STAT 412 – FINAL PROJECT

PREDICTING RECOMMENDATIONS ON NEW YORK TIMES COMMENTS

PRESENTED BY JEREMY GUINTA

JUNE 7, 2018

NAVIGANT

TABLE OF CONTENTS

1. QUESTION AND THEORY

- Question – Predict the total number of recommendations that any comment made on a New York Time's article will receive.

- Theory – Well written, timely comments made on popular articles will receive more views and thus more recommendations.

- Challenge –
  - Determine popular articles.
  - Determine timely comments.
  - Determine well written comments.

DATA RELIED UPON

Article Data (train_article.csv)

- 3,445 observations organized by articleID
- Contains information regarding an article:
  - Who wrote it?
  - When it was published?
  - What it was about?

Comment Data (train_comments.csv)

- 665,396 observations organized by commentID and articleID
- Contains information regarding a comment made to an article
  - What was the contents of the Comment?
  - Who wrote it?
  - When it was written?

DATA RELIED UPON (CONTINUED)

Data Exclusions

- 240 articleIDs were found exclusively in the train_comments.csv data. These were removed.
- The train_comments.csv was reduced from 665,396 to 640,904.
- It is unknown why these articleIDs were missing.

Data Splitting

- The 640,904 observations from the train_comments.csv data were split into a 70%/30% train/validation set for feature and model development.
- 70% (448,554 observations)
  - Feature development
  - Model training and tuning
- 30% (192,350 observations)
  - Final Mean Absolute Error (MAE) validation before Kaggle submission

FEATURE BUILDING

The contents of an article can be used to determine how popular the article is. More popular articles should get more views, more comments, and depending on how the comment is written more recommendations.

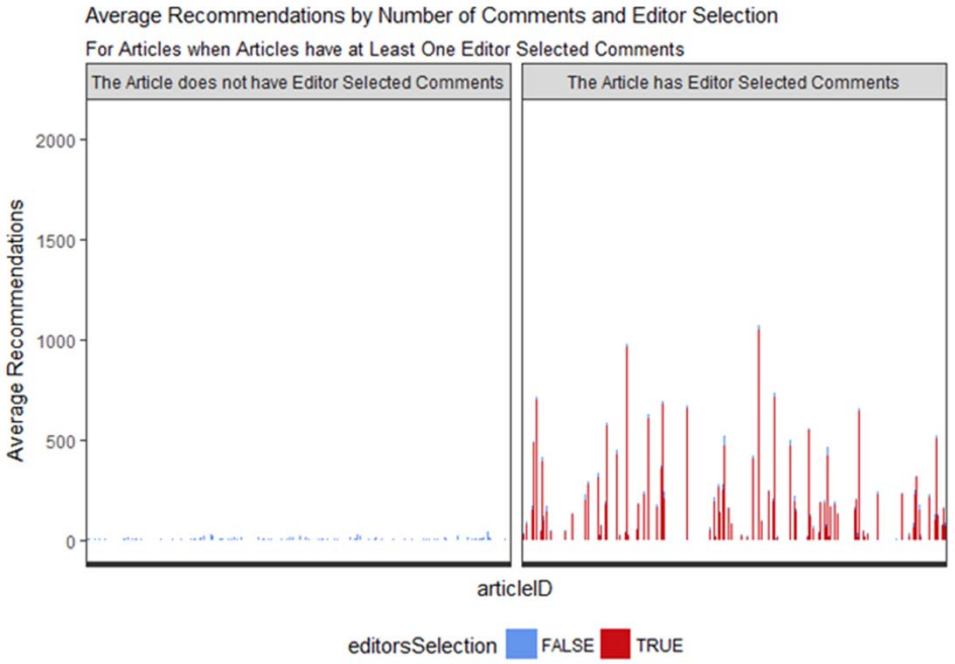
- **Challenge 1 – Popular Articles**
  - Editor Selection (existing variable)
  - Key Word and Key Work Rank
  - Topic Analysis
  - Article Time of Day / Day of Week

- **Challenge 2 – Timely Comments**
  - Comment position
  - Comment timing

- **Challenge 3 – Well Written Comments**
  - Comment sentiment
  - Comment reading grade level

FEATURE BUILDING (AN ASIDE) – EDITOR SELECTION

- This graph shows the average number of recommendations for a comment divided by on whether or not the article had editor selected comments.
- As the **red** very clearly shows, comments that are selected by the editor perform much better than other comments.
- However, it is unclear of the order of operations: Do high scoring comments become editor selected or is a comment selected by the editor, and then people start recommending it?
- **Forward looking prediction on a comment with this variable may not be proper.**



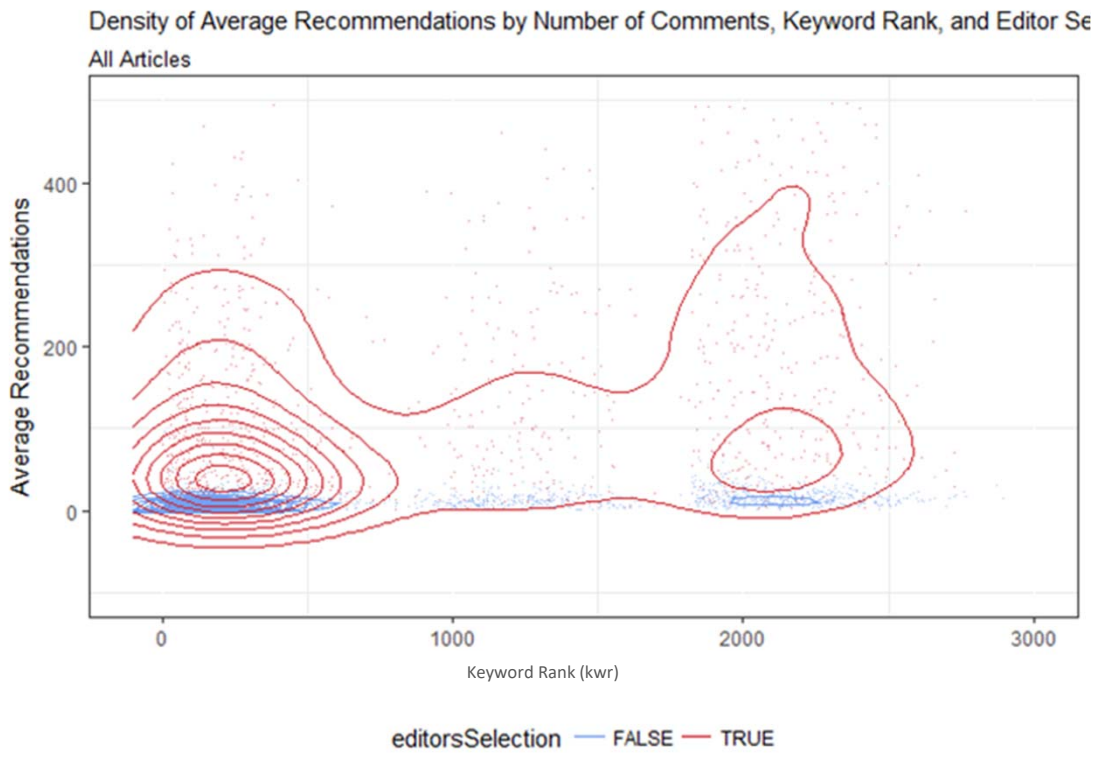
FEATURE BUILDING (CHALLENGE 1 – POPULAR ARTICLES)

Key Word Ranking – (a.k.a., kwr, minkwr, kw)

- This is my measure of popularity of an article.
- It is derived from the keywords field in train_article.csv data
- Keywords that appear more often get the lower ranks (e.g., Trump, Donald J is the most common key word. It gets a score of 1).
- Features:
  - kwr – Composite rank of all key words on an article. The higher the better (numeric)
  - minkwr – Lowest ranking key word by article (numeric)
  - kw1,2,3 – Key word ranked 1st, 2nd, and 3rd (categorical)
  - kwr1,2,3 – Key word ranking value for 1st, 2nd, 3rd ranking keyword (numeric)

- Other features:
  - Topic – Key word determined general topic (e.g., Politics)
  - Specific Topic – Key word determined specific topic (e.g., Donald Trump)

FEATURE BUILDING – KEYWORD RANK AND EDITOR SELECTION



- 2D Density plot of Average Recommendations against kwr.
- Editor selected comments have much higher recommendations than non-editor selected comments (at any keyword rank)
- Average recommendations is higher as the keyword rank increases.
- There are huge outliers in this data.

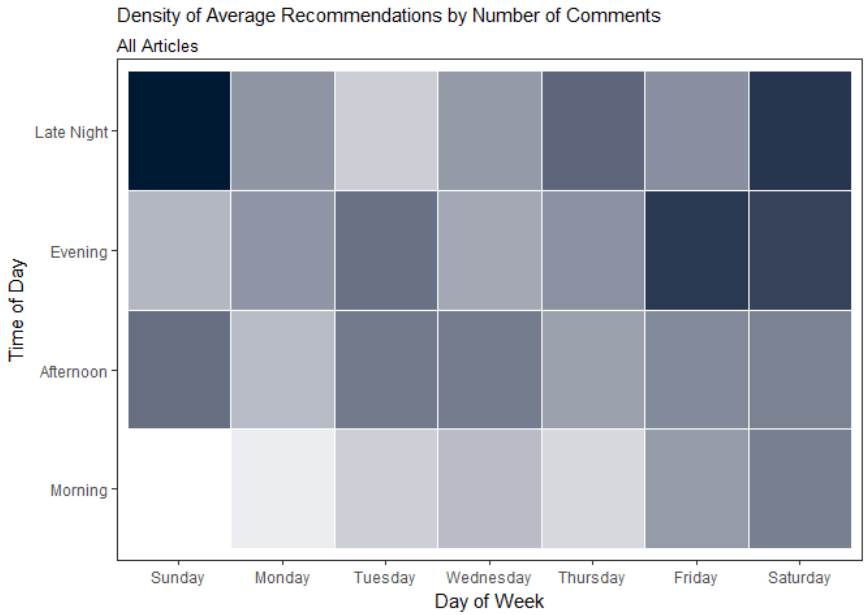
FEATURE BUILDING (CHALLENGE 2 – TIMELY COMMENTS)

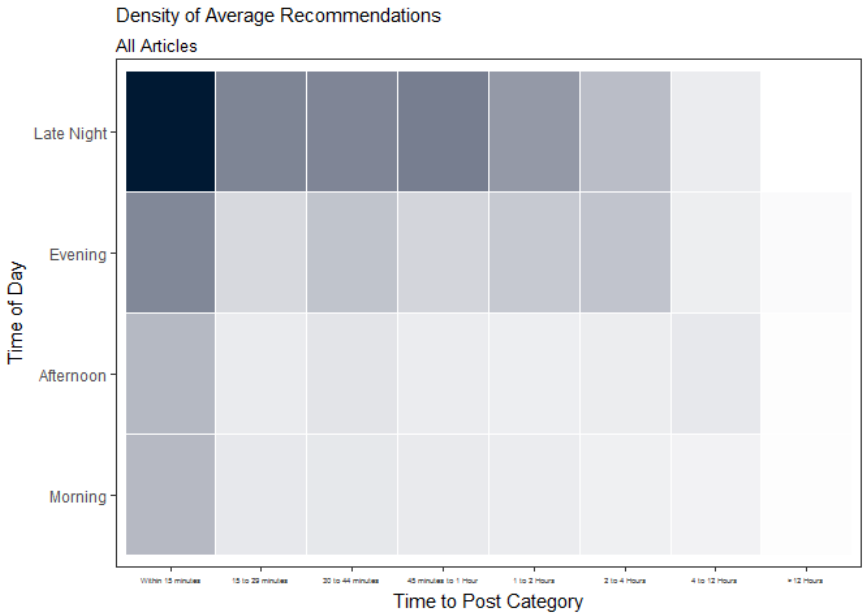
Timing and Position of the Article and Comment – (a.k.a., time_to_post, com_ord, dow, timeofday)

- These are my measures of timing of the comment in relation to the article.
  - time_to_post: Uses the differences between publish date and comment approved date to determine how long it took for the comment to post after the article was published.
  - com_ord: Uses the relative position of the comment to categorize the position of the comment (e.g., top ten comment)

- These are my measures of timing of the article.
  - dow – Day of the week that an article is published
  - timeofday – Time of day that an article is published

FEATURE BUILDING (CHALLENGE 2 – TIMELY COMMENTS)





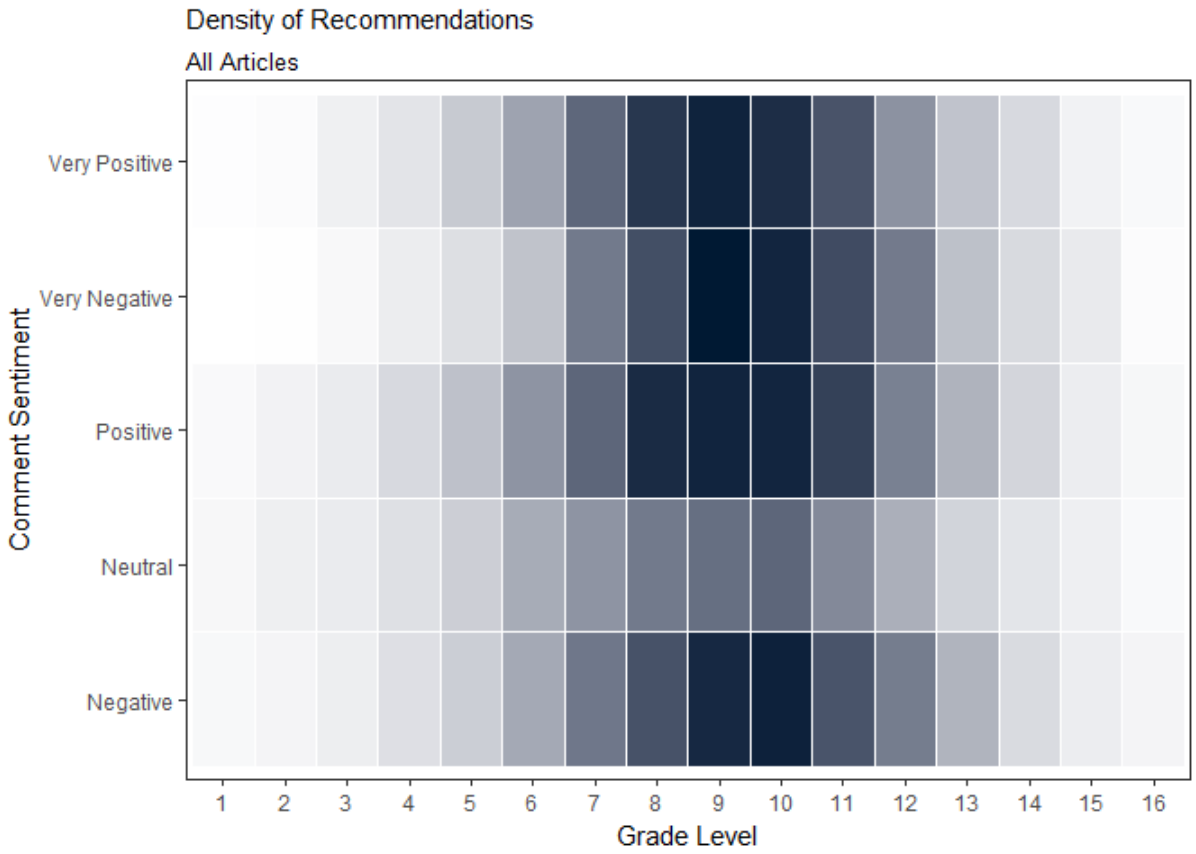
- These heatmaps demonstrate that average recommendations are influenced by the time of day, day of week, and time to post.
- Articles posted Sunday Morning have comments with fewer recommendations on average than Friday Evening or Sunday Late Night.
- As the time to post (from article publishing date) increases, comments get fewer and fewer recommendations.

FEATURE BUILDING (CHALLENGE 3 – WELL WRITTEN COMMENTS)

Sentiment and Writing Score of the Comment (i.e., com_sent, readFL, readCR)

- These are my measures of how well a comment is written.
  - com_sent: Sentiment analysis to score each comment. The variable is also categorized into a range of very positive to very negative.
  - readFL, readCR: Grade level of the comment based on the Flesch Kincaid and Coleman Liau scales.
- Other Features
  - snip_sent: Sentiment analysis to score each article snippet.

FEATURE BUILDING (CHALLENGE 3 – WELL WRITTEN COMMENTS)



- This heatmap demonstrates the spread between written grade level and comment sentiment.
- Comments written between the 7th and 12th grade levels get the majority of recommendations.
- There is little difference in number of recommendations based on the sentiment of the comment.
- Neutral comments are more spread out across all grade levels.

MODEL

Variable	Description
articleWordCount	Word count of the article
commentType	Categorical variable describing the type of comment
depth	The depth of the comment
editorsSelection	Categorical variable TRUE/FALSE
newDesk	The department that published the article
replyCount	The number of replies a comment receives
sectionName	Newspaper section
timespeople	Unknown 0/1 indicator
trusted	Unknown 0/1 indicator
com_ord	Comment order by article
com_pos_cat	Categorical variable of Comment order
time_to_post	Time in minutes to post comment from publish date and time
time_to_post_cat	Categorical variable of time to post
comment_length	String length of entire comment
readFR	Flesch Kincaid grade written level
readCL	Coleman Liau grade written level
com_sent	Comment sentiment
com_cat	Categorical breakdown of comment sentiment
kw1	First keyword after reorganization
kw2	Second keyword after reorganization
kw3	Third keyword after reorganization
kwr1	First keyword rank after reorganization
kwr2	Second keyword rank after reorganization
kwr3	Third keyword rank after reorganization
timeofday	Categorical variable for the time of day of the article being published
dow	Day of the week of the article being published
topic	General topic based on keywords
specific	Specific topic based on keywords
kwr	Composite keyword ranking for the article
minkwr	Lowest ranking keyword for the article
snip_sent	Article snippet sentiment
snip_cat	Categorical breakdown of article sentiment

MODEL – PARAMETER TUNING

- I used a combination of a Gradient Boosted Machine and a Random Forest via h2o (https://www.h2o.ai/)
- I used a random grid search design. This let the computer randomly bounce around a grid of pre-selected ranges of tuning parameters. I let the process run for eight hours for each model.
- I used all my features and modeled against the recommendations field.
- The “best” models had the following parameters:

GBM

Parameter	Value
ntrees	87
max_depth	20
min_rows	5
nbins_cats	64
nbins	20
stopping_metric	MAE
distribution	poisson
sample_rate	0.99
col_sample_rate	0.6
col_sample_rate_per_tree	0.9
learn_rate	0.09
learn_rate_annealing	1

RF

Parameter	Value
ntrees	65
max_depth	20
min_rows	10
nbins_cats	1536
nbins	20
stopping_metric	MAE
distribution	poisson
sample_rate	0.995
col_sample_rate_per_tree	0.8

MODEL – EVALUATION

- The “best” model was selected using the model with the lowest cross validated MAE on the training set.
- I validated each model run using the 30% hold out validation set to confirm I was not overfitting the model.

GBM

MAE Category	MAE
mean	13.18
cv 1	13.01
cv 2	13.20
cv 3	13.52
cv 4	13.09
cv 5	13.10

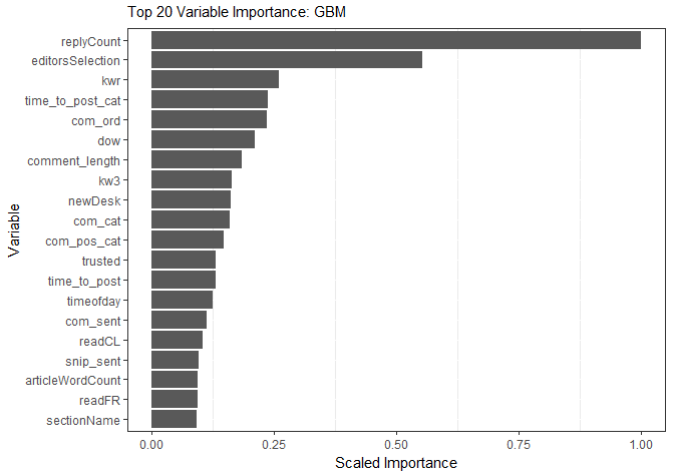
RF

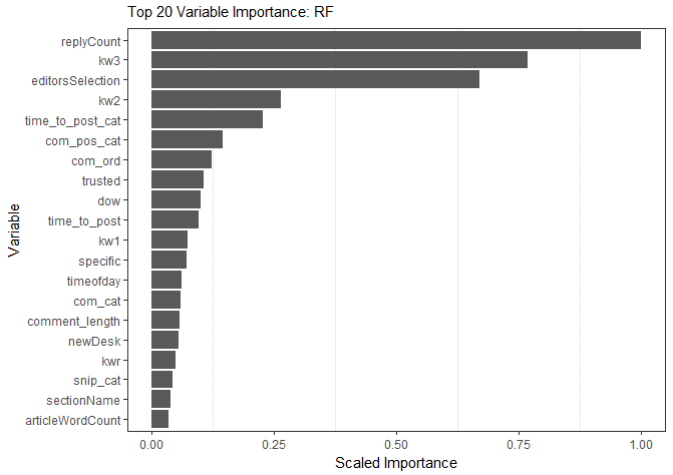
MAE Category	MAE
mean	14.48
cv 1	14.33
cv 2	14.47
cv 3	14.85
cv 4	14.30
cv 5	14.45

- The Random Forest always performed worst than the Gradient Boosted Machine model.
- The final validation MAE was 13.19 for the GBM model and 14.36 for the RF model

MODEL – VARIABLE IMPORTANCE

- My features performed reasonable well in the model, but were not as important as existing variables Reply Count and Editor Selection.
- In the GBM model, Keyword Rank ranked 3rd in importance. Followed by Time to Post, Comment Order, and Day of the Week.
- In the RF Model, Keyword Rank was not very important, but Time to Post, Comment Position, and Day of Week were important in the model.





MODEL – ERROR

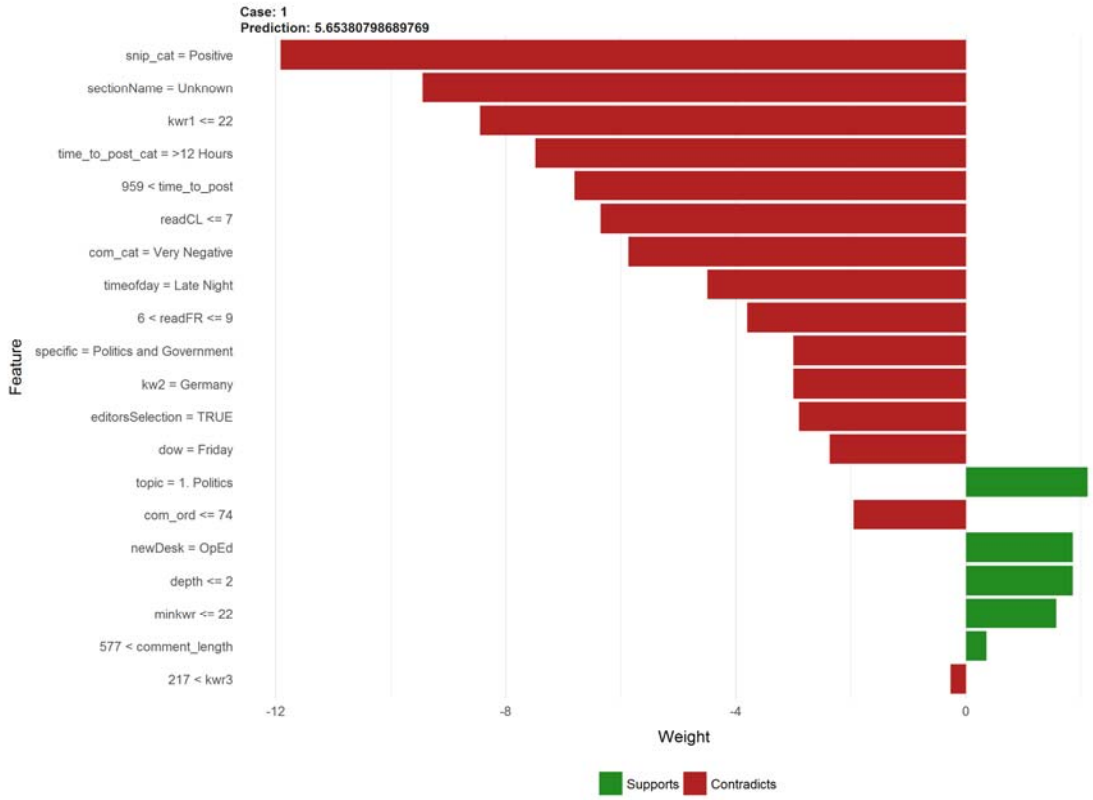
- What was driving the error? In the GBM model, 43.1% of the error was being driven by 3,132 comments and all of the 3,132 comments had 200 or more recommendations.

Recommendation Category	MAE	AE	# of Comments	% of Comments	% of Error
Less than 10	3.34	475,963	142,660	74.2%	18.8%
Between 10 and 19	8.34	195,172	23,388	12.2%	7.7%
Between 20 and 49	18.64	284,918	15,284	7.9%	11.2%
Between 50 and 99	44.33	224,908	5,073	2.6%	8.9%
Between 100 and 199	93.42	262,799	2,813	1.5%	10.4%
Greater than or equal to 200	349.31	1,094,051	3,132	1.6%	43.1%

- Unfortunately, I was unable to determine any specific pattern that would help explain the extreme divergence for this small group of comments.
- **Aside:** Lowering the absolute error in the >=200 category by just 10% would drop the Validation MAE from 13.19 to 12.62.

MODEL – LIME

- This Lime explainer plot is for a single record in the training data.
- The actual number of recommendations for this observation is 3 and the model predicted a value of 5.65.
- In this particular observation, snippet sentiment had the most weight (but this variable had very little importance in the model.)



CONCLUSION

- The two models performed well in terms of the training and validation MAE. Neither was overfit.
- The developed features demonstrate high relative importance in each of the models conducted.
- Both models performed poorly when predicting the number of recommendations for comments when the actual value of recommendations was extremely high.
- The final submitted Kaggle competition models were:
  - GBM model. This model had a public Kaggle score of 14.59.
  - A simple ensemble of the GBM and the RF models. This model had a public Kaggle score of 14.95.

Questions

NAVIGANT