Data Analytics

TMDB Movie

Data Set

Alka Patel - CEBD 1260

Problem Definition

Revenue Prediction

The goal of this project was to determine the revenue of the movie based on the it's:

- **Budget**
- **Production Country**
- **Original Production Language**
- **Genre**

Dataset Description

TMDB and Grouplens

- Movie **Details**
- Movie **Keywords**
- **45,000** Movies
- Released on or before **2017** (not complete data for 2017.

- Movie **ratings**

Dataset Description Cont.

Movie Details

```
Column Names
Index(['adult', 'belongs_to_collection', 'budget', 'genres', 'homepage', 'id',
       'imdb_id', 'original_language', 'original_title', 'overview',
       'popularity', 'poster_path', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'video',
       'vote_average', 'vote_count'],
      dtype='object')
```

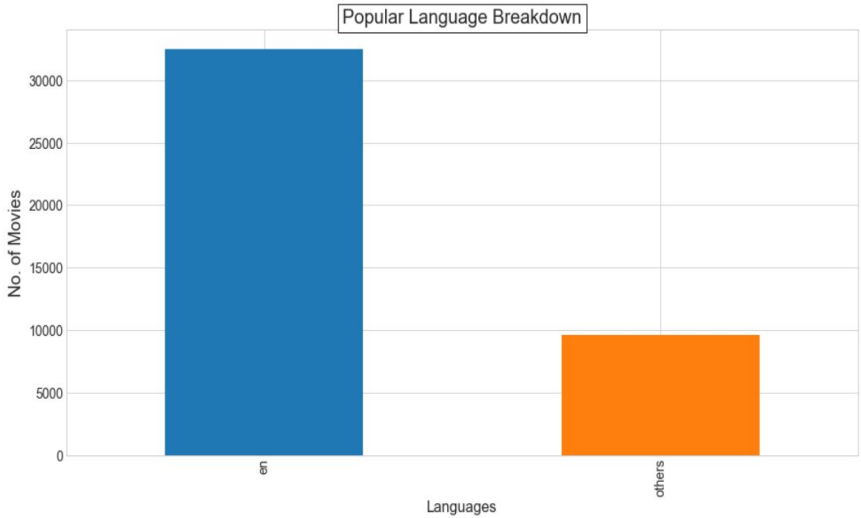
Dataset Description Cont.

Movie Details

- Languages

Took only the top 10 Languages out of 92:

en	32269
fr	2438
it	1529
ja	1350
de	1080
es	994
ru	826
hi	508
ko	444
zh	409



Dataset Description Cont.

Movie Details

- **Status**

Took only movies with status released:

Released	41450
Rumored	200
Post Production	87
In Production	18
Planned	15
Canceled	2

Dataset Description Cont.

Movie Keywords

```
Column Names
Index(['id', 'keywords'], dtype='object')
```

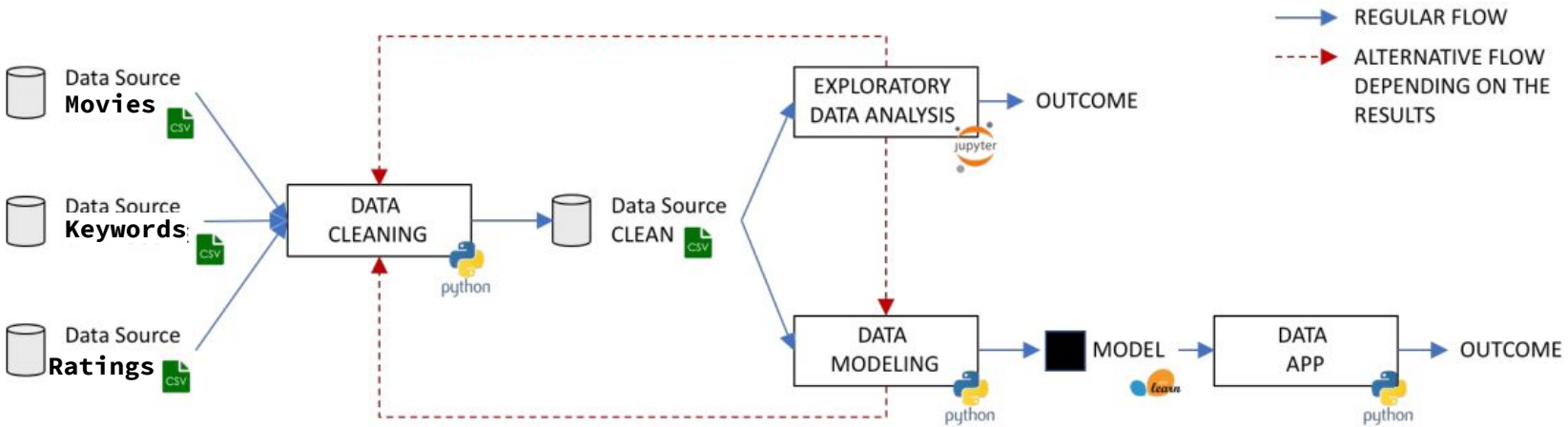
Dataset Description Cont.

Movie Ratings

```
Column Names
Index(['userId', 'movieId', 'rating', 'timestamp'], dtype='object')
```

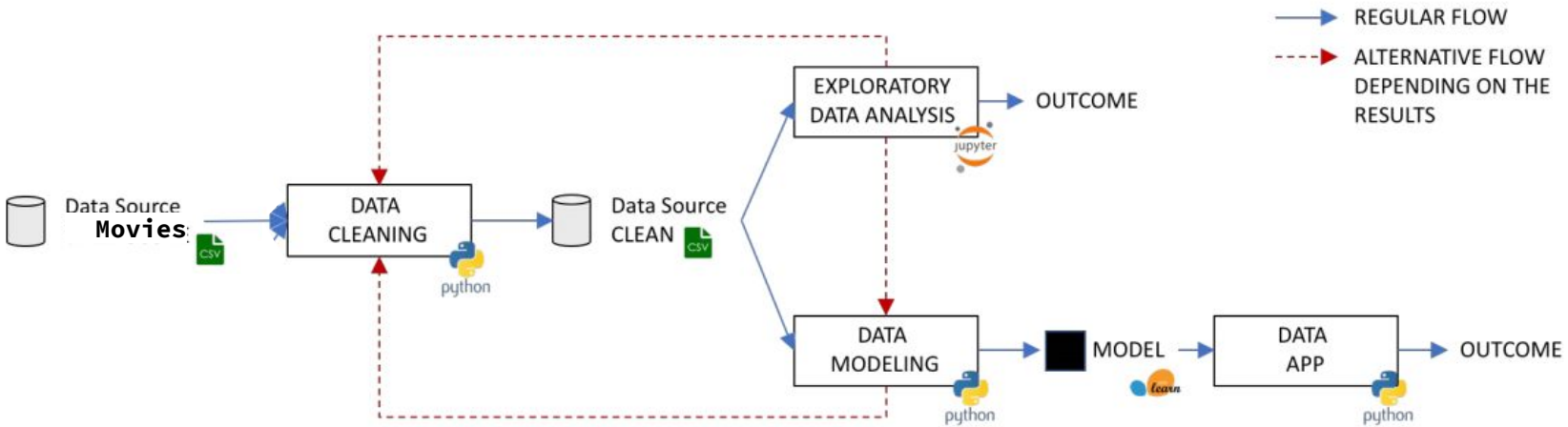
Solution

We all know where this image was borrowed from...



Solution

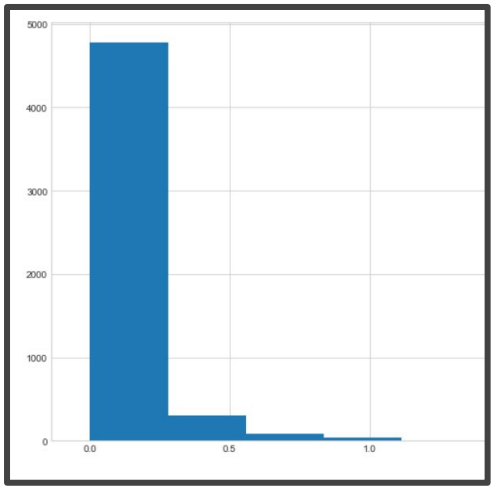
We all know where this image was borrowed from...

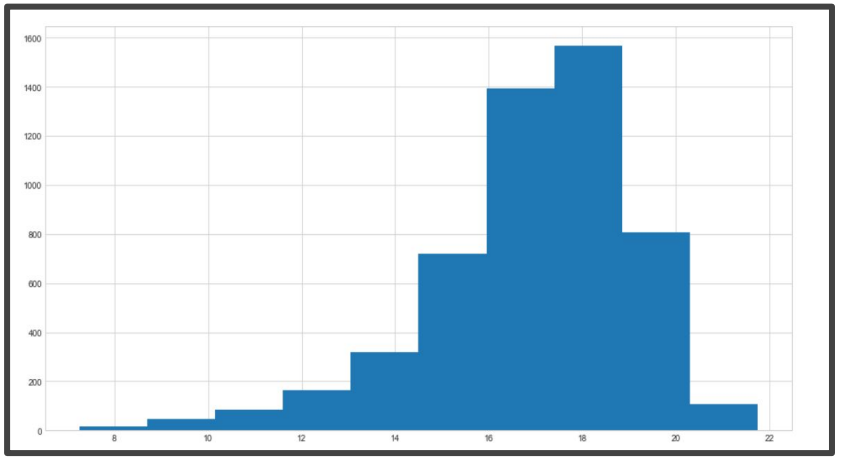


Solution - Random Forest 100

Feature Engineering

- Normalized Budget and Revenue that are over 1000 (dropped the rest).





Solution - Random Forest 100

Feature Engineering

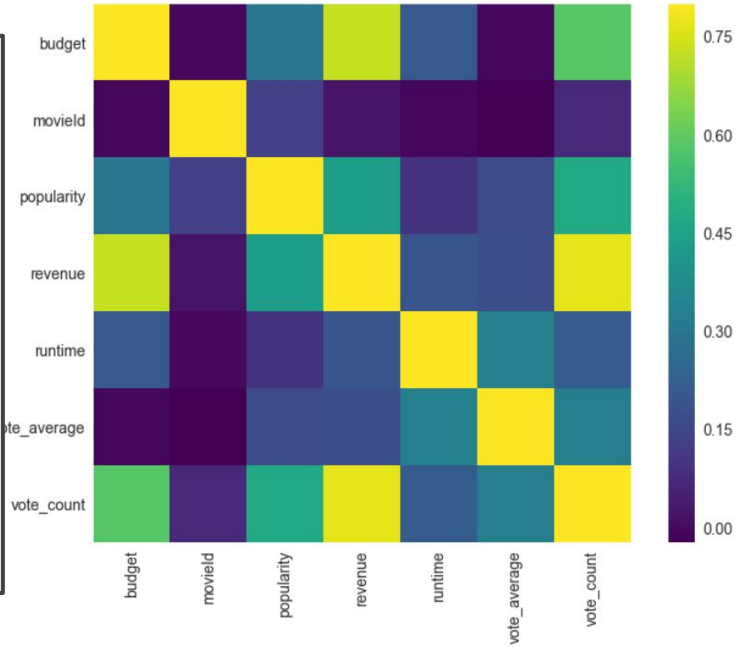
➔ **Dummies Dataframes for :**

- Genre
- Production Countries
- Language
- Production Company

Solution - Choosing X Colums

X_columns = ['budget_log','vote_count','popularity']+ list(df_language.columns) +
list(df_prod_country.columns) + list(df_genres.columns)

	budget	movieId	popularity	revenue	runtime	vote_average	vote_count
budget	1.00	-0.00	0.30	0.73	0.20	-0.00	0.58
movieId	-0.00	1.00	0.13	0.02	-0.01	-0.02	0.07
popularity	0.30	0.13	1.00	0.44	0.10	0.17	0.48
revenue	0.73	0.02	0.44	1.00	0.20	0.18	0.77
runtime	0.20	-0.01	0.10	0.20	1.00	0.33	0.21
vote_average	-0.00	-0.02	0.17	0.18	0.33	1.00	0.33
vote_count	0.58	0.07	0.48	0.77	0.21	0.33	1.00



Solution - Random Forest 100

Best Model

	model	mae	rmse
2	RandomForestRegressor100	0.967	1.434
1	RandomForestRegressor10	1.013	1.501
0	LinearRegression	1.069	1.548
3	KNeighborsRegressor	1.153	1.639
4	DecisionTreeRegressor	1.330	1.988

Results - Random Forest 100

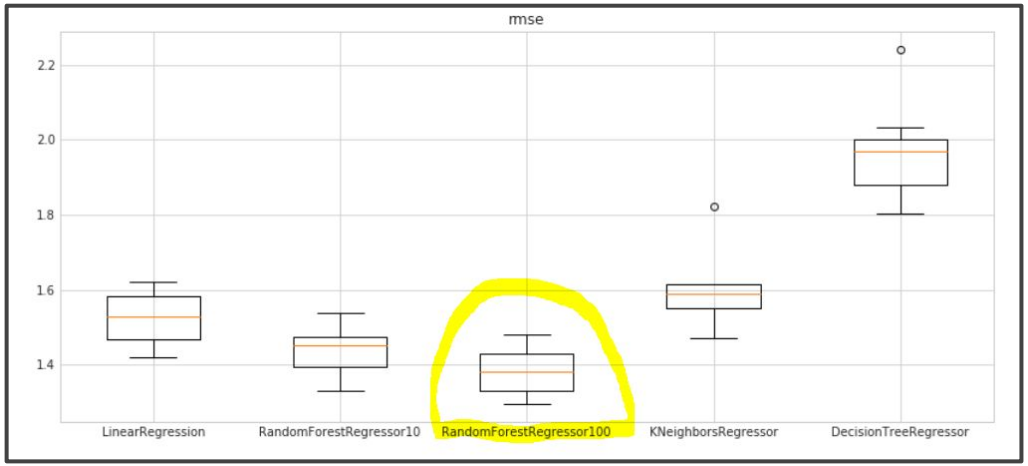
Best Model

Feature Importance

	0	1
1	vote_count	0.497
0	budget_log	0.232
2	popularity	0.078
93	'Comedy'	0.012
96	'Drama'	0.011
107	'Thriller'	0.011
35	'France'	0.009
90	'Action'	0.008
104	'Romance'	0.008
94	'Crime'	0.008

Results - Random Forest 100

Mean Absolute Error and Root Mean Square Error after Shuffling



Discussion

Error Analysis - Where were the highest Errors on the Model?

- Huge Difference between the Mean & Median Profits of that category
  - Ex. Drama Genre
- Huge Difference between the Highest Profits and Lowest Profits
  - Ex. Italy as a Production Country; Foreign Genre
- Very few records on the dataset
  - Ex. Croatia (Just 1 record with a huge loss)

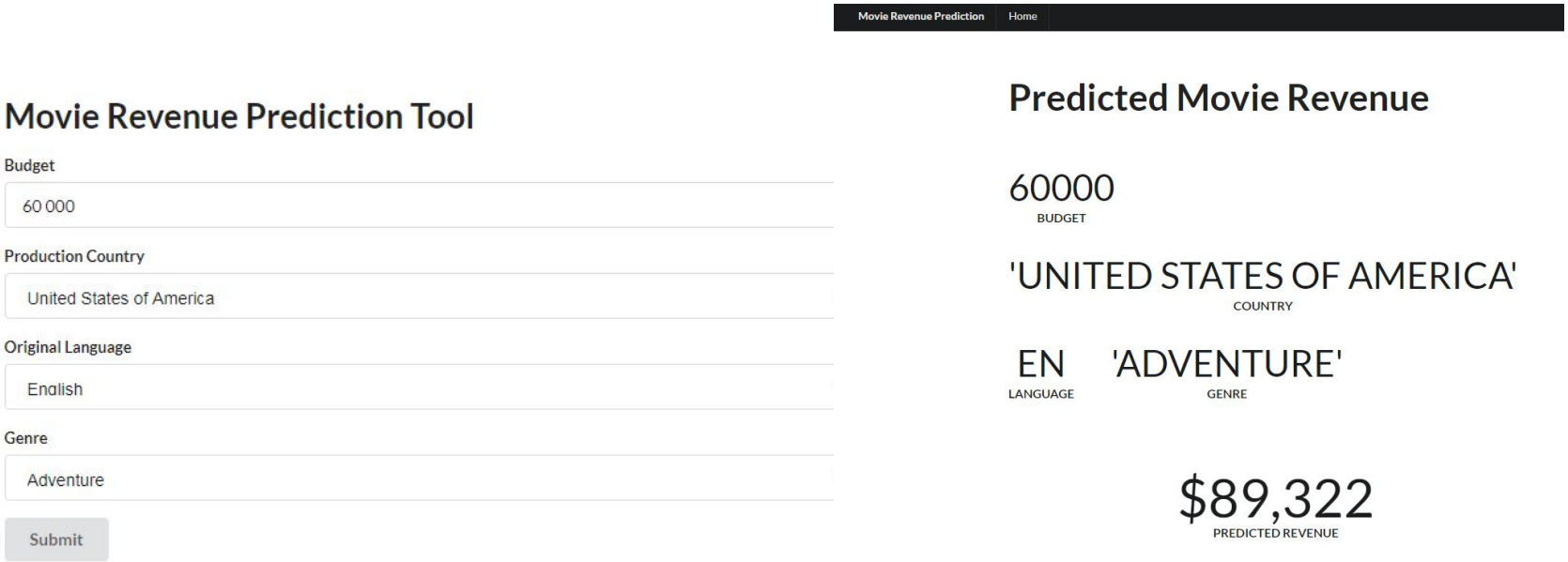
Discussion

Conclusive Thoughts

- Need very large datasets that has many, many examples of all scenarios to have accurate models.
- As a personal afterthought, it would be nice to try and create a recommendation system using the GroupLens Dataset.
- App:
  - https://movie-revenue-prediction.herokuapp.com/

Discussion

My Heroku App



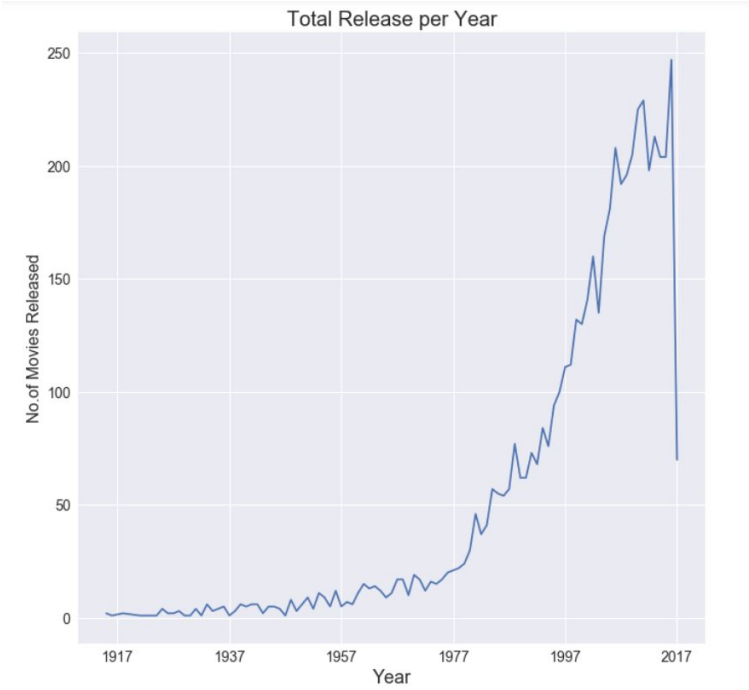
Thank you!!!

References:
https://www.themoviedb.org/documentation/api
https://grouplens.org/datasets/movielens/latest/
https://www.kaggle.com/rounakbanik/the-movies-dataset/home
https://www.themoviedb.org/talk/5141d424760ee34da71431b0

Extra (personal) Notes

Popular Genres for the highest 100 grossing movies.

```
adventure : 80
action : 55
fantasy : 44
family : 35
science : 31
fiction : 31
animation : 20
comedy : 18
thriller : 15
drama : 12
romance : 8
crime : 6
mystery : 3
```



Extra (personal) Notes

My class assignment was done on the Movielens data set from kaggle. This data set is a mix of data from TMDB and Grouplens. The dataset contained metadata of the movies, and also the TMDB popularity, voting average, and count of votes. Additionally, from Grouplens, the rating data of 26, 000, 000 records was available. However, due to computing powers on my end, I used the smaller sample of 100, 000 records. For exploratory purposes, many data fields represented the movie, some of the columns were: Genres, movieId (that linked back to the movieId of a keyword dataset for the movies and the rating dataset from Grouplens), original language, original title, Production Company, release date, spoken languages, production languages and the runtime. The numerical attributes of revenue, budget, average TMDB vote, TMDB vote count, TMDB popularity, revenue, and budget was also given.
To further explain some of the parameters, the TMDB voting system is on a scale of 10, the higher the vote, the better the movie. So the average vote is sum of all the votes divided by the number of votes (provided in another column). The popularity represents how many people saved this movie to watch later, put it in their favorites, or page views etc. It can be from 0 to infinity. The Grouplens rating on the other hand, is on a scale of 0 to 5. The higher the rate, the better liked the movie.