Netflix Rating Recommendation Analysis

Chia Wei Tu 300289967

Introduction and discovery

- Background

Netflix canceled user film review system and use artificial intelligence algorithms to recommend movies in these years. For getting more viewers, Netflix try to obtain viewer's flavor of movie and recommend more feature of movie which match with user's taste. Also, use the most popular result for increasing Netflix's movie variety.

- Problem

Does the rating score really indicate how well Netflix system feel the recommended content fits the specific user? Does the recommendation score really help user find their love shows and movies?

- Developing initial hypotheses

If the Netflix rating recommendation is higher than scores 80, The result is yes (highly recommend).
If the Netflix rating recommendation is lower than scores 80, The result is No (not recommend).

Data Preparation

The Dataset is form Data World https://data.world/chasewillden/netflix-shows

The dataset is about Netflix each title (movie name)'s rating information, including the rating category for different ages(rating), the description of rating, the release year, user rating size (only more than 80 sizes can be record on the user score standard) and the rating score.

- Original Dataset:

title	rating	ratingLevel	ratingDescription	release year	user rating score	user rating size
White Chicks	PG-13	xual humor, language and some	80	2004	82	80
Lucky Number Slevin	R	olence, sexual content and adult l	100	2006	NA	82
Grey's Anatomy	TV-14	oned. May be unsuitable for childr	90	2016	98	80
Prison Break	TV-14	oned. May be unsuitable for childr	90	2008	98	80
How I Met Your Mother	TV-PG	ce suggested. May not be suitable	70	2014	94	80
Supernatural	TV-14	oned. May be unsuitable for childr	90	2016	95	80
Breaking Bad	TV-MA	ces. May not be suitable for chil	110	2013	97	80
The Vampire Diaries	TV-14	oned. May be unsuitable for childr	90	2017	91	80
The Walking Dead	TV-MA	ces. May not be suitable for chil	110	2015	98	80

- Compute length of rating level and add to a new column

- Examine the categorical columns in the netflix rating to 5 ages level - **Preschool, child, Juvenile, Teenager, Adult**

- Categorize columns in the user_rating_score, if user_rating_score >=80, highly recommend=yes, otherwise=No

Create dummy to transformation variables

	ratingDescription	release_year	ratingLevel_len	rating_Child	rating_Juvenile	rating_Preschool	rating_Teenager	Highly_recommend_Yes
count	600	600	600	600	600	600	600	600
mean	73.23666667	2010.785	54.22833333	0.315	0.018333333	0.171666667	0.313333333	0.706666667
std	27.86044381	7.752781288	23.51079453	0.464903458	0.134265661	0.377405105	0.464235659	0.455669779
min	10	1982	3	0	0	0	0	0
25%	60	2007	38	0	0	0	0	0
50%	80	2015	66	0	0	0	0	1
75%	90	2016	77	1	0	0	1	1
max	124	2017	80	1	1	1	1	1

Model Planning and Implementation

#!pip install yellowbrick
 # import yellowbrick library
from yellowbrick.cluster import KElbowVisualizer

The original data have overlapping the same ages but different TV rating names, it can be categorized to 5 different age levels. And the user rating score is from the lowest score 55 to highest score 99 excluding the 0 (unrated movie). So, I planned to divide the score to two results- Not Recommend and Highly Recommend. Over 80 are yes for recommend, others are no for recommend. For the Not Recommend and Highly Recommend target, it can change it to the binary category.

Base on the data transform, I create two series of machine learning pipeline. One is for the clustering analysis which can find the related group in different rating.

The clustering can be analyzed into more detail in each rating group. For example, the teenager can be divided by 7 classes (k=7). So, this can observe the relationship between each specific user's reaction and in different flavor of movies (like release year).

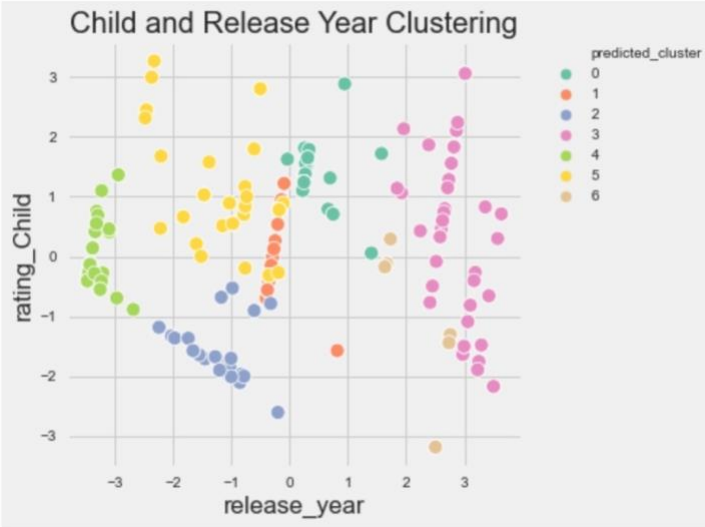
Second is for Logistic regression and classifications algorithms to make predictions for the recommend result. Logistic regression is mostly used to solve binary classification.

After through the classification pipeline model, it can use the best model to predict the release year, different user group or rating description whether will have a good effect on the result of recommendation.

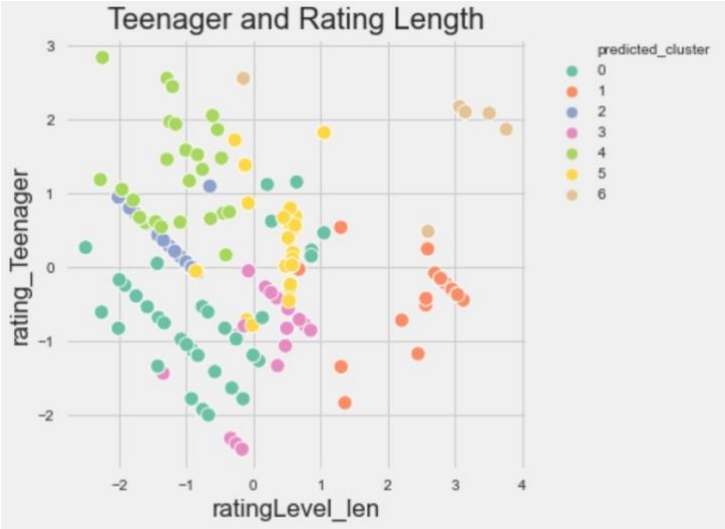
I use list to store every machine learning models and best parameter. And then zip all the list into for loop to use GridSearchCV finding the optimal parameter and pipeline predicting the X test and then append to the result list.

Results Interpretation and Implications

Clustering Analysis



The plot shows that 7 clusters display by child and release year. It is good for Netflix to analyze different release years of movie to the user that in the same child program rated.



The plot shows that 7 clusters display by teenager and rating length. There has a specific length of rating description have fit more teenager. Netflix's program rated description is good for determine the result of recommendation.

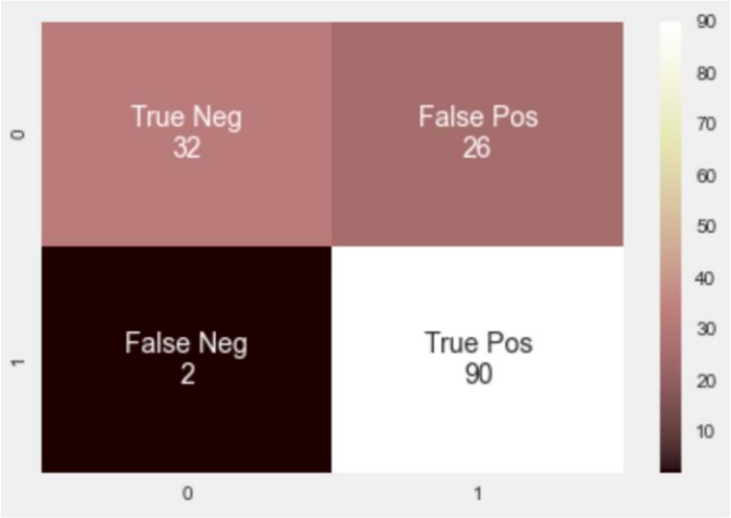
Multiple Regression and Classification Analysis

```
Detailed classification report:
                  precision    recall  f1-score   support

     Not Recommend       0.94      0.55      0.70        58
Highly Recommend         0.78      0.98      0.87        92

          accuracy                           0.81       150
         macro avg       0.86      0.76      0.78       150
      weighted avg       0.84      0.81      0.80       150
```

The accuracy of the data in classification report is 84% which is a nice percentage of correct predictions for the test data in random forest model. The model shows there are 78% accuracy of positive predictions for yes of highly recommend and 98% positives that were correctly identified for yes of highly recommend.
It is a nice result of Netflix rating recommendation; it really can help specific users finding their favorite shows or movie depend on the recommendation system. Most of users will give thumbs up for the movie depend on its release year, the description or the programs rated.

	Classifier	Best Parameters	Accuracy Score	RMSE
0	Logististic Regression	LogisticRegression(max_iter=300, solver='sag')	0.626667	0.611010
1	AdaBoostRegressor	KNeighborsClassifier(algorithm='ball_tree', we...	0.806667	0.439697
2	SVM	SVC(C=1, kernel='poly')	0.660000	0.583095
3	AdaBoost	(DecisionTreeClassifier(max_depth=1, random_st...	0.786667	0.461880
4	MLPClassifier	MLPClassifier(activation='tanh', hidden_layer_...	0.693333	0.553775
5	RandomForest Classifier	(DecisionTreeClassifier(max_depth=8, max_featu...	0.813333	0.432049
6	Decision Tree Classifier	DecisionTreeClassifier(criterion='entropy', ma...	0.780000	0.469042
7	Naive Bayes	GaussianNB(var_smoothing=1.0)	0.660000	0.583095
8	XGBoost	XGBClassifier(base_score=0.5, booster='gbtree'...	0.806667	0.439697
9	Gaussian Process	GaussianProcessClassifier()	0.686667	0.559762
10	SGD Classifier	SGDClassifier(alpha=0.001)	0.666667	0.577350

The best model score is Random Forest Classifier which is 82% and have the lowest RMSE. I use Grid Search CV to find the optimal parameter which the max depths set to 8 and the n estimators set to 10.



There are almost 90 cases was positive and predicted positive. It is good for getting higher correct predictions.
There are 26 and 2 cases of false positive and false negative. Compare to the true positive, it seems has avoid some intolerable mistakes.

I think it should have more data in Netflix dataset, the user will keep growing up and change their interest of movie quickly. The more updated dataset is needed. But at the same time, the old data should be deleted and replaced by new data.

Using clustering and classification algorithms method can answer the question “Does the recommendation score really help user find their love shows and movies?” The Netflix recommendation system can compute the algorithm that the user’s favorite movie such like some people like to watch old school style and some people like to watch cartoon films.
What is more, the question “Does the rating score really indicate how well Netflix system feel the recommended content fits the specific user?” I think it really can help different ages people finding what they are able to watch depend on their program rated category.

Out-of-sample Predictions

Import a new sample csv file ‘X_New_Sample.csv’ to a final model
There have 5 samples to be predict in the model.

Import New Sample

```
#Import the new data for prediction
Xnew = pd.read_csv("X_New_Sample.csv")

#5 samples
Xnew
```

	ratingDescription	release_year	user_rating_size	ratingLevel_len	rating_Child	rating_Juvenile	rating_Preschool	rating_Teenager
0	70	2015	80	66	1	0	0	0
1	60	2016	81	57	1	0	0	0
2	35	2018	80	3	0	0	1	0
3	90	2020	80	77	0	0	0	1
4	60	2014	82	26	1	0	0	0

After Scaling and transform the sample data then perform predictions using new sample data

```
# make a prediction for new sample
#Random Forest is the Best model
ynew = rf.predict(scaled_Xnew)
ynew
```

```
array([0, 1, 1, 1, 1], dtype=uint8)
```

```
#result
i=0
for r in ynew:
    if r ==0:

        print("Sampe ",i,"Predicted result is Not Recommend")
        i += 1
    else:
        print("Sampe ",i,"Predicted result is Highly-Recommend")
        i += 1
```

```
Sampe  0 Predicted result is Not Recommend
Sampe  1 Predicted result is Highly-Recommend
Sampe  2 Predicted result is Highly-Recommend
Sampe  3 Predicted result is Highly-Recommend
Sampe  4 Predicted result is Highly-Recommend
```

The result is 4 out of 5 in highly recommend. It is quite high of result in this model.
It means Netflix have make a nice classification in different category, each age or release years.
It makes the rating score more reliable for the recommendation.

Concluding Remarks

As the number of people subscribing and watching Netflix grew, the task became a big data project. Netflix rating recommendation system can help viewers by choosing among numerous options available to them through the streaming service. It also can filter the program rated group and remove unnecessary information from the data stream before it reaches to them.
Moreover, through the highly recommendation system, it can increase the viewer's trust of selecting the films from recommendation which they like a lot and then giving a nice feedback.
If Netflix rating recommendation system obtain more updated and enough size of data, it seems to be a ideally cycle between user and business strategy.