Academy Award Prediction

CZ1016 Mini-Project

Chang Heen Sunn, Ooi Wei Chern, Saw William Joseph

Project Summary

Motivation

To discover the underlying features that determine an Oscar winning movie and predict winners for the Best Picture Award.

Problem Statements

- What are the key features of a film to win the Academy Award?
- Which prediction model perform best with our dataset?

Type of Problem

Classification type. To determine whether the movie is a winner.

Table of Contents

01

The Opening

Why Oscars?

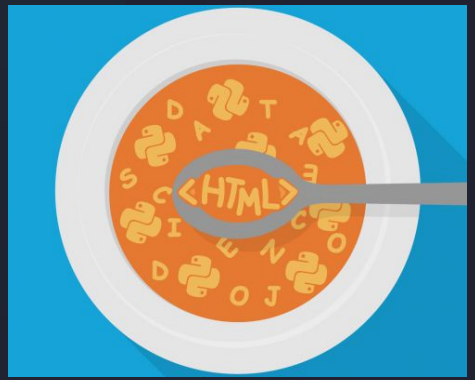
The Academy Awards, Oscars

The Academy Award is the oldest and biggest entertainment awards ceremony in the world.

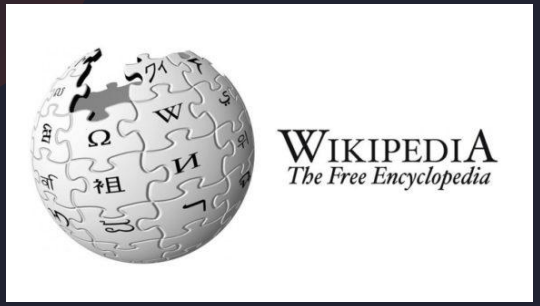
Movie makers and fans, are equally thrilled with anticipation to see their favorite movies lift the biggest award at the end of the night:
The Best Picture Award.

02

Data Collection



Through Web Scraping
Using BeautifulSoup
... ...
& manual filling



IMDb

Box Office Mojo by IMDbPro

Rotten Tomatoes®

metacritic

THE NUMBERS®

Where Data and the Movie Business Meet

News | Box Office | Home Video | Movies | People | Oscars | Research & Data

1. **Joker** (2019)

15 | 122 min | Crime, Drama, Thriller

8.5 Rate this 59 Metascore

In Gotham City, mentally troubled comedian Arthur Fleck is disregarded and mistreated by society. He then embarks on a downward spiral of revolution and bloody crime. This path brings him face-to-face with his alter-egо: the Joker.

Director: Todd Phillips | Stars: Joaquin Phoenix, Robert De Niro, Zazie Beetz, Frances Conroy

Votes: 752,509 | Gross: $335.45M

IMDb

Movie name

Genre

IMDb rating

```
<h3 class="lister-item-header">
        <span class="lister-item-index unbold text-primary">1.</span>

    <a href="/title/tt7286456/?ref_=adv_li_tt"
>Joker</a>
    <span class="lister-item-year text-muted unbold">(2019)</span>
</h3>
    <p class="text-muted ">
            <span class="certificate">15</span>
                <span class="ghost">|</span>
               <span class="runtime">122 min</span>
                <span class="ghost">|</span>
            <span class="genre">
Crime, Drama, Thriller            </span>
    </p>
    <div class="ratings-bar">
    <div class="inline-block ratings-imdb-rating" name="ir" data-value="8.5">
        <span class="global-sprite rating-star imdb-rating"></span>
        <strong>8.5</strong>
    </div>
            <div class="inline-block ratings-user-rating">
                <span class="userRatingValue" id="urv_tt7286456" data-tconst="t
                    <span class="global-sprite rating-star no-rating"></span>
```

Data Scraped

IMDb / Metacritic

General info
User rating
Number of votes
*Metascore

Rotten Tomatoes

Tomatometer
Number of reviews

The Numbers

Budget
Domestic (US) gross
International gross
Worldwide gross

Wikipedia

Film awards winner
Film awards nominations

Box Mojo

- For checking

We scraped through,

1999-2019

2100 x 61 columns

2,100

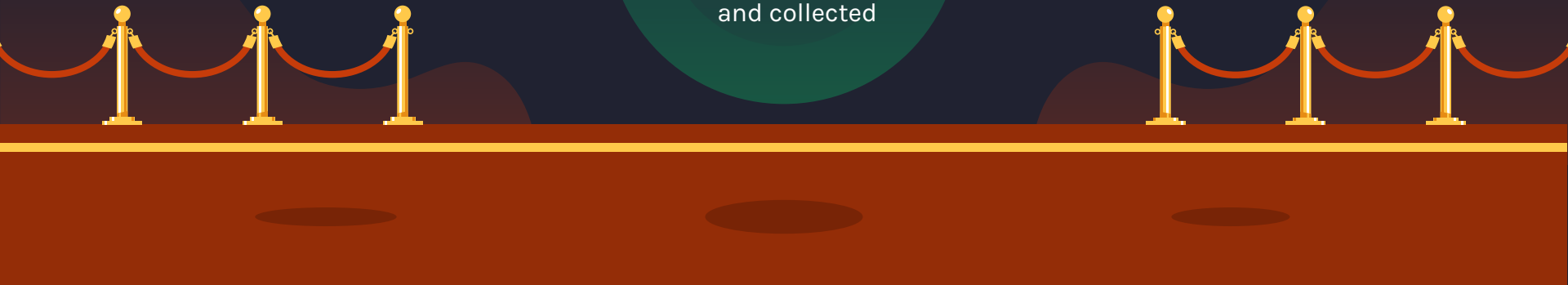
movies scraped from online sources

53

features formed and collected

128,100

data collected in total



Manual Filling, Cleaning & Checking

1. Movies naming format different

- Stemming & Lemmatization
- Movie name consists of different languages, weird symbols

Examples:
(2001) Sen to Chihiro no kamikakushi -Spirited Away
(2002) AstÃ©rix & ObÃ©lix: Mission ClÃ©opÃ¢tre - Asterix & Obelix: Mission Cleopatre

Manual Filling, Cleaning & Checking

2019 New York Film Critics Circle Awards

From Wikipedia, the free encyclopedia

The **84th New York Film Critics Circle Awards**, honoring the best in film for 2019, were announced on December 4, 2019.[1]

Winners

- **Best Film**
  - *The Irishman*
- **Best Director**
  - Josh and Benny Safdie – *Uncut Gems*
- **Best Actor**
  - Antonio Banderas – *Pain and Glory*
- **Best Actress**
  - Lupita Nyong'o – *Us*

2. More Efficient

- Wikipedia table
- Need to check manually nevertheless

03

Data Exploration & Analysis

Category of Features Collected

01

Ratings

IMDb, Tomatometer & Metacritic

02

Movie Elements

Genre, Runtime

03

Commercial

Box office & Budget

04

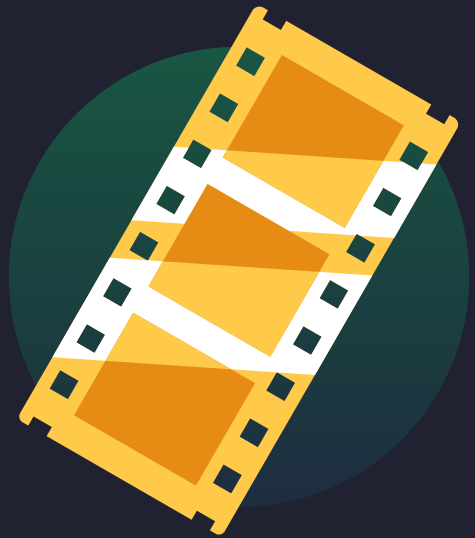
Film Awards

International Awards (BAFTA, DGA, PGA, etc.)

What Features Show A Good Movie?

Critical Ratings

IMDb User Rating, Tomatometer and Metascore all good indicators to tell which movies are Oscar-worthy.

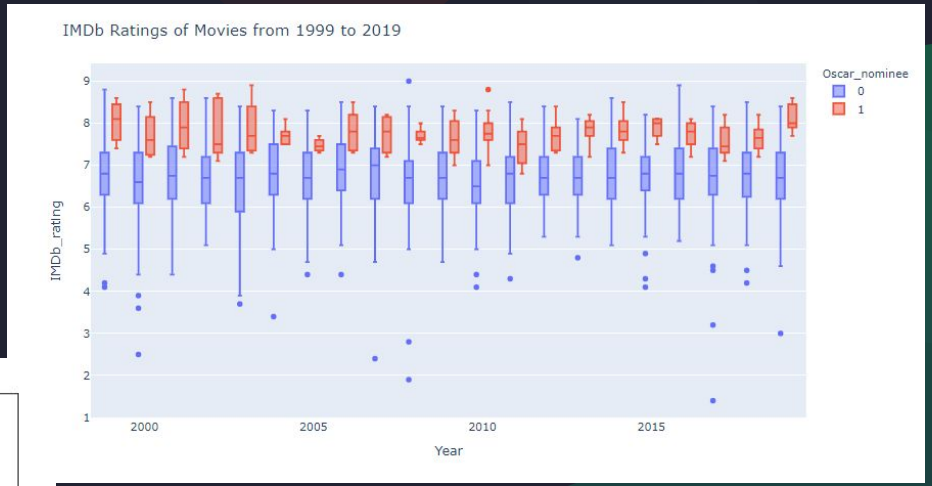


Awards Correlation

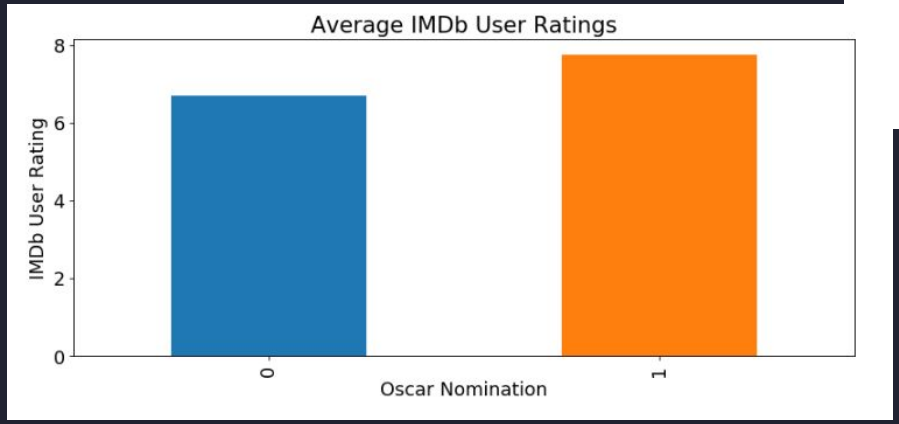
Awards ceremonies have one thing in common: Good eye for appraising movies.

IMDb Rating

Boxplot

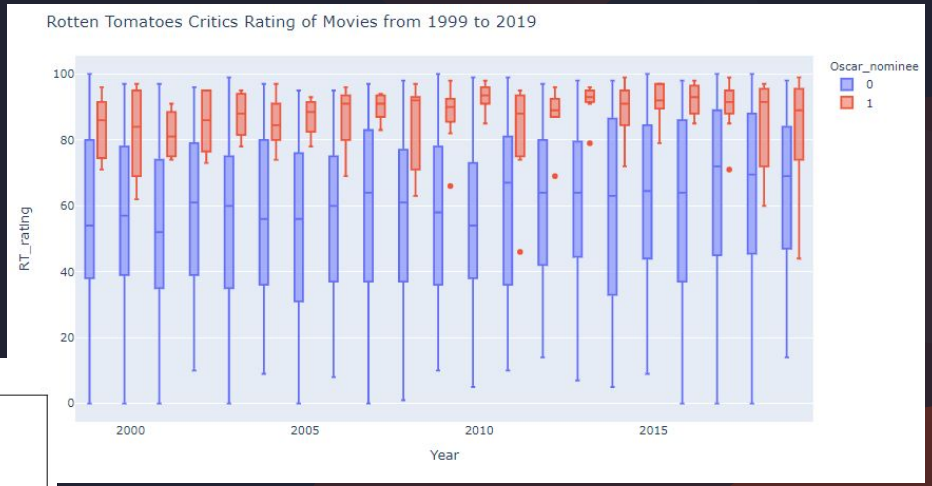


Barchart of Average

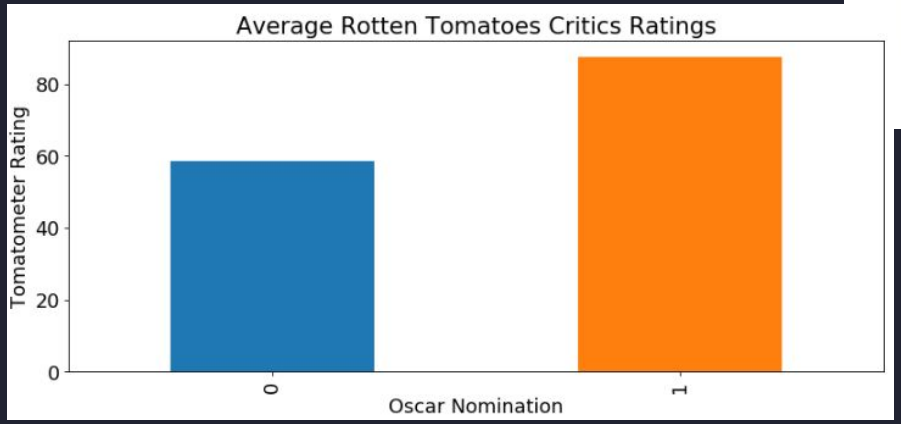


Tomatometer

Boxplot

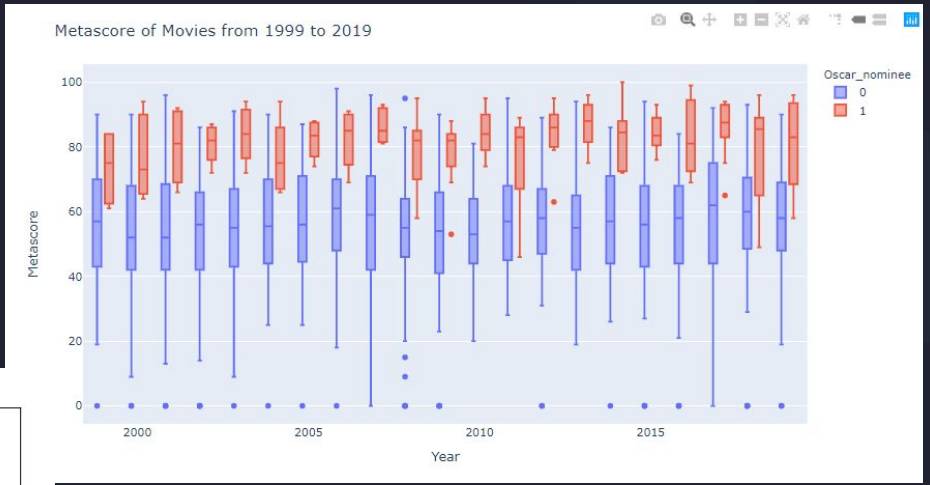


Barchart of Average

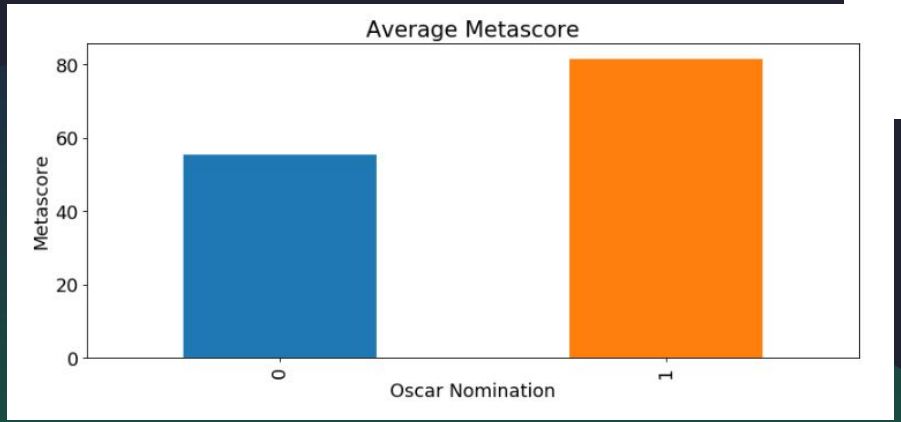


Metascore

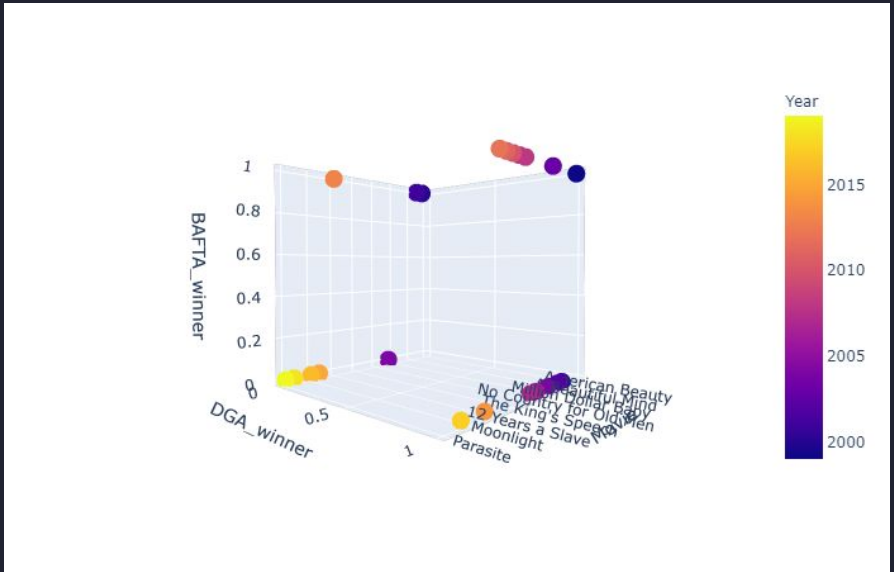
Boxplot



Barchart of Average



Correlation Between Awards



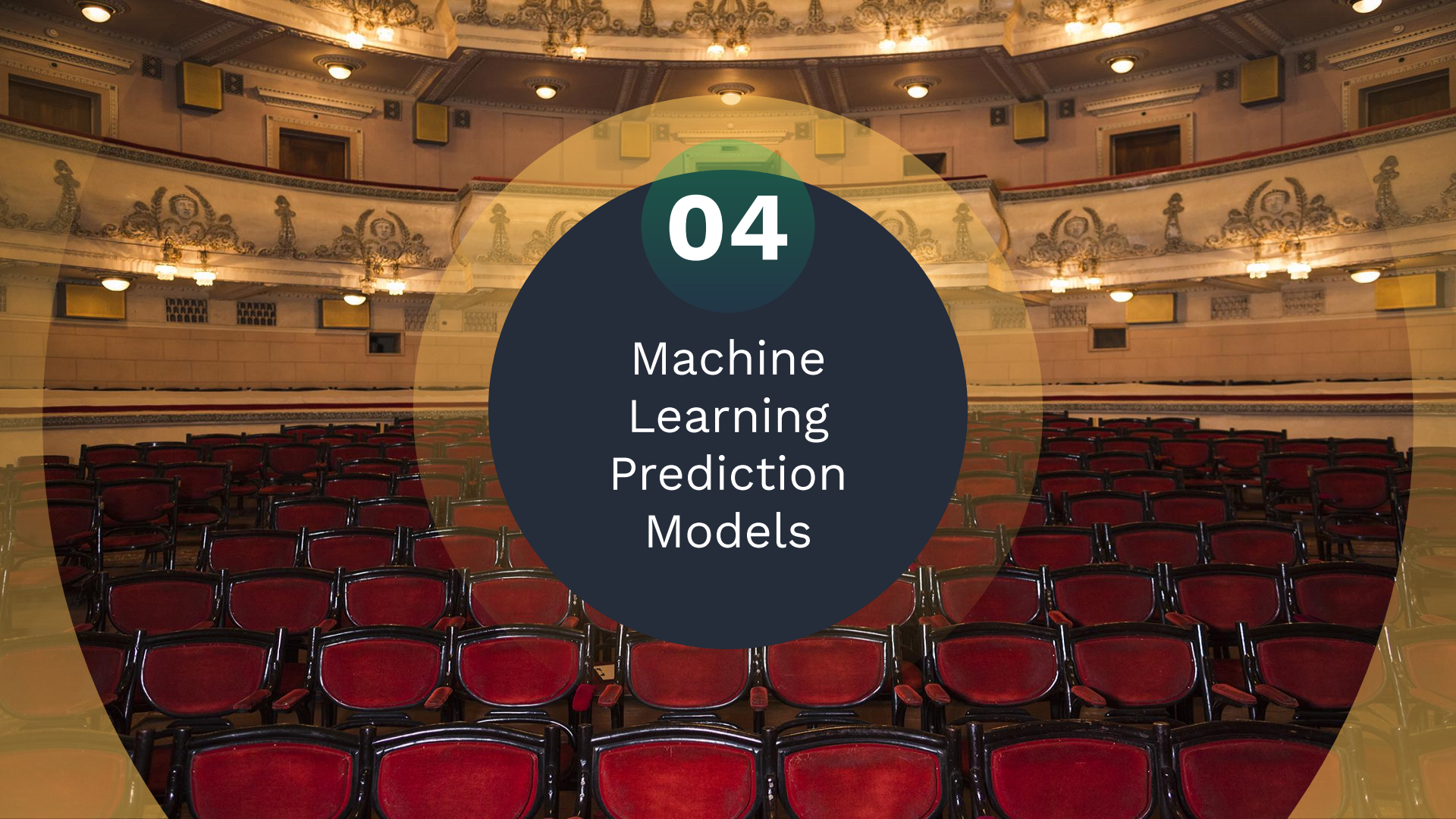
3D plot to correlate DGA, BAFTA with Oscars

DGA shows better correlation with Oscars than BAFTA.

Correlation Between Awards

	Oscar Win	Oscar Nomination
Director's Guild Award (DGA)	+0.65	+0.76
Producer Guild Award (PGA)	+0.63	+0.77
Golden Globes Movie Award	+0.37	+0.63
Critics Choice Movie Award	+0.66	+0.76
British Academy of Film and Television Arts (BAFTA)	+0.46	+0.67

Awards and nomination with correlation above +0.60 are taken as feature for machine learning



04

Machine Learning Prediction Models

The Classification Accuracy is a Joke

- 2100 movies, 21 winners (2100:21 = 100:1)
- Classic problem with imbalanced data
- Easily 99% accuracy rate
- True Positive rate, a.k.a recall

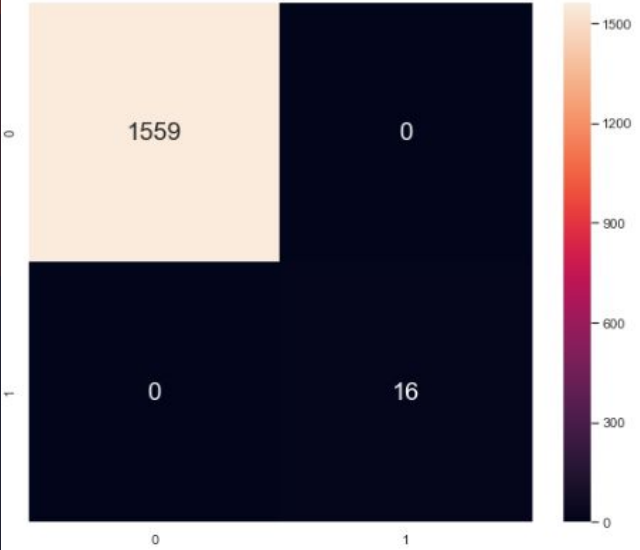
The Models

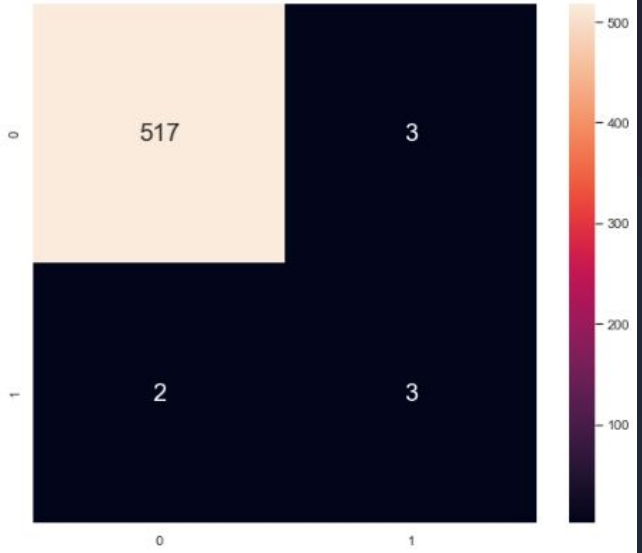
1. Multinomial Naive Bayes
2. Gaussian Naive Bayes
3. Bernoulli Naive Bayes
4. Decision Tree
5. Support Vector Classifier
6. Random Forest
7. Logistic Regression
8. AdaBoost with MultinomialNB as base estimator
9. AdaBoost with Decision Tree as base estimator
10. AdaBoost with SVC as base estimator

Our Top 3 Models

3. Decision Tree Classifier

```
Classification accuracy for train set:  1.0
Classification accuracy for test set:  0.99
```



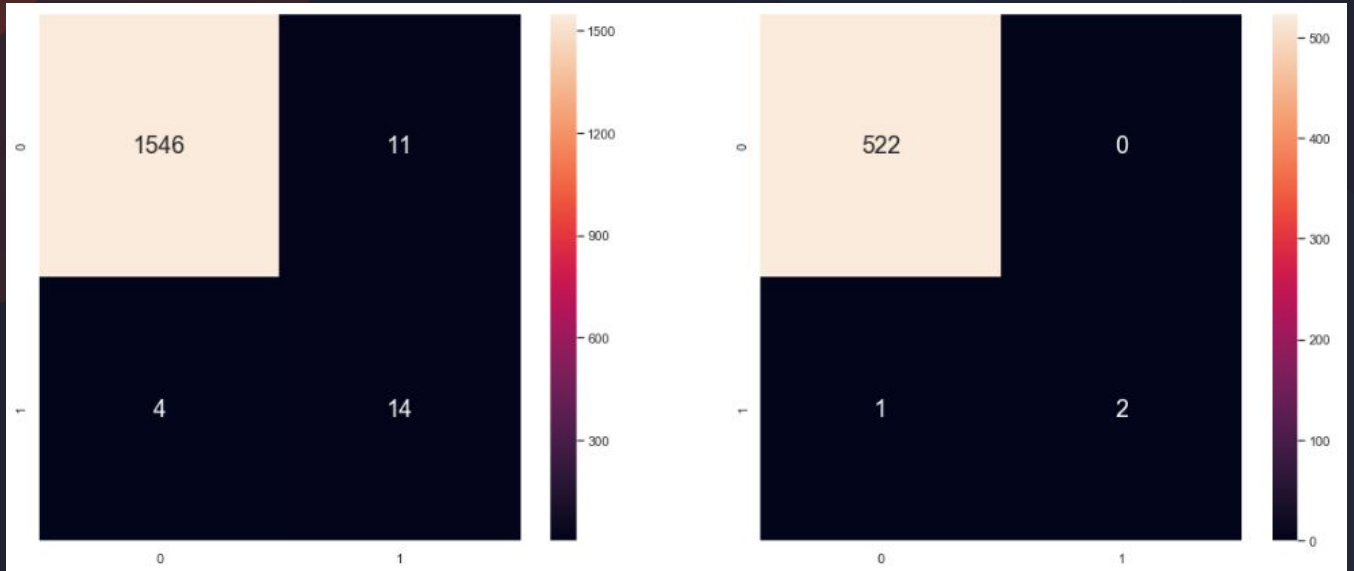



```
For train set:
True negative rate:  1.0
False negative rate:  0.0
True positive rate:  1.0
False positive rate:  0.0
```

```
For test set:
True negative rate:  0.99
False negative rate:  0.4
True positive rate:  0.6
False positive rate:  0.01
```

2. AdaBoost(base_estimator = SVC)

```
Classification accuracy for train set:  0.99
Classification accuracy for test set:  0.998
```

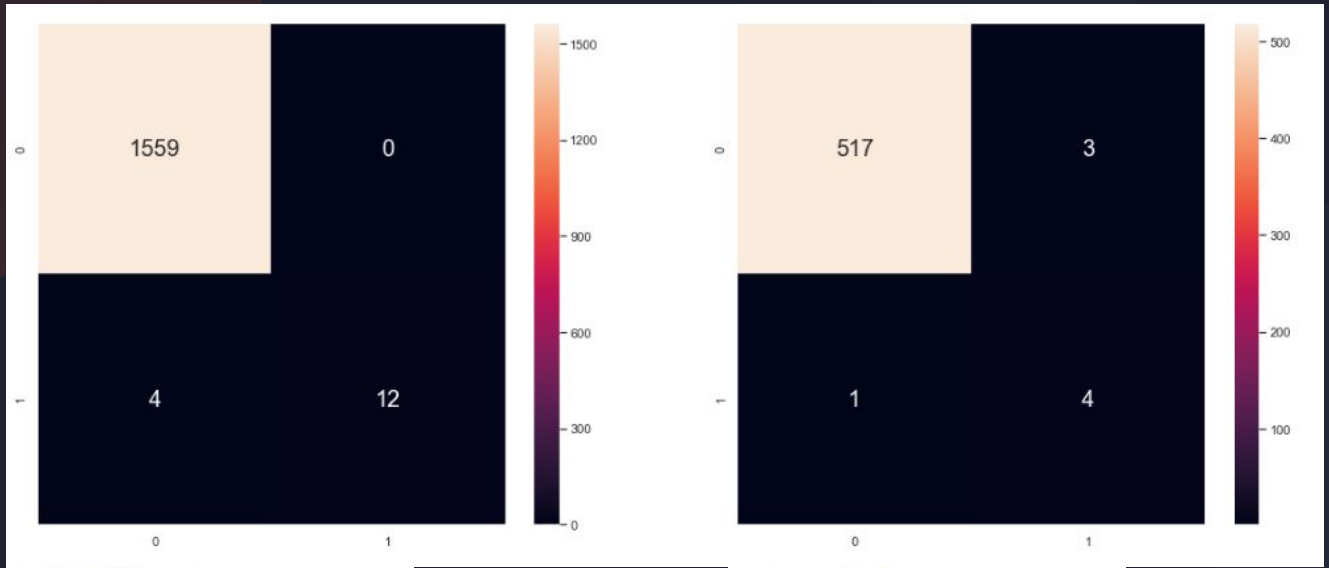



```
For train set:
True negative rate:  0.99
False negative rate:  0.22
True positive rate:  0.78
False positive rate:  0.01
```

```
For test set:
True negative rate:  1.0
False negative rate:  0.33
True positive rate:  0.67
False positive rate:  0.0
```

1. Random Forest

```
Classification accuracy for train set:  0.997
Classification accuracy for test set:  0.992
```




```
For train set:
True negative rate:  1.0
False negative rate:  0.25
True positive rate:  0.75
False positive rate:  0.0
```

```
For test set:
True negative rate:  0.99
False negative rate:  0.2
True positive rate:  0.8
False positive rate:  0.01
```

Comparing Prediction Models

	Train error	Validation error	Time in seconds
Dummy	0.023	0.016	0.0110
Decision Tree	0.000	0.018	0.0120
Random Forest	0.000	0.012	0.1641
Extra Trees	0.000	0.012	0.1212
K-Nearest Neighbors	0.005	0.012	0.2549
Linear SVC	0.000	0.016	0.0309
Logistic Regression	0.000	0.016	0.0249
Bagging	0.002	0.016	0.0429
XGBoost	0.001	0.014	0.1891
AdaBoost	0.000	0.014	0.1616
Light Gradient Boosting Machine (LGBM)	0.000	0.012	0.1513
Gradient Boosting	0.000	0.014	0.3846
Gaussian Naive Bayes	0.024	0.034	0.0090
Bernoulli Naive Bayes	0.028	0.044	0.0110
Multi Layer Perceptron (Neural Network)	0.000	0.016	0.5947
DecisionTreeRegressor	0.000	1.616	0.0110
Random Forest Regressor	0.047	1.093	0.3280
K-Nearest Neighbors Regressor	0.268	1.131	0.2140

05

Predict Winners

Extremely Randomized Trees
vs.
Random Forest

Top 2 Features Categories

Critical Ratings

IMDb ratings & votes

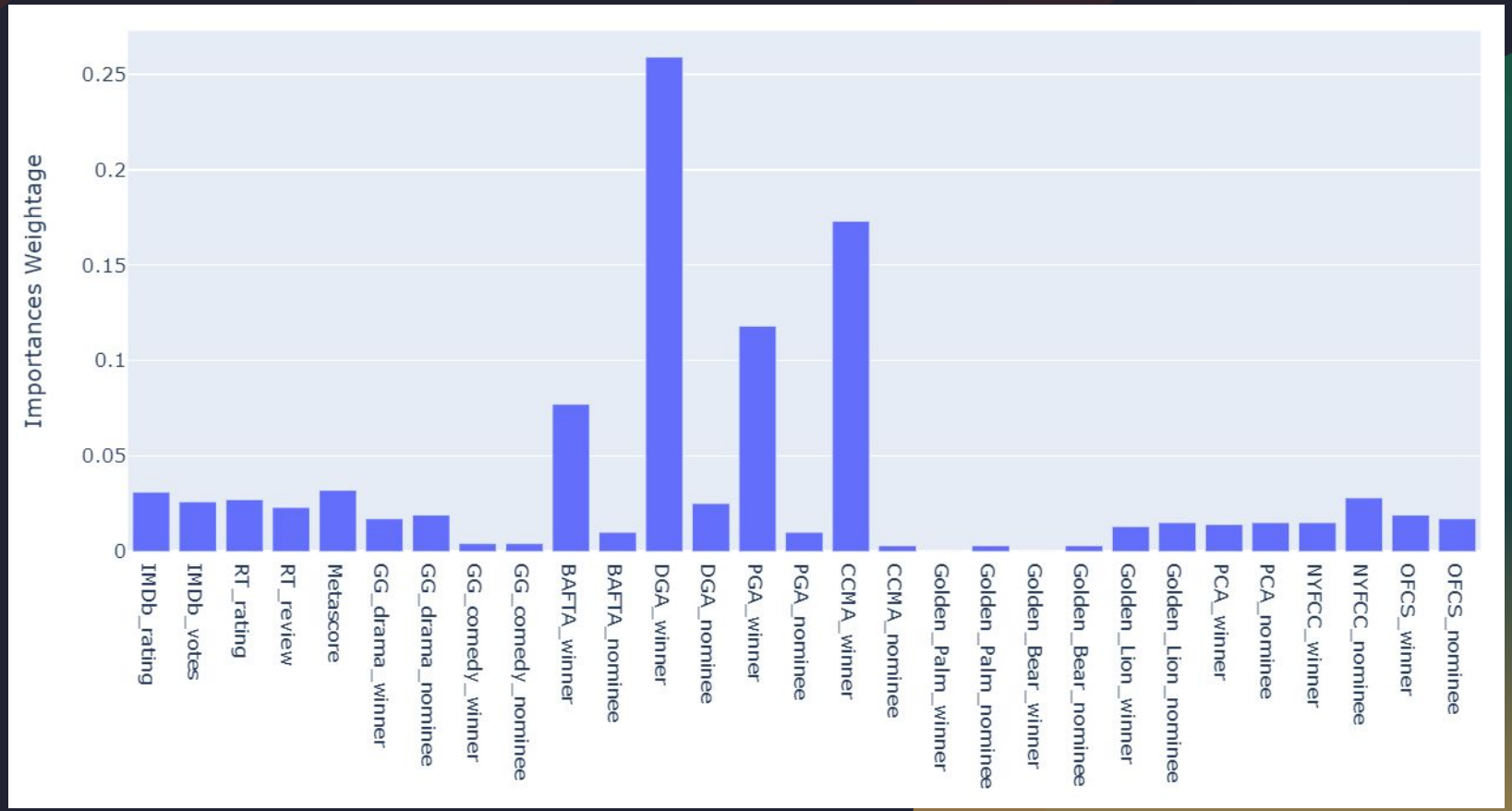
Rotten Tomatoes tomatometer & reviews

Metascore

Film Awards

BAFTA
DGA
PGA
CCMA
GG_drama
GG_comedy
Golden Palm
Golden Bear
Golden Lion
PCA
NYFCC
OFCS

Importance of Features (Movie Critics + Film Awards)



Top 3 Important Individual Features

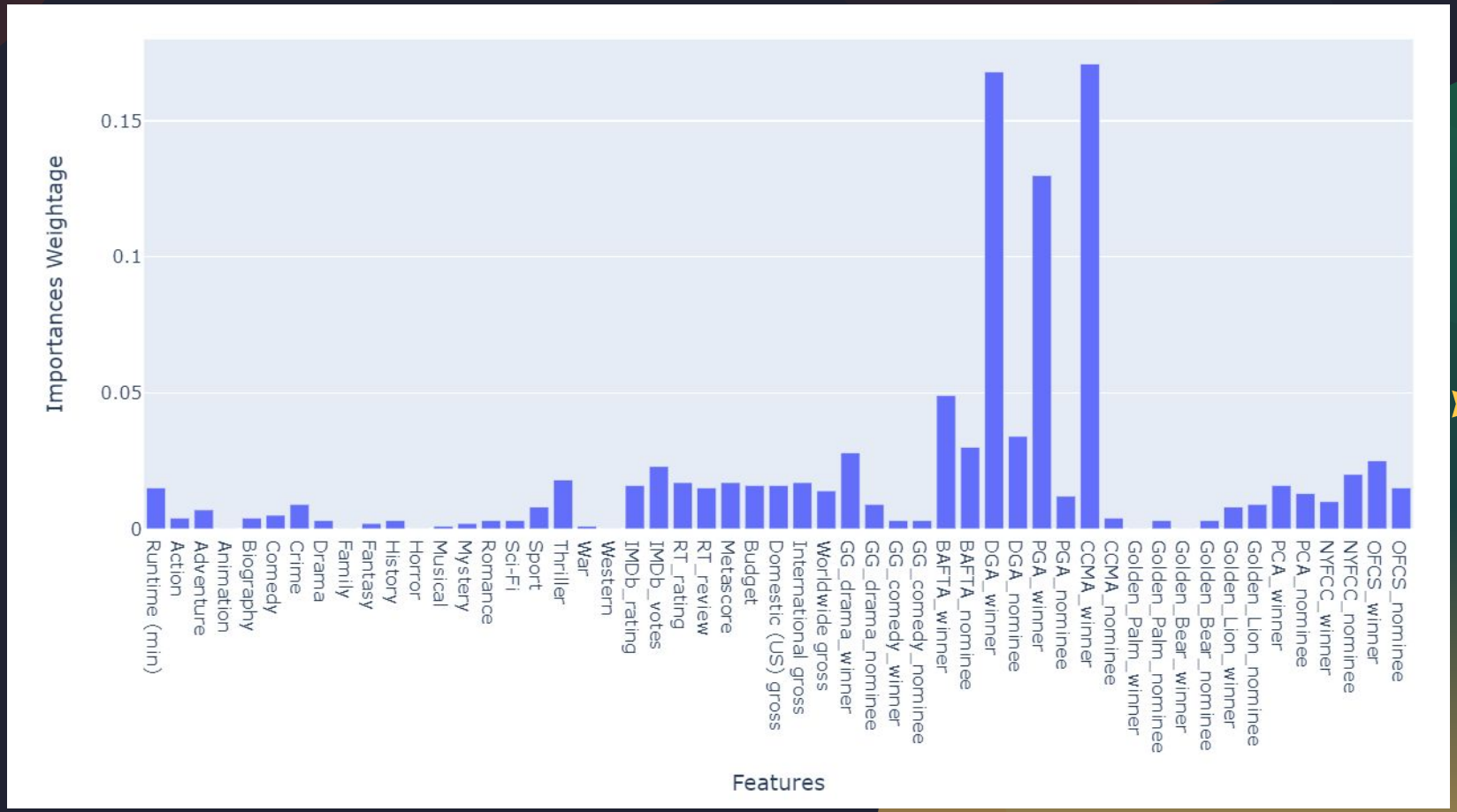
If we only consider the 3 most important features in the algorithm...

	Importances Weightage
DGA_winner	0.405
PGA_winner	0.247
CCMA_winner	0.348

Score = 0.9975

	Year	Movie	Oscar_nominee	Oscar_winner	Predicted Win Rate
408	2019	1917	1	0	0.915
400	2019	Joker	1	0	0.001
463	2019	Gemini Man	0	0	0.001
473	2019	The Lego Movie 2: The Second Part	0	0	0.001
472	2019	Happy Death Day 2U	0	0	0.001
471	2019	Velvet Buzzsaw	0	0	0.001
470	2019	Cold Pursuit	0	0	0.001
469	2019	Crawl	0	0	0.001
468	2019	Bombshell	0	0	0.001
467	2019	Fighting with My Family	0	0	0.001

Importance of Features (All 53 features)



Prediction for 2017

	Year	Movie	Oscar_nominee	Oscar_winner	Predicted Win Rate
213	2017	The Shape of Water	1	1	0.556
211	2017	Three Billboards Outside Ebbing, Missouri	1	0	0.232
207	2017	Get Out	1	0	0.101
230	2017	Call Me by Your Name	1	0	0.030
224	2017	Lady Bird	1	0	0.030
205	2017	Dunkirk	1	0	0.020
238	2017	Darkest Hour	1	0	0.010
236	2017	I, Tonya	0	0	0.010
246	2017	The Post	1	0	0.010
200	2017	Logan	0	0	0.000

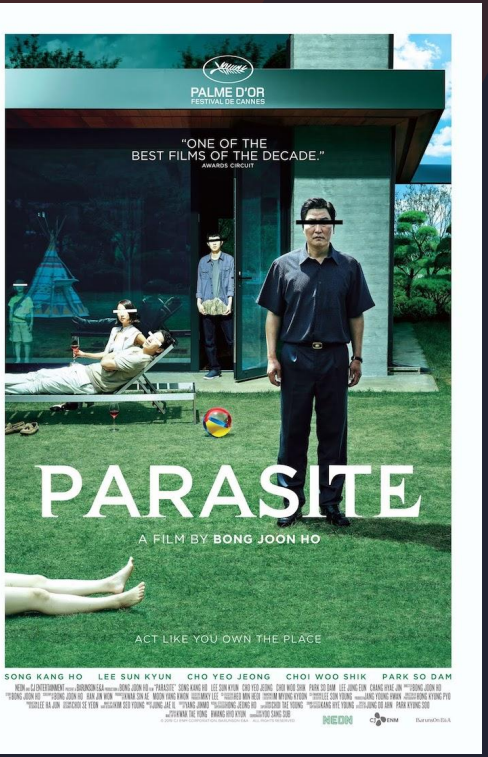
The predicted winner wins!

Prediction for 2018

	Year	Movie	Oscar_nominee	Oscar_winner	Predicted Win Rate
309	2018	Green Book	1	1	0.341
330	2018	Roma	1	0	0.341
325	2018	The Favourite	1	0	0.082
303	2018	Bohemian Rhapsody	1	0	0.071
321	2018	BlacKkKlansman	1	0	0.059
310	2018	A Star Is Born	1	0	0.035
326	2018	First Man	0	0	0.024
344	2018	Vice	1	0	0.012
350	2018	Creed II	0	0	0.012
361	2018	Mary Poppins Returns	0	0	0.012

The predicted winner wins!

Prediction for 2019



	Year	Movie	Oscar_nominee	Oscar_winner	Predicted Win Rate
408	2019	1917	1	0	0.444
402	2019	Once Upon a Time in Hollywood	1	0	0.299
404	2019	Parasite	1	1	0.145
418	2019	Jojo Rabbit	1	0	0.034
407	2019	The Irishman	1	0	0.026
400	2019	Joker	1	0	0.026
413	2019	Marriage Story	1	0	0.017
442	2019	Little Women	1	0	0.009
472	2019	Happy Death Day 2U	0	0	0.000
473	2019	The Lego Movie 2: The Second Part	0	0	0.000

The actual winner, Parasite came 3rd.

		Random Forest		Extra Tree	
	Actual Oscar Winner	Predicted placing	Predicted Oscar Winner	Predicted placing	Predicted Oscar Winner
2015	Spotlight	2nd	The Revenant	2nd	The Revenant
2016	Moonlight	2nd	La La Land	3rd	La La Land
2017	The Shape of Water	1st	The Shape of Water	1st	The Shape of Water
2018	Green Book	2nd	Roma	1st	Green Book
2019	Parasite	3rd	1917	3rd	1917

Extra Exploration of Interest

Balloting Demo

Basic idea of Balloting

- simulating the Oscar academy with a voting body of 10,000 members
- purpose is to pick the top 3 most probable winners for the Best Picture Award.

	Votes
1917	3062
Parasite	1332
Once Upon a Time in Hollywood	800
The Irishman	583
Joker	336
Jojo Rabbit	324
Marriage Story	211
Little Women	179
Ford v Ferrari	173

Extracting the year in prediction

```
# Training Set - Excluding 2019
train = data.loc[((data['Year'] >= 1999) & (data['Year'] < 2019))]
test = data.loc[(data['Year']==2019)]

print('training set contains:', train.shape[0], 'movies')
print('Prediciting on:', test.shape[0], 'movies')
```

```
training set contains: 2000 movies
Prediciting on: 100 movies
```

Simulating a voter

```
[8]: def simulate_a_vote(model, train_df, to_predict_df, features):
        """
        This function creates, trains, and predicts with a DecisionTree to simulate an Academy voter.
        Each tree only sees a part of the data and gets Noise to decorrelate them from each other.
        The prediction is then ranked to create our ballot for Preferential Balloting
        """

        train = train_df.copy()
        test = to_predict_df.copy()

        # A noise column, randomly generated each time represents a voter's bias
        train.loc[:,'Noise'] = np.random.rand(train_df.shape[0])
        test.loc[:,'Noise'] = np.random.rand(to_predict_df.shape[0])

        # Looking at a random amount of awards shows (similar to bootstrapping)
        # This reflects a voter's attention to the season
        # num_features is how many of the features they care about
        num_features = np.random.choice(int(len(features)/1.7))
        voter_features = list(np.random.choice(features, num_features)) + ['Noise']

        x = np.array(train[voter_features])
        y = np.array(train['Oscar_winner'])

        model.fit(x,y)

        # ProbA of the voter will represent the ranked votes
        ballot_clean = model.predict_proba(np.array(test[voter_features]))[:,1]
        # Add small random values to break up ties
        ballot = ballot_clean + np.random.rand(len(ballot_clean))/10000

        # Use np.argsort() to rank the order of the probA
        # The Academy uses ranked votes calculate winner
        temp = ballot.argsort()
        ranks = np.empty_like(temp)
        ranks[temp] = np.arange(len(ballot))
        ranks = np.abs(ranks - len(ballot))
        return ranks
```

```python
def remove_least(voting_body, list_of_nominees):
    """
    A function used for the elimination step of Preferential Balloting
    This function determines which film has the least #1 rankings and removes it
    """
    # List of nominees must be in the same order as the vote index
    firsts = np.where(voting_body==1,1,0)
    tally = np.sum(firsts, axis = 0)
    least_votes_index = np.argmin(tally)

    # Removes the least voted entry (from # 1 to 0)
    voting_body = np.delete(voting_body, least_votes_index, axis = 1)
    list_of_nominees.remove(list_of_nominees[least_votes_index])
    return voting_body, list_of_nominees
```

> To remove the last movie which got the least vote in the list

```python
def re_rank_ballots(voting_body):
    """
    Another function used for the elimination step of Preferential Balloting
    Takes a voting body (numpy array)
    Makes sure each row goes from 1 to shape[1]
    """
    re_ranked = np.zeros(voting_body.shape)
    for i in range(voting_body.shape[0]):
        temp = voting_body[i,:].argsort()
        ranks = np.empty_like(temp)
        ranks[temp] = np.arange(len(voting_body[i,:]))
        re_ranked[i,:] = ranks + 1
    return re_ranked
```

> To rearrange the list of movies in descending number of votes

```python
def run_one_round_of_eliminations(voting_body, list_of_nominees):
    """
    A function which runs one elimination step of Preferential Balloting
    Takes in a Voting Body and List of Nominess and returns them,
    but the film with the least #1 votes has been removed
    """
    voting_body, list_of_nominees = remove_least(voting_body, list_of_nominees)
    voting_body = re_rank_ballots(voting_body)
    return voting_body, list_of_nominees
```

> Voting simulation

```
def run_preferential_voting(voting_body,list_of_nominees, show_steps = False):
    """
    Runs the process of Preferential Balloting on a voting_body(matrix)
    Terminates when one movie has greater than 50% of the total votes
    """
    top_pick_percent = tally_votes(voting_body,list_of_nominees).max()[0]/tally_votes(voting_body,list_of_nominees).sum()[0]

    while top_pick_percent < 0.5:
        voting_body,list_of_nominees = run_one_round_of_eliminations(voting_body, list_of_nominees)
        top_pick_percent = tally_votes(voting_body,list_of_nominees).max()[0]/tally_votes(voting_body,list_of_nominees).sum()[0]

        if show_steps:
            print(tally_votes(voting_body, list_of_nominees),'\n')

    return voting_body, list_of_nominees
```

Terminates once a movie has more than 50% of votes

Lets Simulate the Oscars! ٩(｡•ω•｡)و

```
print('training set contains:', train.shape[0], 'Movie')
print('Prediciting on:', test.shape[0], 'Movie')

# PicK the model we want for each random voter
voter_model = DecisionTreeClassifier(splitter='random',
                                     max_depth=3,
                                     min_samples_leaf=3,
                                     random_state = 92)

num_voters_academy = 10000
print(f'\nSimulating an Academy with {num_voters_academy} random voters.....')
academy_sim = simulate_voting_body(num_voters=num_voters_academy, model = voter_model, train_df = train, to_predict_df = test, fe

print('\nInitial Rankings:\n----------------------------------------')
print(tally_votes(academy_sim, list(test.Movie)),'\n')

print("Now we start eliminating films until one has more than 50% of the top picks:\n-----------------------------------------
final_ballot, final_films = run_preferential_voting(academy_sim, list(test.Movie), True)
```

Using 10,000 voters, we ran the balloting

2017

And the Oscar goes to...
🎉🏆 The Shape of Water 🏆🎉

2016

	Votes
La La Land	7375
Moonlight	812

2018

	Votes
Roma	5090
Green Book	2606

Conclusion

At the end of the night ...

Conclusion

To answer our initial problems,

Best Feature: Film Awards (susceptible to biases) and Ratings

Best Prediction Model: Extra Tree Classifier

Our accomplishments:

- Predicted two winners correctly in 2017 and 2018
- Other winners in 2nd or 3rd place.
- Able to replicate the results using balloting.

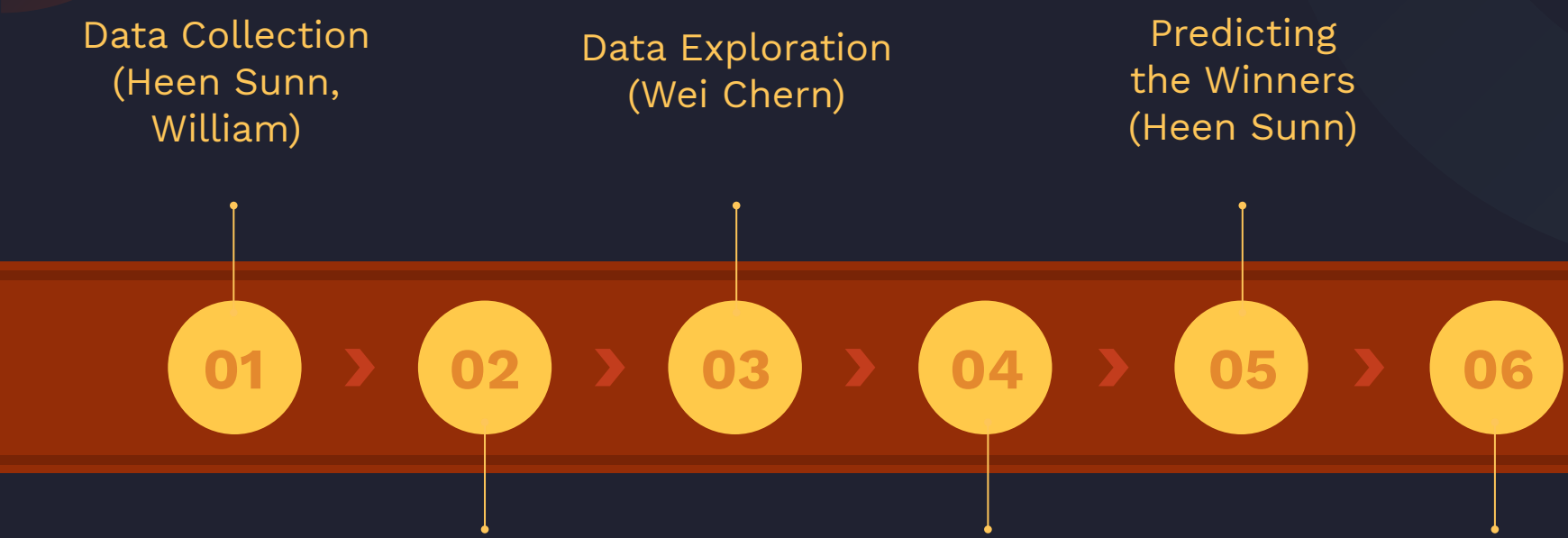
Additional Insight: Director strongly influences a movie's chances at winning the awards.

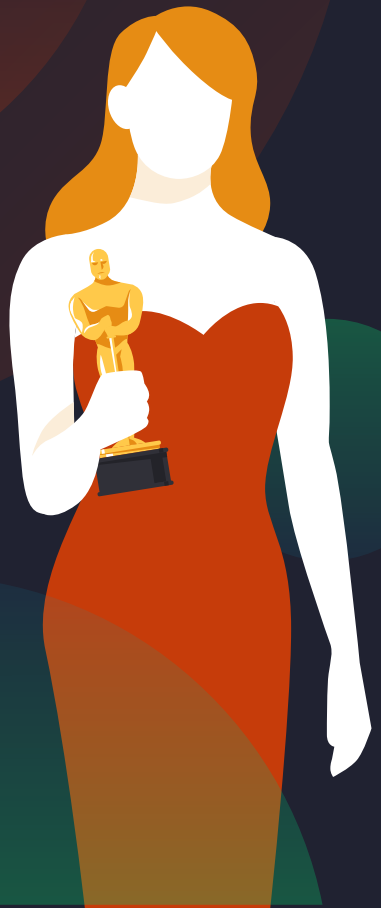
Future Improvements

Expand our predictions to other awards.

Previous Achievements of Directors and Actors

Our Workflow





Thanks!

Do you have any questions?

“I’m very enthusiastic about the Academy Award because if there weren’t Oscars, we wouldn’t have as many good movies as we have now. ”

—Robert Osborne

Whoa!

It could be the part of the presentation where you can introduce yourself, write your email…

4 Main Feature Categories

Film Elements [20]	Movie Critics Rating [5]	Commercial [4]	Film Awards (win & nom) [24]
Runtime (min)	IMDB_rating	Budget	BAFTA
Genre	IMDb votes	Domestic (US) gross	Director Guild Awards (DGA)
	Rotten Tomatoes rating	International gross	Producer Guild Awards (PGA)
	Rotten Tomatoes review	Worldwide gross	Critics' Choice Movie Award (CCMA)
	Metascore		Golden Globes - Comedy & Drama

and 6 more!

Work Distribution

Infographics Make Your Idea Understandable...

Mercury

Mercury is the closest planet to the Sun

Venus

Venus is the second planet from the Sun

Mars

Despite being red, Mars is actually a cold place

Neptune

Neptune is the farthest planet from the Sun

Maybe You Need to Divide the Content

Mercury

Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than our Moon.

Jupiter

Jupiter is a gas giant and the biggest planet in our Solar System. It's named after the Roman god of the sky and lightning

Contents of This Template

Here's what you'll find in this **Slidesgo** template:

1. A slide structure based on a multi-purpose presentation, which you can easily adapt to your needs. For more info on how to edit the template, please visit **Slidesgo School** or read our **FAQs**.
2. An assortment of illustrations that are suitable for use in the presentation can be found in the **alternative resources** slide.
3. A **thanks** slide, which you must keep so that proper credits for our design are given.
4. A **resources** slide, where you'll find links to all the elements used in the template.
5. **Instructions for use**.
6. Final slides with:
    - The **fonts and colors** used in the template.
    - More **infographic resources**, whose size and color can be edited.
    - Sets of **customizable icons** of the following themes: general, business, avatar, creative process, education, help & support, medical, nature, performing arts, SEO & marketing, and teamwork.

You can delete this slide when you're done editing the presentation.

What about Four Columns?

Mercury

Mercury is the closest planet to the Sun and the smallest one in the Solar System

Mars

Despite being red, Mars is actually a cold place. It’s full of iron oxide dust

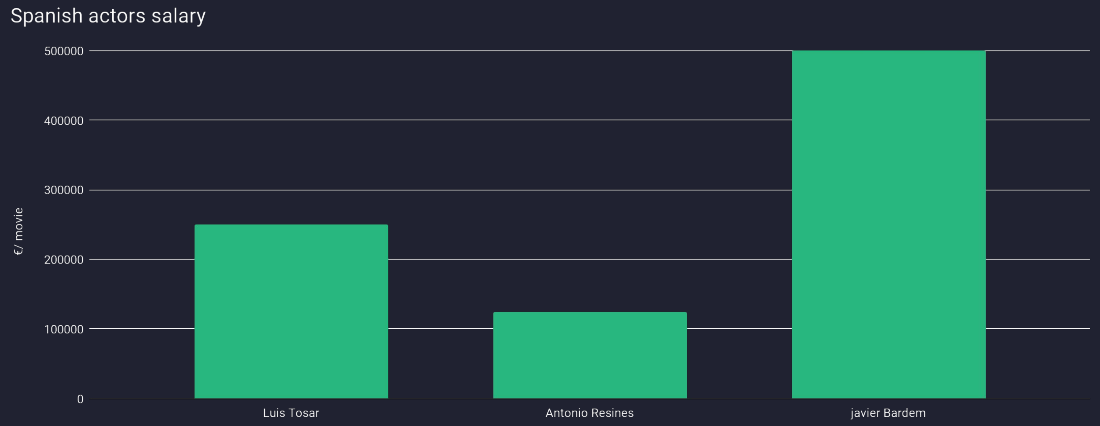
Venus

Venus is the second planet from the Sun. It’s terribly hot and its atmosphere is poisonous

Saturn

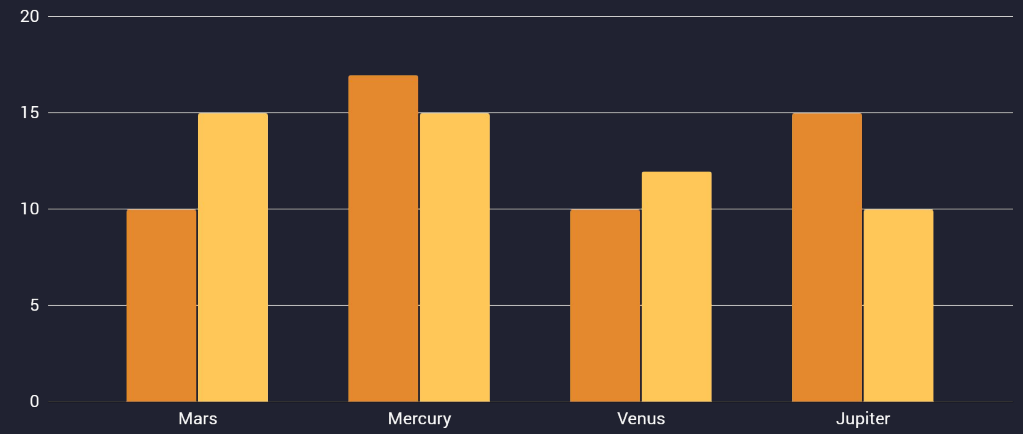
Saturn is the ringed planet. It’s a gas giant, composed mostly of hydrogen and helium

Use a Graph to Show Your Data



If you want to modify this graph, click on it, follow the link, change the data and replace it

Do You Prefer This Graph?



If you want to modify this graph, click on it, follow the link, change the data and replace it

Mercury

Mercury is the closest planet to the Sun

Venus

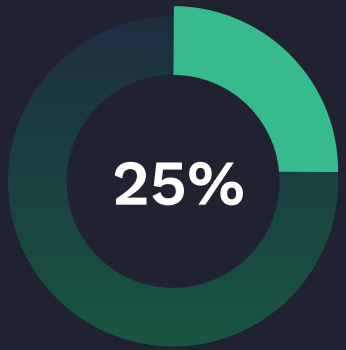
Venus is the second planet from the Sun

You Can Show Some Percentages

Mercury

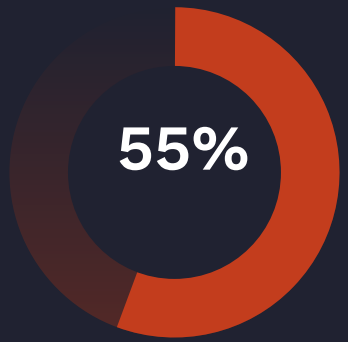
Mercury is the smallest planet in the Solar System

Saturn



Saturn is composed of hydrogen and helium

Venus



Venus has a beautiful name, but it’s terribly hot

Sometimes, Reviewing Concepts Is a Good Idea

Mercury

Mercury is the closest planet to the Sun

Venus

Venus is the second planet from the Sun

Mars

Despite being red, Mars is actually a cold place

Jupiter

It’s the biggest planet in the Solar System

Saturn

Saturn is a gas giant and has rings

Neptune

Neptune is the farthest planet from the Sun

… And the Same Goes for Tables

	Mass (Earths)	Diameter (Earths)	Surface Gravity (Earths)
Mercury	0.06	0.38	0.38
Mars	0.11	0.53	0.38
Saturn	95.2	9.4	1.16

This Is a Map

Venus

Venus has a beautiful name, but it’s terribly hot

Mercury

Mercury is the closest planet to the Sun

Mars

Despite being red, Mars is actually a cold place

A Timeline Always Works Fine

01

Mercury

Mercury is the closest planet to the Sun

02

Venus

Venus has a beautiful name, but it’s terribly hot

03

Saturn

Saturn is composed of hydrogen and helium

04

Neptune

Neptune is the farthest planet from the Sun

How about This Timeline?

Mercury

Mercury is the closest planet to the Sun

01

Saturn

Saturn is composed of hydrogen and helium

02

Venus

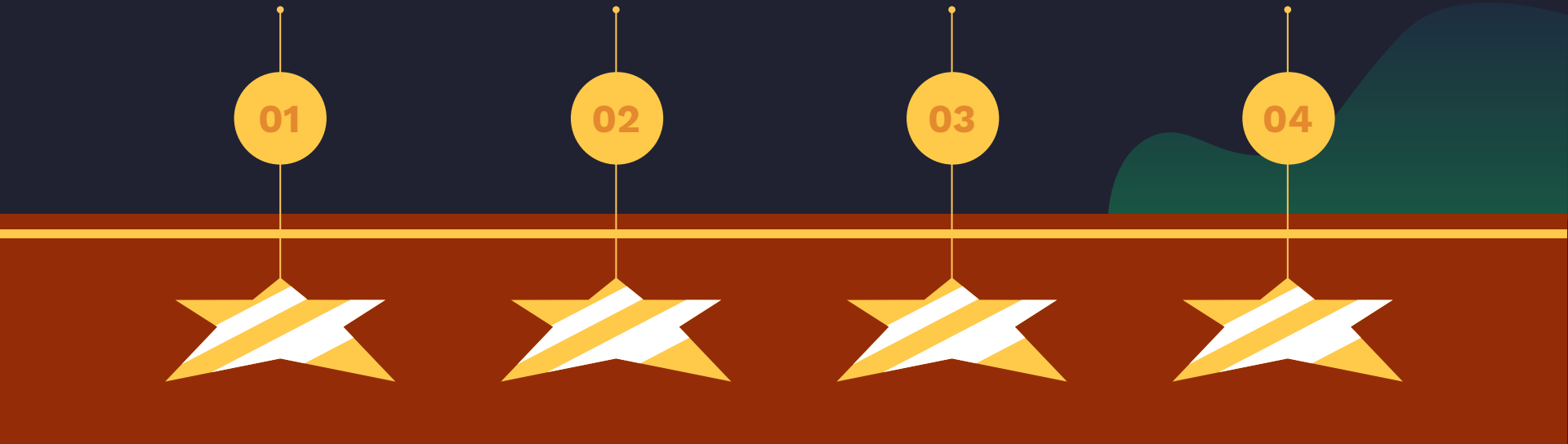
Venus has a beautiful name, but it’s very hot

03

Neptune

Neptune is the farthest planet from the Sun

04




4,498,300
Big numbers catch your audience’s attention

333,000.00

earths is the Sun’s mass

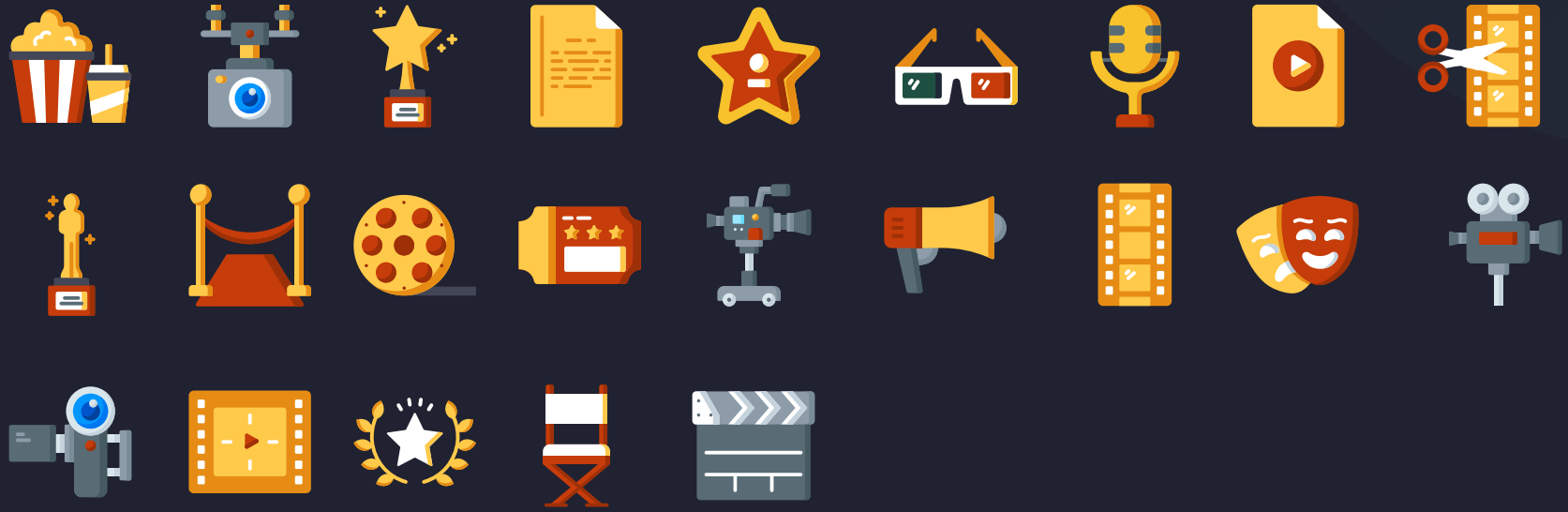
24h 37m 23s

is Jupiter’s rotation period

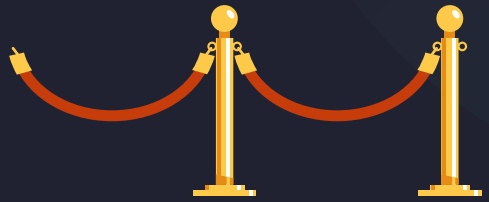
386,000 km

is the distance between Earth and the Moon

Alternative Icons

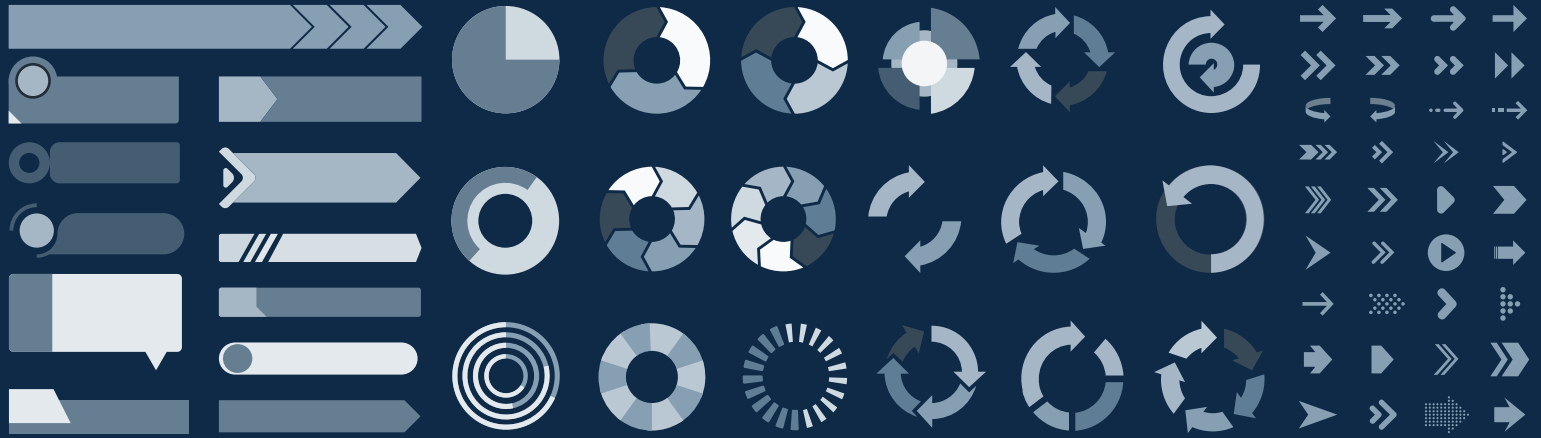


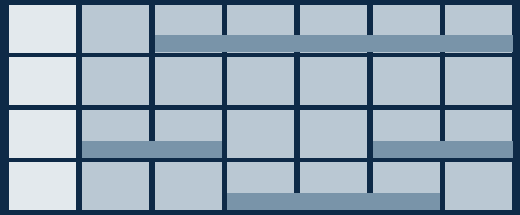
Alternative Resources

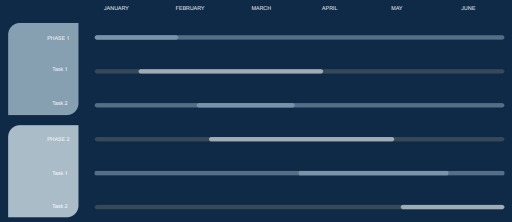


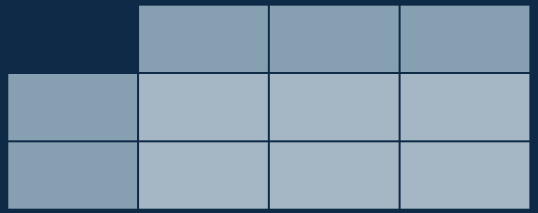
Use our editable graphic resources...

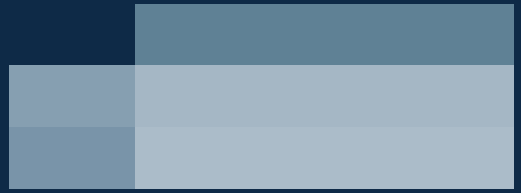
You can easily resize these resources, keeping the quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Don't forget to group the resource again when you're done.

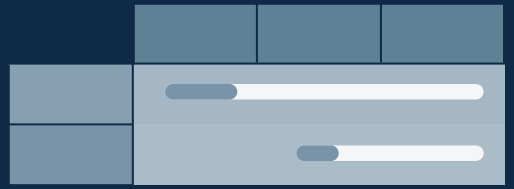


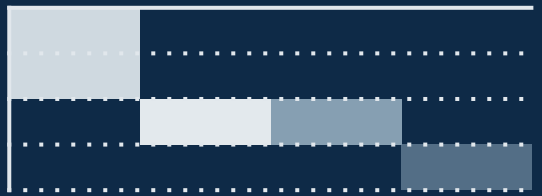


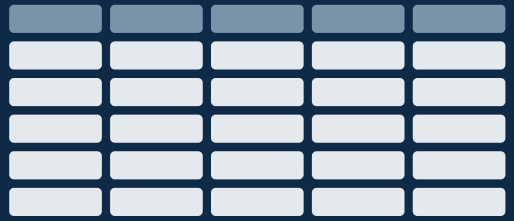

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
PHASE 1
Task 1
Task 2
PHASE 2
Task 1
Task 2

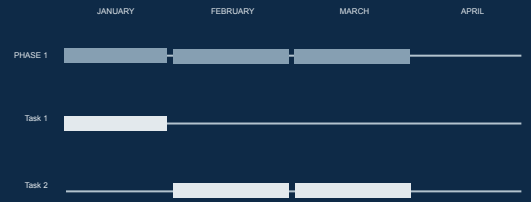











JANUARY
FEBRUARY
MARCH
APRIL
PHASE 1
Task 1
Task 2