# How do Gamers Feel?

Joshua Sonnen
sonnenj@wwu.edu

# Original Data

900,000 reviews across 242 games. Gathered from a web scrape of Steam. Posted on Kaggle.

Review Features

- ★ review message
- ★ hours_played
- ★ funny and helpful votes
- ★ recommendation (binary)
- ★ Date

Game Features

- ★ Game_name (290)
- ★ Publisher (173)
- ★ Developer (216)
- ★ Genres (list[str])
- ★ overall_player_rating (11)
- ★ Number of review from purchased people
- ★ Number of english reviews

# TextBlob Generated

I threw the data through TextBlob which generated some useful attributes for each review.

Review Features

- ★ Number of words
- ★ Number of sentences
- ★ Polarity (3)
- ★ Objectivity/Subjectivity (2)

–"How opinionated the review is"

# Game Sales

Gathered from a web scrape of Steam DB. Posted on Kaggle.

Game Features

- ★ NA sales
- ★ EU sales
- ★ JP sales
- ★ Other sales
- ★ Global sales

Sales are in millions of copies.

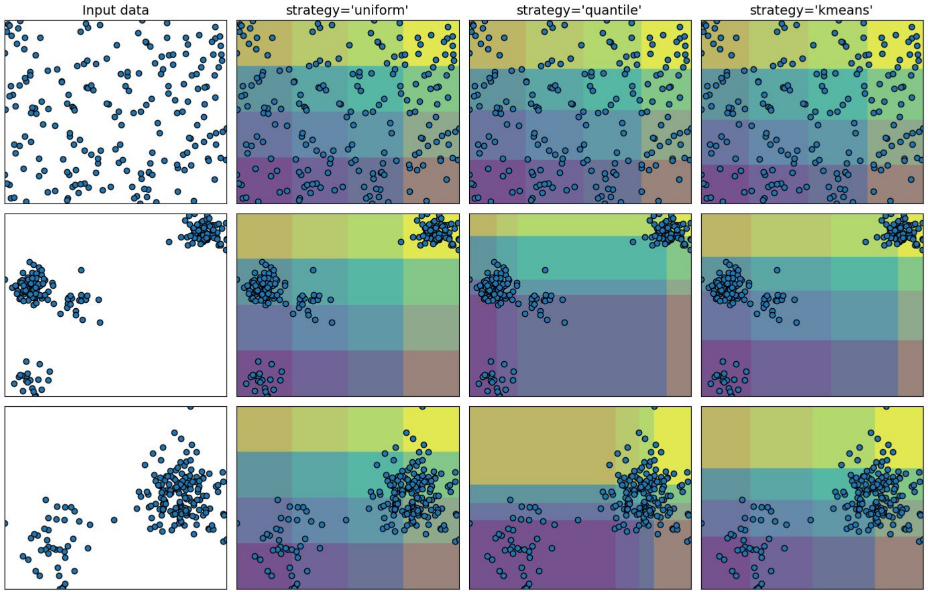
# Is TextBlob an accurate tool?

| Polarity | Objectivity | Message |
|---|---|---|
| ['positive', | 'subjective', | 'I feel so happy today.'] |
| ['neutral', | 'objective', | 'The sky is blue.'] |
| ['neutral', | 'objective', | "I'm worried about the future."] |
| ['positive', | 'subjective', | 'She seems very kind.'] |
| ['neutral', | 'objective', | "I don't know what to do."] |
| ['neutral', | 'objective', | 'The meeting starts at 3 PM.'] |
| ['negative', | 'subjective', | 'That movie was terrible!'] |
| ['positive', | 'objective', | 'I love spending time with my family.'] |
| ['neutral', | 'objective', | 'It rained for two hours yesterday.'] |
| ['neutral', | 'objective', | "I'm feeling really stressed right now."] |

TextBlob

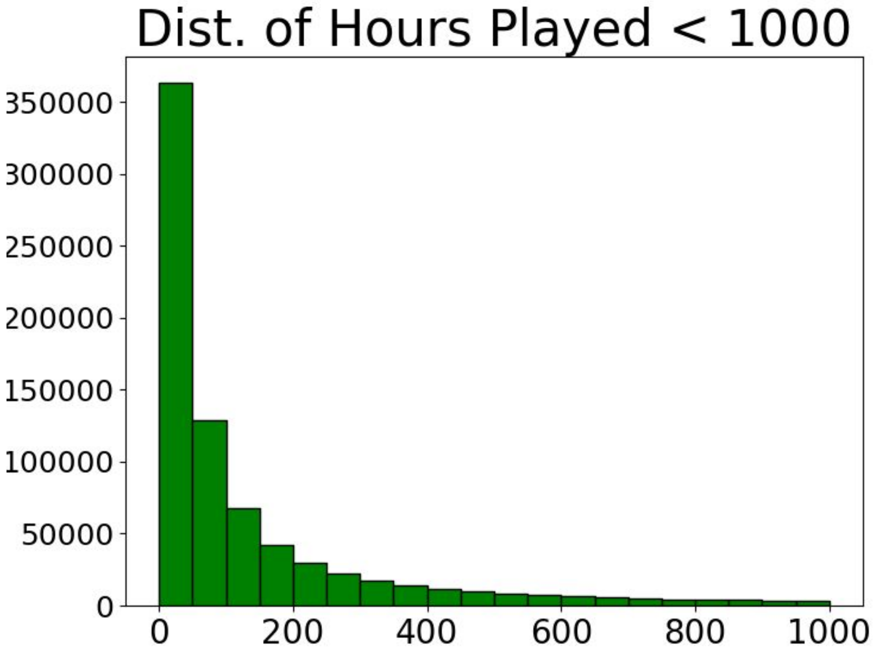
# Hours Played Discretization

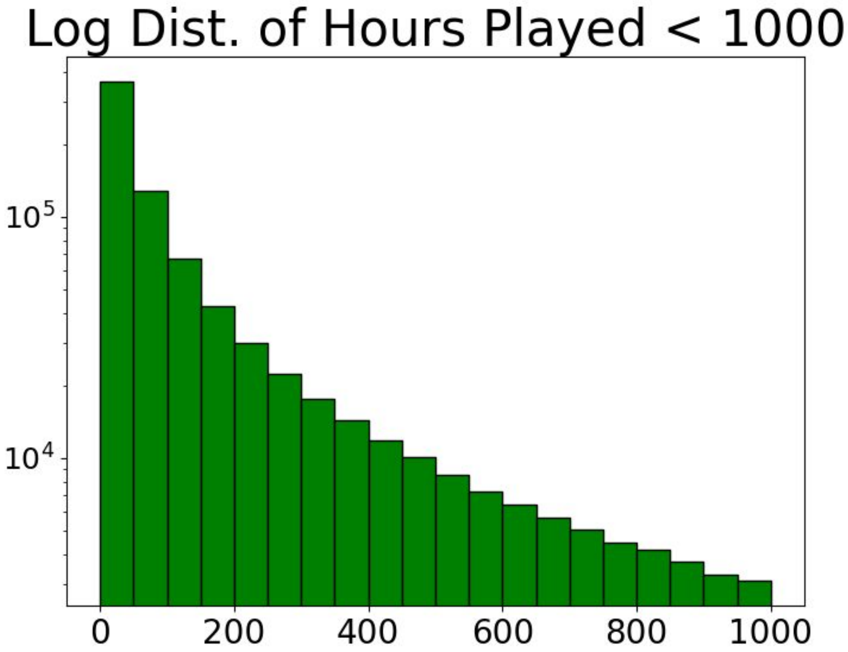
- Split by equal frequency
- Split by equal ranges
  - Remove outlier: Hours_played < 1,000
- K Means Classify hours played
- log(hours_played) then discretize by equal ranges

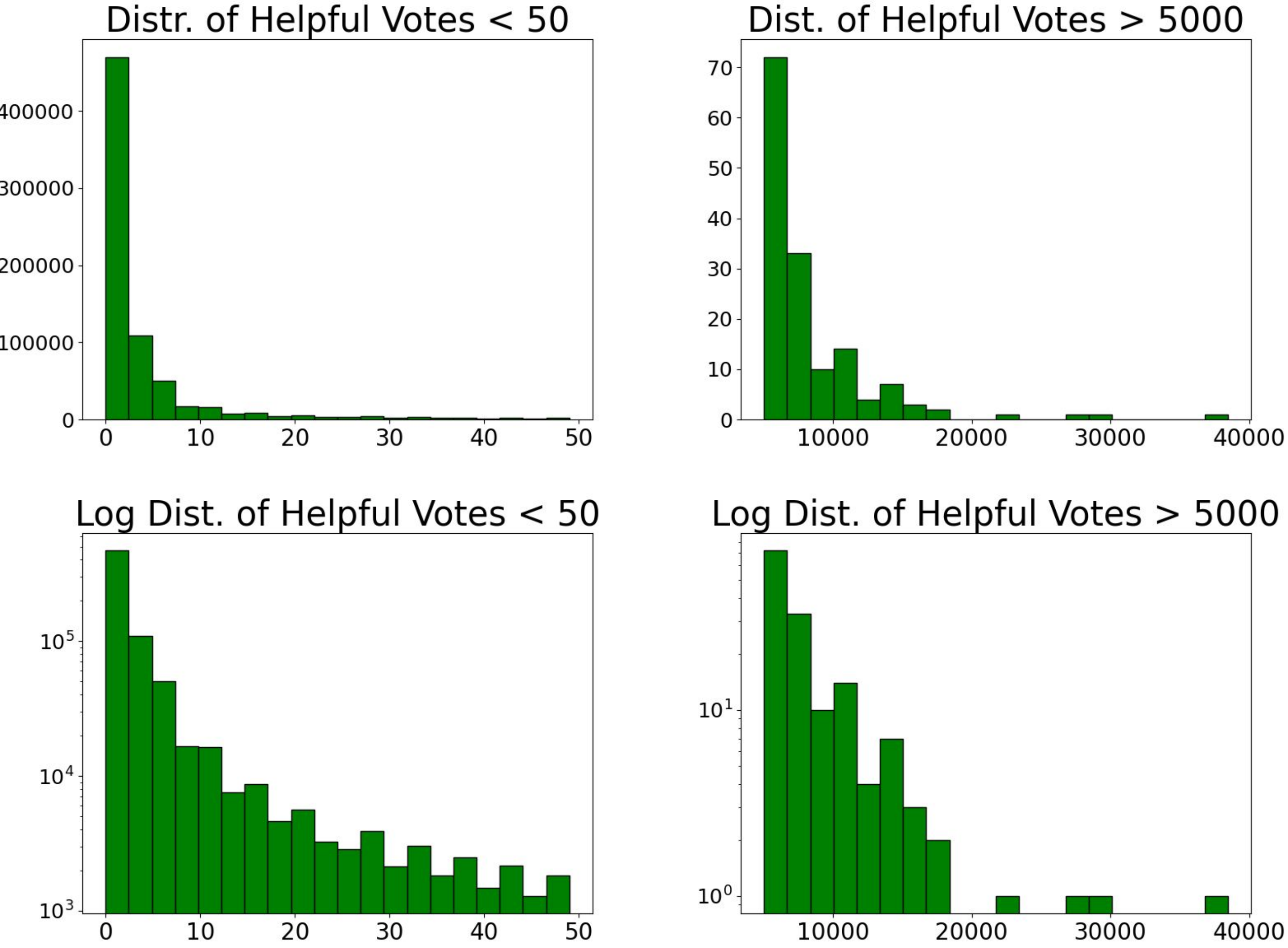


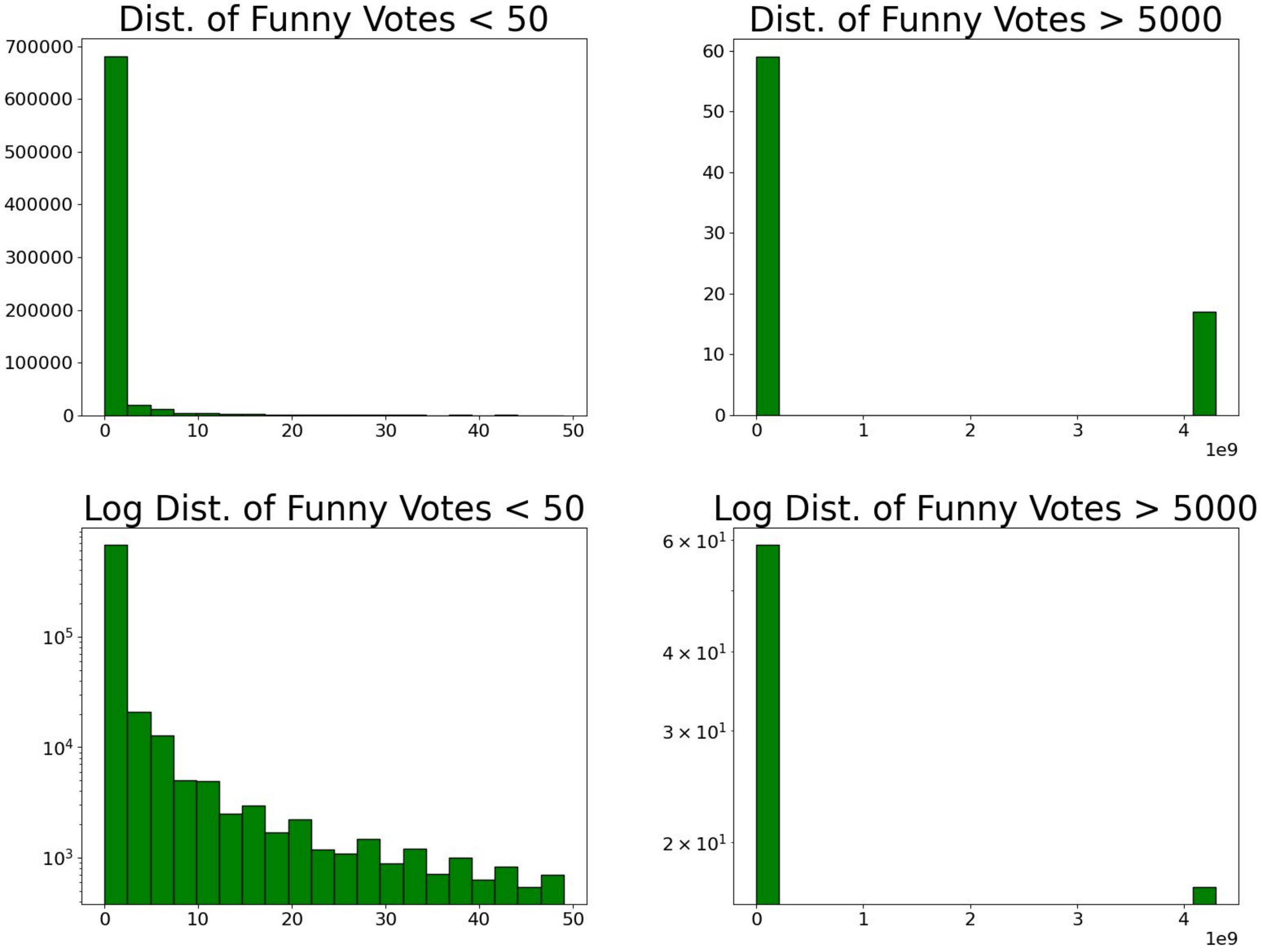

Source: sklearn “Demonstrating the different strategies of KBinsDiscretizer”

Throughout all analysis and classification, I use the subset of data where hours_played < 1000. This data nicely follows a logarithmic distribution.




Distr. of Helpful Votes < 50
Dist. of Helpful Votes > 5000
Log Dist. of Helpful Votes < 50
Log Dist. of Helpful Votes > 5000
400000
300000
200000
100000
0
10
20
30
40
50
70
60
10000
20000
30000
40000
$10^5$
$10^4$
$10^3$
$10^1$
$10^0$




There is a sparsity in reviews with funny votes > 5000.

# Bootstrapping

With 1,000,000 data points, my first step is to remove some data

1. Sample with replacement. These indices are dropped
2. Set difference → Return not sampled indices
3. 80% train; 20% validation split

Sampling k points on the range (0, N) I will expect

Unique data points = $n * \left(1 - \left(1 - \frac{1}{n}\right)^k\right)$

k = n : $n * (1 - e^{-1}) = 0.632n \rightarrow 279{,}605$

k = 2n : $n * (1 - e^{-2}) = 0.865n \rightarrow 102{,}572$

k = 3n : $n * (1 - e^{-3}) = 0.950n \rightarrow 37{,}990$

For K Nearest Neighbors classification, I use a bootstrap factor of 3.

For Random Forest classification, I use a bootstrap factor of 5.

# Welch's T Test

Welch's t-test is a variant to Student's t-test that overcomes the assumption of equal variance. Welch's t-test still assumes normality between samples.

$H_0$: $\mu_1 = \mu_2$

$H_a$: $\mu_1 \neq \mu_2$

Welch's *t*-test defines the statistic *t* by the following formula:

$$t = \frac{\Delta \overline{X}}{s_{\Delta \bar{X}}} = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{s_{\bar{X}_1}^2 + s_{\bar{X}_2}^2}}$$

$$s_{\bar{X}_i} = \frac{s_i}{\sqrt{N_i}}$$

$$\text{d.f.} \approx \frac{\left( \frac{s_1^2}{N_1} + \frac{s_2^2}{N_2} \right)^2}{\frac{s_1^4}{N_1^2 \nu_1} + \frac{s_2^4}{N_2^2 \nu_2}}.$$

Wikipedia: welch's t-test

# K Nearest Neighbors

## Predicting Hours Played

# K Nearest Neighbor Feature Subsets

<u>Five hopeful groups</u>

- All

  'helpful','funny','polarity','objectivity','n_words','n_sentences'

- Vote count

  'helpful','funny'

- TextBlob

  ‘polarity’,'objectivity'

- TextBlob extra

  ‘n_words’,‘n_sentences’,‘polarity’, 'objectivity'

- Pass chi-squared p = 0.05
  - 'helpful','funny',objectivity,’n_words’,’n_sentences’
    - Dropped polarity

# K Nearest Neighbors Trials

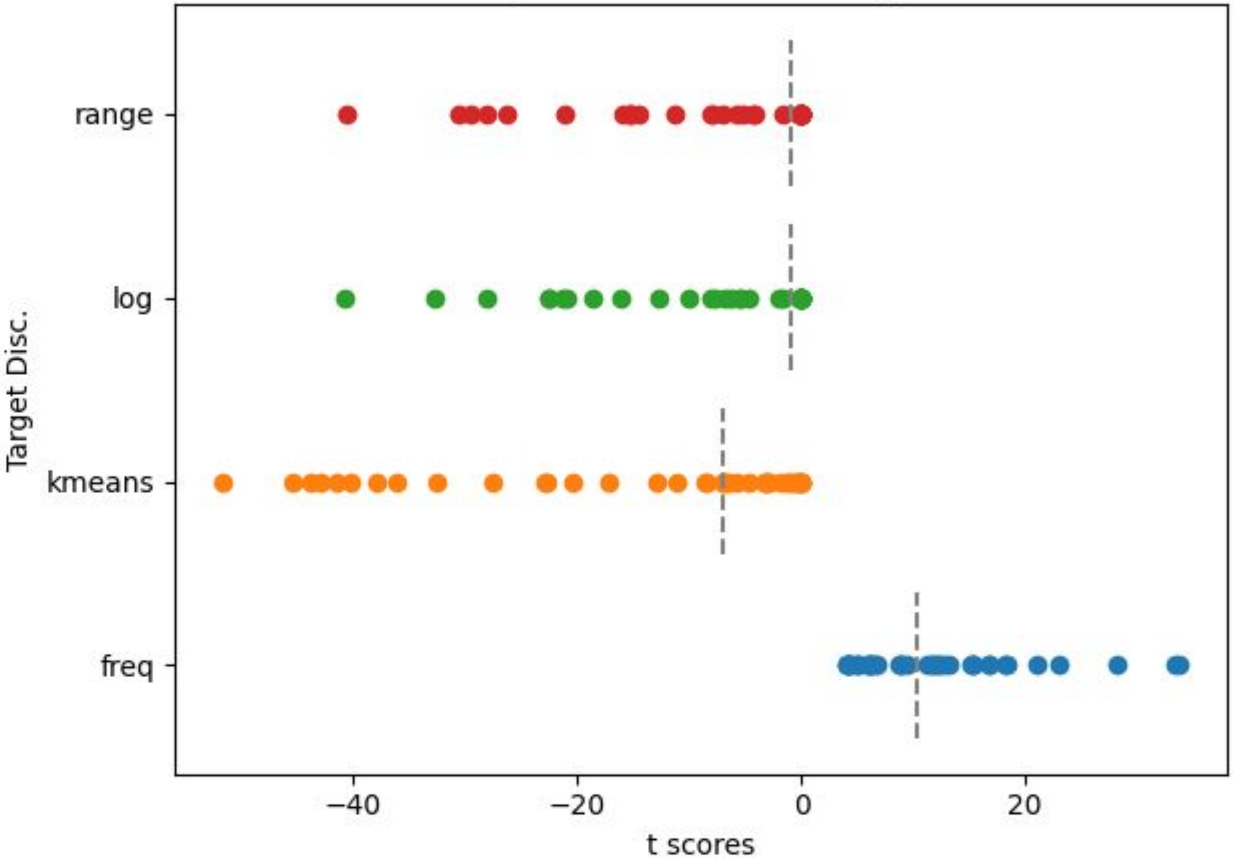
## Trial 1– 120 runs

- 5 feature subsets
- 4 Target Discretization Strategies
  - Range
  - Frequency
  - K-Means
  - Equal Range on log(hours_played)
- 4 bin sizes to discretize targets into
  - 3, 8, 12, 20
- 2 choices for k
  - 5, 87

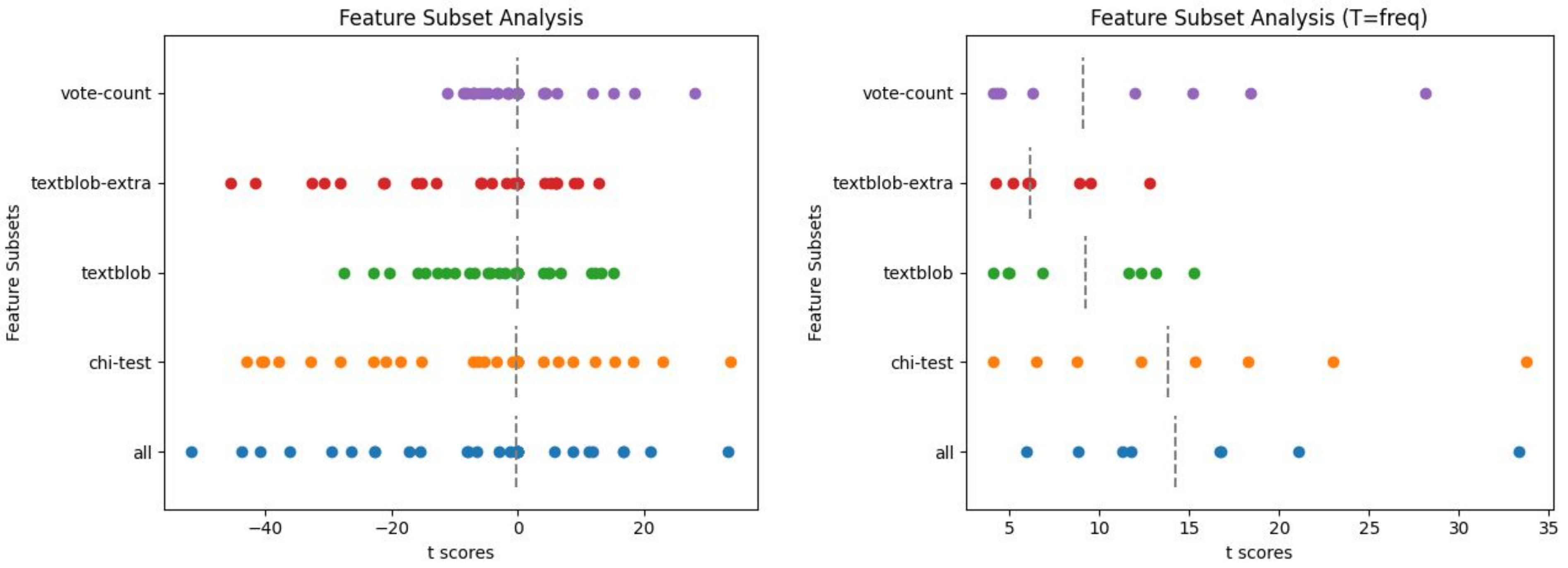
## Trial 2– 300 runs

- 5 feature subsets
- 1 Target Discretization Strategie
  - Frequency
- 20 bin sizes to discretize targets into
  - [ 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100]
- 3 choices for k
  - 5, 100, 1000

# Target Discretization Analysis




Equal frequencies was the best target discretization strategy.


Feature Subset Analysis
vote-count
textblob-extra
textblob
chi-test
all
Feature Subsets
−40
−20
0
20
t scores
Feature Subset Analysis (T=freq)
5
10
15
20
25
30
35

# Trial 1 classifiers which improve over Majority Class Baseline

F=all_T=freq_GROUP=3_K=5
F=all_T=freq_GROUP=8_K=5
F=all_T=freq_GROUP=12_K=5
F=all_T=freq_GROUP=20_K=5

F=chi_test_T=freq_GROUP=3_K=5
F=chi_test_T=freq_GROUP=8_K=5
F=chi_test_T=freq_GROUP=12_K=5
F=chi_test_T=freq_GROUP=20_K=5

F=textblob_extra_T=freq_GROUP=3_K=5
F=textblob_extra_T=freq_GROUP=8_K=5
F=textblob_extra_T=freq_GROUP=12_K=5
F=textblob_extra_T=freq_GROUP=20_K=5

F=textblob_T=freq_GROUP=3_K=5
F=textblob_T=freq_GROUP=8_K=5
F=textblob_T=freq_GROUP=12_K=5
F=textblob_T=freq_GROUP=20_K=5

F=vote_count_T=freq_GROUP=3_K=5
F=vote_count_T=freq_GROUP=8_K=5
F=vote_count_T=freq_GROUP=12_K=5
F=vote_count_T=freq_GROUP=20_K=5

F=all_T=freq_GROUP=3_K=87
F=all_T=freq_GROUP=8_K=87
F=all_T=freq_GROUP=12_K=87
F=all_T=freq_GROUP=20_K=87

F=chi_test_T=freq_GROUP=3_K=87
F=chi_test_T=freq_GROUP=8_K=87
F=chi_test_T=freq_GROUP=12_K=87
F=chi_test_T=freq_GROUP=20_K=87

F=textblob_extra_T=freq_GROUP=3_K=87
F=textblob_extra_T=freq_GROUP=8_K=87
F=textblob_extra_T=freq_GROUP=12_K=87
F=textblob_extra_T=freq_GROUP=20_K=87

F=textblob_T=freq_GROUP=3_K=87
F=textblob_T=freq_GROUP=8_K=87
F=textblob_T=freq_GROUP=12_K=87
F=textblob_T=freq_GROUP=20_K=87

F=vote_count_T=freq_GROUP=3_K=87
F=vote_count_T=freq_GROUP=8_K=87
F=vote_count_T=freq_GROUP=12_K=87
F=vote_count_T=freq_GROUP=20_K=87

Statistically significant at $p = 1e\text{-}5$

Others were statistically Insignificant at $p = 0.05$

Best classifier accuracy improvement over majority class is 5.8%

Most knn classifiers using k-means or equal range discretization performed significantly **worse** than the Majority Classifier

# Pairwise T-Tests on Feature Subset

- Welch's unequal variance t-test
- Cutoff p-value of 0.05
- Summed across all 120 trials:
  - k = [5, 87]
  - Target Discretize = range, frequency, k means
  - Group size = [3, 8, 12, 20]

Key Takeaways

- Textblob-extra and textblob perform poorly as feature subsets.

Losers

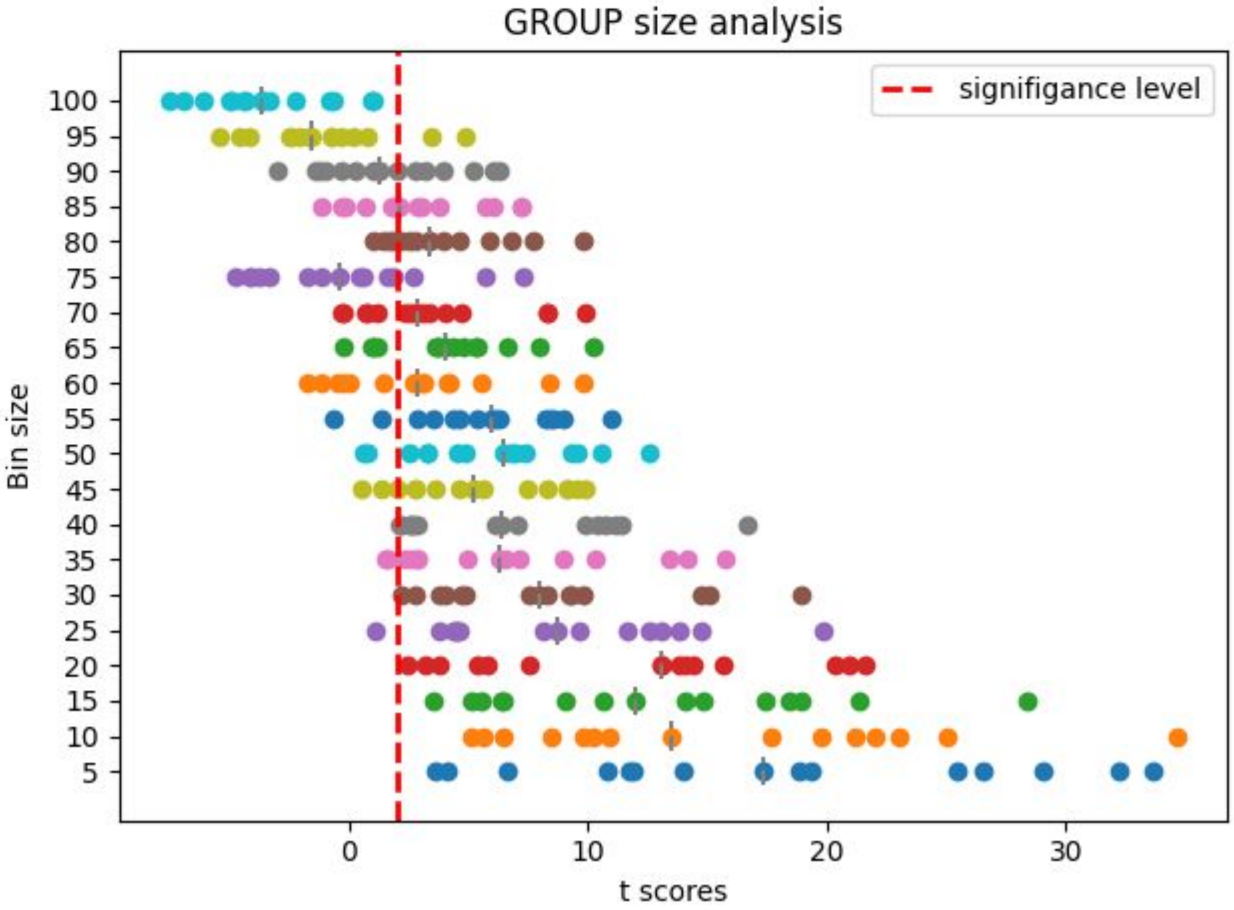
| Winners | vote-count | all | textblob | chi-test | textblob-extra |
|---|---|---|---|---|---|
| vote-count | 0 | 4 | 7 | 5 | 7 |
| all | 1 | 0 | 6 | 1 | 7 |
| textblob | 0 | 4 | 0 | 3 | 6 |
| chi-test | 0 | 0 | 5 | 0 | 8 |
| textblob-extra | 1 | 0 | 1 | 0 | 0 |

Feature Subset Rankings

1. Vote Count
2. All
3. Chi-Test
4. Textblob
5. Textblob Extra

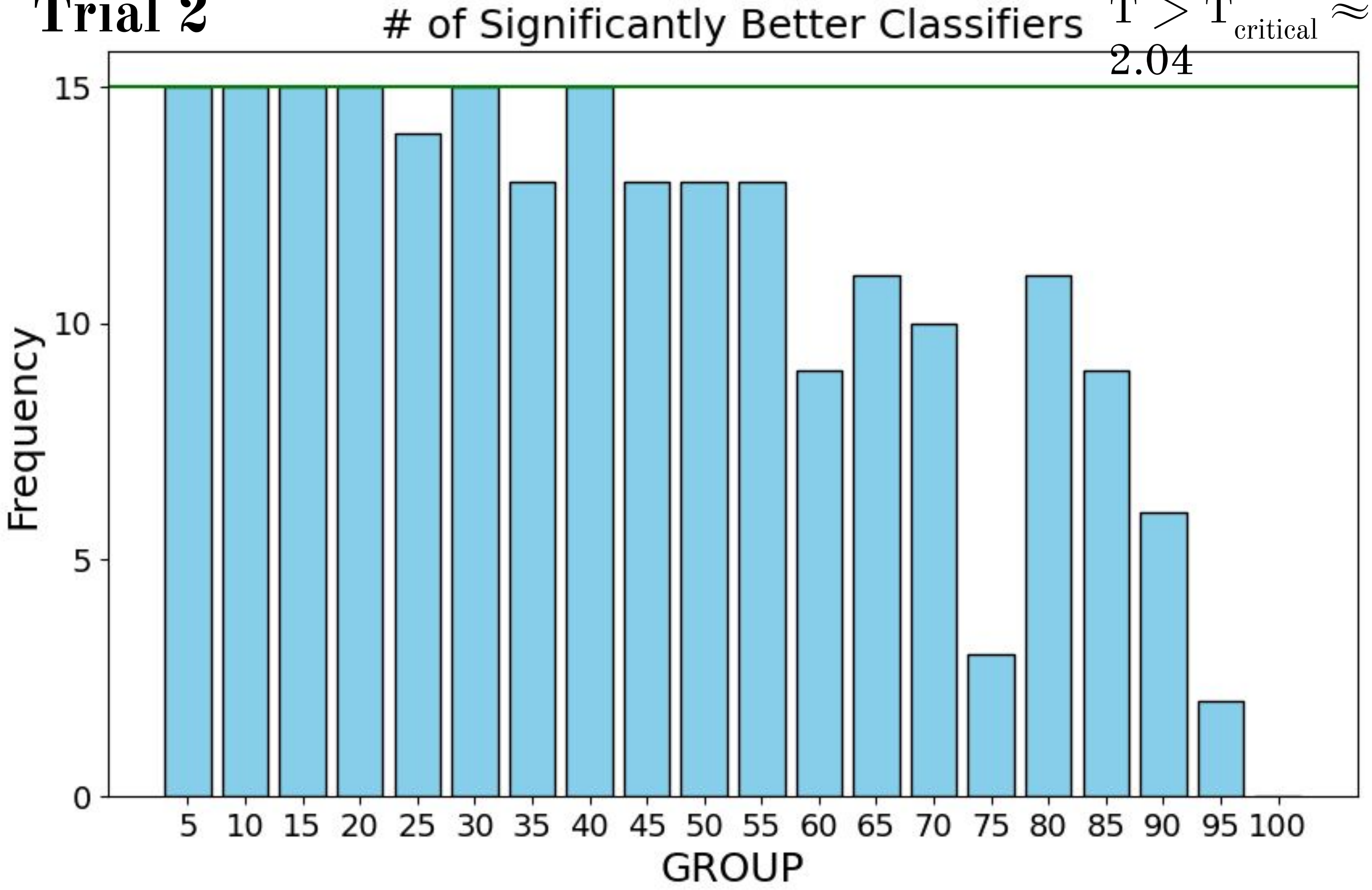
# Helpful and Funny Votes are the most important features when determining Hours Played, using a K Nearest Neighbors Classifier

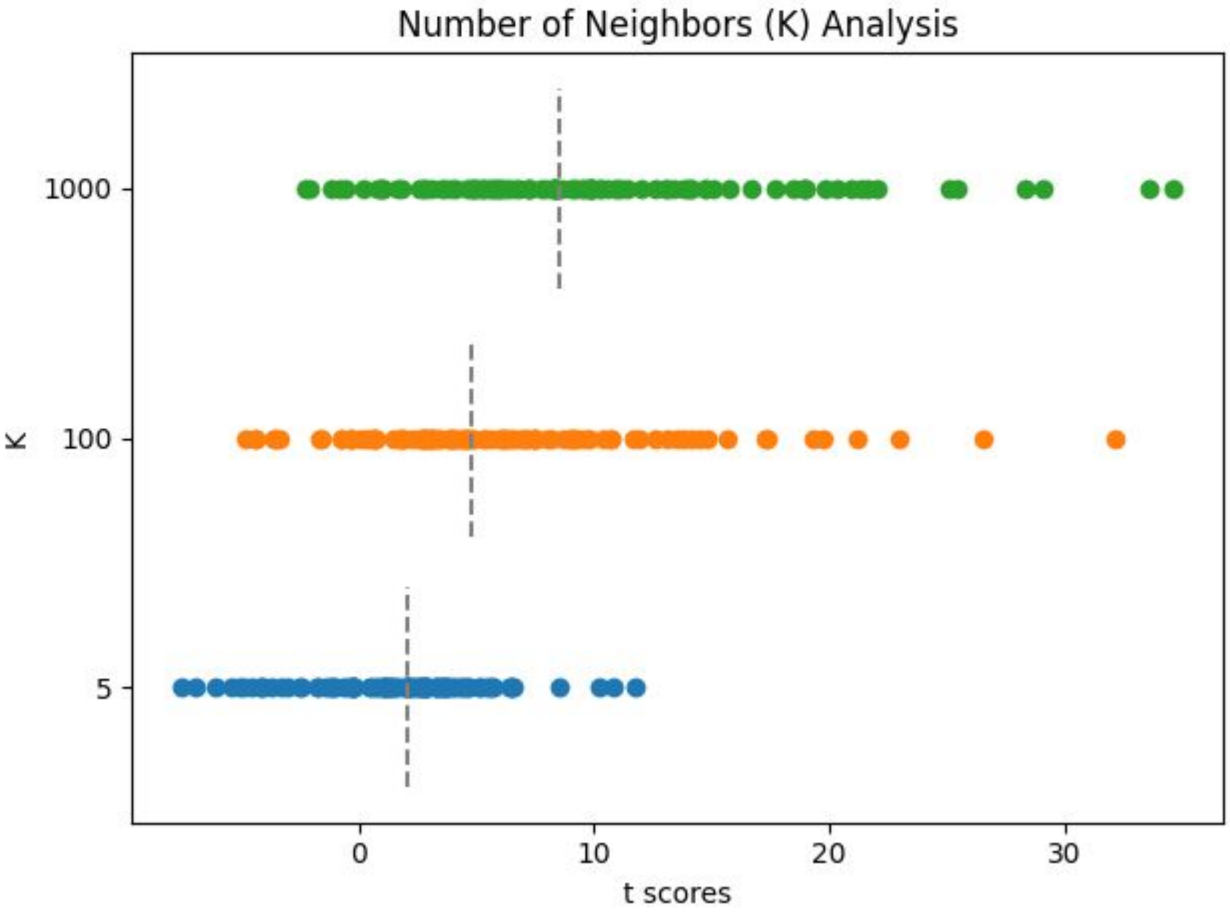
- Hours played is public on the review. Higher hours played could give credibility to the review, increasing the community's votes.
- The Steam algorithm which orders the reviews on its page could preference reviews which higher hours played.
- An individual who played the game for longer might put more effort into making a well-written review.


GROUP size analysis
signifigance level
Bin size
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
10
20
30
t scores

# Trial 2

$T > T_{critical} \approx 2.04$




Number of Neighbors (K) Analysis
1000
100
5
K
0
10
20
30
t scores

# Trial 2– K and Target Discretization Analysis

2nd Hyperparameter search for KNN. Focusing on the best group size

- GROUP = [5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100]
- 5 feature subsets
- 2 target Discretize strategies
- 3 choices for K

Log_Range Disc – discretize on equal ranges after taking the logarithm of hours_played.

| Target Disc | K | # of Classifies which are significantly better than the Majority Classifier (max=100) |
|---|---|---|
| Log_Range | 5 | 0 |
| Log_Range | 10 | 0 |
| Log_Range | 1000 | 0 |
| Freq | 5 | 53 |
| Freq | 100 | 76 |
| Freq | 1000 | 88 |

# Evaluation on the Test Set

Model:

- Feature: vote count
- Target disc: equal frequency
- Group: 8
- K = 5 and 1000

K = 5
Class Acc = 0.1322
Majority Acc = 0.1250

0.72% improvement over majority classifier

K = 1000
Class Acc = 0.1394
Majority Acc = 0.1256

1.38% improvement over majority classifier

# Random Forests

## Predicting Hours Played

# Random Forest Parameters

Four Legacy Groups

- Continuous All– 'helpful','funny','polarity','objectivity','n_words','n_sentences'
- Vote count– 'helpful','funny'
- TextBlob– ‘polarity’,'objectivity'
- TextBlob extra– ‘n_words’,‘n_sentences’,‘polarity’, 'objectivity'

Nine hopeful groups

1. game_name, overall_player_rating
2. publisher, overall_player_rating
3. developer, overall_player_rating
4. game_name, publisher, developer, overall_player_rating
5. recommendation, game_name, overall_player_rating
6. recommendation, publisher, overall_player_rating
7. recommendation, developer, overall_player_rating
8. recommendation, game_name, publisher, developer, overall_player_rating
9. All categorical AND continuous features.

Each trial is ran 30 times to provide enough samples for t-testing

Bootstrap factor is now 5.

Ensemble 100 trees
Max Features = # features

Criterion

- Entropy
- Gini
- Log
- Random

Target Disc

1. Uniform
2. K means
3. Frequency
4. Equal range on log(hours_played)

# **Random Forest Analysis**– Target Disc. and Split Criterion

| Target Disc | # pass t-tests<br>Max = 52 |
|---|---|
| Uniform | |
| K means | |
| Frequency | 49 |
| Equal Range on log(hours_played) | 0 |

Frequency is the only statistically justified choice for target discretization.

| Split Criterion | # pass t-tests<br>Max = 52 |
|---|---|
| Entropy | 11 |
| Gini | 12 |
| Log | 13 |
| Random | 13 |

Almost no variation in the number of passed t-tests are accounted for by the criterion choice.

# Random Forest Analysis– Pairwise T-Test on Feature Subsets

Max = 16

| | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | All Continuous | Textblob | Textblob Extra | Vote Count |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | 0 | 4 | 2 | 0 | 2 | 0 | 2 | 1 | 8 | 8 | 8 | 8 | 8 |
| 2. | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 8 | 8 | 8 | 8 | 7 |
| 3. | 0 | 2 | 0 | 0 | 1 | 0 | 1 | 1 | 8 | 8 | 8 | 8 | 7 |
| 4. | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 8 | 8 | 8 | 8 | 7 |
| 5. | 3 | 4 | 5 | 2 | 0 | 2 | 2 | 2 | 8 | 8 | 8 | 8 | 8 |
| 6. | 0 | 4 | 0 | 0 | 1 | 0 | 1 | 1 | 8 | 8 | 8 | 8 | 7 |
| 7. | 0 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 8 | 8 | 8 |
| 8. | 1 | 4 | 4 | 2 | 0 | 3 | 0 | 0 | 8 | 8 | 8 | 8 | 7 |
| 9. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 5 | 7 | 4 |
| All Continuous | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 2 | 0 |
| Textblob | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 4 | 0 |
| Textblob Extra | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 |
| Vote Count | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 8 | 8 | 8 | 0 |

# Evaluation on the Test Set

Model:

- Feature = group 8
    - Recommendation, game_name, publisher, developer, overall_player_rating
- Target disc = equal frequency
- Group = 8
- Criterion = Random

K = 5
Class Acc = 0.1322
Majority Acc = 0.1250

0.72% improvement over majority classifier

K = 1000
Class Acc = 0.1394
Majority Acc = 0.1256

1.38% improvement over majority classifier

# Random Forest

## Predicting Recommendation

# Predict Recommendation using Random Forests

<u>TextBlob Groups</u>

1. TextBlob 1 – 'n_words'
2. TextBlob 2 – 'n_sentences'
3. TextBlob 3 – 'polarity'
4. TextBlob 4 – 'subjectivity'
5. TextBlob_length – 'n_words', 'n_sentences'
6. TextBlob_sentiment – 'polarity', 'subjectivity'
7. TextBlob_pair1 – 'n_words', 'polarity'
8. TextBlob_pair2 – 'n_words', 'subjectivity'
9. TextBlob_pair3 – 'n_sentences', 'polarity'
10. TextBlob_pair4 – 'n_sentences', 'subjectivity'
11. TextBlob_all – 'polarity', 'subjectivity', 'n_words', 'n_sentences'

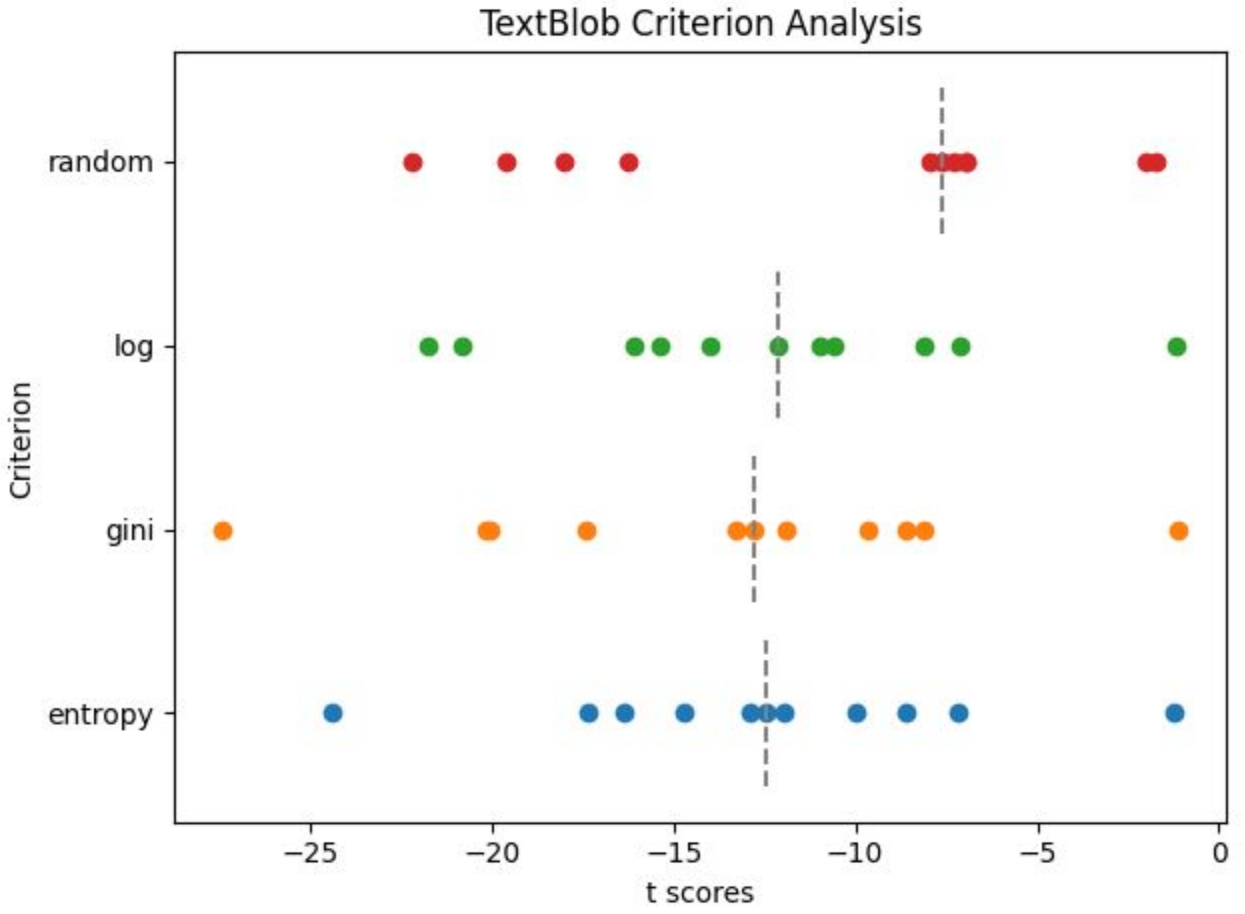
Bootstrap factor of 5.

Each trial is ran 30 times to provide enough samples for t-testing

Ensemble 100 trees
Max Features = # features

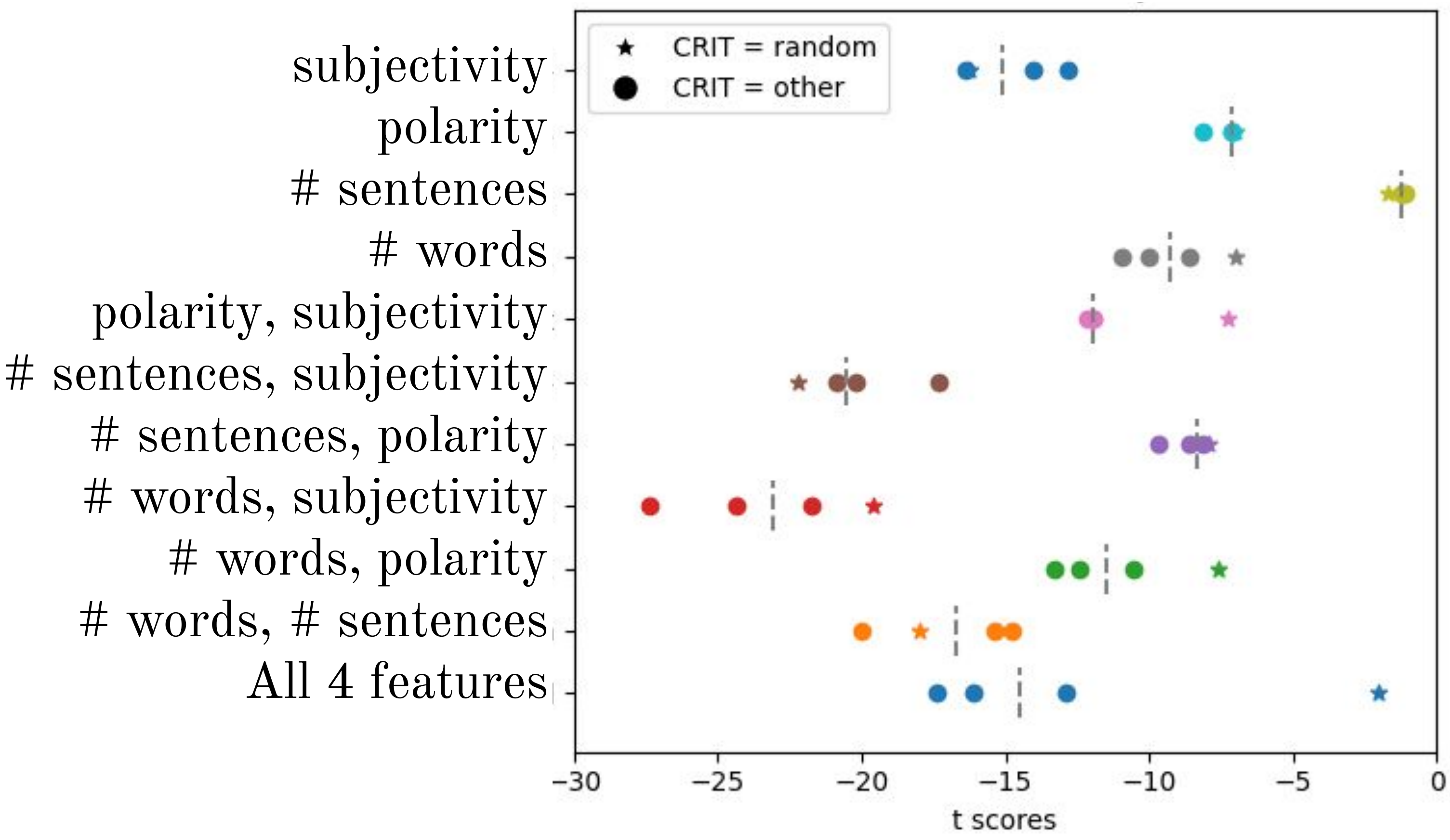
Criterion

- Entropy
- Gini
- Log
- Random




Across 11 feature subsets, splitting the tree using a random attribute maximizes validation performance. The other split criterion are deterministic, that is, each ensembled tree would have the same split decisions using the same bootstrapped data. Random criterion introduces stochasticity, allowing the ensembling to actually do something.

# Feature Subset Analysis



# Predict Recommendation using Random Forests

Nine hopeful groups

1. game_name, overall_player_rating
2. publisher, overall_player_rating
3. developer, overall_player_rating
4. game_name, publisher, developer, overall_player_rating
5. hours_played, game_name, overall_player_rating
6. hours_played, publisher, overall_player_rating
7. hours_played, developer, overall_player_rating
8. hours_played, game_name, publisher, developer, overall_player_rating
9. hours_played, game_name, publisher, developer, overall_player_rating, Helpful, funny, n_words, n_sentences, polarity, subjectivity

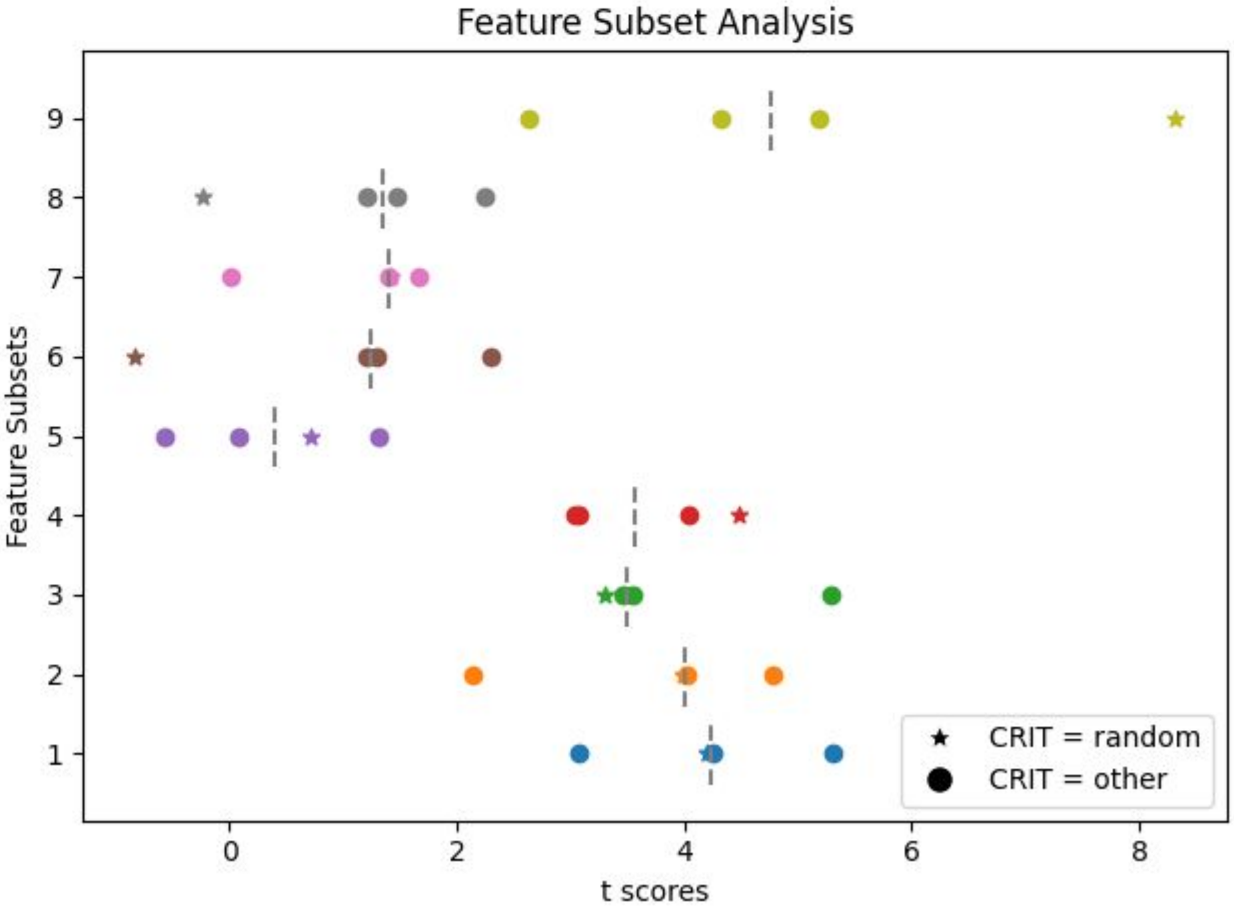
Bootstrap factor of 5.

Each trial is ran 30 times to provide enough samples for t-testing

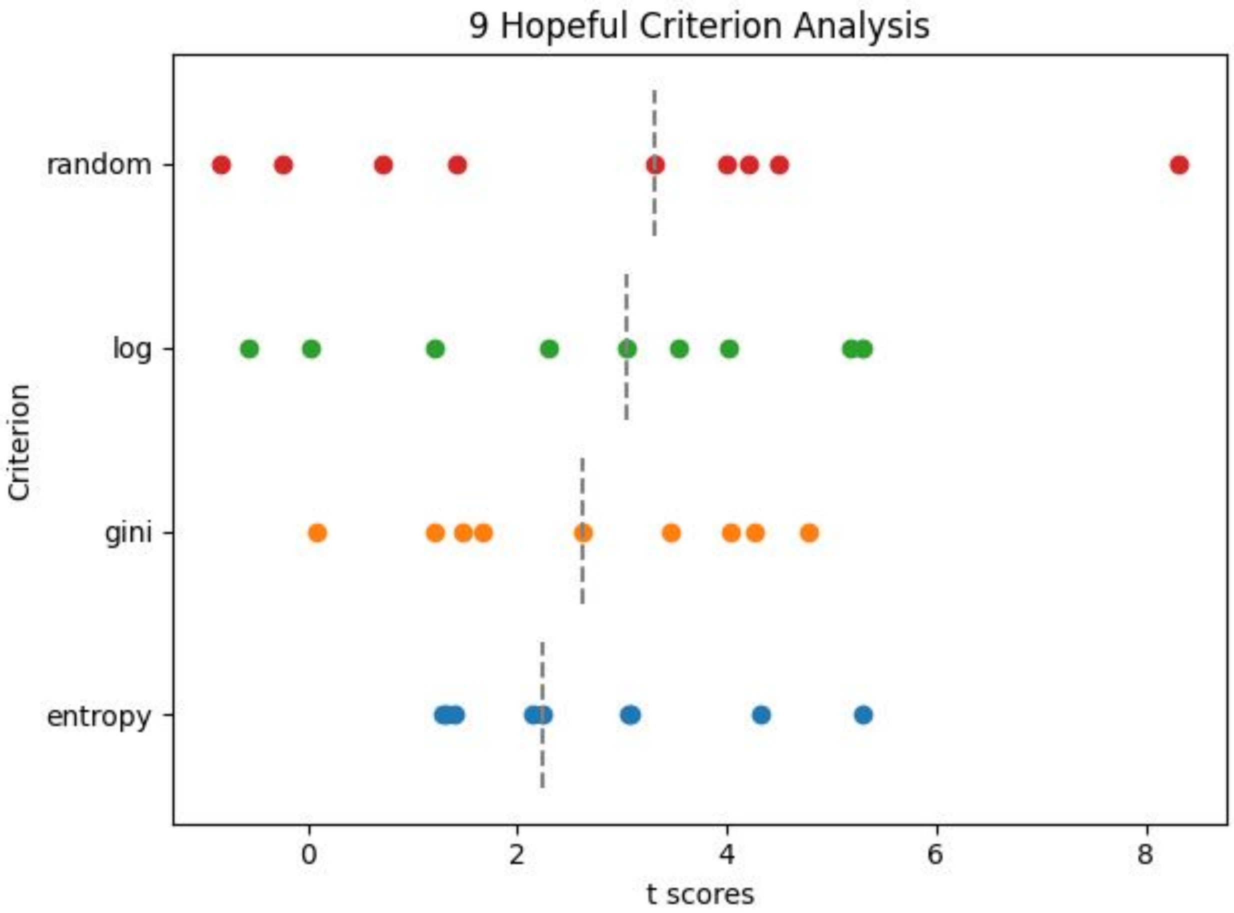
Ensemble 100 trees
Max Features = # features

Criterion

- Entropy
- Gini
- Log
- Random


Feature Subset Analysis
Feature Subsets
t scores
CRIT = random
CRIT = other
9
8
7
6
5
4
3
2
1
0
2
4
6
8

analysis


9 Hopeful Criterion Analysis
random
log
gini
entropy
Criterion
0
2
4
6
8
t scores

analysis

# Incorporating Helpful and Funny Votes for Final Feature Subset Test

<u>Final Set</u>

1. Helpful
2. Funny
3. Helpful, Funny
4. Helpful, funny, n_sentences
5. Helpful, funny, n_words, n_sentences, polarity, subjectivity
6. 

– best performing among TextBlob Feature Subsets

# Evaluation on the Test Set

Model:

- Features
  - Recommendation, game_name, publisher, developer, overall_player_rating
- Criterion = Random

# Conclusions