**Introduction**

Leaving money on the table is a problem most companies try to avoid. It’s an issue that presents itself to business in many forms. As a business gains more customers it can become increasingly difficult to determine which of them are likely to order more products, and why. Our focus will be on maximizing revenue from current customers who are likely to make additional purchases.

My client on this project is StandDesk, Inc. They manufacture the most cost effective automatic height adjustable desks on the market ([Receive a $50 discount by ordering here!](http://go.standdesk.co/dT8sl)). After raising over $800,000 on Kickstarter in 2014, they quickly began taking orders from businesses who were trying to create a healthier workspace for their employees. For a lean startup like StandDesk, it can become increasingly difficult to maximize revenue from a rapidly expanding business to business customer base without utilizing a laser like focus of marketing and sales resources on the right customers at the right times. We will attempt to determine which customers to focus on by creating a model that predicts the total revenue that can be generated from each customer.

**Data Set**

The StandDesk dataset consists of 3 csv files:

* **Sales by customer 6.24.csv (Shopify)**
* **hubspot-crm-view-companies2016-06-24.csv (Hubspot)**
* **hubspot-crm-view-contacts2016-06-24.csv (Hubspot)**

The data was downloaded from StandDesk’s CRM and Shopify account. The CRM contains information related to interactions between StandDesk and their customers as well as their customers’ demographic information. Data downloaded from Shopify contains information related to purchases from StandDesk’s customers.

The data from Shopify was downloaded as a csv file and contains information on each purchase a customer has made. The Order report downloaded from Shopify includes Purchase Date, Purchase Amount, Contact’s Email, etc. This file was imported into R as a data frame called “or\_df.”

The data from the CRM was downloaded as two separate csv files. The first csv file was downloaded from the Companies tab and contains data on all of the companies which have purchased from StandDesk. This data mostly consists of information about each company such as Phone Number, Website, Address, etc., along with demographic data including Employee Count, Industry, Year Founded, etc. This file was imported into R as a data frame called “c\_df.”

The second csv file was downloaded from the Contacts tab which contains data on the contacts related to the companies who have purchased from StandDesk. This data contains information about the contacts such as Phone Number, Email Address, Title, etc. Activity data related to how the contact has interacted with StandDesk is also included in this file. The activity data includes Email communications, calls, website visits, form submissions, etc. This file was imported into R as a data frame called “contact\_df.”

**Data Wrangling**

Each csv file was uploaded to R:

Order\_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/Sales by customer 6.24.csv", header=TRUE, sep=",", na.strings = "")

or\_df <- data.frame(Order\_Report)

Company\_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-companies2016-06-24.csv", header=TRUE, sep=",", na.strings = "")

c\_df <- data.frame(Company\_Report)

Contact\_Report <- read.csv(file = "C:/Users/GamingFoSho/Documents/wdR/hubspot-crm-view-contacts2016-06-24.csv", header=TRUE, sep=",", na.strings = "")

contact\_df <- data.frame(Contact\_Report)

It was necessary to combine the data sets so that each row represents a different company that has made a purchase from StandDesk. Unnecessary and inaccurate variables were eliminated. The company data frame already contained a field labelled “Domain.” A field labeled “Domain” was created in or\_df and contact\_df by separating the domain out from the email fields:

or\_df <- separate(or\_df, email, c("Email Prefix", "Domain"), sep = "@")

or\_df <- filter(or\_df, total\_sales > 0)

or\_df$Domain <- tolower(or\_df$Domain)

contact\_df <- separate(contact\_df, Email, c("Email Prefix", "Domain"), sep = "@")

contact\_df$Domain <- tolower(contact\_df$Domain)

The order dates of each order in the or\_df data frame were aggregated based on the domain field.

or\_day\_vec <- as.Date(or\_df$day, "%m/%d/%Y")

or\_df <- mutate(or\_df, day = or\_day\_vec)

or\_df <- mutate(or\_df, Orders = 1)

attach(or\_df)

or\_df <- or\_df[order(or\_df$Domain, or\_df$day),]

or\_do\_df <- select(or\_df, Domain, Orders)

detach(or\_df)

or\_do\_df <- ddply(or\_do\_df, .(Domain), mutate, Order\_Number = cumsum(Orders))

or\_df <- mutate(or\_df, Order\_Number = or\_do\_df$Order\_Number)

or\_df$Orders <- NULL

or\_df <- filter(or\_df, Order\_Number <= 14)

order\_date\_df <- select(or\_df, Domain, day, Order\_Number)

Order\_Date\_vec <- order\_date\_df$Order\_Number

Order\_Date\_vec1 <- ifelse((Order\_Date\_vec == 1), c("Order\_One\_Date"), c("Unknown"))

Order\_Date\_vec2 <- ifelse((Order\_Date\_vec == 2), c("Order\_Two\_Date"), Order\_Date\_vec1)

Order\_Date\_vec3 <- ifelse((Order\_Date\_vec == 3), c("Order\_Three\_Date"), Order\_Date\_vec2)

Order\_Date\_vec4 <- ifelse((Order\_Date\_vec == 4), c("Order\_Four\_Date"), Order\_Date\_vec3)

Order\_Date\_vec5 <- ifelse((Order\_Date\_vec == 5), c("Order\_Five\_Date"), Order\_Date\_vec4)

Order\_Date\_vec6 <- ifelse((Order\_Date\_vec == 6), c("Order\_Six\_Date"), Order\_Date\_vec5)

Order\_Date\_vec7 <- ifelse((Order\_Date\_vec == 7), c("Order\_Seven\_Date"), Order\_Date\_vec6)

Order\_Date\_vec8 <- ifelse((Order\_Date\_vec == 8), c("Order\_Eight\_Date"), Order\_Date\_vec7)

Order\_Date\_vec9 <- ifelse((Order\_Date\_vec == 9), c("Order\_Nine\_Date"), Order\_Date\_vec8)

Order\_Date\_vec10 <- ifelse((Order\_Date\_vec == 10), c("Order\_Ten\_Date"), Order\_Date\_vec9)

Order\_Date\_vec11 <- ifelse((Order\_Date\_vec == 11), c("Order\_Eleven\_Date"), Order\_Date\_vec10)

Order\_Date\_vec12 <- ifelse((Order\_Date\_vec == 12), c("Order\_Twelve\_Date"), Order\_Date\_vec11)

Order\_Date\_vec13 <- ifelse((Order\_Date\_vec == 13), c("Order\_Thirteen\_Date"), Order\_Date\_vec12)

Order\_Date\_vec14 <- ifelse((Order\_Date\_vec == 14), c("Order\_Fourteen\_Date"), Order\_Date\_vec13)

order\_date\_df <- mutate(order\_date\_df, Order\_Number = Order\_Date\_vec14)

order\_date\_df <- spread(order\_date\_df, Order\_Number, day)

order\_date\_df <- order\_date\_df[c("Domain", "Order\_One\_Date", "Order\_Two\_Date", "Order\_Three\_Date", "Order\_Four\_Date", "Order\_Five\_Date", "Order\_Six\_Date", "Order\_Seven\_Date", "Order\_Eight\_Date", "Order\_Nine\_Date", "Order\_Ten\_Date", "Order\_Eleven\_Date", "Order\_Twelve\_Date", "Order\_Thirteen\_Date", "Order\_Fourteen\_Date")]

The order amounts for each order in the or\_df data frame were aggregated based on the domain field.

or\_num\_df1 <- or\_df

or\_num\_df1$company <- NULL

or\_num\_df1$shipping\_city <- NULL

or\_num\_df1$shipping\_province <- NULL

or\_num\_df1$day <- NULL

or\_num\_df1$year <- NULL

or\_num\_df1$email <- NULL

or\_num\_df1$traffic\_source <- NULL

or\_num\_df1$host <- NULL

or\_num\_df1$referrer <- NULL

or\_num\_df1$name <- NULL

or\_num\_df1$`Email Prefix` <- NULL

or\_num\_df1$order\_count <- NULL

Ord\_Num\_vec <- or\_num\_df1$Order\_Number

Ord\_Num\_vec1 <- ifelse((Ord\_Num\_vec == 1), c("Order\_One\_Amount"), c("Unknown"))

Ord\_Num\_vec2 <- ifelse((Ord\_Num\_vec == 2), c("Order\_Two\_Amount"), Ord\_Num\_vec1)

Ord\_Num\_vec3 <- ifelse((Ord\_Num\_vec == 3), c("Order\_Three\_Amount"), Ord\_Num\_vec2)

Ord\_Num\_vec4 <- ifelse((Ord\_Num\_vec == 4), c("Order\_Four\_Amount"), Ord\_Num\_vec3)

Ord\_Num\_vec5 <- ifelse((Ord\_Num\_vec == 5), c("Order\_Five\_Amount"), Ord\_Num\_vec4)

Ord\_Num\_vec6 <- ifelse((Ord\_Num\_vec == 6), c("Order\_Six\_Amount"), Ord\_Num\_vec5)

Ord\_Num\_vec7 <- ifelse((Ord\_Num\_vec == 7), c("Order\_Seven\_Amount"), Ord\_Num\_vec6)

Ord\_Num\_vec8 <- ifelse((Ord\_Num\_vec == 8), c("Order\_Eight\_Amount"), Ord\_Num\_vec7)

Ord\_Num\_vec9 <- ifelse((Ord\_Num\_vec == 9), c("Order\_Nine\_Amount"), Ord\_Num\_vec8)

Ord\_Num\_vec10 <- ifelse((Ord\_Num\_vec == 10), c("Order\_Ten\_Amount"), Ord\_Num\_vec9)

Ord\_Num\_vec11 <- ifelse((Ord\_Num\_vec == 11), c("Order\_Eleven\_Amount"), Ord\_Num\_vec10)

Ord\_Num\_vec12 <- ifelse((Ord\_Num\_vec == 12), c("Order\_Twelve\_Amount"), Ord\_Num\_vec11)

Ord\_Num\_vec13 <- ifelse((Ord\_Num\_vec == 13), c("Order\_Thirteen\_Amount"), Ord\_Num\_vec12)

Ord\_Num\_vec14 <- ifelse((Ord\_Num\_vec == 14), c("Order\_Fourteen\_Amount"), Ord\_Num\_vec13)

or\_num\_df1 <- mutate(or\_num\_df1, Order\_Number = Ord\_Num\_vec14)

or\_num\_df1 <- spread(or\_num\_df1, Order\_Number, total\_sales, fill = 0)

or\_num\_df1 <- mutate(or\_num\_df1, total\_revenue = Order\_One\_Amount + Order\_Two\_Amount +

Order\_Three\_Amount + Order\_Four\_Amount + Order\_Five\_Amount + Order\_Six\_Amount +

Order\_Seven\_Amount + Order\_Eight\_Amount + Order\_Nine\_Amount + Order\_Ten\_Amount+

Order\_Eleven\_Amount + Order\_Twelve\_Amount + Order\_Thirteen\_Amount +

Order\_Fourteen\_Amount)

company\_df is created by joining c\_df with or\_num\_df1 and then order\_date\_df

c\_df <- mutate(c\_df, Domain = Company.Domain.Name)

c\_df$Company.Domain.Name <- NULL

c\_df$Domain <- tolower(c\_df$Domain)

or\_num\_df1$Domain <- tolower(or\_num\_df1$Domain)

company\_df <- inner\_join(or\_num\_df1, c\_df, by = "Domain")

company\_df <- inner\_join(company\_df, order\_date\_df, by = "Domain")

Several variables (Emails Delivered, Emails Opened, and Emails Clicked) for each contact in the contact\_df data frame were aggregated based on the domain field and then joined to company\_df:

# Prepping contact\_df in order to aggregate several variables

contact\_df <- mutate(contact\_df, Count = 1)

contact\_df <- ddply(contact\_df, .(Domain), mutate, Domain\_Count = cumsum(Count))

contact\_df <- filter(contact\_df, Domain\_Count <= 12)

contact\_df <- filter(contact\_df, Domain != "standdesk.co")

Contact\_Num\_vec <- contact\_df$Domain\_Count

Contact\_Num\_vec1 <- ifelse((Contact\_Num\_vec == 1), c("Contact\_One"), c("Unknown"))

Contact\_Num\_vec2 <- ifelse((Contact\_Num\_vec == 2), c("Contact\_Two"), Contact\_Num\_vec1)

Contact\_Num\_vec3 <- ifelse((Contact\_Num\_vec == 3), c("Contact\_Three"), Contact\_Num\_vec2)

Contact\_Num\_vec4 <- ifelse((Contact\_Num\_vec == 4), c("Contact\_Four"), Contact\_Num\_vec3)

Contact\_Num\_vec5 <- ifelse((Contact\_Num\_vec == 5), c("Contact\_Five"), Contact\_Num\_vec4)

Contact\_Num\_vec6 <- ifelse((Contact\_Num\_vec == 6), c("Contact\_Six"), Contact\_Num\_vec5)

Contact\_Num\_vec7 <- ifelse((Contact\_Num\_vec == 7), c("Contact\_Seven"), Contact\_Num\_vec6)

Contact\_Num\_vec8 <- ifelse((Contact\_Num\_vec == 8), c("Contact\_Eight"), Contact\_Num\_vec7)

Contact\_Num\_vec9 <- ifelse((Contact\_Num\_vec == 9), c("Contact\_Nine"), Contact\_Num\_vec8)

Contact\_Num\_vec10 <- ifelse((Contact\_Num\_vec == 10), c("Contact\_Ten"), Contact\_Num\_vec9)

Contact\_Num\_vec11 <- ifelse((Contact\_Num\_vec == 11), c("Contact\_Eleven"), Contact\_Num\_vec10)

Contact\_Num\_vec12 <- ifelse((Contact\_Num\_vec == 12), c("Contact\_Twelve"), Contact\_Num\_vec11)

contact\_df <- mutate(contact\_df, Contact\_Count = Contact\_Num\_vec12)

# Creating Total\_Emails\_Delivered variable in company\_df

emails\_del\_df <- select(contact\_df, Domain, Emails.Delivered, Contact\_Count)

emails\_del\_df <- spread(emails\_del\_df, Contact\_Count, Emails.Delivered, fill = 0)

emails\_del\_df <- mutate(emails\_del\_df, Total\_Emails\_Delivered = Contact\_One + Contact\_Two + Contact\_Three + Contact\_Four + Contact\_Five + Contact\_Six + Contact\_Seven + Contact\_Eight + Contact\_Nine + Contact\_Ten)

emails\_del\_df <- select(emails\_del\_df, Domain, Total\_Emails\_Delivered)

company\_df <- inner\_join(company\_df, emails\_del\_df, by = "Domain")

company\_df$Emails.Delivered <- NULL

# Creating Total\_Emails\_Opened variable in company\_df

emails\_open\_df <- select(contact\_df, Domain, Emails.Opened, Contact\_Count)

emails\_open\_df <- spread(emails\_open\_df, Contact\_Count, Emails.Opened, fill = 0)

emails\_open\_df <- mutate(emails\_open\_df, Total\_Emails\_Opened = Contact\_One + Contact\_Two + Contact\_Three + Contact\_Four + Contact\_Five + Contact\_Six + Contact\_Seven + Contact\_Eight + Contact\_Nine + Contact\_Ten)

emails\_open\_df <- select(emails\_open\_df, Domain, Total\_Emails\_Opened)

company\_df <- inner\_join(company\_df, emails\_open\_df, by = "Domain")

# Creating Total\_Emails\_Clicked variable in company\_df

emails\_clicked\_df <- select(contact\_df, Domain, Emails.Clicked, Contact\_Count)

emails\_clicked\_df <- spread(emails\_clicked\_df, Contact\_Count, Emails.Clicked, fill = 0)

emails\_clicked\_df <- mutate(emails\_clicked\_df, Total\_Emails\_Clicked = Contact\_One + Contact\_Two + Contact\_Three + Contact\_Four + Contact\_Five + Contact\_Six + Contact\_Seven + Contact\_Eight + Contact\_Nine + Contact\_Ten)

emails\_clicked\_df <- select(emails\_clicked\_df, Domain, Total\_Emails\_Clicked)

company\_df <- inner\_join(company\_df, emails\_clicked\_df, by = "Domain")

**The following new variables were created:**

**total\_revenue:** The sum of the revenue from all orders related to each of the companies.

**total\_order\_count:** The sum of the number of orders related to each company.

**Days\_Between\_All\_Orders:** The number of days between the first and last order from each company.

**Ave\_Days\_Between\_Orders:** Total days between first and last order divided by Days\_Between\_All\_Orders minus one.

**days\_since\_last\_order:** The number of days since the most recent order for each company.

**days\_since\_first\_order:** The number of days since the first order date for each company and today’s date.

**Ave\_Order\_Amount:** The average amount spent per order for each company.

**Ave\_Reorder:** The average amount spent per order (excluding the first order) for each company.

**after\_cutoff\_date:** A Boolean field where 1 represents companies that made their first order after 2016-01-12, and 0 represents all other companies.

**Total\_Emails\_Delivered:** The sum of emails delivered to contacts related to each company

**Total\_Emails\_Opened:** The sum of emails opened by contacts related to each company.

**Total\_Emails\_Clicked:** The sum of emails clicked by contacts related to each company.

**Emails\_Opened\_Percent:** Emails opened divided by emails delivered.

**Emails\_Clicked\_Percent:** Emails clicked divided by emails delivered.

**Order\_One\_Date – Order\_Fourteen\_Date:** Fourteen date fields for each of the company’s order dates.

**Order\_One\_Amount – Order\_Fourteen\_Amount:** Fourteen fields for the revenue amount of each company’s orders.

**First\_Order\_Traffic\_Source:** The traffic source of each company’s initial order.

**Analysis of Model**

As mentioned previously, the total lifetime value of a Business to Business customer is what I am attempting to predict. The linear regression model will be created for this purpose using the dependent variable total\_revenue which represents total lifetime value.

StandDesk began using their current CRM on January 13th, 2016. The data related to interactions between contacts and StandDesk is fairly accurate for companies that made a first purchase on January 13th or later. Unfortunately, this is not the case for the data related to companies that purchased prior to January 13th, 2016. This lack of accurate data has become a significant limitation to creating an accurate model.

Sub\_company\_df was created by filtering out companies with less than $2,500 in total revenue and any company that made a first purchase prior to Jan. 12th, 2016.

sub\_company\_df <- filter(company\_df, total\_revenue > 2500)

sub\_company\_df$after\_cutoff\_date <- ifelse(sub\_company\_df$Order\_One\_Date > "2016-01-12", 1, 0)

sub\_company\_df <- filter(sub\_company\_df, after\_cutoff\_date == 1)

The best model for predicting total revenue for sub\_company\_df:

**modelsub1 <- lm(total\_revenue ~ days\_since\_last\_order + Industry + Total\_Emails\_Opened\*Total\_Emails\_Delivered, data = sub\_company\_df)**

**summary(modelsub1):**

Pr(>|t|)

(Intercept) 0.007163 \*\*

days\_since\_last\_order 0.073802 .

IndustryArchitecture & Planning 0.655837

IndustryBanking 2.31e-06 \*\*\*

IndustryComputer Software 0.394870

IndustryConstruction 0.171733

IndustryConsumer Goods 0.025273 \*

IndustryEvents Services 0.162865

IndustryHospital & Health Care 0.000893 \*\*\*

IndustryInformation Technology and Services 0.256142

IndustryInternational Trade and Development 0.373463

IndustryMachinery 0.033101 \*

IndustryMarket Research 0.433913

IndustryMarketing and Advertising 0.019597 \*

IndustryMusic 0.585836

IndustryNon-Profit Organization Management 0.018171 \*

IndustryOil & Energy 0.804494

IndustryPrimary/Secondary Education 0.561998

IndustryReal Estate 0.053572 .

IndustryRenewables & Environment 0.046556 \*

IndustrySemiconductors 0.820650

IndustrySports 0.000140 \*\*\*

IndustryTelecommunications 0.112001

IndustryUnknown 0.169794

Total\_Emails\_Opened 0.000817 \*\*\*

Total\_Emails\_Delivered 0.257120

Total\_Emails\_Opened:Total\_Emails\_Delivered 2.32e-05 \*\*\*

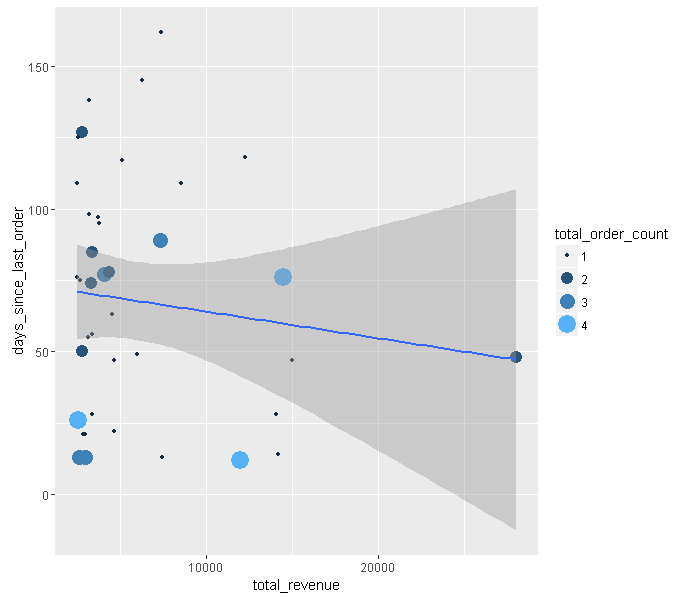
---

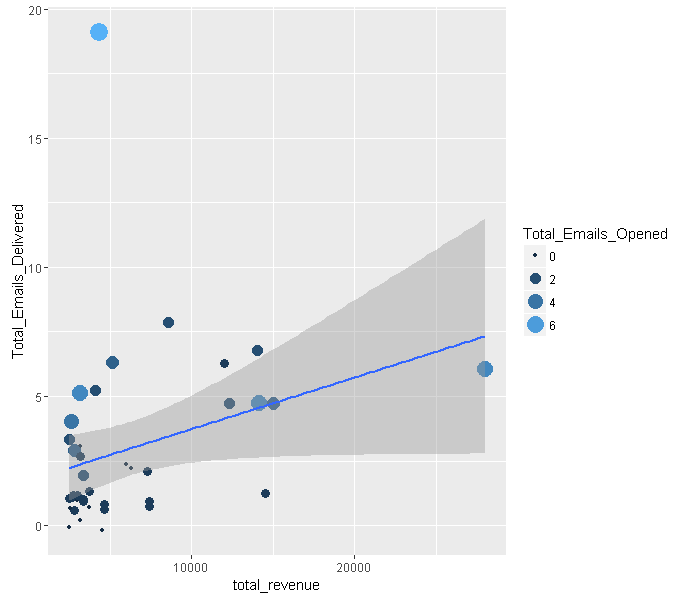
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

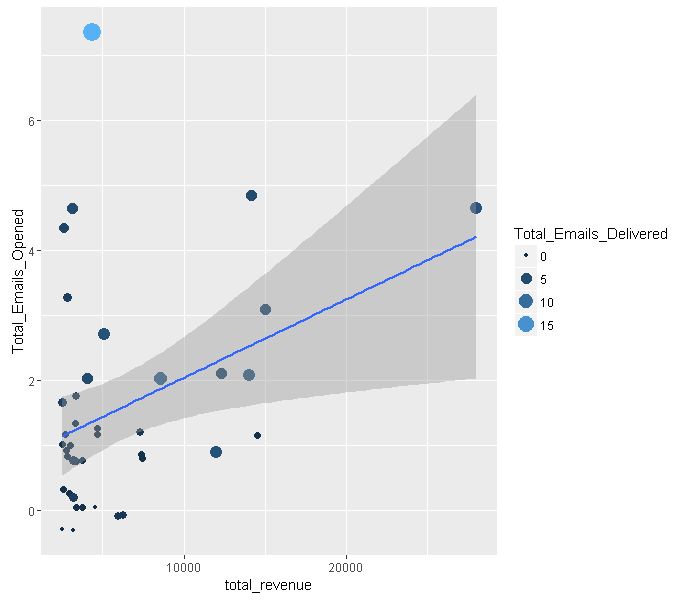
Residual standard error: 2292 on 13 degrees of freedom

Multiple R-squared: 0.9356, Adjusted R-squared: 0.8069

F-statistic: 7.269 on 26 and 13 DF, p-value: 0.0002812



****

****



**Next Steps**

1. Update Model to Predict Revenue for the next 30 days
2. Improve model by adding demographic data

**Recommended Use**

Once the Model is updated to predict revenue for the next 30 days, import new data once per month. Run model on updated data to determine which companies the model predicts will have an increase in total revenue.

Create a specific marketing campaign to reach out to companies that are predicted to have an increase in revenue.