Predicting Oscar Winners With Machine Learning

Sarah, Ruth, Kiana, Nicole,
Gabriela and Kieran

Table of Contents

Who Is Our Targeted Client:

- ★ The Oscars are a highly visible event, gaining mass attention from the media, individuals in the arts and entertainment industries and movie-goers alike.
- ★ Research shows that more than 40 million viewers tune in to watch the Oscars every year.
- ★ The film and entertainment industries are highly profitable and having predictive models such as this will continue to support and promote success in the arts (Gold et al., 2013).
- ★ The Targeted audience for this project could be filmmakers, highly profitable production companies and streaming services or independent movie producers looking to make a return on their investment.

Data Samples:

```python
# import dependencies
import numpy as np
import pandas as pd
from imblearn.over_sampling import RandomOverSampler
from pathlib import Path
from sklearn.metrics import balanced_accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
```

```python
# Read the CSV file from the Resources folder into a Pandas DataFrame
oscars_df = pd.read_csv("Resources/oscars_df.csv")
# Review the DataFrame
pd.set_option('display.max_columns', None)
oscars_df
```

Preview Code Blame 572 lines (572 loc) · 676 KB Raw

```
,Film,Oscar Year,Film Studio/Producer(s),Award,Year of Release,Movie Time,Movie Genre,IMDB Ratir
0,Wings,1927/28,Famous Players-Lasky,Winner,1927,144,"Drama,Romance,War",7.5,"12,221","With Worl
1,7th Heaven,1927/28,Fox,Nominee,1927,110,"Drama,Romance",7.7,"3,439",,,,,,,,,,,,,,,,,,,19ed329
2,The Racket,1927/28,The Caddo Company,Nominee,1928,84,"Crime,Drama,Film-Noir",6.7,"1,257",,,,,,
3,The Broadway Melody,1928/29,Metro-Goldwyn-Mayer,Winner,1929,100,"Drama,Musical,Romance",5.7,"6
4,Alibi,1928/29,Feature Productions,Nominee,1929,91,"Action,Crime,Romance",5.8,765,,,,,,,,,,,,,,
5,Hollywood Revue,1928/29,Metro-Goldwyn-Mayer,Nominee,1929,130,"Comedy,Music",5.7,"2,004",,,,,,,
6,In Old Arizona,1928/29,Fox,Nominee,1928,95,Western,5.6,"1,019","In this early Western, notoric
7,The Patriot,1928/29,Paramount Famous Lasky,Nominee,1928,113,"Drama,History,Thriller",7.4,18,,,
```

Preview Code Blame 572 lines (572 loc) · 344 KB Raw

Search this file

Film	Film Studio/Producer(s)	Award	Year of
Wings	Famous Players-Lasky	Winner	1927
Battleground	Metro-Goldwyn-Mayer	Nominee	1949
7th Heaven	Fox	Nominee	1927
The Racket	The Caddo Company	Nominee	1928
Alibi	Feature Productions	Nominee	1929
Hollywood Revue	Metro-Goldwyn-Mayer	Nominee	1929
The Patriot	Paramount Famous Lasky	Nominee	1928
All Quiet on the Western Front	Universal	Winner	1930
Disraeli	Warner Bros.	Nominee	1929
The Divorcee	Metro-Goldwyn-Mayer	Nominee	1930
The Love Parade	Paramount Famous Lasky	Nominee	1929
East Lynne	Fox	Nominee	1931
The Front Page	The Caddo Company	Nominee	1931
Skippy	Paramount Publix	Nominee	1931
Trader Horn	Metro-Goldwyn-Mayer	Nominee	1931
Arrowsmith	Samuel Goldwyn Productions	Nominee	1931

What We Created:

- First we identified our dataset , cleaned, and then preprocessed our target and features variables using python.
- Our target variable is the 'Award' column (a categorical variable indicating either 'Winner' or 'Nominee'), which we one hot encoded with get dummies.
- We dropped the nominee column and were left with a column called 'Winner' indicating either a positive class 1= Winner, or negative class 0= Nominee.
- We then used a number of functions including regex to split and clean our feature variables including IMDB rating, movie time, production studio, genre, and director nominations

Machine Learning Model

- Model used
  - Logistic Regression Model
- Features/Target
  - Target or y is the winner column
  - Features or X is IMDB rating, movie time, production studio, genre, and director nominations columns
- Split the data
  - Used train-test-split to split the data.
- Fit model and make predictions
  - Used Logistic Regression classifier
- Evaluate Model's performance
  - Calculate the accuracy score
  - Generate a confusion matrix
  - Print the classification report
- Optimize the model
  - Random Over Sampler
  - Random Forest Classifier

Categories We Analyzed:

Top Ten Directors

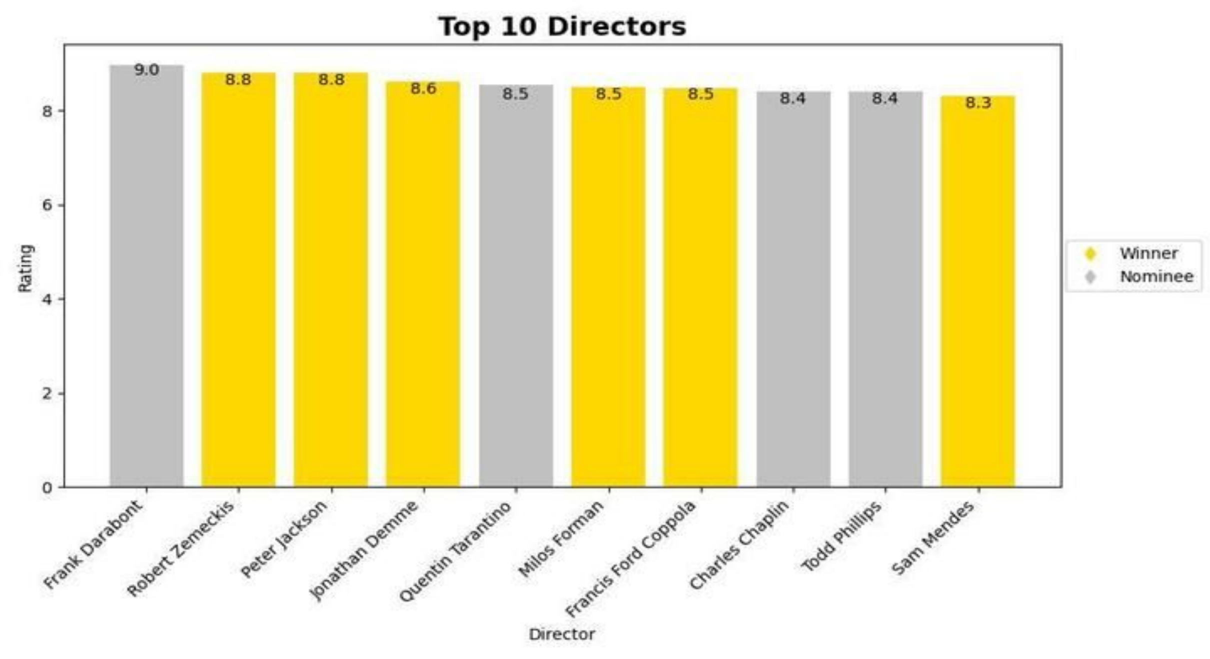
Top Ten Movies

Movie Length

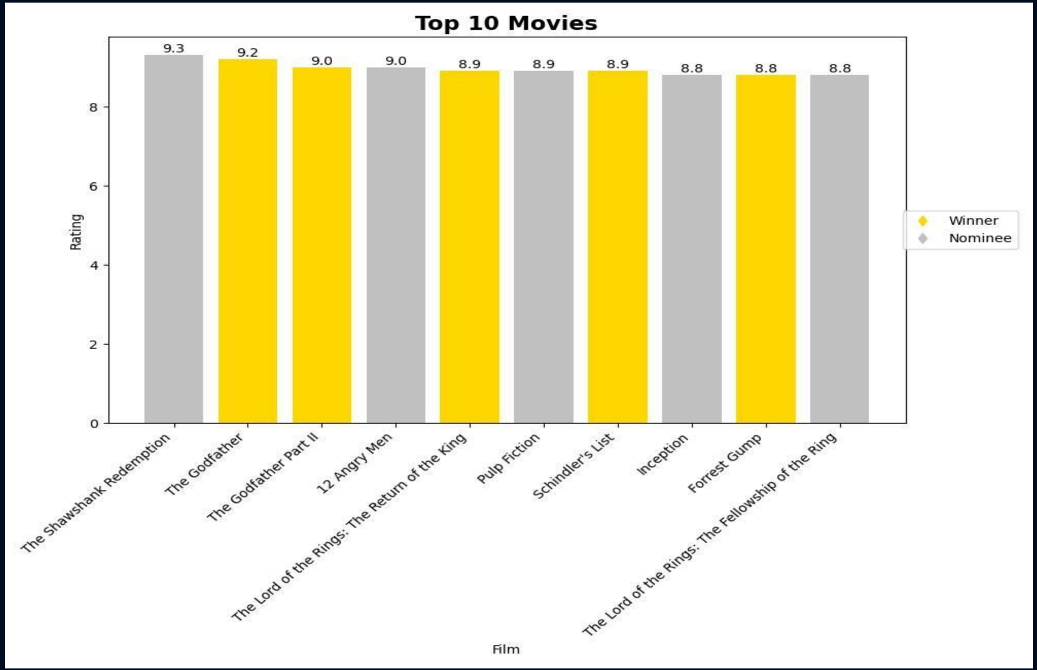
Production Companies

Genres

Top Ten Directors:

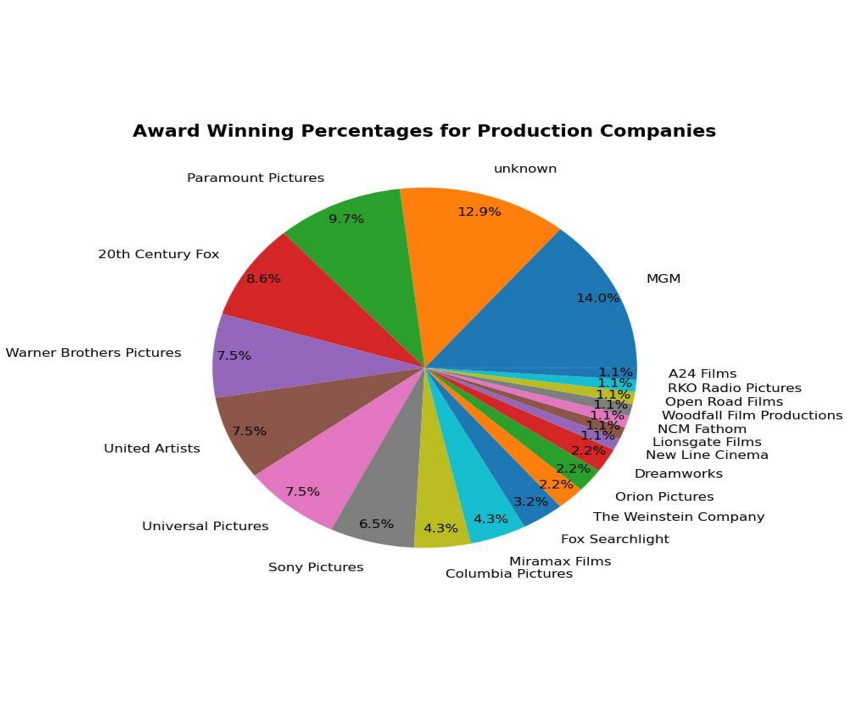


Top Ten Movies:

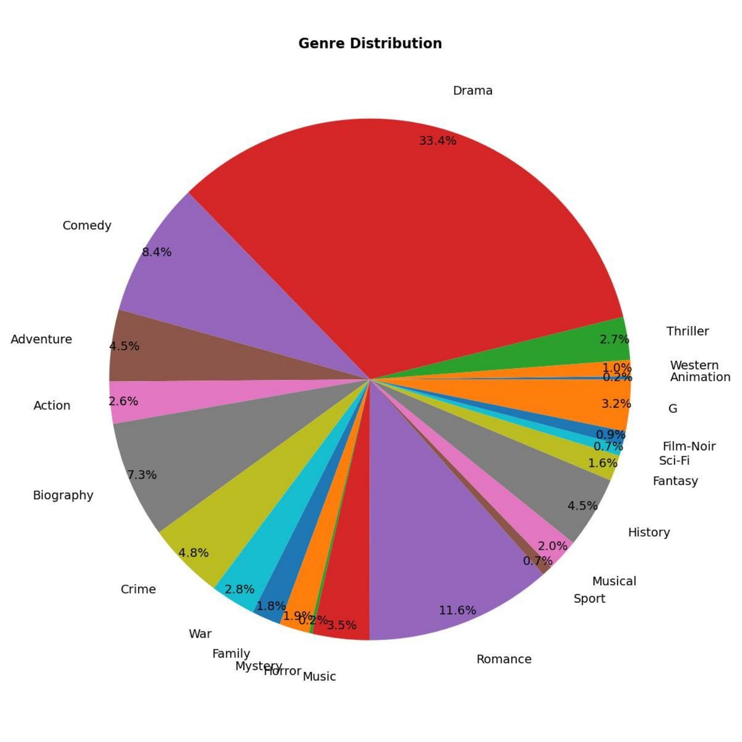


Movie Length:

Production Company:



Genre:


Genre Distribution
Drama
33.4%
Comedy
8.4%
Adventure
4.5%
Action
2.6%
Biography
7.3%
Crime
4.8%
War
2.8%
Family
1.8%
Mystery
1.9%
Horror
0.2%
Music
3.5%
Romance
11.6%
Sport
0.7%
Musical
2.0%
History
4.5%
Fantasy
1.6%
Sci-Fi
0.7%
Film-Noir
0.9%
G
3.2%
Animation
0.2%
Western
1.0%
Thriller
2.7%

Model Optimization

First attempt: LogisticRegression

```
# Print the classification report for the model
model_classification = classification_report(y_test, test_predictions)

print(model_classification)
```

```
              precision    recall  f1-score   support

         0.0       0.86      0.98      0.92       122
         1.0       0.33      0.05      0.08        21

    accuracy                           0.85       143
   macro avg       0.60      0.52      0.50       143
weighted avg       0.78      0.85      0.79       143
```

Second attempt: RandomOverSampler

```
# Print the classification report for the model
model_classification = classification_report(y_test, test_predictions_resamp)

print(model_classification)
```

```
              precision    recall  f1-score   support

         0.0       0.84      0.71      0.77       122
         1.0       0.12      0.24      0.16        21

    accuracy                           0.64       143
   macro avg       0.48      0.48      0.47       143
weighted avg       0.74      0.64      0.68       143
```

Final Model and Findings

Third and Final Attempt: RandomForestClassifer

[84] 0s

```
# print classification report
print(classification_report(y_test, test_pred_rfc))
```

	precision	recall	f1-score	support
0.0	0.93	1.00	0.96	617
1.0	1.00	0.49	0.66	97
accuracy			0.93	714
macro avg	0.96	0.75	0.81	714
weighted avg	0.94	0.93	0.92	714

Challenges:

- Combining the production companies with the same names.
  - There were multiple production companies who had the same name but slightly different spelling. Using regex we were able to change the names of the production companies to be concise and get rid of any duplications.
- Genres column
  - The genres column had to be changed from a string to an array. We were able to separate all of the genres and combine them so there were no duplications. We also changed some spelling errors in the column to ensure no duplicates were present.

Next Steps:

- ★ We could progress this project further by making it more equitable and having the data analyzed independent films in relationship to independent film awards (Example: Sundance film festival)
- ★ We could also include additional features in our model such as actors, the musical score, budget of the film, or prop-bets by exploring different datasets and websites
- ★ Additionally, we could deploy our machine learning model using a flask app to increase front-facing usability

Citations

Gold, M., McClarren, R., & Gaughan, C. (2013). The lessons Oscar taught us: data science and media & entertainment. Big Data, 1(2), 105-109.

https://www.kaggle.com/datasets/martinmraz07/oscar-movies