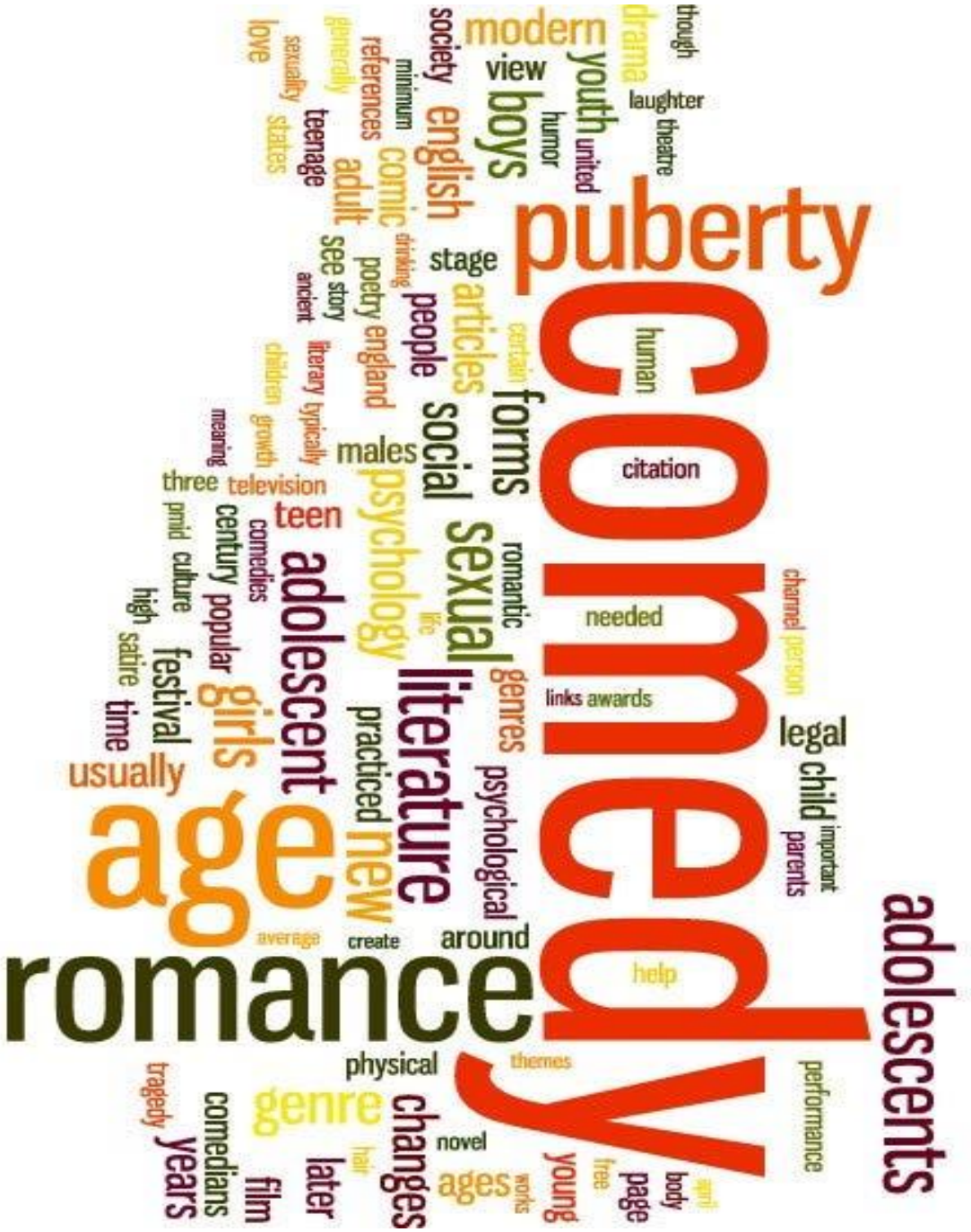


Movie Popularity Prediction

Report

Team ID: CS_7

Team members:

Member:	ID
هبة محمد صالح عقيلي محمد	20201700961
دنيا ايهاب عبد البديع علي	20201700245
مريم محمد محمود أحمد	20201700818
سما محمد العشري السيد	20201700359
ايه محمد عثمان حسن	20201700173
ريهام هارون موسى بطرس	20201700297

Our Dataset Definition:

For our analysis, we got a dataset contains 3039 rows each row consist of 17 independent features and 1 dependent feature (vote_average).

Dataset Sample:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	budget	genres	homepage	id	keywords	original_la	original_ti	overview	viewercou	productio	productio	release_da	revenue	runtime	spoken_la	status	tagline	title	vote_cour	vote_averag
2	2.5E+07	[{"id": 18,	http://ww	33870	[{"id": 432	en	Mao's Las	At the age	1.87681	[{"name":	[{"iso_316	#######	2.1E+07	117	[{"iso_639	Released	[illegible]	Mao's Las	28	6.8
3	3.8E+07	[{"id": 878, "name": '		193	[{"id": 109	en	Star Trek:	Captain Je	14.779	[{"name":	[{"iso_316	11/17/199	1.2E+08	118	[{"iso_639	Released	Boldly go.	Star Trek:	452	6.4
4	2E+07	[{"id": 36,	http://foc	10139	[{"id": 237	en	Milk	The story	30.9097	[{"name":	[{"iso_316	11/26/200	5.5E+07	128	[{"iso_639	Released	Never Ble	Milk	612	7.1
5	2.3E+07	[{"id": 18, "name": "D		11632	[{"id": 212	en	Vanity Fai	Beautiful,	6.61815	[{"name":	[{"iso_316	#######	1.6E+07	141	[{"iso_639	Released	On Septen	Vanity Fai	73	5.5
6	5.2E+07	[{"id": 28,	http://ww	26389	[{"id": 90,	en	From Paris	James Ree	27.9163	[{"name":	[{"iso_316	#######	5.3E+07	92	[{"iso_639	Released	Two agent	From Paris	675	6.1
7	2.8E+07	[{"id": 18,	http://ww	277216	[{"id": 380	en	Straight O	In 1987, fi	61.7623	[{"name":	[{"iso_316	8/13/2015	2E+08	147	[{"iso_639	Released	The Story	Straight O	1355	7.7
8	2.6E+07	[{"id": 80, "name": "C		14181	[{"id": 611	en	Boiler Roc	A college	11.2331	[{"name":	[{"iso_316	2/18/2000	2.9E+07	118	[{"iso_639	Released	Welcome	Boiler Roc	201	6.5
9	0	[{"id": 28, "name": "A		10413	[{"id": 156	en	Nowhere	Escaped c	11.6893	[{"name":	[{"iso_316	1/15/1993	0	94	[{"iso_639	Released	When the	Nowhere	119	5.5
10	4000000	[{"id": 28, "name": "A		2370	[{"id": 242	en	Topaz	A French i	5.9756	[{"name":	[{"iso_316	12/18/196	6000000	143	[{"iso_639	Released	Hitchcock	Topaz	77	6.1

Introduction

What can we say about the success of a movie before it is released? Are there certain companies (Pixar?) that have found a consistent formula? Given that major films costing over $100 million to produce can still flop, this question is more important than ever to the industry. Can we predict which films will be highly rated, whether or not they are a commercial success?

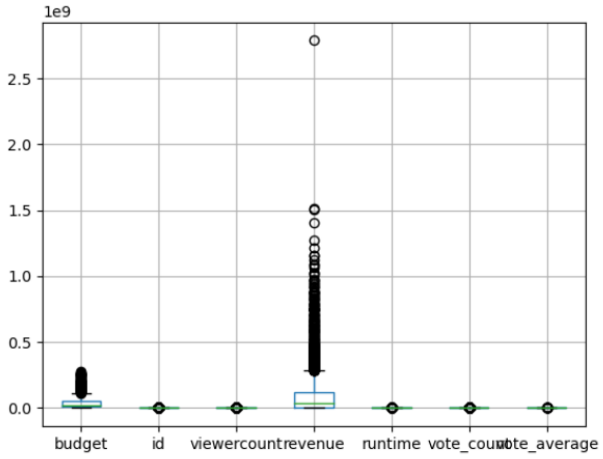
Using machine learning algorithms: Linear Regression and Polynomial Regression, we can predict which films will be highly rated.

Preprocessing Techniques:

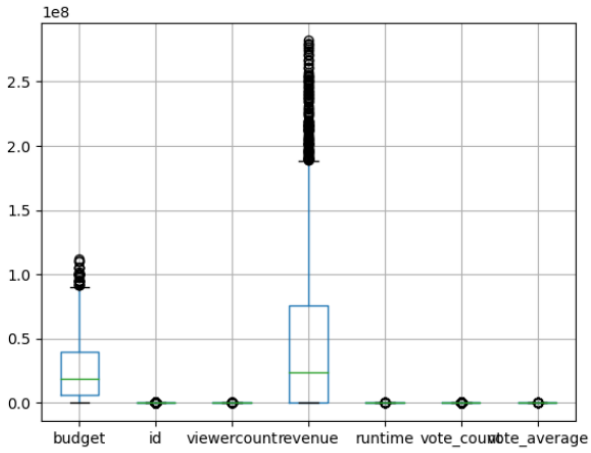
- ## Outliers Removal

We can improve the performance of machine learning models by removing the outliers. So, we defined a function called **outliers (data)** that takes the dataset as a parameter, it first calculates the first and third quartiles (Q1 and Q3) and the interquartile range (IQR). It then removes any values outside this range Q1 - 1.5 x IQR to Q3 + 1.5 x IQR.

Box Plot before removing the outliers.



Box Plot after removing the outliers.



- ## Extracting year from date column:

We extracted the year from the date column as the year will be more useful and significant in our project than month and day

- ## Encoding:

We used encoding to transform non-numeric labels to numeric labels

as in columns (genres - original_language - homepage ..etc)

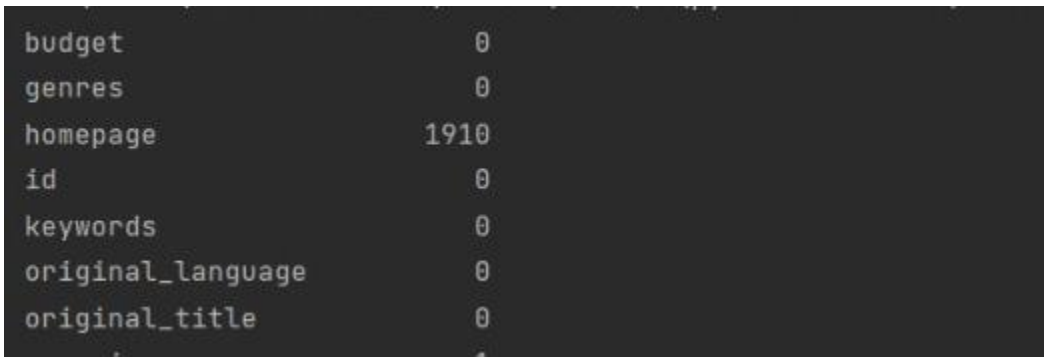
- ## Feature scaling:

We used this technique to re-scale a feature or observation value as the values of the features varies between columns

- ## Dropping columns:

After preprocessing we used the function DataFrame.isna().sum() to know which columns contain null

So, we dropped the column (homepage) as it contains too many nulls (1910 rows)



Then we dropped the column (status) as it will not affect the module because most of the columns have the same value

- ## Filling the missing values:

We filled the remaining columns that contain nulls with the mean value of the column by using the function:

data ["columnName"].fillna(X["columnName"].mean(),inplace=True)

- ## Feature selection:

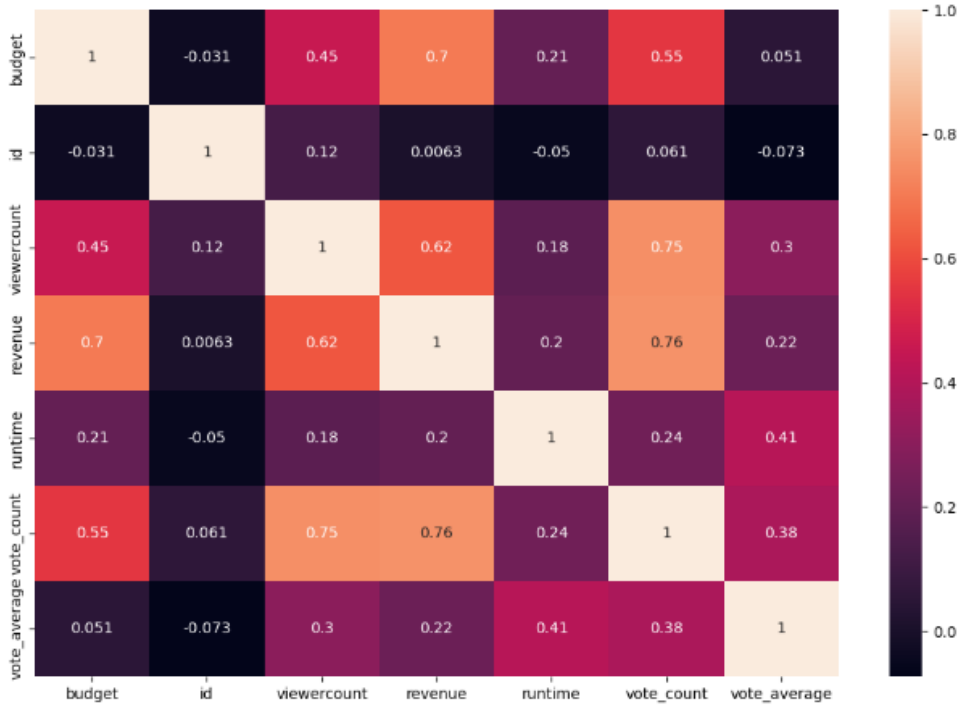
We used correlation to choose the top features that are highly correlated with Y and affects the module the most.

We applied SelectKBest with the f_classif scoring function and k=6 to select the top 6 features.

Data Set Analysis:

From this correlation Matrix, we can conclude that features that are highly correlated are:

Budget, id, viewercount, revenue, runtime, vote_count.



Regression Techniques:

We used 2 different models of regression: **Linear Regression** and **Polynomial Regression** to predict whether the film success or not.

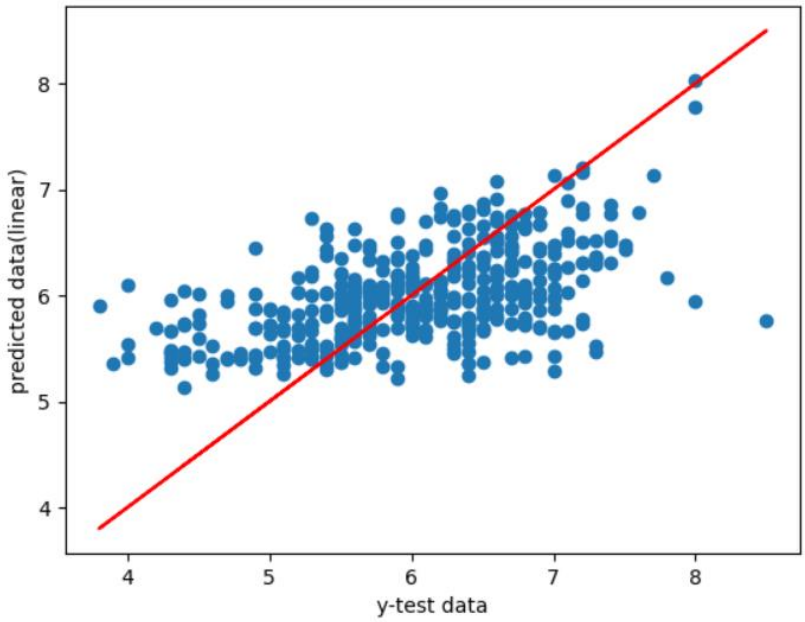
With **Linear Regression**:

We find that the Mean Square Error using the validation data is: 49%

, Mean Square Error using the test data is: 49%

, the Mean Absolute Error using the test data is: 55%

And R2_Score using the test data is: 0.26



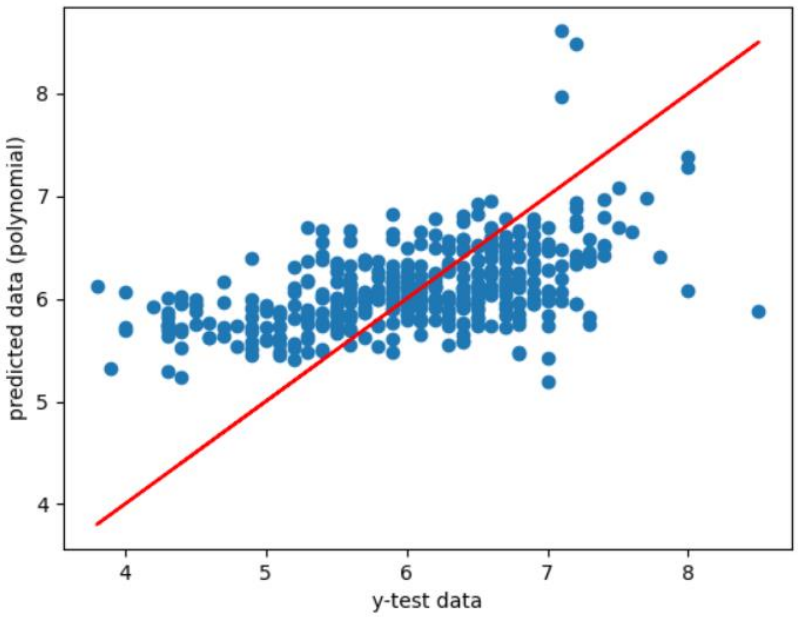
With **Polynomial Regression**:

We find that the Mean Square Error using the validation data is: 73%

, Mean Square Error using the test data is: 49%

, that the Mean Absolute Error using the test data is: 55%

And R2_Score using the test data is: 0.25



Data set Splitting:

It is important to apply the train-test split on a dataset before applying preprocessing because preprocessing techniques such as feature scaling, normalization, and encoding are typically learned from the training set and then applied to both the training and test sets.

If we apply these preprocessing techniques on the entire dataset before splitting it into training and testing sets, we risk introducing bias into our model. This is because the preprocessing methods will be applied to the test set as well, which means that the test set will contain information that was used to inform the model. As a result, the model's performance on the test set may be overly optimistic, and it may not generalize well to new, unseen data.

By applying the train-test split first, we ensure that the preprocessing methods are only learned from the training set, and then applied to the test set. This way, we can be confident that the model's performance on the test set is a fair representation of its ability to generalize to new, unseen data.

Using the train_test_split function from scikit-learn. We use a test size of 20% to split the data into a test set, and then we split the remaining data into a training set and a validation set using a test size of 25%.

Conclusion:

This phase of the project provides a good foundation for building a machine learning model to predict the vote average of a movie.