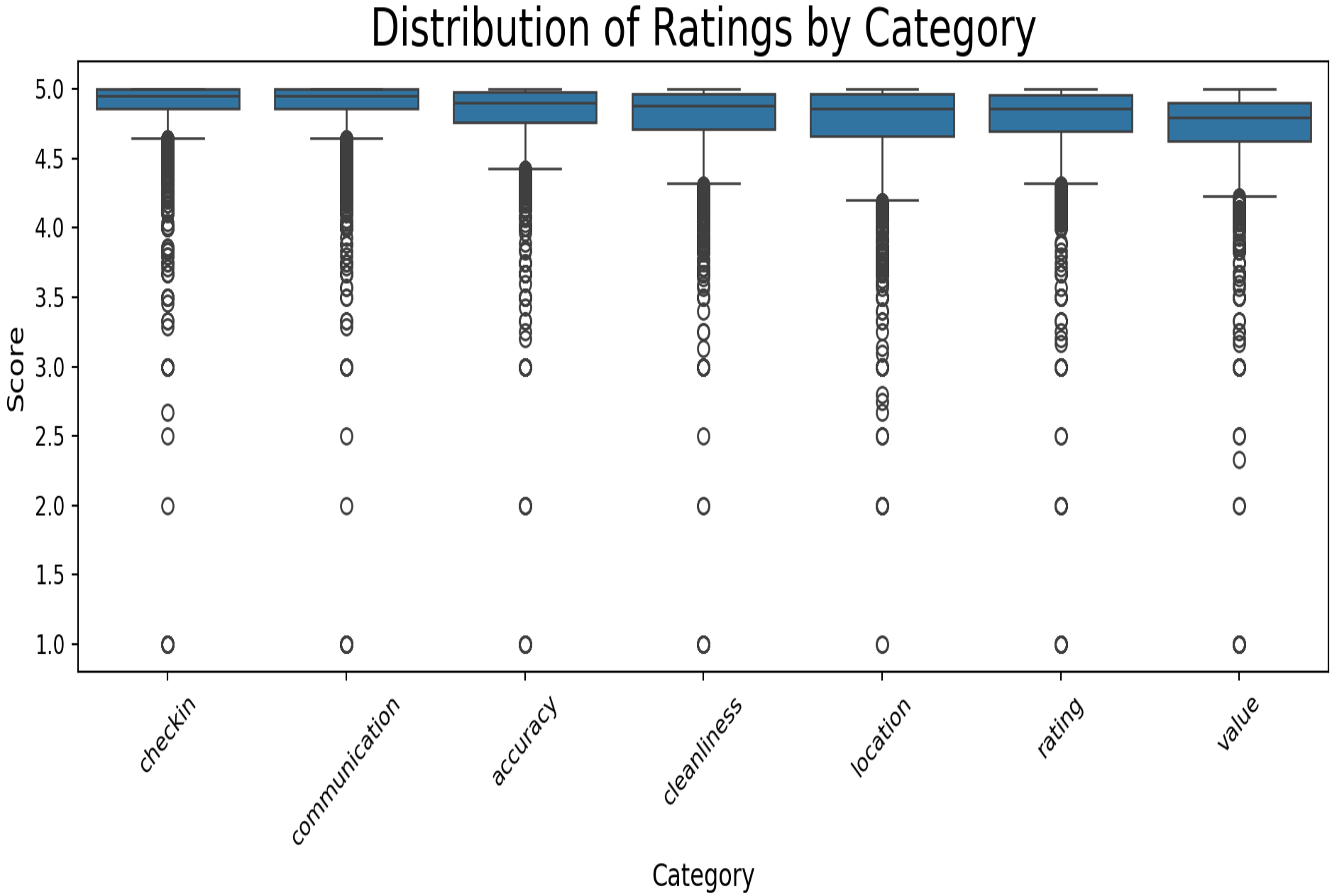
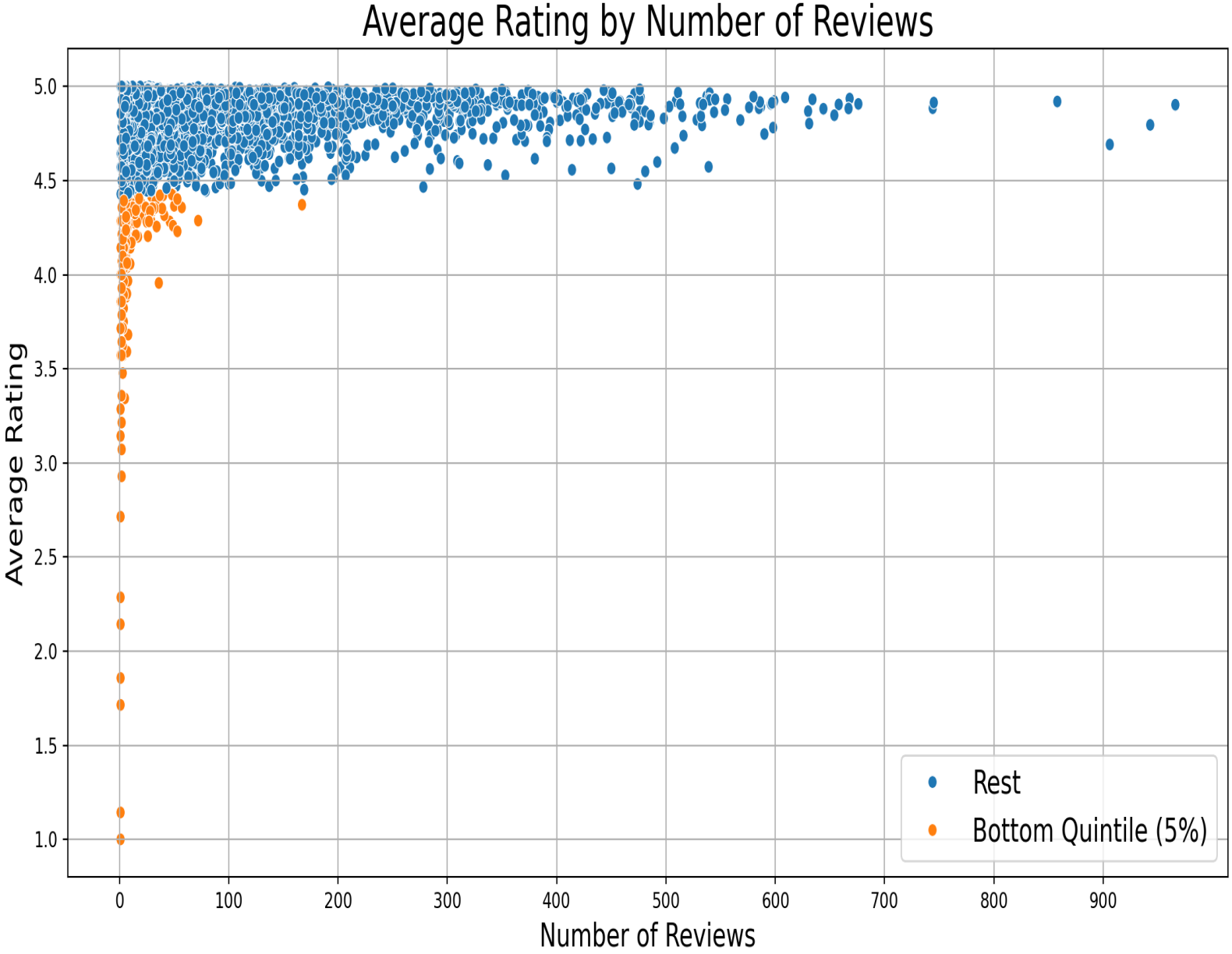
Predicting Poorly Rated AirBnB Listings in the DC Area Using Natural Language Processing

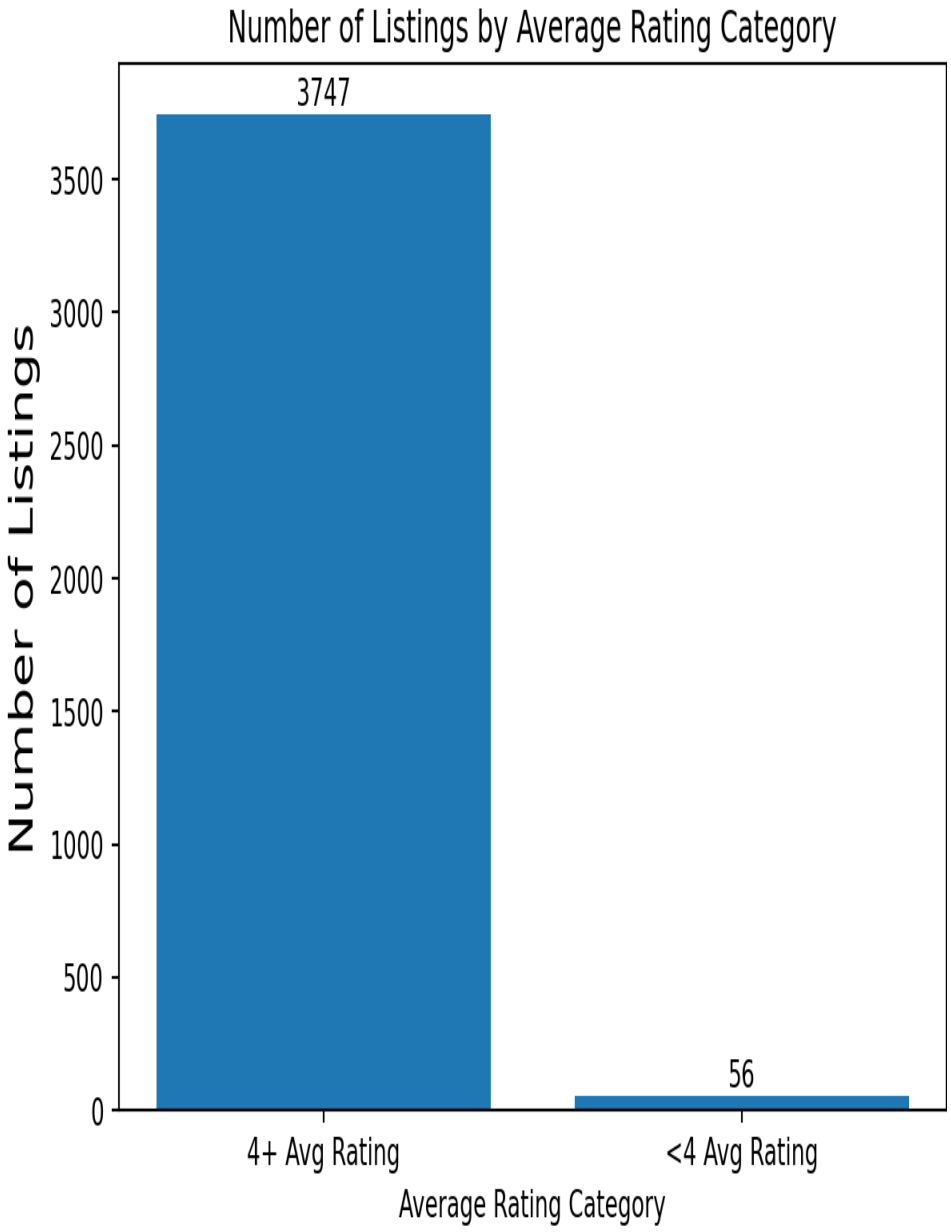
David Kanevsky, david@davidkanevsky.com

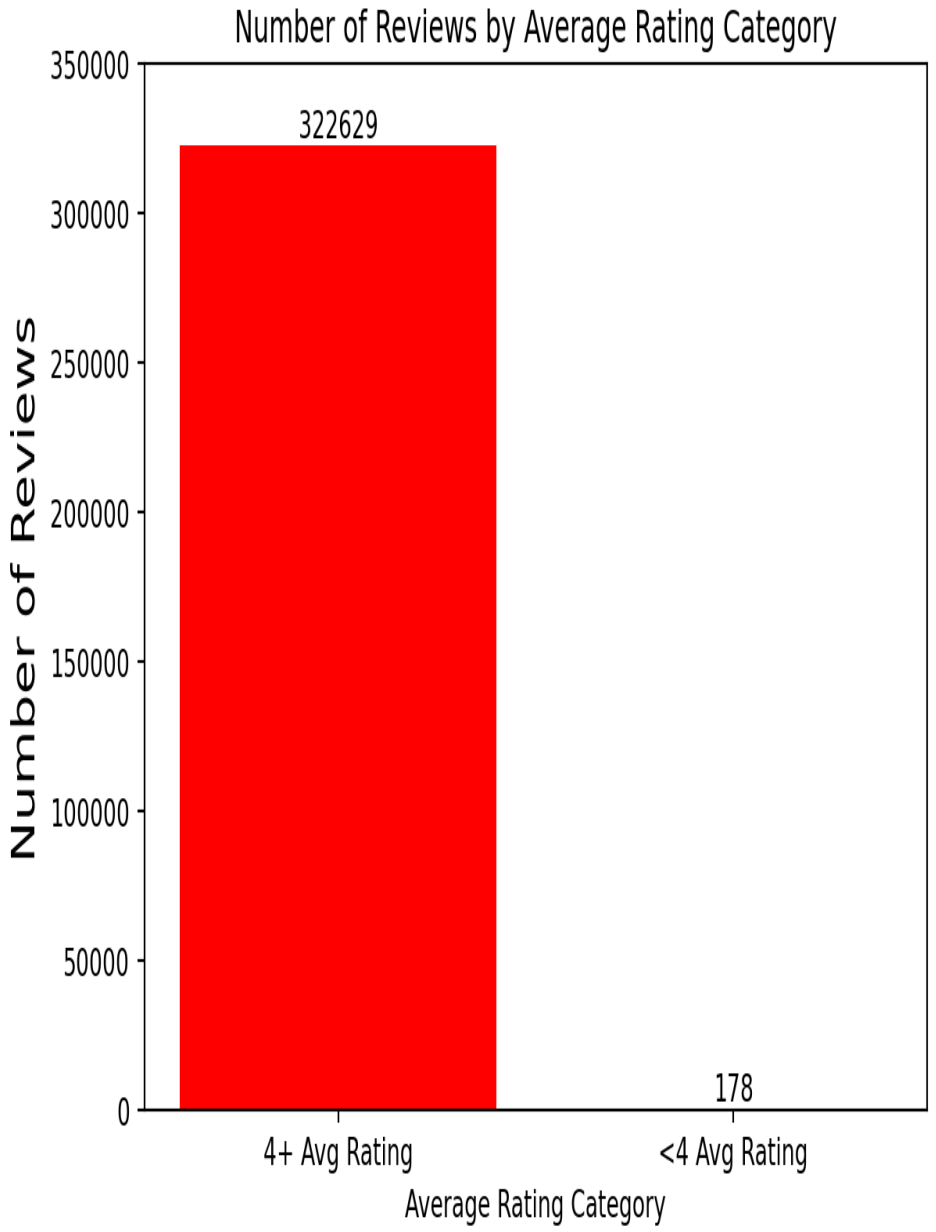
2024-06-13

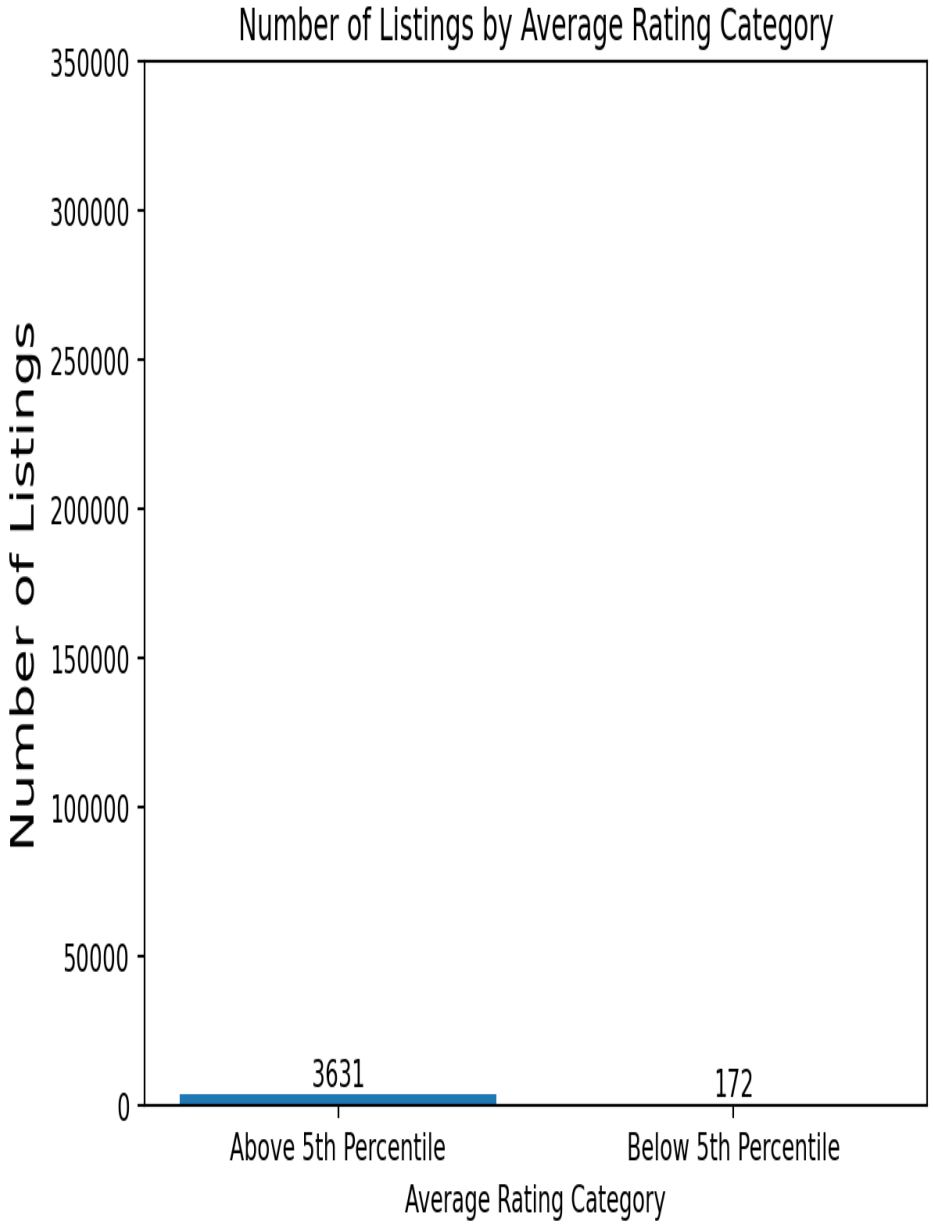
DSB 318

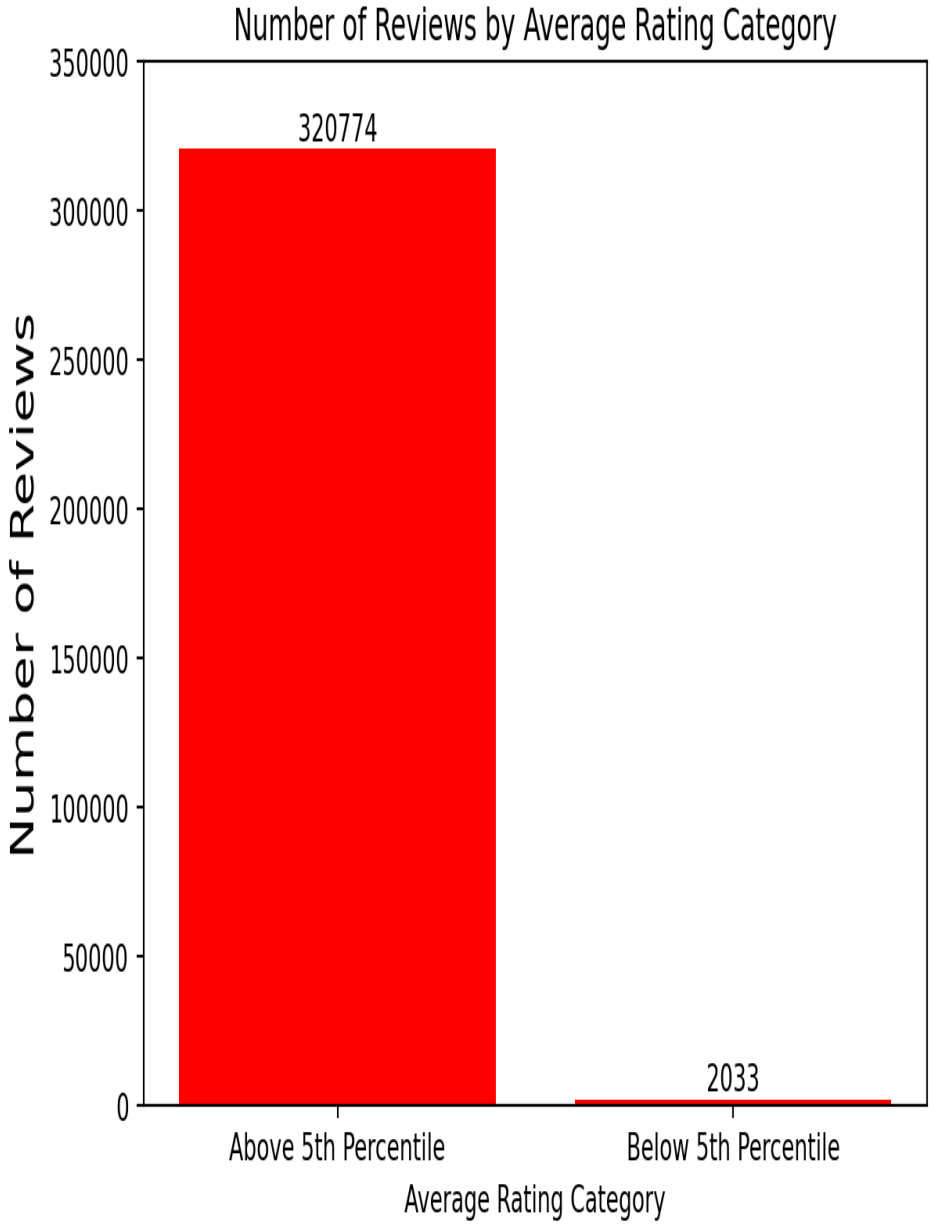

Distribution of Ratings by Category
Score
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
checkin
communication
accuracy
cleanliness
location
rating
value
Category


Average Rating by Number of Reviews
Average Rating
Number of Reviews
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0
100
200
300
400
500
600
700
800
900
Rest
Bottom Quintile (5%)


Number of Listings by Average Rating Category
Number of Listings
3747
56
0
500
1000
1500
2000
2500
3000
3500
4+ Avg Rating
<4 Avg Rating
Average Rating Category

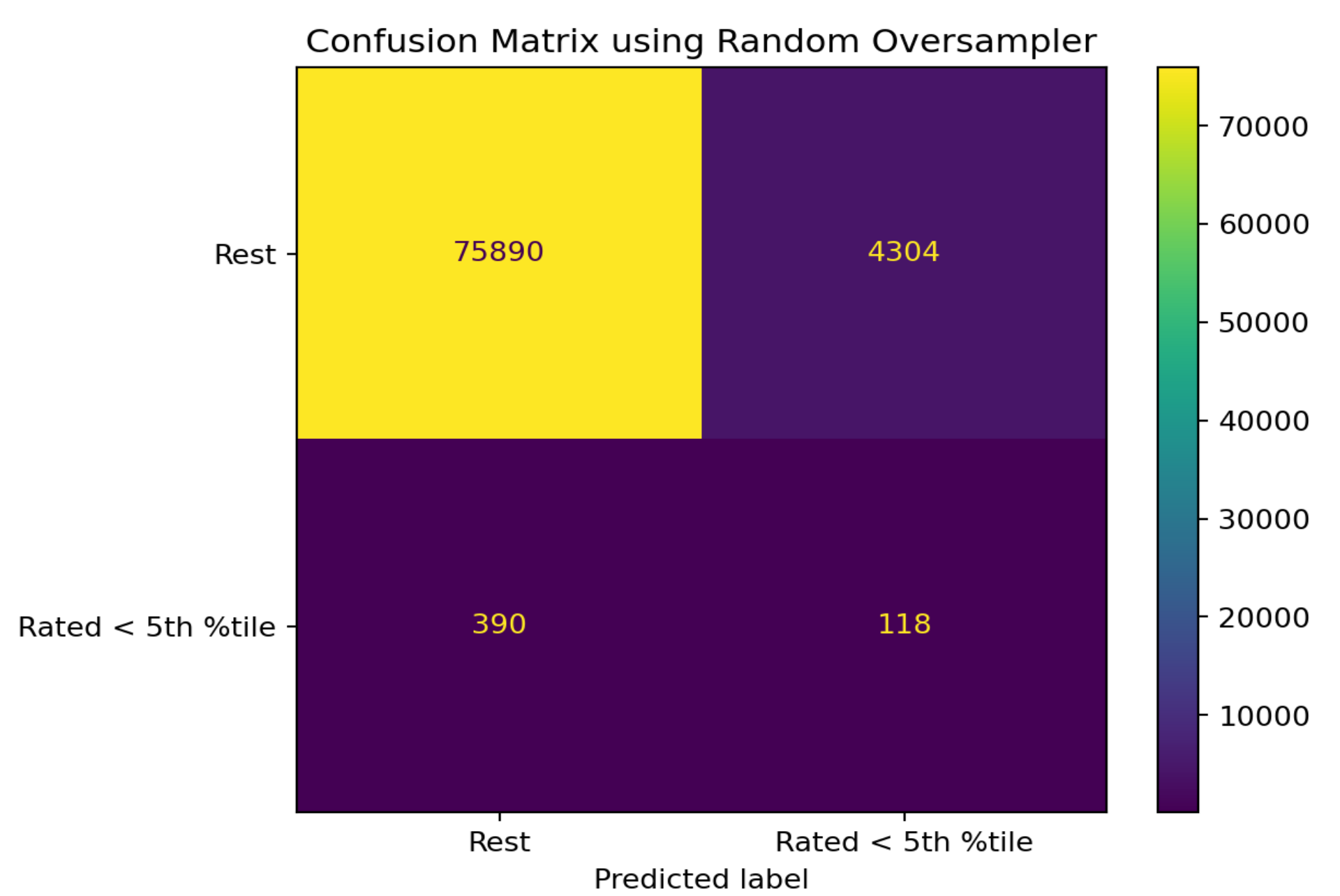

Number of Reviews by Average Rating Category
Number of Reviews
322629
178
0
50000
100000
150000
200000
250000
300000
350000
4+ Avg Rating
<4 Avg Rating
Average Rating Category

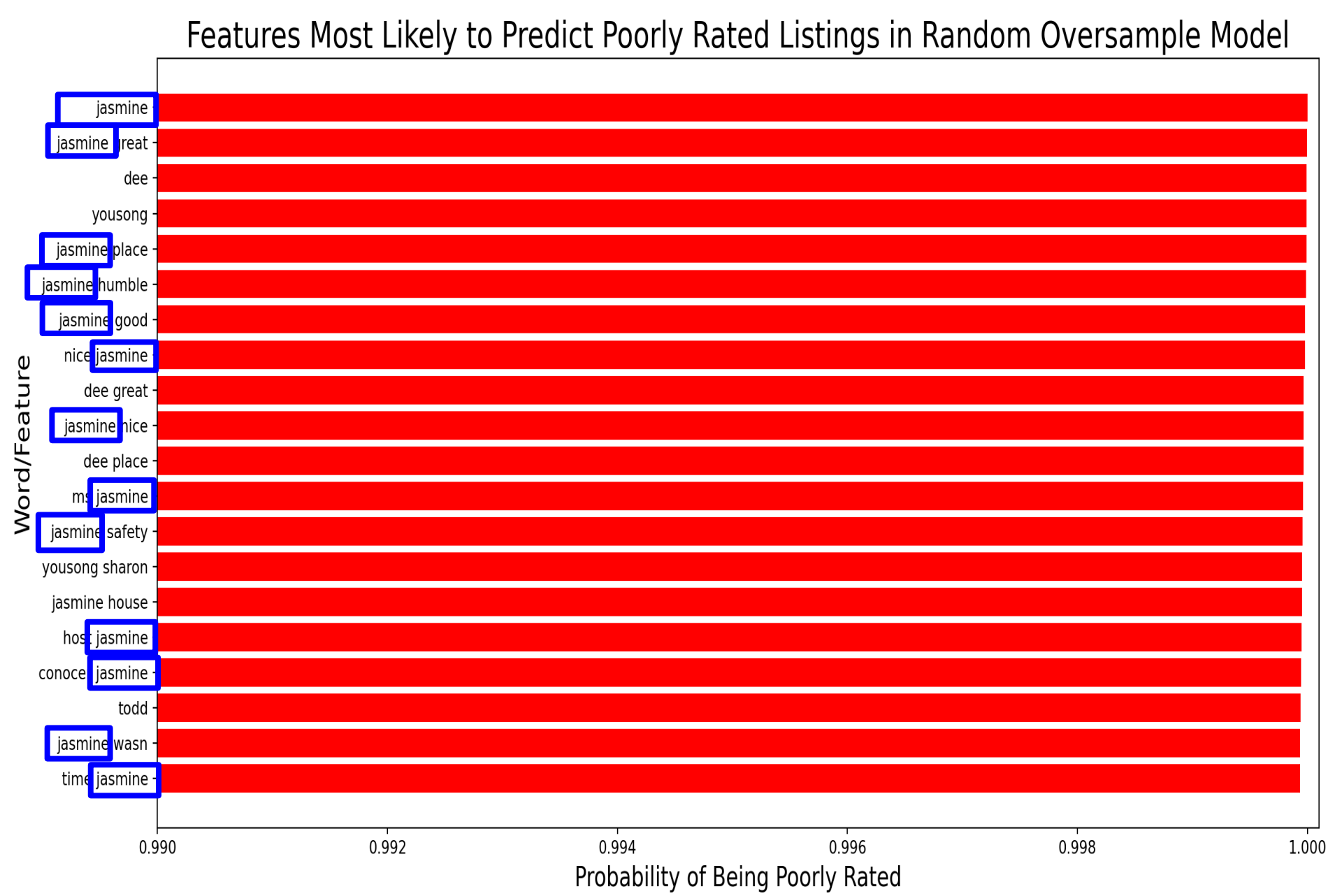

Number of Listings by Average Rating Category
Number of Listings
350000
300000
250000
200000
150000
100000
50000
0
3631
172
Above 5th Percentile
Below 5th Percentile
Average Rating Category

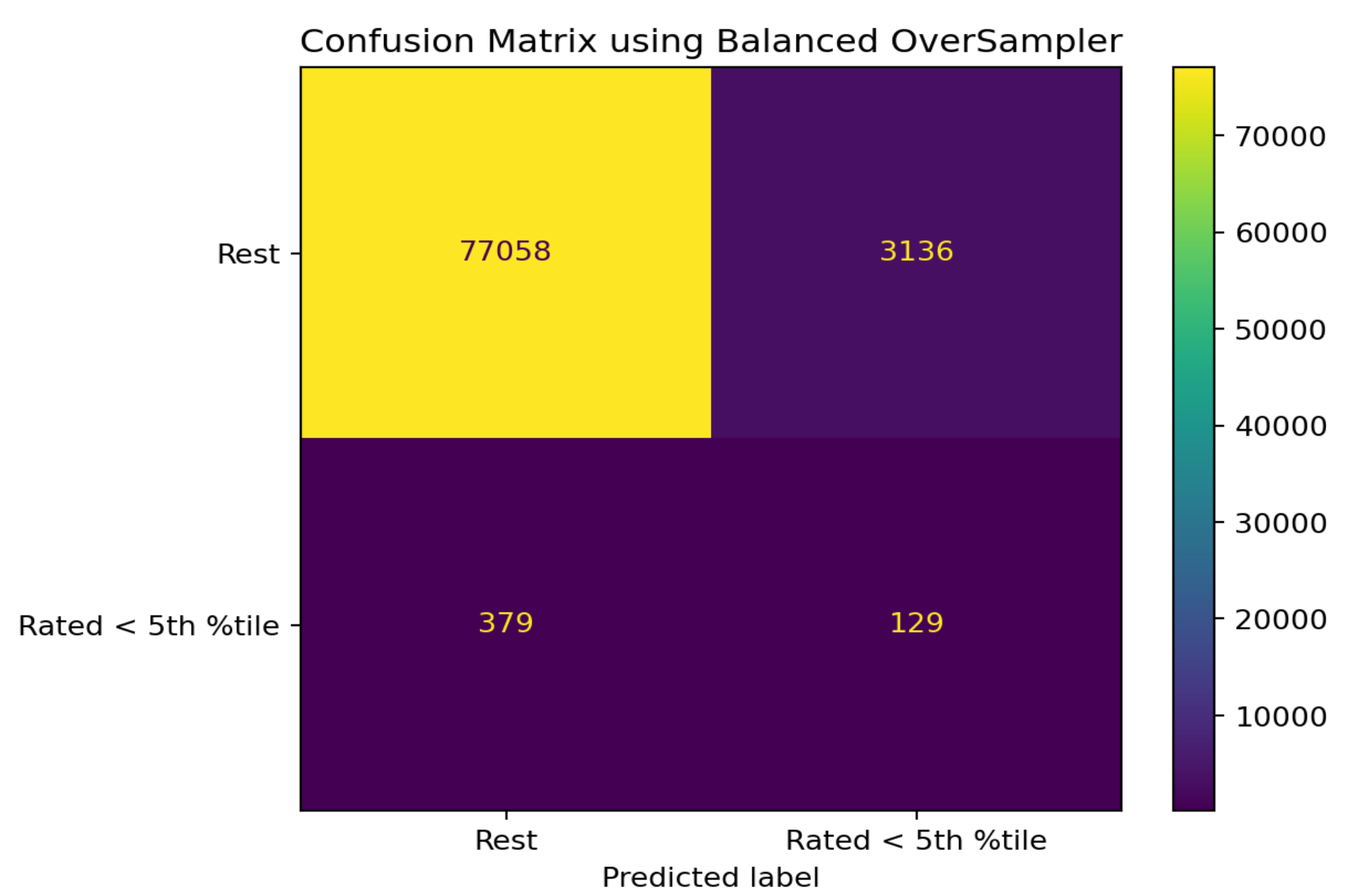

Number of Reviews by Average Rating Category
Number of Reviews
350000
300000
250000
200000
150000
100000
50000
0
320774
2033
Above 5th Percentile
Below 5th Percentile
Average Rating Category

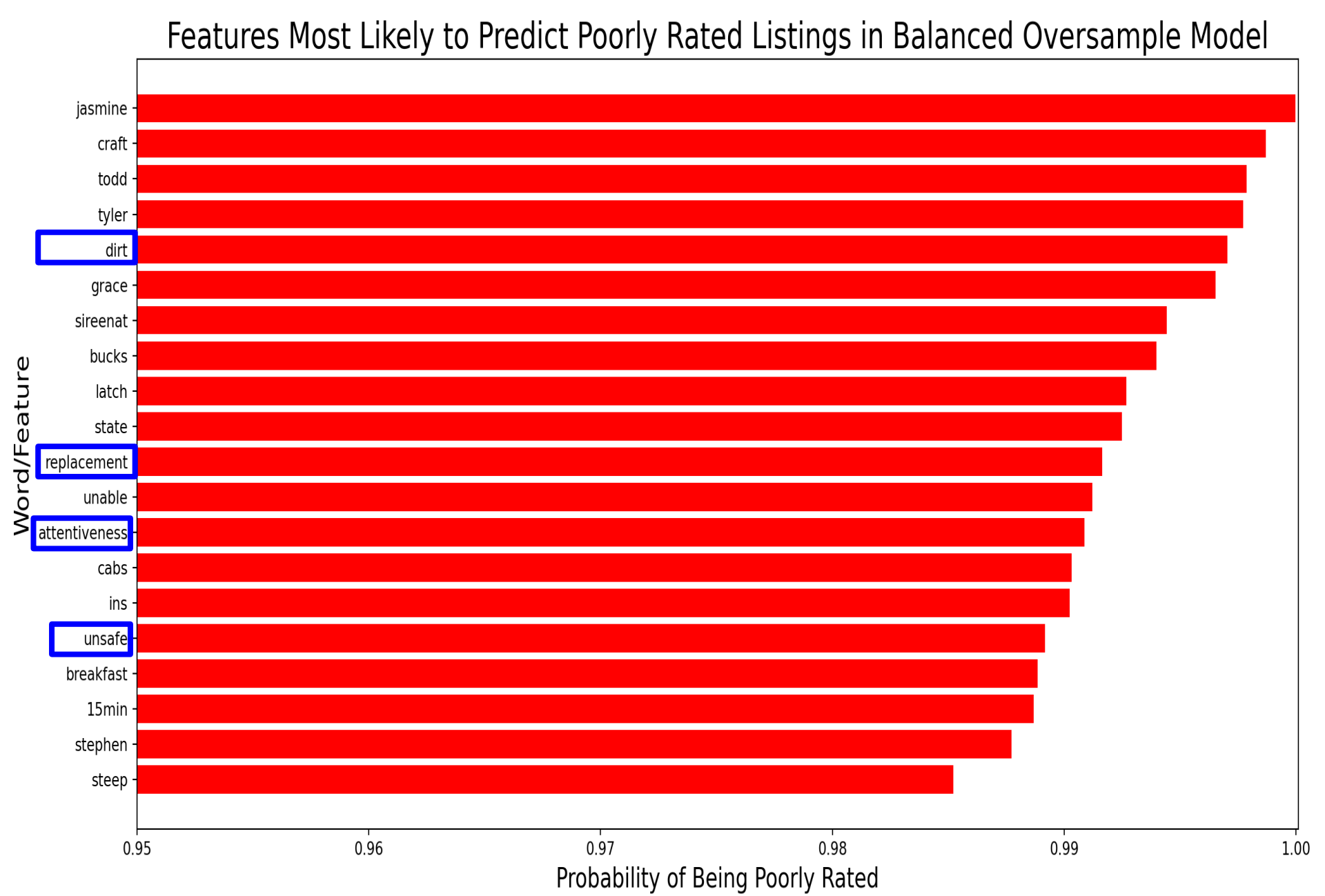
Model Iteration

Rating	Model	Recall Score	F1 Score
+/- 4.0 Avg Rating	Initial Model: TFIDF Vectorizer and Logistic Regression	0.000	0.000
+/- 4.0 Avg Rating	TFIDF & Logistic Regression ***with Balanced Classes***	0.111	0.009
5th Percentile	TFIDF & Logistic Regression with Balanced Classes	0.343	0.059
5th Percentile	TFIDF & Logistic Regression ***with Multinomial Naïve Bayes***	0.000	0.000
5th Percentile	***Count Vectorizer*** and ***Logistic Regression*** with Balanced Classes	0.242	0.047
5th Percentile	***TFIDF Vectorizer*** and Logistic Regression with ***Random Over Sampler***	0.467	0.048
5th Percentile	TFIDF Vectorizer and Logistic Regression with Balanced Classes and ***Custom Stop Words***	0.252	0.052
5th Percentile	TFIDF Vectorizer, Logistic Regression Scoring and Random Over Sampler, but ***Changing Sampling Strategy and Scoring to Balanced Accuracy***	0.254	0.068


Confusion Matrix using Random Oversampler
Rest
Rated < 5th %tile
75890
4304
390
118
Rest
Rated < 5th %tile
Predicted label
70000
60000
50000
40000
30000
20000
10000


Features Most Likely to Predict Poorly Rated Listings in Random Oversample Model
Word/Feature
jasmine
jasmine great
dee
yousong
jasmine place
jasmine humble
jasmine good
nice jasmine
dee great
jasmine nice
dee place
ms jasmine
jasmine safety
yousong sharon
jasmine house
host jasmine
conoce jasmine
todd
jasmine wasn
time jasmine
0.990
0.992
0.994
0.996
0.998
1.000
Probability of Being Poorly Rated


Confusion Matrix using Balanced OverSampler
Rest
Rated < 5th %tile
77058
3136
379
129
Rest
Rated < 5th %tile
Predicted label
70000
60000
50000
40000
30000
20000
10000


Features Most Likely to Predict Poorly Rated Listings in Balanced Oversample Model
jasmine
craft
todd
tyler
dirt
grace
sireenat
bucks
latch
state
replacement
unable
attentiveness
cabs
ins
unsafe
breakfast
15min
stephen
steep
Word/Feature
0.95
0.96
0.97
0.98
0.99
1.00
Probability of Being Poorly Rated

Conclusions and Recommendations

- The issue with imbalanced classes made it difficult to produce a model that had a good recall score and balanced accuracy. Models had an issue with false positives due to over-learning the names of hosts/listings, particularly when over-sampling the minority class of poorly reviewed listings.

Caveats

- The rating is for the listing's average rating across ALL reviews, not the rating each reviewer gave to their stay. Someone who stayed at a listing that is overall poorly reviewed but could give a positive review themselves (though it should be cancelled out by other negative reviews). There's no data on how each individual reviewer rated the listing.

Future Considerations

- Started looking at SMOTE to synthesize minority class of poor reviews.

- Additionally, might look at lemmatizing/stemming words (to make “safety” and “unsafe” both be “safe”) and/or spaCy to develop context around words.

- Get data from multiple cities, especially on poorly reviewed listings. This could help the model avoid learning about specific poorly reviewed listings/hosts in one city and instead look at what the commonalities are about poorly reviewed listings across multiple cities.

- Was reluctant to do random under sampling and get rid of so much data from positively rated listings. But the solution may be to look at just listings with under 100 reviews. Since all poorly rated listings have under 100 stays, looking at poorly rated listings with under 100 stays and well rated listings with under 100 stays may be a better comparison.

- Created binary classification by drawing a semi-arbitrary line. But since most listings are positively rated, difference is less between above/below 4.0 or above/below bottom quartile, and more sliding scale on difference between a 4.9 and 4.8. Tobit regression may work better to determine what features/words improve/reduce a listing’s rating rather than classification model.