# The report of Movie analysis and prediction

**Contributors**:
Chuxuan Zhang(301267261)
Git Username: Shelly0814
Git email: shellyz@sfu.ca

Site Li (301244297)
Git Username: NoNameAA
Git email: bbmmddkk@gmail.com and sitl@sfu.ca

**Git URL:** https://github.com/NoNameAA/Movie_analysis (old URL)
https://csil-git1.cs.surrey.sfu.ca/sitel/CMPT_353_Project (new URL)

Because we missed the hint on the project requirement page, we use our previous git count to build the whole project. If TA and pro want to check the process, you can log to our old URL which is public.

**Git Tag:** Last_submitted_code (git tag on old URL)
submitted_code (git tag on new URL)

## Introduction

In this project, we chose the topic "Wikidata, Movie, success" to do some researches about movies with movie wiki data. We came up with 4 hypotheses about movie data mentioned below.

1. Do audience average and critic average have different means? Are they normal distribution? Do they have equal variance? Which test is better for testing means in this case? Do they have linear relationship?
2. If we have known that audience average and critic average have different means, what about audience average, critic average, audience percent and critic percent? Do audience average, critic average, audience percent and critic percent have a relationship? Are they normal distribution? Can we use different statistic tests to analyze if each two of them have different means or we cannot tell?
3. Can we use audience rating and critic rating data to predict whether the movies made profit? Is there a correlation between audience and critic? Which features are more important that effect models? Which models are the best for this problem and data?
4. Can we use the description of movies to predict the genres of movies? How to deal with multi-genres movies?

Basically, we use statistics models, machine learning models and natural language tools in order to deal with these problems.

## Implementation

**Part 1. The relationship between audience average and critic average - <u>compare_rating.py</u>**

In this file, we wanted to know that whether audience average and critic average have different means or not. We used process_data.py and rotten-tomatoes.json.gz to get the required data and read these data into a dataframe (rating_df).

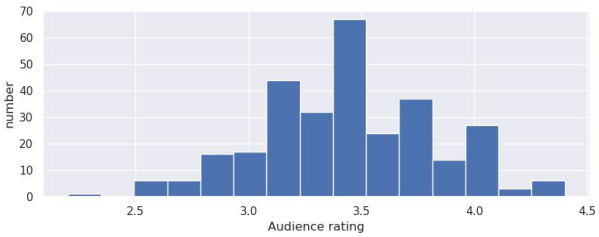
First, we cleaned the data. Because there were some Null data, we dropped the empty data in column audience average and critic average. We sorted the dataframe by audience_ratings in descending order. As audience_rating presents the count of reviews, the bigger counts of reviews mean the ratings are more accurate and normal. After a few attempts, we found that choosing the first 300 lines was suitable, so we chose the first 300 lines as the tested data. Because the audience rating was out of 5 and critic rating was out of 10, the standards of them were different. We converted the column critic_average into half so that both of them were out of 5. Rounded two column values to one-digit numbers. The data are as follow:

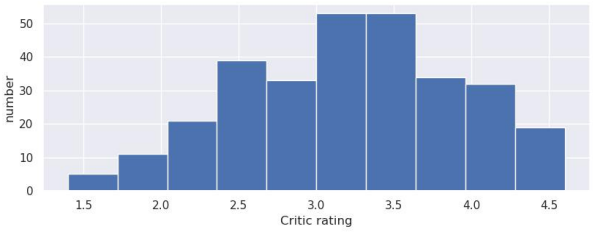
```
----- Data -----
audience_average      300
audience_percent      300
audience_ratings      300
critic_average        300
critic_percent        300
imdb_id               300
rotten_tomatoes_id    300
dtype: int64


2310    3.3
116     3.7
2698    3.3
8422    3.5
283     3.4
Name: audience_average, dtype: float64


2310    4.0
116     4.4
2698    3.8
8422    3.8
283     3.8
Name: critic_average, dtype: float64
```

Then we checked if audience average and critic average are normal distribution and have equal variance. We not only calculated the p-value of their normality and equal variance, but also printed some plots for helping understand.




```
----- T-test -----
pvalue of audience rating normality:  0.950683248464949
pvalue of critic rating normality:  0.0004153006379763149
pvalue of equal variance:  5.855281588473541e-23
Because the data n >= 40, it may ok for T-test
mean of audience rating:  3.4499999999999993
mean of critic rating:  3.199666666666668
pvalue of T-test:  1.3290841433004512e-07
Because pvalue < 0.05, they have different means
```

In T-test, We got that audience average is perfect normal distribution and critic average is similar to normal distribution. Because the data is 300 > 40, we can assume that it is normal too. The p-value of T-test is  1.3290841433004512e-07 which is smaller than 0.05, so audience average and critic average have different means. We used some if-sentences in the file to check that if two variables can do T-test. If audience_rating.count() >= 40 and critic_rating.count() >= 40, the system will print "Because the data n >= 40, it may ok for T-test" and do T-test. After getting the p-value of T-test, if the p-value is smaller than 0.05, the system will print “Because pvalue < 0.05, they have different means”. Otherwise, the system will print “Cannot use T-test”.

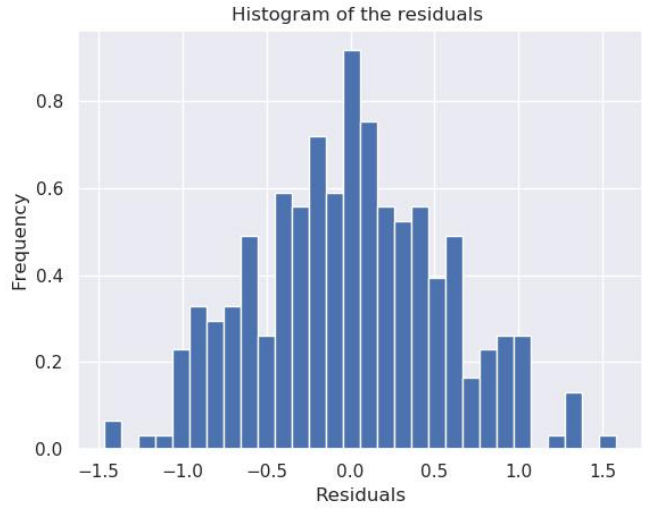
In U-test, both two variables do not need to check if they are normal distribution or have equal variance. The result of p-value is as follow:

```
----- U-test -----
pvalue of U-test:  2.966023328881923e-06
Because pvalue < 0.05, they have different means
```

Finally, we did an ordinary least squares for them. We created a scatter plot for comparing two variables and get a slope and intercept for a best fit line. We calculate the difference between the

critic average and the fit line which is residual and plotted a histogram of it. In addition, we got the slope, intercept, p-value, r-value, r-squared-value, stderr of variables. As the r-squared is 0.3476966583955792 which is smaller than 0.5, so there is no strong linear relationship between them.


```
----- Regression -----
LinregressResult(slope=1.0434737062644028, intercept=-0.4003176199455236, rvalue=0.5896580860088151, pvalue=1
.7546206286100598e-29, stderr=0.08279382008235293)
correlation coefficient r:  0.5896580860088151
r squared:  0.3476966583955792
```




In conclusion, audience average and critic average have different means, because p-value in all tests is smaller than 0.05. U-test is the most suitable for this case because it does not need normal distribution and equal variance. These two variables have no linear relationship because for r-squared.

**Part 2. The relationship between audience average, critic average, audience percent and critic percent - <u>compare_audi_crit.py</u>**

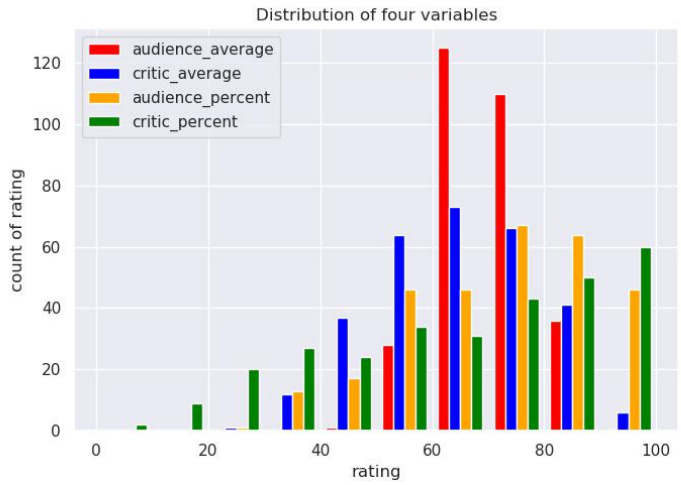
In this file, we have known that audience average and critic average have different means. What about audience average, critic average, audience percent and critic percent? We wanted to know if all of them have different means. We used process_data.py and rotten-tomatoes.json.gz to get the required data and read these data into a dataframe (rating_df).

First, we cleaned the data. Because there were some Null data, we dropped these empty lines. We sorted the dataframe by audience_ratings in descending order. As audience_rating presents the count of reviews, the bigger counts of reviews mean the ratings are more accurate and normal. After a few attempts, we found that choosing the first 300 lines is suitable, so we chose the first 300 lines as the tested data. Because the audience average is out of 5, critic average is out of 10 and both audience percent and critic percent are out of 100, the standard of them are different. For balancing their standard, the column of audience average multiple 20 and the column of critic average multiple 10 so that all the values are out of 100. The head() of the four variables are as follow:

```
----- Data -----
2310    66.0
116     74.0
2698    66.0
8422    70.0
283     68.0
Name: audience_average, dtype: float64
2310    80.0
116     87.0
2698    76.0
8422    77.0
283     75.0
Name: critic_average, dtype: float64
2310    69.0
116     86.0
2698    67.0
8422    69.0
283     74.0
Name: audience_percent, dtype: float64
2310    88.0
116     93.0
2698    89.0
8422    88.0
283     88.0
Name: critic_percent, dtype: float64
```

Then I checked if these four variables are normal distribution by calculating their p-value and plotted a histogram to show distribution. In this step, I tried to use log, exp, sqrt and times to make data normal distribution, but all these functions did not work. According to these p-values and the plot, audience average is perfect normal distribution. Because our data is 300 which is larger than 40, critic average and audience percent can be thought of as normal distribution. However, critic percent is not normal distribution which cannot use Anova test.

```
pvalue of audience average normality:  0.9506832484649477
pvalue of critic average normality:  0.000364324100140016
pvalue of audience percent normality:  0.00021976741208139873
pvalue of critic percent normality:  1.7030866813049488e-08
```

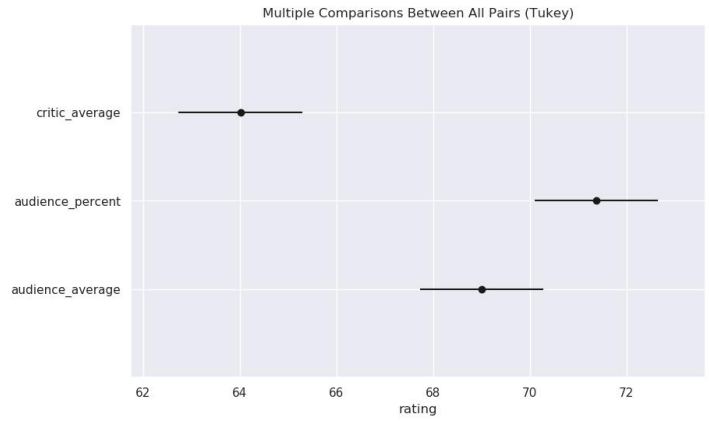



Next, we did anova between audience average, critic average and audience percent and got their normality p-value. If the p-value of anova is less than 0.05, the system will print "Because of pvalue < 0.05, there is a difference between the means of the groups.".As we can see, the p-value is 7.835753420783089e-11, so reject H0. We know that there is a difference between the means of the groups, but do not know which group it is.

```
----- Anova -----
pvalue of anova:  7.835753420783089e-11
Because of pvalue < 0.05, there is a difference between the means of the groups.
```

Finally, we did post hoc analysis to do pairwise comparisons between each variable. The result of post hoc Tukey test are as follow:

```
Do post hoc Tukey test
       Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================================
     group1           group2      meandiff  lower   upper  reject
-------------------------------------------------------------
audience_average audience_percent  2.3733  -0.1787  4.9253 False
audience_average  critic_average  -4.9867  -7.5387 -2.4347  True
audience_percent  critic_average   -7.36    -9.912  -4.808  True
-------------------------------------------------------------
```



From the table and plot, we can conclude that audience average and critic average have different means; so do audience percent and critic average. We cannot tell that audience average and audience percent have different means.

**Part 3. The relationship between rating and profit - movie_predict.py**

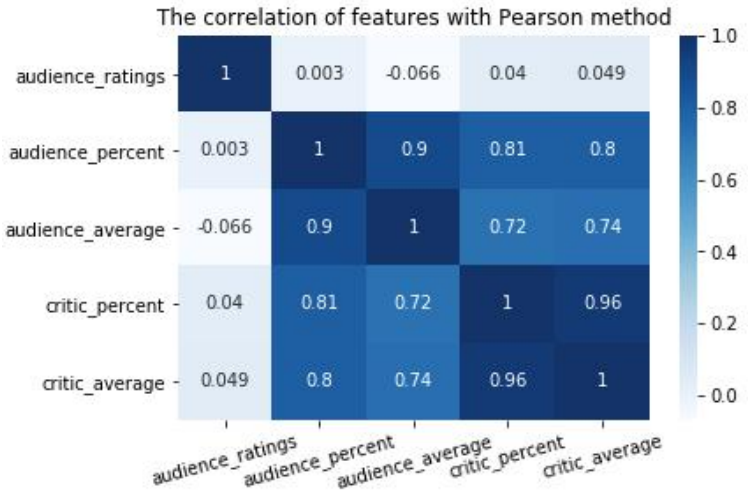
In this part, we verify whether a movie can make profit according to the rating of audience and critic. We used the datasets Wikidata-movies.json.gz and rotten-tomatoes.json.gz. we read the two datasets into dataframes, and joined the two tables by imdb_id. Then we selected all the columns about audience rating and critic rating which are audience_rating, audience_percent, audience_average, critic_percent, critic_average and made_profit. We found there are some missing values in this new table, so we selected the rows which have no missing values. Finally, we got a clean data table about rating and profit.

| | audience_ratings | audience_percent | audience_average | critic_percent | critic_average | profit |
|---|---|---|---|---|---|---|
| 0 | 4479.0 | 57.0 | 3.3 | 74.0 | 6.4 | True |
| 1 | 2421.0 | 60.0 | 3.4 | 76.0 | 7.0 | True |
| 2 | 26832.0 | 96.0 | 4.4 | 98.0 | 9.1 | True |
| 3 | 6801.0 | 57.0 | 3.4 | 89.0 | 7.3 | True |
| 4 | 39909.0 | 95.0 | 4.3 | 100.0 | 9.0 | False |

Next, we trained and test performance 3 machine learning models with the data. The 3 models are SVM Classifier model, Logistic Regression model, and Naive Bayesian models. Before training models, Since the features have different norms which cause the higher norm can effect model significantly, we normalize the features to the same norm. Also, we split data into training part and testing part. The last step is using training data to train models and using testing data to get the accuracy of models. The result showed below.

The accuracy of model with all rating features by Logistic Regression: 0.8424
The accuracy of model with all rating features by SVM Classifier: 0.8397
The accuracy of model with all rating features by Naive bayes Classifier: 0.5571

Depending on common sense, the percent of audience and average of audience have a correlation between each other, the same situation appears on the percent of critic and the average of critic, because if there are many critics like this movie, the average rating of this movie should be high. To verify our assumption, we used pandas correlation to calculate the correlation coefficients as well as shown by heatmap. The result showed the correlation coefficient between the percent of audience and average of audience is 0.9, and the percent of critic and average of critic is 0.96. That proves we can increase training speed and robust of models.




For removing features, we had two strategies which are feature selection and feature reduction. The random forest model is a better choice for feature selection, because this model has the importance coefficients of features, we can select the top-2 features according to the importance coefficients. PCA is another way to reduce the number of features, but it lacks interpretability of feature, because this model manufactures features by covariance matrix and selects the top-k importance features, so the features lost the information of original features. But it is still a good model to feature selection.

We kept top-2 important features to train the models. Even if the number of features decreases, the accuracy will not decrease and for different datasets, the data with two features are better than the data with all features. The accuracy figure sorted by accuracy showed below.

```
The accuracy of model with Top-2 important features by SVM Classifier: 0.856
The accuracy of model with Top-2 important features by Logistic Regression: 0.856
The accuracy of model with PCA transformed features by Logistic Regression: 0.8533
The accuracy of model with PCA transformed features by SVM Classifier: 0.8505
The accuracy of model with all rating features by Logistic Regression: 0.8424
The accuracy of model with all rating features by SVM Classifier: 0.8397
The accuracy of model with PCA transformed features by Naive Bayes Classifier: 0.6087
The accuracy of model with all rating features by Naive bayes Classifier: 0.5571
The accuracy of model with Top-2 important features by Naive Bayes Classifier: 0.3179
```

In this part, we conclude that we can use audience rating and critic rating data to predict whether the movies made profit by machine learning models; and according to the heatmap, we know the audience_percent correlates with audience_average, critic_percent correlates with critic_average; also we know the importance of features with random forest model; through comparing the accuracy figures we know the SVM and Logistic Regression model are the best for this problem.

## Further Discussion

Due to the limitation of time, we still have some good ideas or undetermined things that have not finished. We will discuss these in this part.

We have written a code (NLP_predict.py) which uses description of movie to predict the genres of movies, but we did not finished because the accuracy is pretty low(accuracy=0.2). We followed the tutorial "Working With Text Data" to try to build a machine learning model "MultinomialNB" to predict, it includes using countVectorizer and TfidfVectorizer to transform description and using LabelEncoder to encode genres of movie. However a movie maybe have more than one genres, if we encode the genres "Action, Adventure, Fantasy" to one class, the accuracy is only 0.05. So we did some attempts.

Attempt 1. Exploding the genres of movie into multiclasses which is the same as the exercise 12. And we can split 1 original row into multiple rows with same description and single genres, training model with new data, the accuracy increases to 0.2, it is seems better.

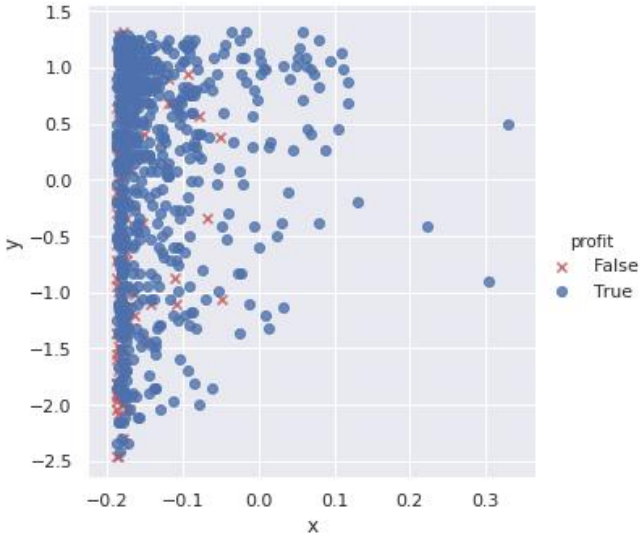
```
Prediction:
['Action' 'Adventure' 'Action' 'Adventure' 'Drama' 'Action' 'Drama' 'Adventure' 'Adventure' 'Action']

Observation:
['Thriller' 'Action' 'Adventure' 'Fantasy' 'Biography' 'Adventure' 'Biography' 'Action' 'Fantasy' 'Adventure']
```

Even if the accuracy increase to 0.2, the model only knows action and adventure movies.

Attempt 2. We still think 0.2 is pretty low. We considered if a movie has multiple genres in original data, the data in new dataframes has multiple rows, but some of rows are selected for train data, some are test data. So a new idea is sorting the prediction probability for every genres and select the top-k genres probability as predicted genres. Also we should write a new 'score' function to evaluate the models. We did not finish this NLP_predict.py due to limited time. And we are not sure that the attempt 2 will improve our machine learning model. At least, we know how to use natural language tools to transform text to matrix through writing this code, it is a great experience.

We realized unbalanced data is a bug of data during training models. After we finished **movie_predict.py** which uses rating to predict whether a movie made profit, we create a figure to show the distribution that which kind of movie can made profit and which not. The figure is with two feature and showed below.




There is a huge amount of points which are the movies made profit, and a small number of points which are the movies did not make profit. It causes the model only "remembers" the successful movie and "forgets" failed movies. We also came up with some new ideas to solve this problems.

1. Increasing the size of data to train models.
2. Depends on distribution of data, sampling subset of data to train model.
3. Simple evaluation of model do not fit this problem, use confusion matrix and ROC curve to re-evaluate model fairly.

These new methods can evaluate models fairly and increase the correct accuracy. Even if we did not finish this part, we got more experiences about training machine learning models and evaluating models.

## Project Experience Summary

Chuxuan Zhang:

- Reading, cleaning and manufacturing movie data with dataframe
- Checking if data is normal distribution and two variables have equal variance
- Applying T-test, U-Test and Ordinary least squares to analyze whether two variables have different means and linear relationship
- Compare T-test, U-test in different situations
- Applying Anova and Post Hoc Tukey Test to analyze multiple variables whether they have different means and which pairs have different means.

Site Li:

- Reading, cleaning and manufacturing movie data with dataframes
- Using rating features to training models and predict profit, evaluate different models performance
- Applying feature selection and feature reduction to original data with random forest and PCA in order to increase robustness and decrease overfitting
- Using Tfidf vectorizer of NLTK to transform the description of movies to matrix, applying LabelEncoder to preprocess genres of movies.