# CS418 Project 3

# **Regression, Classification and Clustering**

Frank Errichiello, Tomasz Hulka, Ankit Kumar Singh

## Overview

The purpose of this report is to use the steps of the data science pipeline to solve a problem we identified using real-world data.

## Problem Selection

Every year hundreds of movies are released, with only some making it to mainstream theaters. The problem we set out to solve is to find the best attributes to predict the popularity of a given movie and find the best performing model. Our data science solution will use the details involved in the production and release of a movie to make a prediction on how they influence the popularity of a movie. Using the data modeling techniques of linear regression and clustering we can compare the effectiveness of certain variable selectors for projecting popularity.

## Data Selection

For our analysis we used the film data API by The Movie Database (TMDB) which contains information on 4803 movies released before October 2017. The database is hosted on Kaggle (https://www.kaggle.com/tmdb/tmdb-movie-metadata?select=tmdb_5000_movies.csv). It contains the following variables:

- budget: The budget used to make the movie.
- genre: The genre of the movie.
- homepage: A link to the website of the movie.
- id: This matches movie_id in the first dataset.
- keywords: Keywords related to the movie.
- original_language: The language in which the movie was made.
- original_title: The title of the movie before translation or adaptation.
- overview: Synopsis of the movie.
- popularity: A numeric quantity specifying the movie popularity.
- production_companies: The production house of the movie.
- production_countries: The countries in which the movie was made.
- release_date: The date on which the movie was released.
- revenue: The revenue generated by the movie.
- runtime: The time of the movie in minutes.
- spoken_languages: a list of the languages spoken in the film.
- status: If the movie has been released or rumored to be released
- tagline: Movie's tagline.

- title: Title of the movie.
- vote_average: average ratings the movie received.
- vote_count: the number of votes received.

## Data Preparation

For our preparation of the data source we removed the bulk of columns we wouldn't be using first. This included "homepage", "tagline", "overview", and "keywords". After removing these we dropped the rows with null values. For the third step we removed any rows with the status of "Rumored" or "Post Production" as this means they weren't released and would lack data.

We then created more useful column values using the preset columns of "budget" and "revenue" to create a profit column. We also used "genres", "production_companies", "production_countries", and "spoken_languages" to create new columns giving us the total of each of these columns. After these columns were created we dropped the original "genres", "production_companies", "production_countries", and "spoken_languages".

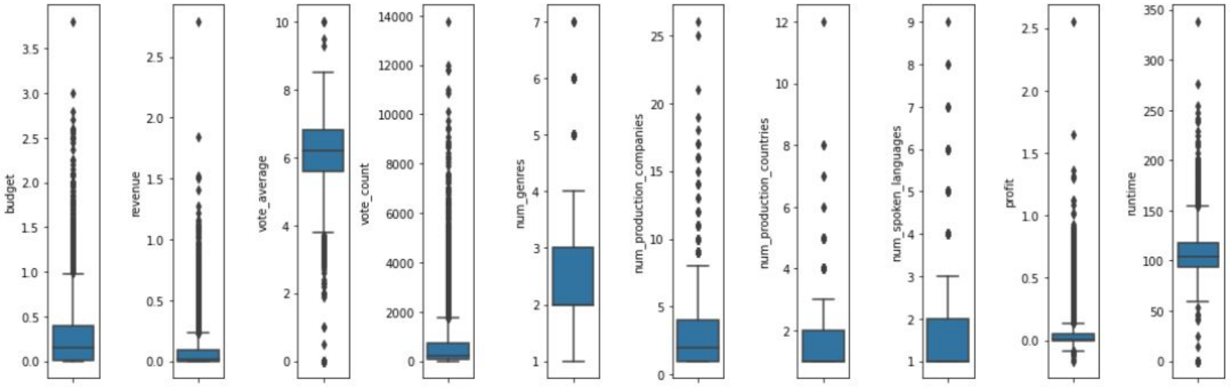
The final variables we were left with were:

- budget: The budget used to make the movie.
- id: This matches movie_id in the first dataset.
- original_language: The language in which the movie was made.
- original_title: The title of the movie before translation or adaptation.
- popularity: A numeric quantity specifying the movie popularity.
- release_date: The date on which the movie was released.
- revenue: The revenue generated by the movie.
- runtime: The time of the movie in minutes.
- status: If the movie has been released or rumored to be released
- title: Title of the movie.
- vote_average: average ratings the movie received.
- vote_count: the number of votes received.
- Num_genres: the number of genres the movie falls under
- Num_production_companies: the number of companies that produced the movie
- num_production_countries: the number of countries the movie was made in
- num_spoken_languages: the number of languages the movie is available in
- profit: This is the budget subtracted from the revenue

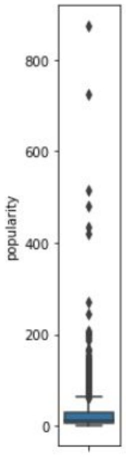
After processing the data we split the data into a training and test set using the holdout method with a training size of 80% and a test size of 20%. We then standardized the set using a standard scaler.

# Data Exploration

Box plots of the variables being used to predict the popularity of the movies



Box plot of the variable being predicted (popularity)

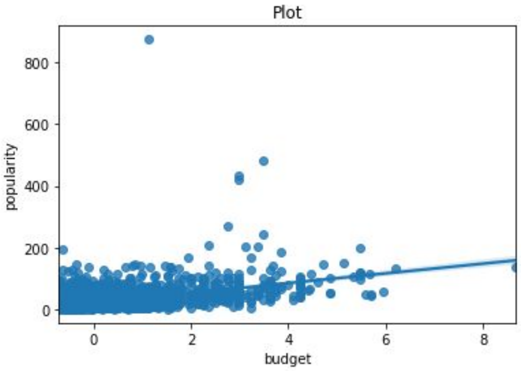


From these plots we can see that budget, revenue, and profit all have similar plots, this makes sense as the profit column is completely based on the revenue and budget. We also see that the profit and vote_count columns are very similar to the popularity column and lead us to believe that these would be the best for predicting the popularity.

# Data Modeling

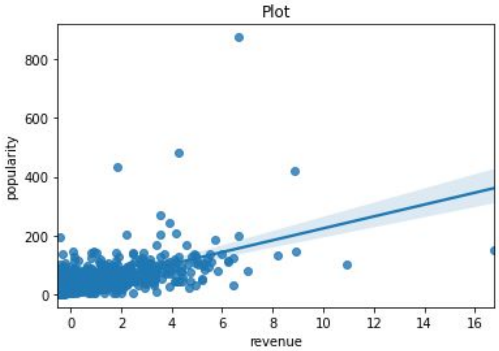
## Simple linear regression

Using **budget** as the predictor variable.



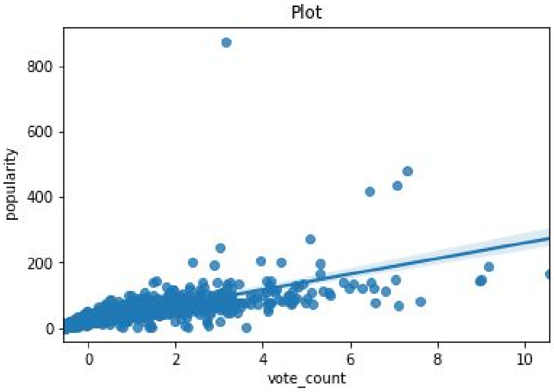
Model coefficient [15.80220667]
$R^2$ value:  0.22630617459143088
Adjusted $R^2$ value:  0.22610421848978413

Using **revenue** as the predictor variable.



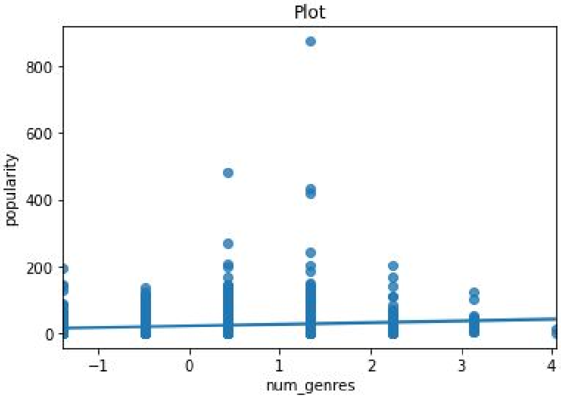
Model coefficient [20.27355683]
$R^2$ value:  0.3550460973010354
Adjusted $R^2$ value:  0.3548777459821373

Using **vote_count** as the predictor variable.



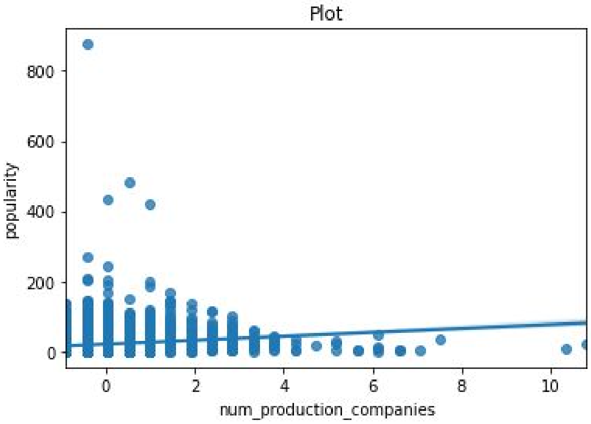
Model coefficient [23.80864697]
$R^2$ value:  0.6300683998976688
Adjusted $R^2$ value:  0.6299718372247107

Using **num_genres** as the predictor variable.



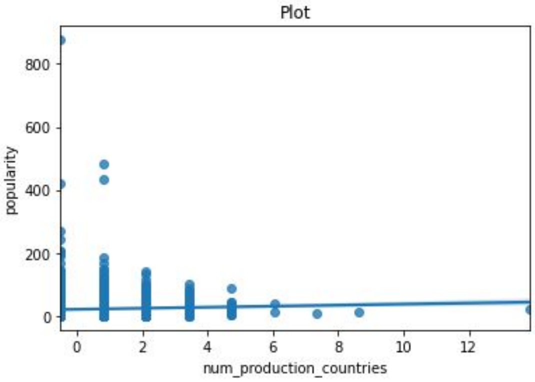
Model coefficient [5.08273789]
$R^2$ value:  0.011735850028733206
Adjusted $R^2$ value:  0.011477884967398988

Using **num_prudction_companies** as predictor variable.



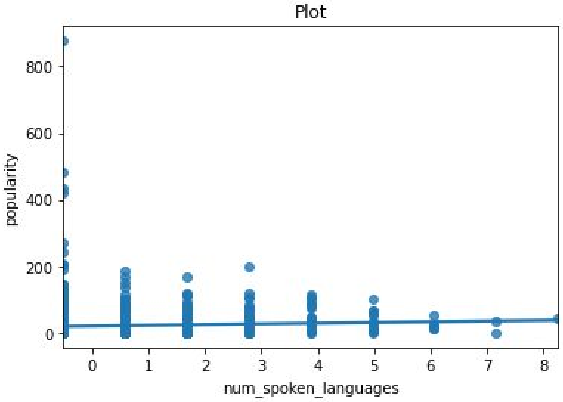
Model coefficient [5.58037696]
$R^2$ value: 0.02470487104907019
Adjusted $R^2$ value: 0.024450291271218227

Using **num_prudction_countries** as the the predictor variable.



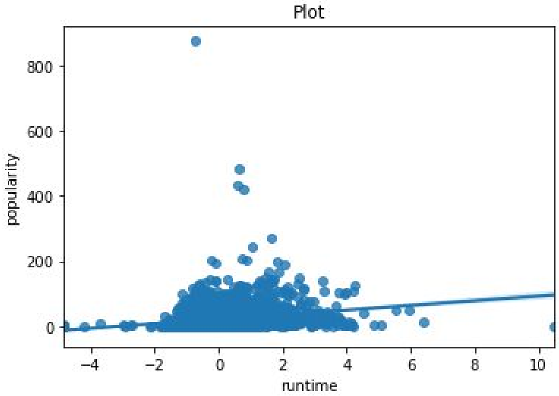
Model coefficient [1.64132386]
$R^2$ value: 0.0061964533838043045
Adjusted $R^2$ value: 0.005937042382338276

Using **num_spoken_languages** as the predictor variable.



Model coefficient [2.08176019]
$R^2$ value: 0.0005629631593411366
Adjusted $R^2$ value: 0.0003020816566419171

Using **runtime** as the predictor variable.



Model coefficient [7.0555049]
$R^2$ value: 0.04410121715497607
Adjusted $R^2$ value: 0.0438517003753246

## Multiple linear regression

1. Using all features as the predictor variables.

   Model coefficient [ 1.57008307,  1.73541017,  1.65110193, 19.56095742,  0.91619063, 1.56916315, -0.11768996, -0.81849668,  1.60672731, -0.35812605]
   R_squared value:  0.6123210704933378
   Adjusted R_squared value:  0.6113067352513005

2. Using **budget**, **num_production_companies**, and **runtime** as the predictor variables.

   Model coefficient [14.60618505  2.36935714  2.66320166]
   R_squared value:  0.23495926005789286
   Adjusted R_squared value:  0.2343598549338849

3. Using **vote_count** and **profit** as the predictor variables.

   Model coefficient [21.23768393  3.38028349]
   R_squared value:  0.6188602511867185
   Adjusted R_squared value:  0.6186612225972599

### Conclusions based on regression models

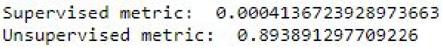
Among all variables the "vote_count" alone predicts the popularity with highest accuracy as expected with Adjusted $R^2$ value of 0.629.

Using all the features known before the release of the movie, the "budget", "num_production_companies", and "runtime" are the three variables that predict the popularity with best Adjusted $R^2$ value of 0.234, which is still low but is best among these features to reflect the popularity.

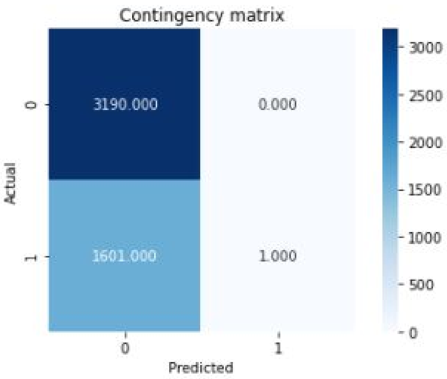
We observed that the best performance was given by the multiple linear regression model with "vote_count" and "profit" as the predictor variables. As we added more predictor variables to these two, the performance slightly decreased which can be explained by the overfitting of the model.

# Clustering

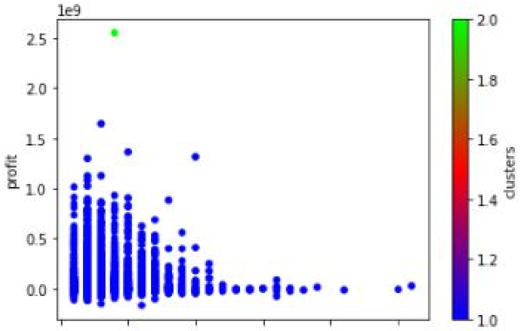
Hierarchical clustering with single linkage and variables **profit, num_production _companies, num_spoken_languages**

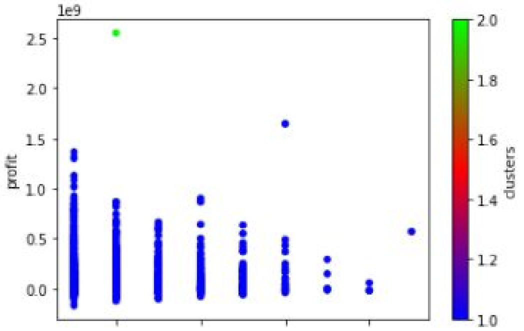


```
Supervised metric:  0.0004136723928973663
Unsupervised metric:  0.893891297709226
```

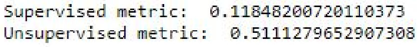


Contingency matrix

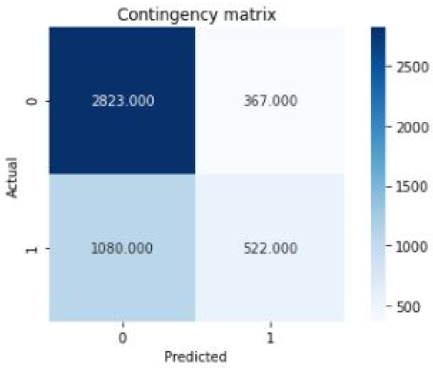




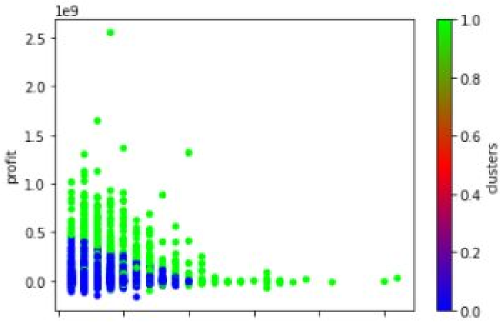
K-means clustering with k = 2 and variables **profit, num_production_companies, num_spoken_languages**

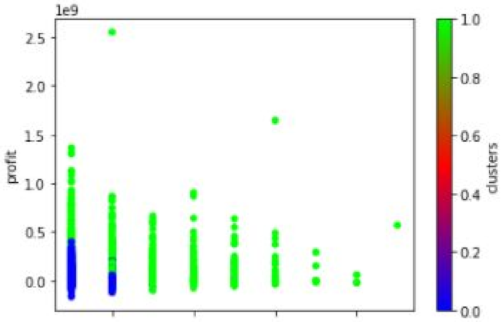


```
Supervised metric:  0.11848200720110373
Unsupervised metric:  0.5111279652907308
```



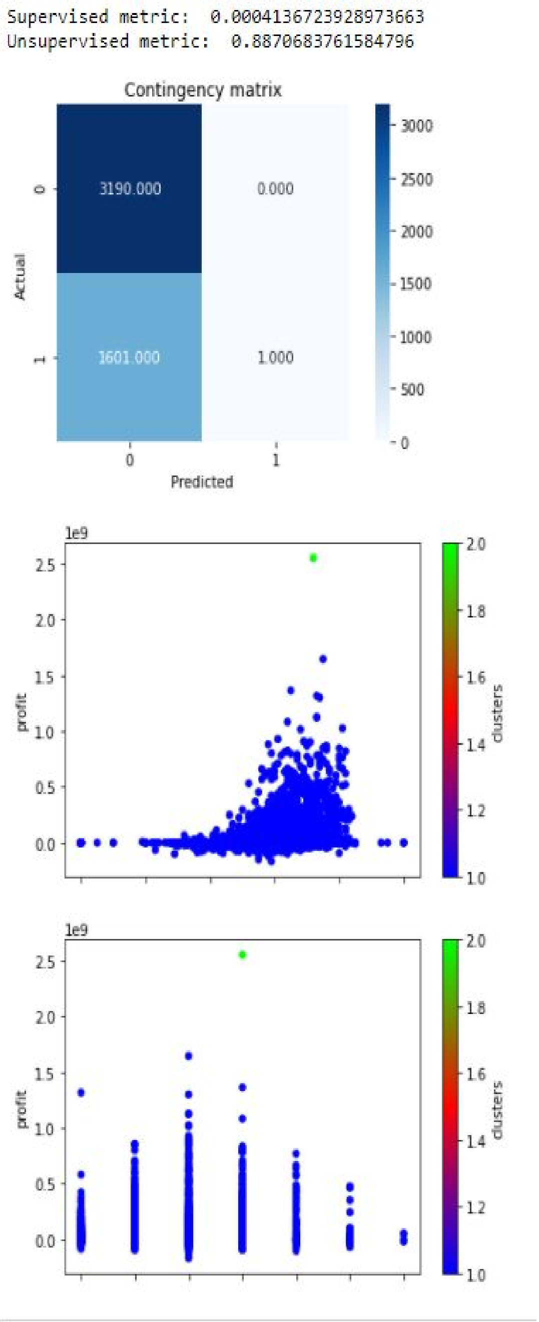
Contingency matrix



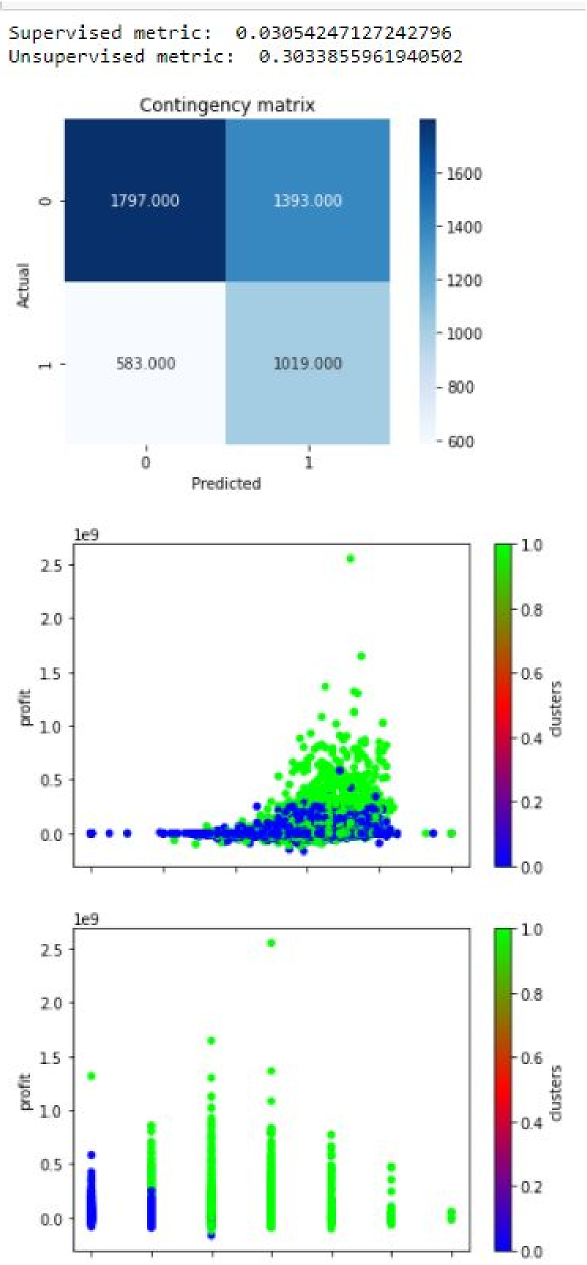


Hierarchical clustering with single linkage and variables **profit, vote_average, num_genres**

K-means clustering with k = 2 and variables **profit, vote_average, num_genres**



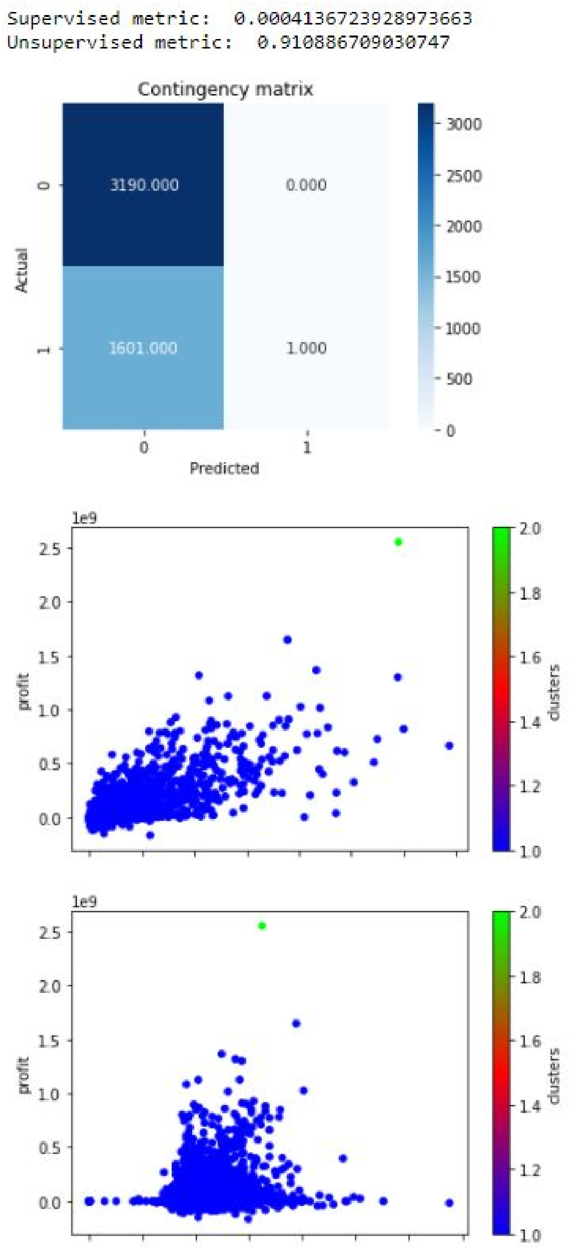
```
Supervised metric:  0.0004136723928973663
Unsupervised metric:  0.8870683761584796
```



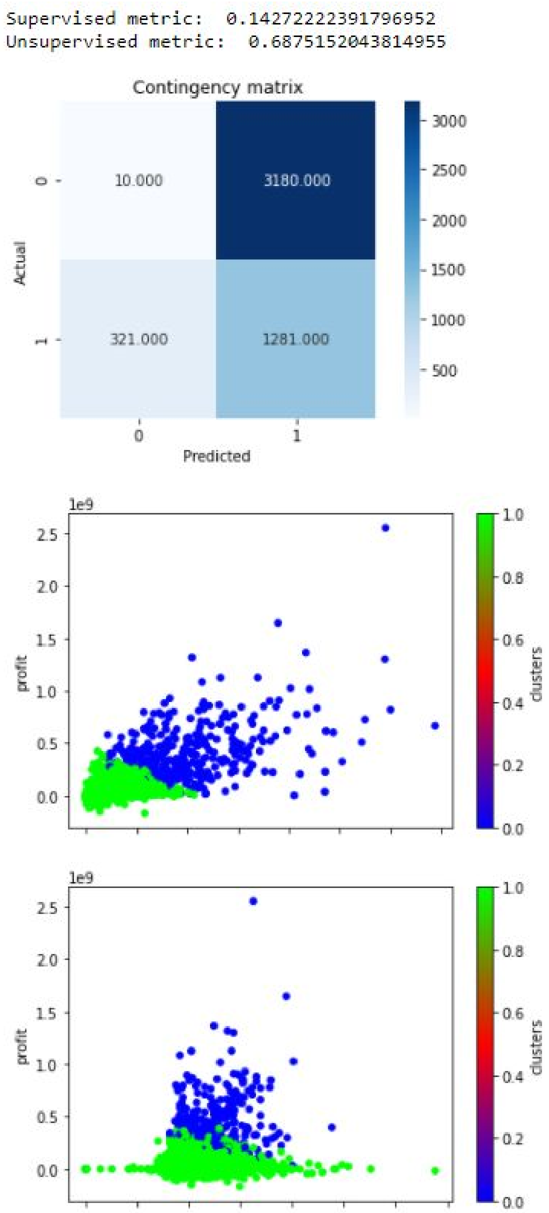
```
Supervised metric:  0.03054247127242796
Unsupervised metric:  0.3033855961940502
```

Hierarchical clustering with single linkage and variables **profit, vote_count, runtime**

K-means clustering with k = 2 and variables **profit, vote_count, runtime**



```
Supervised metric:  0.0004136723928973663
Unsupervised metric:  0.910886709030747
```



```
Supervised metric:  0.14272222391796952
Unsupervised metric:  0.6875152043814955
```

### Conclusions based on clustering models

The best performing model was K-means clustering with two clusters and using variables 'profit', 'vote_count', 'runtime'. It had a supervised metric of .143 and unsupervised metric of .688. The supervised metric was the highest out of all of our clustering models. This can be explained by the use of predictor 'vote_count' which would suggest more popular movies have more votes.

In general the K-means clustering always outperformed the Hierarchical clustering with single linkage. For the models using Hierarchical clustering, the supervised metric was almost always 0, suggesting that the clusters were not very similar.

## Overall Conclusion

Both data modeling techniques of linear regression and clustering support that the best predictors for a movie's popularity are the number of ratings it received (vote_count) and the total profit generated (profit).