# amazon Reviews

Curtis, Joe, Nate, and Ryan

# Overview

- Project Objective
  - Outline
- Methods and Tools
  - AWS
  - R
  - Python
- Build Model
- Descriptive Analysis

# Project Goal

- Summary :

This project will entail web scraping, text mining, and predictive models with the objective of predicting "Review" (y/n) and/or the rating (number of stars). This approach seeks to help sellers target the reviewers most likely to review their product with a high rating, which will also be seen as helpful to other shoppers.

- Key Idea: Want to find text that leads to a good review.

# Team Timeline Outline

**Get Data**
- October 23rd

**Clean and Design**
- October 30th

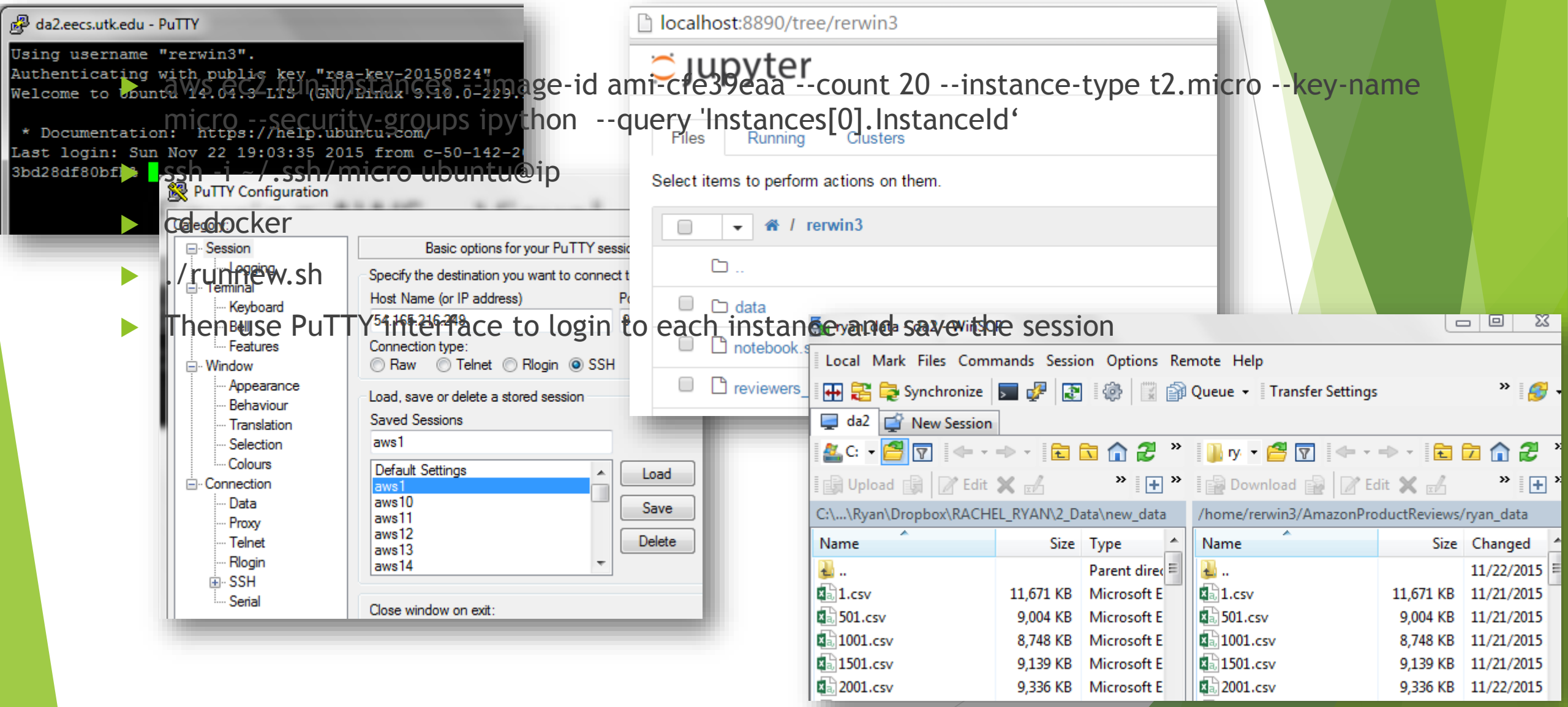
**Build Model**
- November 13th

**Analyze Results and Present**
- November 25th

| Team Member | Responsibilities |
| --- | --- |
| Ryan | • Project Manager tasks<br>• Utilize R to pull in Data<br>• AWS<br>• Support Role |
| Curtis & Nate | • Utilize Python to pull data<br>• Tasks that require python<br>• Support Role |
| Joe | • Model Developer<br>• Interpret Results<br>• Support Role |

# Configuring AWS

- aws ec2 run-instances --image-id ami-cfe39eaa --count 20 --instance-type t2.micro --key-name micro --security-groups ipython  --query 'Instances[0].InstanceId‘
- ssh -i ~/.ssh/micro ubuntu@ip
- cd docker
- ./runnew.sh
- Then use PuTTY interface to login to each instance and save the session



rerwin21/amazon_re | AmazonProductRevi | ItemLookup - Produ | Amazon Web Service | Operations — pytho | Web Scraping: worki

ec2-54-152-30-129.compute-1.amazonaws.com:8787

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function

data_last_step.R | call_last_step.R | Untitled1 *

Source on Save | Run | Source

```r

# load the list of top reviewers ------------------------------------------
reviewers <- read.csv("reviewers_list.csv", stringsAsFactors = F)


# what page to scrape to for reviews
reviewers <- reviewers %>%
  mutate(
    page_num = ceiling(Reviews/10),
    page_num = ifelse(page_num <= 10, page_num, 10)
  )


# grab the url for each person I'm assigned
reviewers <- reviewers %>%
  filter(page_num > 0)


# create the review links
review_links <- .create_links(reviewers$url_name,
                              reviewers$page_num)


# call and store the results ----------------------------------------------
setwd("~/data")

```

1:1 | where are my functions | R Script

Console ~/data/

```
times[j] <- end_loop
```

Environment | History

Import Dataset

Global Environment

| Data | |
|---|---|
| review_links | 487280 obs. |
| reviewers | 9986 obs. |
| **Values** | |
| end_loop | |
| file_name | "2001.csv" |

Files | Plots | Packages | Help | Viewer

New Folder | Upload | Delete

Home

Name

- .Rprofile
- call_last_step.R
- data
- data_last_step.R
- notebook.sh
- products.csv
- R
- reviewers_list.csv
- start.sh

# Python Extraction

- What to Extract
  - Reviewer ID
  - Item Price
  - URL
  - Item Description
  - Rating
  - Summary of Review
  - Item Category

# Troubleshooting

- Separate Fields
- Comments
- VPN
- Run time

## ≈ 300,000 Reviews

- Code
  1. Class
  2. Driver
  3. Function

# Snip it- Driver

```
#Driver
with open('/home/jhughe39/AmazonProductReviews/python_implementation/reviews4.csv', 'w', newline='
') as csvfile:

    # header for csv file
    fn = ['reviewerName', 'reviewerUrl', 'itemNo', 'itemTitle', 'itemUrl', 'itemBrand', 'itemPrice
',
          'itemCategory', 'dateReviewed', 'Rating', 'Summary', 'Description']

    # open csv file for writing
    reviewWriter = csv.DictWriter(csvfile, delimiter=',', fieldnames=fn)
    reviewWriter.writeheader()

    f = open("/home/jhughe39/AmazonProductReviews/python_implementation/ReviewURLs.txt","r")

    for url in f:
        review = get_review(url)
        time.sleep(3)
        if review.itemNo != 'none' and review.reviewerUrl != 'none':
            reviewWriter.writerow(review.__dict__)
```

Scrape Data

Find Most Common Words (Cluster Analysis)

Build Model

- What words lead to a positive review

Random Forest Model
Response: 5 Star Rating

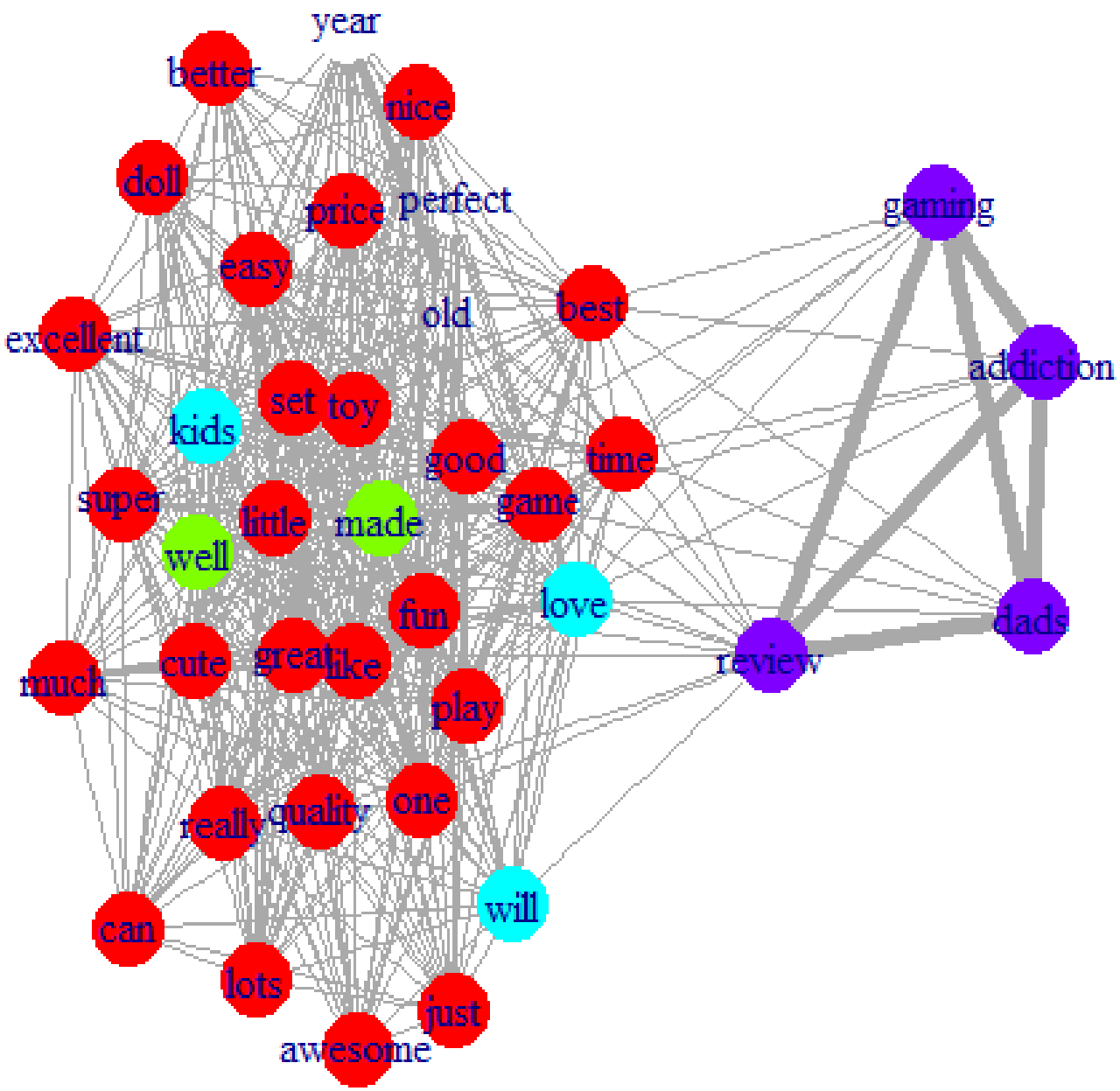
# Top Item Categories

- Books
- Kindle Store
- Toys and Games
- Health and Beauty
- Electronics
- Amazon Video

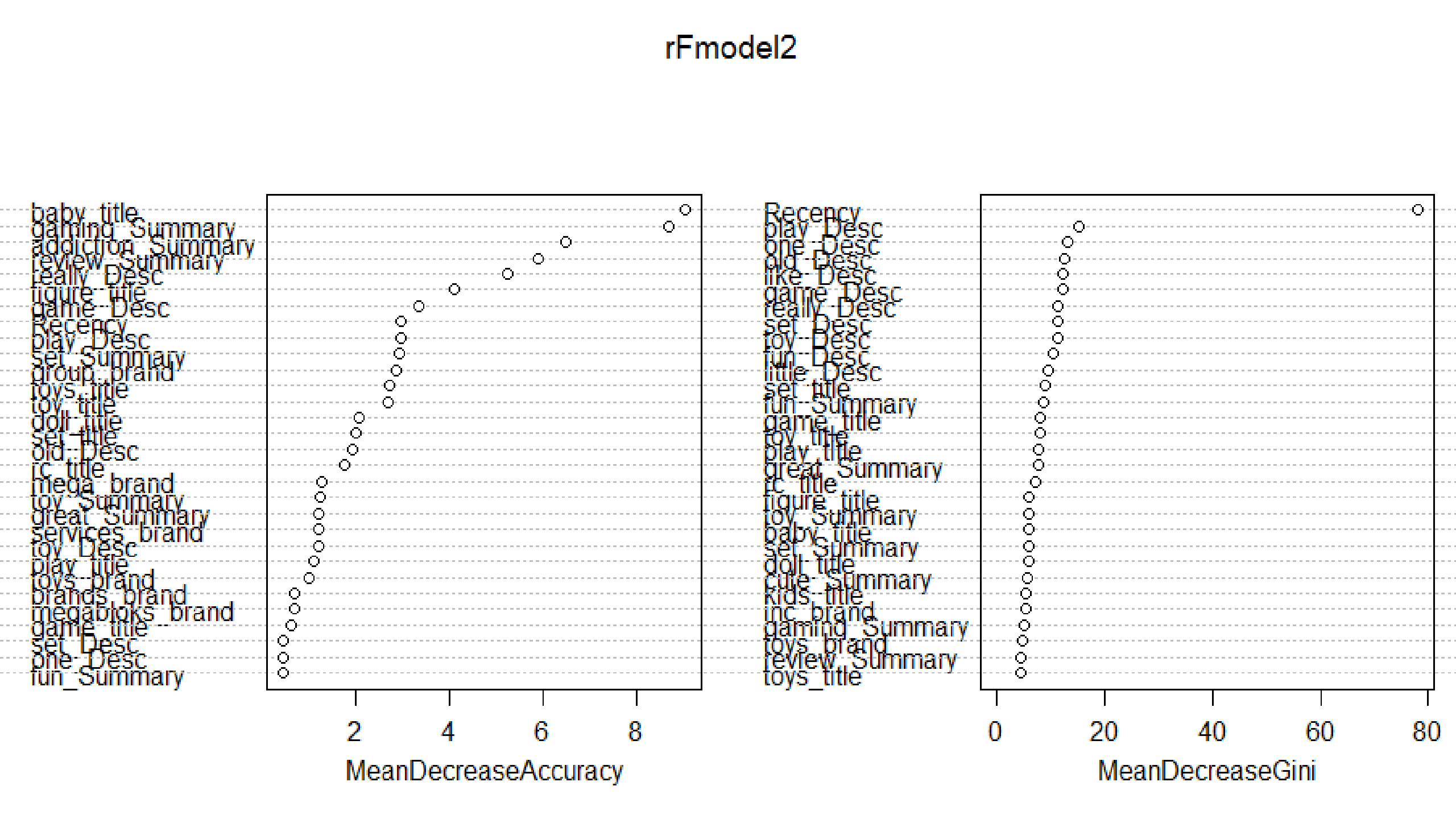
# Example - Toys and Games

- Text analysis
  - Review Summary
  - Item Description
  - Brand Names
  - Product Names

  - Response – 5 Star Rating


year
better
nice
doll
price
perfect
easy
old
best
excellent
set
toy
kids
good
time
super
little
made
game
well
fun
love
much
cute
great
like
play
really
quality
one
will
can
lots
just
awesome
gaming
addiction
dads
review

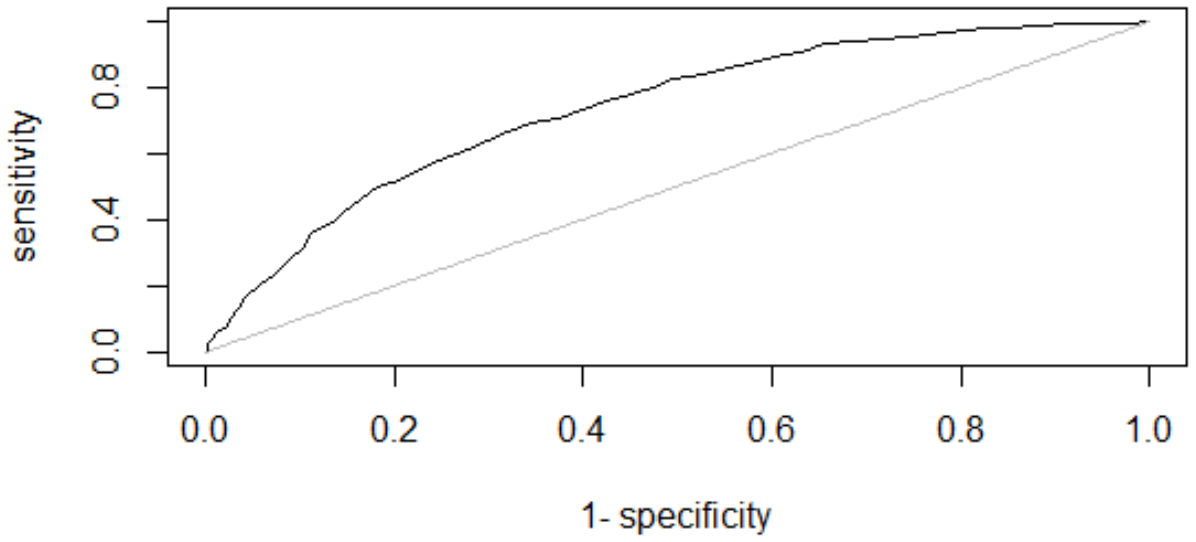

awesome
dads
cute
best
play
kids
really
doll
quality
old
made
just
lots
much
price
time
easy
well
excellent
good
year
will
love
nice
can
little
perfect
toy
great
one
better
super
like
game
review
gaming
addiction

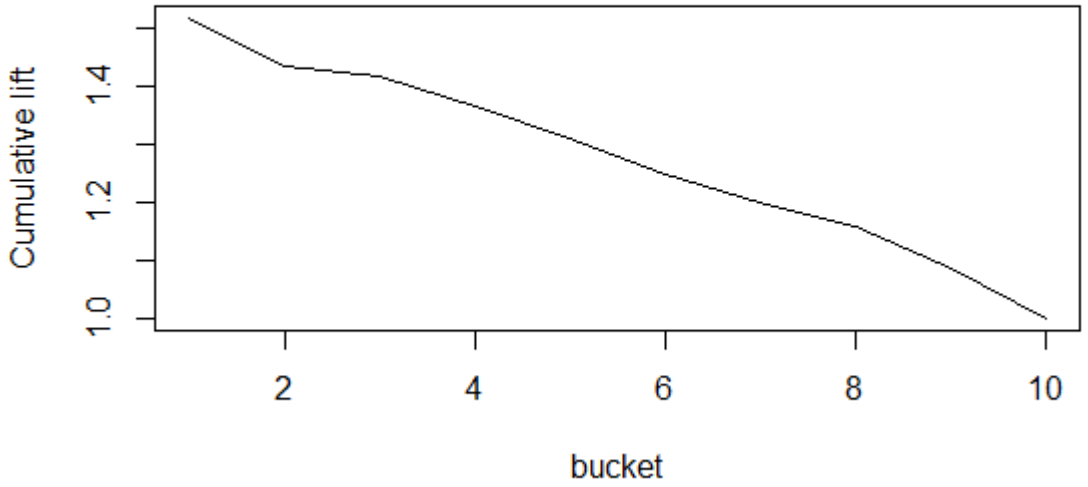

rFmodel2
baby_title
gaming_Summary
addiction_Summary
review_Summary
really_Desc
figure_title
game_Desc
Recency
play_Desc
set_Summary
group_brand
toys_title
toy_title
doll_title
set_title
old_Desc
rc_title
mega_brand
toy_Summary
great_Summary
services_brand
toy_Desc
play_title
toys_brand
brands_brand
megabloks_brand
game_title
set_Desc
one_Desc
fun_Summary
2
4
6
8
MeanDecreaseAccuracy
Recency
play_Desc
one_Desc
old_Desc
like_Desc
game_Desc
really_Desc
set_Desc
toy_Desc
fun_Desc
little_Desc
set_title
fun_Summary
game_title
toy_title
play_title
great_Summary
rc_title
figure_title
toy_Summary
baby_title
set_Summary
doll_title
cute_Summary
kids_title
inc_brand
gaming_Summary
toys_brand
review_Summary
toys_title
0
20
40
60
80
MeanDecreaseGini

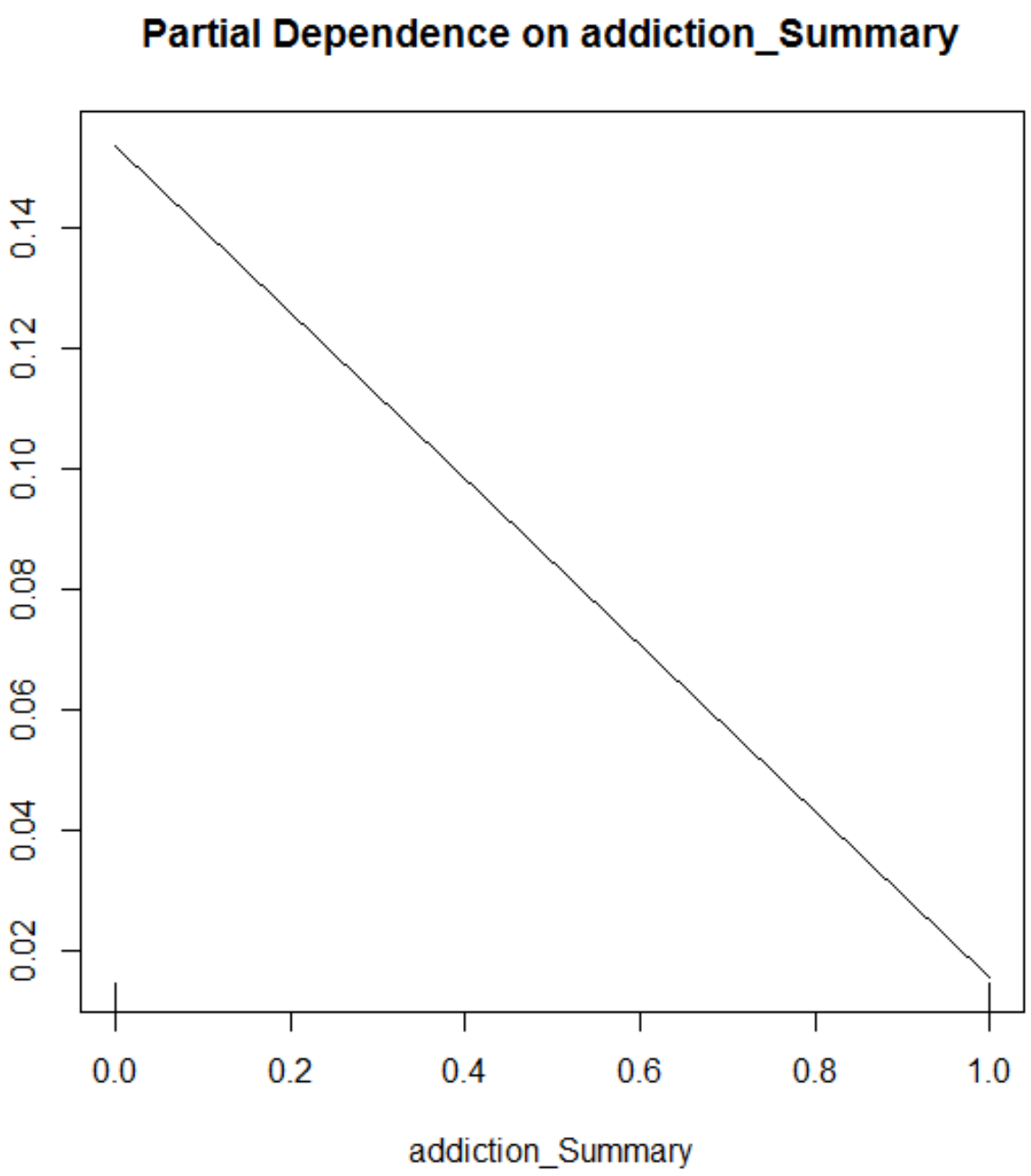
# Model Performance

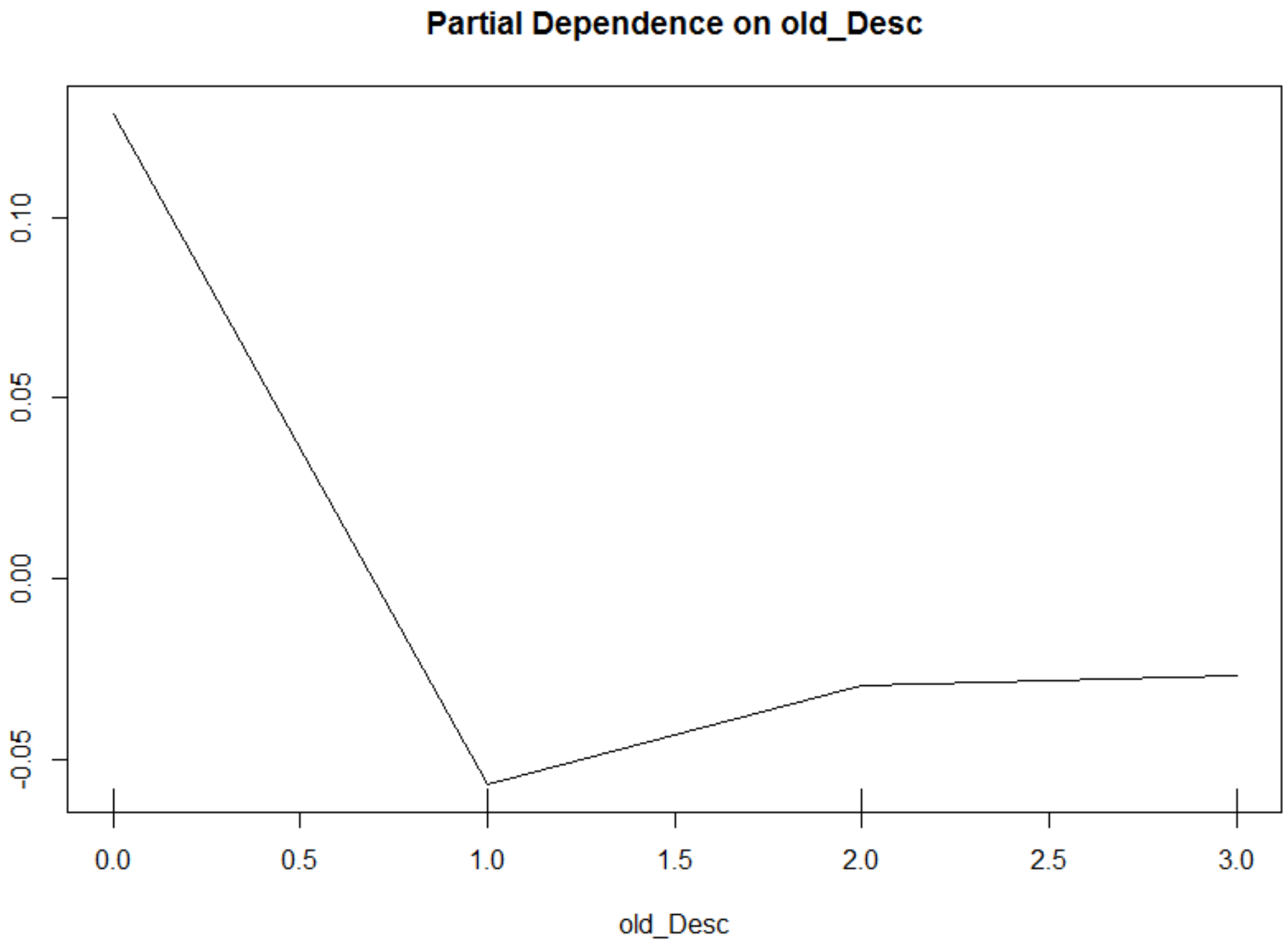
AUC - 0.75 - 0.80

Lift Curve - 1.5




Partial Dependence on addiction_Summary
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.0
0.2
0.4
0.6
0.8
1.0
addiction_Summary


Partial Dependence on old_Desc
0.10
0.05
0.00
-0.05
0.0
0.5
1.0
1.5
2.0
2.5
3.0
old_Desc

# End

Any questions?