"Who said that?" Building classifiers that predict which TV show character said a given line!

Christine Yang, Juliana Fakhoury, Duanchen (Dora) Liu, Jiaxuan (Isabell) Huang and Wenyu (Emma) Huang

Link to Github: www.github.com/yingyangle/whosaidthat

INTRODUCTION

Motivation: Understand which techniques are useful for authorship attribution on shorter utterances

Goal: Building classifiers that predict which character said a given line of dialogue

Methods:

- Using features
- Word embeddings

Idea: Try 3 powerful classifiers!

- Random Forests
- Logistic Regression
- Neural Networks

Hypothesis: Genre will affect accuracy

DATA

The Big Bang Theory [sitcom]

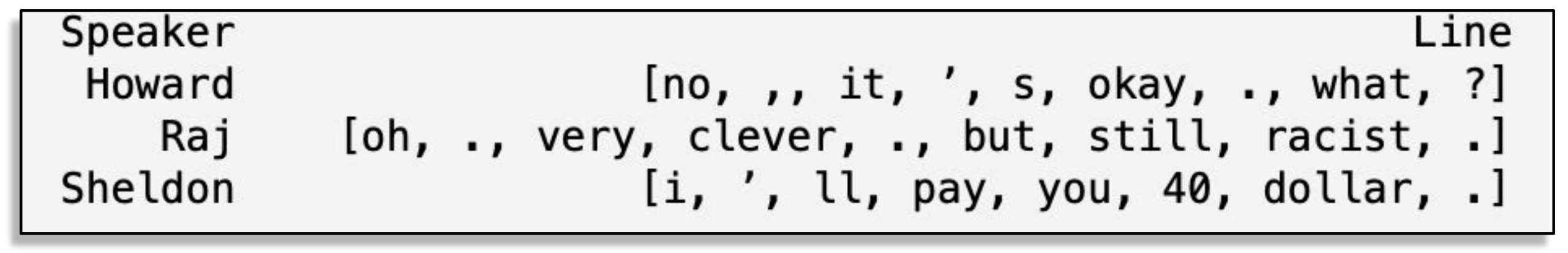
- scrape scripts from WordPress website
- split lines and label the speaker for each line
- 45,825 lines, 7 characters

The Simpsons [cartoon]

- found existing labeled scripts in .csv format
- 67,955 lines, 5 characters

Desperate Housewives [drama]

- convert .doc scripts to .csv
- standardize speaker names and formatting
- 18,437 lines, 4 characters

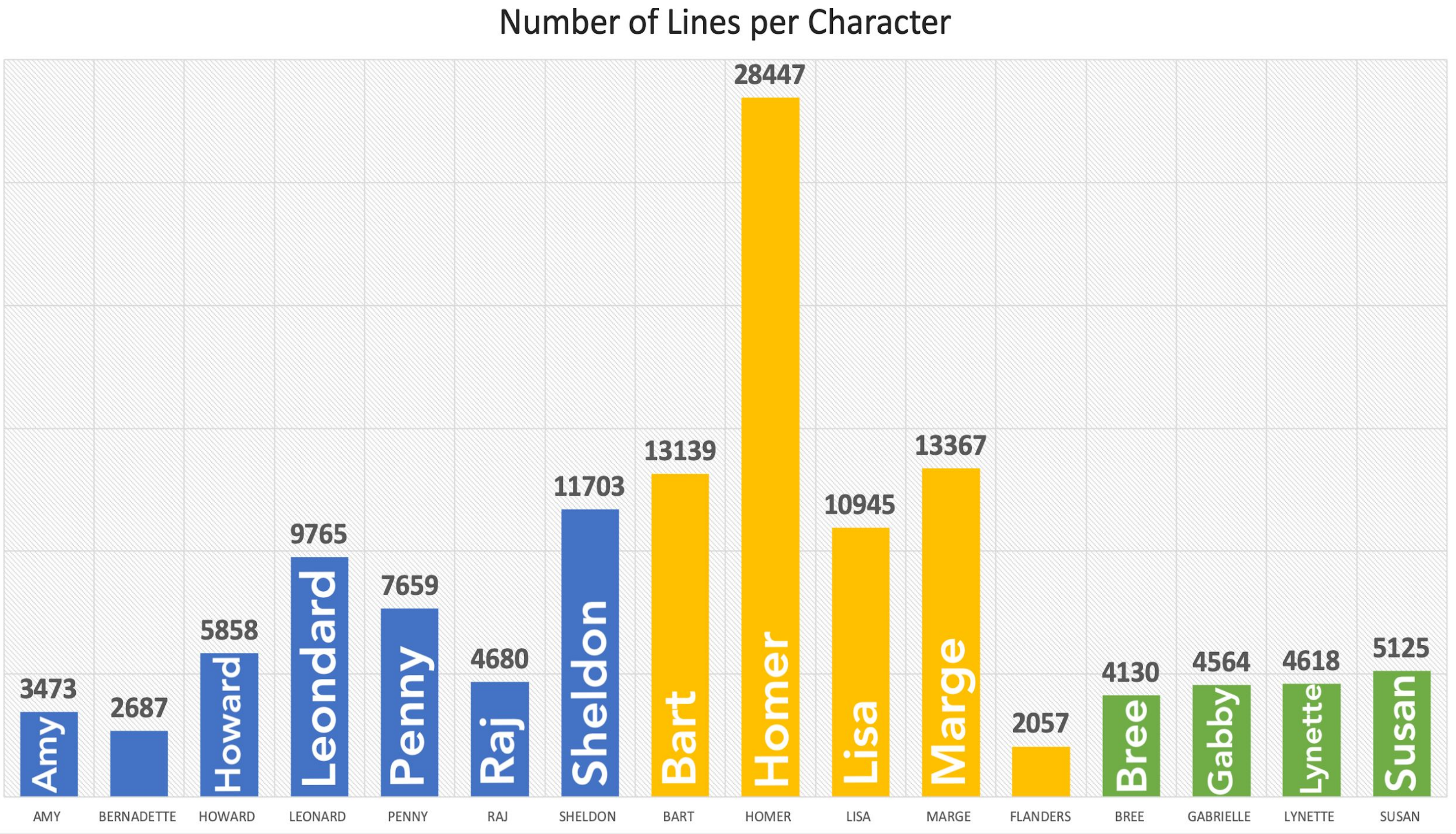


```
Speaker                                            Line
 Howard              [no, ,, it, ', s, okay, ., what, ?]
    Raj   [oh, ., very, clever, ., but, still, racist, .]
Sheldon               [i, ', ll, pay, you, 40, dollar, .]
```

Normalization & Preprocessing:

- Convert all letters to **lowercase**
- **Tokenize** and **lemmatize** words
- Convert number words into **digits**
- Test/Train → **20/80 split**

The Simpsons had the largest disparity in lines per character
Desperate Housewives had the most even dataset



METHODOLOGY

FEATURES: 2 APPROACHES

Manual Feature Selection

- **type-token** ratio, **punctuation** use
- utterance length, average **word length**
- **polarity** & **subjectivity** (textblob)
- # of stop words, **neologisms**, number words, **profanity** words
- # of words in utterance that are also in each character's 20 **most frequent words**

Word Embeddings

- get word vector embeddings from **word2vec**
- word vectors → **sentence vector**
- constructed embeddings using 2 corpora:
  - **Google News** corpus
  - show script (training data)

BEST
CALL
BEER
DOLLAR
NOTHING
JOB
MINUTE
WOO
MUST
WIFE
HOO
HEH
FINE
CAR
EVERY
WORRY
EH
OW
EAT
WORLD

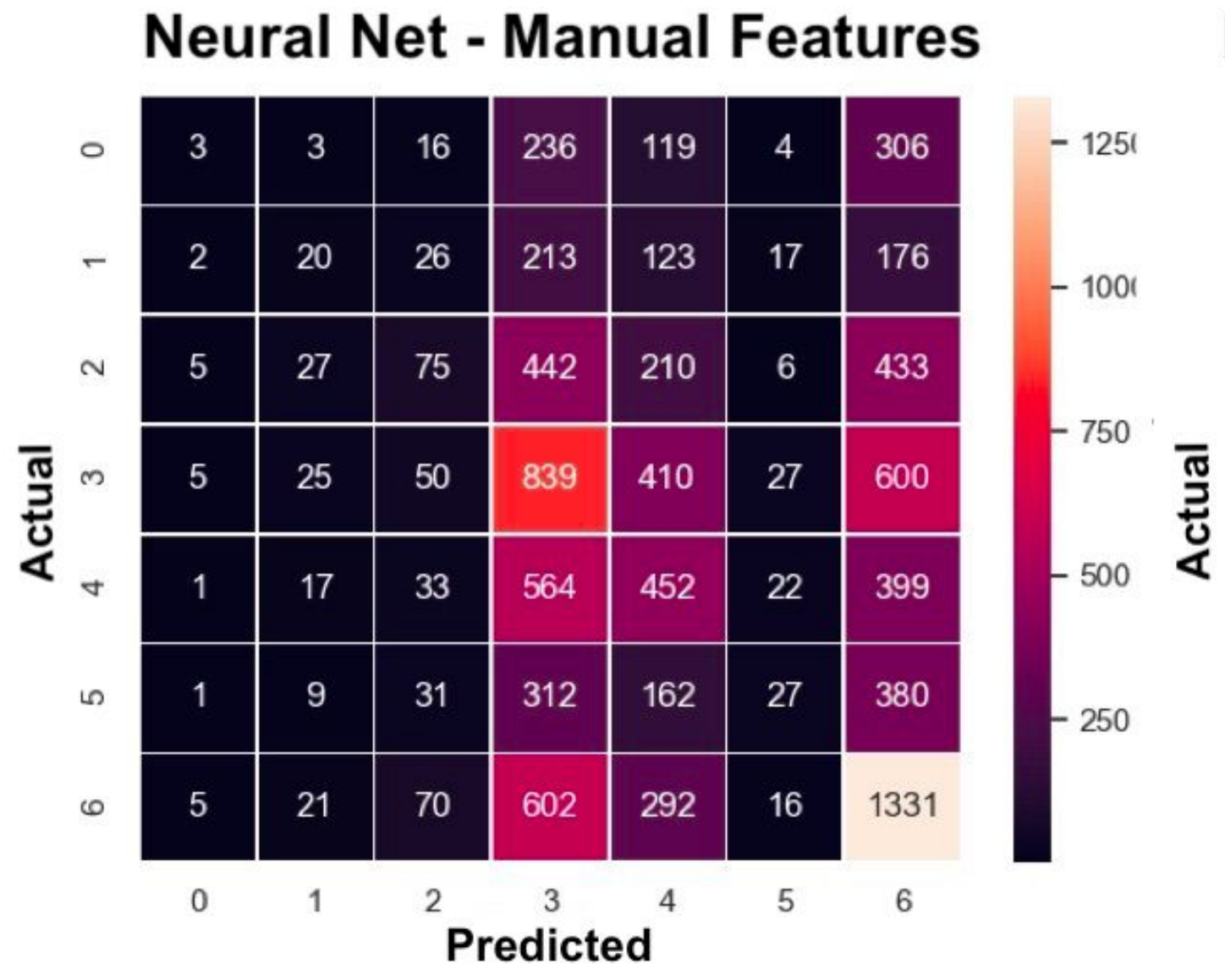
Capturing the 20 most frequent words for **Homer** (left) and **Flanders** (right)

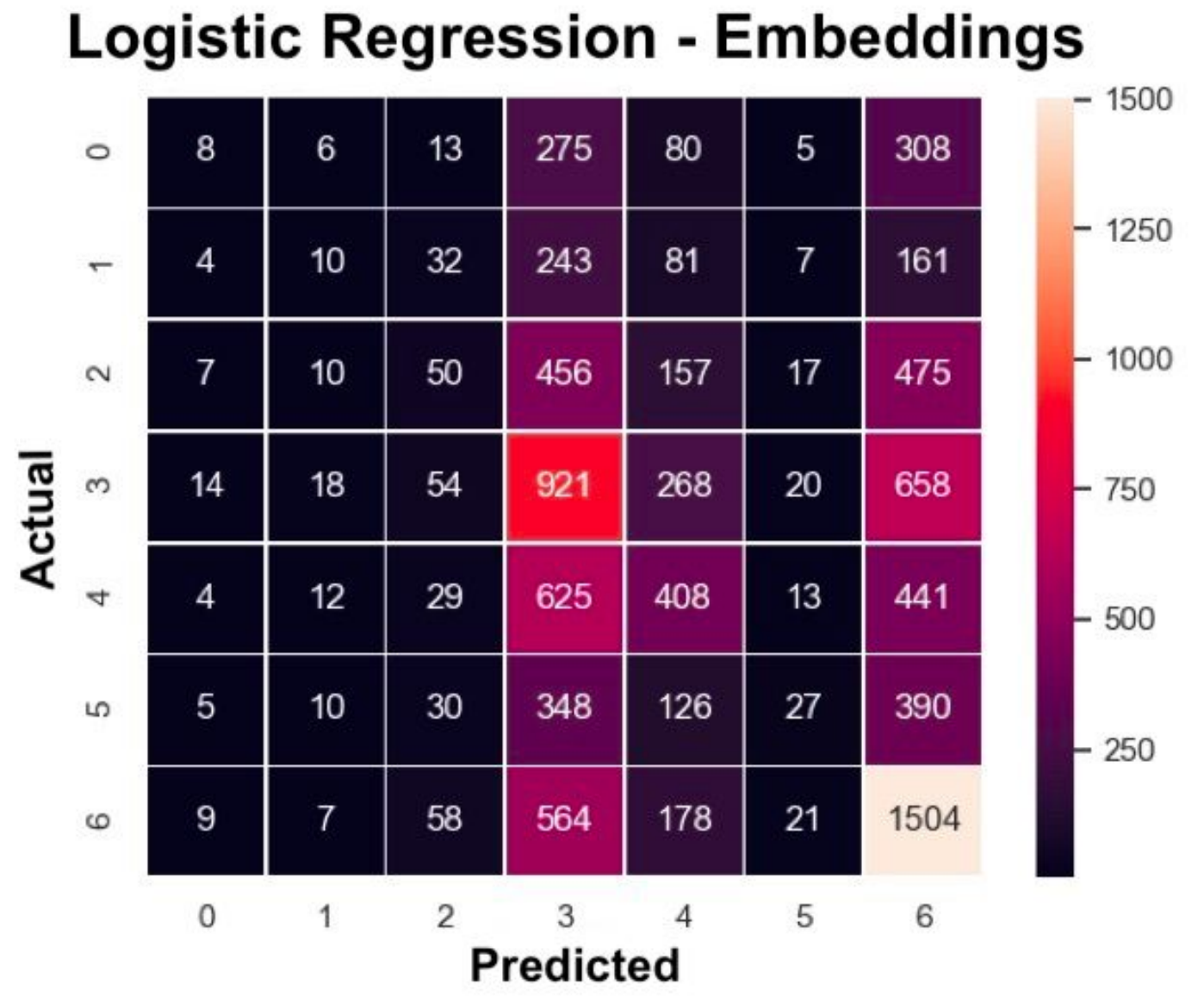
BIBLE
DIDDILY
REVEREND
CHURCH
NED
MAUDE
AH
NICE
YA
LORD
HI
HO
SIR
CALL
NEIGHBOR
FUN
DONE
TODD
WORD
ROD

RESULTS

Big Bang Theory

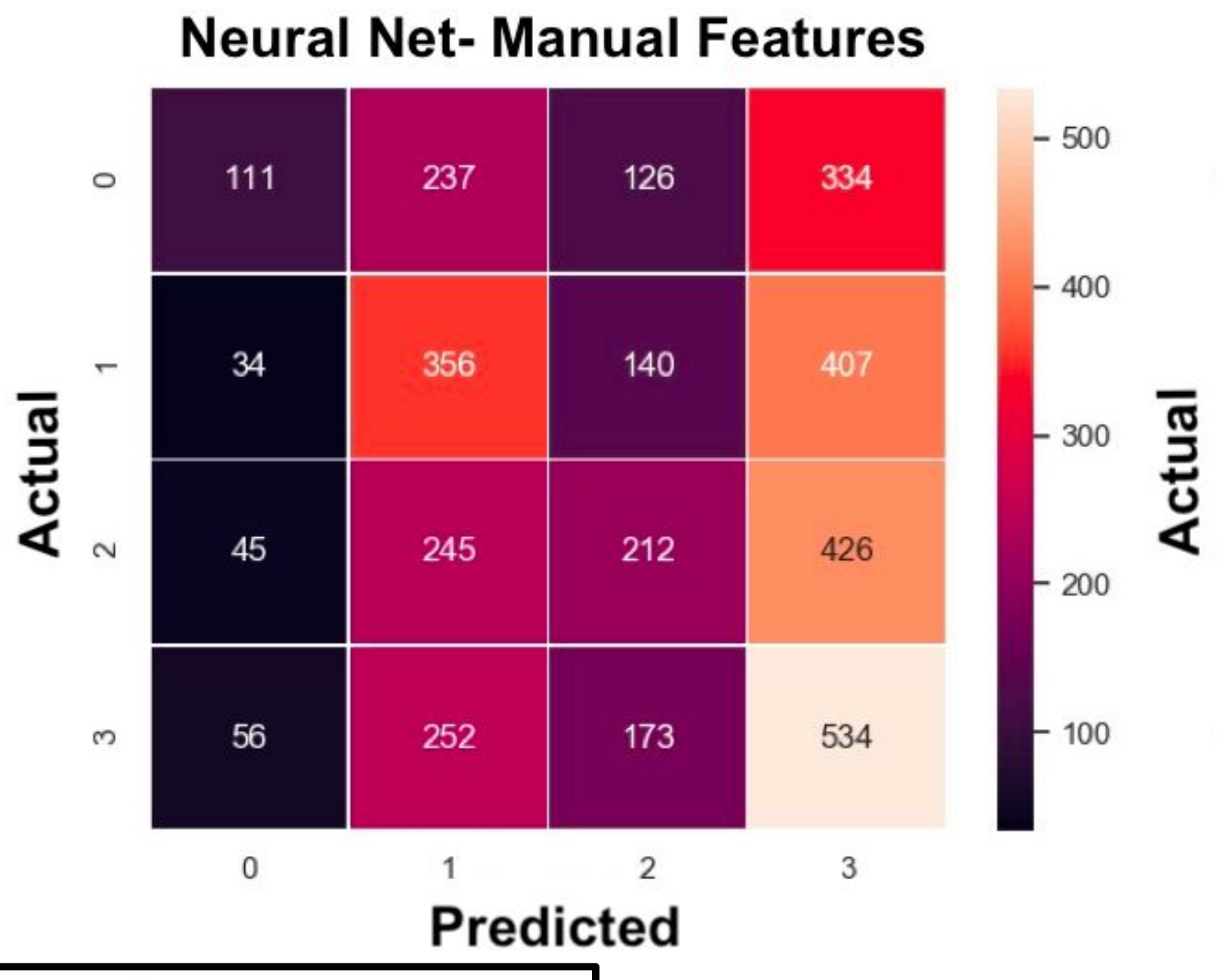
		Accuracy	Precision	Recall
	Random Baseline	0.14		
	Majority Baseline	0.26		
Manual Features	Random Forest	0.27	0.24	0.27
	Logistic Regression	0.29	0.28	0.29
	Neural Net	0.30	0.27	0.30
Embeddings	Random Forest	0.31	0.26	0.20
	Logistic Regression	0.32	0.25	0.21
	Neural Net	0.27	0.20	0.20

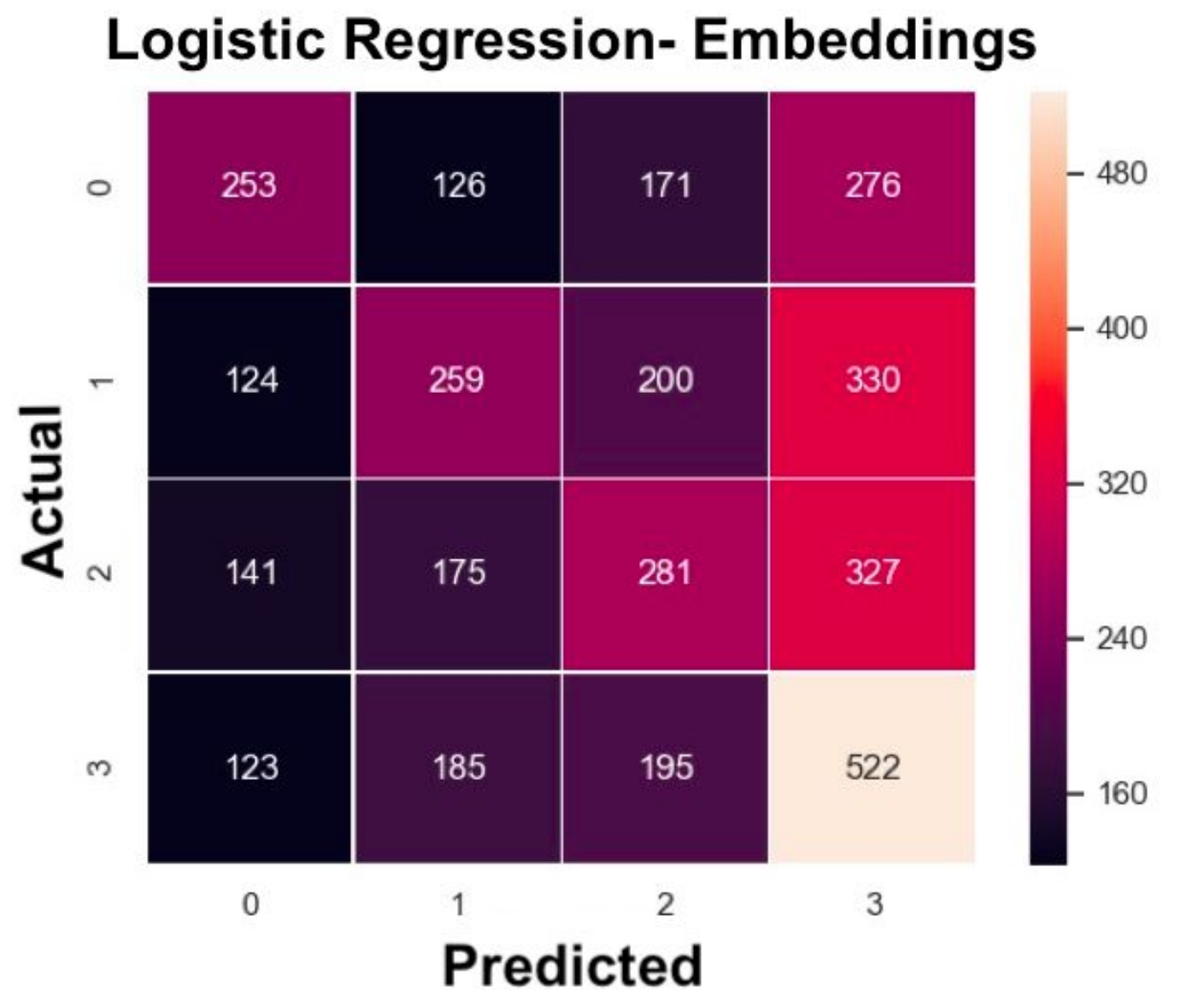




The Simpsons

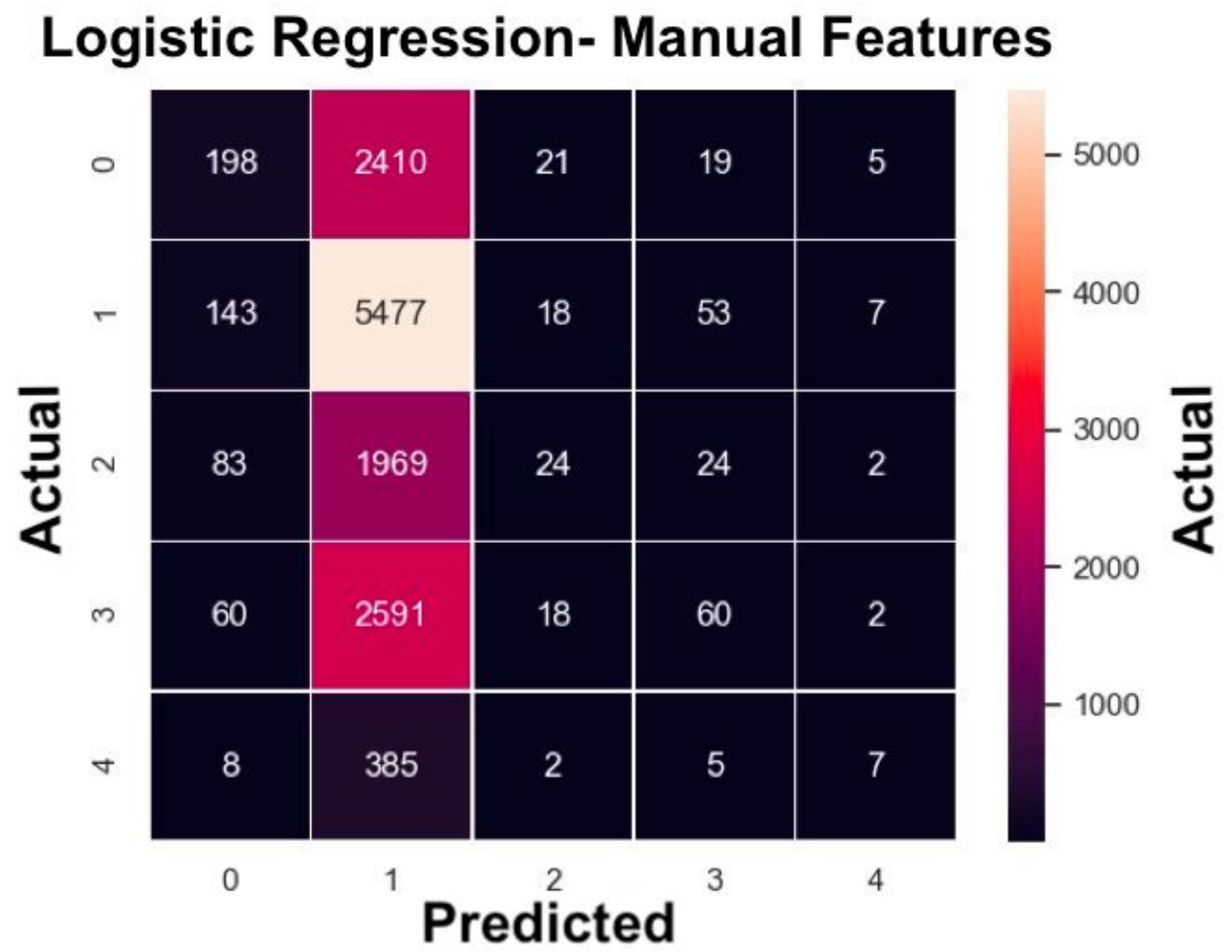
		Accuracy	Precision	Recall
	Random Baseline	0.20		
	Majority Baseline	0.42		
Manual Features	Random Forest	0.40	0.35	0.40
	Logistic Regression	0.42	0.39	0.42
	Neural Net	0.43	0.39	0.43
Embeddings	Random Forest	0.44	0.42	0.24
	Logistic Regression	0.45	0.39	0.26
	Neural Net	0.42	0.31	0.29

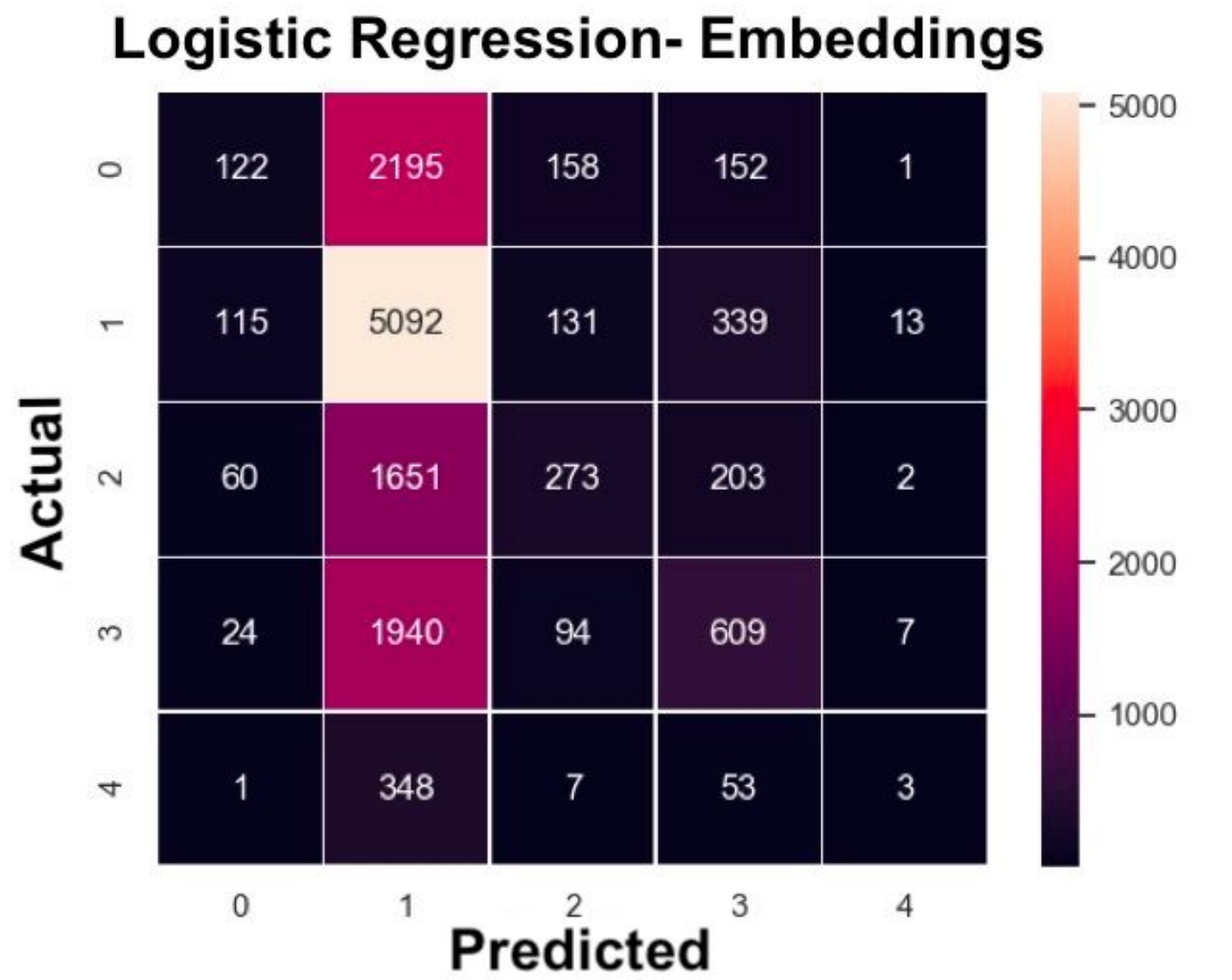


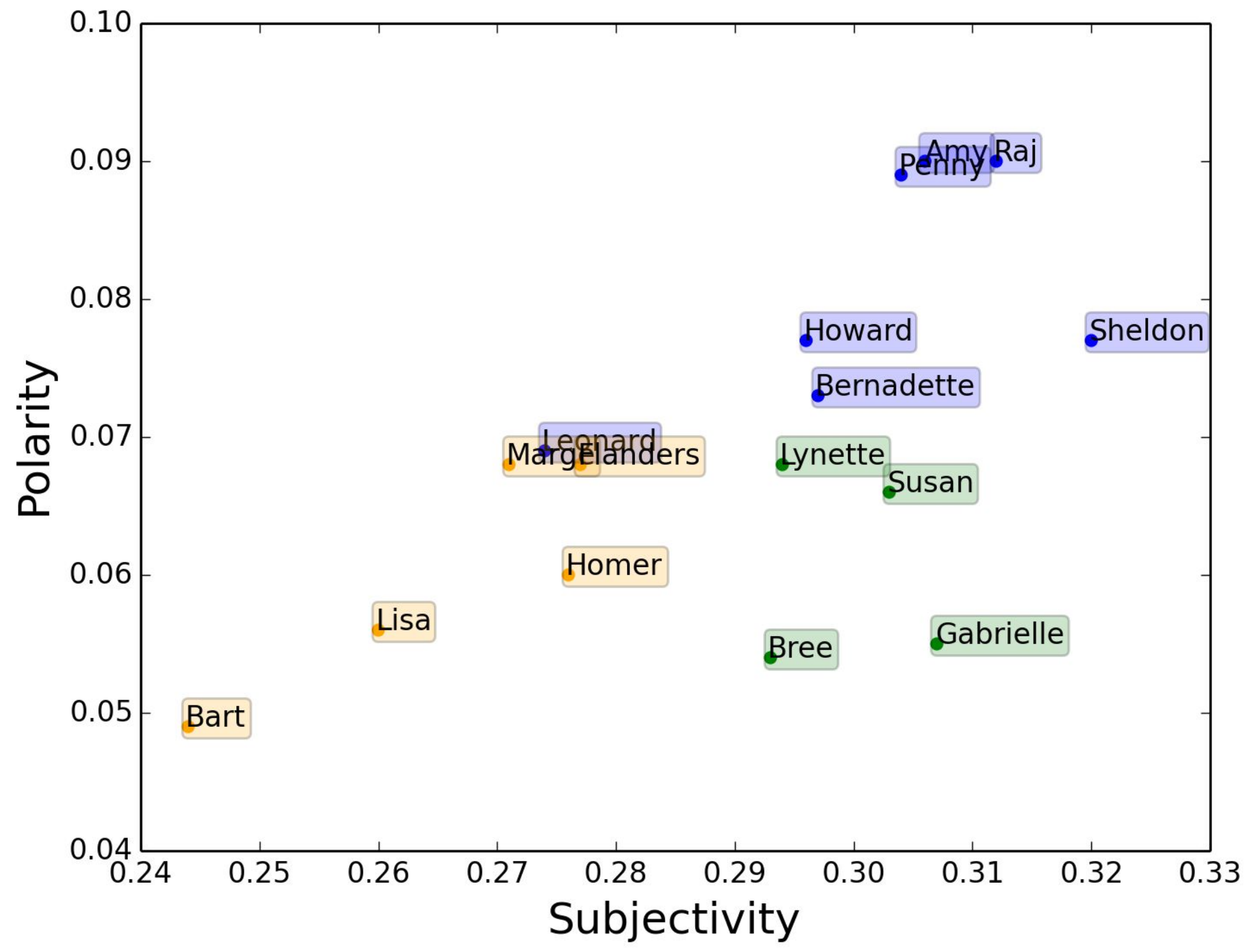


Desperate Housewives

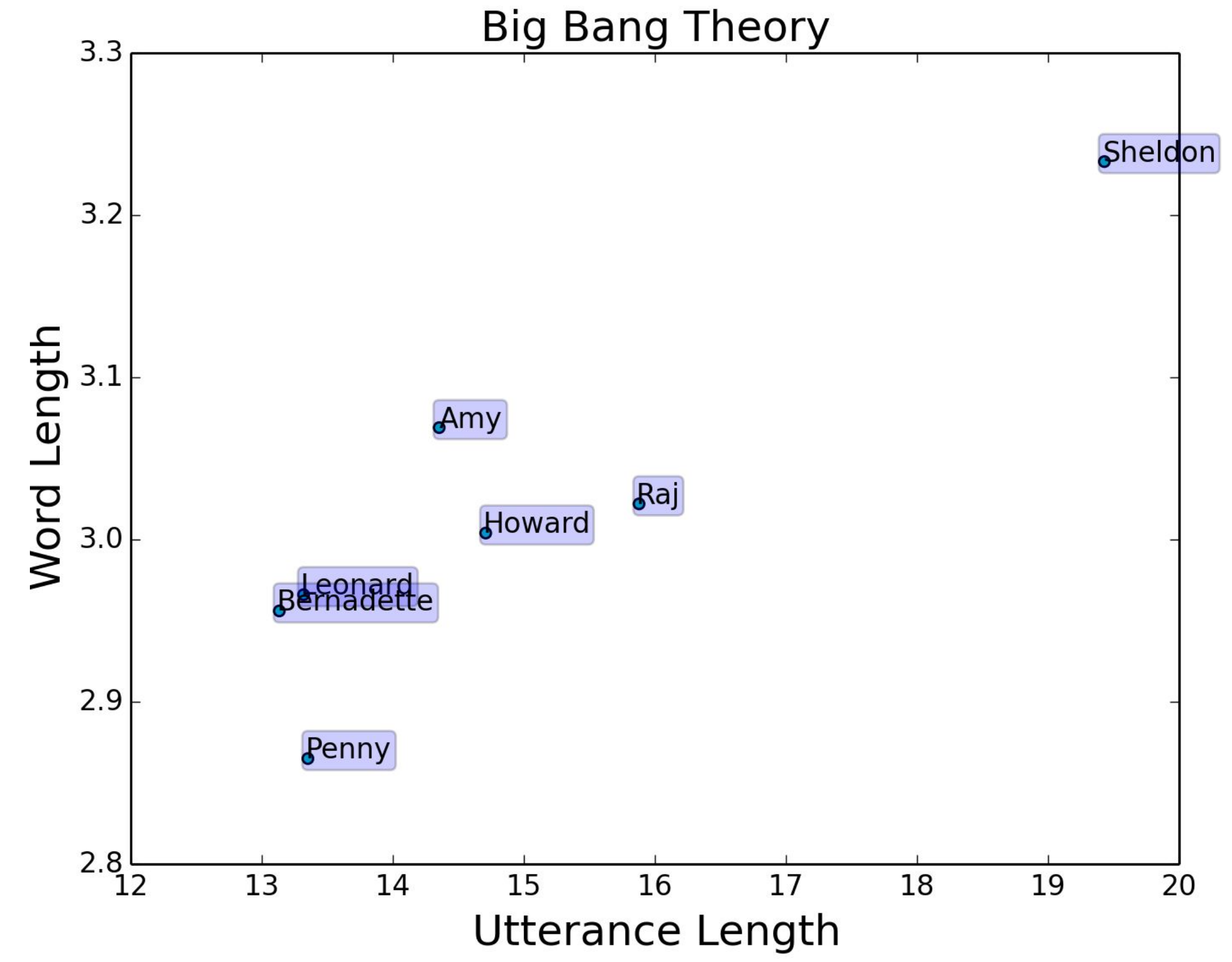
		Accuracy	Precision	Recall
	Random Baseline	0.25		
	Majority Baseline	0.28		
Manual Features	Random Forest	0.34	0.34	0.34
	Logistic Regression	0.34	0.35	0.34
	Neural Net	0.34	0.36	0.34
Embeddings	Random Forest	0.34	0.34	0.33
	Logistic Regression	0.36	0.36	0.35
	Neural Net	0.33	0.33	0.33







Polarity & **Subjectivity** differ greatly **across shows** (above). **Word Length** & **Utterance Length** make **Sheldon** easily identifiable (below).



CONCLUSION

Best model: Logistic Regression
Best features: Word Embeddings

- Higher accuracy for almost all cases!
- On average, 6.6 percentage points above majority baseline

Hypothesis: Genre will affect accuracy

- **Easiest show** to predict: Desperate House
- **Dramas** have subplots for each character, topic might help distinguish characters

Future steps!

- Explore **additional features**: Arousal, Dominance, POS and Topic
- Try **BERT** and Google's universal sentence encoder
- **Tune parameters** to optimize classification
- Try **more TV shows** to understand if indeed genre plays a role in classification