Movie projections using Relational Databases and Classification

The Wonderbolts

Chris Arcand, Rich Jeffery, and Brian Salter

Higher. Faster. Further.

Motivation

Using existing data, can we accurately predict how well upcoming movies will do?

(And will this make us lots of $$$?)

Motivation

Using existing data, can we accurately predict how well upcoming movies will do?

(And will this make us lots of $$$?)

Known metrics, such as:
Cast, Director, Release date, Rating, Duration, Subject matter...

Motivation

Using existing data, can we accurately predict how well upcoming movies will do?

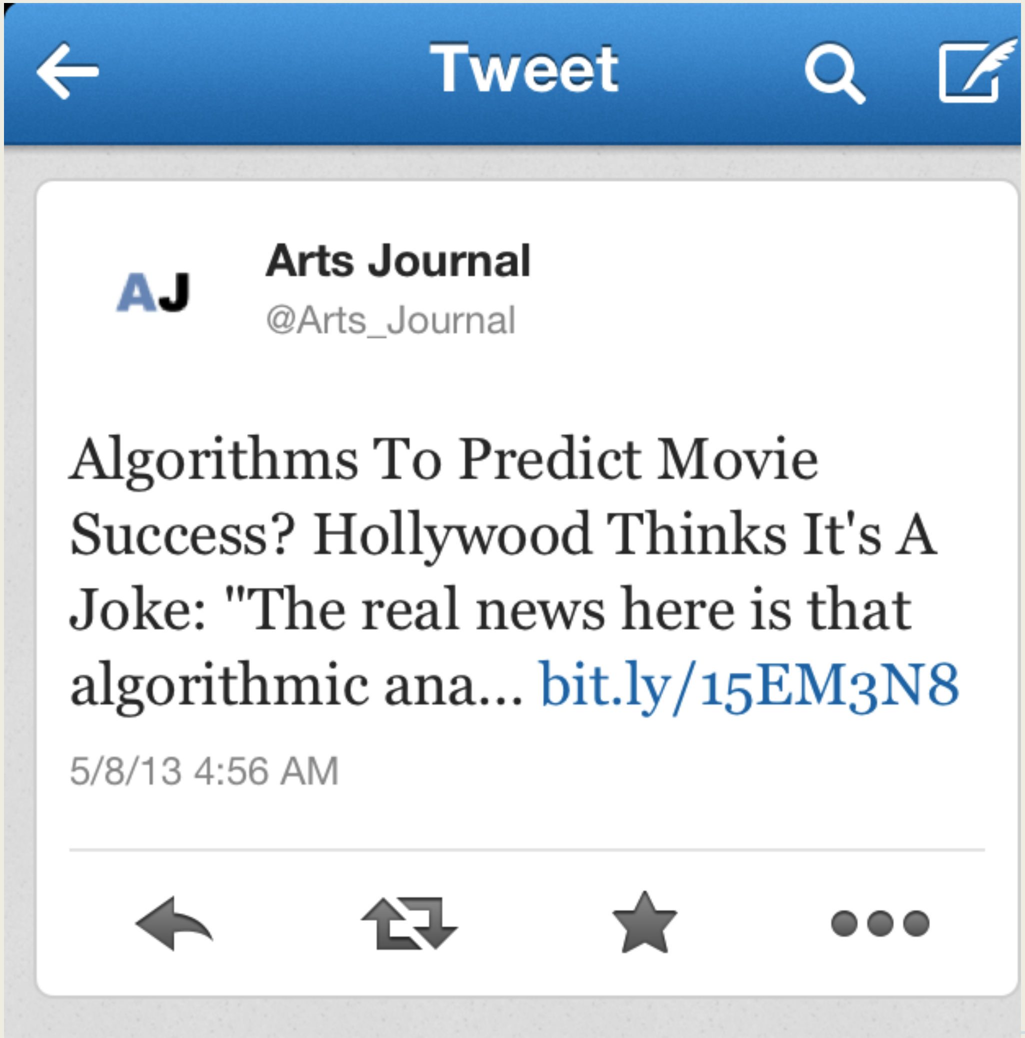
(And will this make us lots of $$$?)

Known metrics, such as:

Cast, Director, Release date, Rating, Duration, Subject matter...

Find a pattern in these metrics, and use it to make a prediction

CHALLENGE ACCEPTED!!!

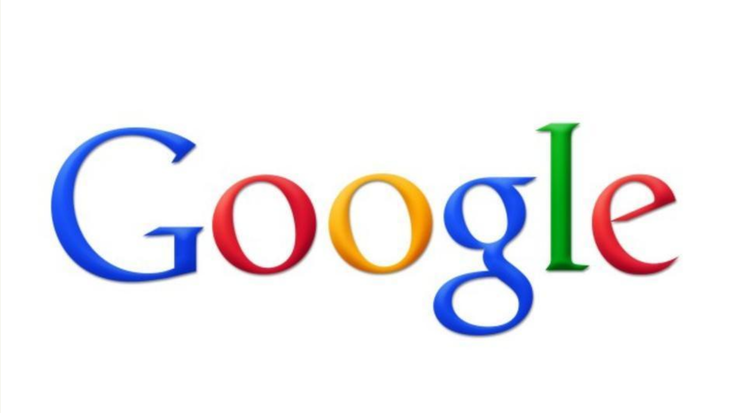


Arts Journal
@Arts_Journal

Algorithms To Predict Movie Success? Hollywood Thinks It's A Joke: "The real news here is that algorithmic ana... bit.ly/15EM3N8

5/8/13 4:56 AM

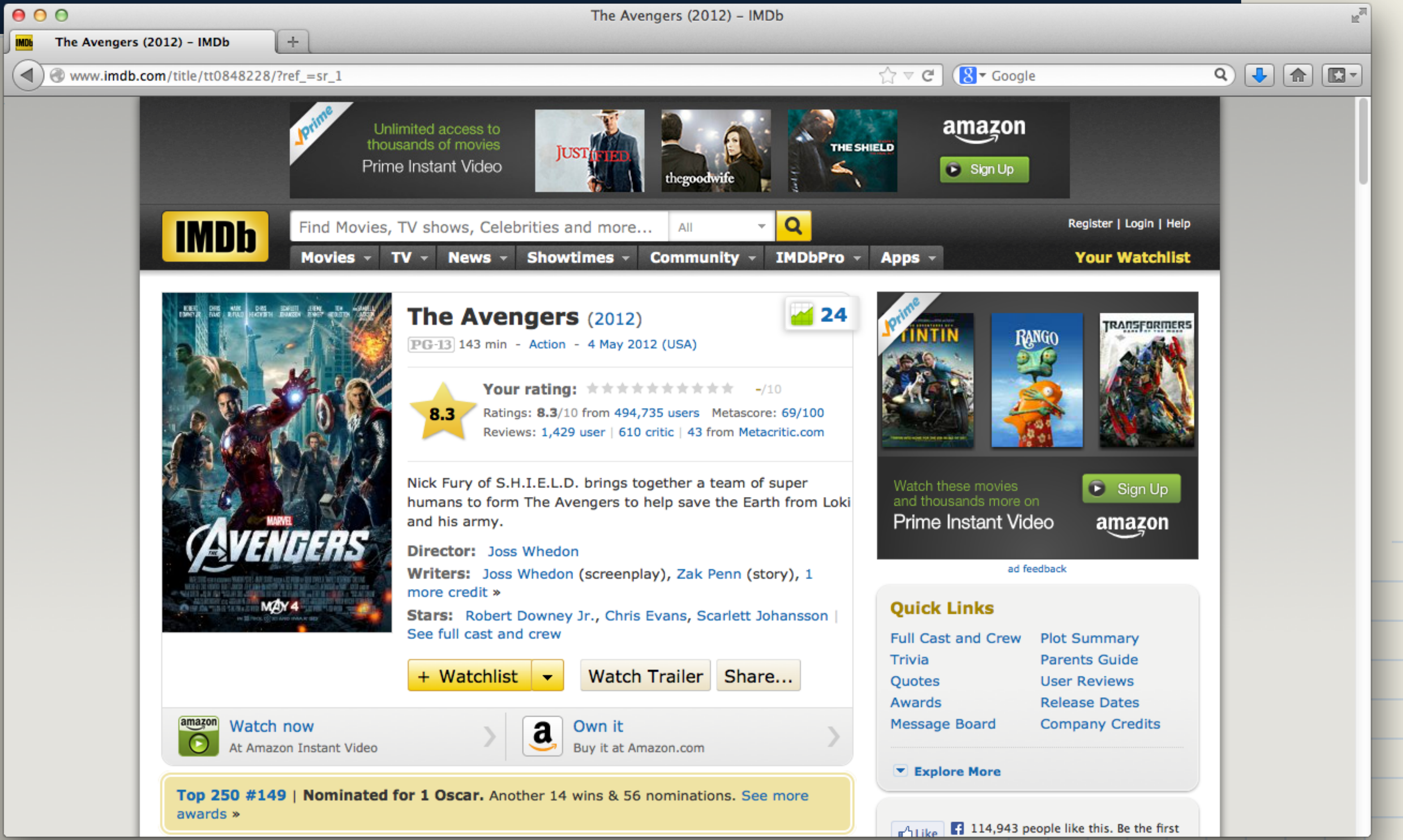
Where do you go when you need information on movies?



Follow up question: When you use Google, what is the first link that appears?

IMDb

Internet Movie Database



The Avengers (2012)

PG-13 143 min - Action - 4 May 2012 (USA)

Your rating: ★★★★★★★★★★ -/10

8.3

Ratings: 8.3/10 from 494,735 users Metascore: 69/100

Reviews: 1,429 user | 610 critic | 43 from Metacritic.com

Nick Fury of S.H.I.E.L.D. brings together a team of super humans to form The Avengers to help save the Earth from Loki and his army.

Director: Joss Whedon

Writers: Joss Whedon (screenplay), Zak Penn (story), 1 more credit »

Stars: Robert Downey Jr., Chris Evans, Scarlett Johansson | See full cast and crew

Watch now
At Amazon Instant Video

Own it
Buy it at Amazon.com

Top 250 #149 | **Nominated for 1 Oscar.** Another 14 wins & 56 nominations. See more awards »

Quick Links

Full Cast and Crew	Plot Summary
Trivia	Parents Guide
Quotes	User Reviews
Awards	Release Dates
Message Board	Company Credits

Explore More

114,943 people like this. Be the first

How to get the data?

IMDB API

A lightweight, RESTful web interface to IMDb

www.imdbapi.com

(Make JSON queries to fill our database)

How to get the data?

~~IMDB API~~

~~A lightweight, RESTful web interface to IMDb~~

~~www.imdbapi.com~~

-Lots of overhead in automating queries

-Huge data. Bandwidth isn't cheap (or plentiful)

-3,000 queries per hour limit

4,115,950 rows in 'person' table -> 57 days of processing for ONE table

IMDB API would have been slow. Very, very slow.

How to get the data?

IMDbPY

Open source project to retrieve and manage IMDb data
imdbpy.sourceforge.net

-Written in pure Python (with a few lines of C!)
-Released under GPL 2 License
-Can both retrieve data from IMDb directly (web)
-Can also parse data in IMDb's plain text files

How to get the data?

IMDbPY

Open source project to retrieve and manage IMDb data
imdbpy.sourceforge.net

-Written in pure Python (with a few lines of C!)
-Released under GPL 2 License
-Can both retrieve data from IMDb directly (web)
-Can also parse data in IMDb's <u>plain text files</u>

Minor forgotten detail...

IMDb releases subsets of the plain text file dumps of their database!
www.imdb.com/interfaces/

49 files
6.75GB when uncompressed

Part 3/3 - The Database

-IMDbPY doesn't support Oracle

(open source, yo)

Suggestion: SQLite -> Oracle

Part 3/3 - The Database

-IMDbPY doesn't support Oracle

(open source, yo)

~~Suggestion: SQLite -> Oracle~~

-SQLite doesn't work well with large data

-In addition, 4GB limit on Oracle tablesize == bad

MySQL to the rescue

Finally, loading the data

IMDb text files + IMDbPY + MySQL

~1.5 hours to process in final run

Many attempts required to get it right:
-Database issues
-Dependency issues within Python
-Resource issues (VPS vs. decent machine)

Lesson learned: Big data requires thoughtful planning and knowledge to set up and maintain effectively.

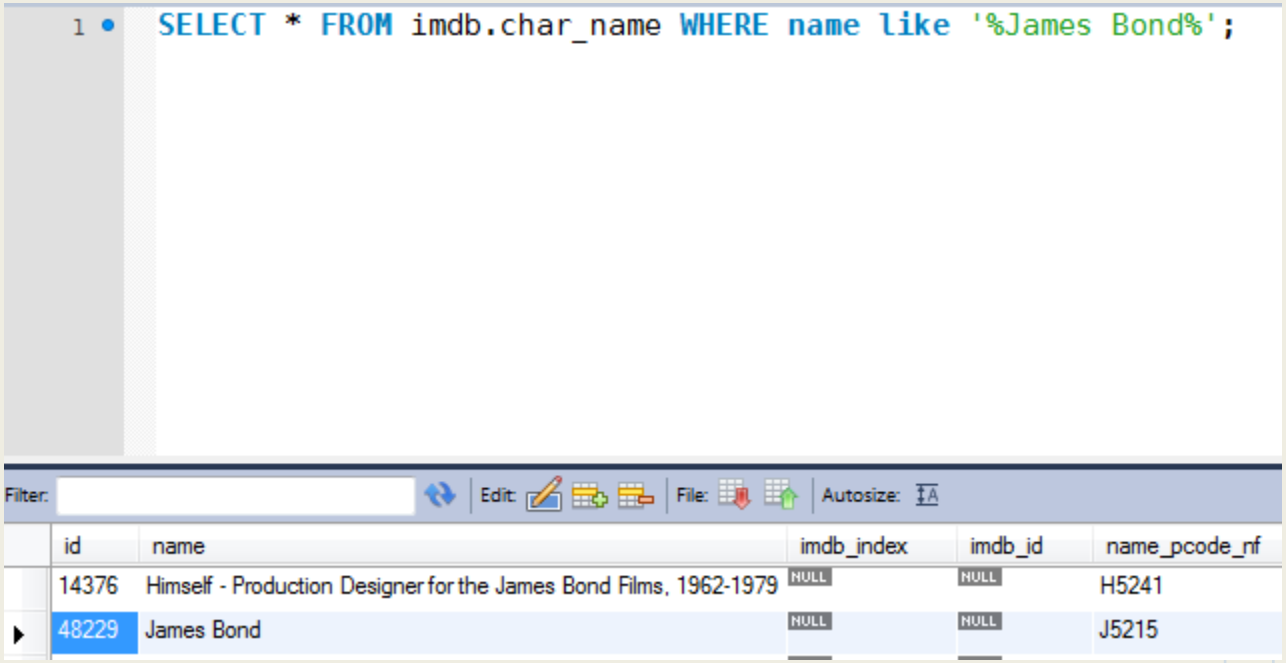
What does the data look like?

Let's see an example!

What if we wanted to actually do something with this data as is? Suppose we wanted to find out which actor played the best James Bond.

First things first...

Firstly we need to find the role_id of James Bond from the char_name table. Easy enough.

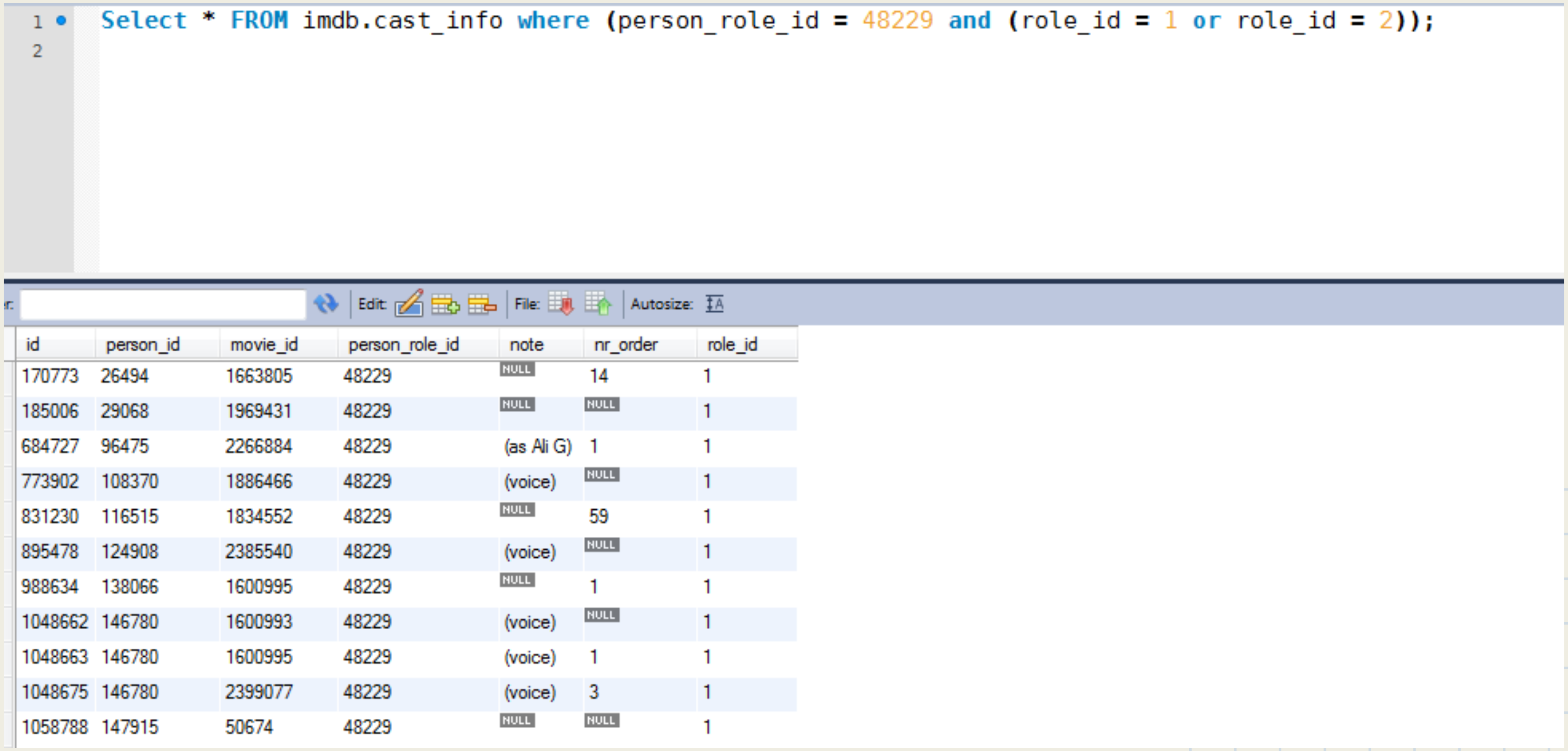


```
SELECT * FROM imdb.char_name WHERE name like '%James Bond%';
```

id	name	imdb_index	imdb_id	name_pcode_nf
14376	Himself - Production Designer for the James Bond Films, 1962-1979	NULL	NULL	H5241
48229	James Bond	NULL	NULL	J5215

Now what?

Well now we need to find the person_id for every person who has played James Bond. We do this by looking in the cast_info table for the role_id.



```
Select * FROM imdb.cast_info where (person_role_id = 48229 and (role_id = 1 or role_id = 2));
```

id	person_id	movie_id	person_role_id	note	nr_order	role_id
170773	26494	1663805	48229	NULL	14	1
185006	29068	1969431	48229	NULL	NULL	1
684727	96475	2266884	48229	(as Ali G)	1	1
773902	108370	1886466	48229	(voice)	NULL	1
831230	116515	1834552	48229	NULL	59	1
895478	124908	2385540	48229	(voice)	NULL	1
988634	138066	1600995	48229	NULL	1	1
1048662	146780	1600993	48229	(voice)	NULL	1
1048663	146780	1600995	48229	(voice)	1	1
1048675	146780	2399077	48229	(voice)	3	1
1058788	147915	50674	48229	NULL	NULL	1

...Many queries later...

- Once we have the person_id list for people that played James Bond, we need to get scores for each movie where that actor played James Bond and find their average review score.
- We also need to limit our search to theatrical releases only.
- This involves a query that joins role_id, cast_info, title, movie_info_idx, and info_type tables...
- After taking half an hour to run the query we get the following data:

Actor	Average James Bond Film Score
Sean Connery	7.0
George Lazenby	6.8
Roger Moore	6.6
Timothy Dalton	6.6
Pierce Brosnan	6.5
Daniel Craig	7.5

What if we wanted to do something simpler?

What if we just wanted to decide how good an actor is by examining the average scores of his movies?

The query for this question looks like:

```
Select AVG(info)  FROM (Select imdb.cast_info.movie_id, info, info_type_id, person_id, role_id
FROM imdb.cast_info inner join imdb.movie_info_idx on imdb.cast_info.movie_id = imdb.
movie_info_idx.movie_id) As T where (person_id = 1468525 and (role_id = 1 or role_id = 2) and
info_type_id = 101);
```

Running this query takes upwards of 10 minutes and tells us that the late Rod Steiger has an average review score of 6.2.

That seems like a lot of time and work...

If we want to look at hundreds of thousands of movies, and millions of actors we need to change something...

We need to change the world... or at least the structure of our database.

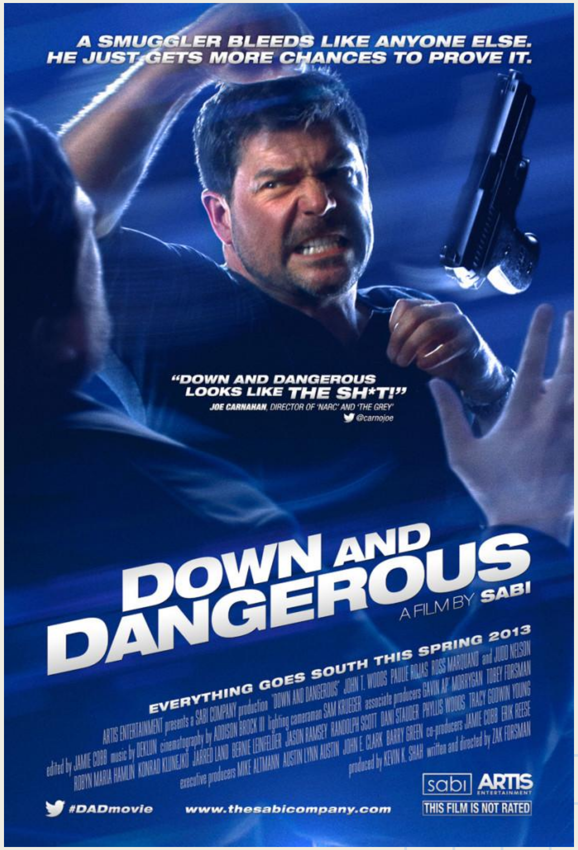
Data Manipulation

- 2.7 million columns
- 100K rows

How do we deal with this much data?

What we lost...

- Actors not in a movie
- Redundant entries
- Independent films
- Low-budget films
  - < $1,000,000


A SMUGGLER BLEEDS LIKE ANYONE ELSE.
HE JUST GETS MORE CHANCES TO PROVE IT.
"DOWN AND DANGEROUS LOOKS LIKE THE SH*T!"
JOE CARNAHAN, DIRECTOR OF 'NARC' AND 'THE GREY'
@carnojoe
DOWN AND DANGEROUS
A FILM BY SABI
EVERYTHING GOES SOUTH THIS SPRING 2013
ARTIS ENTERTAINMENT presents a SABI COMPANY production 'DOWN AND DANGEROUS' JOHN T. WOODS PAULIE ROJAS ROSS MARQUAND and JUDD NELSON
edited by JAMIE COBB music by DEKLUN cinematography by ADDISON BROCK III lighting cameraman SAM KRUEGER associate producers GAVIN AP' MORRYGAN TOBEY FORSMAN
ROBYN MARIA HAMLIN KONRAD KLINEJKO JARRED LAND BERNIE LEINFELDER JASON RAMSEY RANDOLPH SCOTT DANI STAUDER PHYLLIS WOODS TRACY GODWIN YOUNG
executive producers MIKE ALTMANN AUSTIN LYNN AUSTIN JOHN E. CLARK BARRY GREEN co-producers JAMIE COBB ERIK REESE
produced by KEVIN K. SHAH written and directed by ZAK FORSMAN
sabi ARTIS ENTERTAINMENT
THIS FILM IS NOT RATED
#DADmovie
www.thesabicompany.com

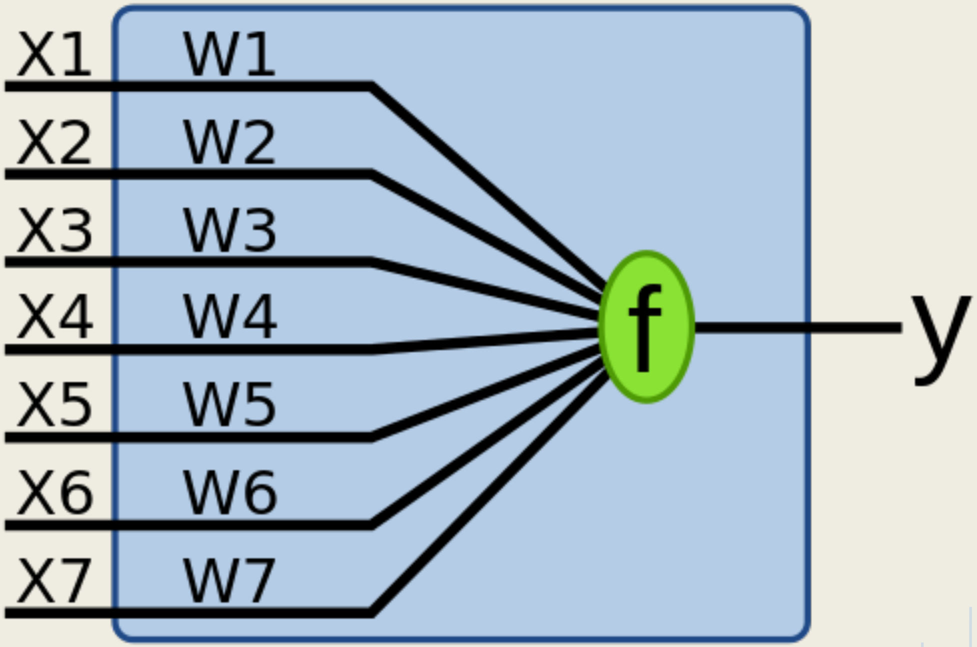
dataGet.py

- Matlab hates us
- Store data in text files
  - data_2000.txt
  - target_2000.txt

- Read in data

Perceptron

- Many inputs
- Many weights



How it works...

- Dot products
  - Feature vector
  - Weight vector

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Minimize test error
  - misclassified / total

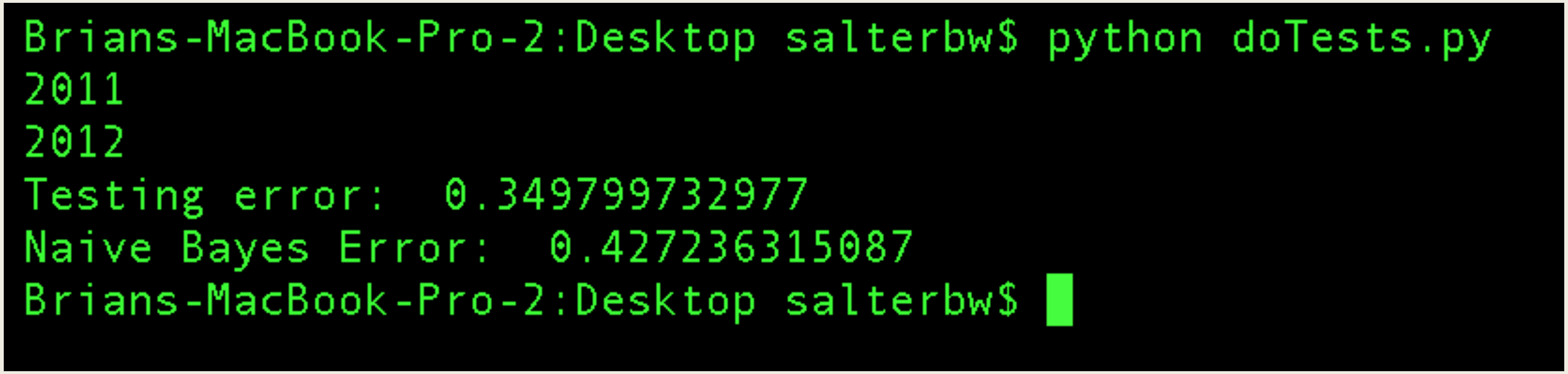
Naive Bayes Classifier

- Probabilities
  - P(Class|Features)
- Learns P(F|C)

$$p(C|F_1,\dots,F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

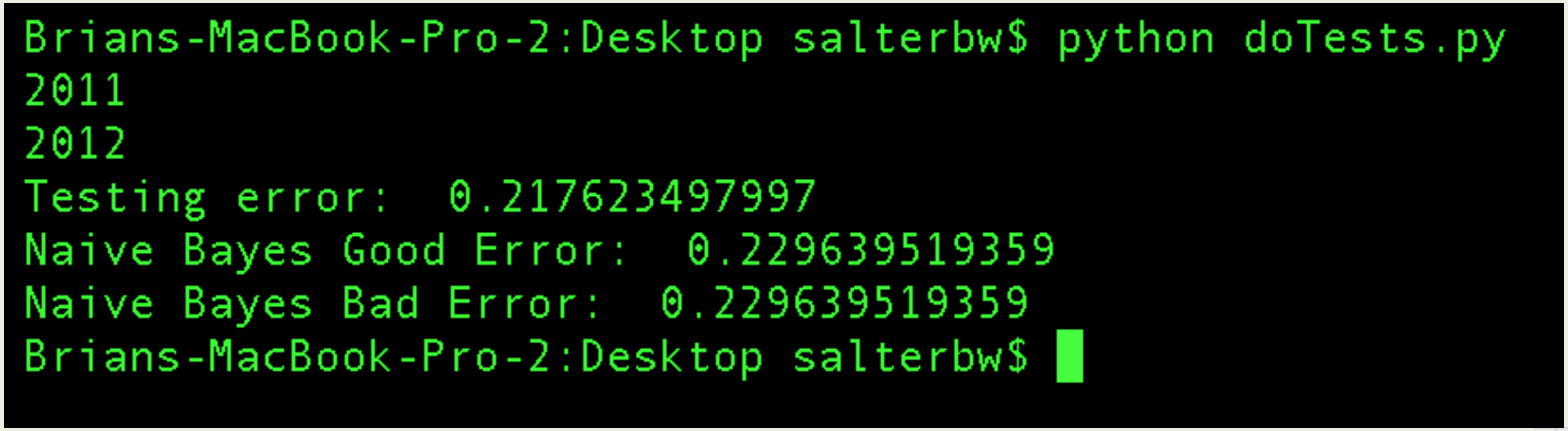
Computes P(C|F) to predict

6.5 Star Threshold



```
Brians-MacBook-Pro-2:Desktop salterbw$ python doTests.py
2011
2012
Testing error:  0.349799732977
Naive Bayes Error:  0.427236315087
Brians-MacBook-Pro-2:Desktop salterbw$ 
```

7 Star Threshold



```
Brians-MacBook-Pro-2:Desktop salterbw$ python doTests.py
2011
2012
Testing error:  0.217623497997
Naive Bayes Good Error:  0.229639519359
Naive Bayes Bad Error:  0.229639519359
Brians-MacBook-Pro-2:Desktop salterbw$ 
```

What comes next...

1. Increase Timespan
2. More Features
3. Multiclass
4. Multi-layers

Summary

1. Big data == Lots of time to load

Summary

1. Big data == Lots of time to load
2. Big data == Lots of time to maintain

Summary

1. Big data == Lots of time to load
2. Big data == Lots of time to maintain
3. Big data == Lots of time to analyze

4. Make good, smart choices about how/what you are doing

5. Challenge defeated, @ArtsJournal

FIN