ISE 5103 Intelligent Data Analytics

Homework 6 - Modeling Competition

Daniel Carpenter, Sonaxy Mohanty, & Zachary Knepp

October 2022

Contents

General Data Prep

For general data preparation, please see conceptual steps below. See `.rmd` file for detailed code.

Read Training Data

Clean data to ensure each read variable has the correct data type (factor, numeric, Date, etc.)

Create `numeric` and `factor` *base* `data frames`

Make data set of `numeric` variables called `df.train.base.numeric`

Make data set of `factor` variables called `df.train.base.factor`

`(a, i)` - Data Understanding

Create a data quality report of `numeric` and `factor` data
Created function called `dataQualityReport()` to create factor and numeric QA report

Numeric Data Quality Report

- `pageviews` has some null values, but there are an insignificant amount, so we will just drop those rows.

Num_Numeric_Variables	Total_Observations
4	70071

variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
visitNumber	0	1	3.1	8.7	1	1	1	2	155
timeSinceLastVisit	0	1	256450.2	1164717.4	0	0	0	10375	30074517
revenue	0	1	10.2	99.5	0	0	0	0	15981
pageviews	8	1	6.3	11.7	1	1	2	6	469

Factor Data Quality Report

- Location data unknown, so add an `Unknown` label for `null` values
- Appears that few people use website from the ads, which cause many null values. See more details below.

Num_Factor_Variables	Total_Observations
28	70071

variable	n_missing	complete_rate	n_unique	top_counts
sessionId	0	1.00	70071	200: 1, 400: 1, 600: 1, 700: 1
custId	0	1.00	47249	234: 155, 558: 135, 455: 129, 818: 115
channelGrouping	0	1.00	8	Org: 27503, Soc: 13528, Ref: 13482, Dir: 11824
deviceCategory	0	1.00	3	des: 53986, mob: 13868, tab: 2217
isTrueDirect	0	1.00	2	0: 42026, 1: 28045
bounces	0	1.00	2	0: 40719, 1: 29352
newVisits	0	1.00	2	1: 46127, 0: 23944
browser	1	1.00	27	Chr: 51584, Saf: 12007, Fir: 2407, Int: 1357
source	2	1.00	131	goo: 29233, you: 12708, (di: 11825, mal: 10840
continent	85	1.00	5	Ame: 42508, Asi: 13697, Eur: 11992, Oce: 901
subContinent	85	1.00	22	Nor: 38860, Sou: 4823, Nor: 3601, Wes: 3563
country	85	1.00	176	Uni: 36941, Ind: 3044, Uni: 2330, Can: 1918
operatingSystem	307	1.00	15	Mac: 23970, Win: 23707, And: 8074, iOS: 7487
medium	11827	0.83	5	org: 27503, ref: 27010, cpc: 2085, aff: 911
networkDomain	33448	0.52	5014	com: 2890, ver: 1372, rr.: 1319, com: 1247
topLevelDomain	33448	0.52	183	net: 15027, com: 6297, tr: 874, in: 868
region	38485	0.45	309	Cal: 11254, New: 3468, Ill: 1047, Tex: 909
city	39028	0.44	477	Mou: 4569, New: 3465, San: 2183, Sun: 1362
referralPath	43062	0.39	383	/: 11419, /yt: 4359, /yt: 842, /an: 836
metro	49183	0.30	72	San: 10072, New: 3526, Los: 1050, Chi: 1047
campaign	67310	0.04	6	AW : 1229, Dat: 911, AW : 575, tes: 35
keyword	67412	0.04	415	6qE: 997, 1hZ: 213, Goo: 183, (Re: 182
adwordsClickInfo.gclId	68245	0.03	1405	Cj0: 14, Cjw: 10, CIy: 9, Cj0: 9
adwordsClickInfo.page	68260	0.03	5	1: 1806, 2: 2, 3: 1, 5: 1
adwordsClickInfo.slot	68260	0.03	2	Top: 1771, RHS: 40, emp: 0
adwordsClickInfo.adNetworkType	68260	0.03	1	Goo: 1811, emp: 0
adwordsClickInfo.isVideoAd	68260	0.03	1	0: 1811
adContent	69230	0.01	27	Goo: 449, Dis: 82, Goo: 79, Ful: 49

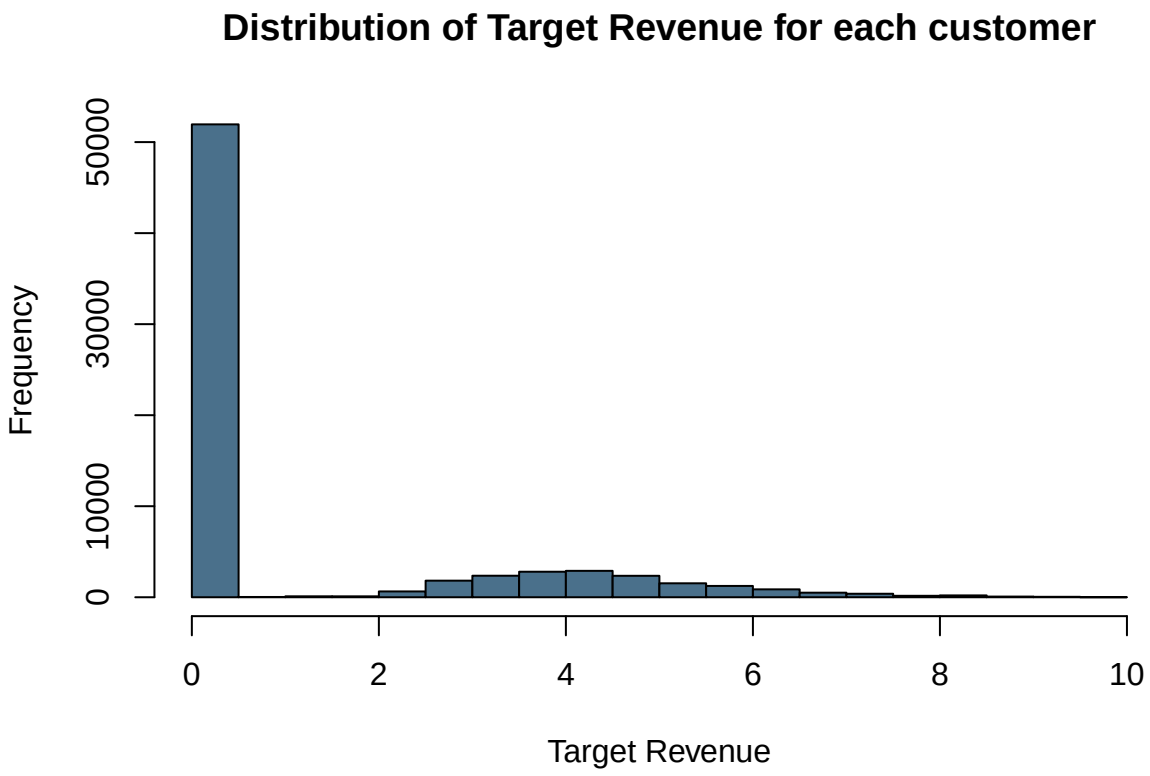
Exploratory Analysis

```
#aggregate revenue
CustRev <- stats::aggregate(df.train.base$revenue,
                     by=list(df.train.base$custId),
                     FUN = sum,
                     na.rm = TRUE)

#renaming fields
names(CustRev) <- c('custId', 'totalRevenue')

#merging datasets
df.train.merge <- merge(df.train.base, CustRev, by='custId')

#applying transformation
df.train.merge$totalRevenue <- df.train.merge$totalRevenue + 1
df.train.merge$totalRevenue <- log(df.train.merge$totalRevenue)
```

Analysis 1:

- Checking the distribution of the transformation of the aggregrate customer-level sales value based on the natural log:

```
hist(df.train.merge$totalRevenue,
     col = 'skyblue4',
     main = 'Distribution of Target Revenue for each customer',
     xlab = 'Target Revenue')
```

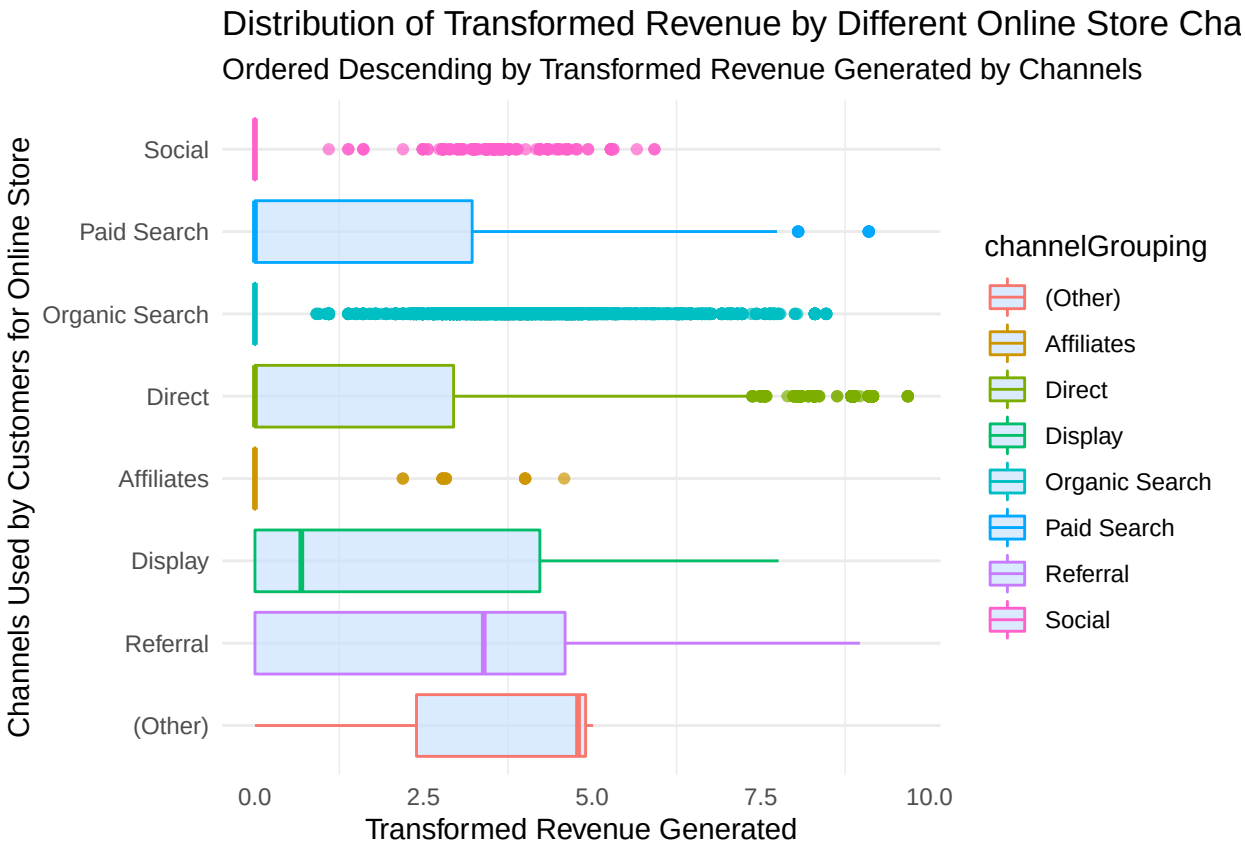


- We can see that the transformed revenue doesn't look like a normal distribution with a spike at 0 revenue which means it can be an outlier.

Analysis 2:

- Correlation between features in the dataset

```
df.train.merge %>%
  ggplot(aes(x = fct_reorder(channelGrouping, desc(totalRevenue) ),
             y = totalRevenue) ) +
  # Boxplots
  geom_boxplot(aes(color = channelGrouping), fill = 'lightsteelblue1', alpha = 0.7) +
  coord_flip() +

  # Theme, y scale format, and labels
  theme_minimal() +
  theme(panel.grid.major.x = element_blank()) +

  #scale_y_continuous(labels = comma) +
  labs(title = 'Distribution of Transformed Revenue by Different Online Store Channels',
       subtitle = 'Ordered Descending by Transformed Revenue Generated by Channels',
       x = 'Channels Used by Customers for Online Store',
       y = 'Transformed Revenue Generated')
```


Distribution of Transformed Revenue by Different Online Store Cha
Ordered Descending by Transformed Revenue Generated by Channels
Channels Used by Customers for Online Store
Social
Paid Search
Organic Search
Direct
Affiliates
Display
Referral
(Other)
0.0
2.5
5.0
7.5
10.0
Transformed Revenue Generated
channelGrouping
(Other)
Affiliates
Direct
Display
Organic Search
Paid Search
Referral
Social

(a, ii) - Data Preparation

For general data preparation, please see conceptual steps below. See `.rmd` file for detailed code.

Clean up Null Data

See that when `region` is `Osaka Prefecture` and `city` is `Osaka` some location details are `NULL`

- Implication: the other fields can be manually set to correct values based on region and city criteria
- So, set `location related` null fields to `know` description for the above `region` and `city` condition

See that when `continent` is `null`, then other `location` related fields are also null

- Implication: these other fields depend on the `continent` variable
- So, set `location related` null fields to `Unknow` description

See that when `medium` is `null`, then other `ad`, `keyword` and `campaign` related fields are (mostly) null

- Implication: these other fields depend on the `medium` variable
- So, set these null fields to `None` description, since a null value indicates the user did not has `no traffic source`

See that when `campaign` is `null`, then some `ad` related fields are (mostly) null

- Implication: these other fields depend on the `campaign` variable
- So, set `adwordsClickInfo.page` null fields to `None` description, since a null value indicates the user did not come using an advertisement

Similar to campaign, whenever `keyword` is NA, some `ads` is null

Similar to the campaign data, if the `adContent` is null, label as `No Ad`.

- Implications: If there is no ad Content of the traffic source then there is no no referral path

Similar to the campaign data, if the `adwordsClickInfo.adNetworkType` is null, then all `ad` related variables are also `NULL`.

- Implications: If there is no ad search then customer didn't see any ad.

Similar to the adwordsClickInfo.adNetworkType data, if the `adwordsClickInfo.page` is null, then some `ad` related variables are also `NULL` and there is no referral source.

- Implications: If there is no ad published on a page then customer didn't see any ad.

If `network domain` is `NULL` then all the related domains are also NULL.

Setting `referralPath` for NAs.

Setting `adwordsClickInfo.gclId` for NAs.

Now we have very few null values rows. Let's simply remove them. See below for how many.

```
## [1] "There are 318 rows with nulls"
```

```
## [1] "That equates to 0.5% rows with nulls"
```

```
## [1] "Total Rows Remaining: 69753"
```

```
## [1] "Before cleaning, there are 24 factor columns with more than 4 unique values"
```

```
## [1] "After cleaning, there are 2 columns with more than 5 unique values (omitting NA's)"
```

Group by Customer

Get list of customers who visited once and twice

Group by customer & Sum up all numeric data

- Filter to only the customers who visited twice
- Get the unique visits and choose the first visit
- THis is just an assumption! Not the best, but we have to make a choice.
- Append unique customers to non-unique customers (that are now unique)
- Note not using all columns, only columns NOT specific to the model

```
## [1] 46967
```

```
## [1] 46967
```

```
## [1] 28
```

```
## [1] 28
```

Create `targetRevenue` Variable

```
df.train.clean.cust <- df.train.clean.cust %>%
  mutate(targetVariable = log(revenue + 1)) %>%
  dplyr::select(-revenue)
```

Create dataset without the `custID` field called `df.train.clean.noCust`

(a, iii) - Modeling

OLS Model

Fit the Model

- Initially created a model with all variables, then used `stepAIC()` to identify important variables
- Implemented in the OLS model to realize a better fit model.

```
# The OLS model
# See RMD for stepAIC function that generated these relevant variables for the model
ols <- lm(targetVariable ~ operatingSystem + country + metro + city + networkDomain +
            source + keyword + isTrueDirect + referralPath + bounces +
            newVisits + pageviews,
          data = df.train.clean.noCust)
```

View and Interpret Results

Model	Notes	Hyperparameters	RMSE	Rsquared
OLS	lm	N/A	0.93	0.5

Model 2: PCR Model

Fit the Model

- Based on model testing, highest R^2 is around 68 number of components.
- Fits data much better than the former model.

View and Interpret Results

Model	Notes	Hyperparameters	RMSE	Rsquared
PCR	pcr	ncomp = 36	0.94	0.49

Model 3: MARS

Fit the Model

- Use MARS model from earth package.
- Fits data similarly to the former models.

View and Interpret Results

Model	Notes	Hyperparameters	RMSE	Rsquared
MARS	caret and earth	Degree = 1 , nprune = 8	0.77	0.66

Model 4: Elastic Net Model

Fit the Model

View and Interpret Results

Model	Notes	Hyperparameters	R
Elastic Net	caret and elasticnet	Alpha = 0.2 , Lambda = 0.000381198688071757	

(a, iv) - Debrief

Summary Table

Model	Notes	Hyperparameters	RMSE	Rsquared
OLS	lm	N/A	0.93	0.50
PCR	pcr	ncomp = 36	0.94	0.49
MARS	caret and earth	Degree = 1 , nprune = 8	0.77	0.66
Elastic Net	caret and elasticnet	Alpha = 0.2 , Lambda = 0.000381198688071757	0.93	0.50

Interpretations of Debrief

Apply to Test Data

- Need to clean test data like we did in the train
- Note all comments for the main model apply here
- Then apply the models to this dataset
- Outputs a CSV with predicted customer log revenue
- For general data preparation, please see conceptual steps below. See `.rmd` file for detailed code.

```
## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
```

```
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".

## Warning in sorted_count(x): Variable contains value(s) of "" that have been
## converted to "empty".
```