Product review analysis

ML - Penalized logistic regression modeling

Hampus Nordholm

2024-09-22

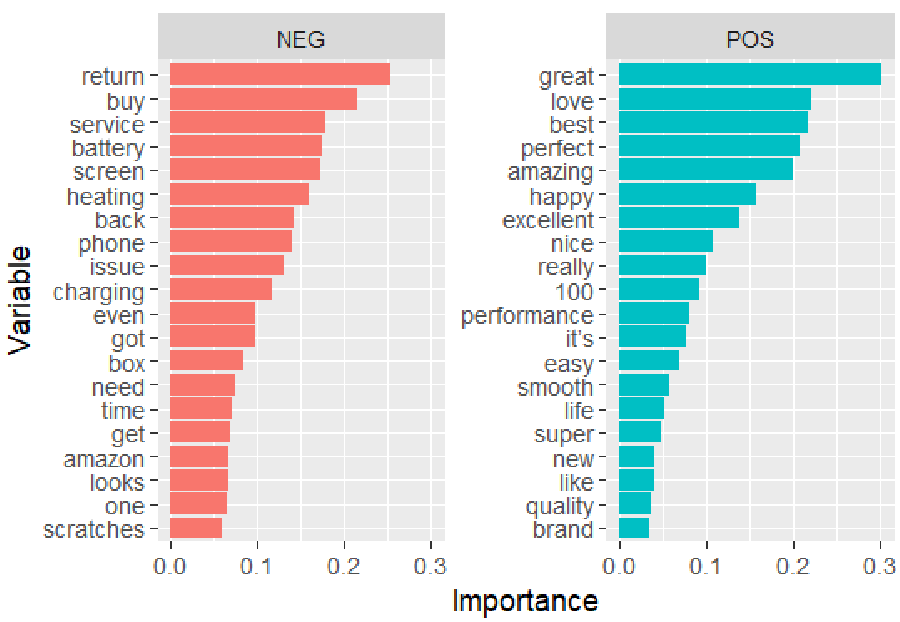
Intro

Analyzing iPhone reviews to predict whether a review description corresponds to a five-star rating or not. For classification, a penalized logistic regression model (LASSO) was used to identify which words in the review descriptions influence a five-star rating and which words that pushes agaianst a lower rating. Additionally, exploratory data analysis (EDA) was conducted prior to ML-modeling to understand and explore patterns within the dataset.

Solution summary

The final logistic model achieved an accuracy of 0.689 and a ROC AUC of 0.75 after tuning. The model revealed several important words that influence whether a review receives a five-star rating. The visualization below highlights the 20 most significant words that either support or detract from achieving a five-star rating.

Notably, words like "battery," "heating," and "charging" are associated with lower ratings, indicating areas where improvements can be made to improve the product's performance and customer satisfaction.



Core syntax for analysis

```r
# LIBRARIES --

#Data analysis
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(skimr)
library(tidytext)

# Machine learning --
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.2.0 --
## v broom        1.0.6      v rsample      1.2.1
## v dials        1.2.1      v tune         1.2.1
## v infer        1.0.7      v workflows    1.1.4
## v modeldata    1.4.0      v workflowsets 1.1.0
## v parsnip      1.2.1      v yardstick    1.3.1
## v recipes      1.0.10
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```r
library(textrecipes)
library(vip)
```

```
##
## Attaching package: 'vip'
##
## The following object is masked from 'package:utils':
##
##     vi
```

```r
# READ DATA --

iphone_tbl <- read_csv("iphone.csv")
```

```
## Rows: 3062 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): productAsin, country, date, reviewTitle, reviewDescription, reviewU...
## dbl (1): ratingScore
## lgl (1): isVerified
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# DATA EXAMINATION --

iphone_tbl %>% skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	3062
Number of columns	11
Column type frequency:	
character	9
logical	1
numeric	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
productAsin	0	1.00	10	10	0	7	0
country	0	1.00	5	20	0	7	0
date	0	1.00	10	10	0	789	0
reviewTitle	0	1.00	1	150	0	2018	0
reviewDescription	86	0.97	1	3885	0	2297	0
reviewUrl	16	0.99	102	108	0	2460	0
reviewedIn	0	1.00	31	57	0	1255	0
variant	0	1.00	22	58	0	86	0
variantAsin	0	1.00	10	10	0	99	0

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
isVerified	0	1	0.93	TRU: 2850, FAL: 212

Variable type: numeric

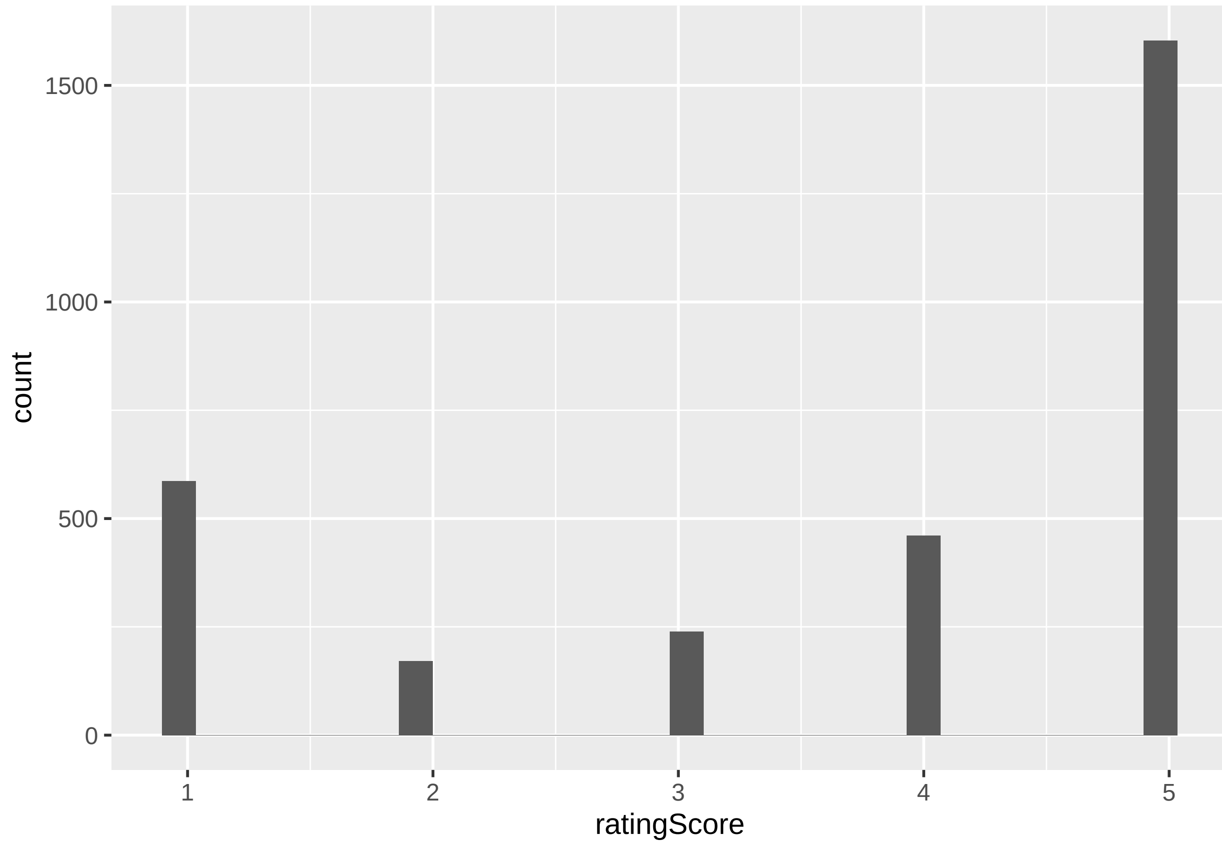
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ratingScore	0	1	3.76	1.58	1	3	5	5	5	

```r
iphone_tbl %>% glimpse()
```

```
## Rows: 3,062
## Columns: 11
## $ productAsin       <chr> "B09G9BL5CP", "B09G9BL5CP", "B09G9BL5CP", "B09G9BL5C~
## $ country           <chr> "India", "India", "India", "India", "India", "India"~
## $ date              <chr> "11-08-2024", "16-08-2024", "14-05-2024", "24-06-202~
## $ isVerified        <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ ratingScore       <dbl> 4, 5, 4, 5, 5, 5, 5, 5, 4, 5, 5, 5, 5, 5, 4, 5, 3, 5~
## $ reviewTitle       <chr> "No charger", "iPhone 13 256GB", "Flip camera option~
## $ reviewDescription <chr> "Every thing is good about iPhones, there's nothing ~
## $ reviewUrl         <chr> "https://www.amazon.in/gp/customer-reviews/R345SEIPU~
## $ reviewedIn        <chr> "Reviewed in India on 11 August 2024", "Reviewed in ~
## $ variant           <chr> "Colour: MidnightSize: 256 GB", "Colour: MidnightSiz~
## $ variantAsin       <chr> "B09G9BQS98", "B09G9BQS98", "B09G9BQS98", "B09G9BQS9~
```

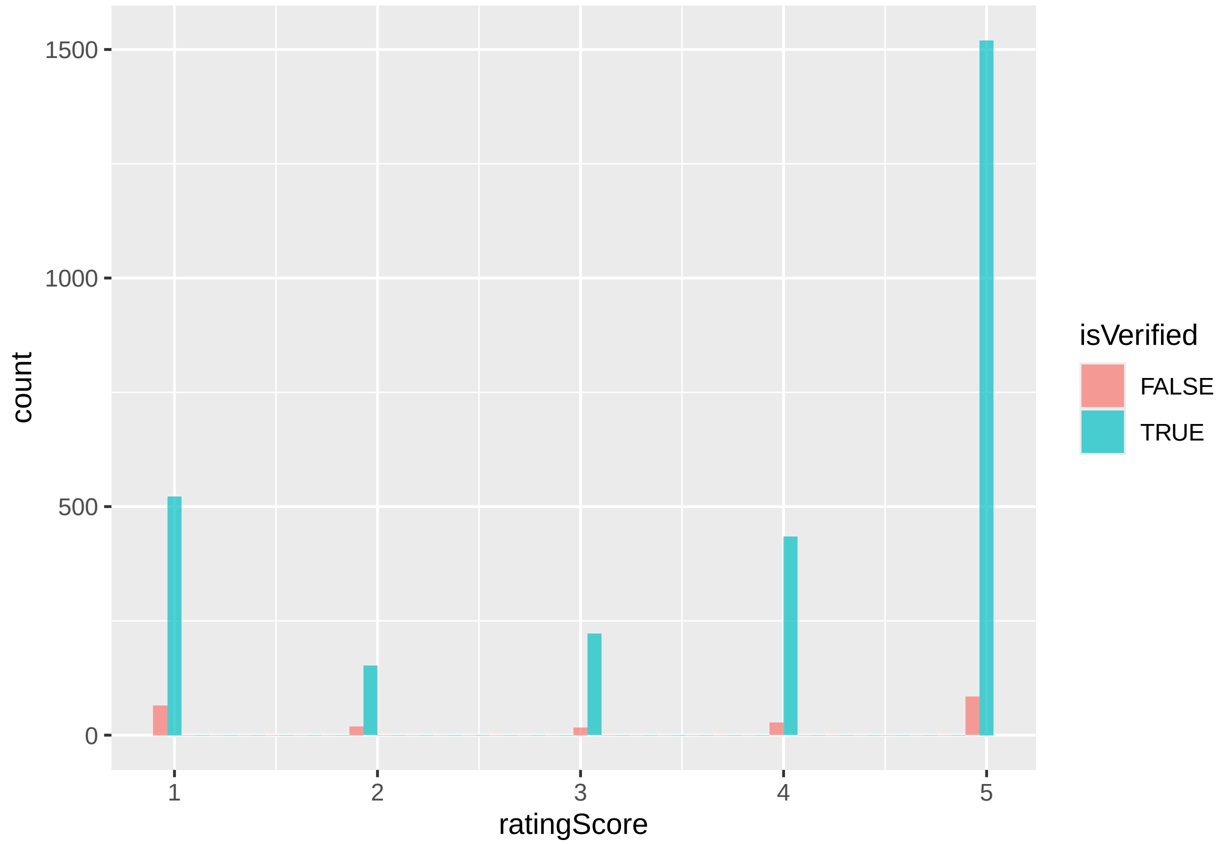
```r
# EXPLORATORY DATA ANALYSIS -- (EDA)

# Ratingscore distribution --

iphone_tbl %>%
  ggplot(aes(ratingScore))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



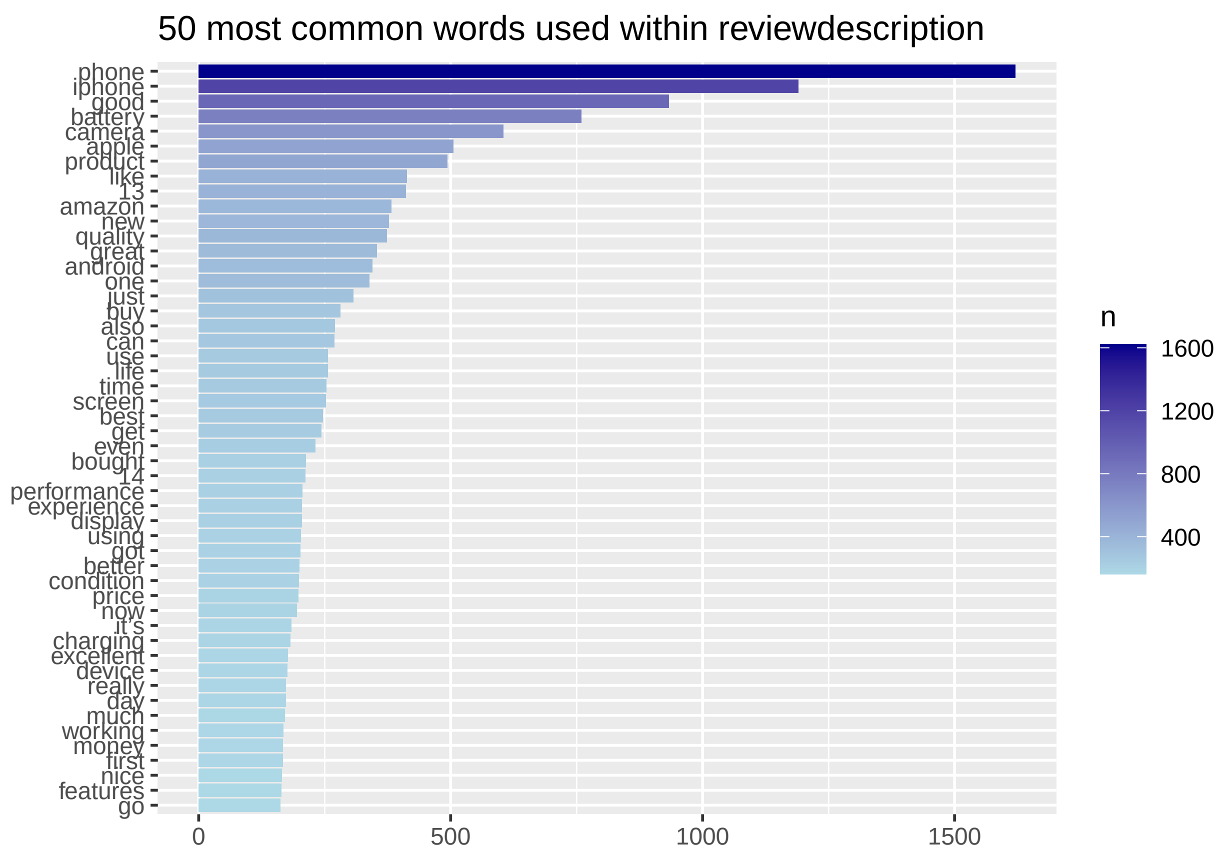

```
iphone_tbl %>%
  ggplot(aes(ratingScore,fill=isVerified))+
  geom_histogram(alpha=0.7,position="Dodge")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
iphone_tbl %>%
  unnest_tokens(word,reviewDescription) %>%
  anti_join(get_stopwords()) %>%
  count(word,sort=TRUE) %>%
  slice_max(n,n=50) %>%
  ggplot(aes(n,fct_reorder(word,n),fill=n))+
  geom_col()+
  scale_fill_gradient(low="lightblue",high="darkblue")+
  labs(title="50 most common words used within reviewdescription",
       y=NULL,x=NULL)
```

```
## Joining with `by = join_by(word)`
```




```r
# ** PENALIZED LOGISTIC REGRESSION MODEL ** (LASSO)

# Creating binary variable for 5 star rating TRUE/FALSE -- NA removal description --

iphone_tbl <- iphone_tbl %>%
  filter(!is.na(reviewDescription)) %>%
  mutate(toprated=if_else(ratingScore==5,"TRUE","FALSE"))

iphone_tbl %>%
  count(toprated)
```

```
## # A tibble: 2 x 2
##   toprated     n
##   <chr>    <int>
## 1 FALSE     1423
## 2 TRUE      1553
```

```r
# ML train and testing split **

set.seed(123)
iphone_split <- initial_split(data=iphone_tbl,strata=toprated)
iphone_training <- training(iphone_split)
iphone_testing <- testing(iphone_split)
```

```r
# Model recipe --

iphone_rec <- recipe(toprated ~ reviewDescription,data=iphone_training) %>%
  step_tokenize(reviewDescription) %>%
  step_stopwords(reviewDescription) %>%
  step_tokenfilter(reviewDescription,max_tokens=100) %>%
  step_tfidf(reviewDescription) %>%
  step_normalize(all_predictors())
```

```r
# Penalized logistic (lasso) model spec --

lasso_spec <- logistic_reg(penalty=tune(),mixture=1) %>%
  set_engine("glmnet")
```

```r
# Recipe and model spec into workflow --

lasso_wf <- workflow() %>%
  add_recipe (iphone_rec) %>%
  add_model (lasso_spec)

lasso_wf
```

```
## == Workflow ====================================================================
## Preprocessor: Recipe
## Model: logistic_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 5 Recipe Steps
##
## * step_tokenize()
## * step_stopwords()
## * step_tokenfilter()
## * step_tfidf()
## * step_normalize()
##
## -- Model -----------------------------------------------------------------------
## Logistic Regression Model Specification (classification)
##
## Main Arguments:
##   penalty = tune()
##   mixture = 1
##
## Computational engine: glmnet
```

```r
# Model tuning --

lambda_grid <- grid_regular(penalty(),levels=30)

# Bootstraps resampling --

set.seed(123)
iphone_folds <- bootstraps(iphone_training,strata=toprated)
```

```r
# Lasso grid --

set.seed(2020)
lasso_grid <- tune_grid(lasso_wf,
          resamples=iphone_folds,
          grid=lambda_grid)
```

```
## Warning: package 'stopwords' was built under R version 4.3.3

## Warning: package 'glmnet' was built under R version 4.3.3
```

```r
lasso_grid
```

```
## # Tuning results
## # Bootstrap sampling using stratification
## # A tibble: 25 x 4
##    splits             id          .metrics          .notes
##    <list>             <chr>       <list>            <list>
##  1 <split [2231/808]> Bootstrap01 <tibble [90 x 5]> <tibble [0 x 3]>
##  2 <split [2231/827]> Bootstrap02 <tibble [90 x 5]> <tibble [0 x 3]>
##  3 <split [2231/809]> Bootstrap03 <tibble [90 x 5]> <tibble [0 x 3]>
##  4 <split [2231/846]> Bootstrap04 <tibble [90 x 5]> <tibble [0 x 3]>
##  5 <split [2231/834]> Bootstrap05 <tibble [90 x 5]> <tibble [0 x 3]>
##  6 <split [2231/843]> Bootstrap06 <tibble [90 x 5]> <tibble [0 x 3]>
##  7 <split [2231/825]> Bootstrap07 <tibble [90 x 5]> <tibble [0 x 3]>
##  8 <split [2231/815]> Bootstrap08 <tibble [90 x 5]> <tibble [0 x 3]>
##  9 <split [2231/821]> Bootstrap09 <tibble [90 x 5]> <tibble [0 x 3]>
## 10 <split [2231/830]> Bootstrap10 <tibble [90 x 5]> <tibble [0 x 3]>
## # i 15 more rows
```

```r
lasso_grid %>%
  collect_metrics()
```

```
## # A tibble: 90 x 7
##     penalty .metric     .estimator  mean     n std_err .config
##       <dbl> <chr>       <chr>      <dbl> <int>   <dbl> <chr>
##  1 1    e-10 accuracy    binary     0.672    25 0.00299 Preprocessor1_Model01
##  2 1    e-10 brier_class binary     0.215    25 0.00114 Preprocessor1_Model01
##  3 1    e-10 roc_auc     binary     0.725    25 0.00235 Preprocessor1_Model01
##  4 2.21e-10 accuracy    binary     0.672    25 0.00299 Preprocessor1_Model02
##  5 2.21e-10 brier_class binary     0.215    25 0.00114 Preprocessor1_Model02
##  6 2.21e-10 roc_auc     binary     0.725    25 0.00235 Preprocessor1_Model02
##  7 4.89e-10 accuracy    binary     0.672    25 0.00299 Preprocessor1_Model03
##  8 4.89e-10 brier_class binary     0.215    25 0.00114 Preprocessor1_Model03
##  9 4.89e-10 roc_auc     binary     0.725    25 0.00235 Preprocessor1_Model03
## 10 1.08e- 9 accuracy    binary     0.672    25 0.00299 Preprocessor1_Model04
## # i 80 more rows
```

```r
# BEST PENALTY FROM LASSO GRID --

best_auc <- lasso_grid %>%
```

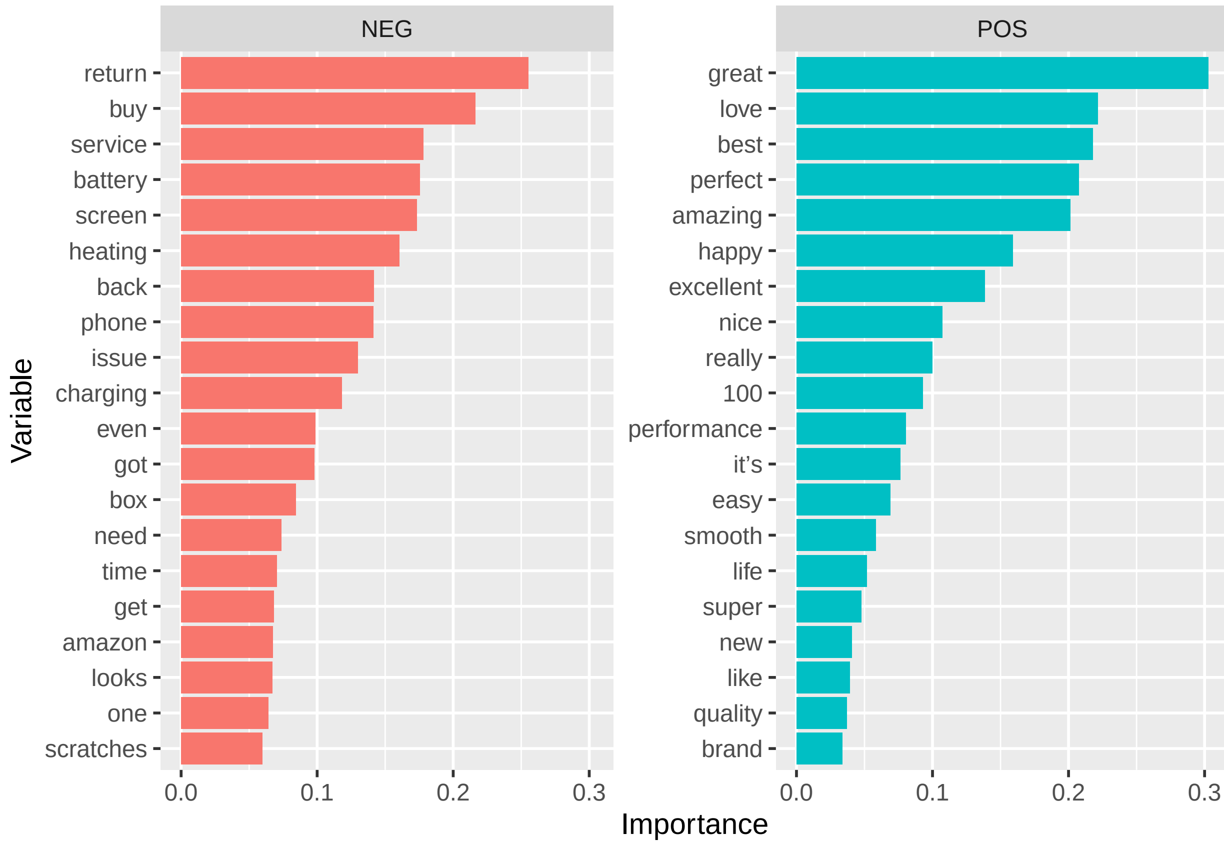
```r
  select_best(metric="roc_auc")

best_auc
```

```
## # A tibble: 1 x 2
##   penalty .config
##     <dbl> <chr>
## 1 0.00853 Preprocessor1_Model24
```

```r
# Final workflow --

final_lasso <- finalize_workflow(lasso_wf,best_auc)
```

```r
# Fit training data --

final_fit <- final_lasso %>%
  fit(iphone_training) %>%
 pull_workflow_fit()%>%
 vi(lambda=best_auc$penalty)
```

```r
# Logistic reg model word importance visual --

final_fit %>%
  group_by(Sign) %>%
  top_n(20,wt=abs(Importance)) %>%
  ungroup() %>%
  mutate(Importance=abs(Importance),
         Variable=str_remove(Variable,"tfidf_reviewDescription_"),
         Variable=fct_reorder(Variable,Importance)) %>%
  ggplot(aes(Importance,Variable,fill=Sign))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~Sign,scales="free_y")
```




```r
# -- Final logistic model evaluation on testing DATA --

iphone_final <- last_fit(final_lasso,iphone_split)

#Accuracy , ROC_AUC
iphone_final %>%
  collect_metrics()
```

```
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.689 Preprocessor1_Model1
## 2 roc_auc     binary         0.750 Preprocessor1_Model1
## 3 brier_class binary         0.202 Preprocessor1_Model1
```

```r
#Predictions
iphone_final %>%
  collect_predictions()
```

```
## # A tibble: 745 x 7
##    .pred_class .pred_FALSE .pred_TRUE id                .row toprated .config
##    <fct>             <dbl>      <dbl> <chr>            <int> <fct>    <chr>
##  1 TRUE             0.329       0.671 train/test split     9 FALSE    Preproces~
##  2 TRUE             0.0731      0.927 train/test split    21 TRUE     Preproces~
##  3 TRUE             0.496       0.504 train/test split    25 TRUE     Preproces~
```

```
##   4 TRUE              0.436       0.564 train/test split    34 TRUE     Preproces~
##   5 FALSE             0.785       0.215 train/test split    38 FALSE    Preproces~
##   6 TRUE              0.349       0.651 train/test split    39 FALSE    Preproces~
##   7 TRUE              0.492       0.508 train/test split    41 TRUE     Preproces~
##   8 TRUE              0.228       0.772 train/test split    43 TRUE     Preproces~
##   9 FALSE             0.583       0.417 train/test split    51 FALSE    Preproces~
##  10 TRUE              0.404       0.596 train/test split    52 FALSE    Preproces~
## # i 735 more rows
```

```
#Confusion matrix
iphone_final %>%
  collect_predictions() %>%
  conf_mat(truth=toprated,estimate=.pred_class)
```

```
##             Truth
## Prediction FALSE TRUE
##      FALSE   215   91
##      TRUE    141  298
```