionality 4](#_Toc446348694)

[2.1 Screenshots with explanation of each controls 4](#_Toc446348695)

[3 Architecture 7](#_Toc446348696)

[3.1 Architecture Overview 7](#_Toc446348697)

[3.2 Component Structure 7](#_Toc446348698)

[4 Design 8](#_Toc446348699)

[4.1 Data Dictionary 8](#_Toc446348700)

[5 CODE 8](#_Toc446348701)

# INTRODUCTION

## Objectives

**Churn Prediction: Identify customers which are most likely to churn**

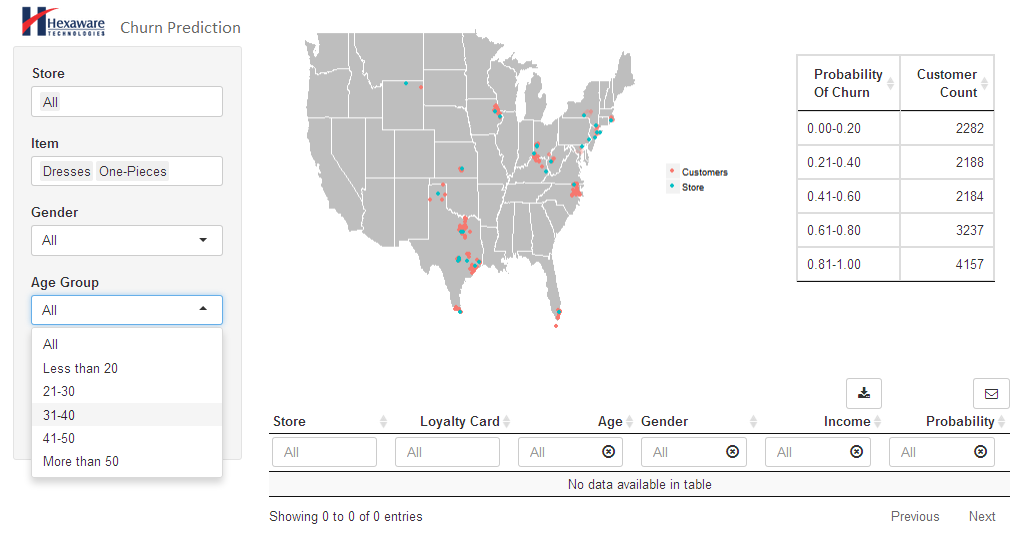
## Scope

It works for transactional level point of sales data of a retail store.

# Functionality

## Screenshots with explanation of each controls

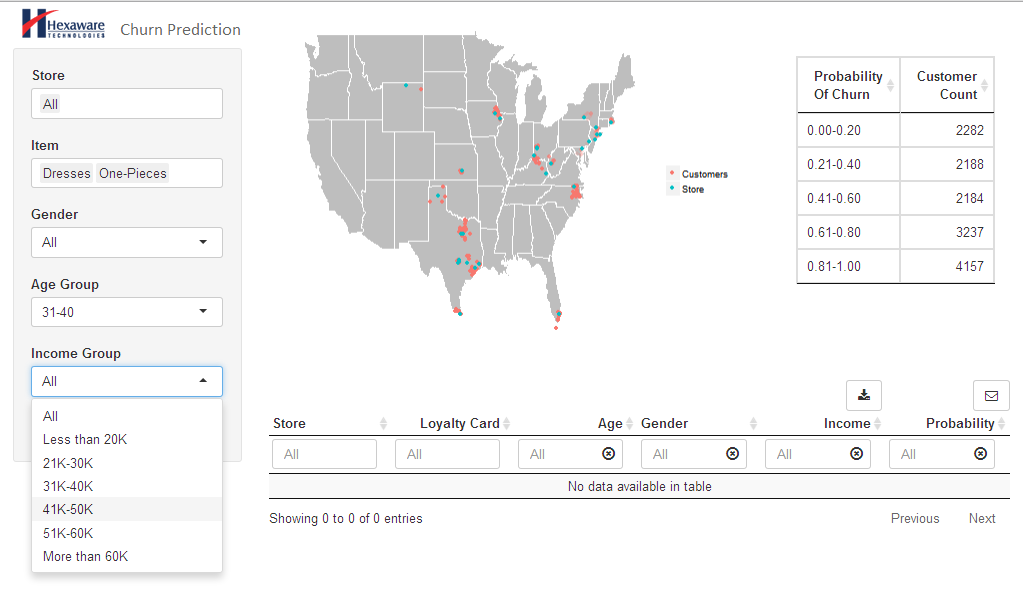
Age Filter



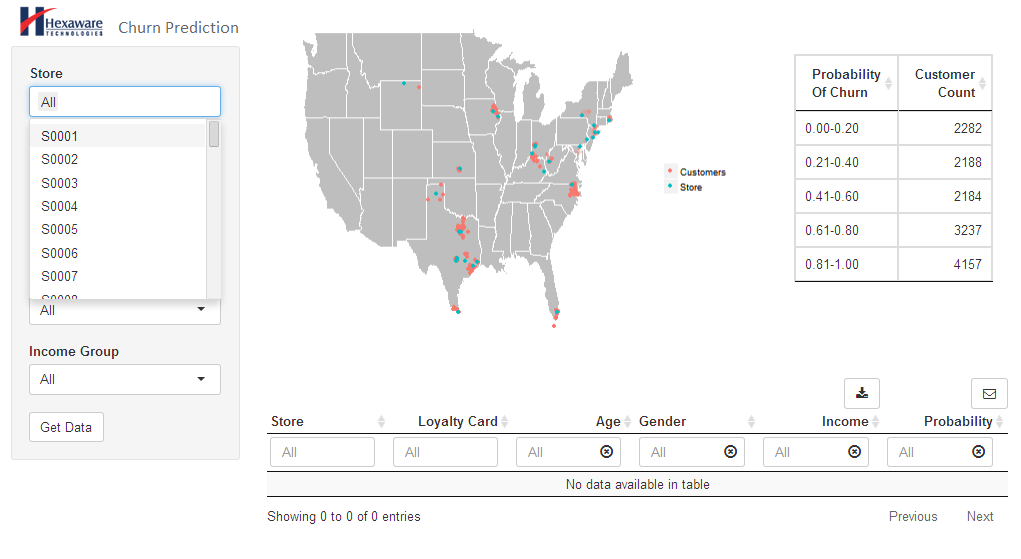
Gender Filter



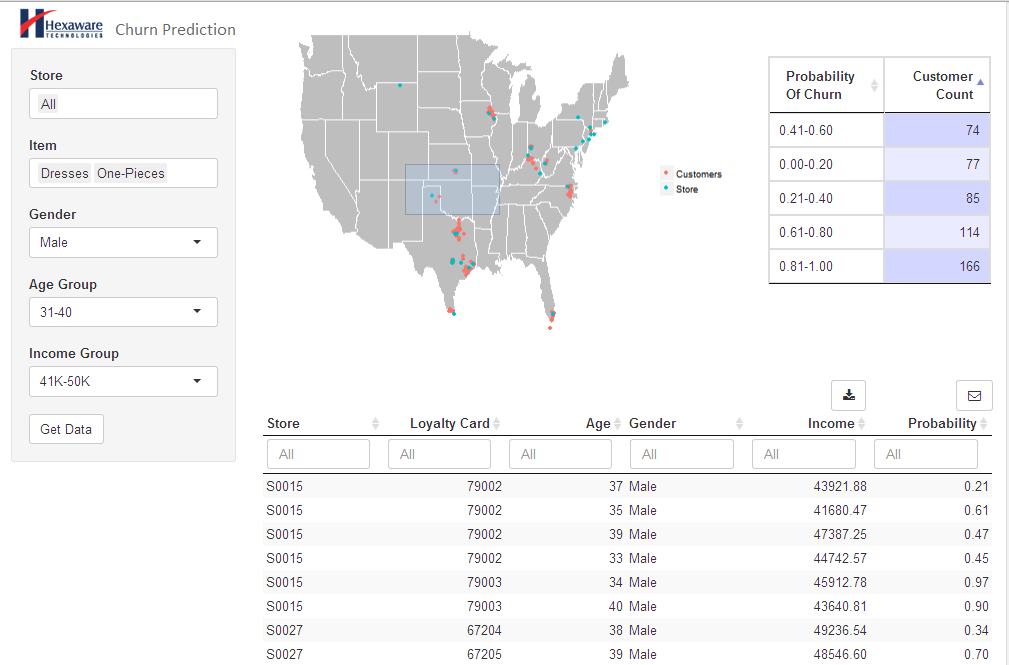
Income Filter



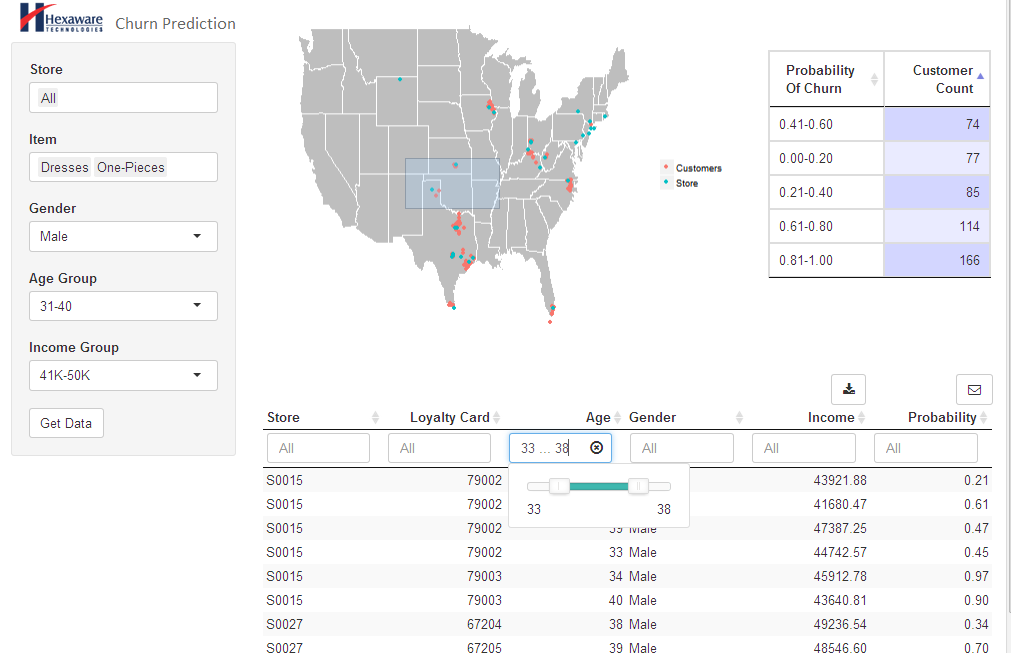
Store Filter



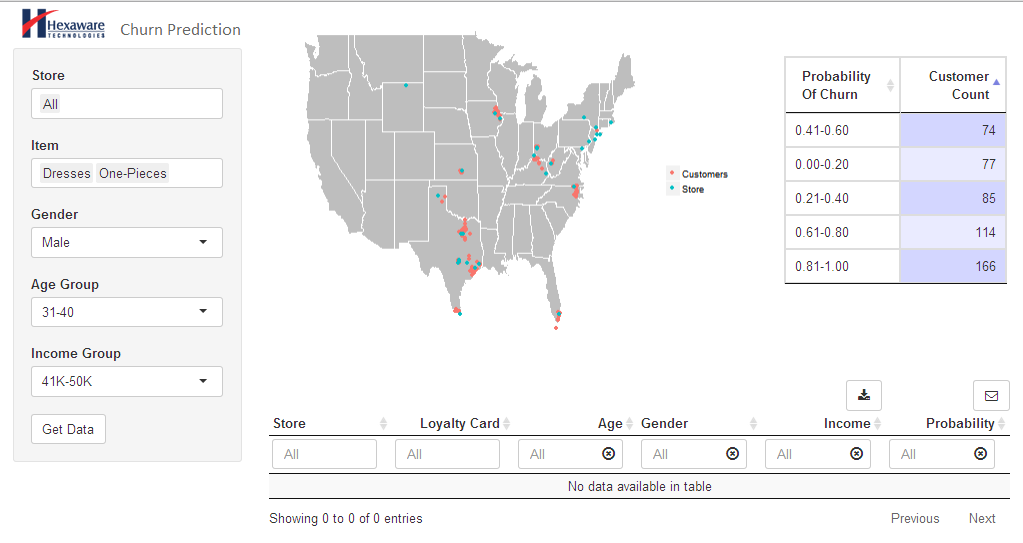
Selection from Map



Further Selection from Map



Churn Probability



# Architecture

## Architecture Overview

Front end – R Shiny

Middle tier – R

Backend – SQL Server 2012

**Front-end / Client Side:**  
The top most visible layer is what is called the Frontend. This is written in R Shiny for styling, interactivity and function. The data from this layer has to make it to the user's browser which then turns all the code into something beautiful and interactive.  
  
**Middle tier:**  
Here we have the "Middle tier" which is where we usually use a programming language R. The middle tier usually decides how to render the front-end depending on it's business logic.

**Back-end:**  
In order for most web application to function, we need to be able to store data somewhere. That is served by a database of some sort. Here we are using SQL Server 2012.

## Component Structure

Every Shiny app is composed of a two parts: a web page that shows the app to the user, and a computer that powers the app. The computer that runs the app can either be your own laptop (such as when you’re running an app from RStudio) or a server somewhere else. You, as the Shiny app developer, need to write these two parts (you’re not going to write a computer, but rather the code that powers the app). In Shiny terminology, they are called UI (user interface) and server.

# Design

## Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Datatype** | **Description of the field** |
| Sequence No | float | Sequence identifier |
| Store No | nvarchar | Store identifier |
| Loyalty Card No | nvarchar | Customer identification number |
| Receipt No | float | Receipt identifier |
| Transaction date | date | Date of the transaction as recorded in the POS system |
| Acquisition date | date | Date of the acquisition of the customer as recorded in the POS system |
| Price | float | Price of the products purchased |
| Item Category code | float | Product Category identification number |
| Item Category description | nvarchar | Product Category description |

# CODE

INPUT PARAMETERS- Item Category, Store, Gender, Age group, Income group

OUTPUT PARAMETER- Dataframe with churn probability for all loyalty card nos

FILTERS

1. Filter the data according to item category
2. Filter the data according to store
3. Filter the data according to gender
4. Filter the data according to income group
5. Filter the data according to Age group

Churn Probability Calculation

1. – Identify the churned customers in training data (who have not purchased in 2 years or more)
2. – Train the Logistic model on this training data set
3. – Test the model for accuracy
4. – Fit the model on remaining/new data

> glm.fit(churn~r+f+m,family=binomial)

> glm.fit=glm(churn~r+f+m,data=logictic,family=binomial)

> pairs(logictic,col=churn)

> plot(churn~r)

> plot(churn~f)

> plot(churn~m)

> plot(r~f)

> plot(r~m)

> plot(f~m)

> plot(f~m,col=”red”)

> summary(logictic)

> glm.probs=predict(glm.fit,type="response")

> glm.probs[1:10]

> glm.pred=ifelse(glm.probs>.5,1,0)

> attach(logictic)

> table(glm.pred,churn)

> mean(glm.pred==churn)

> train=lcn>1000315927

> glm.fit=glm(churn~r+f+m,data=logictic,family=binomial,subset=train)

> glm.probs=predict(glm.fit,newdata=logictic[!train,],type="response")

> glm.pred=ifelse(glm.probs>.5,1,0)

createBins<-function(help.churn.data){  
   
   
 for (i in 1:nrow(help.churn.data)){  
 if(help.churn.data$Age[i]<=20){  
 help.churn.data$AgeBins[i]<-"Less than 20"  
 }  
 else if(help.churn.data$Age[i]>20 & help.churn.data$Age[i]<=30){  
 help.churn.data$AgeBins[i]<-"21-30"  
 }  
 else if(help.churn.data$Age[i]>30 & help.churn.data$Age[i]<=40){  
 help.churn.data$AgeBins[i]<-"31-40"  
 }  
 else if(help.churn.data$Age[i]>40 & help.churn.data$Age[i]<=50){  
 help.churn.data$AgeBins[i]<-"41-50"  
 }  
   
 else {  
 help.churn.data$AgeBins[i]<-"More than 50"  
 }  
   
   
 }   
   
   
   
 for (i in 1:nrow(help.churn.data)){  
 if(help.churn.data$Income[i]<=20000){  
 help.churn.data$IncomeBins[i]<-"Less than 20K"  
 }  
 else if(help.churn.data$Income[i]>20000 & help.churn.data$Income[i]<=30000){  
 help.churn.data$IncomeBins[i]<-"21K-30K"  
 }  
 else if(help.churn.data$Income[i]>30001 & help.churn.data$Income[i]<=40000){  
 help.churn.data$IncomeBins[i]<-"31K-40K"  
 }  
 else if(help.churn.data$Income[i]>40000 & help.churn.data$Income[i]<=50000){  
 help.churn.data$IncomeBins[i]<-"41K-50K"  
 }  
 else if(help.churn.data$Income[i]>50000 & help.churn.data$Income[i]<=60000){  
 help.churn.data$IncomeBins[i]<-"51K-60K"  
 }  
 else {  
 help.churn.data$IncomeBins[i]<-"More than 60K"  
 }  
   
   
 }  
   
   
  
 help.churn.data  
}