Analysis of Customer Preferences for Zooplus' Candy Offering

Neha Savant

Table of Contents

1. Scenario

Zooplus in addition to its pet supplies, plans to expand its offering in the confectionery segment. Within the business, it wants to strategize the introduction of a brand-new label for candies. These will be sold alongside other candy brands on Zooplus' online shopping website.

Internal brainstorming sessions rendered several preferences. However, the Category Manager wants to opt for a data driven approach to address this problem. She collaborated with a market research group and mined a dataset that contained a list of 85 competitor confectionery brands along with their characteristics based on contents, flavor, form, and packaging. It also contains price points and the overall win percentage according to 269,000 matchups.

2. Objective

The objective of this study is to identify which characteristics of a candy drives the overall purchase tendency of the customer and thus, recommend the business to include those characteristics in Zooplus' new candy label.

The focus of decision making is to analyse customer preferences and zero in on evaluated options on analytical grounds.

3. Getting Started with Data

3.1 Understanding the Data

The dataset 'candy-data.csv' consists of 13 variables and 85 observations. The 85 observations represent 85 distinct candies of competitor brands described over ten physical characteristics and one price point.

The table below describes all the variables from the dataset. The characteristic variables hold binary values where 1s indicate a presence and 0 indicates an absence of that characteristic in the candy.

Col. No.	Data Header	Value Description
1	chocolate	Does the candy contain chocolate?
2	fruity	Is it fruit flavoured?
3	caramel	Is there caramel in the candy?
4	peanutyalmondy	Does it contain peanuts, peanut butter or almonds?
5	nougat	Does it contain nougat?
6	crispedricewafer	Does it contain crisped rice, wafers, or a cookie component?
7	hard	Is it a hard candy?
8	bar	Is it a candy bar?
9	pluribus	Is it one of many candies in a bag or box?
10	sugarpercent	The percentile of sugar it falls under within the data set.
11	pricepercent	The unit price percentile compared to the rest of the set.
12	winpercent	The overall win percentage according to 269,000 matchups.

Table 3.1

The attributes - chocolate, caramel, peanutyalmondy, nougat, and crisped rice water are associated with the ingredients in the candies. The attribute fruity indicates whether the candy has a fruit flavour to it. A hard candy is determined by 1 in the attribute hard, a 0 value is indicative of a soft candy. If the candy is in a bar shaped, the attribute bar is set to 1. If the candy is packaged in a box along with other candies, it is indicated as 1 in the pluribus column. The attribute sugarpercent determines the percentile of sugar content in a candy with respect to the other candies in the dataset. Similarly, the pricepercent is the unit price percentile when compared to the rest.

The winpercent denotes the percentage of people who prefer a particular candy over another randomly chosen candy from the dataset.

The first nine attributes hold binary values 0 or 1, sugarpercent and pricepercent range from 0 to 1, and winpercent ranges from 0 to 100.

At the outset, the candy data-set is fairly clean. There are no duplicates, missing or NULL values in it. None of the columns are redundant and all of them contain a meaningful distribution.

3.2 Proposing the Hypothesis

A combination of characteristics basis the first 11 candy attributes is to be recommended such that the probability of a customer picking Zooplus' candy is maximum. Customer sentiments are associated with the win percent of a candy.

Thus, the hypothesis –

H_o: There is no significant relationship between the characteristics of a candy in the dataset and its ranking

H_a: There is a significant relationship between all or some characteristics of a candy in the dataset and its ranking.

The intent of this study is to put the null hypothesis H_o to test and deduce reproducible results and conclusions with the help of statistical tests and machine learning algorithms.

4. Statistical Instruments

4.1 Statistical Tools

The statistical tools used for data analysis are

i. R Version 3.6.1 'Action of the Toes'
ii. R Studio Version 1.2.5019 © 2009-2019 RStudio, Inc.
iii. Microsoft Excel (Microsoft Office ProPlus) Version 1712
   a. Analysis ToolPak ANALYS32.XLL
   b. Solver Add-in SOLVER.XLAM

The data models deployed in this study can be found in the R code here.

4.2 Statistical Techniques

In addition to a variety of summary statistics, below is the list of the statistical and machine learning techniques used in this study.

i. Pearson, Polyserial, and Polychoric Correlations
ii. Multivariate Linear Regression
iii. Polynomial Regression
iv. Ridge Regression
v. Lasso Regression
vi. Elastic Net
vii. Random Forest
viii. Principal Component Analysis
ix. Boruta search
x. RPART (Recursive Partitioning And Regression Trees)
xi. Genetic Algorithm Feature Selection
xii. Simulated Annealing Feature Selection
xiii. DALEX (Descriptive mAchine Learning EXplanations)

5. Exploring Data

5.1 Setting up the R Environment

The environment is first cleaned and required packages are installed in the environment.

```
# Clean environment
rm(list=ls())

# Load Packages
packages = c("dplyr","ggplot2","polycor","reshape2", "lattice", "boot", "MASS",
             "cvTools", "polycor", "DAAG", "randomForest", "glmnet", "caret",
             "rpart", 'FactoMineR', "factoextra", "Boruta", 'RRF', 'ggpubr', 'DALEX')

ipak <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)}

ipak(packages)
```

Figure 5.1

Next, the working directory is set, and the candy dataset is loaded. The first column, competitorname is unbranched from the working dataset 'candy' since it is nominal in nature and less useful statistically.

```
# Set Working Dir
setwd("F:/N_Docs/DE/Zooplus")

# Load Data
candy.data = read.csv('candy-data.csv')

# Remove the first column - competitorname (candy names)
candy = candy.data[,-1]

# Check data, first hand analysis
head(candy)
```

Figure 5.2

5.2 Data Manipulation

Upon checking the structure of the dataset candy, it is seen that the variables from chocolate to pluribus are considered as integers and therefore, they need to be converted into factor data type with 2 levels.

```
# convert columns 1 to 9 to factors
for(i in 1:9){
  candy[,i] = as.factor(as.character(candy[,i]))
}

str(candy)


# 'data.frame': 85 obs. of  12 variables:
# $ chocolate       : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 1 1 1 ...
# $ fruity          : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 2 ...
# $ caramel         : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 2 ...
# $ peanutyalmondy  : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 2 1 1 ...
# $ nougat          : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 2 1 1 1 ...
# $ crispedricewafer: Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
# $ hard            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
# $ bar             : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 1 1 1 ...
# $ pluribus        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 1 ...
# $ sugarpercent    : num  0.732 0.604 0.011 0.011 0.906 ...
# $ pricepercent    : num  0.86 0.511 0.116 0.511 0.511 ...
# $ winpercent      : num  67 67.6 32.3 46.1 52.3 ...
```

Figure 5.3

For ease of calculations, the columns sugarpercent and pricepercent are rounded up to two decimal places.

```
candy$sugarpercent = round(candy$sugarpercent,2)
candy$pricepercent = round(candy$pricepercent,2)
```

Figure 5.4

Over the course of this study, at a stage where feature engineering (Section 7.1.4) is implemented, three new attributes are added to the dataset. These are custom transformed attributes derived from manipulation of one or more columns in the dataset.

These derived attributes are –

i. **sugarbyprice**: When two candies are compared over this attribute, a higher value will suggest that the candy is sweeter and cheaper as compared to the other that is less sweet and expensive. It is derived by the formula below.

$$\text{sugarbyprice} = \frac{\text{sugarpercent}}{\text{pricepercent}}$$

ii. **winbyprice**: A higher winbyprice indicates the candy is much liked and cheaper than the candy it is compared with.

$$\text{winbyprice} = \frac{\text{winpercent}}{\text{pricepercent}}$$

iii. **pricepercent.sq**: This was added in the dataset because a quadratic/polynomial distribution was observed on the plot of winpercent vs pricepercent.

$$\text{pricepercent.sq} = (\text{pricepercent})^2$$

6. Statistical Analysis – I

6.1 Descriptive Statistics

6.1.1 Numerical Variables

The descriptive statistics of the three numerical variables are broken down into measures of central tendency and measures of variability (spread) in the table below.

statistic	sugarpercent	pricepercent	winpercent
Count	85	85	85
Range	0.98	0.97	61.73
Minimum	0.01	0.01	22.45
1st Quartile	0.22	0.255	39.1
Mean	0.48	0.47	50.32
Median	0.47	0.47	47.83
3rd Quartile	0.73	0.65	59.90
Maximum	0.99	0.98	84.18
Standard Error	0.03	0.03	1.60
Standard Deviation	0.28	0.29	14.71
Variance	0.08	0.08	216.51
Kurtosis	-1.13	-1.15	-0.58
Skewness	0.10	0.13	0.33

Table 6.1

Central Tendency: The mean and the median of all the three attributes are approximately same. This indicates that the data has a fairly normal distribution and there must be very few or no outliers in the data.

Variance: The measure of dispersion is close to 0 about 0.08 for sugarpercent and pricepercent that indicates that there is little variability in them.

Symmetry and Peaked-ness: The values of Kurtosis and Skewness are well within -2 and +2 and this is indicative of the fact that the data is normally distributed for all the three attributes. A distribution with a negative kurtosis value for these attributes signifies that the distribution has lighter tails and a flatter peak.

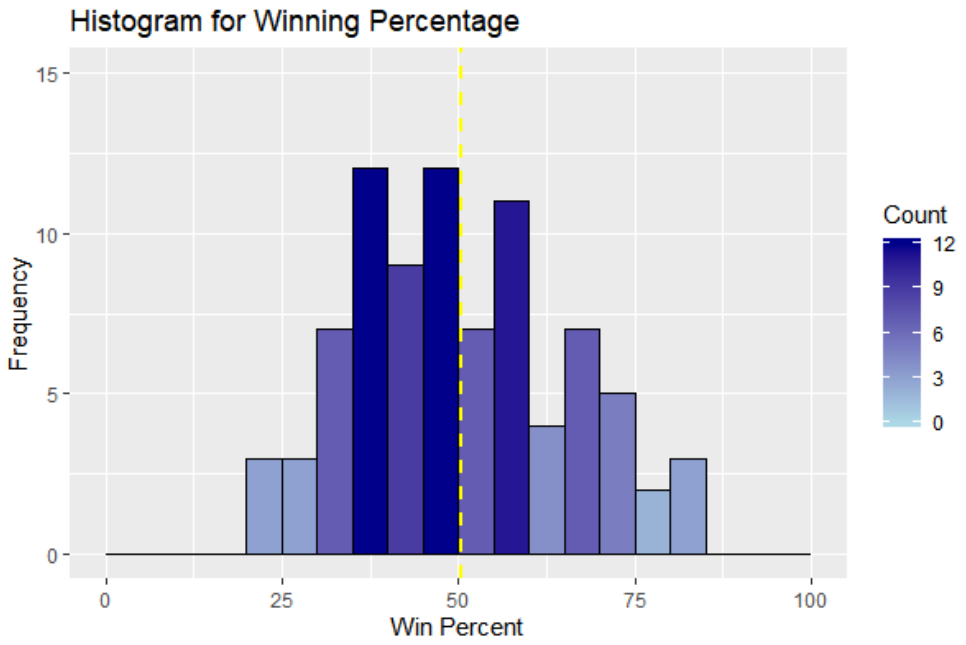

Figure 6.1

Figure above is the histogram of the attribute winpercent. The yellow dashed line denotes the mean and it implies that on an average, a candy with randomly picked characteristics has a 50% chance of winning.

Notably, this exercise aims at recommending candy characteristics that have more than the average chances of winning.

6.1.2 Categorical Variables

The nine categorical variables have two levels and the distribution counts of these variables are in the table below.

The candies in the dataset have an even distribution of non-chocolate as it is to chocolate, fruity and non-fruity, and pluribus or non-pluribus. There are lesser caramel, nutty, nougat filled, and cookie-contained candies in the dataset. Additionally, the dataset contains more number of hard and bar-shaped candies.

0	48	47	71	71	78
1	37	38	14	14	7

0	78	70	64	41
1	7	15	21	44

Table 6.2

6.2 Exploratory Analysis

6.2.1 Categorical Variables

The distribution of candies with chocolate as an ingredient in them is shown in the box-plot below. There is a clear split in the rankings, and this implies that the winning percentage is above average if there is a presence of chocolate in the candy. Besides, the notches of the box plots do not overlap and thus, it can be said that true medians for a chocolaty and a non-chocolaty candy differs.

This analysis tells that chocolate can be a good characteristic in determining the ranking for a candy.

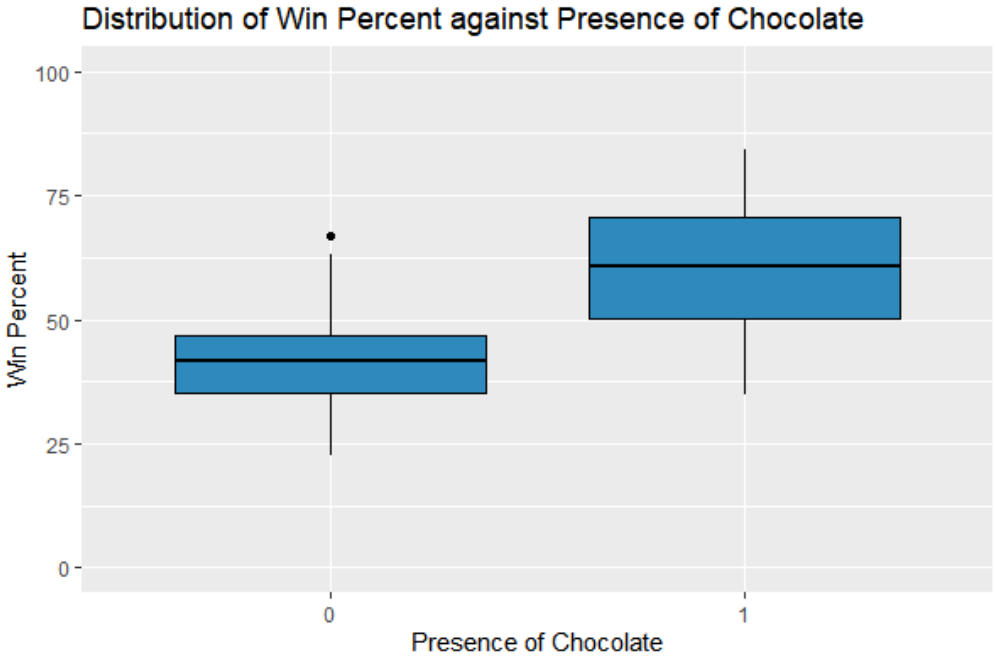

Figure 6.2

Similarly, the box plots of the other eight characteristics are in the figure below. Let's look at each characteristic box plot.

The probability of a non-fruity candy being picked is more than that of a fruity candy. The ranking of fruity candies, on a median statistic is 42% whereas that of a non-fruity one, is about 57%. However, the notches overlap and thus, findings beyond descriptive analysis need to be pursued.

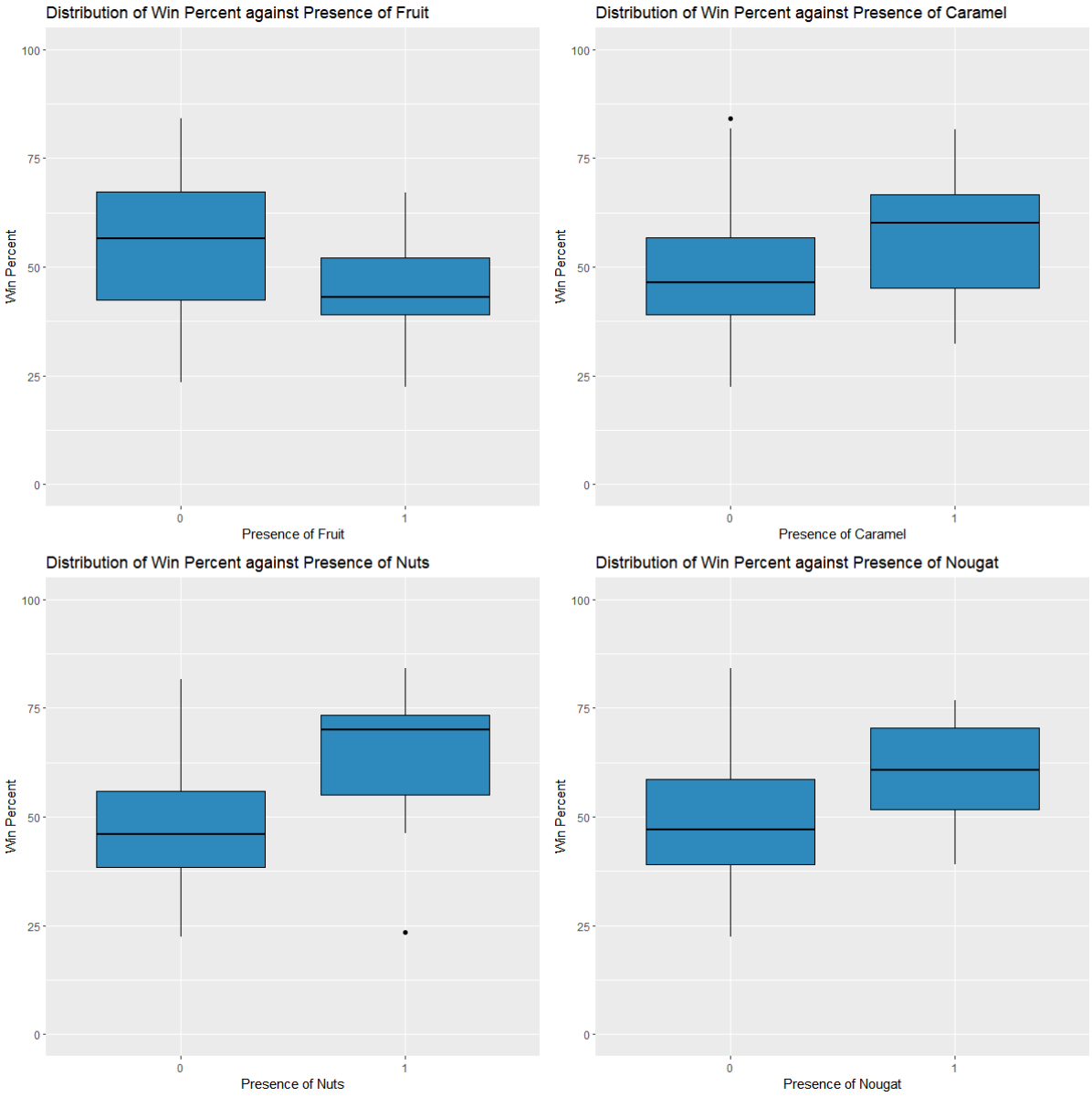

Figure 6.3

On the same note, the attributes caramel, nougat and pluribus show similar overlapping of the boxes at two levels 0 and 1. Therefore, a deeper analysis will be done to statistically note their significance in the recommendation engine.

Subsequently, a good split of levels can be observed in the box plots of attributes peanutyalmondy (nuts), crispedricewafer (cookie), hard (hardness), and bar shaped candies. Their distributions are clearly separated for higher and lower ranking candies and therefore, they can be potential features in our recommended list.

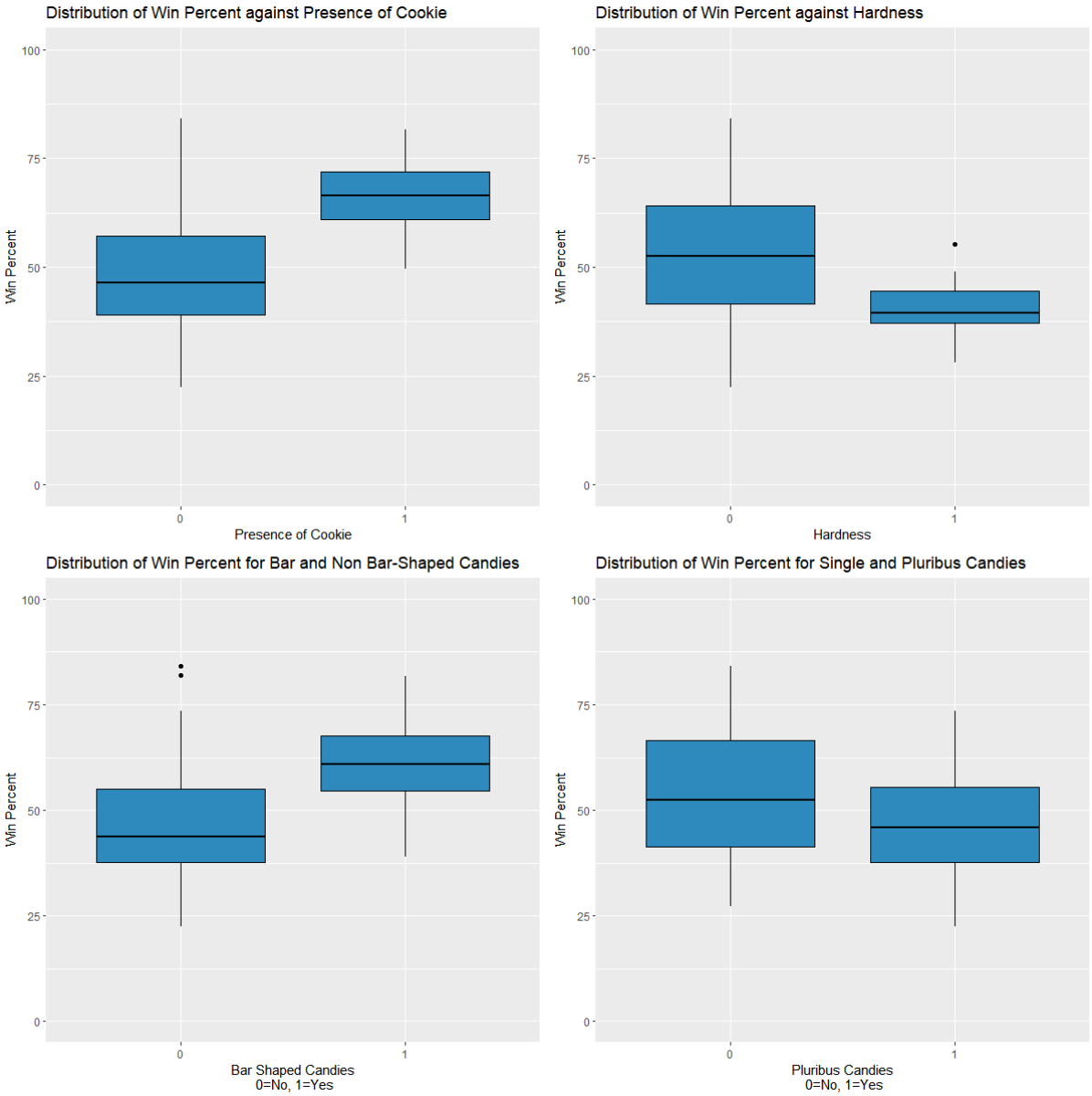

Figure 6.4

Hence, through a high-level observation of candy characteristics from the box plots, a recommended candy can be proposed to have chocolate, nuts and cookie contents, it shall be soft, and in a bar shape.

6.2.2 Numerical Variables

The scatter-plot of sugarpercent against the rankings of the candies show that the points in the association space cluster very loosely. A vague trend can be perceived with the black dashed line on the scatter-gram. It indicates that there is a poor correlation between the two attributes.

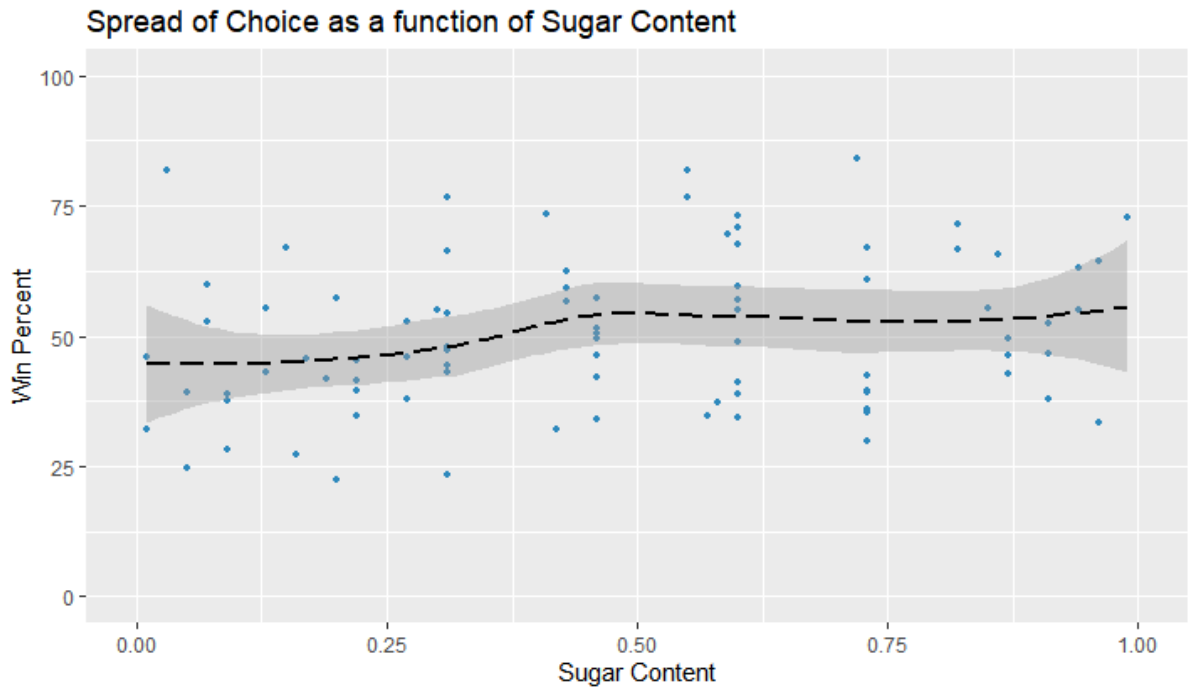

Figure 6.5

This is also evident from candy positions with respect to the sugar content below. The two highest ranked candies – Reese's peanut butter cup and Reese's miniatures have a dramatic difference in sugar contents yet are the highest ranked.

On the other hand, most candies are in the low sugar – low ranking or high sugar – high ranking buckets.

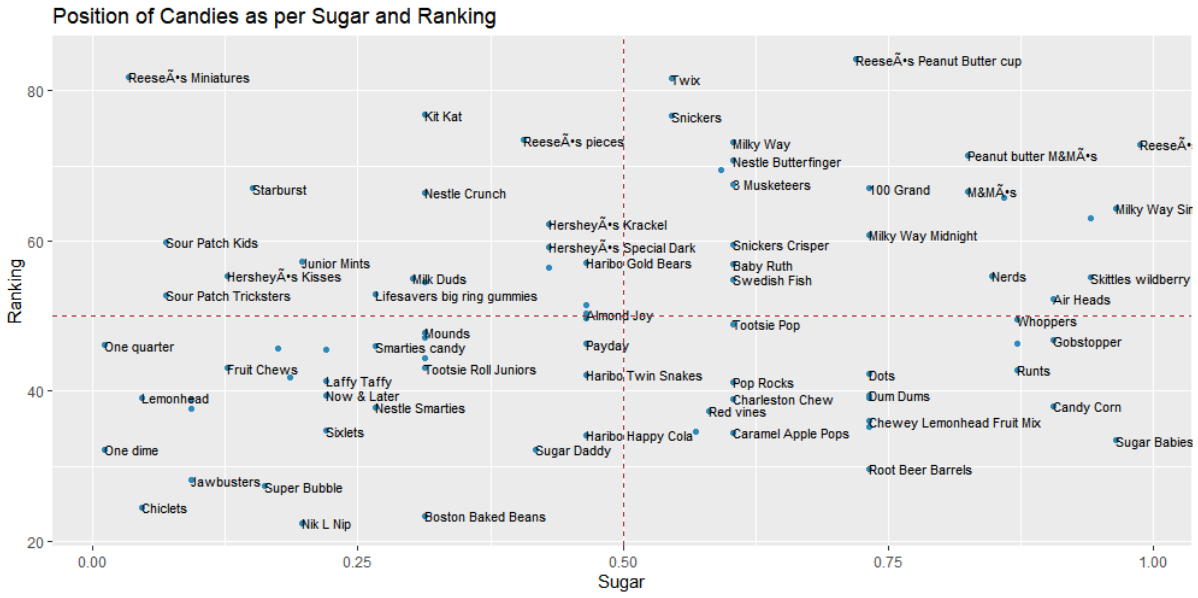

Figure 6.6

Let's observe the relationship between price percent and ranking.

There is a clear non-linear relationship between price percent and ranking. An inverted U-shaped trend can be noticed. Hence, a polynomial pricing component can be added while statistically testing the significance of this characteristic on winning percent.

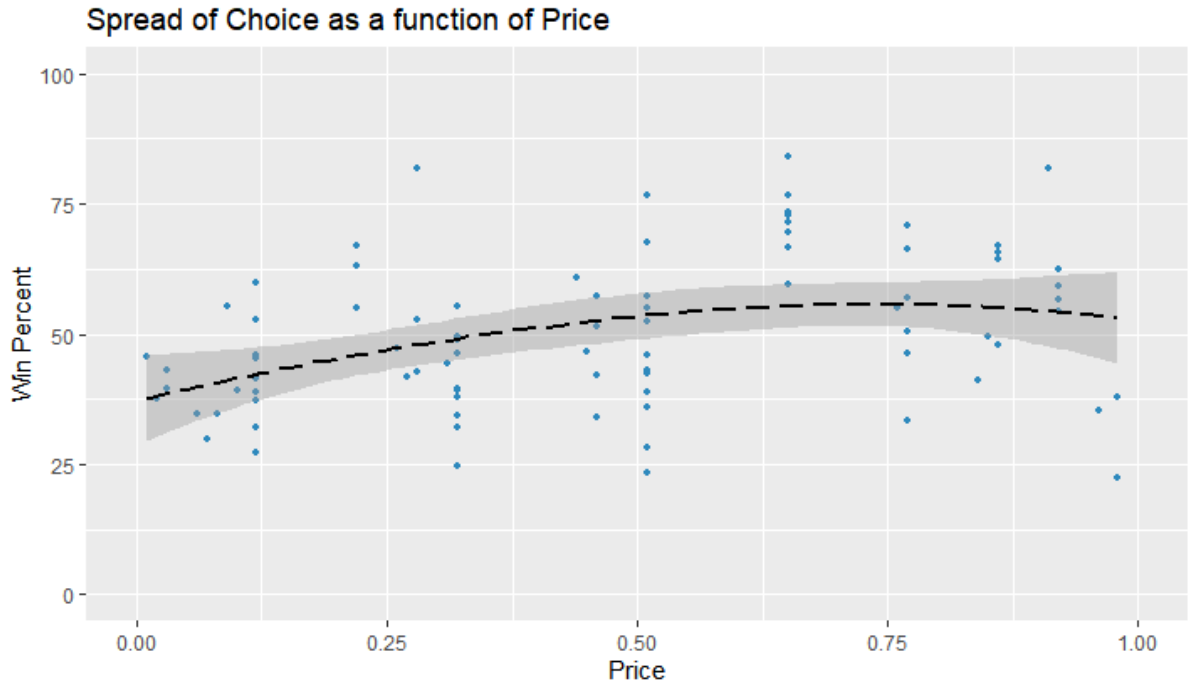

Figure 6.7

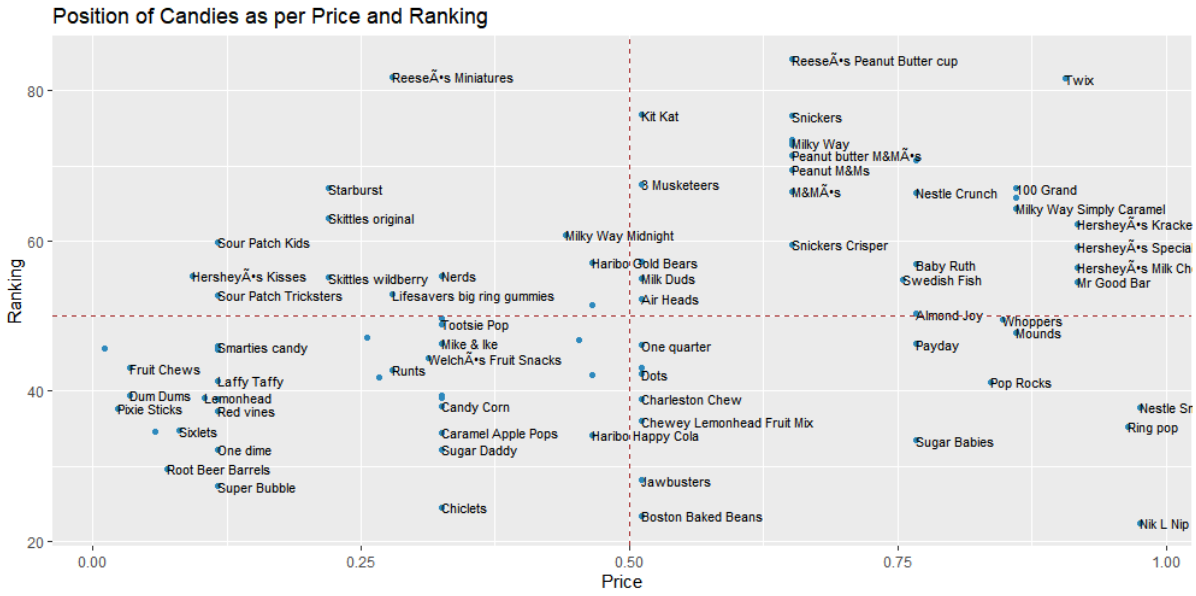

Figure 6.8

6.2.3 Competitor Variable

The top 30 candies based on their ranks is in the lollipop chart below.

Reese's, Hershey's, and Milky Way are the champions in this list. The brands have all their candies listed in the Top 30 list.

Reese's have all 4 variants in Top 10 rankings.

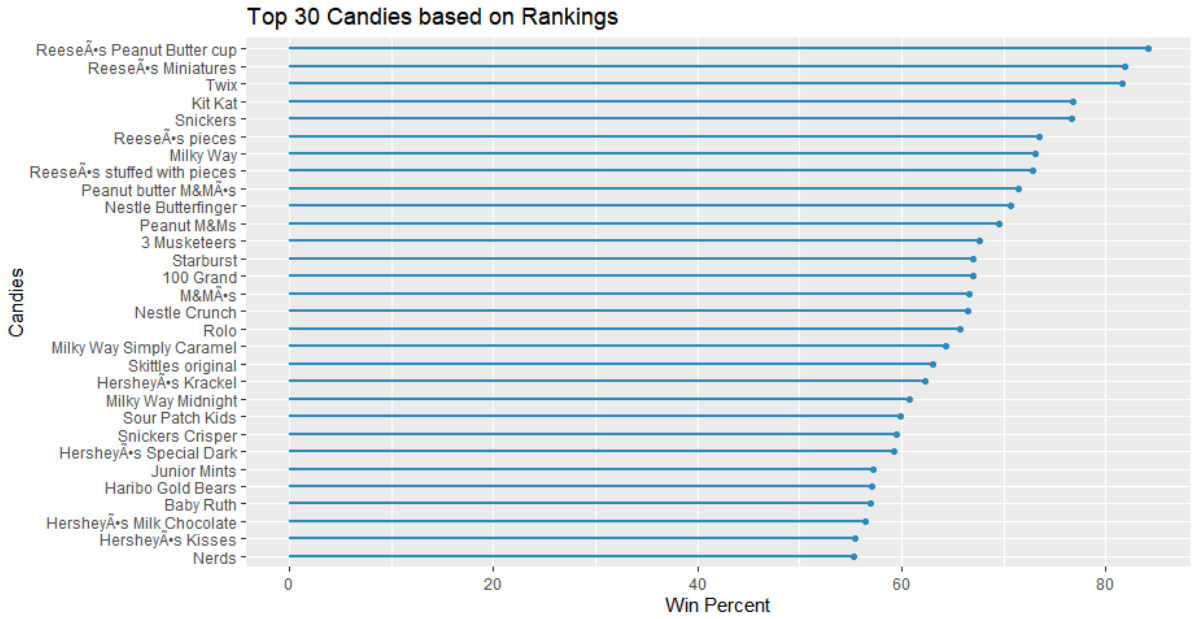

Figure 6.9

7. Statistical Analysis - II

7.1 Inferential Statistics

While descriptive statistics and exploratory analysis have helped to summarize the data, inferential statistics can be deployed to reach conclusions based on statistical evidences.

7.1.1 Correlation

The dataset contains a combination of categorical and continuous variables. Thus, a combination of Pearson, Polyserial, and Polychoric correlation tests are used for estimating the correlation between variables.

```
# Correlation
cor = hetcor(candy)
```

Figure 7.1

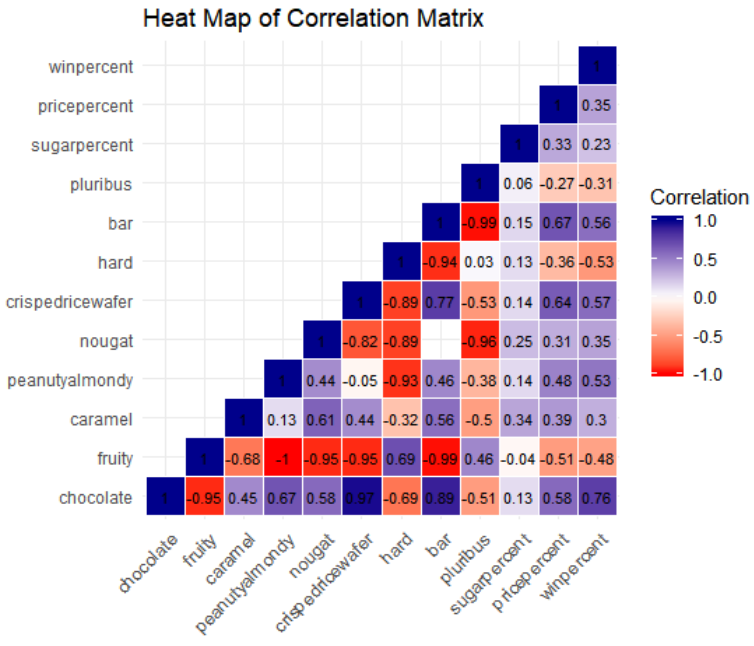

Figure 7.2

The degree of correlation can be estimated with the visual aid of the heat map.

Observing the rightmost vertical bar of the heat map, it can be noted that chocolate and ranking are positively and strongly correlated amongst other attributes. The attributes – bar, peanutyalmondy, and crispedricewafer too have a positive impact on the win percentage.

The hardness of a candy is inversely or negatively correlated to the ranking which implies softer candies in the dataset have higher rankings than harder ones. Also, non-fruity candies have better chances of winning than fruit flavoured ones. Such an association was previously observed through box plots too.

Multicollinearity: A considerable degree of association between independent variables (all the variables except winpercent) is observed in the heat map of correlation matrix.

The attribute bar is substantially correlated with chocolate (+60%) and pluribus (-59%). Similarly, the attribute chocolate and fruity have a correlation coefficient of -0.74 (74%) and can be considered as a red flag, a case of multicollinearity.

Fundamentally, sophisticated machine learning algorithms implemented in the further chapters of this study, will eliminate one of these from the feature selection engine.

7.1.2 Multivariate Linear Regression

To begin with feature selection, let's start with the simplest model - linear regression where the attribute winpercent is the dependent variable and candy characteristics are the independent variables.

Model Name: lm.model.1

```
# Model 1: Multivariate Regression Model
lm.model = lm(winpercent~. , data = candy)
summary(lm.model)
```

Figure 7.3

The p-values in the model summary indicate chocolate, peanutyalmondy, fruity, crispedricewafer, and negative coeff. of the three attributes - hard, pricepercent and sugarpercent are good predictors of winpercent. For instance, the presence of chocolate in the

candy increases its ranking by 19.74 times while a nutty candy will improve the chances of winning by 10 times. Also, negative coefficients of attributes such as hard indicate that a softer candy is recommended. Having said that, the model makes use of all the variables to predict the outcome whether or not it is significant.

The adjusted R-squared quoted is 47.1% which is lower than the average winning threshold set for this dataset (that is, 50%). Hence, the adjusted co-efficient of determination (adjusted R-squared) needs to be improved.

```
Call:
lm(formula = winpercent ~ ., data = candy)

Residuals:
    Min      1Q  Median      3Q     Max
-20.224  -6.625   0.199   6.842  23.868

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         34.534      4.320    7.99  1.4e-11 ***
chocolate1          19.748      3.899    5.07  3.0e-06 ***
fruity1              9.422      3.763    2.50   0.0145 *
caramel1             2.224      3.657    0.61   0.5449
peanutyalmondy1     10.071      3.616    2.79   0.0068 **
nougat1              0.804      5.716    0.14   0.8885
crispedricewafer1    8.919      5.268    1.69   0.0947 .
hard1               -6.165      3.455   -1.78   0.0785 .
bar1                 0.442      5.061    0.09   0.9307
pluribus1           -0.854      3.040   -0.28   0.7794
sugarpercent         9.087      4.659    1.95   0.0550 .
pricepercent        -5.928      5.513   -1.08   0.2858
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.7 on 73 degrees of freedom
Multiple R-squared:  0.54,      Adjusted R-squared:  0.471
F-statistic:  7.8 on 11 and 73 DF,  p-value: 9.5e-09
```

Figure 7.4

To improve the adjusted R-squared value, let's take a stepwise elimination approach on the previous model.

7.1.3 Stepwise Regression

Model Name: lm.model.2

```
# Model 2 : Stepwise Regressed Version of Model 1,
#           penalises insignificant attributes to quote Adusted R squared
lm.model.2 = step(lm.model)
summary(lm.model.2)
```

Figure 7.5

The output summary of the step model shows a slight improvement in the adjusted R squared value (49.2%) though it is still lesser than the average ranking for a candy whose characteristics are randomly selected.

```
Call:
lm(formula = winpercent ~ chocolate + fruity + peanutyalmondy +
    crispedricewafer + hard + sugarpercent, data = candy)

Residuals:
     Min       1Q   Median       3Q      Max
-21.4825  -6.7038   0.5828   5.9098  24.0312

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        32.921      3.516   9.365 2.12e-14 ***
chocolate1         19.145      3.586   5.339 8.92e-07 ***
fruity1             8.878      3.559   2.494  0.01474 *
peanutyalmondy1     9.483      3.445   2.753  0.00735 **
crispedricewafer1   8.379      4.483   1.869  0.06536 .
hard1              -5.682      3.288  -1.728  0.08794 .
sugarpercent        8.052      4.135   1.948  0.05507 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.49 on 78 degrees of freedom
Multiple R-squared:  0.5283,    Adjusted R-squared:  0.492
F-statistic: 14.56 on 6 and 78 DF,  p-value: 4.5e-11
```

Figure 7.6

The stepwise regression has modelled a function based on statistically significant variables based on p-values and backward elimination. The recommended characteristics as per this model are chocolate, fruity, peanutyalmondy, crispedricewafer, negative co-eff. of hardness that is softness, and sugarpercent.

7.1.4 Stepwise Multivariate Linear Regression with Feature Engineering

In the pursuit to attain a better adjusted R-Squared value, two new features sugarbyprice and winbyprice are added to the dataset. A multivariate regression algorithm is then implemented with stepwise elimination to model this dataset.

Model Name: lm.model.3 and lm.model.4

```
# Feature Engineering
candy$sugarbyprice = candy$sugarpercent/candy$pricepercent
candy$winbyprice = candy$winpercent/candy$pricepercent

# Model 3: With new features
lm.model.3 = lm(winpercent~. , data = candy)
summary(lm.model.3)

lm.model.4 = step(lm.model.3)
summary(lm.model.4)
```

Figure 7.7

The model summary suggests that one of the two newly added features, sugarbyprice is included in the model although its p-value 0.1 suggests otherwise. Apart from the recommended variables from lm.model.2, this model includes pricepercent based on a significant p-value. It is to be noted that the negative coefficient of pricepercent indicates that price is inversely related to ranking in this model.

```
Call:
lm(formula = winpercent ~ chocolate + fruity + peanutyalmondy +
    crispedricewafer + hard + sugarpercent + pricepercent + sugarbyprice,
    data = candy)

Residuals:
     Min       1Q   Median       3Q      Max
-22.2294  -5.1722   0.2126   6.7034  23.3550

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      36.4610     3.9528   9.224 4.98e-14 ***
chocolate1       20.4267     3.6419   5.609 3.15e-07 ***
fruity1           8.0639     3.5462   2.274  0.02579 *
peanutyalmondy1   9.8206     3.4509   2.846  0.00569 **
crispedricewafer1 9.6222     4.5412   2.119  0.03737 *
hard1            -4.8691     3.3548  -1.451  0.15079
sugarpercent     12.1179     4.6130   2.627  0.01042 *
pricepercent    -10.0656     5.8287  -1.727  0.08825 .
sugarbyprice     -0.6535     0.4084  -1.600  0.11372
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.37 on 76 degrees of freedom
Multiple R-squared:  0.5502,    Adjusted R-squared:  0.5029
F-statistic: 11.62 on 8 and 76 DF,  p-value: 1.233e-10
```

Figure 7.8

Furthermore, the Adjusted R-Squared value has only increased by 1% from the previous model. It means about 50.29% variability in the dependent variables is accounted by the model. However, it is still low.

7.1.5 Stepwise Multivariate Regression with Quadratic Term

Taking a step forward with feature engineering, a polynomial distribution between price percent and win percent was observed while exploring the data graphically. Thus, let's observe the result of a stepwise quadratic regression model by adding a squared term of price percent to the data set.

The model output claims that the quadratic feature is a significant predictor of the outcome variable and has included it in the model function. It has a coefficient weight of -9.48 which implies a unit decrease in pricepercent.sq will increase the ranking by 9.48 times! There model has thus eliminated the linear variable pricepercent and replaced it with the quadratic term pricepercent.sq.

Model Name: lm.model.5

```
# Add a quadratic pricepercent term
candy$pricepercent.sq = (candy$pricepercent)^2


lm.model.5 = lm(winpercent~., data = candy)
summary(lm.model.5)


lm.model.6 = step(lm.model.5)
summary(lm.model.6)
```

Figure 7.9

The adjusted R-Squared value has slightly increased to 50.6% but it still can be potentially improved for recommending characteristics of the candy such that the winning probabilities are higher.

```
Call:
lm(formula = winpercent ~ chocolate + fruity + peanutyalmondy +
    crispedricewafer + hard + sugarpercent + sugarbyprice + pricepercent.sq,
    data = candy)

Residuals:
     Min       1Q   Median       3Q      Max
-21.6000  -5.5076   0.7404   6.4672  23.2571

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      34.8607     3.5939   9.700 6.16e-15 ***
chocolate1       20.4915     3.6253   5.652 2.63e-07 ***
fruity1           8.0918     3.5308   2.292  0.02469 *
peanutyalmondy1   9.5260     3.4243   2.782  0.00681 **
crispedricewafer1 9.8087     4.5341   2.163  0.03366 *
hard1            -4.7371     3.3462  -1.416  0.16096
sugarpercent     11.0615     4.3330   2.553  0.01269 *
sugarbyprice     -0.5461     0.3764  -1.451  0.15093
pricepercent.sq  -9.4810     5.0815  -1.866  0.06593 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.34 on 76 degrees of freedom
Multiple R-squared:  0.553,     Adjusted R-squared:  0.506
F-statistic: 11.75 on 8 and 76 DF,  p-value: 9.852e-11
```

Figure 7.10

Checking the validity of a Linear Regression Model:

So far, for the recommendation of candy characteristics, the data was modelled on a linear regression algorithm. These models accounted for a variance of about 50.6% at maximum. Let's check the validity of a linear model on this data to understand if it was a good choice to start with.

Test 1: Histogram of the residuals should be normally distributed.
Test 2: Sum of residuals should be zero.

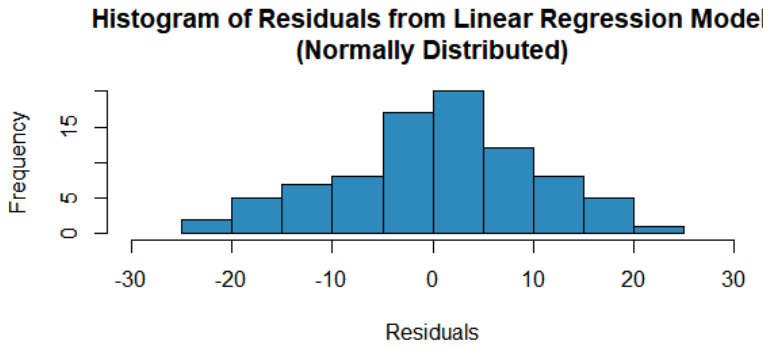

Figure 7.11

The histogram of residuals has a normal distribution. The sum of residues from the R-Code is equal to 5.009881e-15 which is approximately 0. Thus, it can be stated that the linear regressor is a suitable model, but it fails to explain degrees of variance in the data set effectively. (~50.6%)

7.1.6 Random Forest

A Random Forest model involves bagging or bootstrap aggregation to model data and that can be leveraged to increase the predictability of the variables in the dataset.

Model Name: rf.model.7

```
# Model 7: Random Forest
set.seed(3)

rf.model.7=randomForest(winpercent ~ ., data = candy, importance=TRUE)
rf.model.7
```

Figure 7.12

The model summary indicates a rise of precisely 3.82% in the variance and so, the adjusted R-Squared value quoted by the random forest model is 54.11%. Though there is a good rise in the variance explained, it can still be improved.

```
Call:
 randomForest(formula = winpercent ~ ., data = candy, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 98.18751
                    % Var explained: 54.11
```

Figure 7.13

Let's check which are the most important candy characteristics that the RF model has selected. The dot-chart of variable importance as measured by Random Forest is demonstrated below.

Depending on the type of importance, the left plot shows variable importance as a measure of mean decrease in accuracy while the right shows it based on a mean decrease in node impurity.

According to both the measures, nougat, caramel, hardness, and pluribus are the least determinative recommendations for the characteristics of the candy.

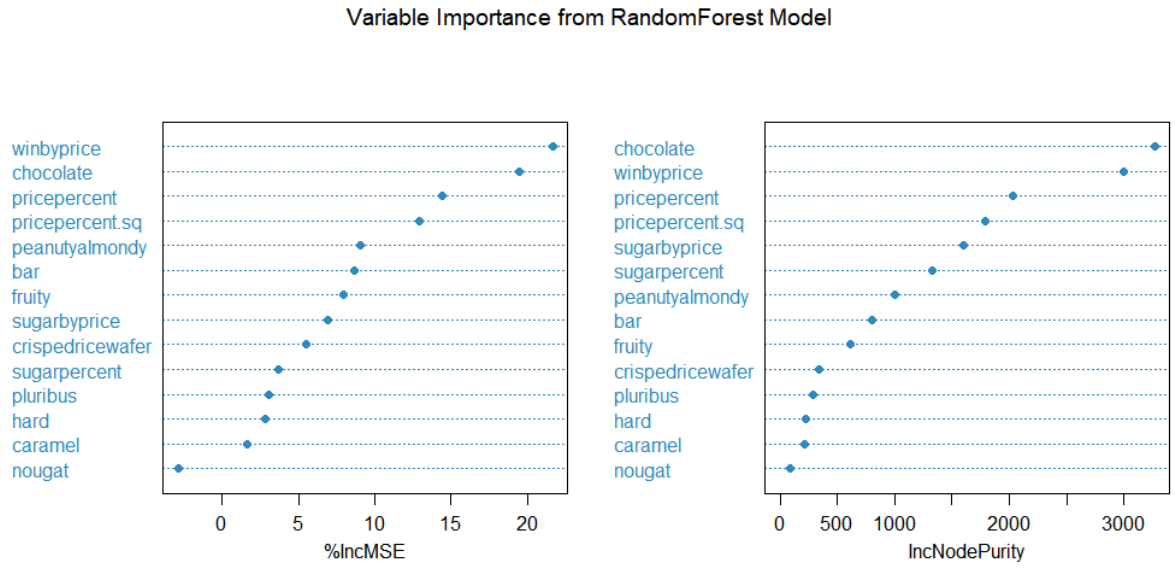

Figure 7.14

As a next step, a combination of 4 features from the top 10, 5, and 3 variables as per Inc MSE are deployed to re-run the random forest model.

Model Names: rf.model.8

```
# Considering a combination of top 4 (optimum) features for the next tree as per Inc MSE
set.seed(3)
rf.model.8=randomForest(winpercent ~  winbyprice + chocolate + pricepercent + pricepercent.sq
                        , data = candy, importance=TRUE)

rf.model.8
#% Var explained: 63.43%
```

Figure 7.15

It is observed that the top 4 variables winbyprice, chocolate, pricepercent, and pricepercent.sq make an optimum model with the highest variance explained i.e. 63.43%, highest so far. This means the top 4 variables are capable of explaining a variance of 63.43% in the dataset, which is a rise of 9.32% from the previous RF model.

Model Name: rf.model.9

```
# Considering top 6 (optimum) features for the next tree as per Inc Node Impurity
set.seed(3)
rf.model.9=randomForest(winpercent ~
                          winbyprice + chocolate + pricepercent + pricepercent.sq + sugarbyprice
                        +peanutyalmondy
                        , data = candy, importance=TRUE)

rf.model.9
#% Var explained: 60.56%
```

Figure 7.16

Similarly, as per Inc Node Impurity, the top 6 variables could explain a variance of 60.56% in the dataset.

7.1.7 Penalised Regression Method – Lasso Regression

A modification of linear regression, Lasso penalizes the model for the sum of absolute values of the weights. It introduces a hyperparameter alpha which penalizes weights.

Model Name: lasso.model.10

```
#################### Computing lasso regression

# Choosing the best lambda using cross-validation
set.seed(3)
cv = cv.glmnet(x, y, alpha = 1)

# Display the best lambda value
cv$lambda.min
#0.394

# Fit the lasso model on data
lasso.model.10 = glmnet(x, y, alpha = 1, lambda = cv$lambda.min)
lasso.model.10

# % Deviance Explained = 53.87%

# Display regression coefficients
coef(lasso.model.10)
```

Figure 7.17

The summary of the Lasso model denotes that 53.8% variance is explained by the model. There is a considerable drop in the variance from the RF Model.

The characteristics of a potential best-selling candy according to this model are chocolate, sugarpercent, crispedricewafer, peanutyalmondy, pricepercent.sq, fruity, and soft (negative hard). Nougat, pricepercent, and winbyprice have no significance as per Lasso.

```
> lasso.model.10

Call:  glmnet(x = x, y = y, alpha = 1, lambda = cv$lambda.min)

  Df   %Dev Lambda
1 11 0.5387 0.4322
> # Display regression coefficients
> coef(lasso.model.10)
15 x 1 sparse Matrix of class "dgCMatrix"
                            s0
(Intercept)        37.50532172
chocolate1         16.99933728
fruity1             4.62996862
caramel1            0.44445097
peanutyalmondy1     7.83542154
nougat1             .
crispedricewafer1   7.15514369
hard1              -3.43312975
bar1                0.61388277
pluribus1          -0.06964536
sugarpercent        8.32020761
pricepercent        .
sugarbyprice       -0.33217758
winbyprice          .
pricepercent.sq    -4.83208890
```

Figure 7.18

7.1.8 Penalised Regression Method – Ridge Regression

The Ridge Regression penalises the model for the sum of squared value of the weights. Therefore, the weights not only tend to have smaller absolute values, they are more evenly distributed, and tend to be as close as possible to zero.

Model Name: ridge.model.11

```
########### Computing ridge regression

# Find the best lambda using cross-validation

set.seed(3)
cv = cv.glmnet(x, y, alpha = 0)

# Display the best lambda value
cv$lambda.min
# 2.59

# Fit the final model on the data
rigde.model.11 = glmnet(x, y, alpha = 0, lambda = cv$lambda.min)
rigde.model.11
# % Deviance Explained = 53%

# Display regression coefficients
coef(rigde.model.11)
```

Figure 7.19

The Ridge model output summarizes that the model explains 53.19% variance in the dataset. Like Lasso, the recommended features of the candy in the order of decreasing importance are chocolate, sugarpercent, peanutyalmondy, crispedricewafer, negative coeff. of the attributes pricepercent.sq and hardness (which implies the candy should be softer), bar, and caramel.

```
> rigde.model.11

Call:  glmnet(x = x, y = y, alpha = 0, lambda = cv$lambda.min)

  Df   %Dev Lambda
1 14 0.5319  2.591
> # Display regression coefficients
> coef(rigde.model.11)
15 x 1 sparse Matrix of class "dgCMatrix"
                             s0
(Intercept)         39.313902288
chocolate1          13.060736290
fruity1              3.498485939
caramel1             1.040198801
peanutyalmondy1      8.064647408
nougat1              0.145092500
crispedricewafer1    7.636042477
hard1               -4.102308935
bar1                 2.250067980
pluribus1           -0.855086620
sugarpercent         9.486144025
pricepercent        -0.259738563
sugarbyprice        -0.499901481
winbyprice           0.001147846
pricepercent.sq     -5.050828071
```

Figure 7.20

7.1.9 Penalised Regression Method – Elastic Net Regression

Elastic Net includes both the absolute value penalization and squared penalization. It is a hybrid of Ridge and Lasso algorithms.

Model Name: elasticnet.model.12

```
############ Computing elastic net regession

# Build the model
set.seed(3)
elasticnet.model.12 = train(
  winpercent ~., data = candy, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneLength = 10
)
# Best tuning parameter
elasticnet.model.12$bestTune

#     alpha lambda
# 68   0.7   1.51
# The best alpha and lambda values are those values
# that minimize the cross-validation error

# Coefficient of the final model. You need
# to specify the best lambda

coef(elasticnet.model.12$finalModel, elasticnet.model.12$bestTune$lambda)
```

Figure 7.21

The model has aggressively set insignificant variables to zero as seen below. The attributes chocolate, peanutyalmondy, sugarpercent, crispedricewafer, (negative coeff.) hardness, and (negative coeff.) sugarbyprice are the selected features for a potent brand of candies, according to Elastic Net.

```
> coef(elasticnet.model.12$finalModel, elasticnet.model.12$bestTune$lambda)
15 x 1 sparse Matrix of class "dgCMatrix"
                             1
(Intercept)         41.44806917
chocolate1          12.88096833
fruity1              .
caramel1             .
peanutyalmondy1      5.69622408
nougat1              .
crispedricewafer1    4.65075894
hard1               -1.79728195
bar1                 .
pluribus1            .
sugarpercent         5.12876398
pricepercent         .
sugarbyprice        -0.09898057
winbyprice           .
pricepercent.sq      .
```

Figure 7.22

7.1.10 Principal Components Analysis

The use of PCA - a dimensionality reduction technique can help emphasize variation and bring out strong patterns in the dataset. Let's check how much variance is explained by the principal components.

Name: res.pca

```
###### Principal Compnents Analysis

res.pca <- PCA(candy.ohe[, -12], graph = FALSE)

#The proportion of variation retained by the principal components
#(PCs) can be extracted as follow :

eigenvalues <- res.pca$eig
```

Figure 7.23

The amount of variation retained by each PC is called eigenvalues. The first PC corresponds to the direction with the maximum amount of variation in the data set. The importance of PCs can be visualized from the scree plot below.

The first principal component corresponds to 33% of variance in the data, the second PC corresponds to 12.7%, while the third explains 9% and so on.

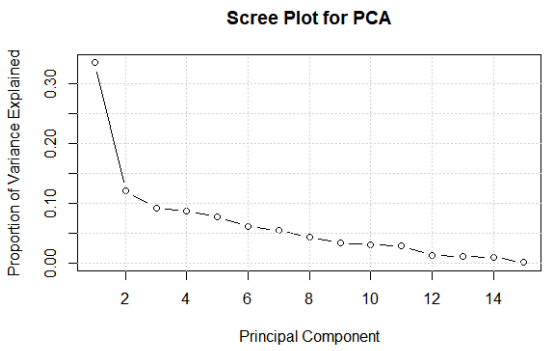

Figure 7.24

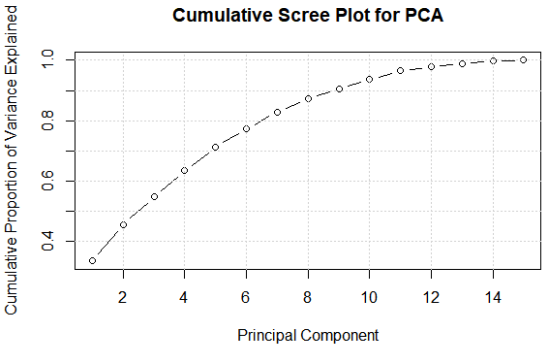

Figure 7.25

Overall, the first 5 Principal Components are capable of explaining about 70% variability in the dataset as seen in Table 7.1.

Variable	PC	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
pricepercent	1	**0.82**	-0.40	0.10	0.22	0.10
bar	1	**0.80**	0.27	0.10	-0.14	-0.22
pricepercent.sq	1	**0.78**	-0.33	0.10	0.31	0.05
chocolate	1	**0.76**	0.33	-0.13	0.24	0.08
caramel	1	**0.46**	0.16	0.35	-0.21	-0.13
pluribus	1	**-0.51**	-0.28	-0.11	0.38	0.39
fruity	1	**-0.71**	-0.38	0.20	-0.17	-0.14
winbyprice	2	-0.41	**0.73**	-0.06	0.36	0.03
sugarbyprice	2	-0.45	**0.60**	0.44	0.34	0.12
sugarpercent	3	0.23	-0.19	**0.74**	0.18	0.38
hard	3	-0.42	-0.10	**0.57**	**-0.08**	-0.16
nougat	4	0.42	0.35	0.20	**-0.60**	0.19
peanutyalmondy	5	0.46	0.13	-0.20	-0.09	**0.55**
crispedricewafer	5	0.45	0.01	0.07	0.43	**-0.60**

Table 7.1

Let's have a look at the loadings. The loadings are the correlation between a variable and a PC. The variables are plotted as points in the component space using their loadings as coordinates.

Cos2 are squared loadings for the variables. Statistically, if a variable is perfectly represented by only two components, the sum of the cos2 is equal to one. In this case the variables will be positioned on the circle of correlations.

In the circle of correlations in figure 7.26, the variables are positioned inside the circle. This implies more than 2 components are required to perfectly represent the data.

The variables chocolate, bar, pricepercent.sq, pricepercent, winbyprice, sugarbyprice, fruity are close to the circle circumference and are thus, are important in explaining the variability in the data set according to Principal Components Analysis.

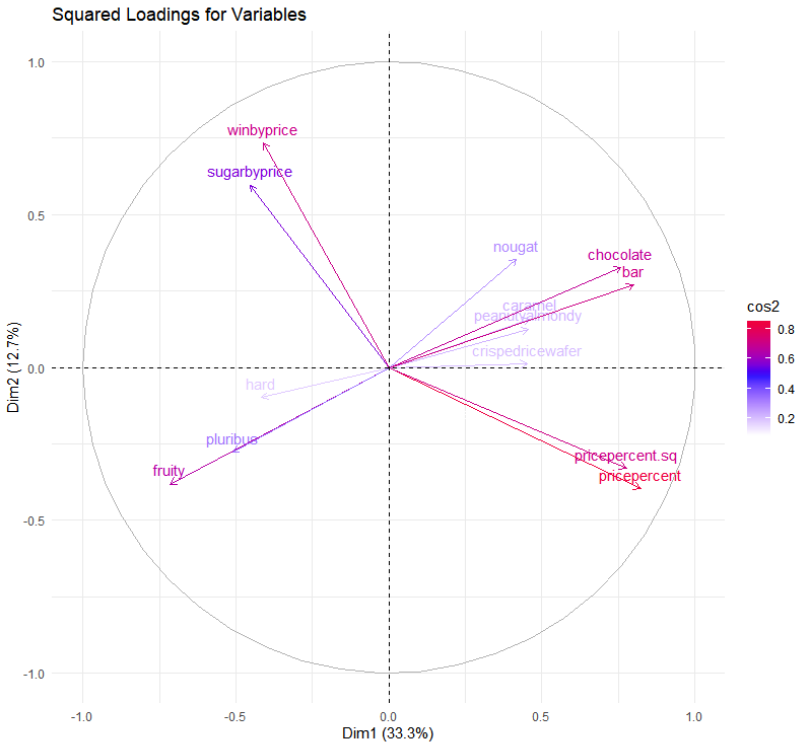

Figure 7.26

7.1.11 Boruta

Boruta is a feature ranking and selection algorithm based on random forests algorithm. Boruta clearly decides if a variable is important or not and helps to select variables that are statistically significant.

The strictness of the algorithm can be changed by adjusting the p-values that defaults to 0.01 and the number of times the algorithm is run. Higher the run, more selective is the algorithm in picking variables.

Model Name: boruta_model.14

```
# Perform Boruta search
boruta_model.14 <- Boruta(winpercent ~ ., data=candy, doTrace=0)
names(boruta_model.14)

boruta_model.14
```

Figure 7.27

Boruta performed 87 iterations and 10 attributes were confirmed important as seen in the Variable Importance chart.

The variables in green are 'confirmed' and the ones in red are not. The blue bars denote ShadowMax and ShadowMin, they are not actual features. They are used by the Boruta algorithm to determine the importance of a variable.

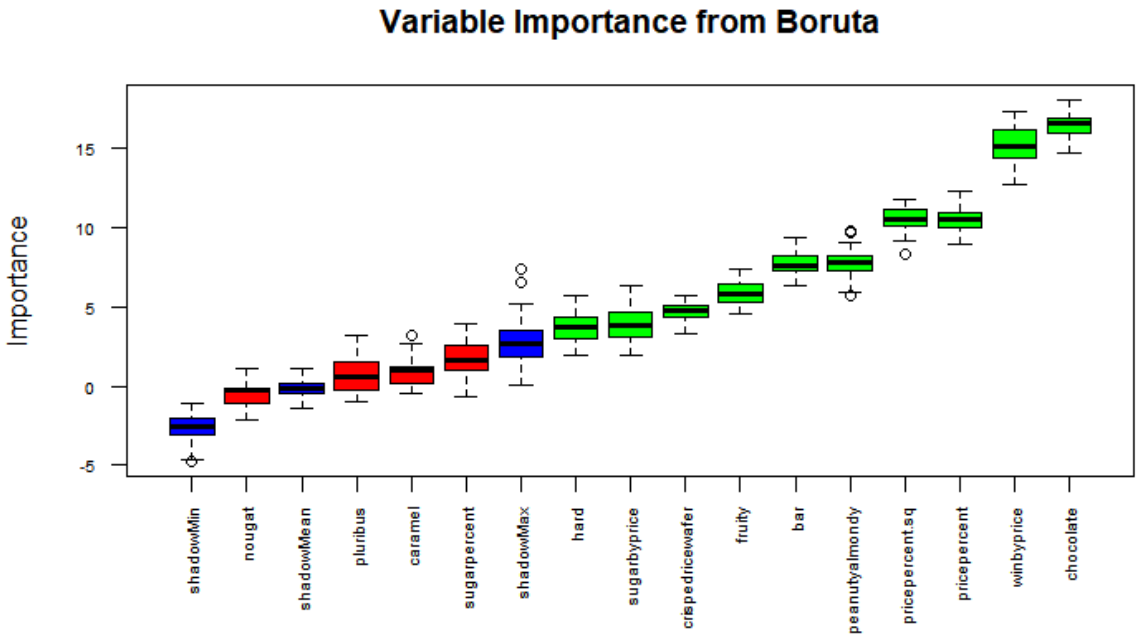

Figure 7.28

Thus, chocolate, fruity, peanutyalmondy, crispedricewafer, hard, bar, pricepercent, sugarbyprice, winbyprice, and pricepercent.sq are importance features that should be recommended for the new brand of candies.

7.1.12 RPART (Recursive Partitioning And Regression Trees)

The RPART algorithm splits the dataset recursively, which means that the subsets arising from a split are further split until a predetermined termination criterion is reached. At every step, the split is made based on the independent variable. This results in the largest possible reduction in heterogeneity of the dependent variable.

Model Name: rpart.model.15

```
# Feature selection using rpart model

set.seed(3)
rpart.model.15 <- train(winpercent ~ ., data=candy, method="rpart")
rpartImp <- varImp(rpart.model.15)
rpartImp
```

Figure 7.29

The summary output of the RPART model indicates five important variables namely, chocolate, pricepercent.sq, pricepercent, bar, and peanutyalmondy.

```
> rpartImp
rpart variable importance

                  Overall
chocolate1         100.00
pricepercent.sq     48.12
pricepercent        48.12
bar1                45.62
peanutyalmondy1     40.72
crispedricewafer1    0.00
caramel1             0.00
sugarbyprice         0.00
winbyprice           0.00
pluribus1            0.00
fruity1              0.00
hard1                0.00
nougat1              0.00
sugarpercent         0.00
```

Figure 7.30

7.1.13 Recursive Feature Elimination (RFE)

Recursive Feature Elimination determines important features in the dataset through subset sizes. In the R code, a subset size of 1 to 6 variables is chosen. The rfeControl parameter is set to Random Forest (rfFuncs) and the resampling method is set to repeated k-Fold cross validation method with repeats=5.

Model Name: lmProfile.16

```
####### Recursive Feature Elimination (RFE)
set.seed(3)
subsets <- c(1:6)

ctrl <- rfeControl(functions = rfFuncs,
                   method = "repeatedcv",
                   repeats = 5,
                   verbose = FALSE)

lmProfile.16 <- rfe(x=candy[, -12], y=candy$winpercent,
                    sizes = subsets,
                    rfeControl = ctrl)
```

Figure 7.31

According to the output of RFE, the least RMSE is achieved for the model subset size 4. The four important features of a candy listed by this algorithm are winbyprice, chocolate, pricepercent.sq, and pricepercent.

```
> lmProfile.16

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold, repeated 5 times)

Resampling performance over subset size:

 Variables   RMSE Rsquared    MAE RMSESD RsquaredSD MAESD Selected
         1 14.844   0.1853 12.391  2.401     0.1704 2.227
         2 10.622   0.5285  8.830  2.118     0.2028 1.860
         3  9.167   0.6609  7.463  2.197     0.1741 1.734
         4  8.783   0.6923  7.015  2.363     0.1588 1.794        *
         5  9.787   0.6232  7.827  2.411     0.1689 1.736
         6  9.218   0.6468  7.306  2.354     0.1634 1.740
        14  9.519   0.6230  7.568  2.426     0.1792 1.836

The top 4 variables (out of 4):
   winbyprice, chocolate, pricepercent.sq, pricepercent
```

Figure 7.32

7.1.14 Genetic Algorithm

A supervised feature selection with genetic algorithms was performed to determine important characteristics of a candy. However, this is a resource expensive algorithm and thus, I have chosen repeats = 2 which normally should be 100+.

Model Name: ga_obj.17

```
# Genetic Algorithm

# Define control function
ga_ctrl <- gafsControl(functions = rfGA,  # another option is `caretGA`.
                       method = "cv",
                       repeats = 2)

# Genetic Algorithm feature selection
set.seed(3)
ga_obj.17 <- gafs(x=candy[, -12],
                  y=candy[, 12],
                  iters = 2,   # normally much higher (100+)
                  gafsControl = ga_ctrl)

ga_obj.17

# Optimal variables
ga_obj.17$optVariables
```

Figure 7.33

The output summary indicates that 72% variability (which is the highest so far) is explained by the genetic model despite having only 2 iterations. The optimal characteristics suggested by this algorithm are pricepercent.sq, winbyprice, chocolate, pricepercent, and caramel.

```
Genetic Algorithm Feature Selection

85 samples
14 predictors

Maximum generations: 2
Population per generation: 50
Crossover probability: 0.8
Mutation probability: 0.1
Elitism: 0

Internal performance values: RMSE, Rsquared
Subset selection driven to minimize internal RMSE

External performance values: RMSE, Rsquared, MAE
Best iteration chose by minimizing external RMSE
External resampling method: Cross-Validated (10 fold)

During resampling:
  * the top 5 selected variables (out of a possible 14):
    pricepercent.sq (100%), winbyprice (100%), chocolate (60%), pricepercent (60%), caramel (50%)
  * on average, 5.4 variables were selected (min = 2, max = 10)

In the final search using the entire training set:
   * 2 features selected at iteration 2 including:
     pricepercent, winbyprice
   * external performance at this iteration is

      RMSE    Rsquared         MAE
    8.3762      0.7262      6.2159
```

Figure 7.34

7.1.15 Simulated Annealing Feature Selection

Simulated annealing is a global search algorithm that works by making small random changes to an initial solution and sees if the performance improved. If it sees an improvement, the changes are incorporated. However, it can still be accepted if the acceptance criteria is met by the difference of performance.

Model Name: sa_obj.18

```
# Simulated Annealing

# Define control function
sa_ctrl <- safsControl(functions = rfSA,
                       method = "repeatedcv",
                       repeats = 3,
                       improve = 5) # n iterations without improvement before a reset

# Genetic Algorithm feature selection
set.seed(3)
sa_obj.18 <- safs(x=candy[, -12],
                  y=candy[, 12],
                  safsControl = sa_ctrl)

sa_obj.18

# Optimal variables
sa_obj.18$optVariables
```

Figure 7.35

The model explains an unappealing 49.6% R-squared value, and suggests chocolate, fruity, peanutyalmondy, and sugarbyprice as important features in the dataset.

```
> sa_obj.18

Simulated Annealing Feature Selection

85 samples
14 predictors

Maximum search iterations: 10
Restart after 5 iterations without improvement (0.3 restarts on average)

Internal performance values: RMSE, Rsquared
Subset selection driven to minimize internal RMSE

External performance values: RMSE, Rsquared, MAE
Best iteration chose by minimizing external RMSE
External resampling method: Cross-Validated (10 fold, repeated 3 times)

During resampling:
  * the top 5 selected variables (out of a possible 14):
    winbyprice (70%), pricepercent.sq (63.3%), pricepercent (60%), chocolate (50%), pluribus (50%)
  * on average, 5.9 variables were selected (min = 2, max = 9)

In the final search using the entire training set:
   * 4 features selected at iteration 10 including:
     chocolate, fruity, peanutyalmondy, sugarbyprice
   * external performance at this iteration is

      RMSE     Rsquared          MAE
   11.2088       0.4962       9.1581

> # Optimal variables
> sa_obj.18$optVariables
[1] "chocolate"      "fruity"         "peanutyalmondy"
[4] "sugarbyprice"
```

Figure 7.36

7.1.16 DALEX Package

DALEX stands for **D**escriptive M**a**chine **Le**arning **Ex**planation. It contains explainers that help in understanding the link between input variables and model output.

Model Name: rf.model.19

```
#*********************************************************************
# Model 19: DALEX Package

# Train random forest model
set.seed(3)
rf_model.19 <- randomForest(candy$winpercent ~ ., data=candy[,-12], ntree=100)
rf_model.19
#   % Var explained: 54.07%


# Variable importance with DALEX
explained_rf <- explain(rf_model.19, data=candy[,-12], y=candy$winpercent)
explained_rf

# Get the variable importances
varimps = variable_dropout(explained_rf, type='raw')
```

Figure 7.37

The explainedrf object explains variable importance from the model output. It denotes how important a variable is based on the dropout loss, that is how much loss is incurred by removing a variable from the model. Thus, the attributes chocolate, winbyprice, pricepercent, sugarbyprice, pricepercent.sq, sugarpercent and fruity are amongst the most important features of this data set.

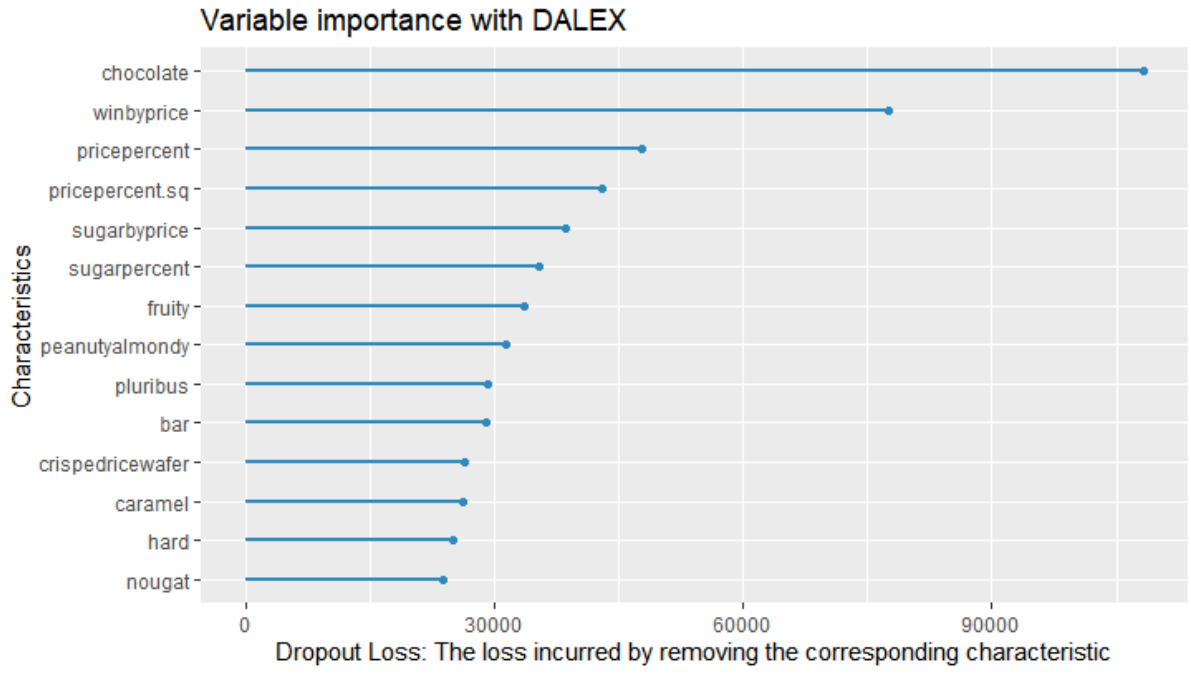

Figure 7.38

8. Model Summary

The table below is a quick view of all the features recommended by each of the nineteen algorithms.

sr.no	model	chocolate	fruity	caramel	peanuty almondy	nougat	crisped ricewafer	hard	bar	pluribus	sugar percent	price percent	sugar byprice	win byprice	price percent.sq
1	lm.model1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-
2	lm.model.2	✓	✓		✓		✓	✓			✓		-	-	-
3	lm.model.3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
4	lm.model.4	✓	✓		✓		✓	✓			✓	✓	✓		-
5	lm.model.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	lm.model.6	✓	✓		✓		✓	✓			✓		✓		✓
7	rf.model.7	✓	✓		✓		✓		✓		✓	✓	✓	✓	✓
8	rf.model.8	✓										✓		✓	✓
9	rf.model.9	✓			✓							✓	✓	✓	✓
10	lasso.model.10	✓	✓		✓		✓	✓			✓				✓
11	ridge.model.11	✓		✓	✓		✓	✓	✓		✓				✓
12	elasticnet.model.12	✓			✓		✓	✓			✓		✓		
13	res.pca	✓	✓						✓			✓	✓	✓	✓
14	boruta_model.14	✓	✓		✓		✓	✓	✓		✓	✓		✓	✓
15	rpart.model.15	✓			✓				✓			✓			✓
16	RFE - lmprofile.16	✓										✓		✓	✓
17	Genetic - ga_obj.17	✓		✓								✓		✓	✓
18	Simulated Annealing sa_obj.18	✓	✓		✓								✓		
19	DALEX - rf_model.19	✓	✓								✓	✓	✓	✓	✓

Table 8.1

Most models have recommended chocolate, peanutyalmondy, and price position as the choicest features that affect a candy's ranking. The least significant of them all are nougat and pluribus. It can also be seen that the derived features added in the dataset during data modelling have also proved significant and as a result, recommended by these algorithms.

9. Conclusion

As it turns out, different models exhibited different variables as important, or at best the degree of importance changed. This need not be called a conflict, as every algorithm offers its own perspective of how the variable can be useful depending on the learnability of algorithms.

Taking into consideration 19 machine learning algorithms, the variable significance in each statistical test was observed from their p-values. With a confidence interval of 95%, the null hypothesis formulated for this study is rejected with statistical evidences. It can be said that there is a significant relationship between the characteristics of a candy in the dataset and its ranking.

In order to identify the most recommended candy characteristics, the preferences of the feature selection engines are hereby summarized.

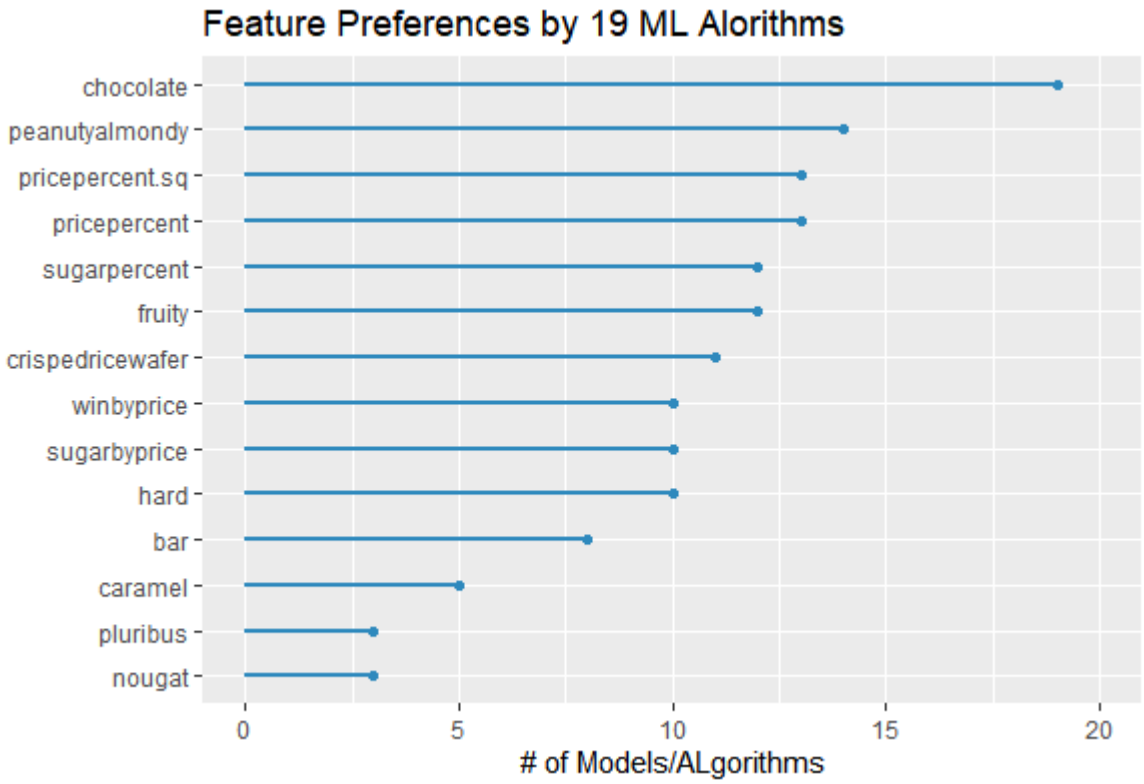

Figure 9.1

It is found that the candy feature chocolate was recommended by all the algorithms. The coefficients of this variable were positive and large in all the case and thus it is said to have a strong positive effect on the candy rankings. It is a must-have feature in the candy – the new label of candies from Zooplus must contain chocolate.

The feature peanutyalmondy is the second highest recommended feature. It is suggested by few of the most sophisticated algorithms applied in this exercise – Random Forest, Penalised Regression, and Boruta. Moreover, seven candies in the top 10 rankings in the candy dataset contain peanuts, peanut butter or almonds. Therefore, it is a significant ingredient to be included in a candy.

The price point of the candy also plays an important role in candy selection. While penalised models have established an inverse relationship between rankings and prices, on the whole, the recommended unit price of the new line of candies should be at 65^{th} percentile (or lesser) compared to the other candies in the given dataset.

The next most recommended attribute is sugarpercent. Sugar is the base ingredient of a confection, and a highly ranked candy from the dataset contains sugar in the 55 to 60 percentile range. Therefore, this should also be the preferred sugar level for the new brand of Zooplus' candies.

The attribute fruity is also a recommended characteristic for a best-selling candy. However, it can be confused to the presence of a fruit flavour. The graphical analysis with box plots, descriptive analysis and the correlation matrix in this study suggest that the absence of a fruit flavour in the candy is a good indicator of higher ranking. The analysis also states that the presence of chocolate and fruit-flavour are inversely related and since chocolate is a must have ingredient in a potential high-ranking candy, that candy should not be fruit-flavour. Besides, the top 20 candies in the dataset had only two fruit flavoured candies (1%).

A chocolaty candy with crisped rice, wafer or cookie component – how does that sound? Delectable, right? That is exactly what the analysis suggested too. The presence of crisped rice, wafers, or a cookie component in the candy should drive the customer sentiment into selecting it. However, data analysis also suggests that a high-ranking candy contains either of the two- peanut-almond or crispy rice, cookie contents.

Last but not the least, a soft and chewy 'easy on the jaws' candy should delight the customer's taste-buds.

Thus, it can be concluded that a chewy chocolaty, nutty or crispy candy that is not fruit flavoured, is moderately sweet and economically priced would drive the overall purchase tendency of the customer and thus, shall be incorporated in Zooplus' new candy offering.