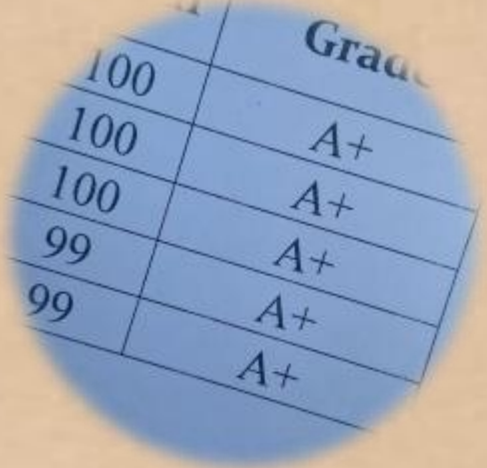


	Grade
100	
100	A+
100	A+
99	A+
99	A+
	A+

Welcome to Animal Crossing™ New Horizons

Classification: Positive + Negative Reviews

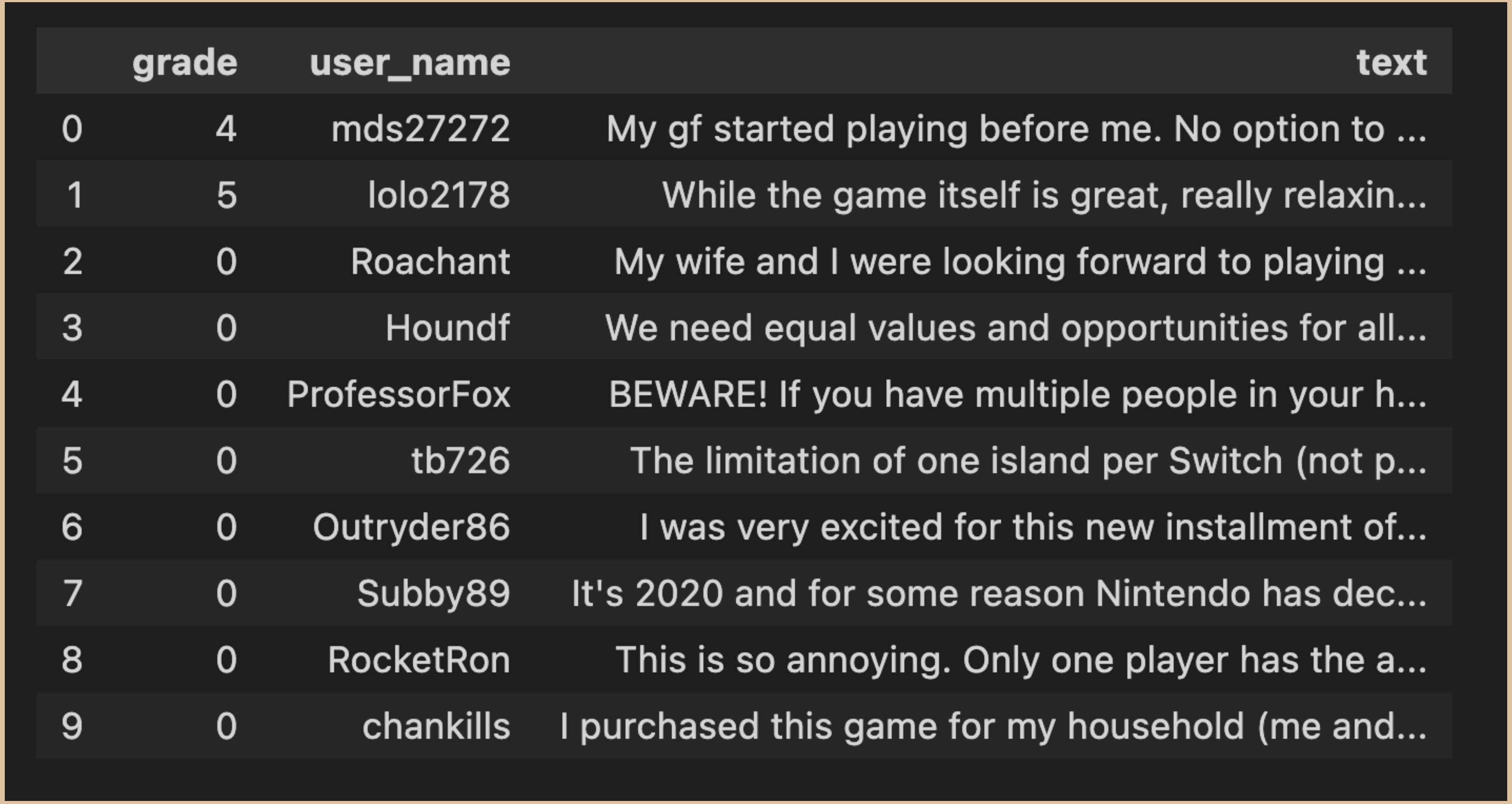
ML Project - *Final Results*

The Problem:

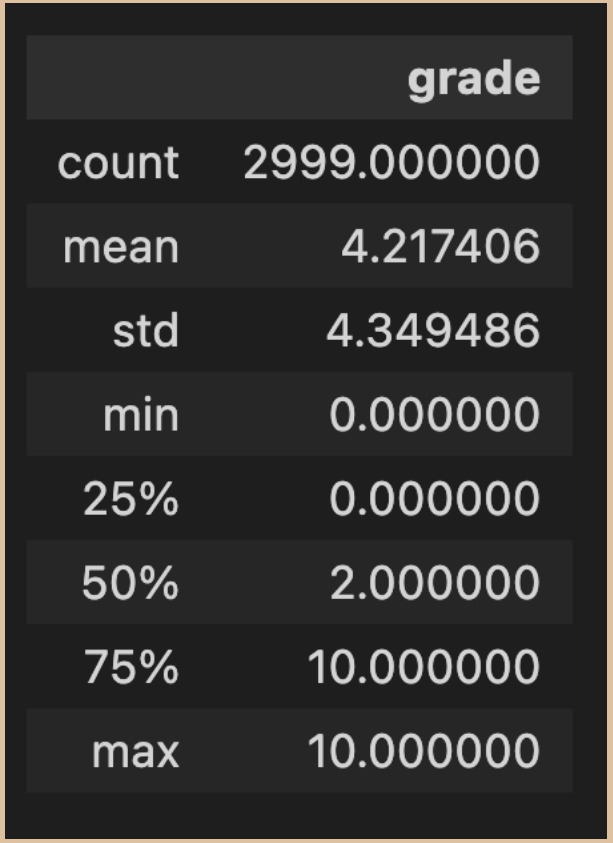
- Classifying user reviews for the video game Animal Crossing New Horizons as holding either negative or positive feelings towards based on the frequency of certain words appearing the review.
- The cost of time and human resources can become very high to sort through every single review manually, and so this tedious task can be made easier for people by automating the detection of whether a review is positive or negative.

employees without ml working overtime:

The Data:



	grade	user_name	text
0	4	mds27272	My gf started playing before me. No option to ...
1	5	lolo2178	While the game itself is great, really relaxin...
2	0	Roachant	My wife and I were looking forward to playing ...
3	0	Houndf	We need equal values and opportunities for all...
4	0	ProfessorFox	BEWARE! If you have multiple people in your h...
5	0	tb726	The limitation of one island per Switch (not p...
6	0	Outryder86	I was very excited for this new installment of...
7	0	Subby89	It's 2020 and for some reason Nintendo has dec...
8	0	RocketRon	This is so annoying. Only one player has the a...
9	0	chankills	I purchased this game for my household (me and...



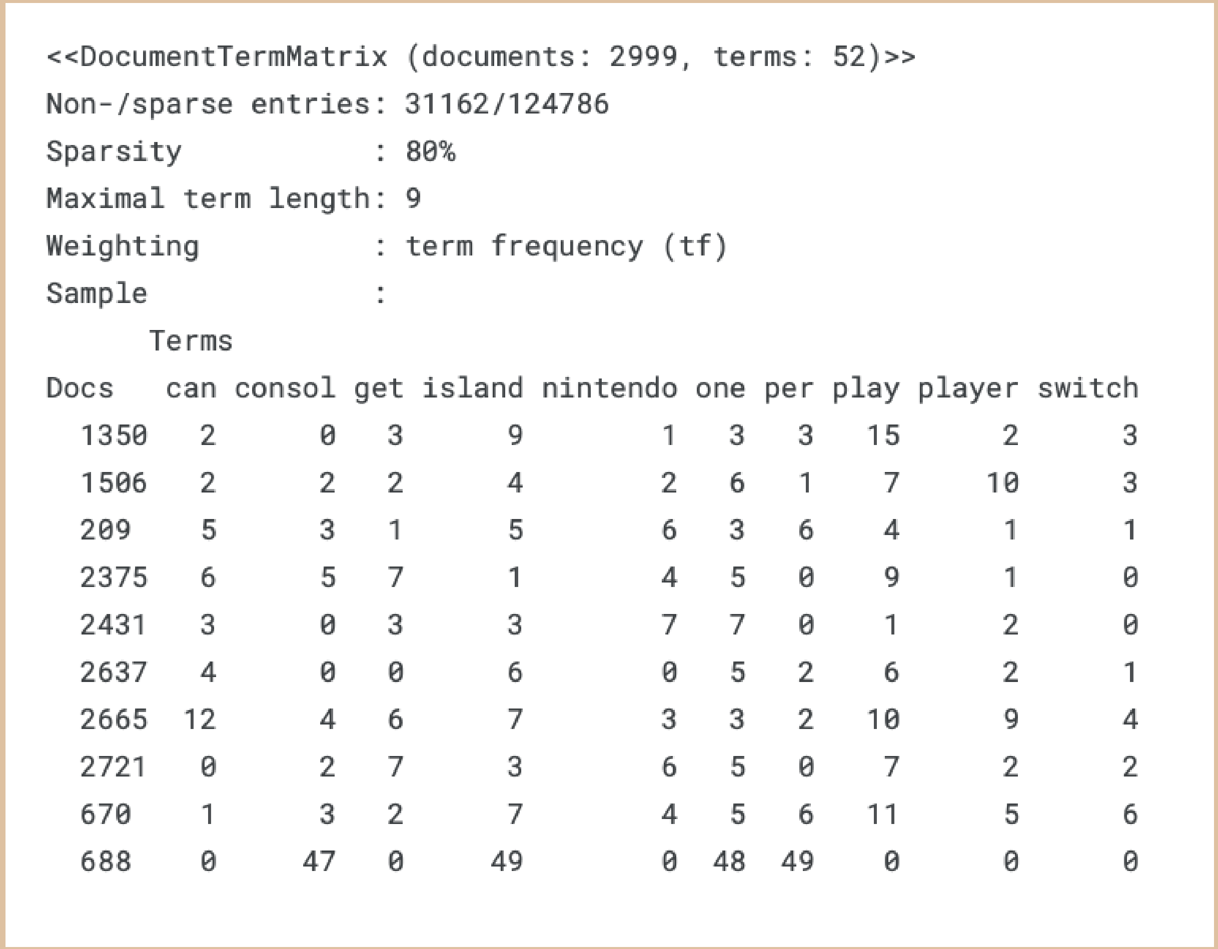
	grade
count	2999.000000
mean	4.217406
std	4.349486
min	0.000000
25%	0.000000
50%	2.000000
75%	10.000000
max	10.000000

ACNH User Review Dataset:
https://www.kaggle.com/datasets/jessemostipak/animal-crossing/data

Descriptive Stats

A Case Study:

- Phyllis Tay on Kaggle split the dataset into positive and negative reviews based on each user's grade: 0-5 as negative and 6-10 positive.
- The text data was preprocessed into a numerical form with a Document Term Matrix.
- Dimension reduction was performed by removing sparse words or words that don't show up much across the dataset.
- The ML algorithm used was hierarchical clustering to classify the words from the Document Term Matrix into positive or negative sentiment groups.



```
<<DocumentTermMatrix (documents: 2999, terms: 52)>>
Non-/sparse entries: 31162/124786
Sparsity           : 80%
Maximal term length: 9
Weighting          : term frequency (tf)
Sample             :
      Terms
Docs   can consol get island nintendo one per play player switch
  1350   2      0   3      9        1   3   3   15      2      3
  1506   2      2   2      4        2   6   1    7     10      3
  209    5      3   1      5        6   3   6    4      1      1
  2375   6      5   7      1        4   5   0    9      1      0
  2431   3      0   3      3        7   7   0    1      2      0
  2637   4      0   0      6        0   5   2    6      2      1
  2665  12      4   6      7        3   3   2   10      9      4
  2721   0      2   7      3        6   5   0    7      2      2
  670    1      3   2      7        4   5   6   11      5      6
  688    0     47   0     49        0  48  49    0      0      0
```

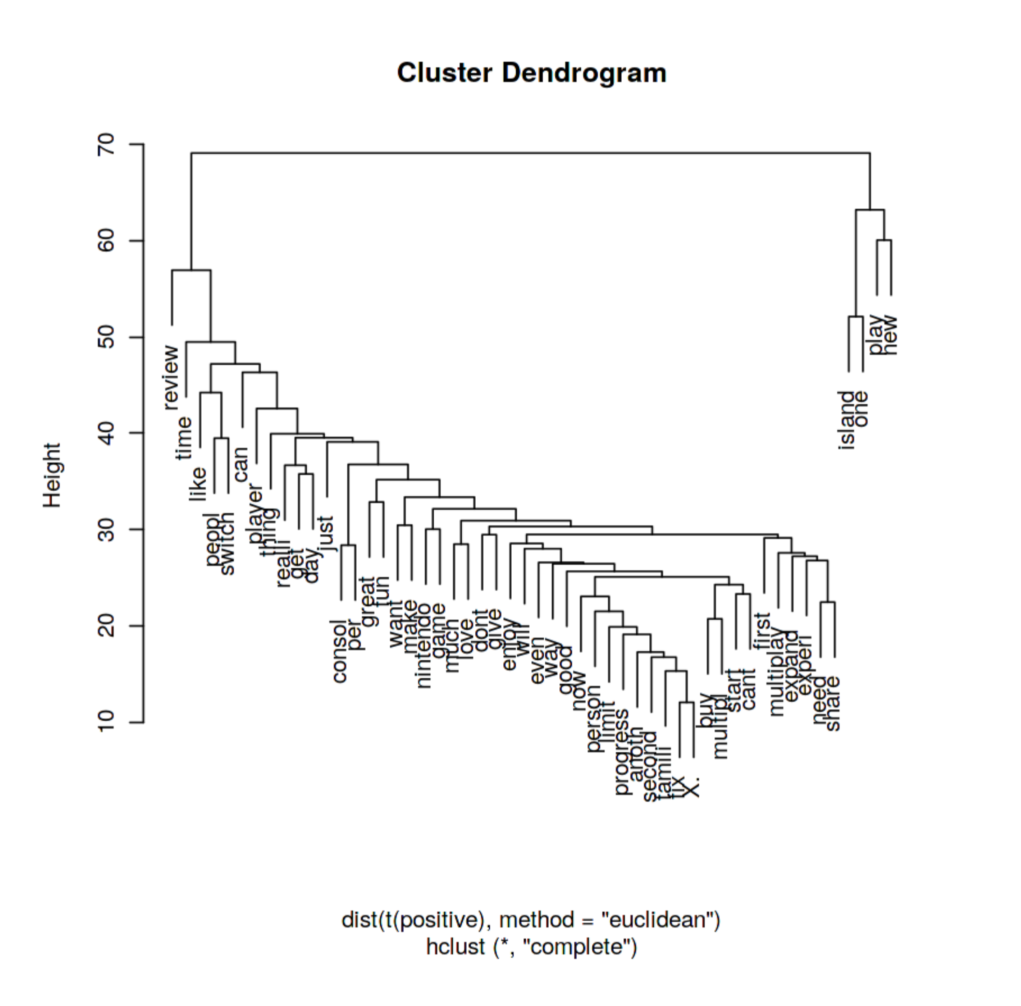
Dataset analysis by Phyllis Tay:

https://www.kaggle.com/code/phyllistay/acnh-text-analytics

https://public.tableau.com/app/profile/phyllis.tay/viz/AnimalCrossingNewHorizonsReviews/ACNHReviews

A Case Study:

- Phyllis Tay on Kaggle split the dataset into positive and negative reviews based on each user's grade: 0-5 as negative and 6-10 positive.
- The text data was preprocessed into a numerical form with a Document Term Matrix.
- Dimension reduction was performed by removing sparse words or words that don't show up much across the dataset.
- The ML algorithm used was hierarchical clustering to classify the words from the Document Term Matrix into positive or negative sentiment groups.

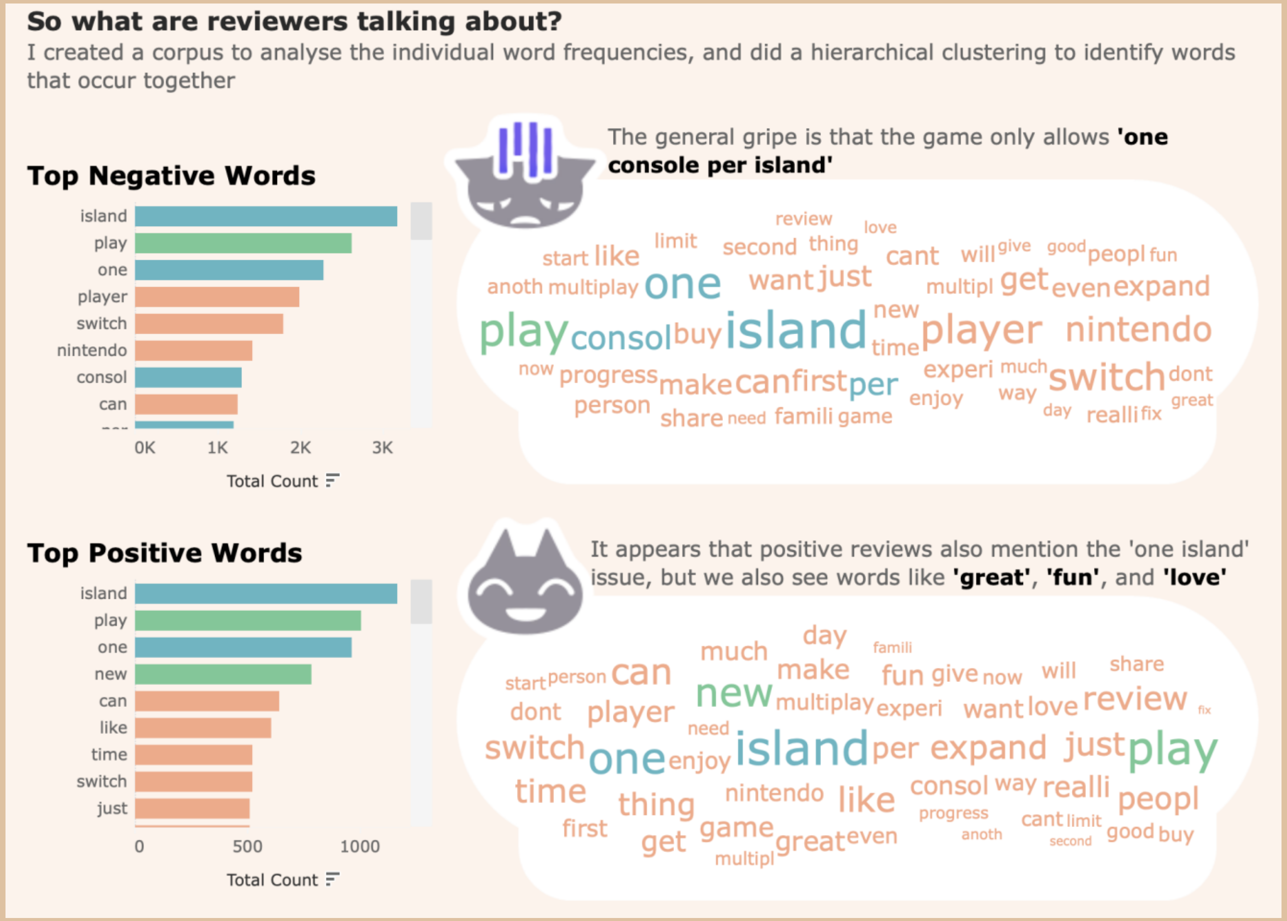


Dataset analysis by Phyllis Tay:

https://www.kaggle.com/code/phyllistay/acnh-text-analytics

https://public.tableau.com/app/profile/phyllis.tay/viz/AnimalCrossingNewHorizonsReviews/ACNHReviews

A Case Study:



So what are reviewers talking about?

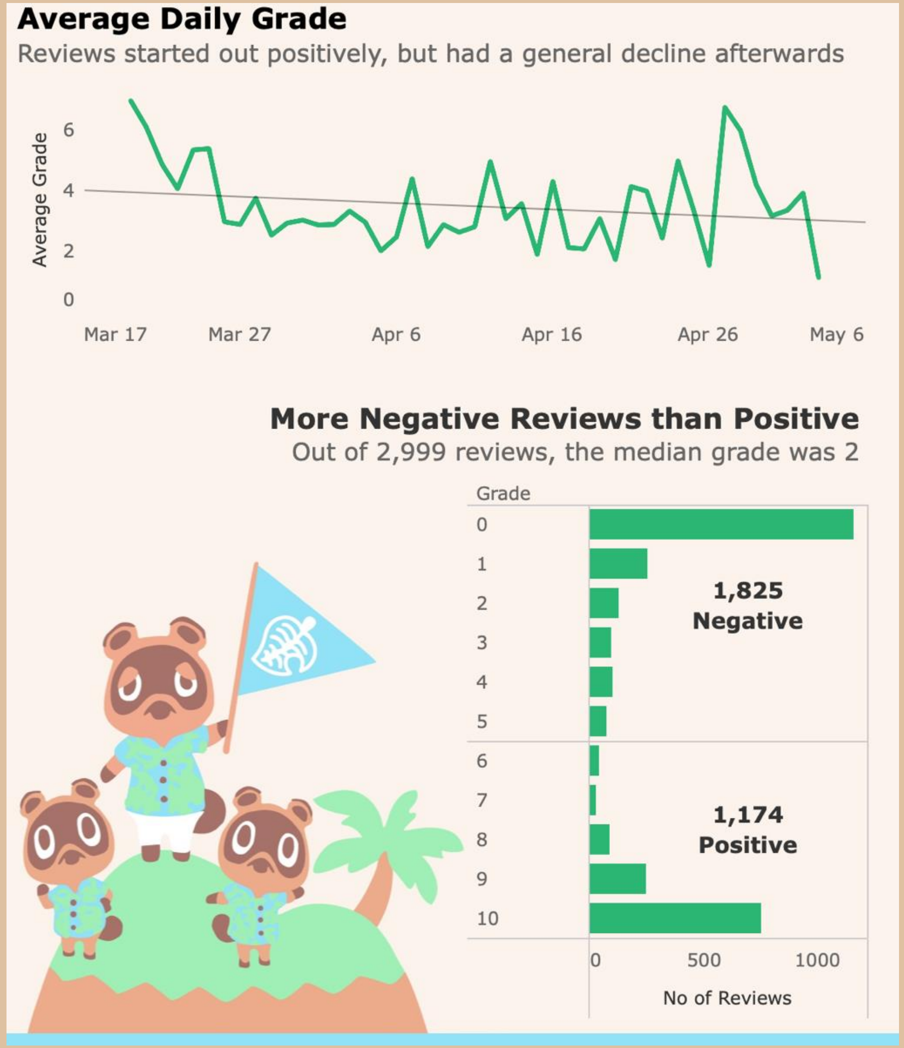
I created a corpus to analyse the individual word frequencies, and did a hierarchical clustering to identify words that occur together

Top Negative Words

The general gripe is that the game only allows **'one console per island'**

Top Positive Words

It appears that positive reviews also mention the 'one island' issue, but we also see words like **'great'**, **'fun'**, and **'love'**



Average Daily Grade

Reviews started out positively, but had a general decline afterwards

More Negative Reviews than Positive

Out of 2,999 reviews, the median grade was 2

Dataset analysis by Phyllis Tay:

https://www.kaggle.com/code/phyllistay/acnh-text-analytics

https://public.tableau.com/app/profile/phyllis.tay/viz/AnimalCrossingNewHorizonsReviews/ACNHReviews

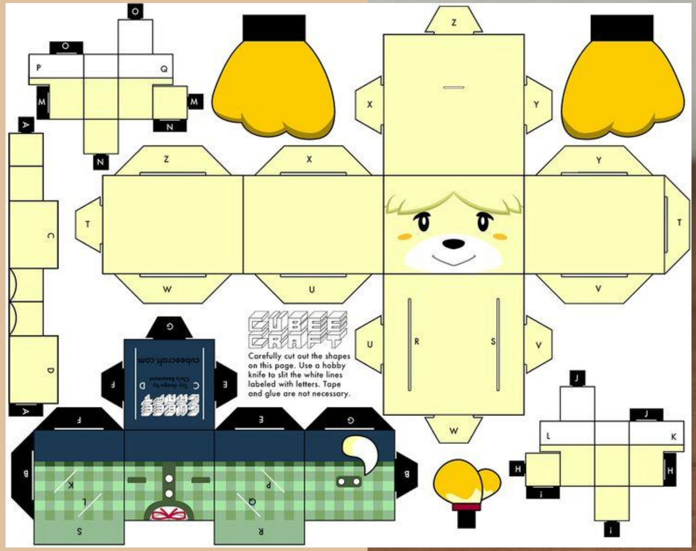
My Process:

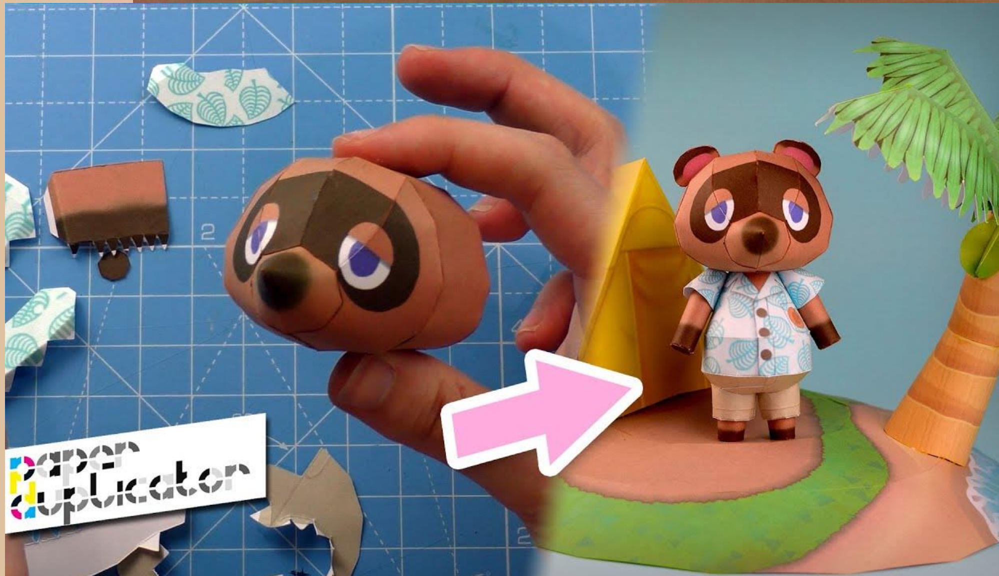
- Clustering the data based on the grade attached to the user's review. A word clouds were generated to see what words appear in each cluster.
- Training a model on random forest classification to predict a grade from 0 to 10, then running the model to predict on test data.
- This was then adjusted to predict only 0 for negative or 1 for positive.

- The final part was never executed, but I hoped the first two steps were able to set up well for this:
  - Mapping the frequency of each word in the vocabulary onto its count in the reviews clustered as positive or negative.

K-Fold: Fold 1 of 2

- K-Fold cross validation to split the data in half for two folds.
- More patterns were to distinguish clusters by with a smaller portion of the dataset rather than in working with the entire dataset at once.

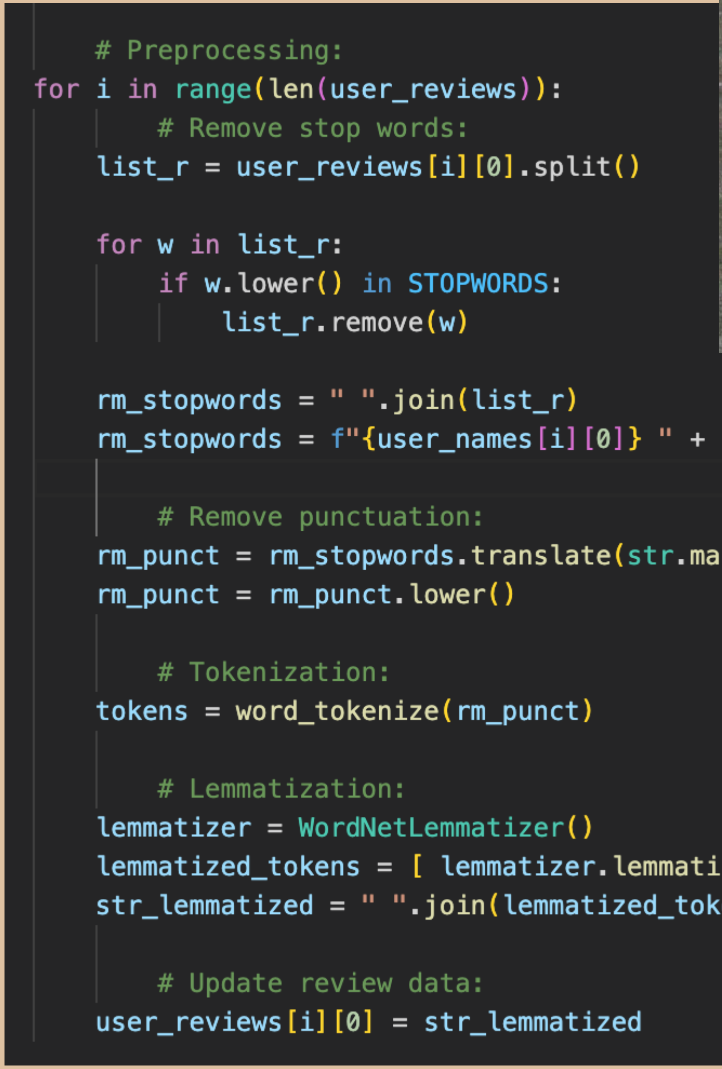

CUBEECRAFT
Carefully cut out the shapes on this page. Use a hobby knife to slit the white lines labeled with letters. Tape and glue are not necessary.



Preprocessing 1: Words

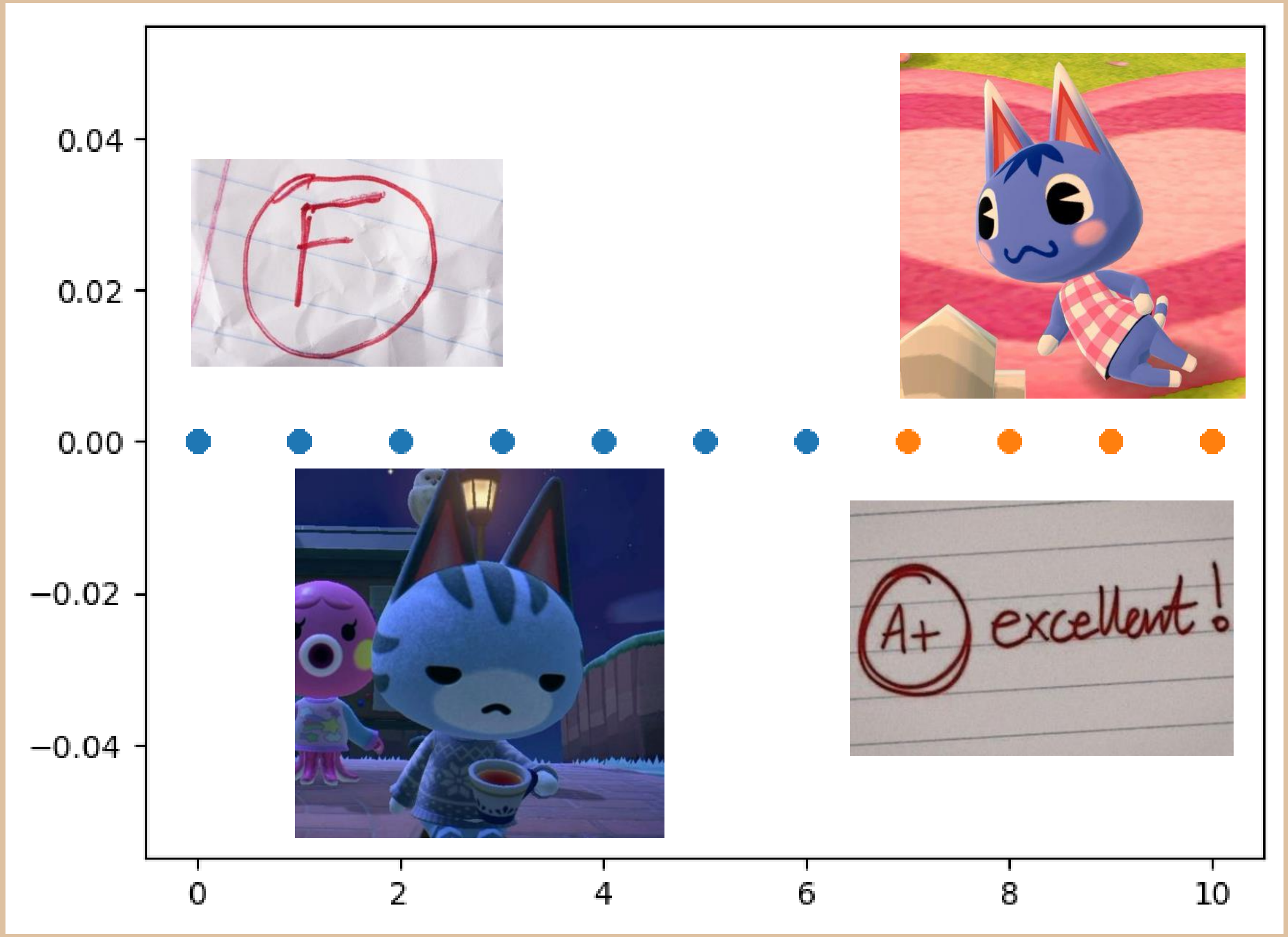
Preprocess review data:

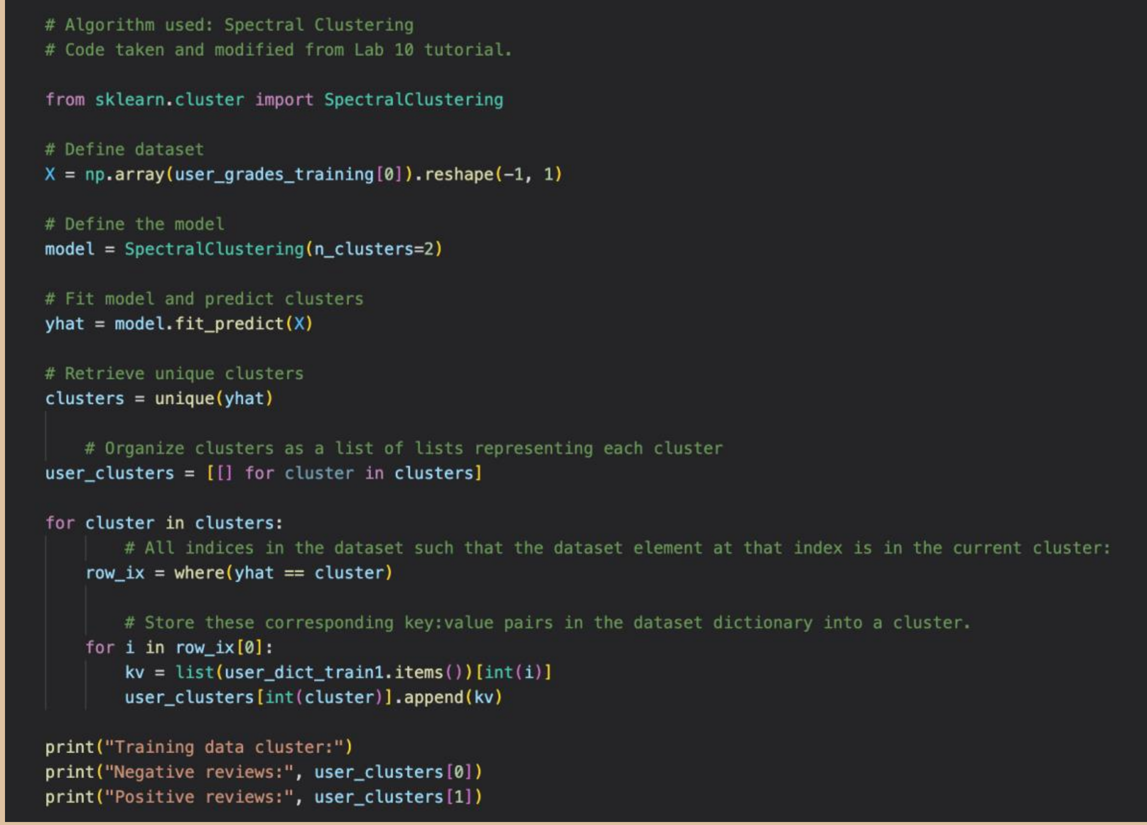
- **Remove stopwords**
- **Remove punctuation**
- **Tokenize reviews**
- **Lemmatize reviews**



```
    # Preprocessing:
for i in range(len(user_reviews)):
        # Remove stop words:
    list_r = user_reviews[i][0].split()

    for w in list_r:
        if w.lower() in STOPWORDS:
            list_r.remove(w)

    rm_stopwords = " ".join(list_r)
    rm_stopwords = f"{user_names[i][0]} " + rm_stopwords

        # Remove punctuation:
    rm_punct = rm_stopwords.translate(str.maketrans('', '', string.punctuation))
    rm_punct = rm_punct.lower()

        # Tokenization:
    tokens = word_tokenize(rm_punct)

        # Lemmatization:
    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [ lemmatizer.lemmatize(token) for token in tokens ]
    str_lemmatized = " ".join(lemmatized_tokens)

        # Update review data:
    user_reviews[i][0] = str_lemmatized
```

Initial Clustering





```
# Algorithm used: Spectral Clustering
# Code taken and modified from Lab 10 tutorial.

from sklearn.cluster import SpectralClustering

# Define dataset
X = np.array(user_grades_training[0]).reshape(-1, 1)

# Define the model
model = SpectralClustering(n_clusters=2)

# Fit model and predict clusters
yhat = model.fit_predict(X)

# Retrieve unique clusters
clusters = unique(yhat)

    # Organize clusters as a list of lists representing each cluster
user_clusters = [[] for cluster in clusters]

for cluster in clusters:
        # All indices in the dataset such that the dataset element at that index is in the current cluster:
    row_ix = where(yhat == cluster)

        # Store these corresponding key:value pairs in the dataset dictionary into a cluster.
    for i in row_ix[0]:
        kv = list(user_dict_train1.items())[int(i)]
        user_clusters[int(cluster)].append(kv)

print("Training data cluster:")
print("Negative reviews:", user_clusters[0])
print("Positive reviews:", user_clusters[1])
```

Spectral clustering predicted based on grade information for each review.

Negative Reviews:

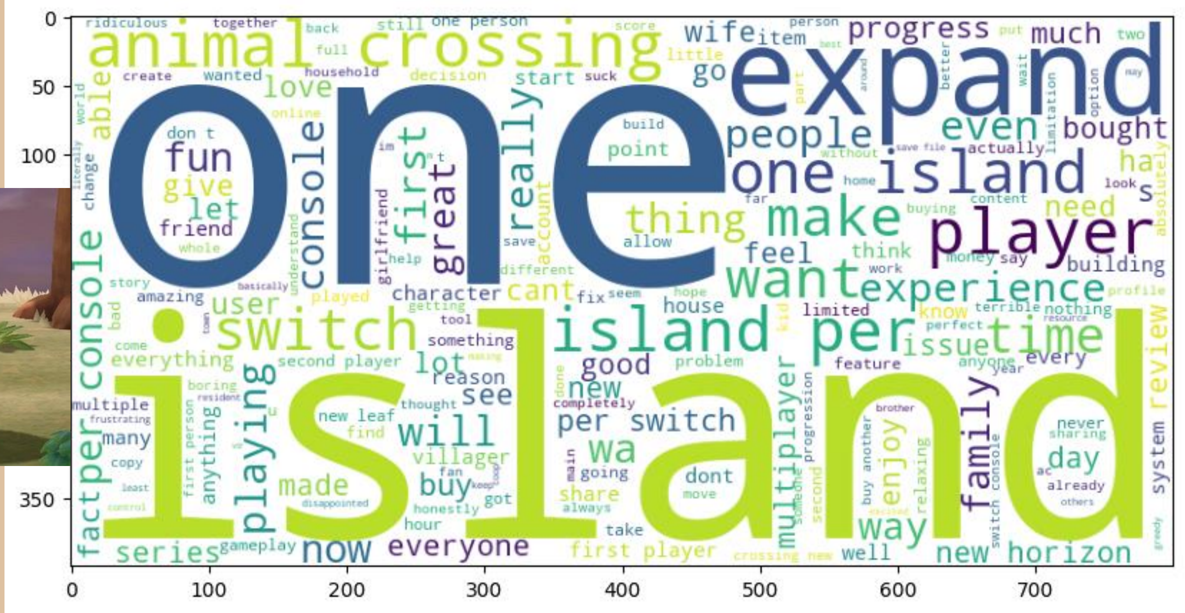
- **Grade 0-6**

Positive Reviews:

- **Grade 7-10**

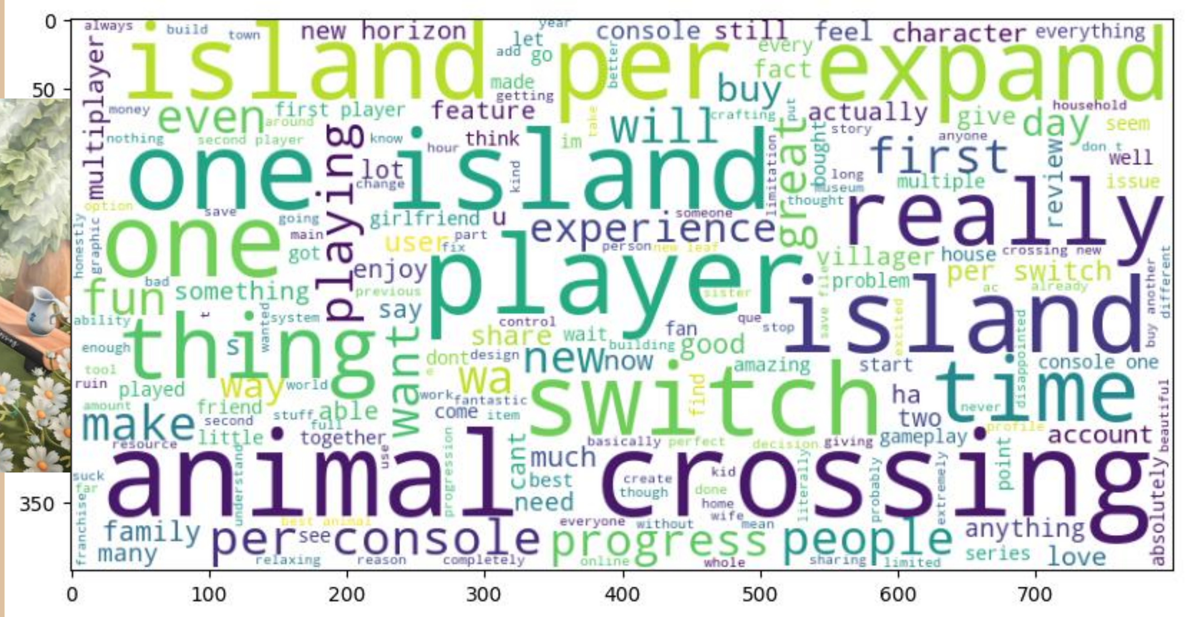
Initial Clustering

- The cluster for the negative reviews has and extremely high frequency for "one", "island", "player", and "expand”.
- The cluster for positive reviews contains a more even distribution across the words in its word cloud.



Negative reviews

- More distinguished than in Results: Stage 1
- Tokenization + Lemmatization in previous slides generalized the data more.
- K-Fold focuses on a smaller portion of data at a time.

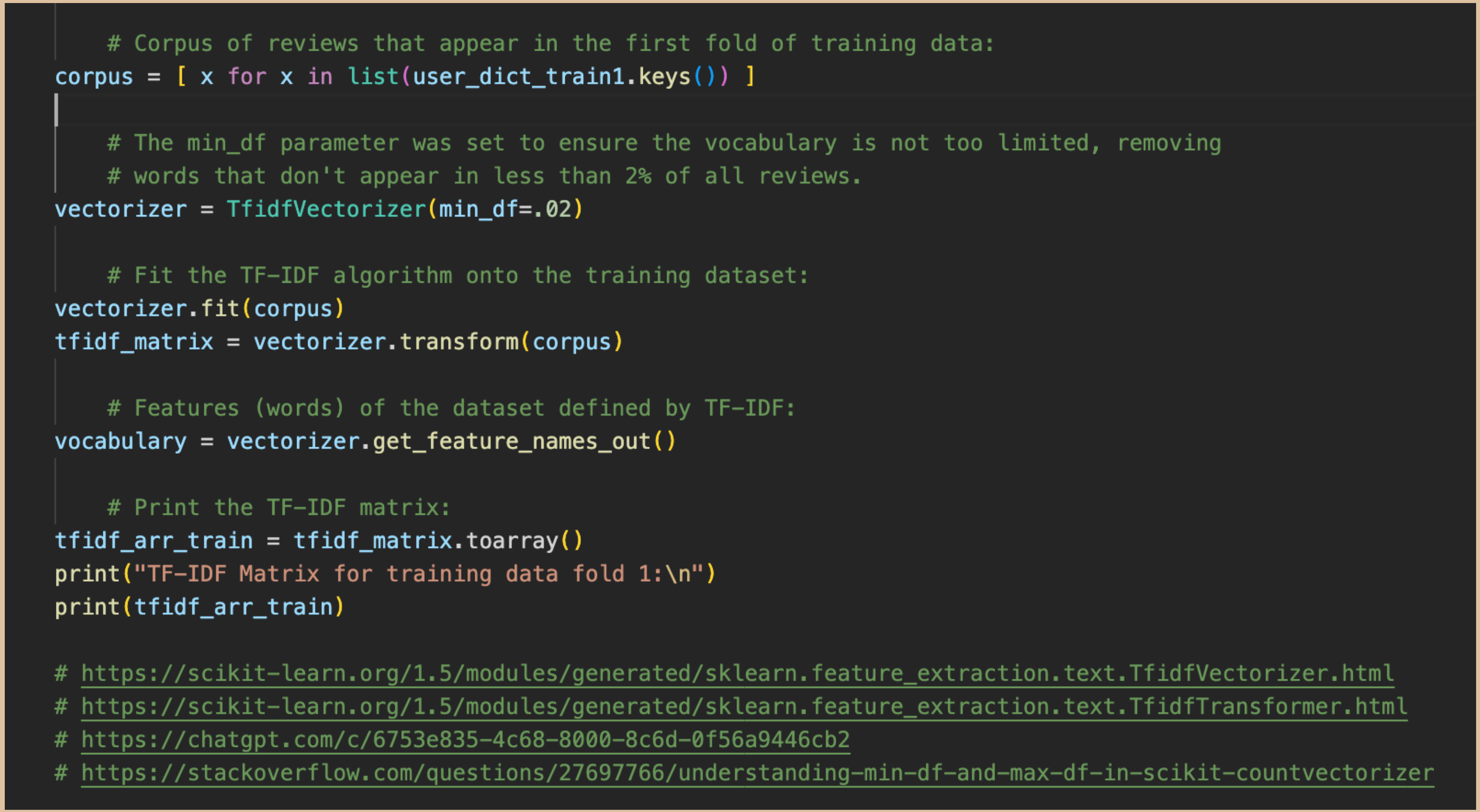


Positive reviews

Preprocessing 2: TF-IDF

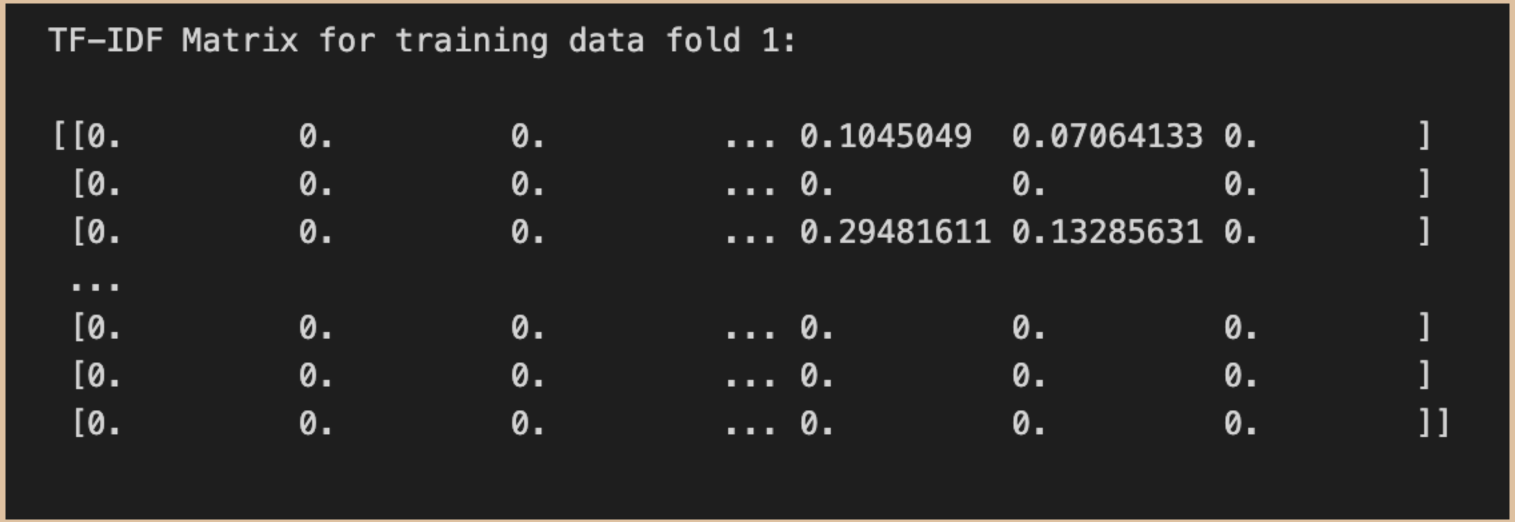
- **Improve upon Results Stage 1 by classifying based on the reviews' text, and not their grades.**
- **Text data as numerical representation with Term Frequency**
- **Defines each word in the dataset's vocabulary as a feature.**
- **Computes a matrix representing each review as a vector.**
- **Each element in the vector represents one feature or vocabulary word and the frequency it appears in this review.**

Preprocessing 2: TF-IDF

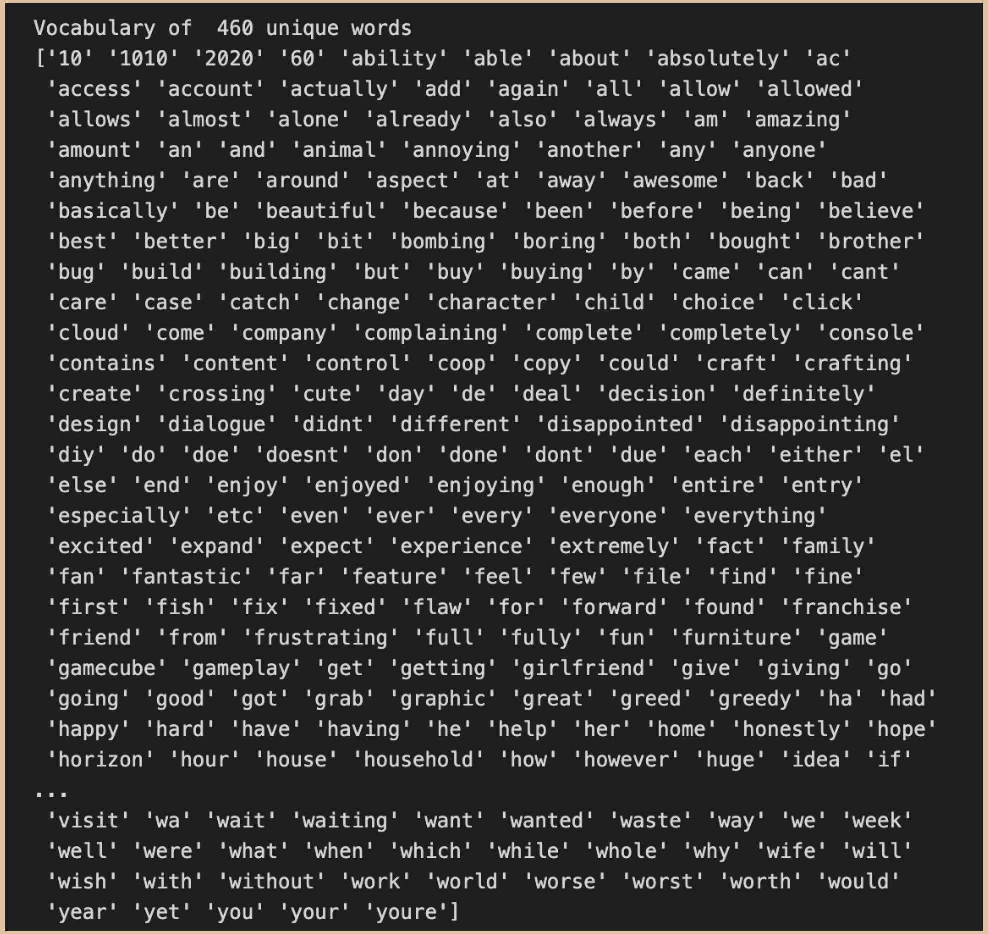


```
    # Corpus of reviews that appear in the first fold of training data:
corpus = [ x for x in list(user_dict_train1.keys()) ]

    # The min_df parameter was set to ensure the vocabulary is not too limited, removing
    # words that don't appear in less than 2% of all reviews.
vectorizer = TfidfVectorizer(min_df=.02)

    # Fit the TF-IDF algorithm onto the training dataset:
vectorizer.fit(corpus)
tfidf_matrix = vectorizer.transform(corpus)

    # Features (words) of the dataset defined by TF-IDF:
vocabulary = vectorizer.get_feature_names_out()

    # Print the TF-IDF matrix:
tfidf_arr_train = tfidf_matrix.toarray()
print("TF-IDF Matrix for training data fold 1:\n")
print(tfidf_arr_train)

# https://scikit-learn.org/1.5/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
# https://scikit-learn.org/1.5/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html
# https://chatgpt.com/c/6753e835-4c68-8000-8c6d-0f56a9446cb2
# https://stackoverflow.com/questions/27697766/understanding-min-df-and-max-df-in-scikit-countvectorizer
```

Preprocessing 2: TF-IDF

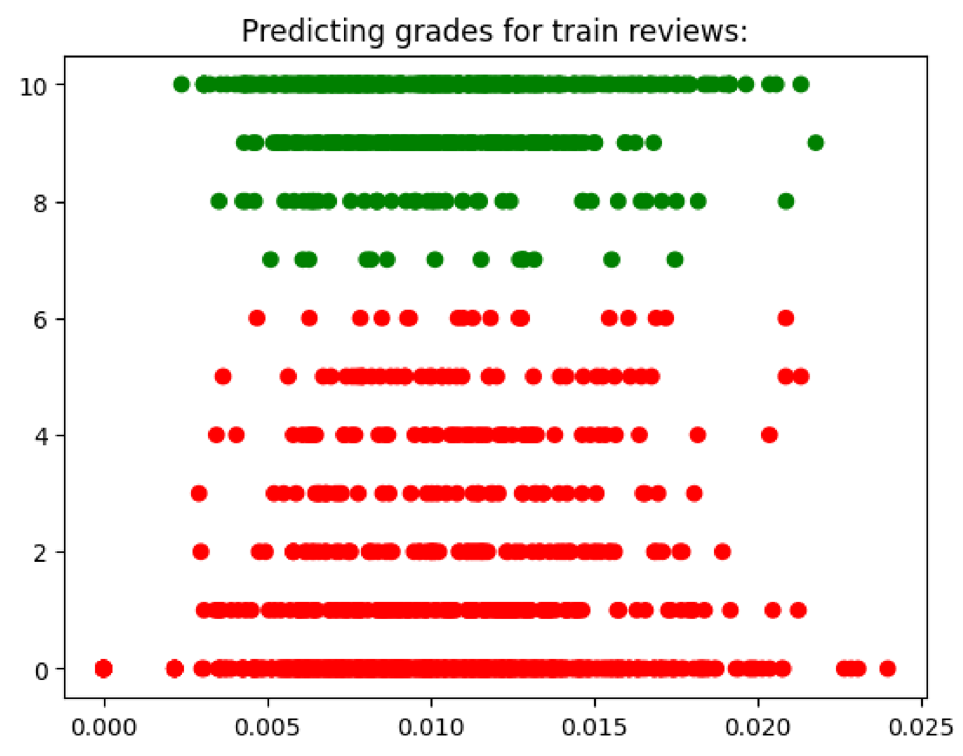


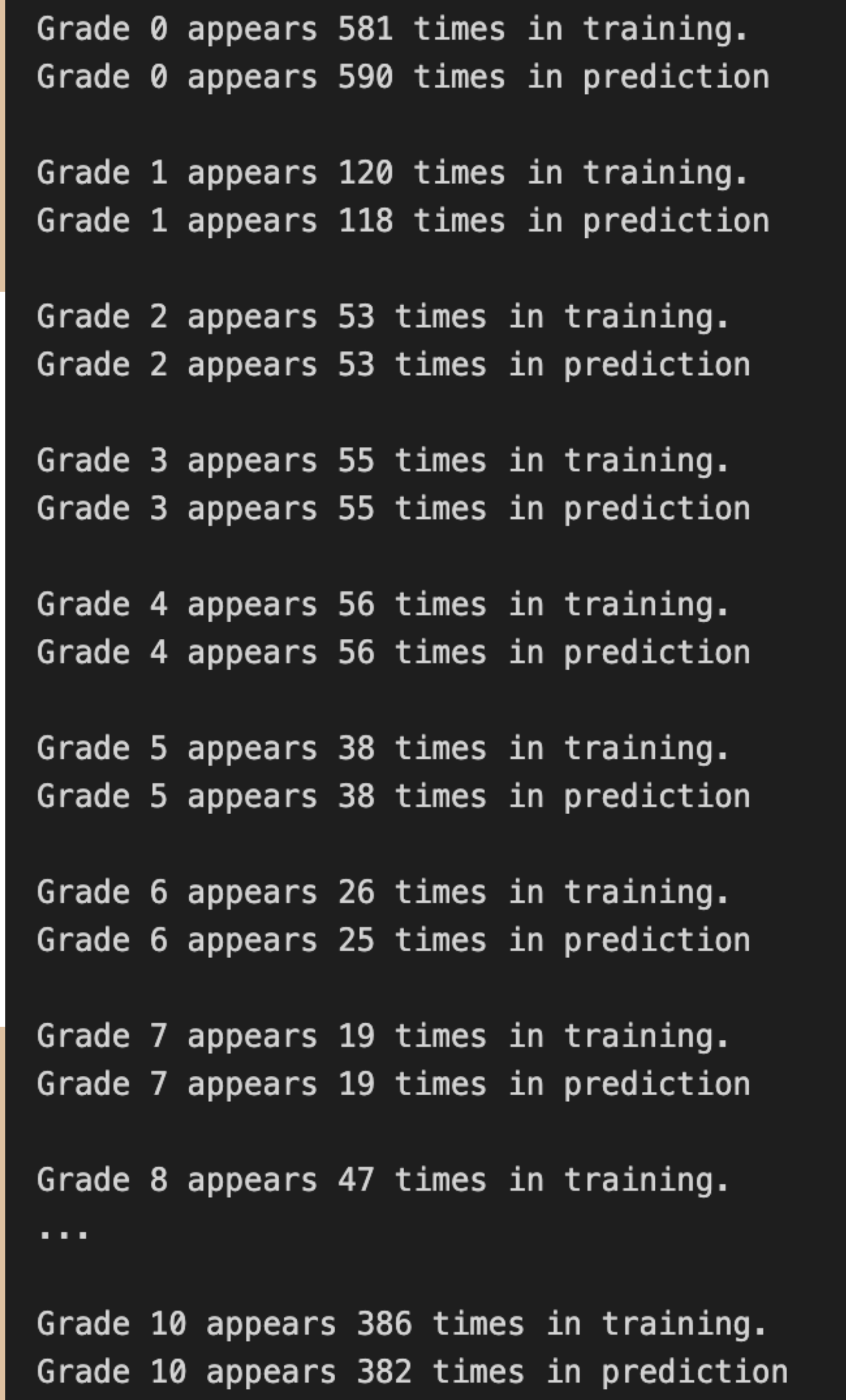
```
TF-IDF Matrix for training data fold 1:

[[0.         0.         0.         ... 0.1045049  0.07064133 0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.29481611 0.13285631 0.        ]
 ...
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]]
```



```
Vocabulary of  460 unique words
['10' '1010' '2020' '60' 'ability' 'able' 'about' 'absolutely' 'ac'
 'access' 'account' 'actually' 'add' 'again' 'all' 'allow' 'allowed'
 'allows' 'almost' 'alone' 'already' 'also' 'always' 'am' 'amazing'
 'amount' 'an' 'and' 'animal' 'annoying' 'another' 'any' 'anyone'
 'anything' 'are' 'around' 'aspect' 'at' 'away' 'awesome' 'back' 'bad'
 'basically' 'be' 'beautiful' 'because' 'been' 'before' 'being' 'believe'
 'best' 'better' 'big' 'bit' 'bombing' 'boring' 'both' 'bought' 'brother'
 'bug' 'build' 'building' 'but' 'buy' 'buying' 'by' 'came' 'can' 'cant'
 'care' 'case' 'catch' 'change' 'character' 'child' 'choice' 'click'
 'cloud' 'come' 'company' 'complaining' 'complete' 'completely' 'console'
 'contains' 'content' 'control' 'coop' 'copy' 'could' 'craft' 'crafting'
 'create' 'crossing' 'cute' 'day' 'de' 'deal' 'decision' 'definitely'
 'design' 'dialogue' 'didnt' 'different' 'disappointed' 'disappointing'
 'diy' 'do' 'doe' 'doesnt' 'don' 'done' 'dont' 'due' 'each' 'either' 'el'
 'else' 'end' 'enjoy' 'enjoyed' 'enjoying' 'enough' 'entire' 'entry'
 'especially' 'etc' 'even' 'ever' 'every' 'everyone' 'everything'
 'excited' 'expand' 'expect' 'experience' 'extremely' 'fact' 'family'
 'fan' 'fantastic' 'far' 'feature' 'feel' 'few' 'file' 'find' 'fine'
 'first' 'fish' 'fix' 'fixed' 'flaw' 'for' 'forward' 'found' 'franchise'
 'friend' 'from' 'frustrating' 'full' 'fully' 'fun' 'furniture' 'game'
 'gamecube' 'gameplay' 'get' 'getting' 'girlfriend' 'give' 'giving' 'go'
 'going' 'good' 'got' 'grab' 'graphic' 'great' 'greed' 'greedy' 'ha' 'had'
 'happy' 'hard' 'have' 'having' 'he' 'help' 'her' 'home' 'honestly' 'hope'
 'horizon' 'hour' 'house' 'household' 'how' 'however' 'huge' 'idea' 'if'
...
 'visit' 'wa' 'wait' 'waiting' 'want' 'wanted' 'waste' 'way' 'we' 'week'
 'well' 'were' 'what' 'when' 'which' 'while' 'whole' 'why' 'wife' 'will'
 'wish' 'with' 'without' 'work' 'world' 'worse' 'worst' 'worth' 'would'
 'year' 'yet' 'you' 'your' 'youre']
```

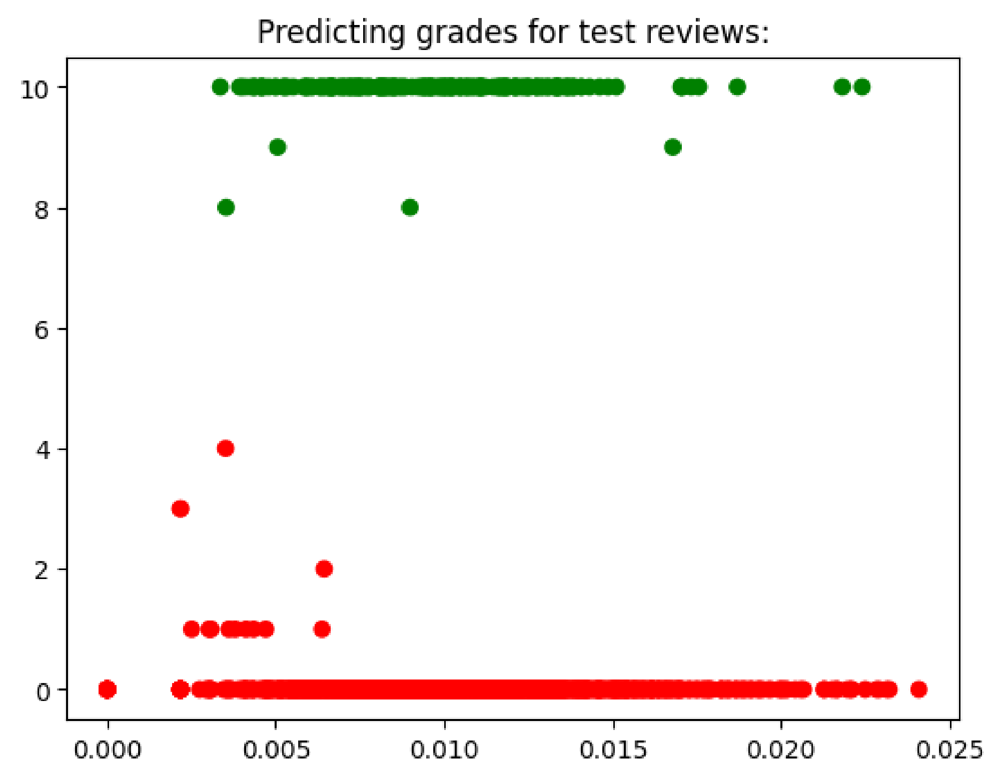
Random Forest - Grades

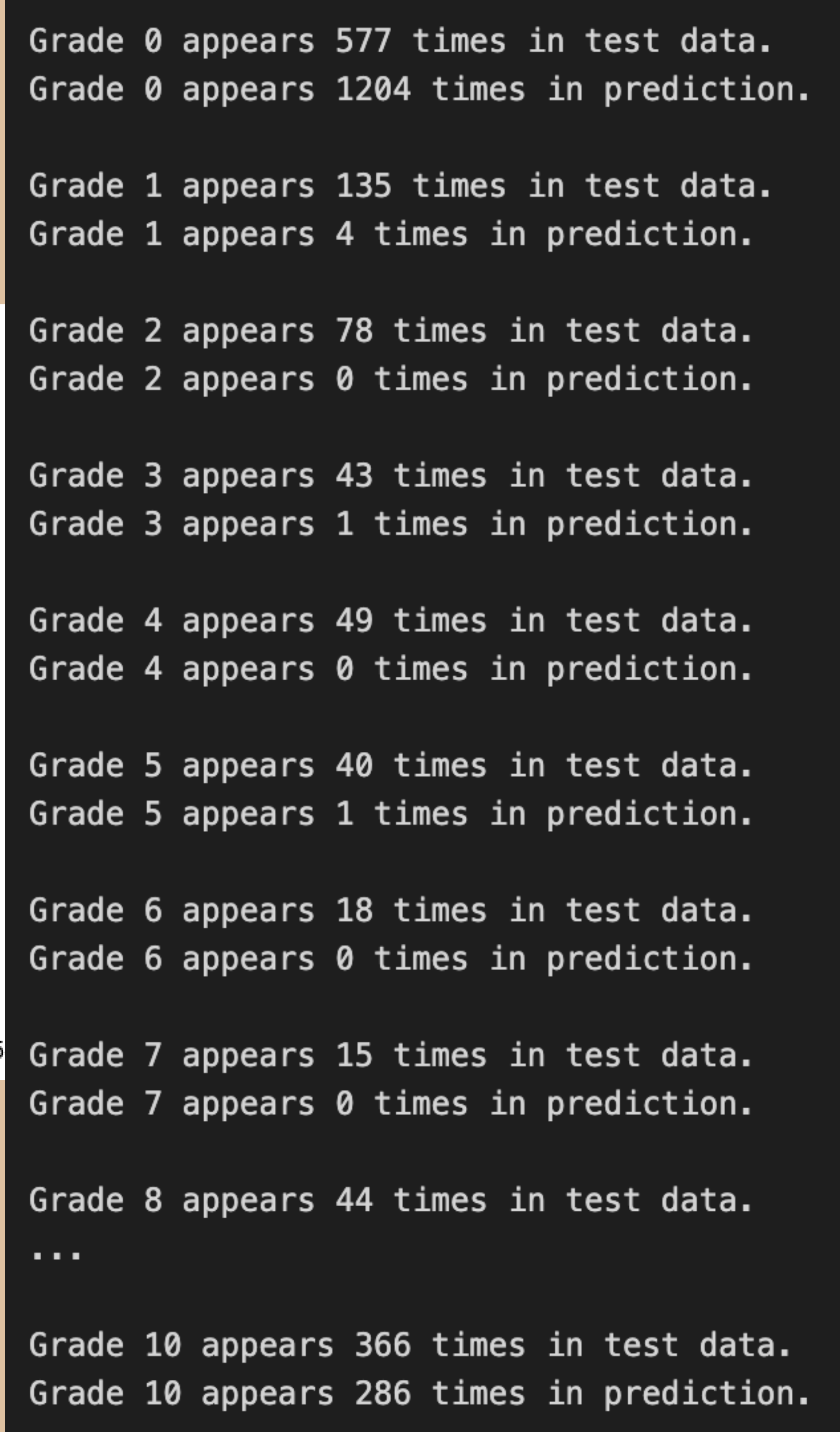




```
Grade 0 appears 581 times in training.
Grade 0 appears 590 times in prediction

Grade 1 appears 120 times in training.
Grade 1 appears 118 times in prediction

Grade 2 appears 53 times in training.
Grade 2 appears 53 times in prediction

Grade 3 appears 55 times in training.
Grade 3 appears 55 times in prediction

Grade 4 appears 56 times in training.
Grade 4 appears 56 times in prediction

Grade 5 appears 38 times in training.
Grade 5 appears 38 times in prediction

Grade 6 appears 26 times in training.
Grade 6 appears 25 times in prediction

Grade 7 appears 19 times in training.
Grade 7 appears 19 times in prediction

Grade 8 appears 47 times in training.
...

Grade 10 appears 386 times in training.
Grade 10 appears 382 times in prediction
```

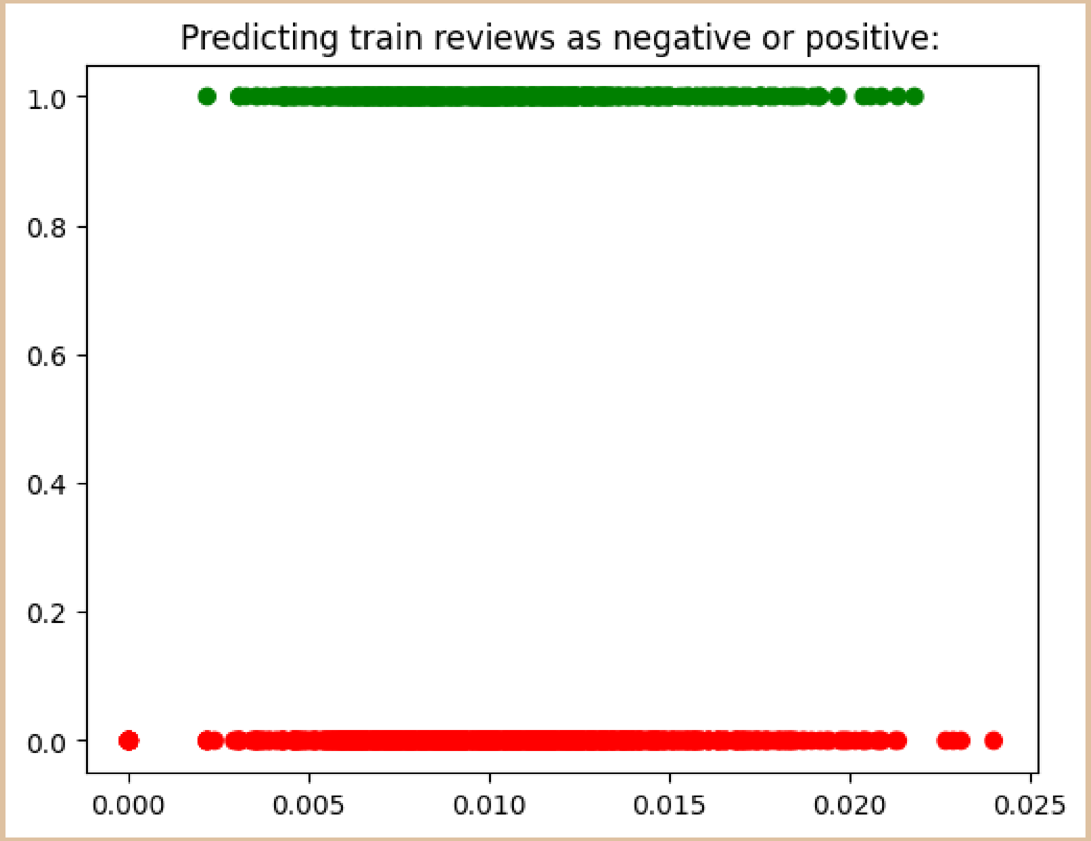
Random Forest - Grades

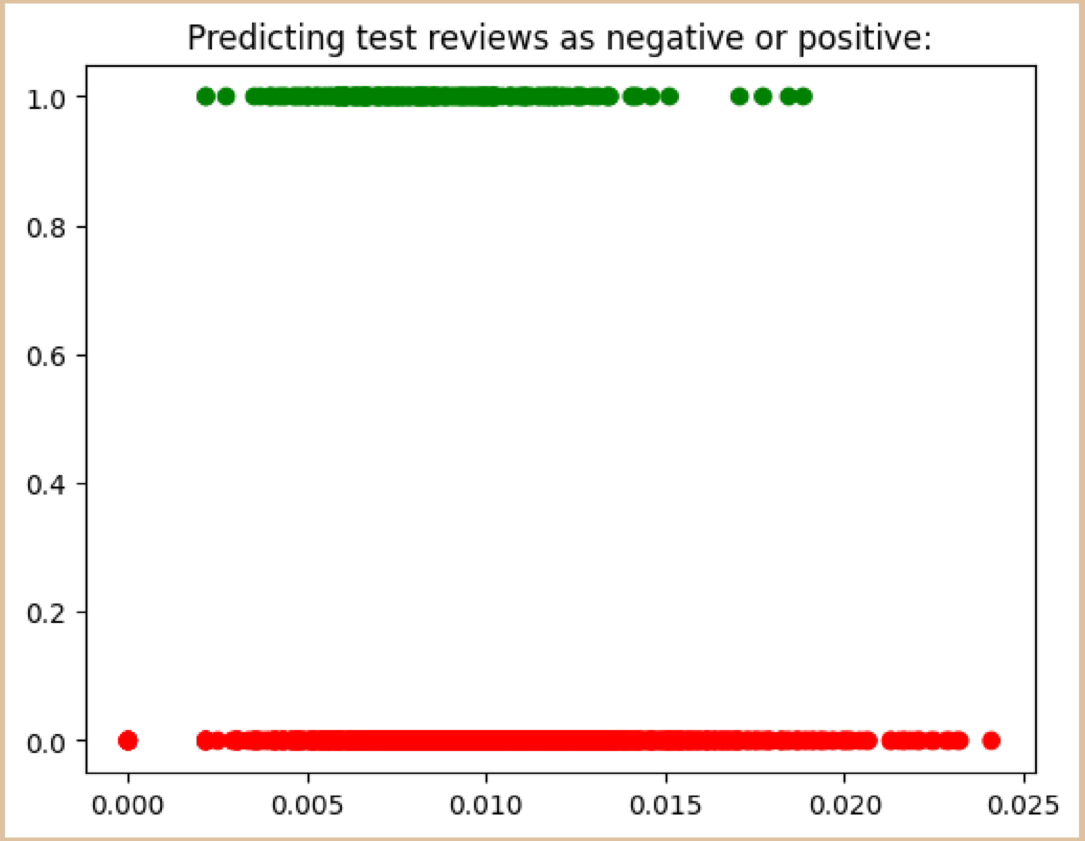




```
Grade 0 appears 577 times in test data.
Grade 0 appears 1204 times in prediction.

Grade 1 appears 135 times in test data.
Grade 1 appears 4 times in prediction.

Grade 2 appears 78 times in test data.
Grade 2 appears 0 times in prediction.

Grade 3 appears 43 times in test data.
Grade 3 appears 1 times in prediction.

Grade 4 appears 49 times in test data.
Grade 4 appears 0 times in prediction.

Grade 5 appears 40 times in test data.
Grade 5 appears 1 times in prediction.

Grade 6 appears 18 times in test data.
Grade 6 appears 0 times in prediction.

Grade 7 appears 15 times in test data.
Grade 7 appears 0 times in prediction.

Grade 8 appears 44 times in test data.
...

Grade 10 appears 366 times in test data.
Grade 10 appears 286 times in prediction.
```

Random Forest – 0 v 1

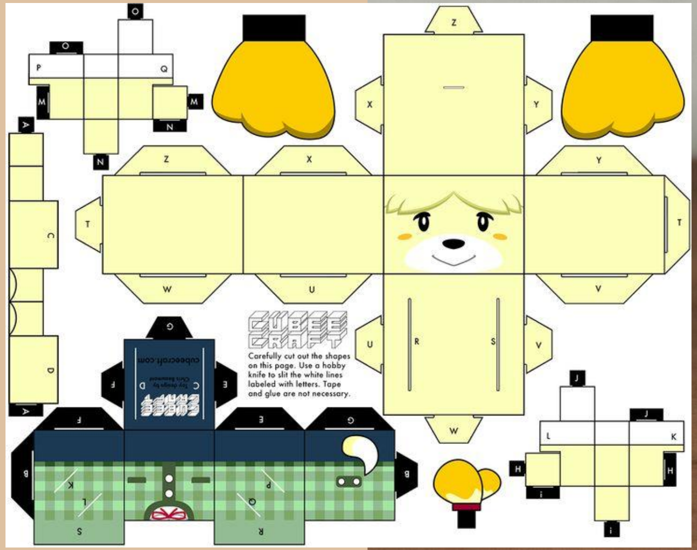


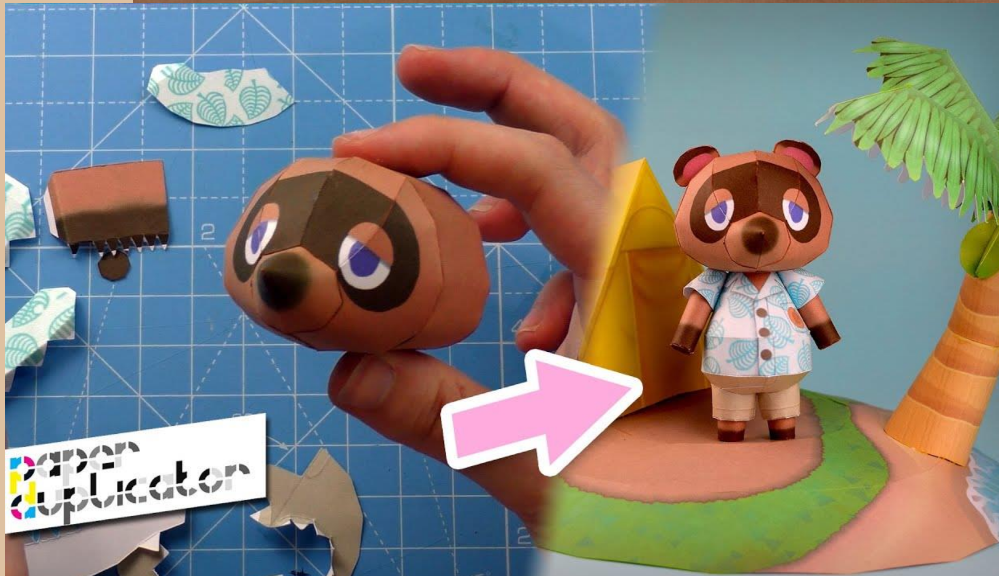


K-Fold: Fold 2 of 2

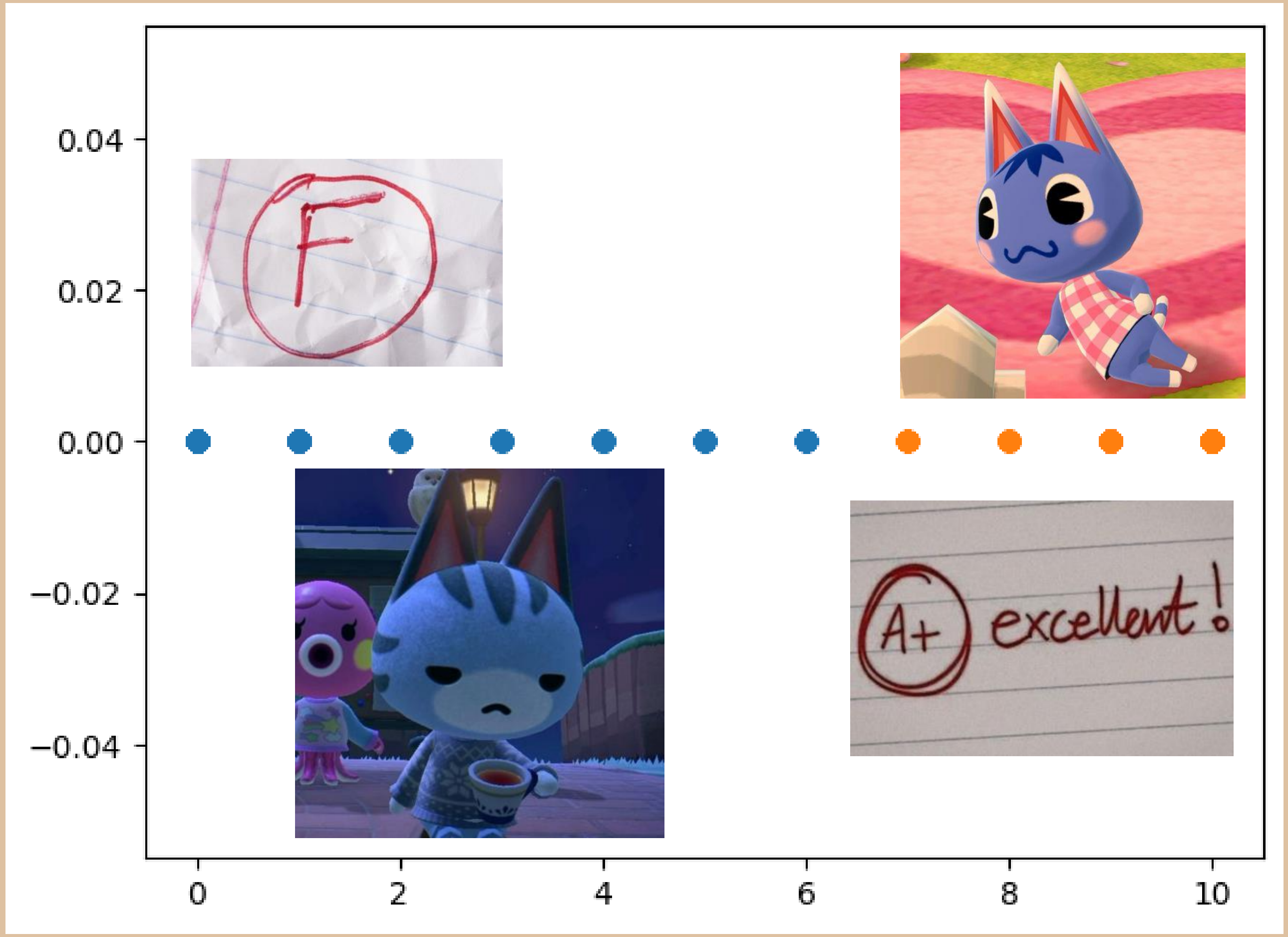
- K-Fold cross validation to split the data in half for two folds.

- Let’s see if the second fold of training and test data catches similar or different patterns.


CUBEE CRAFT
Carefully cut out the shapes on this page. Use a hobby knife to slit the white lines labeled with letters. Tape and glue are not necessary.
cubeecraft.com



Initial Clustering



Spectral clustering predicted based on grade information for each review.

Negative Reviews:

- **Grade 0-6**

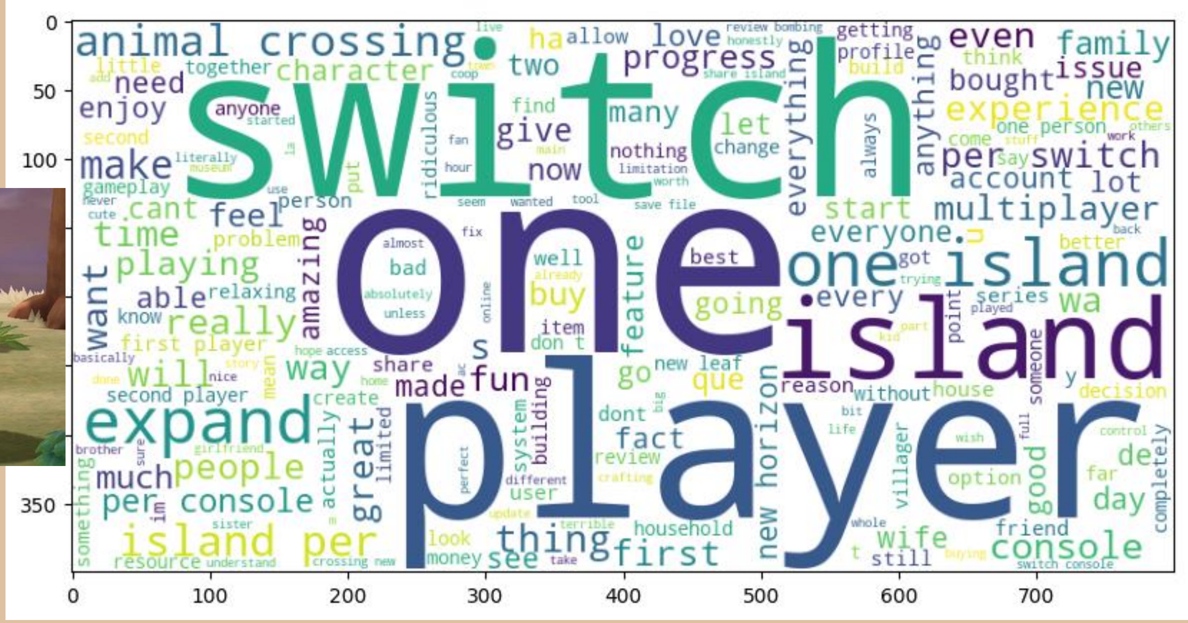
Positive Reviews:

- **Grade 7-10**

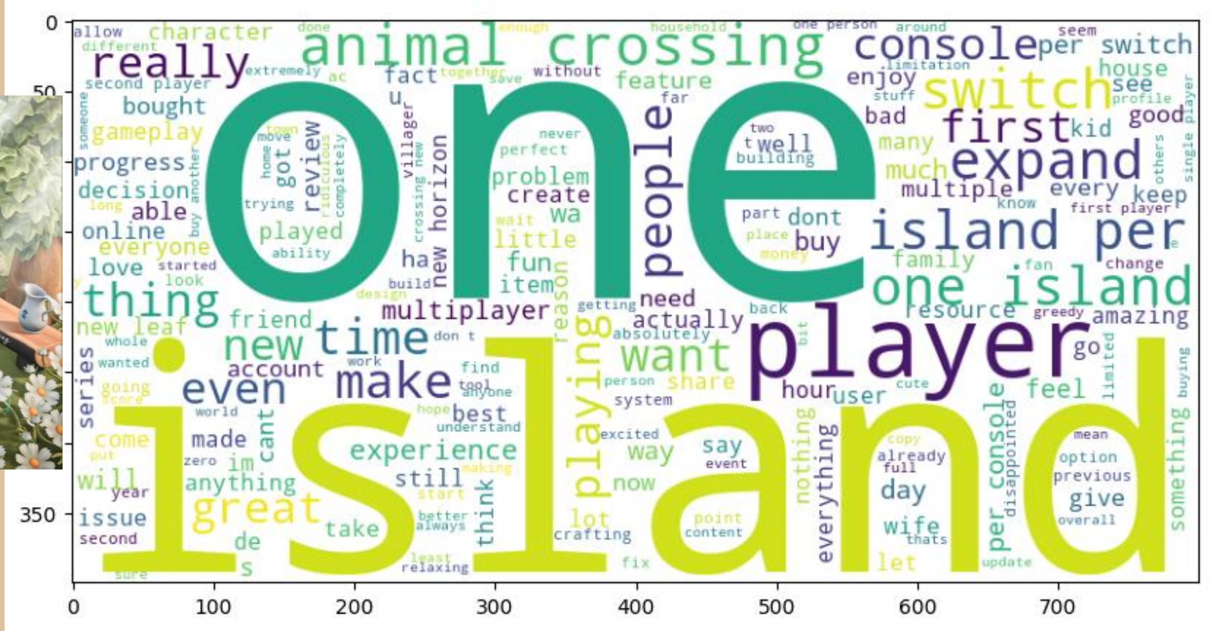
Initial Clustering

- For this second fold, both clusters feature the words "one", "island", and "player" as among their most common words.
- The negative cluster has more of an emphasis on "player" while the positive cluster has more emphasis on "island".

- This similarity contrasts with the clusters in the first fold, where the positive cluster contains a more balanced number of frequencies for each word.

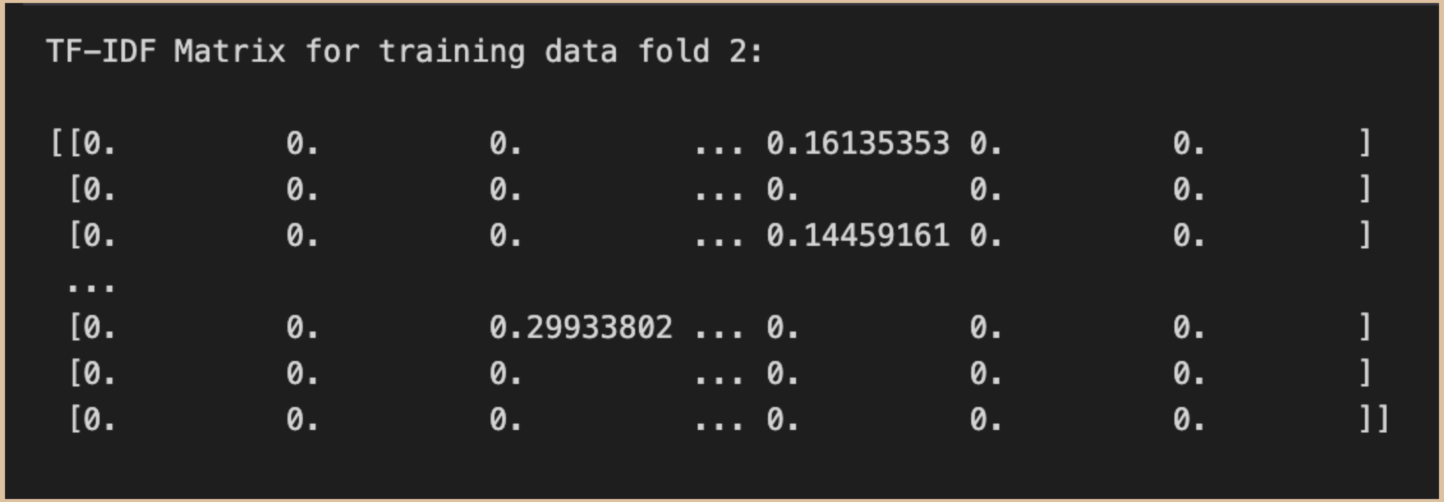


Negative reviews

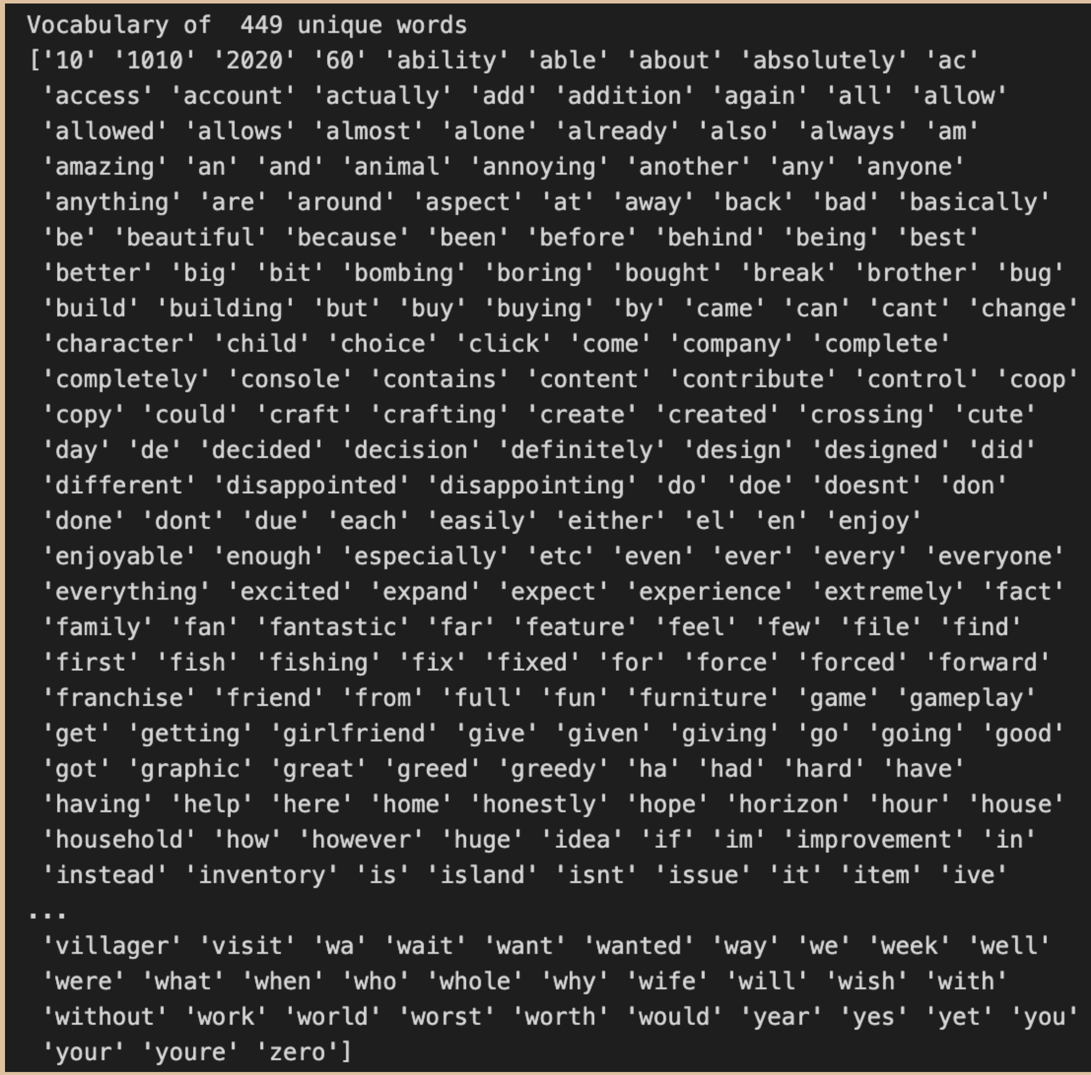


Positive reviews

Preprocessing 2: TF-IDF

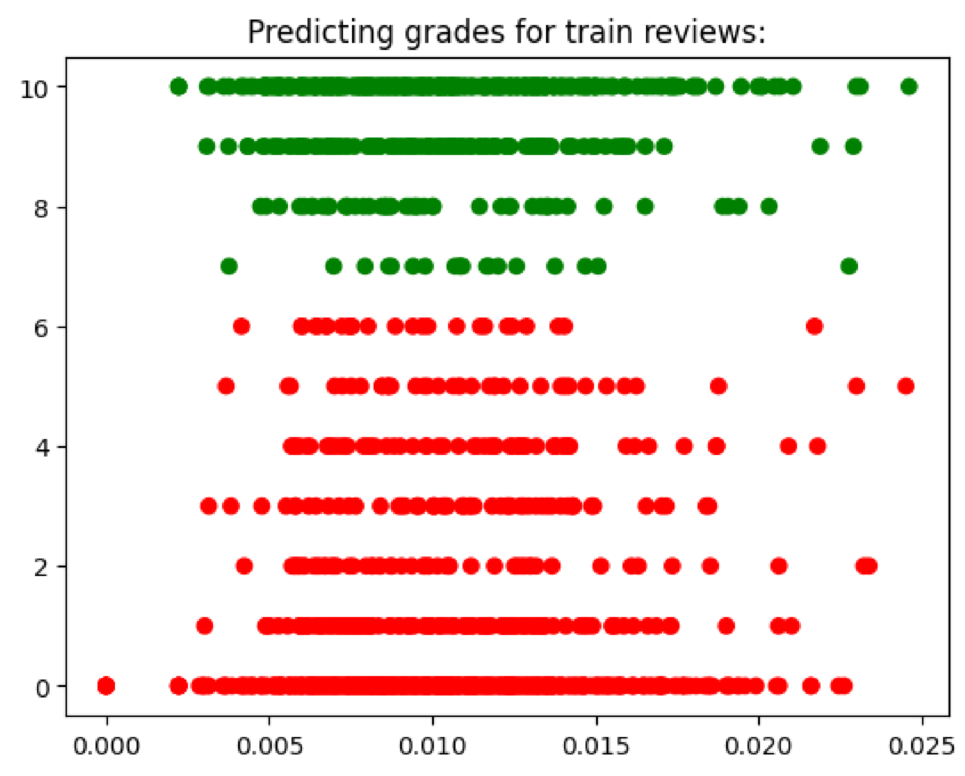


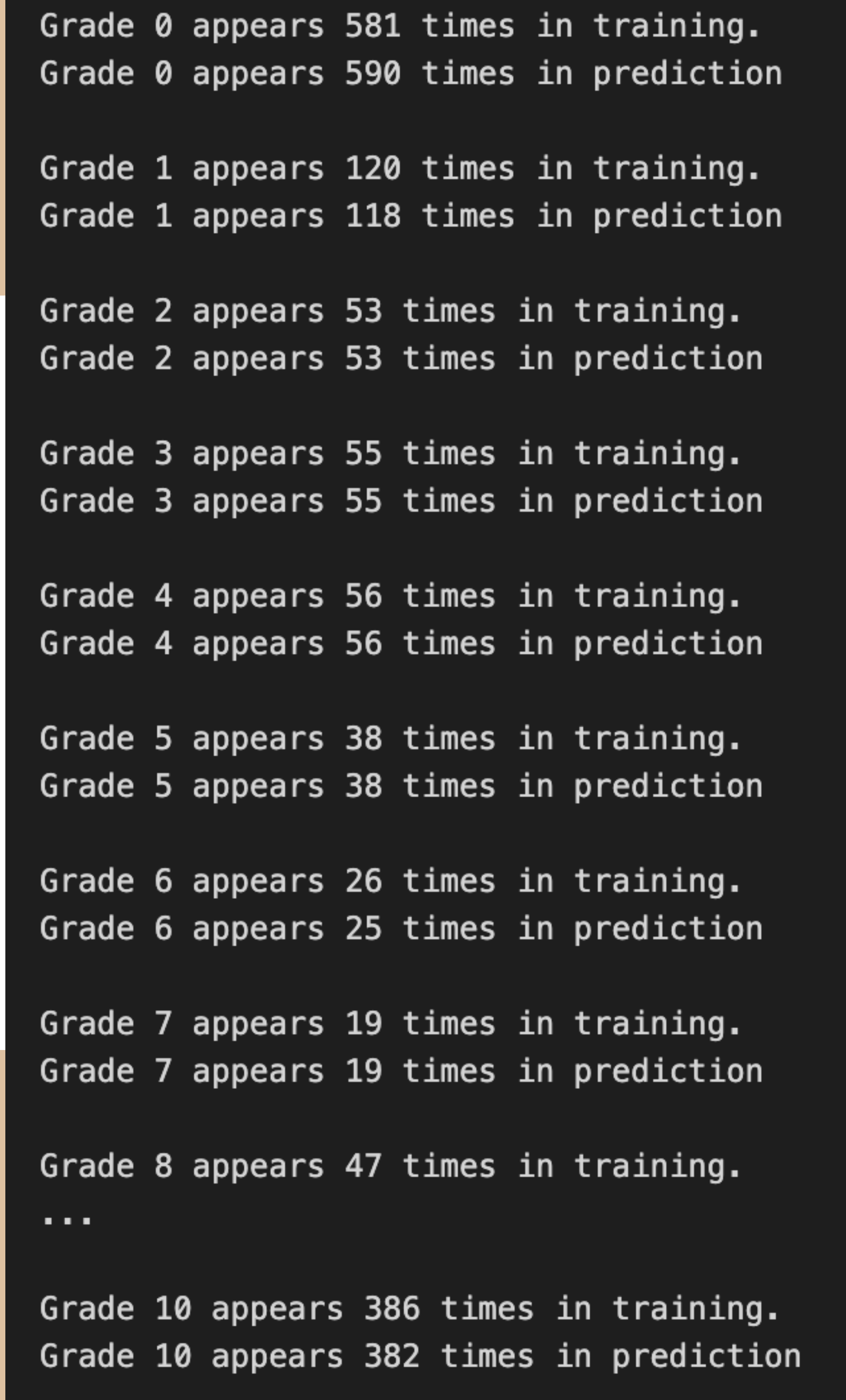
```
TF-IDF Matrix for training data fold 2:

[[0.         0.         0.         ... 0.16135353 0.         0.         ]
 [0.         0.         0.         ... 0.         0.         0.         ]
 [0.         0.         0.         ... 0.14459161 0.         0.         ]
 ...
 [0.         0.         0.29933802 ... 0.         0.         0.         ]
 [0.         0.         0.         ... 0.         0.         0.         ]
 [0.         0.         0.         ... 0.         0.         0.         ]]
```



```
Vocabulary of  449 unique words
['10' '1010' '2020' '60' 'ability' 'able' 'about' 'absolutely' 'ac'
 'access' 'account' 'actually' 'add' 'addition' 'again' 'all' 'allow'
 'allowed' 'allows' 'almost' 'alone' 'already' 'also' 'always' 'am'
 'amazing' 'an' 'and' 'animal' 'annoying' 'another' 'any' 'anyone'
 'anything' 'are' 'around' 'aspect' 'at' 'away' 'back' 'bad' 'basically'
 'be' 'beautiful' 'because' 'been' 'before' 'behind' 'being' 'best'
 'better' 'big' 'bit' 'bombing' 'boring' 'bought' 'break' 'brother' 'bug'
 'build' 'building' 'but' 'buy' 'buying' 'by' 'came' 'can' 'cant' 'change'
 'character' 'child' 'choice' 'click' 'come' 'company' 'complete'
 'completely' 'console' 'contains' 'content' 'contribute' 'control' 'coop'
 'copy' 'could' 'craft' 'crafting' 'create' 'created' 'crossing' 'cute'
 'day' 'de' 'decided' 'decision' 'definitely' 'design' 'designed' 'did'
 'different' 'disappointed' 'disappointing' 'do' 'doe' 'doesnt' 'don'
 'done' 'dont' 'due' 'each' 'easily' 'either' 'el' 'en' 'enjoy'
 'enjoyable' 'enough' 'especially' 'etc' 'even' 'ever' 'every' 'everyone'
 'everything' 'excited' 'expand' 'expect' 'experience' 'extremely' 'fact'
 'family' 'fan' 'fantastic' 'far' 'feature' 'feel' 'few' 'file' 'find'
 'first' 'fish' 'fishing' 'fix' 'fixed' 'for' 'force' 'forced' 'forward'
 'franchise' 'friend' 'from' 'full' 'fun' 'furniture' 'game' 'gameplay'
 'get' 'getting' 'girlfriend' 'give' 'given' 'giving' 'go' 'going' 'good'
 'got' 'graphic' 'great' 'greed' 'greedy' 'ha' 'had' 'hard' 'have'
 'having' 'help' 'here' 'home' 'honestly' 'hope' 'horizon' 'hour' 'house'
 'household' 'how' 'however' 'huge' 'idea' 'if' 'im' 'improvement' 'in'
 'instead' 'inventory' 'is' 'island' 'isnt' 'issue' 'it' 'item' 'ive'
...
 'villager' 'visit' 'wa' 'wait' 'want' 'wanted' 'way' 'we' 'week' 'well'
 'were' 'what' 'when' 'who' 'whole' 'why' 'wife' 'will' 'wish' 'with'
 'without' 'work' 'world' 'worst' 'worth' 'would' 'year' 'yes' 'yet' 'you'
 'your' 'youre' 'zero']
```

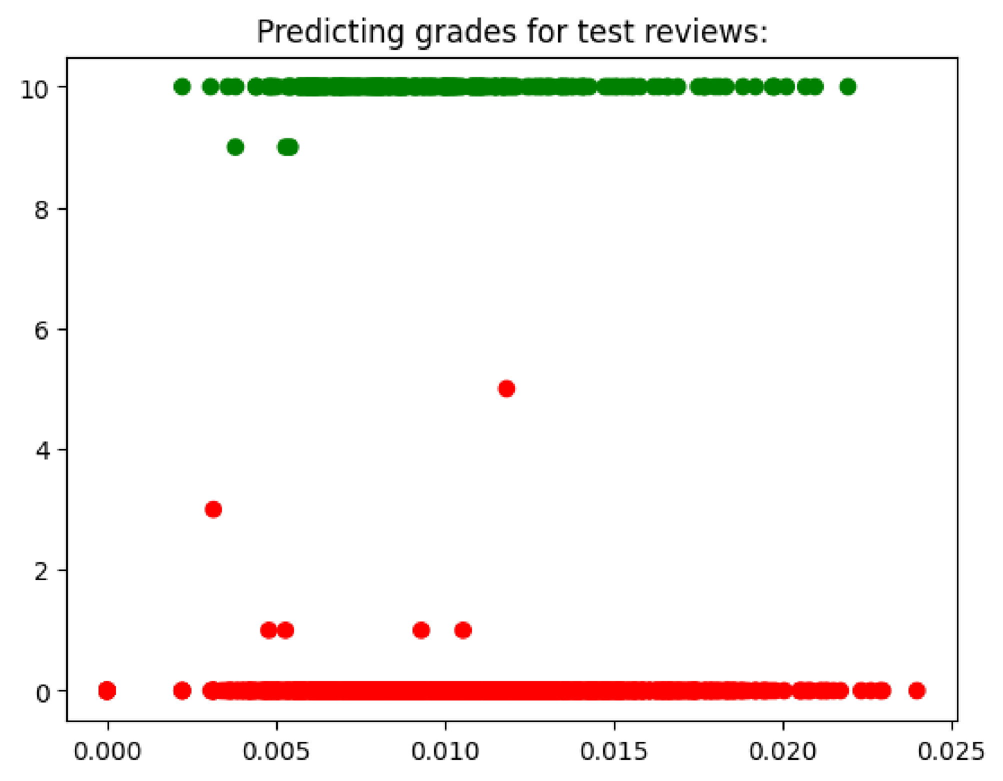
Random Forest - Grades

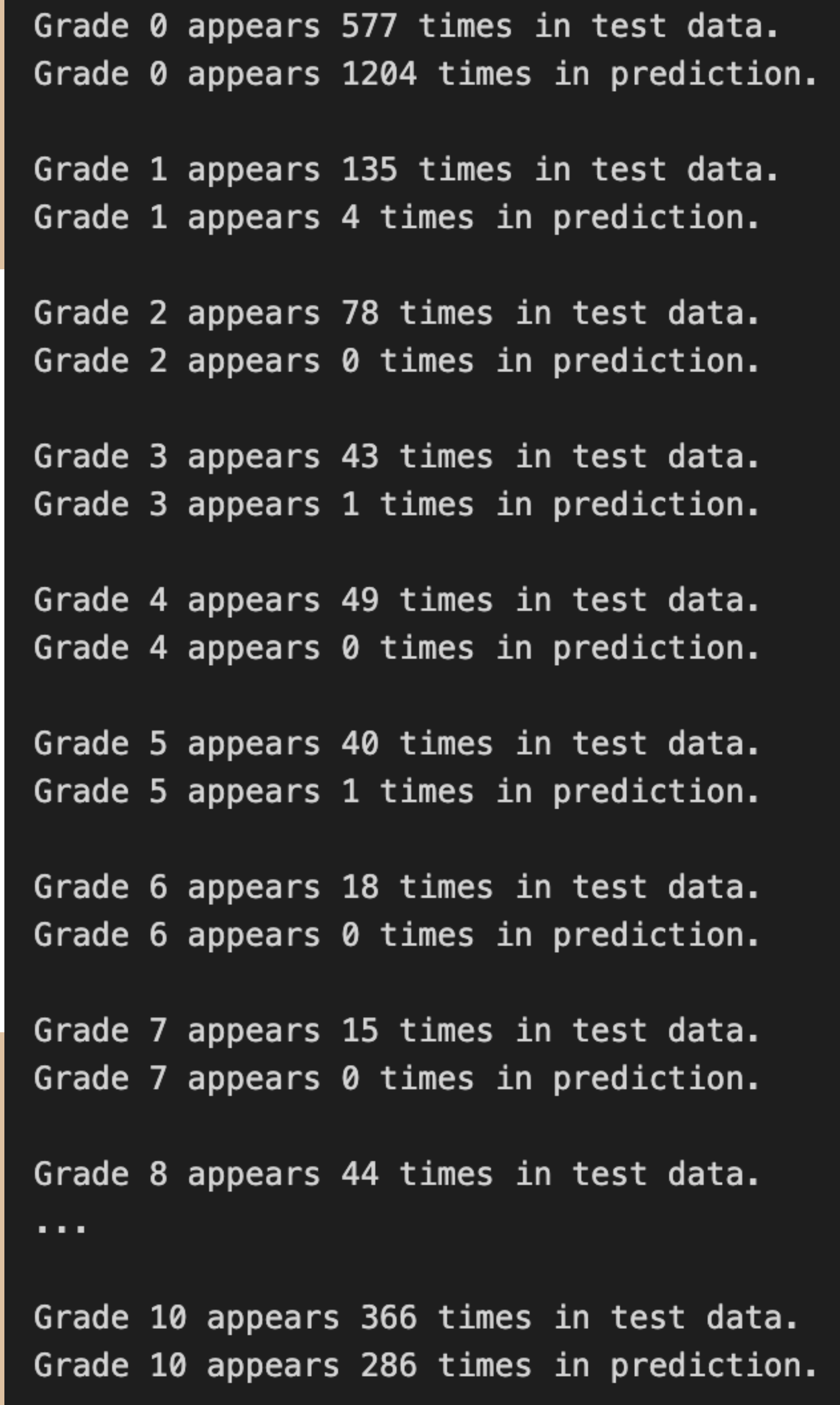




```
Grade 0 appears 581 times in training.
Grade 0 appears 590 times in prediction

Grade 1 appears 120 times in training.
Grade 1 appears 118 times in prediction

Grade 2 appears 53 times in training.
Grade 2 appears 53 times in prediction

Grade 3 appears 55 times in training.
Grade 3 appears 55 times in prediction

Grade 4 appears 56 times in training.
Grade 4 appears 56 times in prediction

Grade 5 appears 38 times in training.
Grade 5 appears 38 times in prediction

Grade 6 appears 26 times in training.
Grade 6 appears 25 times in prediction

Grade 7 appears 19 times in training.
Grade 7 appears 19 times in prediction

Grade 8 appears 47 times in training.
...

Grade 10 appears 386 times in training.
Grade 10 appears 382 times in prediction
```

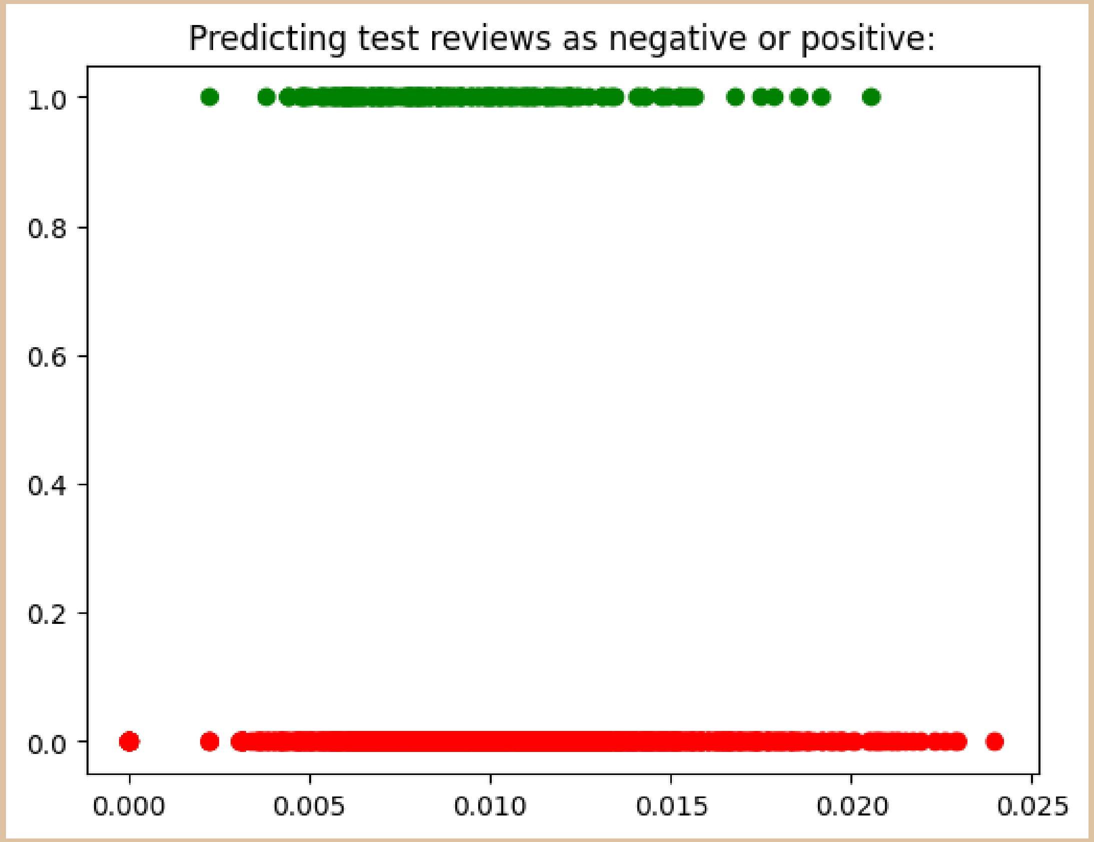
Random Forest - Grades

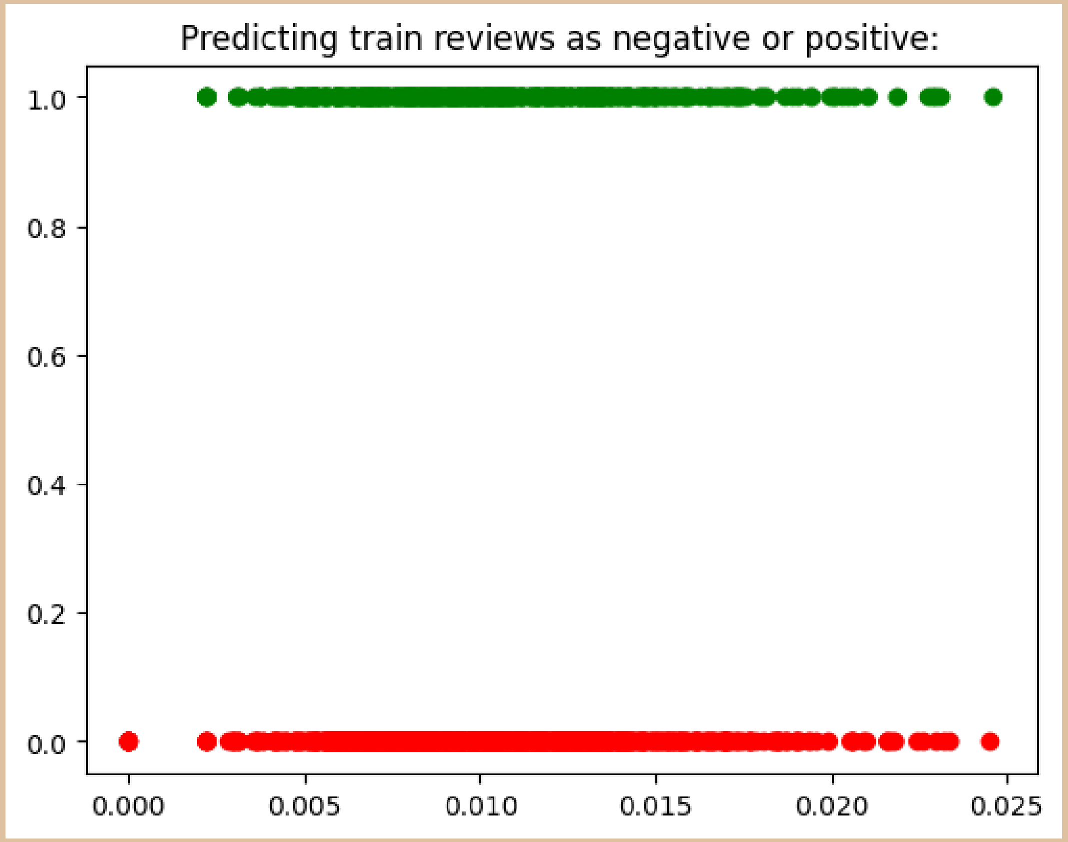




```
Grade 0 appears 577 times in test data.
Grade 0 appears 1204 times in prediction.

Grade 1 appears 135 times in test data.
Grade 1 appears 4 times in prediction.

Grade 2 appears 78 times in test data.
Grade 2 appears 0 times in prediction.

Grade 3 appears 43 times in test data.
Grade 3 appears 1 times in prediction.

Grade 4 appears 49 times in test data.
Grade 4 appears 0 times in prediction.

Grade 5 appears 40 times in test data.
Grade 5 appears 1 times in prediction.

Grade 6 appears 18 times in test data.
Grade 6 appears 0 times in prediction.

Grade 7 appears 15 times in test data.
Grade 7 appears 0 times in prediction.

Grade 8 appears 44 times in test data.
...

Grade 10 appears 366 times in test data.
Grade 10 appears 286 times in prediction.
```

Random Forest – 0 v 1





The Results:

- In both folds, the random forest classifier algorithms produce very similar results, showing patterns are found consistently across the entire dataset of reviews for Animal Crossing New Horizons.
- The thing that was most different were the clusters for positive reviews in each fold. Finding which words define a positive vs. negative cluster may prove to be difficult for the second fold than the first, as the two clusters were more similar in the second fold.

The Results:

- The next step for both folds was to confirm if positive or negative reviews could be defined by the frequency of certain words. I ran of time to implement this.
- When reading reviews on my own, it does seem that many users complain about the fact that players can only have one island for their characters per Nintendo switch console.
- The word clouds seem to confirm that negative reviews feature words referring to this complaint.
- However, even many of the positive reviews, including those that gave the game full marks, also address this issue as something they did not like.

Thank you for your time!