Analysis of Movies Based on Online Popularity and Academy Awards

Anish Timila ,Topa Timilsena, Sulav Poudyal, Jared Inting

Department of Computer Science

Texas Tech University, Lubbock TX

May 7, 2017

ABSTRACT

In this paper, we will use a database to determine if a movie's online popularity has any bearing on its performance at the Academy Awards.

1. INTRODUCTION

The internet has become a prevalent part in the lives of people in the last two or three decades. Online perception of politics, pop culture, and other subjects shapes perception of those things in the real world.

We are interested in seeing if the online perception of movies has significantly affected which movies receive Oscars at the Academy Awards. Our datasets span a century, so we can also compare the types and genres of movies that won awards in the past to the kinds that commonly win awards since the internet and social media became a part of everyday life.

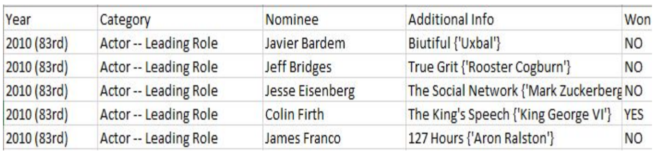
2. DATASETS

The datasets involved in this project are "Top 5000 IMDB Movies around the World" and "Complete List of Oscar Nominees and Winners" from 1927 to 2015.

IMDB stands for the "Internet Movie Database"; on this website, anyone can create an account to rate and discuss movies. People can give a 1 to 10 star rating to movies and the average is displayed as the movie rating. Using this average, we can approximate a movie's online popularity.

We also have information on Facebook likes, so we have a different source of a movie's social media presence and popularity.

Figure 1: A small example of our dataset.



Year	Category	Nominee	Additional Info	Won
2010 (83rd)	Actor -- Leading Role	Javier Bardem	Biutiful {'Uxbal'}	NO
2010 (83rd)	Actor -- Leading Role	Jeff Bridges	True Grit {'Rooster Cogburn'}	NO
2010 (83rd)	Actor -- Leading Role	Jesse Eisenberg	The Social Network {'Mark Zuckerberg	NO
2010 (83rd)	Actor -- Leading Role	Colin Firth	The King's Speech {'King George VI'}	YES
2010 (83rd)	Actor -- Leading Role	James Franco	127 Hours {'Aron Ralston'}	NO

3. OBJECTIVES

Our primary key in the "Top 5000 IMDB Movies around the World" dataset will be 'movie_title' attribute which uniquely identifies each movie in the dataset.

In the "Complete List of Oscar Nominees and Winners" dataset, our primary key will be a numerical ID for each award..

The 'name' attribute from "Complete List of Oscar Nominees and Winners" will be the foreign key linking the two datasets.

We are going to use the "Best Picture" data from the "Complete List of Oscar Nominees and Winners" and ignore the individual actor/writer/composer/etc awards. This is because we are only interested in a movie's overall performance and not the individual performance of specific people.

We are then going to see which movies that have been nominated or won Oscars also have high ratings on IMDB.

Also, since we have data about which movies were only nominated instead of winning, we can see if movies that were the most popular online won more often once the internet became prevalent in society.

4. DATABASE DESIGN

4.1 Conceptual Database Design

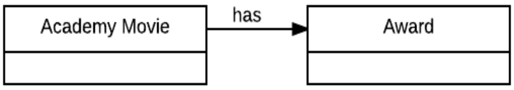


Fig: 4.1(a)

Fig 4.1(a) is the conceptual design model of the "Complete List of Oscar Nominees and Winners" dataset. There are two entities, Academy Movie and Award. The two entities are connected with a 'has' relationship. Each Academy Movie has been nominated for or has won an Award.

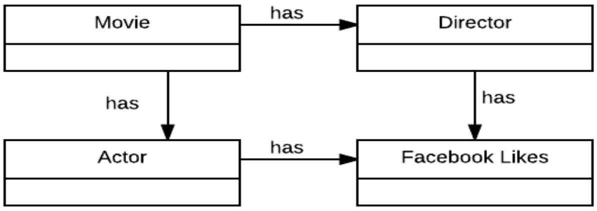


Fig: 4.1(b)

Fig 4.1(b) is the conceptual design model of the "Top 5000 IMDB Movies around the World" dataset. In this dataset, there are four entities: Movie,Director, Actor, Facebook Likes. Each movie has a Director and Actor. Each Director and Actor have Facebook Likes.

4.2 Logical Database Design

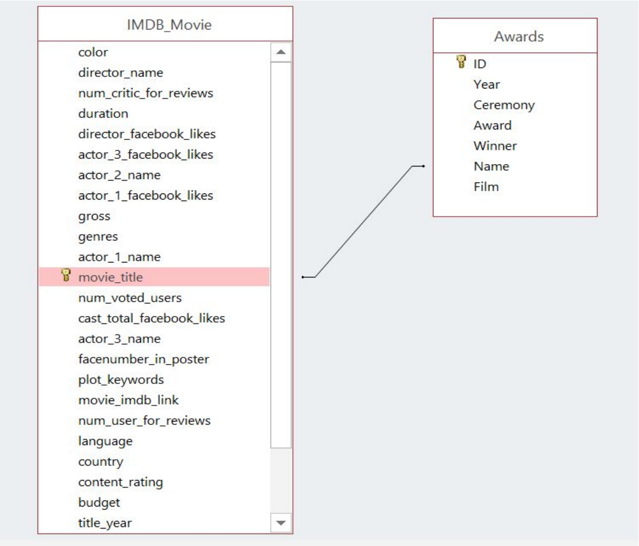
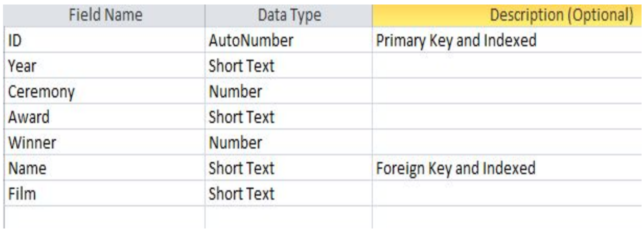


Fig:4.2(a)

The figure, Fig:4.2(a), describes the relationship between the two tables. Both IMDB_Movie and Awards tables have an attribute mentioning the name of movies. The 'movie_title' attribute describes the movie name in the IMDB_Movie table and 'Name' describes the movie that won the 'Best Picture' award. We join the two tables based on this attribute. We use 'Name' instead of 'Film' from Awards table to join the two tables because 'Name' contains the title of movies that won the best picture award whereas 'Film' consists of all the movies which were nominated for categories like 'Best Actor', 'Best Screenwriter' etc. Since we are concerned only with the 'Best Picture', we selected this attribute.

We imported our dataset into the database and noticed that the datasets were inconsistent. In the IMDB_Movie table, the primary key is 'movie_title' and every other attribute is dependent on it. There were no other dependencies, so we didn't need to normalize this table. In the Awards table, there was no unique attribute so we had to generate a primary key for this table. There was no dependency on this table as well so we didn't need to normalize Awards table.

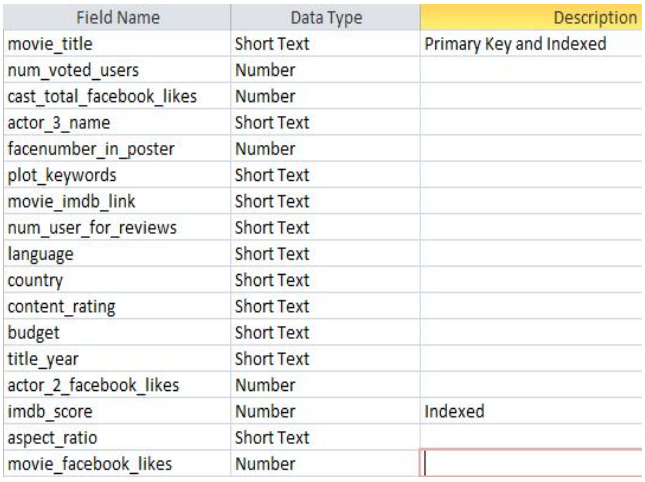
4.3 Physical Database Design



Field Name	Data Type	Description (Optional)
ID	AutoNumber	Primary Key and Indexed
Year	Short Text	
Ceremony	Number	
Award	Short Text	
Winner	Number	
Name	Short Text	Foreign Key and Indexed
Film	Short Text	

Fig: 4.3(a)

Figure 4.3(a) describes the physical model of the Award table. We auto generated the primary key 'ID' while the attribute 'Name' is the foreign key.



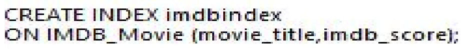
Field Name	Data Type	Description
movie_title	Short Text	Primary Key and Indexed
num_voted_users	Number	
cast_total_facebook_likes	Number	
actor_3_name	Short Text	
facenumber_in_poster	Number	
plot_keywords	Short Text	
movie_imdb_link	Short Text	
num_user_for_reviews	Short Text	
language	Short Text	
country	Short Text	
content_rating	Short Text	
budget	Short Text	
title_year	Short Text	
actor_2_facebook_likes	Number	
imdb_score	Number	Indexed
aspect_ratio	Short Text	
movie_facebook_likes	Number	

Fig: 4.3(b)

For the IMDB_Movie table, the primary key is the attribute 'movie_title' as shown in figure 4.3(b).

4.4 Index and Constraints

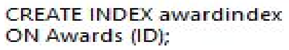
4.4.1 Index



```
CREATE INDEX imdbindex
ON IMDB_Movie (movie_title,imdb_score);
```

Fig: 4.4(a)

In the IMDB_Movie table, the attributes that are mostly used are 'movie_title' and ''imdb_score'. We index these attributes by using the create index statement shown in 4.4(a). We can check that these attributes were indexed by looking at their physical design as shown in figure 4.3(b).

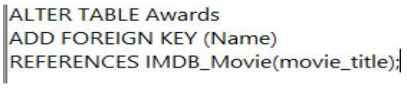


```
CREATE INDEX awardindex
ON Awards (ID);
```

Fig: 4.4(b)

The 'ID' and 'Name' attributes are indexed from the Awards table because we use them in all of our queries. 'ID', being the primary key, is automatically indexed and 'Name' is indexed using the create index statement as shown in figure 4.4(a). We can check that the attribute are indexed by checking the physical design as shown in figure 4.3(a).

4.4.2 Constraints



```
ALTER TABLE Awards
ADD FOREIGN KEY (Name)
REFERENCES IMDB_Movie(movie_title);
```

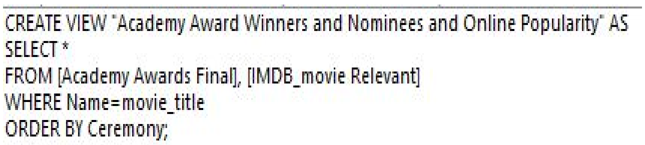
Fig: 4.4.2(b)

We added a foreign key constraint on the Awards table which references the 'movie_title' attribute from the IMDB_Movie table.

5. IMPLEMENTATION

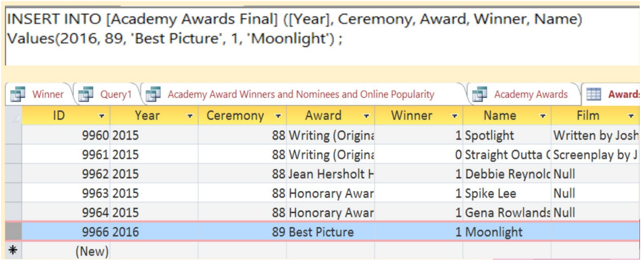
5.1 Validation

We cross validated our datasets by inserting a new row, deleting an existing row and updating existing data in the views we created. After performing the cross validation operations in the views, we made sure that the changes were reflected properly in the base tables' data.



```
CREATE VIEW "Academy Award Winners and Nominees and Online Popularity" AS
SELECT *
FROM [Academy Awards Final], [IMDB_movie Relevant]
WHERE Name=movie_title
ORDER BY Ceremony;
```

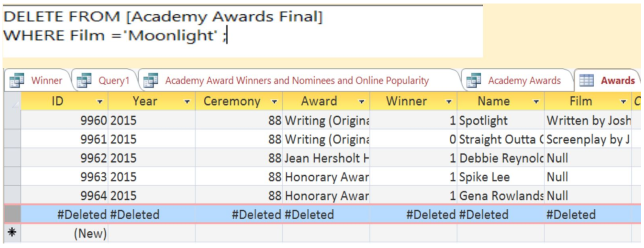
Fig: 5.1(a) : Creating a view from the Academy Award Final and IMDB_movie Relevant



```
INSERT INTO [Academy Awards Final] ([Year], Ceremony, Award, Winner, Name)
Values(2016, 89, 'Best Picture', 1, 'Moonlight') ;
```

ID	Year	Ceremony	Award	Winner	Name	Film
9960	2015	88	Writing (Origina	1	Spotlight	Written by Josh
9961	2015	88	Writing (Origina	0	Straight Outta (	Screenplay by J
9962	2015	88	Jean Hersholt H	1	Debbie Reynolc	Null
9963	2015	88	Honorary Awar	1	Spike Lee	Null
9964	2015	88	Honorary Awar	1	Gena Rowlands	Null
9966	2016	89	Best Picture	1	Moonlight	
(New)						

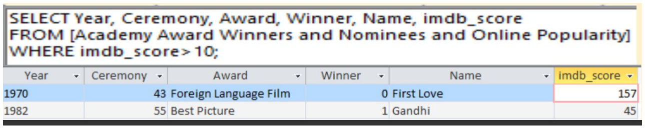
Fig: 5.1(b) Inserting a new row through the view and the updated base table after insertion.



```
DELETE FROM [Academy Awards Final]
WHERE Film ='Moonlight' ;
```

ID	Year	Ceremony	Award	Winner	Name	Film	C
9960	2015	88	Writing (Origina	1	Spotlight	Written by Josh	
9961	2015	88	Writing (Origina	0	Straight Outta (	Screenplay by J	
9962	2015	88	Jean Hersholt H	1	Debbie Reynolc	Null	
9963	2015	88	Honorary Awar	1	Spike Lee	Null	
9964	2015	88	Honorary Awar	1	Gena Rowlands	Null	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	
(New)							

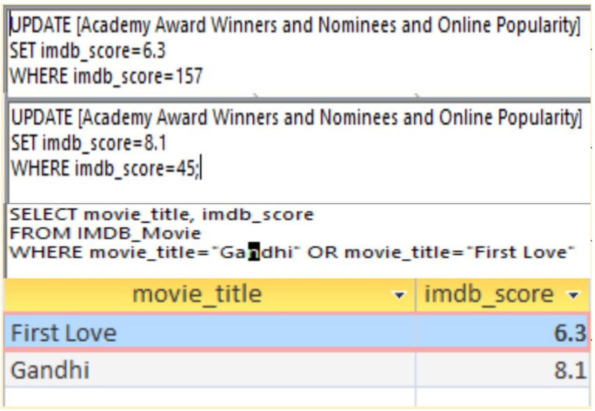
Fig: 5.1(c) Deleting the last row through the view and the updated base table after deletion



```
SELECT Year, Ceremony, Award, Winner, Name, imdb_score
FROM [Academy Award Winners and Nominees and Online Popularity]
WHERE imdb_score>10;
```

Year	Ceremony	Award	Winner	Name	imdb_score
1970	43	Foreign Language Film	0	First Love	157
1982	55	Best Picture	1	Gandhi	45

Fig: 5.1(d) A query is done to check for any false imdb_scores and displaying the result of the query.



```
UPDATE [Academy Award Winners and Nominees and Online Popularity]
SET imdb_score=6.3
WHERE imdb_score=157
```

```
UPDATE [Academy Award Winners and Nominees and Online Popularity]
SET imdb_score=8.1
WHERE imdb_score=45;
```

```
SELECT movie_title, imdb_score
FROM IMDB_Movie
WHERE movie_title="Gandhi" OR movie_title="First Love"
```

movie_title	imdb_score
First Love	6.3
Gandhi	8.1

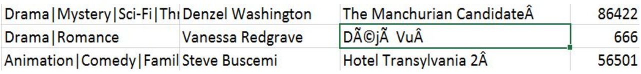
Fig: 5.1(e) Displaying the imdb_scores after updating to the correct scores. Below are the sources for the correct scores.

http://www.imdb.com/title/tt0083987/ (Gandhi)
http://www.imdb.com/title/tt0065703/ (First Love)

6. CONCLUSION

6.1 Challenges

Both the datasets that we used for our research were raw and had to cleaned. There were a lot of special characters and symbols such as Ã, Â, © in the datasets (as shown in Fig 6.1(a)). We analyzed and cleaned them through MS Access with the "Find and Replace Tool".



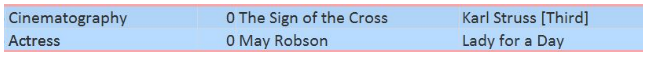
Drama|Mystery|Sci-Fi|Thı	Denzel Washington	The Manchurian CandidateÂ	86422
Drama|Romance	Vanessa Redgrave	DÃ©jÃ VuÂ	666
Animation|Comedy|Famil	Steve Buscemi	Hotel Transylvania 2Â	56501

Fig 6.1(a)

We also found some invalid data as discussed in the cross validation section of our paper; some of the movies had an IMDB rating exceeding 10, which is impossible because the IMDB website only allows users to score movies based on a 1 to 10 star rating. We solved this issue by updating the scores through SQL queries.

There were also data inconsistencies in the Academy Awards Table. The "Award" attribute had many award types such as 'Best Actor','Best Actress', 'Best Director', 'Best Picture', etc.

Because "Award" could be awarded to an individual person, OR a movie, the dataset would place the titles of movies under the "Name" attribute if the award was for a movie, but it would also place an individual actor's or director's name under the "Name" attribute if the award was for an individual person. In those cases, the movie title was placed in the "Film" attribute. This is shown in Fig 6.1(b))



Cinematography	0	The Sign of the Cross	Karl Struss [Third]
Actress	0	May Robson	Lady for a Day

Fig 6.1(b)

This made the table's data very inconsistent and made it impossible to normalize properly.

One final challenge we faced was that the "Best Picture" award category that we were interested in had several name changes since 1927. We had to research to find out what the award's name was over the years and then alter our queries accordingly so that we were actually getting the data we wanted from the datasets. Our research showed that the award's name changed four times since the beginning of the Academy Awards:

1927-1929: **Academy Award for Outstanding Picture**
1929-1940: **Academy Award for Outstanding Production**
1941-1943: **Academy Award for Outstanding Motion Picture**
1944-1961: **Academy Award for Best Motion Picture**
1962-Present: **Academy Award for Best Picture**

6.2 Contribution

Our project’s main goal was to determine whether or not a movie’s online presence affected its performance at the Academy Awards. Obviously, a movie would not have an “online presence” until the internet became a popular medium for sharing thoughts. Thus, we had to pick a cut-off point for when the internet became prevalent. We researched this subject and it seems that there is no real consensus. Because of this, we had to arbitrarily decide on a cut-off year. Our group chose 2006 because this is the year that Facebook became open to the public. The 2006 cut-off point is represented as a green line in Fig 6.2(a) and a blue line in Fig 6.2(b).

We were also interested in seeing if people of today rated movies which won Academy Awards more highly than movies that didn’t win. In Fig.6.2(a), movies that were nominated for awards within one ceremony fall on the same vertical line (the x axis is the ceremony number). Movies marked by an orange dot are the ones which actually won the award.

While our group did not run actual statistical analysis on this data, we can see that the orange dot is very often the movie with the highest rating for each ceremony.

There are interesting cases to look at in this figure. For example, the highest rated movie which was also nominated for an award was The Shawshank Redemption, which has an IMDB rating of 9.2. However, it did not win an Academy Award because in the same year, it was up against Forrest Gump.

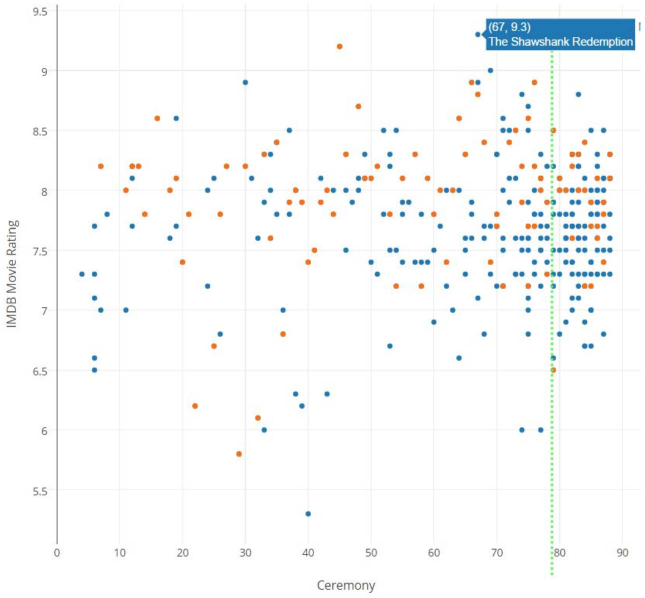


Fig 6.2(a)

We also examined the Facebook likes of movies which won awards as shown in Fig 6.2(b). In this figure, the larger the dot for a movie, the more Facebook likes the movie received. The most liked movie was Birdman from 2014 with 141,000 likes.

It is also interesting to note that movies dating all the way back to the 1930s have had Facebooks created for them. The likes are probably disproportionate in these cases because people of today will not generally seek out the Facebook of an older film unless they already like the movie.

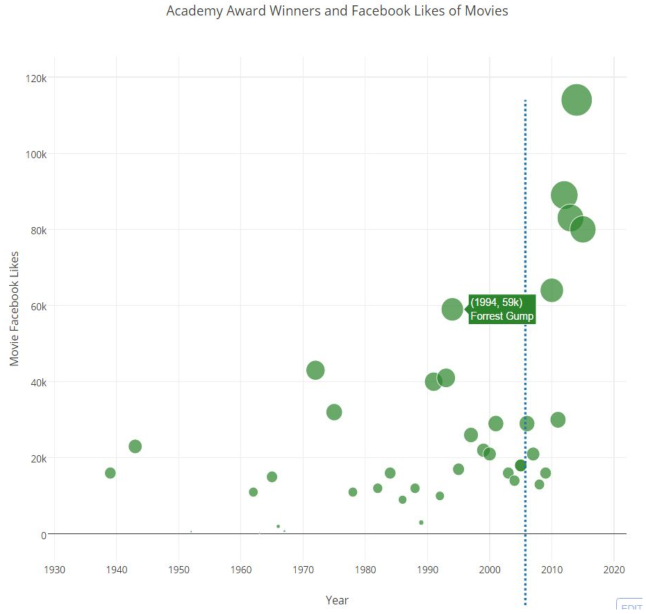


Fig 6.2(b)

The last thing our group analyzed was the types of genres that typically won Academy Awards. This analysis was slightly difficult, because in our dataset, each movie had several genres listed. For example, a movie might have this for a genre: Action|Adventure|Comedy

Thus, we plotted the movies by the first genre listed, which on IMDB is generally the movie’s main genre. While this might mean that this graph is not 100% representative of every genre, the data is still useful to examine.

From this data, we can see clearly that movies in the Drama genre tend to win most awards. We can also see that movies in the Horror genre are completely absent from Academy Award winners that were also rated in the top 5000 movies.

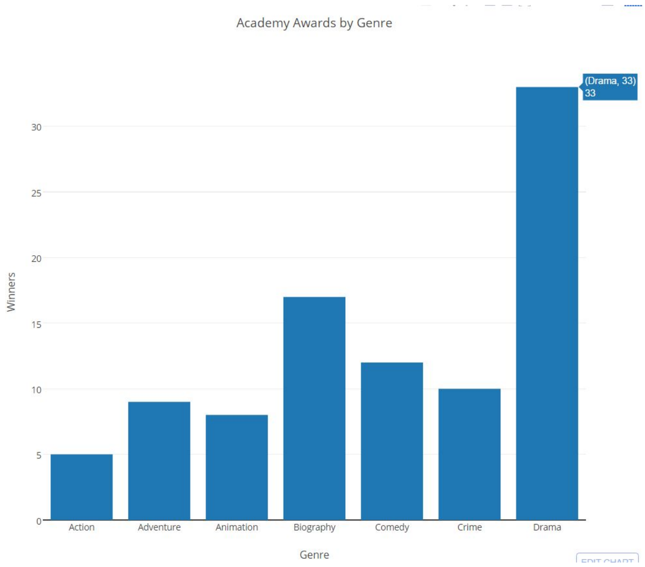


Fig 6.2(c)

7. ACKNOWLEDGEMENT

Sulav Poudyal: Conceptual and Logical Model, SQL Queries
Anish Timila: Data Cleaning, Indexes, Visualizations, Github
Jared Inting: SQL Queries, Constraints
Topa Timilsena: Cross-Validation

8. TOOLS

- Plotly
- Microsoft Access
- GitHub Static Pages
- Github
- Lucid Chart

9. REFERENCES AND CITATIONS

Sun, C. (2016, August 22). IMDB 5000 Movie Dataset |Kaggle. Retrieved March 22, 2017, from https://www.kaggle .com/deepmatrix/imdb-5000-movie-dataset

Complete List of Oscar Nominees and Winners. (2011, March 15). Retrieved March 22, 2017, from https://www.aggdata.com/awards/oscar

10. PROJECT GITHUB RESOURCES

Graphs: https://databaseprojectttu.github.io/

Project Files: https://github.com/DatabaseProjectTTU