Assignment: Final Report
Topic: Movie Review Sentiment Analysis on genres and review ratings
Date: Dec 13, 2022
Name: Vita Huang

Introduction:

The movie review is one of the referenced resources for movies viewers. Many websites such as IMDB or Rotten Tomatoes are encouraging viewers to post their opinion after watching the movie. This information is useful for both readers and movie companies. For example, manufacturers can collect feedback from their customers to improve their products through movie reviews. On the other hand, people could objectively evaluate a product by viewing other people's opinions, which will possibly influence their decisions on whether to watch the movie.

Data Preprocessing:

1. I import two datasets; one is a movie and another one is critics.
2. Drop nan rows
3. Build two binary columns which are "tomato_binary_rating" and "audience_binary_rating". If the rating is higher than 60, the column value would be 1.
4. Create two movie genres. The first genre is "Horror & Suspense" from "Horror" and "Mystery & Suspense" genres, and the second one is "Family movie" from "Romance", "Kids & Family", and "Comedy".
5. 2425 movies are "Horror & Suspense". 3877 movies are "Family movie"
6. For the second dataset, I remove the stop words, digits, and duplicate strings. I found lots of words which should have 'e' at the end but without "e" after I use the PorterStemmer and WordNetLemmatizer functions which cause those words becoming not meaningful. Therefore, I decided not to lemmatize words.
7. Use information gain to calculate word entropy for the Horror & Suspense genre. Then rank words through their entropy
8. Finally, I create 200 words dataset and a 400 words dataset for modeling
9. Top ten information gain words.

```
1812790     thriller
3065515     suspense
311582        horror
1863417       comedy
2731921     charming
3398158        sweet
2451801        funny
3453257        tense
99572      thrillers
3805304      thrills
Name: remove_duplicate_words, dtype: object
```

Related Works:

1. Mukta et al. (2017) extract psychological attributes: personality and value scores from tweets to predict users' movie preferences. Firstly, they find the users who have left the movie review on IMDB and have Twitter accounts. Then, building classification models that take tweets as input, and give movie genre preference as output. After that, training a model by exploiting psychological concepts, which may influence people's performance. My datasets include movie overviews, critics' movie ratings, audience movie ratings, and critics' reviews. Following the paper, I think I could use critics' ratings and audience ratings to predict audience genre preference. Furthermore, the paper indicates that movie ratings lower than seven should not be considered valuable data since they are identifying users' movie preferences. However, I did not use this rating boundary to filter the data because my rating scale is 1 to 100 which is different than the paper's 1 to 10. I should include this setting in my future work. In the classification session, the paper falls into the intersection of a movie having more than one movie genre and I have the same problem. They distribute the genre names into multiple rows, but I choose the genre which has lower instances to be a class label. This is my second future work to think about how to deal with movies that have multiple genres.
   Mukta et al. (2017) apply Naive Bayes, SVM, Random Forest, Random Tree, and RepTree classifiers in the dataset to predict the different genres of movies from users' value dimensions. They conduct performance evaluation through AUC values under the 10-fold cross-validation.

Table 1: Personality based classification of movie genres

Genres	Best classifier	AUC	TPR	TNR
Drama	RepTree	0.62	0.94	0.85
Thriller	RepTree	0.55	0.09	0.03
Comedy	RandomTree	0.59	0.06	0.03
Action	RandomTree	0.61	0.28	0.25
Adventure	RepTree	0.60	0.07	0.03

Table 2: Value based classification of movie genres

Genres	Best classifier	AUC	TPR	TNR
Drama	Random Tree	0.63	0.91	0.81
Thriller	Random Forest	0.59	0.17	0.13
Comedy	RepTree	0.59	0.10	0.02
Action	RandomTree	0.54	0.12	0.07
Adventure	Random Forest	0.59	0.07	0.04

Table 4: Classification result to identify genre by using both personality and value scores

Genres	Best classifier	AUC	TPR	TNR
Drama	RepTree	0.65	0.94	0.85
Thriller	RandomForest	0.59	0.22	0.18
Comedy	RandomTree	0.62	0.27	0.21
Action	RandomTree	0.65	0.52	0.37
Adventure	RepTree	0.64	0.37	0.12

Table 4 is combined the personality and values-based models and shows better AUC results than independent models. The higher AUC can correctly distinguish between classes. Also, these classifiers show substantial improvement in TPR and TNR scores for movie genre classifiers.
I did not include the precision, recall, F1, and accuracy because there is no predicted positive and negative information in the paper, so I include the tables with AUC, TPR, and TNR as the model evaluations.

2. Zhuang et al. (2006) focus on extracting features from movie reviews, summarization, and observing whether the text reviews are positive or negative. Movie reviews are different from product reviews since the audience comments not only on movie elements, such as music, and special visual effects but also related people, for example, actors and directors. Therefore, there are many options for feature selection. In the paper, they process the reviews in a few steps, identifying the relationship between feature words and subjective words, then checking the polarity of opinions. First, the subjective classification distinguishes subjective opinions and evaluations that present objective information, and according to the feature words, they select the relevant subjective words to produce the summary. This classification I think I need to include this in my future work. If I could identify the subjective and objective reviews, the models' predictions are more trustworthy. In the task of sentiment classification, they propose several machine learning approaches to find the semantic distance from "good" words to "bad" as the criterion. The logic of this method is reasonable and I may do it in future work. Identifying the words distances for "Horror & Suspense" and "Family Movie".

Table 4: Results of feature-opinion pair mining

Movie	Hu and Liu's approach			The proposed approach		
	Precision	Recall	F-score	Precision	Recall	F-score
Gone with the Wind	0.462	0.651	0.551	0.556	0.564	0.560
The Wizard of OZ	0.475	0.705	0.568	0.589	0.648	0.618
Casablanca	0.431	0.661	0.522	0.452	0.521	0.484
The Godfather	0.400	0.654	0.496	0.476	0.619	0.538
The Shawshank Redemption	0.443	0.620	0.517	0.514	0.644	0.571
The Matrix	0.353	0.565	0.434	0.468	0.593	0.523
The Two Towers	0.338	0.583	0.428	0.404	0.577	0.476
American Beauty	0.375	0.576	0.454	0.393	0.527	0.450
Gladiator	0.405	0.619	0.489	0.505	0.632	0.562
Wo hu cang long	0.368	0.567	0.447	0.465	0.537	0.498
Spirited Away	0.388	0.583	0.466	0.493	0.567	0.527
Average	0.403	0.617	0.488	0.483	0.585	0.529

Experiment 1 Modeling (Random Forest and Decision Tree):

In the experiment 1, I use the Random Forest and Decision Tree predicting the movie genre based on the word entropy, and I apply the same parameters for two hundred and four hundred information gain datasets. The test size is 0.2:

```
RF = RandomForestClassifier(random_state = 1, max_depth = 46,
                            max_features=70, min_samples_leaf = 20,
                           n_estimators=50)
RF.fit(X_train, y_train)
```

```
RandomForestClassifier(max_depth=46, max_features=70, min_samples_leaf=20,
                       n_estimators=50, random_state=1)
```

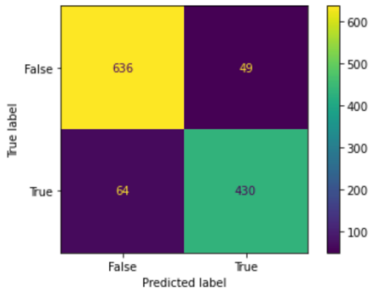
```
DT = DecisionTreeClassifier(random_state = 1, max_depth = 32,
                            max_features=40, min_samples_leaf = 5)
DT.fit(X_train, y_train)
```

```
DecisionTreeClassifier(max_depth=32, max_features=40, min_samples_leaf=5,
                       random_state=1)
```

1. For the two hundred information gain dataset

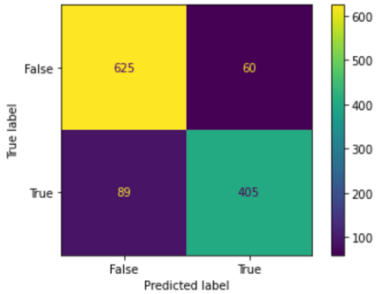
Random Forest:

```
For the two hundred information gain dataset
RF training_set: accuracy:0.924, precision:0.92, recall:0.891, f1:0.905
RF test_set: accuracy:0.904, precision:0.898, recall:0.87, f1:0.884
                    precision    recall  f1-score   support

Horror & Suspense       0.91      0.93      0.92       685
     Family Movie       0.90      0.87      0.88       494

          accuracy                           0.90      1179
         macro avg       0.90      0.90      0.90      1179
      weighted avg       0.90      0.90      0.90      1179
```



Decision Tree:

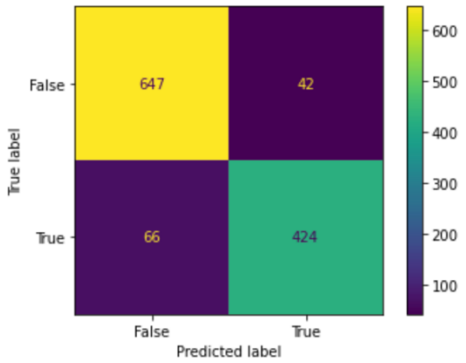
```
DT training_set: accuracy:0.93, precision:0.938, recall:0.887, f1:0.912
DT test_set: accuracy:0.874, precision:0.871, recall:0.82, f1:0.845
                    precision    recall  f1-score   support

Horror & Suspense       0.88      0.91      0.89       685
     Family Movie       0.87      0.82      0.84       494

          accuracy                           0.87      1179
         macro avg       0.87      0.87      0.87      1179
      weighted avg       0.87      0.87      0.87      1179
```



2. For the four hundred information gain dataset

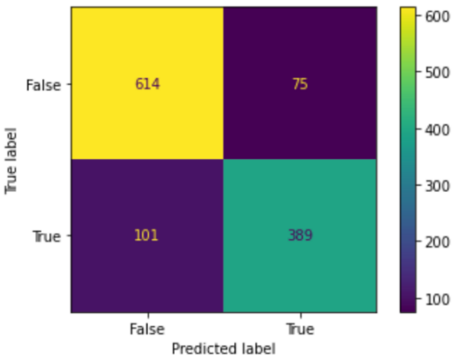
Random Forest:

```
For the four hundred information gain dataset
RF training_set: accuracy:0.92, precision:0.919, recall:0.882, f1:0.9
RF test_set: accuracy:0.908, precision:0.91, recall:0.865, f1:0.887
                   precision    recall  f1-score   support

Horror & Suspense       0.91      0.94      0.92       689
     Family Movie       0.91      0.87      0.89       490

         accuracy                           0.91      1179
        macro avg       0.91      0.90      0.90      1179
     weighted avg       0.91      0.91      0.91      1179
```



Decision Tree:

```
DT training_set: accuracy:0.919, precision:0.924, recall:0.874, f1:0.898
DT test_set: accuracy:0.851, precision:0.838, recall:0.794, f1:0.816
                   precision    recall  f1-score   support

Horror & Suspense       0.86      0.89      0.87       689
     Family Movie       0.84      0.79      0.82       490

         accuracy                           0.85      1179
        macro avg       0.85      0.84      0.85      1179
     weighted avg       0.85      0.85      0.85      1179
```



Thought:

Overall, the Random Forest has better performance than the Decision Tree because of the higher F1 score. First, I tuned the max_features in the Random Forest which significantly increases the F1. The max_features in the Random Forest model means a maximum number of features in one tree. Increasing max_features takes more time to produce outputs, but improves overall performance. Therefore, finding optimal max_features is important to my model's performance. For the Decision Tree, I increased the max_depth and lowered the

min_samples_leaf number. The max_depth indicates how deeper the tree and it captures more information. The min_samples_leaf describes the minimum number of samples at the leaves.

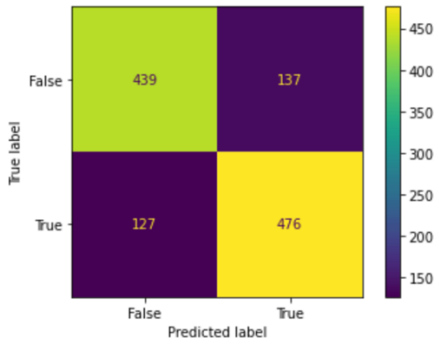
Experiment 2 Modeling (Random Forest, Decision Tree, XGBoost, and Logistic Regression):

In the experiment 2, I use Random Forest, Decision Tree, XGBoost, and Logistic Regression for predicting audience ratings. I apply the same parameters in Random Forest and Decision Tree as I did in experiment 1.

3. For the two hundred datasets for predicting audience ratings

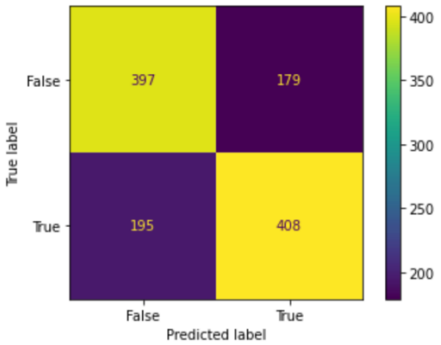
Random Forest:

```
RF training_set: accuracy:0.787, precision:0.79, recall:0.793, f1:0.791
RF test_set: accuracy:0.776, precision:0.777, recall:0.789, f1:0.783
                   precision    recall  f1-score   support

Horror & Suspense       0.78      0.76      0.77       576
     Family Movie       0.78      0.79      0.78       603

         accuracy                           0.78      1179
        macro avg       0.78      0.78      0.78      1179
     weighted avg       0.78      0.78      0.78      1179
```



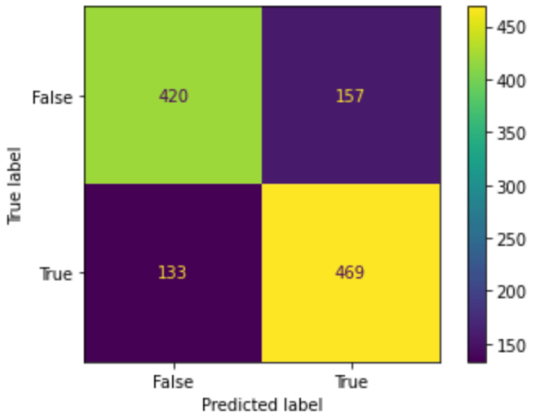
Decision Tree:

```
The two hundred information gain dataset predicts the audience ratings
DT training_set: accuracy:0.845, precision:0.859, recall:0.833, f1:0.846
DT test_set: accuracy:0.683, precision:0.695, recall:0.677, f1:0.686
                   precision    recall  f1-score   support

Horror & Suspense       0.67      0.69      0.68       576
     Family Movie       0.70      0.68      0.69       603

         accuracy                           0.68      1179
        macro avg       0.68      0.68      0.68      1179
     weighted avg       0.68      0.68      0.68      1179
```



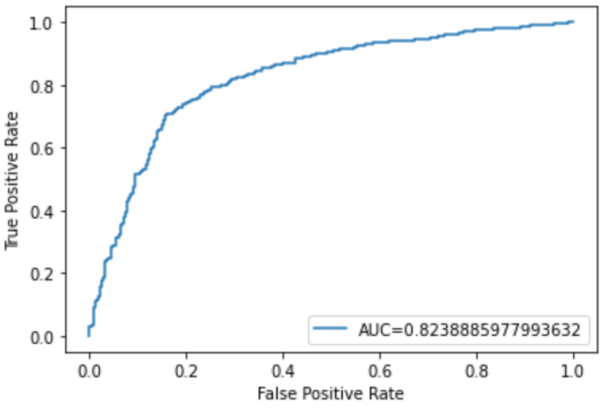
XGBoost:

```
The two hundred information gain dataset predicts the audience ratings
xgbc training_set: accuracy:0.979, precision:0.98, recall:0.98, f1:0.98
xgbc test_set: accuracy:0.754, precision:0.749, recall:0.779, f1:0.764
                 precision    recall  f1-score   support

Audience_rating       0.76      0.73      0.74       577
  Tomato_rating       0.75      0.78      0.76       602

       accuracy                           0.75      1179
      macro avg       0.75      0.75      0.75      1179
   weighted avg       0.75      0.75      0.75      1179
```



Logistic Regression:

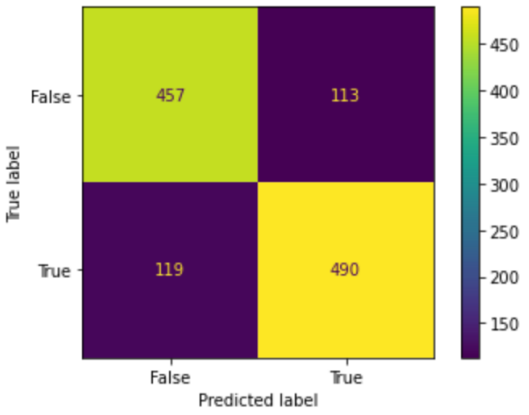
```
#define metrics
y_pred_proba = log_regression.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)

#create ROC curve
plt.plot(fpr,tpr,label="AUC="+str(auc))
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend(loc=4)
plt.show()
```



4. For the four hundred datasets for predicting audience ratings

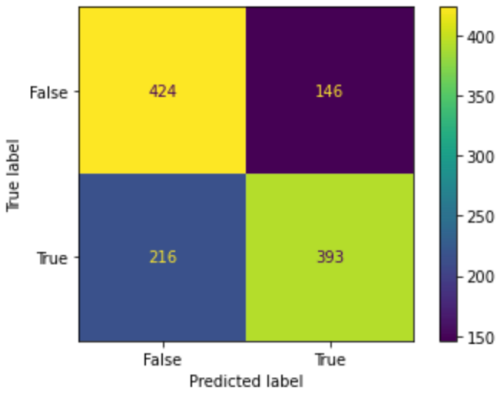
Random Forest:

```
RF training_set: accuracy:0.787, precision:0.787, recall:0.797, f1:0.792
RF test_set: accuracy:0.803, precision:0.813, recall:0.805, f1:0.809
                    precision    recall  f1-score   support

Horror & Suspense       0.79      0.80      0.80       570
     Family Movie       0.81      0.80      0.81       609

          accuracy                           0.80      1179
         macro avg       0.80      0.80      0.80      1179
      weighted avg       0.80      0.80      0.80      1179
```



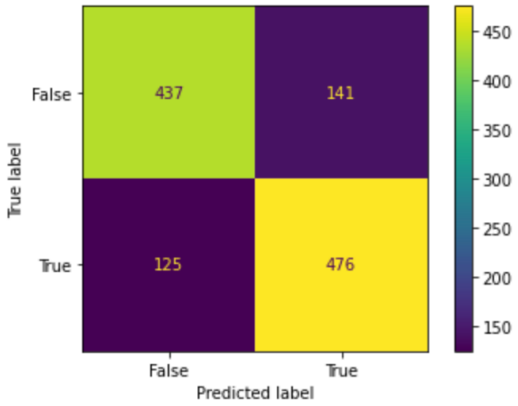
Decision Tree:

```
The four hundred information gain dataset predicts the audience ratings
DT training_set: accuracy:0.834, precision:0.855, recall:0.811, f1:0.832
DT test_set: accuracy:0.693, precision:0.729, recall:0.645, f1:0.685
                    precision    recall  f1-score   support

Horror & Suspense       0.66      0.74      0.70       570
     Family Movie       0.73      0.65      0.68       609

          accuracy                           0.69      1179
         macro avg       0.70      0.69      0.69      1179
      weighted avg       0.70      0.69      0.69      1179
```



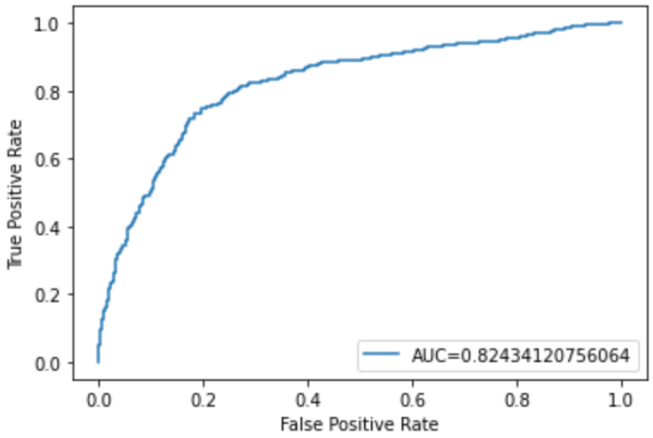
XGBoost:

```
The four hundred information gain dataset predicts the audience ratings
xgbc training_set: accuracy:0.982, precision:0.984, recall:0.981, f1:0.982
xgbc test_set: accuracy:0.774, precision:0.771, recall:0.792, f1:0.782
                  precision    recall  f1-score   support

Audience_rating       0.78      0.76      0.77       578
  Tomato_rating       0.77      0.79      0.78       601

       accuracy                           0.77      1179
      macro avg       0.77      0.77      0.77      1179
   weighted avg       0.77      0.77      0.77      1179
```



Logistic Regression:

```
#define metrics
y_pred_proba = log_regression.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)

#create ROC curve
plt.plot(fpr,tpr,label="AUC="+str(auc))
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend(loc=4)
plt.show()
```



Thought: I chose the XGBoost as my third modeling because it is a popular technique in machine learning, based on Gradient Boosting, and often compared with the Random Forest. The decision trees in Random Forest are built independently at a time with different features. Random Forest considers the average of the output so that if decision trees are more, the accuracy will be higher. However, XGBoost only builds one decision tree at a time and the next tree will be built on what the algorithm learned from the last tree. XGBoost helps developers to work with the decision tree algorithm and gradient algorithm.
I did not set the parameters in the XGBoost because the f1 is not higher than unadjusted. when I adjusted the scale_pos_weight, the f1 will become lower. The scale_pos_weight parameter balances the positive and negative weights and imposes greater penalties on the minor class if there is an error. Furthermore, I adjusted the max_delta_step because I care about right predicting probability; however, the f1 from the setting became lower or did not change at all. Therefore, I decide not to adjust the parameters.
I also did not set the parameters in the Logistic Regression, but the AUC is over 0.8. The higher the AUC, the better the model is at predicting classes. By analogy, the Higher the AUC, the better the model is at distinguishing between review ratings that are under or over 60.
Overall, Random Forest has the best performance among the three models, the second is XGBoost, and the final one is Decision Tree.

Error Analysis:

False Negative cases (Horror & Suspense → Family Movie)
The terms are found in the movie, The Hitcher. The movie genre should be Horror & Suspense, but is predicted as Family Movie:
Abandons, action, atmosphere, blandly, blood, comfortably, creepy, depressingly
Emotional, favorite, fear, frightening, intellectually, teeny, delicious
Fascinating, civil, society, teenage

1. Bigger, louder and far bloodier than the original, but devoid of all the delicious ambiguity that made it so fascinating.
2. Their highest ambition is making teenage girls jump out of their chairs.
3. The original film was nicely atmospheric and gritty, whereas this version is aimed at the t een audience and lacks the tension, relying on car crashes rather than chills.
4. About all the movie is effective at is insulting the audience.

Thought:
When I looked at the reviews, some of them mentioned the movie name and some verbs that help model interpreted movie genre as Horror & Suspense. For example, thriller, killing, bloody, or terrifying. However, some reviews I selected are difficult to interpret as the horror genre, such as teeny, teenage, and mediocre. The movie reviews are about reviewer sentiment, movie actors, actors' actions, and directors, so it is difficult to recognize. Furthermore, there are some punctuations the movie reviews used that I removed in the previous stage which could be the reason causing the wrong interpretation.

False Positive cases (Family Movie →Horror & Suspense)
The terms are found in the movie, Mad Hot Balloon. The movie genre should be Family Movie, but is predicted as Horror & Suspense:

Sensitive, smile, tears, tension, thrilling, treat, uninspired,
Unintentional, physical, poise, tentative, untidy, suspicious, icy, dramatized

1. The kids are remarkably open -- and tentative -- about their feelings.
2. This vastly entertaining documentary follows 10- and 11-year-olds as they gain poise and self-esteem through ballroom dancing.
3. persuasively shows how these lessons help bridge the gap between boys and girls during t hat uneasy period when the sexes are still suspicious of each other.
4. At its least inspired, it's like a dramatized newspaper feature story; at its best, it offers an unscripted slice of inner-city life...

Thought:
When I look at the reviews, the reviews seem to be fine to be interpreted as Family Movie. However, some of the movies in the original datasets are having Horror & Suspense, and Family Movie genres at the same time. I decided to give those movies belonging to the Horror & Suspense genre because the rows of Horror & Suspense movies are fewer than the Family Movie. I think that probably would be one of the reasons
Also, there are some punctuations that I removed in the text-preprocessing stage, which could influence the results.

Future Works:

In future work, I will need to consider how to distribute movie genres if one movie has multiple movie genres instead of simply picking one genre as a class label. After that, I will improve and adjust my approach from several aspects. Firstly, a spelling correction component will be added in the pre-processing of the reviews. Since there are still some words without meaning after data pre-processing. And setting a rating filter to reduce ambiguous reviews in the dataset. Secondly, more context information will be considered to perform word sense and opinion words. Furthermore, I will consider adding a neutral semantic orientation to my reviews more accurately because of the error analysis in experiment 1. There is some neutral semantics that is interpreted wrong and becomes movie genre words. In the machine learning models stage, I will refine the XGBoost parameters because I can obtain better performance if I properly choose the numbers.

References:

Aayush, B., (2019, Dec). *What does your classification metric tell about your data?* Towards Data Science. https://towardsdatascience.com/what-does-your-classification-metric-tell-about-your-data-4a8f35408a8b

Mukta, M. S. H., Khan, E. M., Ali, M. E., & Mahmud, J. (2017, May). Predicting movie genre preferences from personality and values of social media users. *In Eleventh International AAAI Conference on Web and Social Media,* (pp.1-4). https://ojs.aaai.org/index.php/ICWSM/article/view/14910

Nisha, A., (2022, Aug). *Tuning Random Forest Hyperparameters*. KD nuggets. https://www.kdnuggets.com/2022/08/tuning-random-forest-hyperparameters.html

Zhuang, L., Jing, F., & Zhu, X. Y. (2006, November). Movie review mining and summarization. *In Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 43-50). https://dl.acm.org/doi/pdf/10.1145/1183614.1183625