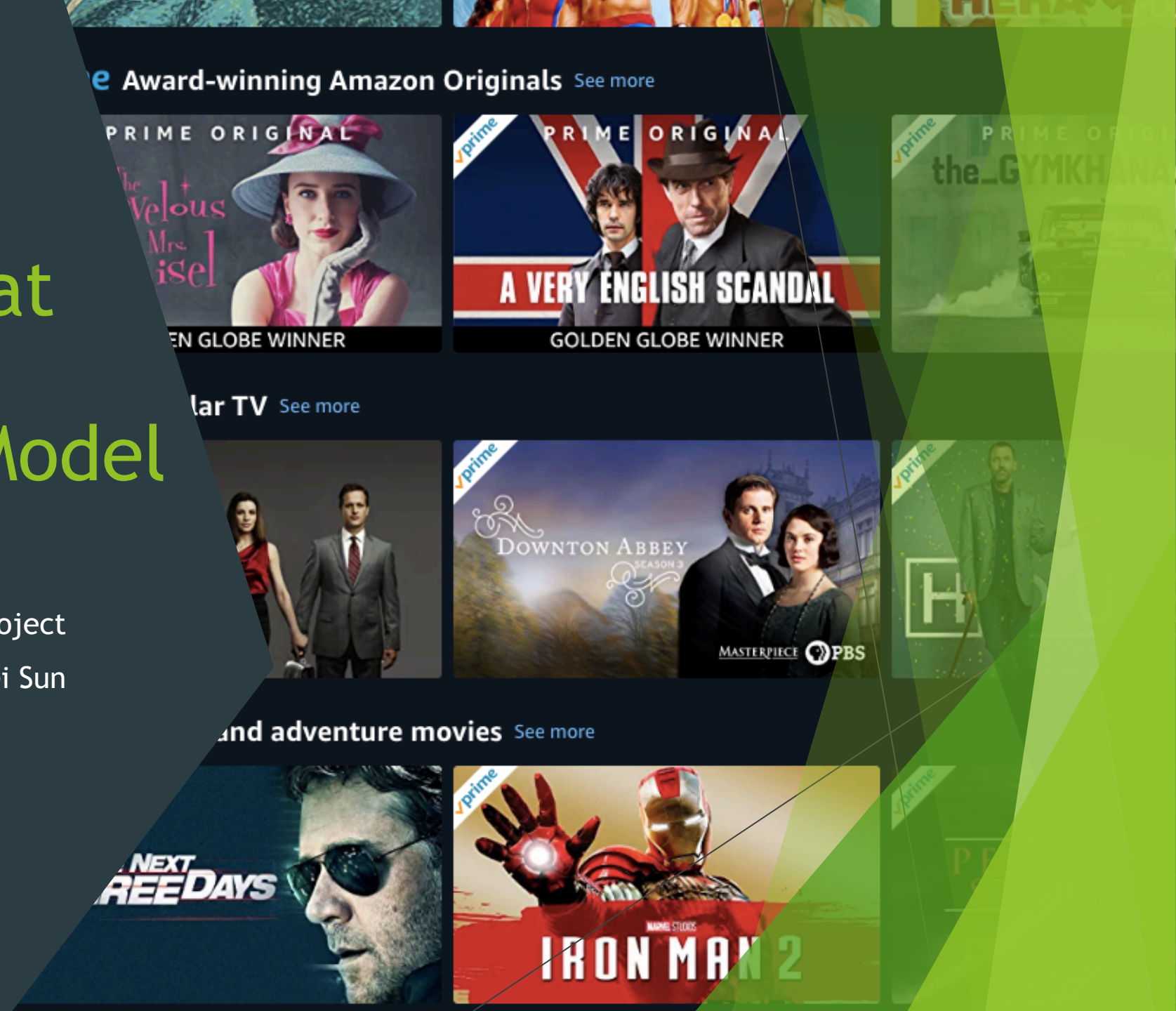
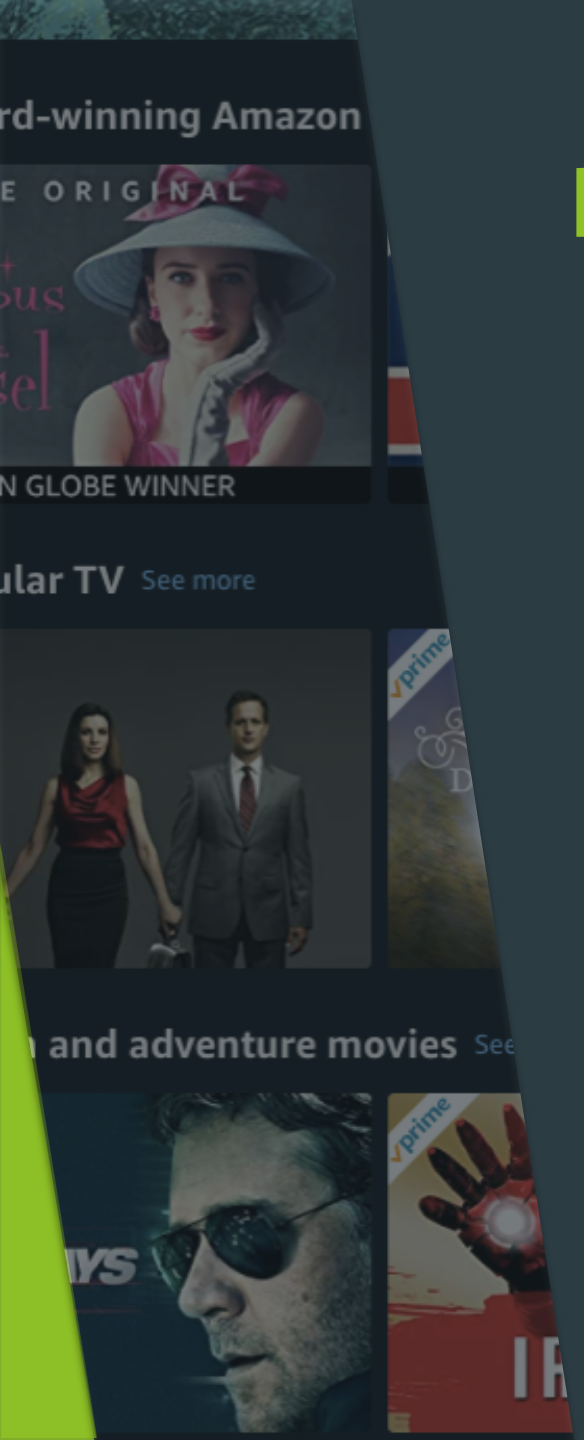
Predicting Great Movies Using a classification Model

EM623 Final Project

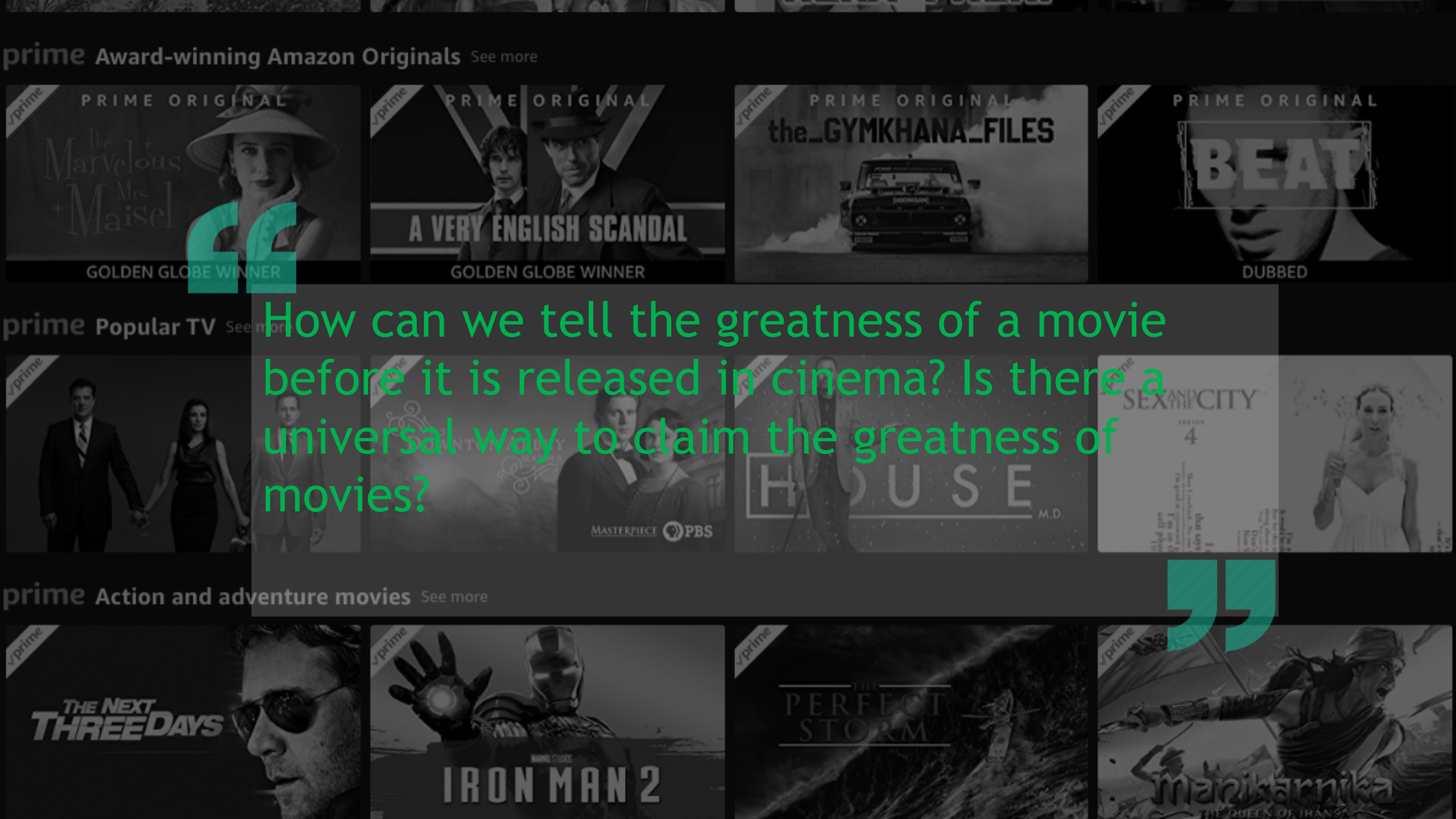
Lei Sun



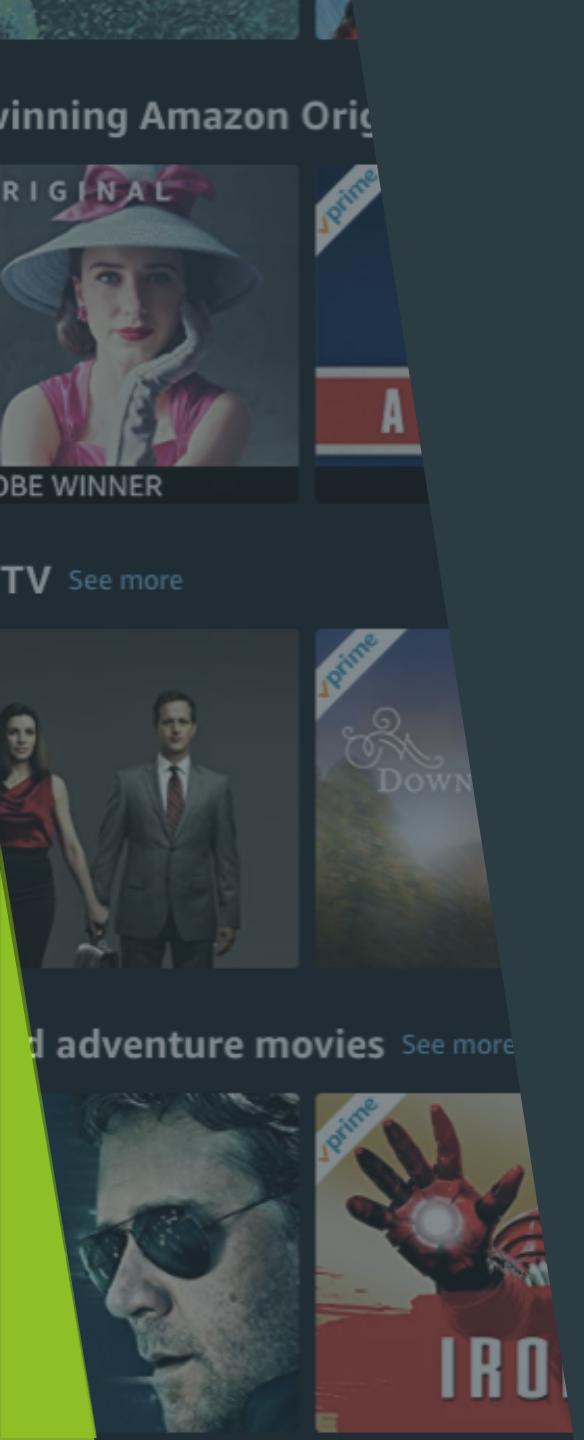


PROJECT GOALS AND CONDITIONS

- Determining what kind of movies are considered as popular can be a challenge. Different kinds of movies are made every year. Some series of movies gain high revenue; some rewards professional academy awards; some are successful for both aspects.
- People usually decide whether or not to see a movie based on subjective opinions or get influenced by reviews of others and social media advertising.
- It is important for Film Studios and Film Producers to create “successful” movies, which may bring sufficient revenues. So to find what kind of movie are audiences’ favorite is helpful.
- This project proposes a way to predict how a great a movie is, using a supervise algorithm to build the predict model.



How can we tell the greatness of a movie before it is released in cinema? Is there a universal way to claim the greatness of movies?



BUSINESS UNDERSTANDING

- To answer this question, I utilize a list of movies with various features to determine what “factors” are essential to let a movie be great, Since people usually decide whether or not to see a movie based on personal favors.
- **Business objectives:** Determine what the factors for creating a great movie.
- **Strategy to meet objectives: I** use a dataset which contains 5,000+ movies in history record to implement a predict model following CRISP-DM approach and then through validation to figure it meets the business objectives.

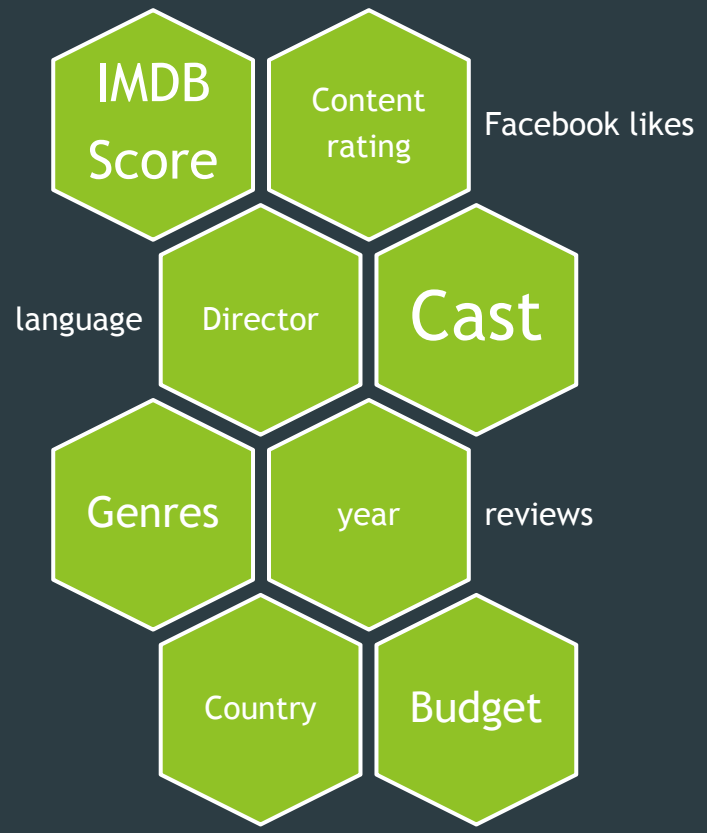
DATA UNDERSTANDING

- **IMDB 5000 Movie Dataset:**
- The dataset obtain 28 variables for 5043 movies, spanning across 100 years in 66 countries. There are 2399 unique director names, and thousands of actors/actresses.

- It was provided by a researcher who applied the human face detection algorithm on all the posters using the Python library called dlib, and extracted the number of faces in posters. Posted on data.world from Chuan Sun (@sundeepblue on Github) scraped tons of metadata using a combination of www.the-numbers.com, IMDB.com, and a Python library called "scrapy".

- Sources: https://data.world/popculture/imdb-5000-movie-dataset

DATA UNDERSTANDING- Characteristics



The dataset has a few numerical columns representing the measures of numbers of Facebook likes in movies, rating, year

It also contains some nominal columns for the information film directors and casts; category of movie genres and text of reviews

DATA PREPARATION- Data Cleaning

- Using KNIME to process the data preparation
- First, perform file loading
- Second, process missing values, outliers and normalization

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
num_critic_for_reviews	1	140.1943	?	813	121.6017	1.5165	2.9134	50	0	0	1 813
duration	7	107.2011	?	511	25.1974	2.3391	22.5658	15	0	0	7 511
director_facebook_likes	0.0	686.5092	?	23,000	2,813.3286	5.2297	27.2563	104	0	0	0 23.000
actor_3_facebook_likes	0.0	645.0098	?	23,000	1,665.0417	7.279	60.5639	23	0	0	0 23.000

- Third, filter and eliminate those variable which obviously irrelevant to the prediction, most of them are string type

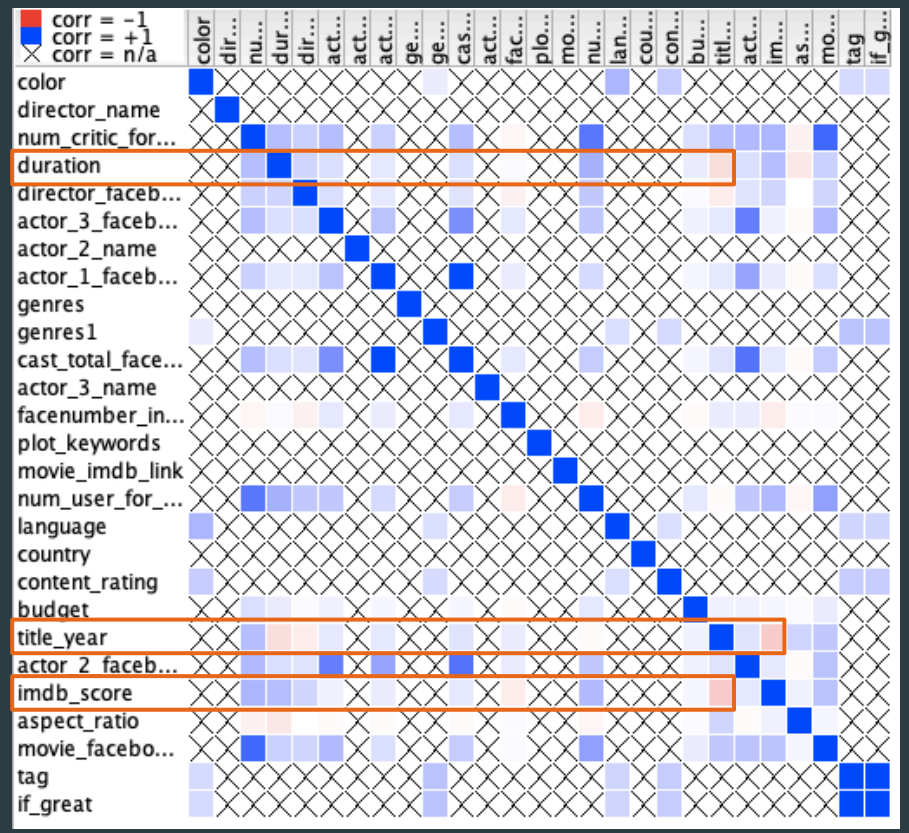
DATA PREPARATION-Data Transformation

- Next, derived a target variable for predict and evaluation purpose: I created a new column with value of Yes/No according to imdb_score. It is target attribute.
- The rule is set above a particular score, the movie is labeled as great. Using the function of =IF(imdb_score>=7.5,"Yes","No") to form the variable value.

director_name	imdb_score	if_great
James Cameron	7.9	Yes
Gore Verbinski	7.1	No
Sam Mendes	6.8	No
Christopher Nolan	8.5	Yes
Doug Walker	7.1	No
Andrew Stanton	6.6	No

- The last column contains category tag “if_great” which represent whether a movie is label as great.

DATA PREPARATION-Correlation Measurement



- Although most variables are independent variables, meaning there are not correlated with each other, some variables may be affected by each other or strongly correlated to the tag
- Using Linear-correlation to identify those variables and then filter them. As is shown, variables: duration,title_year, actor_facebook needs to be eliminated
- Correspondingly, some variables contains duplicated information such as language\country, genres1/2/3, filter them as well
- Then I have the final dataset with 14 variables and 1 label variable.

MODELING-Algorithm Selection

- The goal is to train a Decision Tree to classify whether a movie is a great movie on existing data and use it to classify new data

- Select Supervised Learning (classification) method

  - Supervision: The training data are accompanied by labels indicating the class of the observations

  - New data is classified based on the training set

- Select Decision Tree

  - It is one of the simplest and most successful forms of classification

  - Models are easier to understand

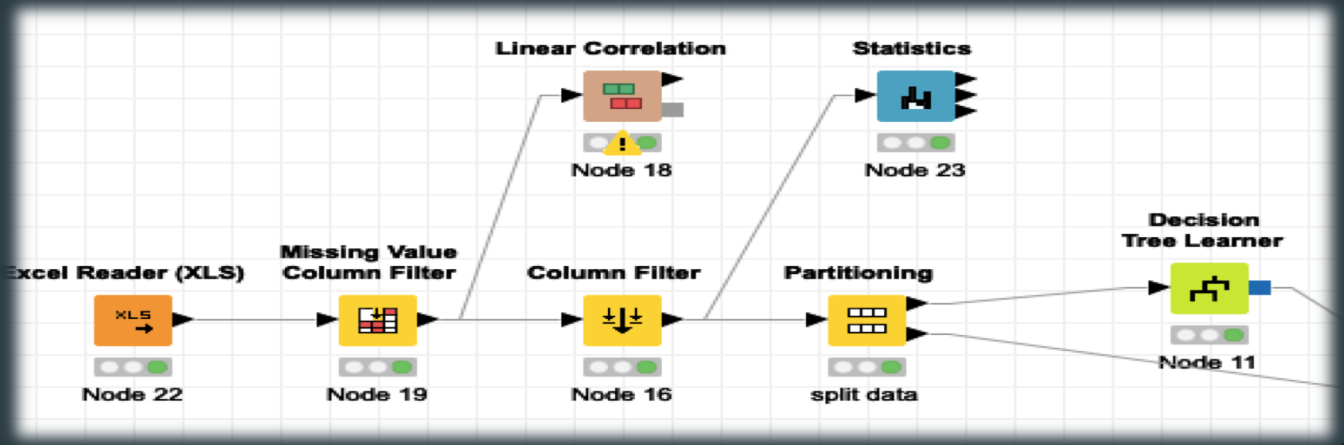
  - Rules are well visualized

- Split it into a training set and a test set. Using Partitioning node to randomly split the data into two sets:70% for training and 30% for testing

MODELING-Model Training

- Set setting parameters: class column is if_great, Min number records per node is 20, leave other parameters as default

- Adjust parameters which negatively effect the accuracy: here I build the tree there are some parameters that can contribute to the size and growth of tree such as country/language so I made adjustment



MODELING-Final Model

- This final Decision Tree model predicts the circumstances that could lead to a great movie from the piles of factors, such as the fame of the actors, the genres, communications on social medias
- How much the movie advertising, like the numbers of Facebook likes
- what the genre is, like action, adventure, comedy, fantasy, biography
- How famous the actor is, like how many Facebook likes of the actor

MODELING-Results Explanation

- How famous the actor is:

Celebrities with large population Facebook likes

- How much the movie advertising: movie Facebook likes

MODELING-Results Explanation

- what the genre are:

action

adventure

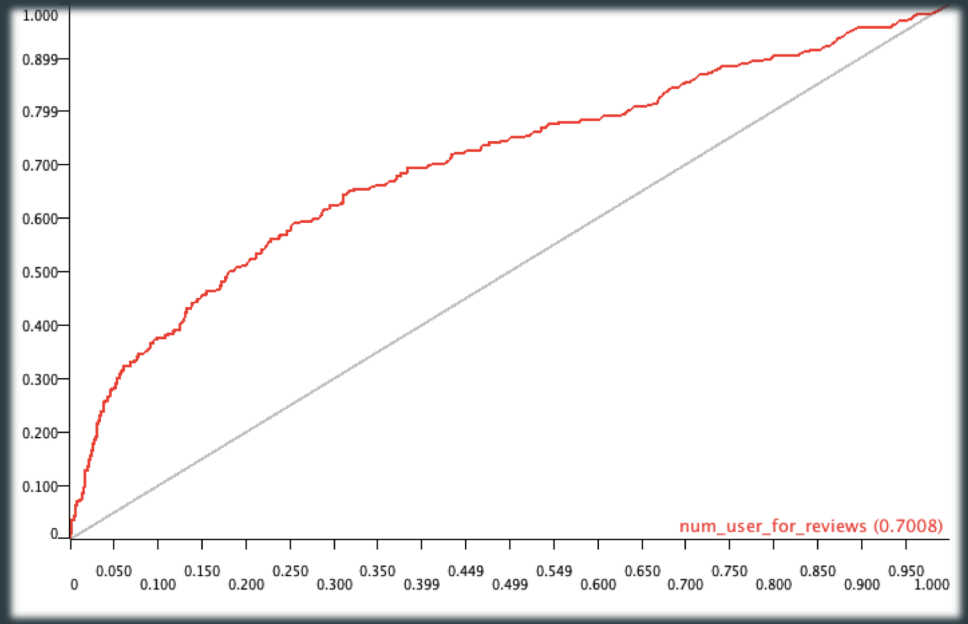
comedy

fantasy

biography

MODELING-Evaluation

- Receiver Operating Characteristic curve (ROC curve) a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
- Using ROC Curve to measure whether the model is effective: Overall 0.7,which represents the model is fairly capable of distinguishing between classes.
- The Result determines the objectives is achieved.



DEPLOYMENT

- The model is ready to predict future data. For future analysis to better assessment for predicting whether a movie will achieved fame and revenues before it is released in cinema

`

- In business, film makers can deployment this model to help them make wise decision based on the model

CONCLUSIONS

- This project proposes a decision tree algorithm to build a predictive model that how a great a movie is, using KNIME tool.
- It is important for Film Studios and Film Producers to create “successful” movies, which may bring sufficient revenues. So to find what kind of movie are audiences’ favorite is helpful.
- The result obtained that it is useful and effectively define what the rule of a great movie.
- It may have limitations because here the collected data only contains around 30 variables of factors and some movies are old fashions and may not suitable to modern film industry business.
- To optimize the model in the future, I may search for up to date data with as many variables as possible to increase the rules.