



# That's Rotten

Predicting Good & Bad Movie Reviews from  
Rotten Tomatoes using Natural Language  
Processing & Classification Models

Tanya Shapiro

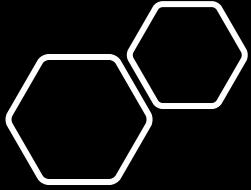
# What's The Problem?

The executives at a well-known movie production company are trying to determine which movies they've released are hits or flops.

Unfortunately, they don't have time to read and synthesize all the online buzz in their own time.

They've hired you to produce an algorithm that will them determine which movies received praise and which ones tanked according to critics.



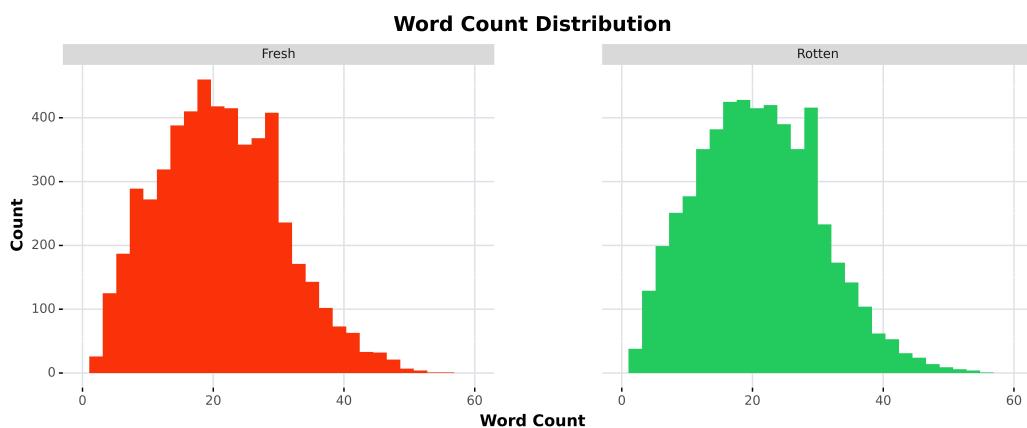


# Behind The Curtain

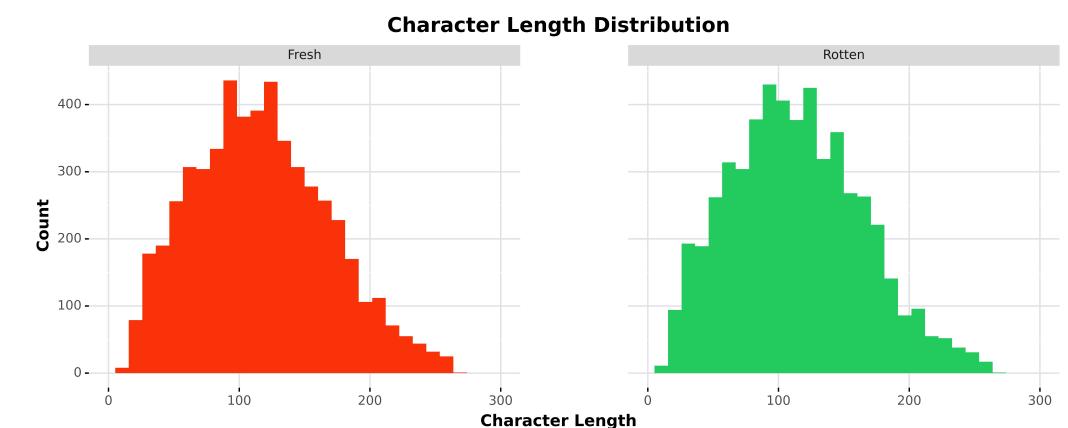
- **Source.** Data from Kaggle, 10K+ movie reviews
- **Balanced.** Even split between classes (50% Rotten, 50% Fresh)
- **Objective.** Build a classification model that beats the baseline accuracy (+50%)

# How verbose are the critics?

No discernable difference between “Rotten” and “Fresh” reviews when it comes to wordiness.



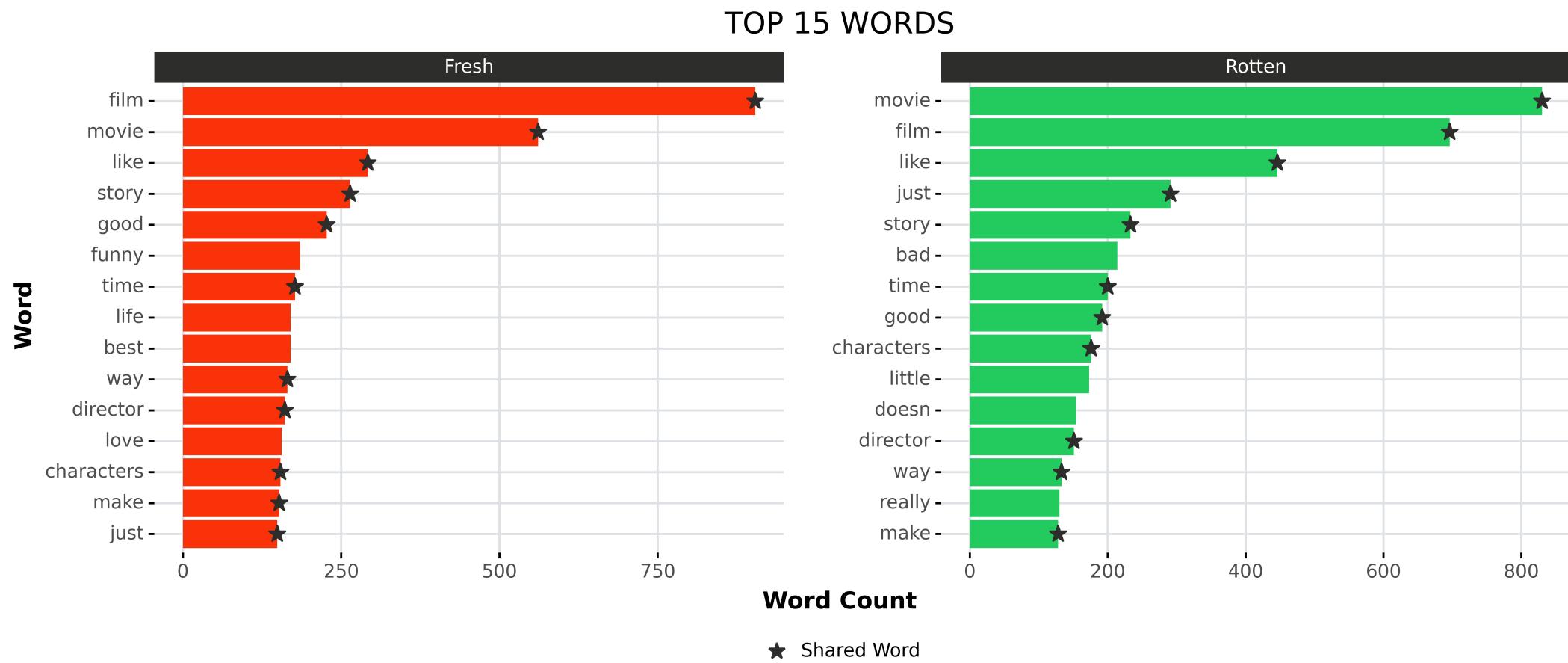
Avg ~21 words



Avg ~115 characters

# Common Words

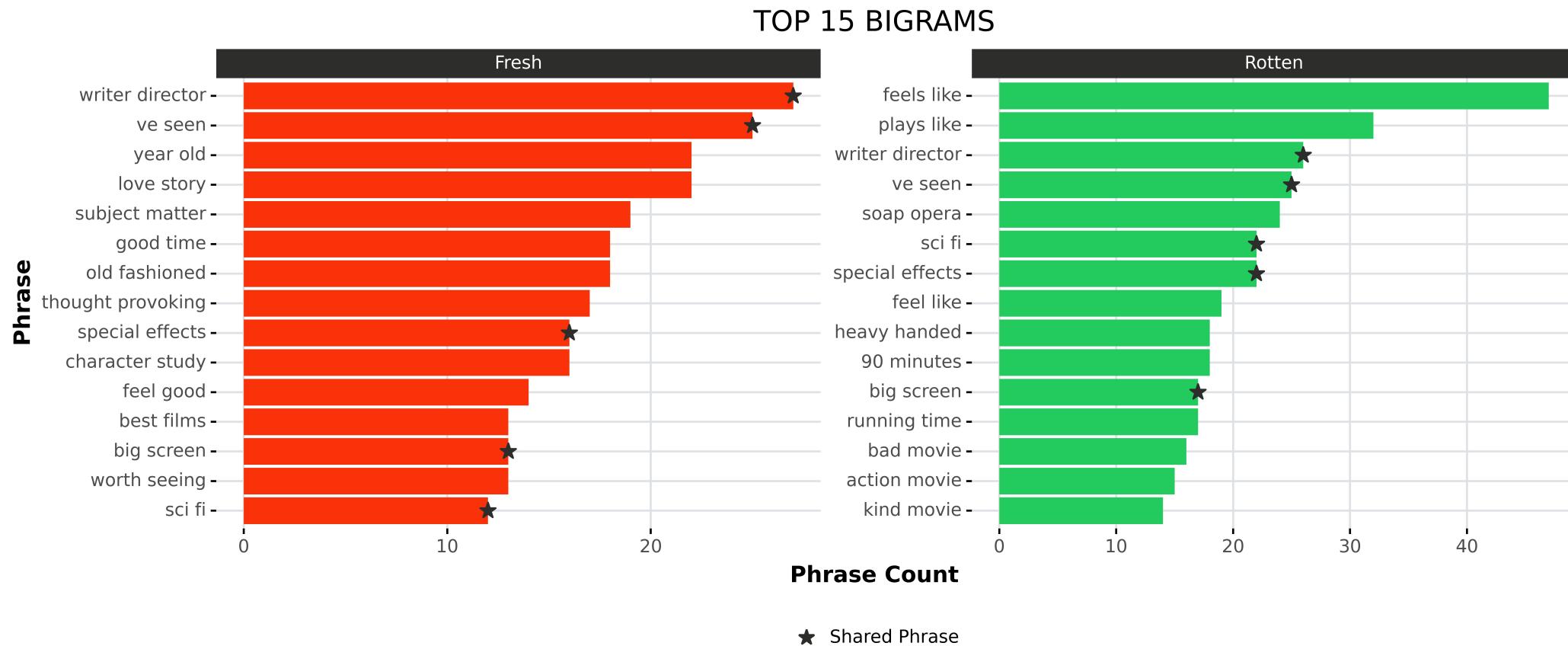
Not surprising, “Fresh” reviews commonly contain positive affinity words like “good”, “best” and “love” while “Rotten reviews” commonly contain negative affinity words like “bad” or “little.”



# Common Phrases

“Fresh” movies are a “good time” and “thought provoking.”

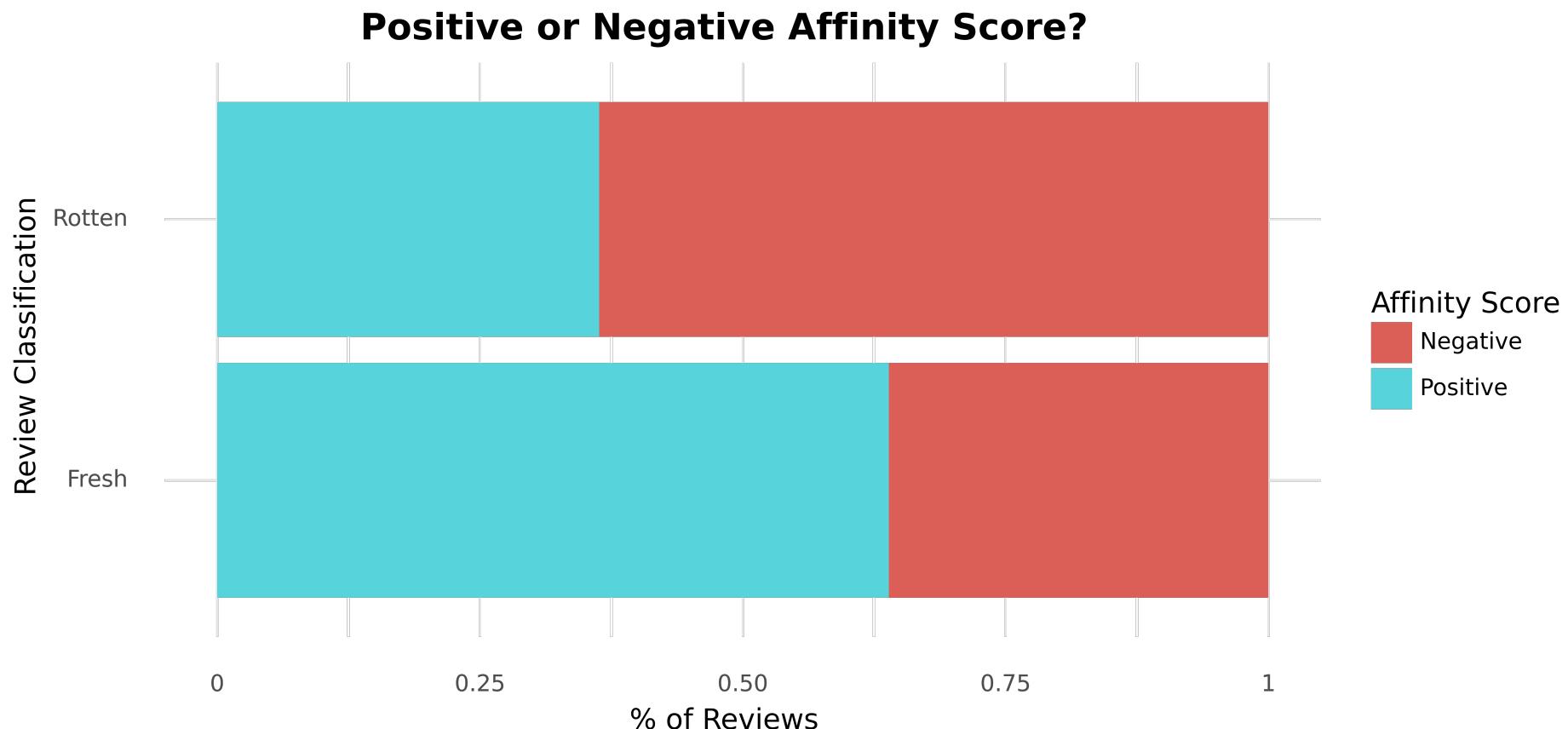
Both reviews contain language specific to film, e.g. “special effects”, “sci fi”, “writer director.”



# Sentiment Analysis

Majority of Rotten reviews generated negative affinity scores.

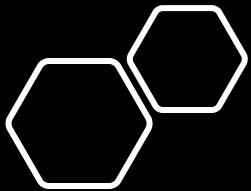
Majority of Fresh reviews generated positive affinity scores.



# And The Award for Best Model Goes To...

Model	Train Accuracy	Test Accuracy	TP	FP	TN	FN	Precision	Sensitivity	Specificity	Recall
SVM with TFV	93.6%	78.5%	645	189	611	155	79.8%	80.6%	76.4%	76.4%
Logistic Regression with TFV	94.1%	78.2%	630	179	621	170	78.5%	78.8%	77.6%	77.6%
MultiNomial NB with TFV	92.7%	77.4%	637	198	602	163	78.7%	79.6%	75.3%	75.3%
Logistic Regression with CV	93.5%	77.3%	623	187	613	177	77.6%	77.9%	76.6%	76.6%
KNN with TFV	82.6%	70.8%	557	224	576	243	70.3%	69.6%	72.0%	72.0%
Random Forest with CV	100.0%	70.2%	574	251	549	226	70.8%	71.8%	68.6%	68.6%
Random Forest with TFV	98.3%	69.6%	567	253	547	233	70.1%	70.9%	68.4%	68.4%

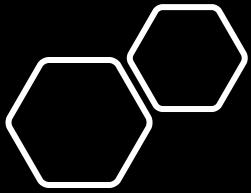
SVM model yields best accuracy results (78.5%).  
Better at predicting “fresh” reviews than “rotten reviews.”



# Plot Twists: Misclassification Analysis

*Rotten? It's  
Actually Fresh!*

- “It’s maker , Steven Spielberg , hasn't had so much fun in two decades , since he was **schlepping** Indiana Jones around the globe in search of a giant **misplaced** ashtray.”
- “Even during the climactic hourlong cricket match , **boredom** never takes hold”
- “Without [De Niro] , City By The sea would slip under the waves . He drags it back , single-handed.”



# Plot Twists: Misclassification Analysis

*Fresh? It's  
Actually Rotten!*

- “However stale the material , Lawrence's delivery remains **perfect** ; his **great** gift is that he can actually trick you into thinking some of this worn-out , pandering palaver is actually **funny** .”
- “Despite **terrific** special effects and funnier gags , harry potter and the chamber of secrets finds a way to make J . K . Rowling's **marvelous** series into a deadly bore.”

# Closing Credits: Notes & Recommendations

- **Improving the model** – more hyperparameter tuning with GridSearch to find best model parameters for SVM
- **Testing other models** – would a Neural Network classifier or XGBoost yield higher accuracy scores?
- **Increasing sample size** – collect more reviews to train the model, changing train/test split from 80/20 to 85/15 improved performance slightly.
- **Defining “Rotten”** – model is for binary-classification, but Rotten Tomatoes reviews are typically scaled (1-4 or %). Problem better suited for multi-class?
- **Sarcasm** – text classification analysis doesn’t seem to perform well on sarcasm or text with mixed sentiments (good and bad remarks)



# Epilogue: Applying The Model



APPLIED TO **SISKEL & EBERT**  
REVIEWS – 66% ACCURACY



DEVELOPED A **STREAMLIT APP**