THE MOVIE DATABASE (TMDb) ANALYSIS AND COMPARISON OF PREDICTVE MODELS

A PROJECT REPORT

Submitted by:

Indrashis Paul (I9MIM10046)
Om Paras Rajani (19MIM10099)
Aniket Bandyopadhyay (19MIM10115)
Aman Jain (19MIM10089)

in partial fulfillment for the award of the degree of

INTEGRATED MASTERS OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Specialization in

Artificial Intelligence and Machine Learning

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH – 466114

OCTOBER 2020

VIT BHOPAL UNIVERSITY,KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled "**THE MOVIE DATABASE (TMDb) ANALYSIS AND COMPARISON OF PREDICTVE MODELS"** is the Bonafide work of "**Indrashis Paul (Register No: 19MIM10046), Om Paras Rajani (Register No: 19MIM10099), Aniket Bandyopadhyay (Register No: 19MIM10115) and Aman Jain (Register No: 19MIM10089)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR,
Dr. V. Pandimurugan, Assistant Professor
Integrated M.Tech, School of AI &ML division
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Mr. Ashish Kr. Sahu, Assistant Professor
School of AI &ML division
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on _____

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to **Dr Manas Kr. Mishra**, Head of the Department, School of Computing Science and Engineering, for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide **Mr. Ashish Kr. Sahu**, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not the least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

Abbreviation	Full Form
1. TMDb	The Movie Database

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this machine learning project, we cleaned, analyzed, and predicted two target variables – both revenue (numerical) and profitability (categorical), from the dataset of THE MOVIE DATABASE (TMDb).

Our aim is to explore the various Data Processing, Analysis and Regression and Classification Modeling techniques required for our Dataset to provide better predictions of the Revenue or the Profitability of a movie before its production.

The dataset contains around 5000 movies with 22 features and is obtained from Kaggle [1].

The information available about each movie include its budget, revenue generated, genres, rating, vote count, popularity, actors and actresses and any more. However, we used an unclean version of the dataset for our project.

In this project, we will use this dataset to **clean**, **analyze** and determine whether any information about a movie can **predict** the **total revenue** of a movie. We will then attempt to predict whether a movie's revenue will exceed its budget (profitability). Also, we will test **two different models** for each prediction to check which predicts our target variable better.

The results obtained from this project will be helpful for the Movie Production Teams to analyze the rubrics of their Movie Idea before it moves on to the Production Phase.

TABLE OF CONTENT

1. INTRODUCTION

1.1 Introduction

The movie industry has grown immensely over the past few decades generating approximately $10 billion of revenue for the stakeholders annually [2]. Nowadays, people can stream Movies online at the comfort of being at their home with the help of Netflix, YouTube and downloads. A Movie's gross revenue prediction is a very important problem in the film industry because it determines all the financial decisions made by producers and investors. Furthermore, a prediction system to assess the success of new movies with the help of the predicted revenue can help the movie producers and directors take proper decisions when making the movie in order to increase the chance of profitability and success. Usually, these types of predictions are made using basic statistical techniques that are described in [3]. While these methods are pretty common, they often only provide a very rough and not a specific estimate of revenue and profitability prediction before a film has been released. The goal of this project is to analyze the data and find the best computational model through evaluating metrics for predicting revenues and profitability rates of a movie based on public data for movies extracted from a popular online movie database called The Movie Database (TMDb) [1].

1.2 Problem Statement and Motivation for the work

Various Websites other than TMDb are present out there providing Movie Trend Analysis and most of them fail to provide proper Movie analysis. This is mostly due to the presence of various kinds of noise in their data.

Furthermore, when a Movie is to be produced, the directors and producers require an estimate of the financial expenditure and the overall success of the movies so that they can proceed with a specific pipeline to follow. But many well-financed movies fail because these estimations were not taken into account and it turned out to be unsuccessful [2].
So, these issues faced by both Website Administrators and Movie Producers motivated us to decide the use an unclean dataset for our project to demonstrate the various Data Wrangling techniques in order to perform proper Analysis and then explore few predictive models and compare them for the sake of improving the accuracy of the prediction [5].

1.3 Objective of the work

In this paper, as per the problem statement discussed above, we have enlisted our objectives in doing this project as follows:

- To showcase a series of Data Wrangling techniques needed in general for cleaning, preparing and transforming a raw Movie Box Office data from the source to a Proper Data ready and suitable for Analytical Methods to be implemented on them.

- To showcase various Univariate, Bivariate and Multivariate Analysis Techniques to be used on the transformed data of Movie Box Office and generate accurate visualisations and insights on the trend of movies and films throughout the past century.

- To showcase the superiority of Ensemble and Tree based Supervised Predictive models like Random Forest over basic Statistical models like Linear and Logistic Regression while Predicting the Success of Movies though Revenue Prediction.

1.6 Organization of the thesis

This dissertation shows how a realistic Data Wrangling, Exploration and Modelling on a Movie Trend Database can help Film Producers and Directors to understand the characteristics of the explored predictions and take curated decisions regarding the production of movies.
The organization of this thesis is as follows:
Chapter 2 introduces and elaborates on the literary details from where we extracted the theoretical basis of our model pipeline. We first brief about the preceding researches done on similar topics and the result they obtained from their detailed analysis of those specific domains. The we describe all the existing algorithms and techniques that we have incorporated in our data pipeline. The last section in Chapter 2 is focusing on the limitations faced by those models and hypotheses and needs to be addressed.
Chapter 3 and 4 introduces our Pipeline in detail. In chapter 3, we first describe the limitations of the existing model and provide an analysis of our proposed pipeline. In chapter 4, we elaborate on the implementation of our proposed pipeline with the help of Insightful Outputs and Obtained Graphs assisting the workflow explanation.
Chapter 5 explores the results our pipeline obtained with detailed descriptions of the evaluation metrics used and the graphical representation of our results.
Chapter 6 throws light into the limitations and shortcomings of our pipeline and enumerates various future enhancements to be done on the model workflow and finally concludes the thesis.
Last but not the least, Appendix A contains the full description of all benchmark methodologies of our Pipeline.

1.7 Summary

This section introduces the topic, objectives and the organization of the project workflow and exactly what we have tried to achieve through our project. It also contains brief descriptions of the following chapters.

1. LITERATURE REVIEW

There has been work done related to certain projects which have referenced the analysis of movie-based data sets and made not only predictive models but analyzed these data sets and made a complete analytical model based on their knowledge of Artificial Intelligence & Machine Learning.

Ibtesam Ahmed in May 2020 made a model relating to a **Movie Recommendation System** using data from multiple sources. Recommendation System is a type of information filtering systems as they improve the quality of search results and provides items that are more relevant to the search item.~(Kaggle ID:Ibtesama)

GSD in 2017 worked on a model and Explored the data set taken from **TMDb Database** which consisted of movies and in the model calculated what highest popularity as well as highest budget movies were based on the algorithms and calculation only after cleaning the data set and made an elaborate analysis based on it.~(Kaggle ID:gsdeeppakkumar)

Yueming Zhang in 2016 worked on a **Data Mining Algorithm** to analyze which movies are successful and made a **Predictive Model.** Taking IMDB scores as response variable and focus on operating predictions by analyzing the rest of variables in the IMDB 5000 movie data. The results can help film companies to understand the secret of generating a commercial success movie.~(Kaggle ID:carolezhangdc)

Simonoff ,J. S. and Sparrow, I. R. in 2000 made a model of predicting movie grosses. They took data from the Internet Movie Database (www.imdb.com). Movies that opened on a limited time in 1997 and then released globally in 1998 are included in the sample as well. This yields a total of 311 films. A shortened version of their research paper appeared in *Chance* , 13(3), 15-24 (Summer 2000).

A research by Timothy King about the effects of criticism affecting box office ratings in 2003. In his findings he found out that two of the highest grossing movies(The Lord of The Rings: Return of The King and Finding Nemo) also received the highest score of any movie released in 2003. His analysis made use of **Statistical Graphs and Representation** to easily grasp his research. The research paper is available at research gate under the title “Does Film Criticism Affect Box Office Ratings”.

A. Chen used **Regression** concepts in his Report and proved Predicting the box-office revenue of a movie before its theatrical release is an important but challenging problem that requires a high level of Artificial Intelligence. Social media has shown its predictive power in various domains, which motivates us to exploit social media content to predict box-office revenues. He employed both Linear and Logistic Regression. ~University of Washington, Seattle, June 2002.

2. PIPELINE DESIGN AND IMPLEMENTATION

3.1 Introduction

Dataset:

The movie dataset on which this case study is based is a database of 5000 movies catalogued by **The Movie Database (TMDb).** This dataset was generated from The Movie Database API. This product uses the TMDb API but is not endorsed or certified by TMDb.

The dataset contains around 5000 movies with 22 features and is obtained from Kaggle [1]. The information available about each movie include data on the plot, cast, crew, budget, revenues, genres, rating, vote count, popularity, and any more. However, we used an unclean version of the dataset for our project which was curated by edX Course creators to be used as an asset for various projects:

https://courses.edx.org/assetv1:HarvardX+PH526x+2T2019+type@asset+block@movie_data.csv

The Case Study:

In this case study, we will use this dataset to **analyze** and determine whether any information about a movie can **predict** the **total revenue** of a movie. We will also attempt to predict whether a movie's revenue will exceed its budget (profitability). Also, we will test **two different models** (mentioned previously) for each prediction to check which predicts our target variable better.

3.2 Proposed Pipeline

The total module Walkthrough is briefed in the following parts:

1. Data Pre-processing: Since we used an unclean dataset, we filtered, cleaned and transformed the data into a usable one. This helped us a lot in making proper Analysis and Predictions.

2. Exploratory Data Analysis: We explored and analysed the dataset and obtain valuable insights on it to determine the importance of each variables in predicting the target variables.

3. Comparison of Predictive Models: We used two models – Basic Statistical Model and the Random Forest Model for both – predicting the numerical feature (revenue) which is a regression task and predicting the categorical feature (profitability) which is a classification task. We also compared the performance of these models using Correlation metrics for Regression and Accuracy metrics for Classification through 10-fold Cross-Validation and determined the most important features for each of the prediction tasks.

3.3 Module 1 – Data Pre-processing

Data preprocessing is an important step in the data mining process. Data-gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income: −100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), and missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and

quality of data is first and foremost before running any analysis. Often, data preprocessing is the most important phase of a machine learning project, especially in computational biology.[6]

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data preprocessing includes cleaning, Instance selection, normalization, transformation, feature extraction and selection, etc. The product of data preprocessing is the final training set.

Data pre-processing may affect the way in which outcomes of the final data processing can be interpreted. This aspect should be carefully considered when interpretation of the results is a key point, such in the multivariate processing of chemical data (chemometrics).

We worked out the following pre-processes in our pipeline:

- Importing Libraries: Here, we imported the required dependencies and libraries that are required for the various techniques we were going to use in our Case Study. The libraries include – pandas, NumPy, matplotlib, seaborn, and various tools from scikit-learn. scikit-learn (**sklearn**) contains helpful statistical models, and we've used the matplotlib.pyplot and seaborn library for visualizations of the determined analysis of data. Of course, we used NumPy and pandas for data representation and manipulation throughout.
- Reading Dataset: Here, we use our previously imported pandas library to read the dataset present as a comma separated values file and convert it into a pandas dataframe to view and represent it in memory in a better format. We used pandas.read_csv() function to implement this and then the dataframe.head() function to view the first 5 elements of our dataset.

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies	production_countries	release_date	revenue	run
0	237000000	Action, Adventure, Fantasy, Science Fiction	http://www.avatarmovie.com/	19995	culture clash, future, space war, space colony...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	Ingenious Film Partners, Twentieth Century Fox...	United States of America, United Kingdom	2009-12-10	2787965087	
1	300000000	Adventure, Fantasy, Action	http://disney.go.com/disneypictures/pirates/	285	ocean, drug abuse, exotic island, east india t...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	Walt Disney Pictures, Jerry Bruckheimer Films,...	United States of America	2007-05-19	961000000	
2	245000000	Action, Adventure, Crime	http://www.sonypictures.com/movies/spectre/	206647	spy, based on novel, secret agent, sequel, mi6...	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	Columbia Pictures, Danjaq, B24	United Kingdom, United States of America	2015-10-26	880674609	
3	250000000	Action, Crime, Drama, Thriller	http://www.thedarkknightrises.com/	49026	dc comics, crime fighter, terrorist, secret id...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	Legendary Pictures, Warner Bros., DC Entertain...	United States of America	2012-07-16	1084939099	
4	260000000	Action, Adventure, Science Fiction	http://movies.disney.com/john-carter	49529	based on novel, mars, medallion, space travel,...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	Walt Disney Pictures	United States of America	2012-03-07	284139100	

Fig 3.3.1: A preview of our dataset

- Defining Target Variables: Now, we defined the regression and classification outcomes. Specifically, we used the **revenue** column as the target for regression. For classification, we have constructed an indicator of **profitability** for each movie. Steps:
    1. We created a new column in our DataFrame called **profitable**, defined as 1 if the movie **revenue** is greater than the movie **budget**, and 0 otherwise.
    2. Next, we defined and stored the outcomes we will use for regression and classification as such

- **regression_target** as the string 'revenue'.
- **classification_target** as the string 'profitable'.

- Removing null, unwanted or infinite values: For simplicity, we will proceed by analysing only the rows without any missing data. Here, we will first remove all unimportant features to reduce high data dimensionality and then remove all the rows with any infinite or missing values.
  Steps:
  1. We check the number of null values or numpy.nan values by using DataFrame.isnull().sum().
  2. We use DataFrame.drop() to remove any columns that are unimportant.
  3. We check if the important columns contain value = 0 and replace those values of the budget column with numpy.nan since the budget of a movie cannot be 0 even if the revenue might be.
  4. We create a column which shows the return gained or lost from the movie production.
  5. We use DataFrame.replace() to replace any cells from the resultant dataset with type numpy.inf or -numpy.inf with numpy.nan.
  6. We drop all rows with any numpy.nan values in that row using DataFrame.dropna().
  7. We use the inplace=True argument to immediately reflect the changes in the current dataset.

BEFORE

```
budget                     0
genres                    28
homepage                3091
id                         0
keywords                 412
original_language          0
original_title             0
overview                   3
popularity                 0
production_companies     351
production_countries     174
release_date               1
revenue                    0
runtime                    2
spoken_languages          87
status                     0
tagline                  844
title                      0
vote_average               0
vote_count                 0
movie_id                   0
cast                      43
profitability              0
dtype: int64
```

Fig 3.3.2 Before Cleaning

AFTER

```
budget                0
genres                0
id                    0
keywords              0
original_language     0
overview              0
popularity            0
release_date          0
revenue               0
runtime               0
status                0
tagline               0
title                 0
vote_average          0
vote_count            0
movie_id              0
cast                  0
profitable            0
return                0
dtype: int64
```

Fig 3.3.3 After Cleaning

- Feature Engineering:
  1. Encoding of Genres Column: Many of the variables in our dataframe contain the names of genre, actors/actresses, and keywords. Here, we add indicator columns for each genre.
     Steps
     - We determine all the genres in the genre column and make use of the list.strip() function on each genre to remove trailing characters.

- Next, we include each listed genre as a new column in the dataframe. Each element of these genre columns is 1 if the movie belongs to that particular genre, and 0 otherwise, keeping in mind, a movie may belong to several genres at once.
- This process is also called as One-Hot Encoding but here it is applied on each data of each row of the Column.

	Action	Adventure	Fantasy	Science Fiction	Crime	Drama	Thriller	Animation	Family	Western	Comedy	Romance	Horror	Mystery	History	War	Mus
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
3	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	
4	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4788	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	
4791	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
4792	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	
4796	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	
4798	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	

3266 rows × 20 columns

Fig 3.3.4 The encoded Genres column

2 Creating the Year column for analysis: Some variables in the dataset can be used to create new and important columns which can be further used for better analysis of the dataset.
Steps
- Create a new column, year where we extract and store the year of production of a movie from the given format of release_date.
- After creating the column, we extract the year from the release_date using the pandas.to_datetime function.

```
0    2009-12-10
1    2007-05-19
2    2015-10-26
3    2012-07-16
4    2012-03-07
Name: release_date, dtype: object
```

Fig 3.3.5 The "release_date" Column

- Feature Selection: Some variables in the dataset are already numeric and perhaps useful for regression and classification. Here, we will store the names of these variables for future use. We will also take a look at some of the continuous variables and outcomes by plotting each pair in a scatter plot. Finally, we will evaluate the skew of each variable.
Steps
1 We call plt.show() to observe the plot shown below to find which of the covariates and/or outcomes are correlated with each other.
2 We call skew() on the columns outcomes_and_continuous_covariates in the dataset and check the features for which the skew is above 1.

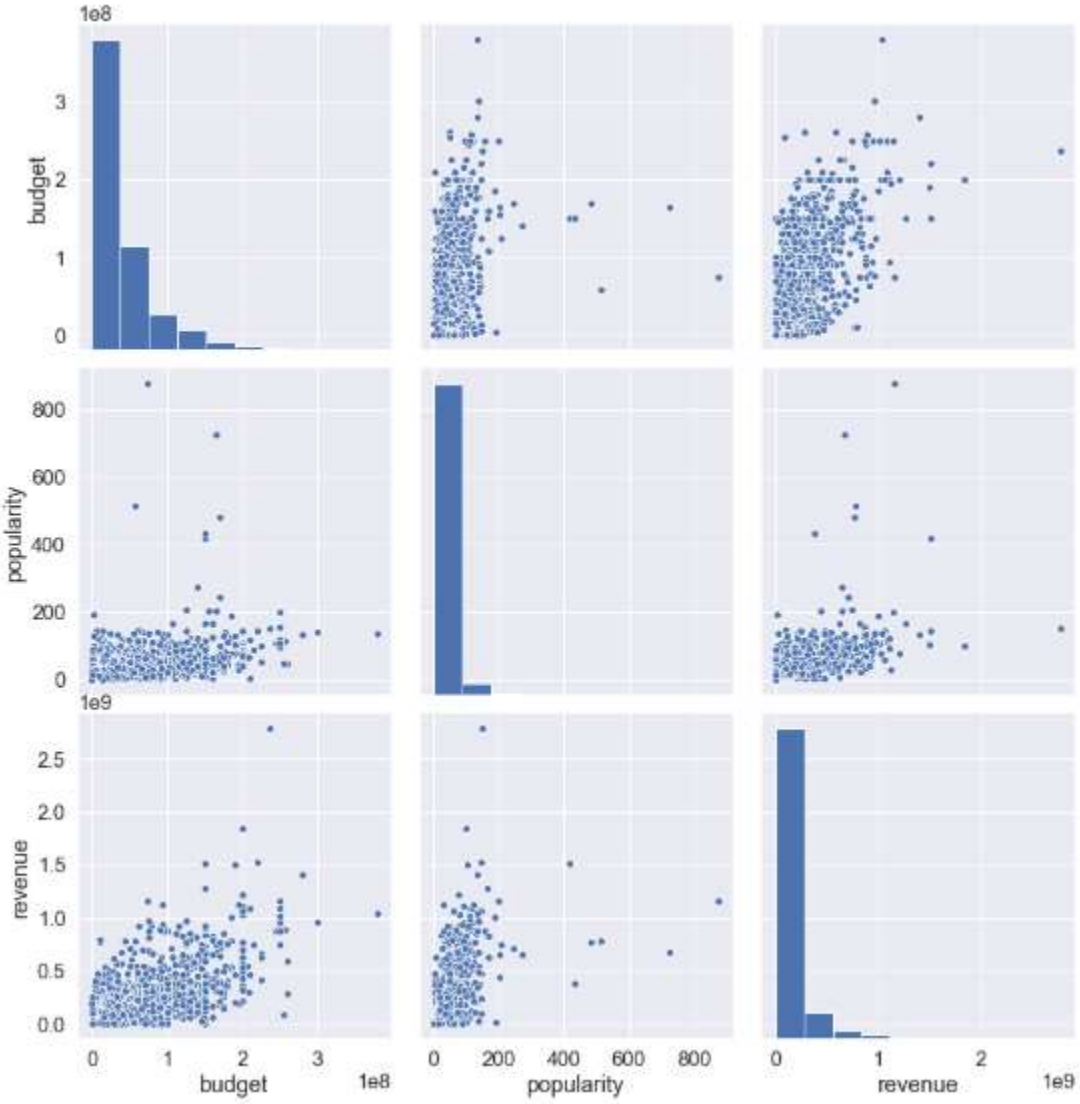

Fig 3.3.6 Distribution plot of numerical covariates
showing skewness of data

- Feature Transformation: Here, we will transform these variables to eliminate this skewness. Specifically, we will use the numpy.log10() method. Because some of these variable values are exactly 0, we will add a small positive value to each to ensure it is defined; this is necessary because log(0) is negative infinity. Steps
    - For each above-mentioned variable in the dataset, we will transform the value x into numpy.log10(1+x).

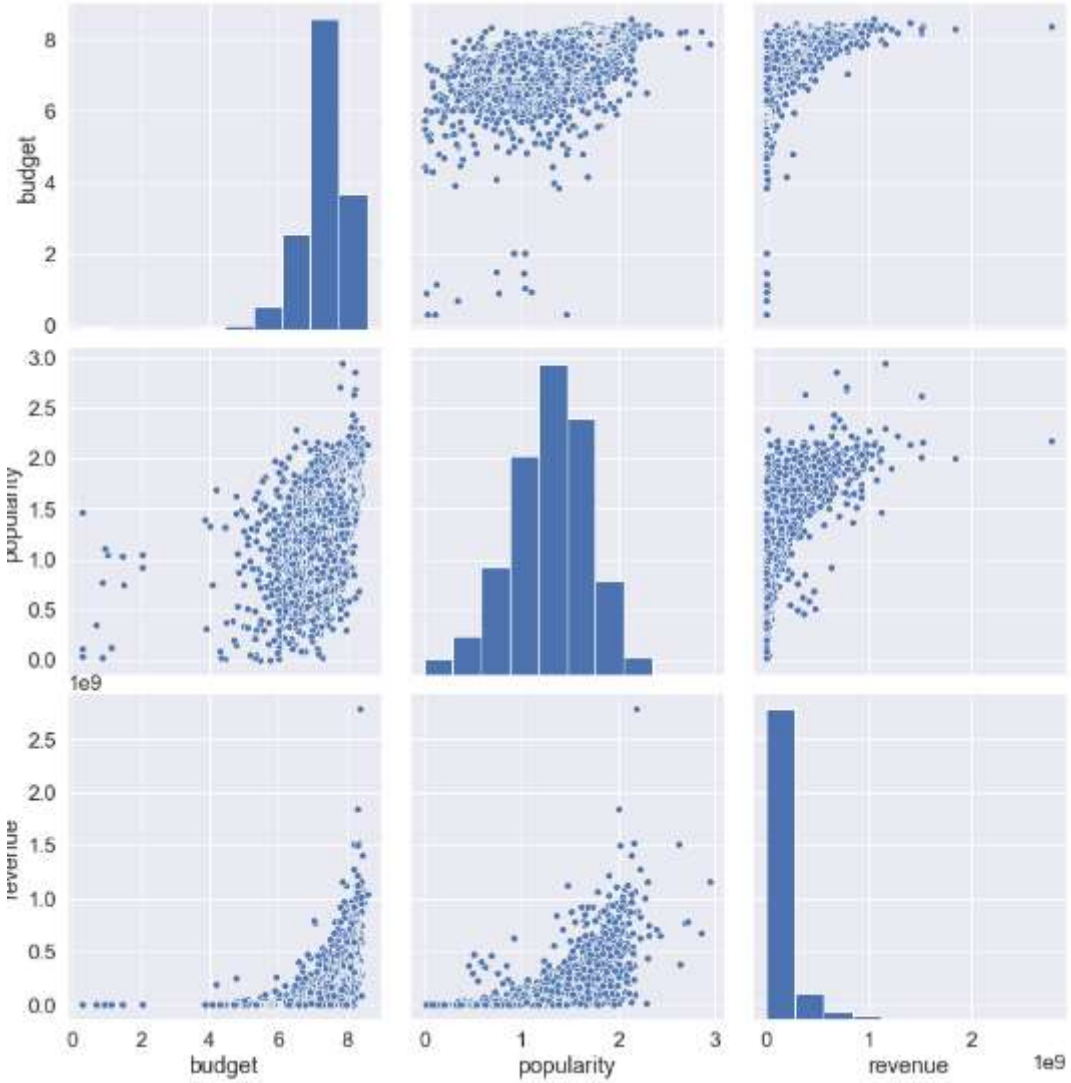

Fig 3.3.7 Distribution plot of the corrected numerical covariates showing rectified skewness of transformed data

- Storing the Transformed Dataset: We will now save our dataset.
  Steps
    - We use DataFrame.to_csv() to save the DataFrame object as "movies_clean.csv".

	budget	genres	id	keywords	original_language	overview	popularity	release_date	revenue	runtime	status	tagline	title	vote_average	vote_count	movie_id	cast	profitable	return	Action	Adventure
0	8.374748	Action, Adventure, Fantasy, Science Fiction	19995	culture clash, future, space war, space colony...	en	In the 22nd century, a paraplegic Marine is di...	2.180234	2009-12-10	9.445287	2.212188	Released	Enter the World of Pandora.	Avatar	7.2	4.071919	19995	Sam Worthington, Zoe Saldana, Sigourney Weaver...	1	11.763566	1	1
1	8.477121	Adventure, Fantasy, Action	285	ocean, drug abuse, exotic island, east india t...	en	Captain Barbossa, long believed to be dead, ha...	2.146384	2007-05-19	8.982723	2.230449	Released	At the end of the world, the adventure begins.	Pirates of the Caribbean: At World's End	6.9	3.653309	285	Johnny Depp, Orlando Bloom, Keira Knightley, S...	1	3.203333	1	1
2	8.389166	Action, Adventure, Crime	206647	spy, based on novel, secret agent, sequel, mi6...	en	A cryptic message from Bond's past sends him o...	2.034936	2015-10-26	8.944815	2.173186	Released	A Plan No One Escapes	Spectre	6.3	3.650016	206647	Daniel Craig, Christoph Waltz, Léa Seydoux, Ra...	1	3.594590	1	1
3	8.397940	Action, Crime, Drama, Thriller	49026	dc comics, crime fighter, terrorist, secret id...	en	Following the death of District Attorney Harve...	2.054280	2012-07-16	9.035405	2.220108	Released	The Legend Ends	The Dark Knight Rises	7.6	3.959375	49026	Christian Bale, Michael Caine, Gary Oldman, An...	1	4.339756	1	0
4	8.414973	Action, Adventure, Science Fiction	49529	based on novel, mars, medallion, space travel,...	en	John Carter is a war-weary, former military ca...	1.652507	2012-03-07	8.453531	2.123852	Released	Lost in our world, found in another.	John Carter	6.1	3.327359	49529	Taylor Kitsch, Lynn Collins, Samantha Morton, ...	1	1.092843	1	1

Fig 3.3.8 The Transformed Dataset

3.4 Module 2 – Exploratory Data Analysis

In statistics, exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task. Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed.
Tukey defined data analysis in 1961 as:

"Procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data."

We worked out the following analysis in our pipeline:

- Statistical Analysis - Analysing all the features form a Statistical point of view

	budget	id	popularity	revenue	runtime	vote_average	vote_count	movie_id	profitable
count	3266.000000	3266.000000	3266.000000	3266.000000	3266.000000	3266.000000	3266.000000	3266.000000	3266.000000
mean	7.278334	45488.042254	1.292131	6.855850	2.038060	6.271341	2.604175	45488.042254	0.694121
std	0.740971	77053.912258	0.405928	2.486905	0.082333	0.894677	0.631083	77053.912258	0.460849
min	0.301030	5.000000	0.000688	0.000000	0.000000	0.000000	0.000000	5.000000	0.000000
25%	7.000000	5307.500000	1.041590	7.030228	1.986772	5.700000	2.204120	5307.500000	0.000000
50%	7.397940	11317.000000	1.317127	7.695402	2.029384	6.300000	2.645913	11317.000000	1.000000
75%	7.740363	44698.250000	1.575985	8.146561	2.082785	6.900000	3.042477	44698.250000	1.000000
max	8.579784	417859.000000	2.942792	9.445287	2.530200	8.500000	4.138397	417859.000000	1.000000

Fig 3.4.1 Descriptive Statistical Analysis of the dataset

Observations:

- It shows the count of all the features as 3266 confirming that the data present in the dataset is constant and doesn't contain any discrepancy.
- It also shows the mean of all the features and we can clearly see that all the numerical features have been normalized to their log10 counterparts.
- It also shows us the maximum and minimum values present in each feature.

- Correlation Analysis - Plotting a heatmap for the correlation of all the features of our Dataset
  Observations:
  - Most of the features have Pearson's Correlation Coefficient between 0.3 and 0.7 meaning that they have some relation between them.
  - return feature is solely created for visualization and so is irrelevant to all the features.

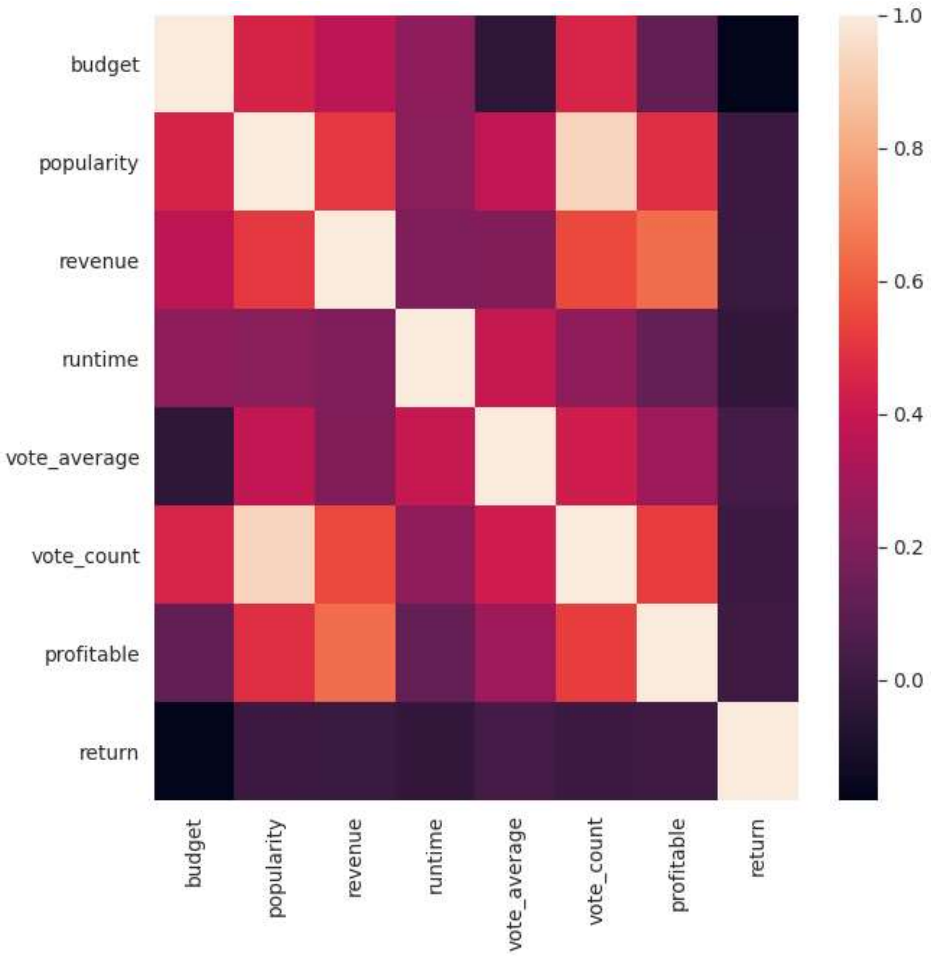

Fig 3.4.2 Correlation Heatmap of the numerical features of the dataset

- Regression Target Analysis –
  1. Plotting the correlation graph between revenue and budget
     Observations:
     - We can definitely see strong correlation between budget and revenue
     - Quite a few numbers of outliers are present
     - The relation is mostly linear

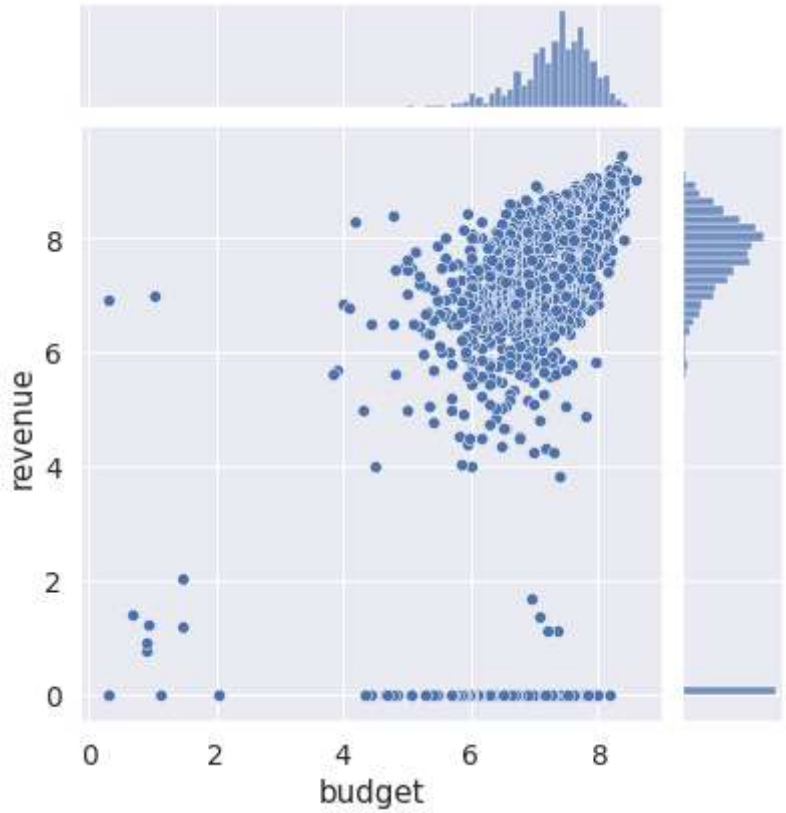

Fig 3.4.3 Correlation graph – revenue vs budget

2 Plotting the correlation graph between revenue and popularity
Observations:

- Again, as expected, we see some correlation between revenue and popularity
- Here, too, quite a few numbers of outliers are present
- The relation is mostly linear

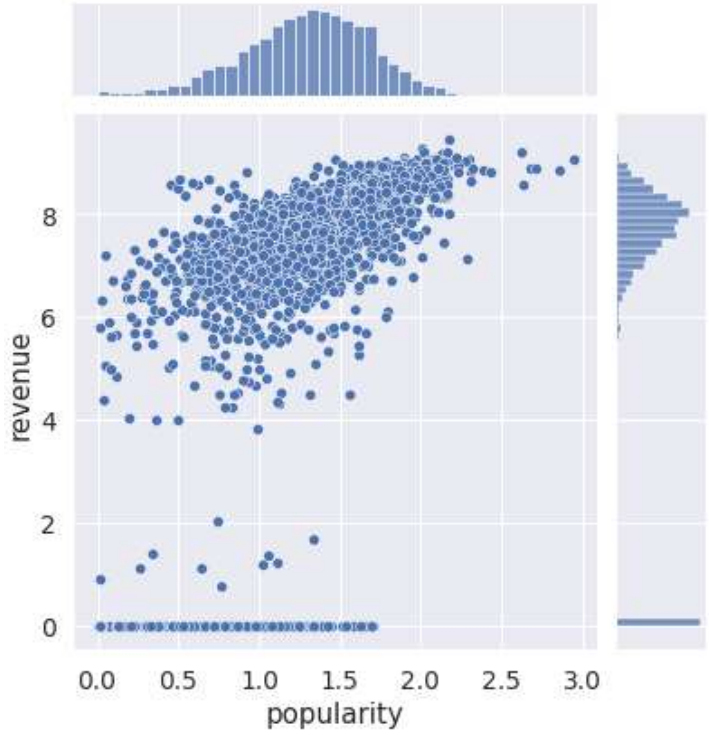

Fig 3.4.4 Correlation graph – revenue and popularity

3 Plotting the correlation graph between revenue and vote_count
Observations:
- Strong correlation between **revenue** and **vote_count** is observed.
- The movies that have high revenue necessarily has a high vote_count.
- There are outliers present, although not too many.
- The relation is strongly **linear**.

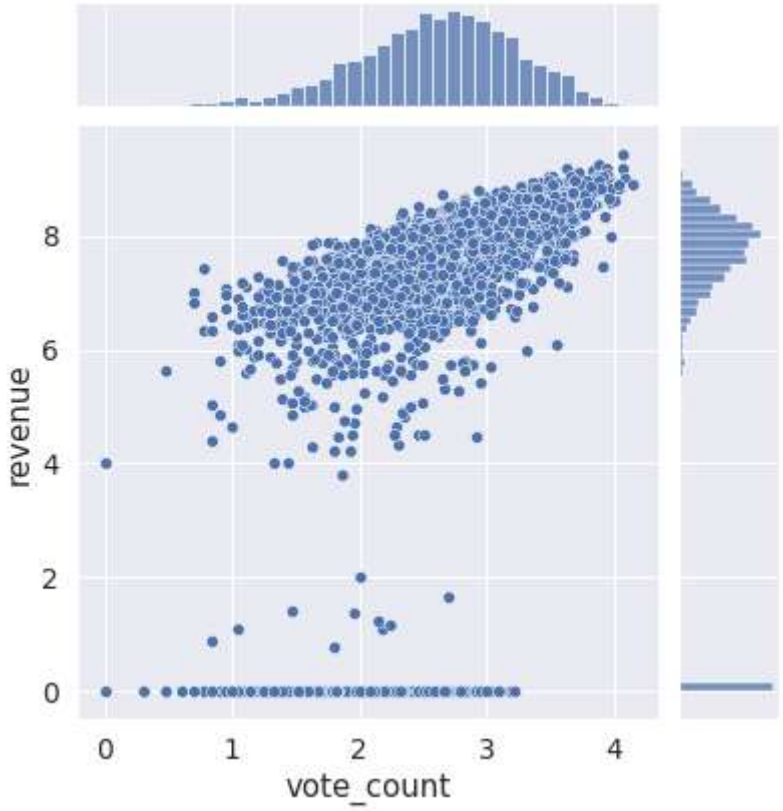

Fig 3.4.5 Correlation graph – revenue and vote_count

- Analysis of Movie Trends over the Years –
  1 Sorting out the title, vote_count and year to check the movies with highest vote_counts ever.

	title	vote_count	year	revenue	return
96	Inception	4.138397	2010	8.916734	5.159580
65	The Dark Knight	4.079290	2008	9.001975	5.430046
0	Avatar	4.071919	2009	9.445287	11.763566
16	The Avengers	4.071035	2012	9.181717	6.907081
788	Deadpool	4.041235	2016	8.893824	13.501948
95	Interstellar	4.036150	2014	8.829381	4.091636
287	Django Unchained	4.004321	2012	8.628765	4.253682
94	Guardians of the Galaxy	3.988693	2014	8.888364	4.548992
426	The Hunger Games	3.975707	2012	8.839610	9.216143
127	Mad Max: Fury Road	3.974420	2015	8.578477	2.525722

Fig 3.4.6 View of the sorted-out Movies

2. Plotting High-revenue (>= 1 billion) and high-vote_count movies according to their release year

   Observations:

   - After Titanic (1997) which was the first movie to reach 1 billion dollar mark, 2010, 2011, 2012, 2013 and 2015 had many movies crossing the billion dollar mark significantly.
   - There has been an increase in the revenue of movies filmed till 2012 which showed the peak revenue generated, and although it decreased sharply for 2013 and 2014 yet, it again increased in 2015

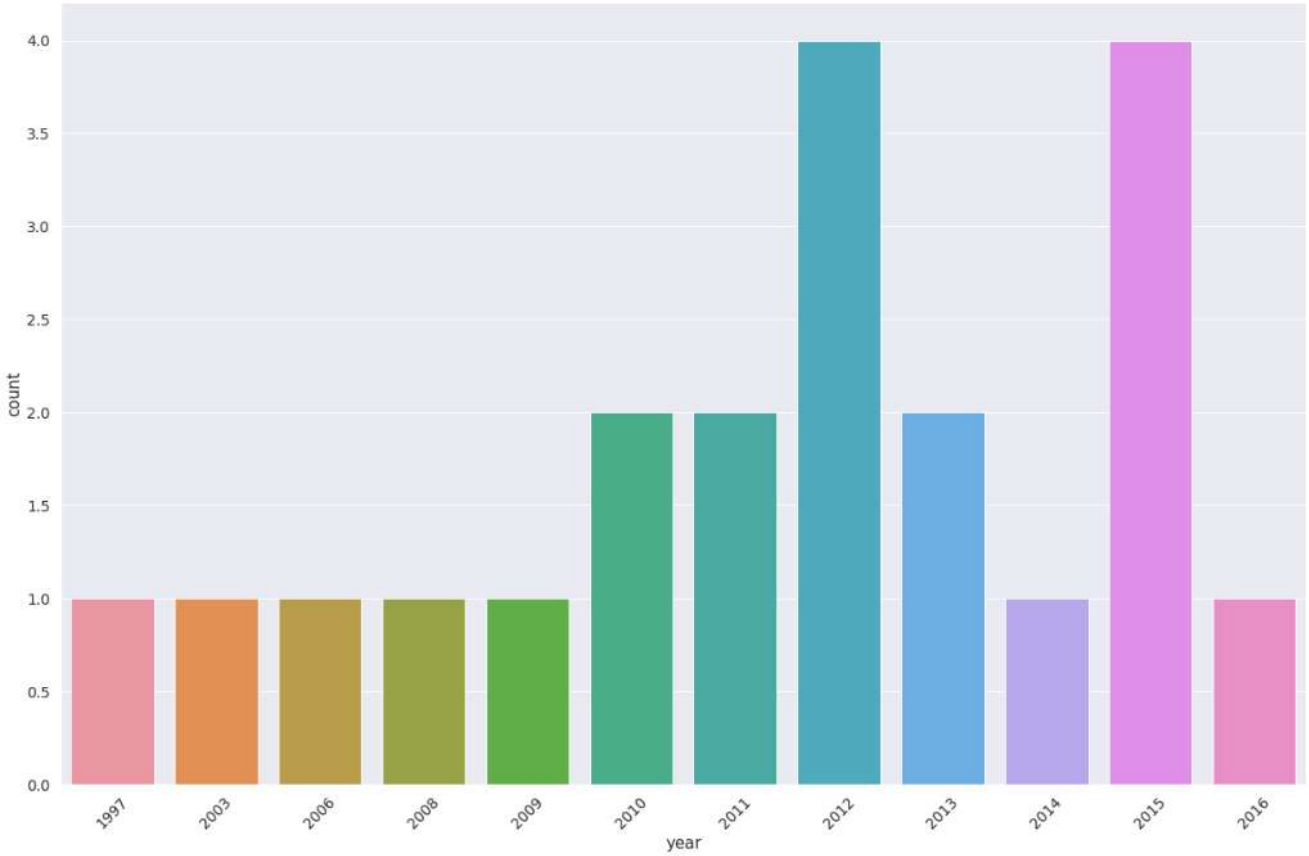

Fig 3.4.7 Trend of the revenue of movies over 1 billion mark over the years

- Trend Analysis of vote-average of movies –
  1. Extracting and printing the sorted sub-dataset of title, vote_average, vote_count and year of the movies for which vote_count is greater than 3000

	title	vote_average	vote_count	year
1881	The Shawshank Redemption	8.5	3.914132	1994
3337	The Godfather	8.4	3.770410	1972
3865	Whiplash	8.3	3.628900	2014
2294	Spirited Away	8.3	3.584444	2001
1818	Schindler's List	8.3	3.636488	1993
3232	Pulp Fiction	8.3	3.925776	1994
662	Fight Club	8.3	3.973774	1999
2731	The Godfather: Part II	8.3	3.523616	1974
809	Forrest Gump	8.2	3.899164	1994
690	The Green Mile	8.2	3.607348	1999

Fig 3.4.8 Sorted sub-dataset of Movies as per their vote_average

Observations:

- These are the 1st 10 movies in rank considering the vote_average feature
- We observe that The Shawshank Redemption (1994) is the movie with the highest vote_average ever.

2 Plotting the correlation graph between vote_average and popularity

Observations:

- Some correlation between vote_average and popularity are observed.
- The movies that have high vote_average doesn't necessarily have to be highly popular.
- There are outliers present, although not too many.
- The relation is mostly linear.

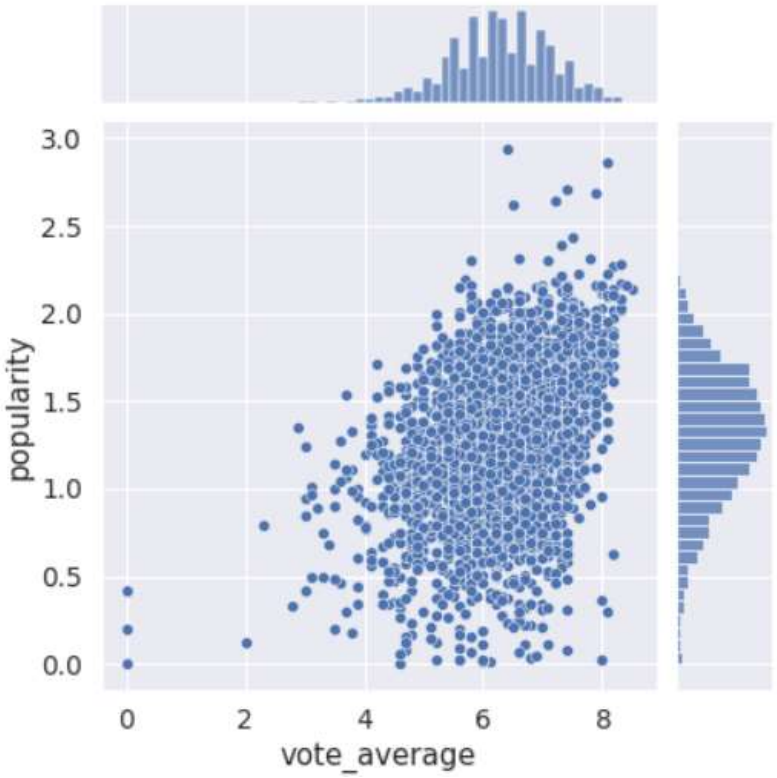

Fig 3.4.9 Correlation graph – popularity vs vote_average

- Genre Feature Analysis –
    1. Extracting the number of each genre used throughout the total dataset
    2. Creating a new dataframe to store the genres and their count in a sorted order
    3. Plotting the first 15 genres and their movie count

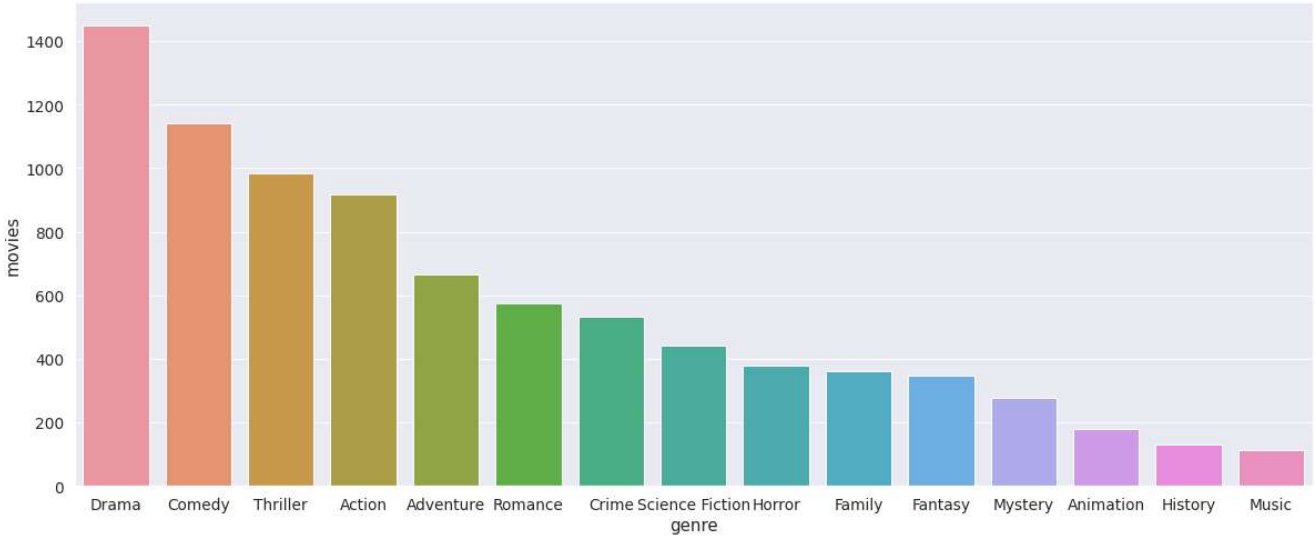

Fig 3.4.10 Genre Trend Analysis

Observations:

- Drama is the most used genre in the Film Industry.
- Comedy and Thriller genres follow after it.
- Foreign and TV Movie genres has the lowest movie count (excluded in the graph)

- General Movie Trend Analysis –
    1. Checking the most expensive movie ever made

	title	budget	revenue	return	year
17	Pirates of the Caribbean: On Stranger Tides	8.579784	9.019413	2.751878	2011
1	Pirates of the Caribbean: At World's End	8.477121	8.982723	3.203333	2007
7	Avengers: Age of Ultron	8.447158	9.147801	5.019299	2015
4	John Carter	8.414973	8.453531	1.092843	2012
6	Tangled	8.414973	8.772171	2.276134	2010
5	Spider-Man 3	8.411620	8.949815	3.452991	2007
13	The Lone Ranger	8.406540	7.950802	0.350157	2013
19	The Hobbit: The Battle of the Five Armies	8.397940	8.980467	3.824079	2014
3	The Dark Knight Rises	8.397940	9.035405	4.339756	2012
8	Harry Potter and the Half-Blood Prince	8.397940	8.970328	3.735837	2009

Fig 3.4.11 Movie Sub-dataset sorted as per decreasing from highest budget

Observations:

- Pirates of the Caribbean: On Stranger Tides, the Johnny Depp starrer is the most expensive movie ever made! The budget was 3.8 Billion which is still the highest!

2 Checking the highest grossing movies of all time

	title	budget	revenue	year
0	Avatar	8.374748	9.445287	2009
25	Titanic	8.301030	9.266004	1997
16	The Avengers	8.342423	9.181717	2012
28	Jurassic World	8.176091	9.179991	2015
44	Furious 7	8.278754	9.177897	2015
7	Avengers: Age of Ultron	8.447158	9.147801	2015
124	Frozen	8.176091	9.105244	2013
31	Iron Man 3	8.301030	9.084734	2013
546	Minions	7.869232	9.063232	2015
26	Captain America: Civil War	8.397940	9.061944	2016

Fig 3.4.12 Movie sub-dataset for highest grossing movies

Observations:

- The mighty Avatar was the highest grossing movie of all time, since it generated the highest revenue ever, which was surpassed recently by The Avengers: End Game

3 Checking the trend of revenue over the years

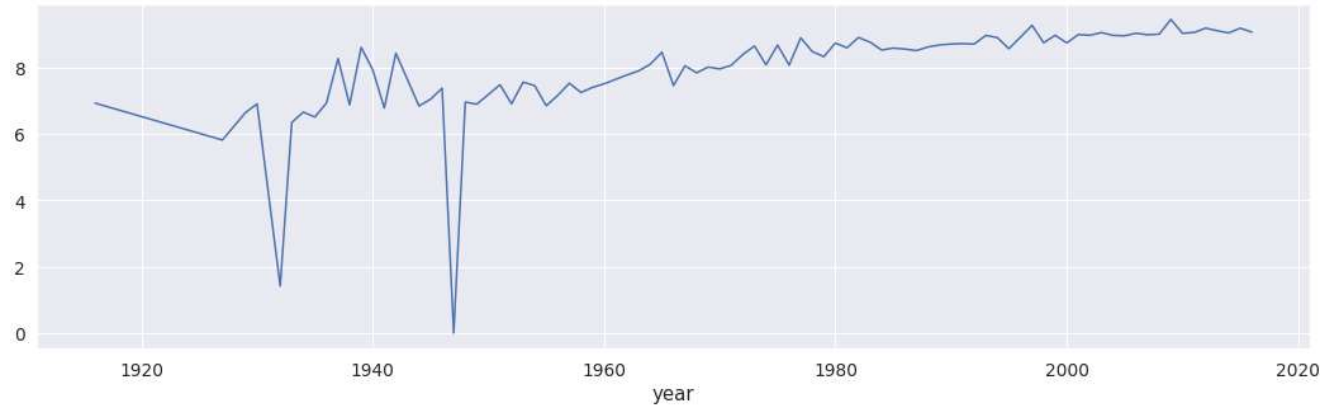

Fig 3.4.13 Movie Trend Analysis over the Years

Observations:

- We can see that the revenue has been steadily increasing over the years
  Revenue Distribution over the years

3.5 Module 3 - Predictive and Comparison Modelling

Predictive modeling uses statistics to predict outcomes. Most often the event one wants to predict is in the future, but predictive modelling can be applied to any type of unknown event, regardless of when it occurred. For example, predictive models are often used to detect crimes and identify suspects, after the crime has taken place.

In many cases the model is chosen on the basis of detection theory to try to guess the probability of an outcome given a set amount of input data, for example given an email determining how likely that it is spam.

Models can use one or more classifiers in trying to determine the probability of a set of data belonging to another set. For example, a model might be used to determine whether an email is spam or "ham" (non-spam).

There are various types of predictive models available but they are broadly classified as – supervised and unsupervised. We are using supervised techniques here and two types among them – Statistical Baseline Model, Linear and Logistic Regression and the more efficient Tree-based Ensemble Model, Random Forest Regressor and Classifier.

We apply these in the following pipeline:

- Defining Variables and Instantiating Models - In this segment, we will instantiate regression and classification models. Code provided prepares the covariates and outcomes we will use for model analysis.
  Steps
  - Instantiate LinearRegression(), LogisticRegression(), RandomForestRegressor(), and RandomForestClassifier() objects from sklearn.linear_model and sklearn.ensemble, and assign them to linear_regression, logistic_regression, forest_regression, and forest_classifier, respectively.
  - For the random forest models, specify max_depth=4 and random_state=0 initially.

- Defining our Scoring Metrics - In this segment, we will create two functions that compute a model's score. For regression models, we will use correlation as the score. For classification models, we will use accuracy as the score.
  Steps
  1. Define a function called correlation with arguments estimator, X, and y. The function should compute the correlation between the observed outcome y and the outcome predicted by the model.
     - To obtain predictions, the function should first use the fit method of estimator and then use the predict method from the fitted object.
     - The function should return the first argument from r2_score comparing predictions and y.
  2. Define a function called accuracy with the same arguments and code, and computing the accuracy_score for comparing predictions and y.

- Making the Baseline Regression Model - In this segment, we will compute the cross-validated performance for the linear and random forest regression models.
  Steps
  1. Call cross_val_score using linear_regression and forest_regression as models. Store the output as linear_regression_scores and forest_regression_scores, respectively.
     - Set the parameters cv=10 to use 10-fold cross-validation and scoring=correlation to use our correlation function defined in the previous segment.
  2. Plotting code has been provided to compare the performance of the two models. Use plt.show() to plot the correlation between actual and predicted revenue for each cross-validation fold using the linear and random forest regression models.
  3. We observe that forest_regression clearly provides a better result than linear_regression.

Linear Regression Scores: [0.34629263 0.18048078 0.26013231 0.25660351 0.29642796 0.18214009 0.23756296 0.27476138 0.36637537 0.38254666]

Forest Regression Scores: [0.82734243 0.58048084 0.53047469 0.5237927 0.61230199 0.47882372 0.45987321 0.54075802 0.56945885 0.55119326]

- Making the Baseline Classification Model - In this segment, we will compute cross-validated performance for the logistic and random forest classification models.

  Steps

  1. Call cross_val_score using logistic_regression and forest_classifier as models. Store the output as logistic_regression_scores and forest_classification_scores, respectively.
     - Set the parameters cv=10 to use 10-fold cross-validation and scoring=accuracy to use our accuracy function defined in the previous exercise.
  2. Plotting code has been provided to compare the performance of the two models. Use plt.show() to plot the accuracy of predicted profitability for each cross-validation fold using the logistic and random forest classification models.
  3. We observe that both models perform good for the classification but forest_classifier works a bit better.

Logistic Regression Scores: [0.91131498 0.85626911 0.89602446 0.87461774 0.84097859 0.83792049 0.84355828 0.82822086 0.80981595 0.80368098]

Forest Classification Scores: [0.95412844 0.95412844 0.97859327 0.96330275 0.92966361 0.91743119 0.96625767 0.92331288 0.88957055 0.84969325]

- Optimizing the Data and Model - In the Baseline Regression and Classification Models, we saw that predicting revenue was only moderately successful while predicting profitability was really successful. It might be the case that predicting movies that generated precisely no revenue is difficult. In the next three segments, we will exclude these movies, and rerun the analyses to determine if the fits improve. In this segment, we will rerun the regression analysis for this subsetted dataset. Although our target is to improve the regression models, we will still re-train our classification models too for an even better prediction than before.
  Steps
  1. Define positive_revenue_df as the subset of movies in df with revenue greater than zero.
  2. Code is provided below that creates new instances of model objects. Change the max_depth parameter of RandomForest models from 4 to 20
  3. to increase the accuracy of our model. Replace all instances of df with positive_revenue_df.

- Making the Optimized Regression Models - In this segment, we will compute the cross-validated performance for the optimized linear and random forest regression models for positive revenue movies only.

  Steps

  1. Call cross_val_score using linear_regression and forest_regression as models. Store the output as linear_regression_scores and forest_regression_scores, respectively.
     - Set the parameters cv=10 to use 10-fold cross-validation and scoring=correlation to use our correlation function defined in the previous exercise.
  2. Plotting code has been provided to compare the performance of the two models. Use plt.show() to plot the correlation between actual and predicted revenue for each cross-validation fold using the optimized linear and random forest regression models.
  3. Here, too, the forest_regression model works better than the linear_regression model. We also find substantial improvement in the performance of our regression models.
  4. Code is provided that prints the importance of each covariate in predicting revenue using the random forests regresssor.
     - We observe that vote_count, budget and popularity are the most important features in predicting revenue

Linear Regression Scores: [0.69499467 0.65469345 0.55611339 0.53935215 0.5282503 0.49612886 0.46010822 0.40262575 0.59526262 0.34963542]

Forest Regression Scores: [0.94248728 0.94417169 0.91656935 0.91537018 0.90063549 0.88887538 0.8927123 0.90383198 0.93086626 0.87472553]

```
[('TV Movie', 0.0),
 ('Foreign', 0.00019451886848061625),
 ('Documentary', 0.0007702474950609l9),
 ('History', 0.0011983868642010062),
 ('Western', 0.001681682600822341),
 ('Animation', 0.002328943152840267),
 ('Music', 0.0029286077957537933),
 ('War', 0.003358498137479063),
 ('Family', 0.0036447881691100453),
 ('Fantasy', 0.0040724246396627835),
 ('Horror', 0.004317475627051731),
 ('Adventure', 0.004332978392311065),
 ('Romance', 0.004526072905643128),
 ('Mystery', 0.004562997619417912),
 ('Action', 0.005457830935637969),
 ('Comedy', 0.005971708468969015),
 ('Crime', 0.006154015801259207),
 ('Science Fiction', 0.0062509706471508865),
 ('Thriller', 0.007793558727763475),
 ('Drama', 0.008296379636712207),
 ('vote_average', 0.053015556866495034),
 ('runtime', 0.05777698041422592),
 ('popularity', 0.08364158834413118),
 ('budget', 0.293240442597374),
 ('vote_count', 0.43448334529244653)]
```

Fig 3.5.1 Features along with their importance for Revenue Prediction using RandomForest

- Making the Optimized Classification Models - In this segment, we will compute cross-validated performance for the logistic and random forest classification models for positive revenue movies only.

  Steps

  1. Call cross_val_score using logistic_regression and forest classifier as models. Store the output as logistic_regression_scores and forest_classification_scores, respectively.
     - Set the parameters cv=10 to use 10-fold cross-validation and scoring=accuracy to use our accuracy function defined in the previous segment.
  2. Plotting code has been provided to compare the performance of the two models. Use plt.show() to plot the accuracy between actual and predicted profitability for each cross-validation fold using the logistic and random forest classification models.
  3. forest classifier obviously performs far better than logistic_regression. In fact, logistic_regression doesn't improve at all and is stuck at the same range but forest classifier improves to a perfection.
  4. Code is provided that prints the importance of each covariate in predicting profitability using the random forests classifier.
     - We observe that here too vote_count, popularity and budget are the most important features but the importance of popularity and budget are reversed

```
Logistic Regression Scores: [0.9112628 0.86986301 0.85273973 0.85958904 0.8390411 0.8390411
0.8630137 0.85273973 0.81164384 0.80479452]

Forest Classification Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

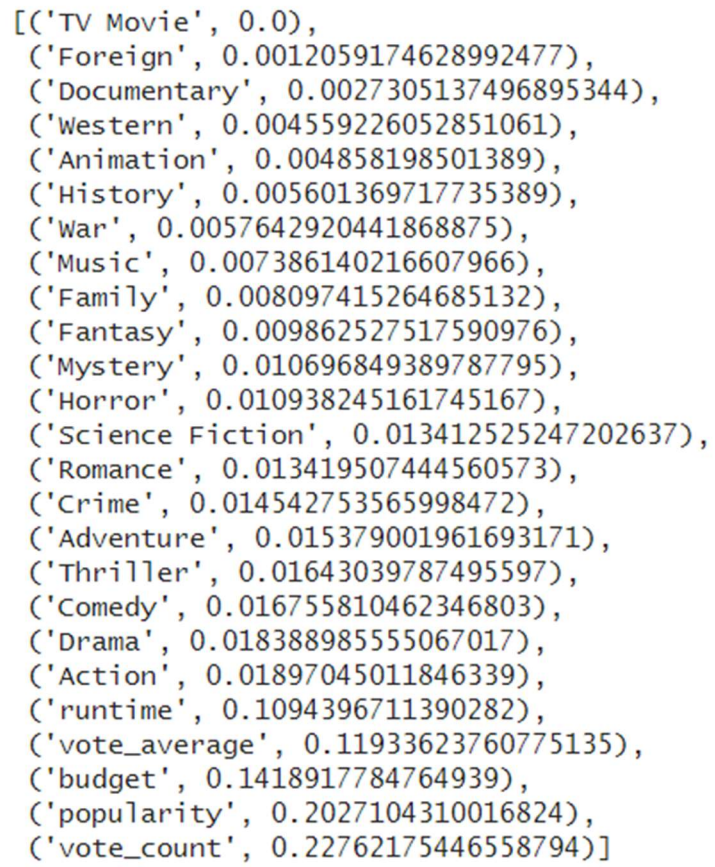

Fig 3.5.2 Features along with their importance for Profitability Prediction using RandomForest

3.6 Summary

This section explains in detail the processes and techniques we followed and used in our Pipeline and makes a walkthrough our code, how we implemented the intended, and the observations we achieved through those using finely-visualizable graphs and representations.

4. PERFORMANCE ANALYSIS

4.1 Introduction

Here, we will perform a detailed analysis of the results we obtained from using the above-mentioned techniques with the means of various graphical representations and getting observations from those.

4.2 Performance Analysis

Similar to the study conducted in various Papers mentioned in [2], [3] and [5], we used k-fold cross-validation rather than running a single experiment, specifically 10-fold cross-validation.
As mentioned previously, we used two fundamentally different types of predictive models – Basic Statistical Model, Linear and Logistic Regression and Tree-based Ensemble Model, Random Forest Model. Also, we used Correlation (r2_score) for the Regression task and Accuracy (accuracy_score) for the Classification task.

Type of Model	Task	Model	Evaluation Metric	Score (Mean)
BASELINE MODEL	1. Regression	Linear Regression	Correlation	0.278
		Random Forest Regressor		0.567
	2. Classification	Logistic Regression	Accuracy	0.85
		Random Forest Classifier		0.932
OPTIMIZED MODEL	1. Regression	Linear Regression	Correlation	0.527
		Random Forest Regressor		0.911
	2. Classification	Logistic Regression	Accuracy	0.85
		Random Forest Classifier		1

Table 4.2.1 Model Evaluation Score Table

- Performance of Baseline Regression Models – We previously implemented the Baseline regression models of Linear Regression and Random Forest Regressor (with max_depth = 4). The results it showed is provided in Table 4.1.
    - Plotting the results from the 10-fold cross validation for both the models –

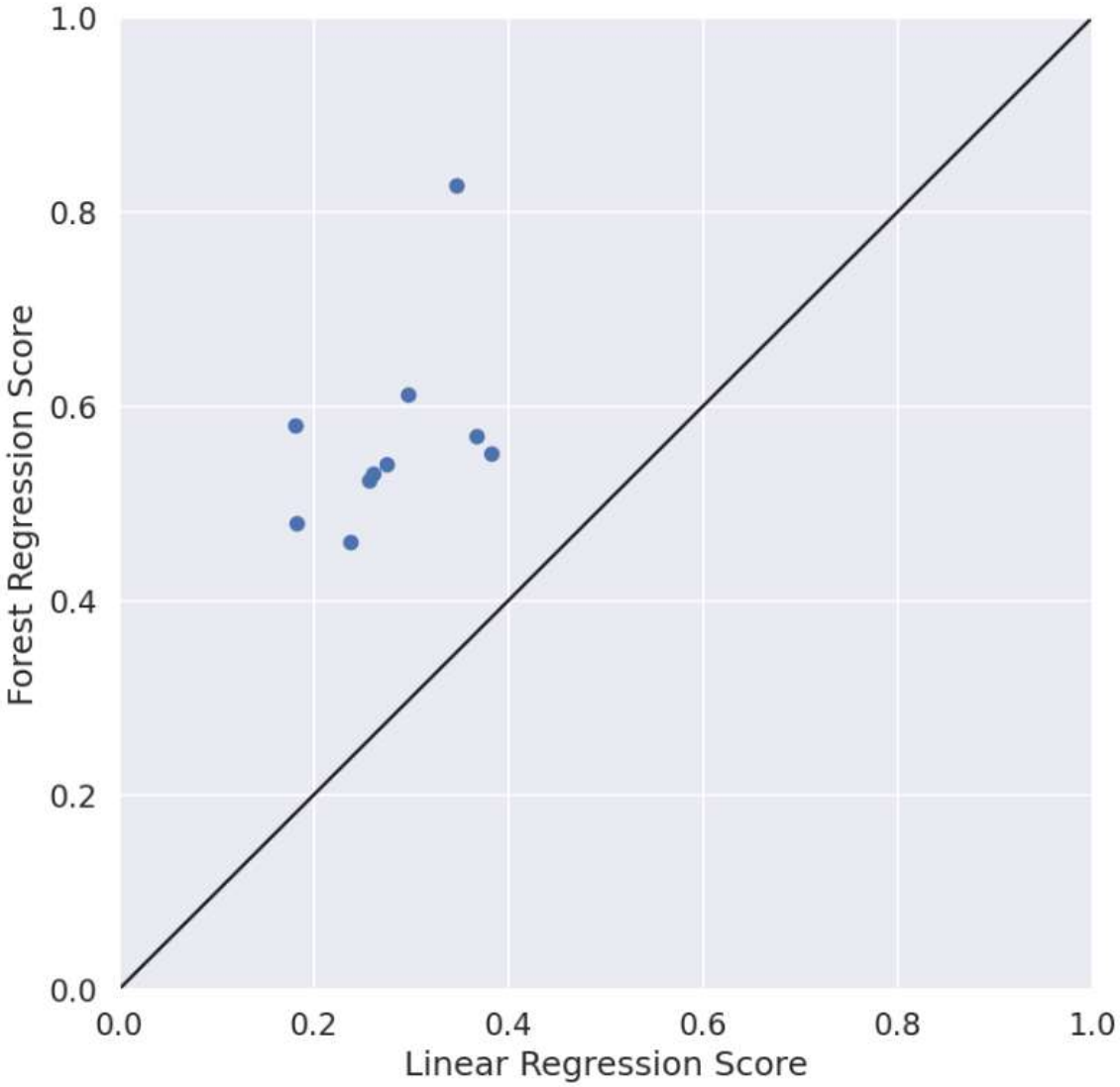

Fig 4.2.1 Baseline Linear Regression vs Random Forest Regression Scores

- We observe that 10-fold cross-validation of -
    - forest_regression scores range from 0.4 to 0.6 and we also have one above 0.8
    - linear_regression scores range from 0.2 to 0.4

    This clearly shows that the baseline model for forest_regression works better than that of linear_regression but predicting Revenue as a whole was only moderately successful.

- Performance of Baseline Classification Models – We previously implemented the Baseline classification models of Logistic Regression and Random Forest Classifier (with max_depth = 4). The results it showed is provided in Table 4.1.
    - Plotting the results from the 10-fold cross validation for both the models –

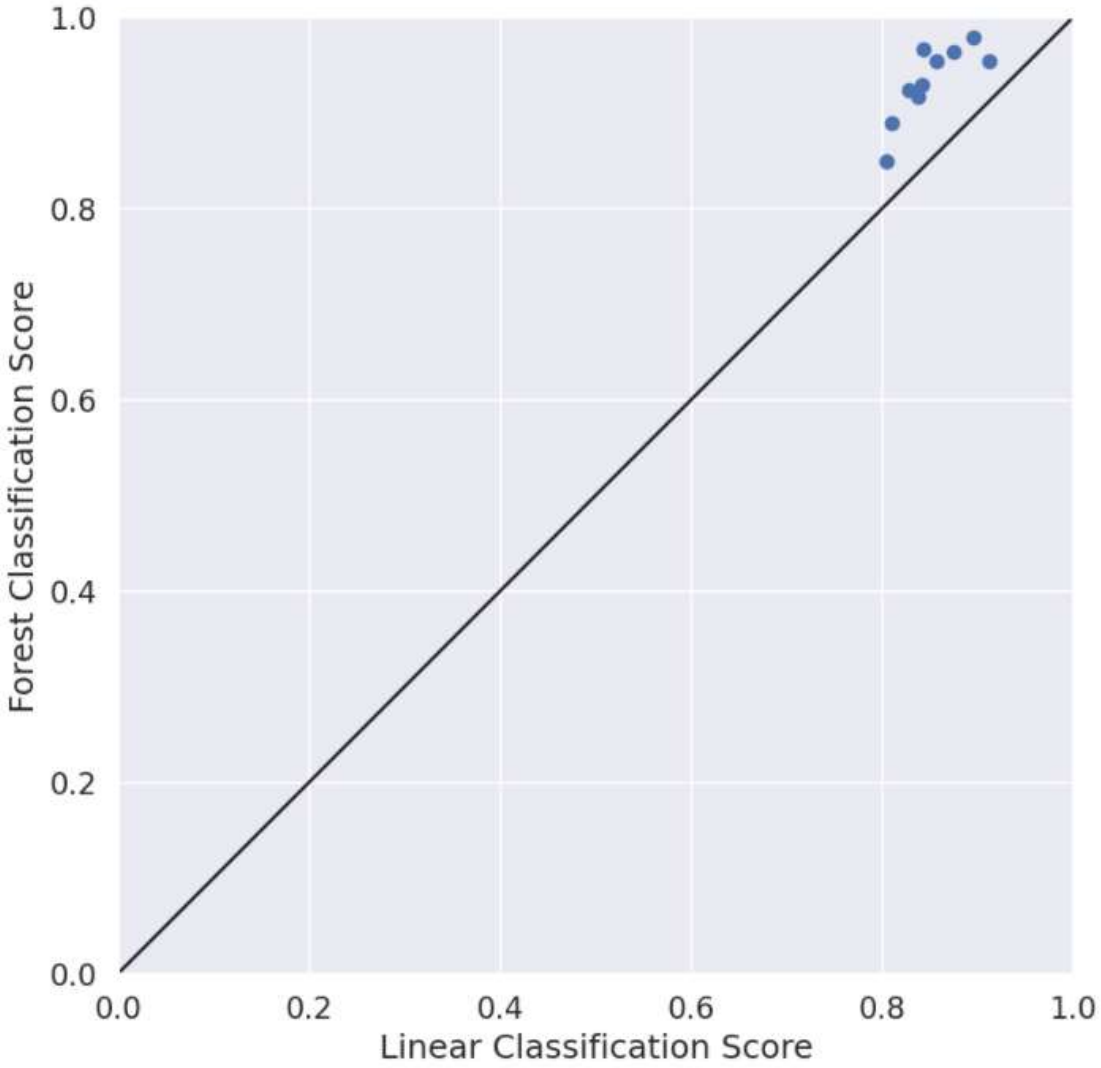

Fig 4.2.2 Baseline Logistic Regression and Random Forest Classifier Scores

We observe that 10-fold cross-validation of -

- forest_classification scores range mainly from 0.9 to 1
- logistic_regression scores range from 0.8 to 0.9

This clearly shows that forest_classification works a bit better than logistic_regression

- Performance of Optimized Regression Models – We previously implemented the Optimized regression models of Linear Regression and Random Forest Regressor (with max_depth = 20) only on movies that has positive revenue. The results it showed is provided in Table 4.1.

  - Plotting the results from the 10-fold cross validation for both the models –

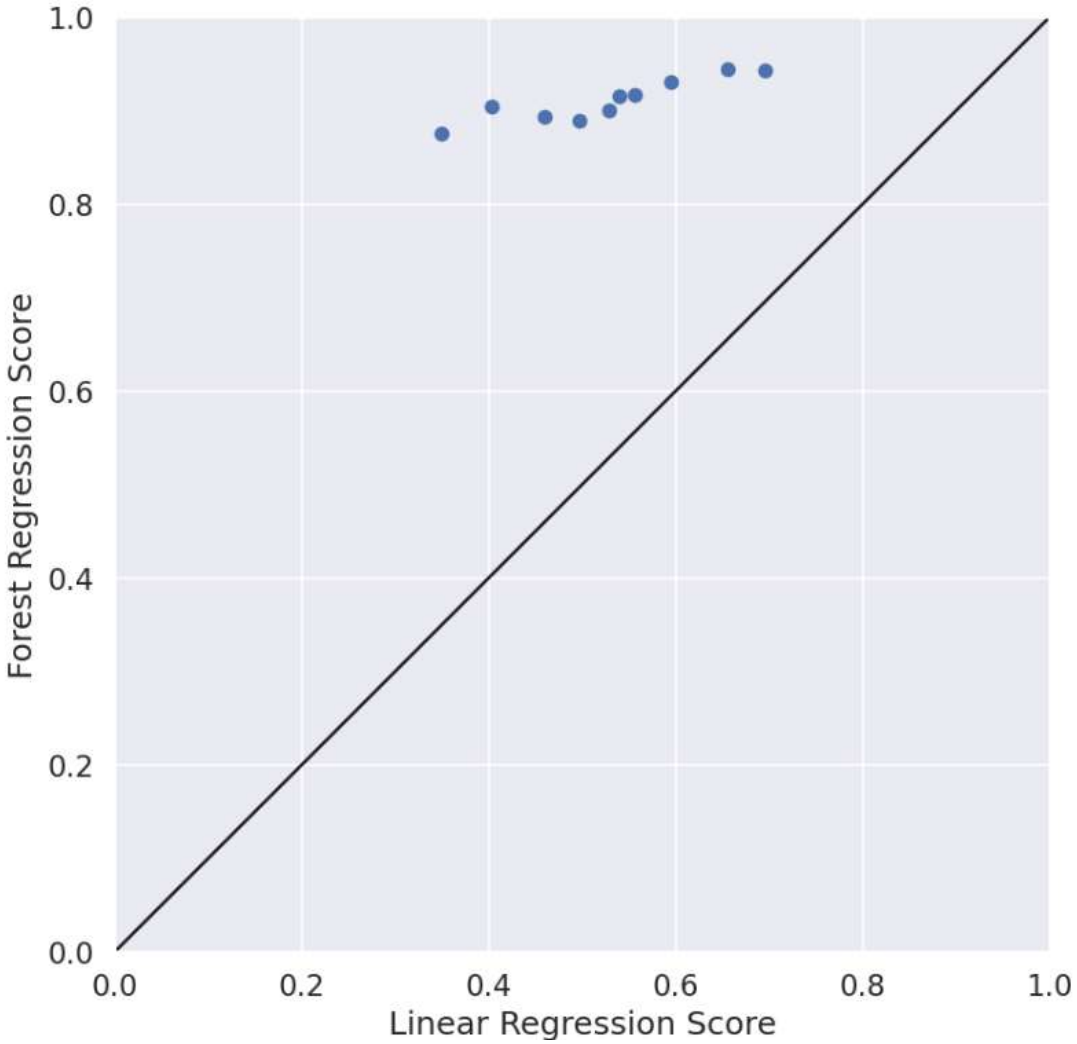

Fig 4.2.3 Optimized Linear Regression vs Random Forest Regression Scores

We observe that 10-fold cross-validation of -

- Optimized forest_regressor score for positive revenue data range around 0.9 which is a very high score
- linear_regression score for positive revenue data range from 0.4 to 0.7 which is quite good for a linear model

This again shows that the optimized model for forest_regression works better than that of linear_regression and considering only positive revenue movies has increased the score considerably.

We also print the feature importance of all the features using RandomForest and found out that vote_count, budget and popularity are the most important of them all.

- Performance of Optimized Classification Models – We previously implemented the Optimized classification models of Logistic Regression and Random Forest Classifier (with max_depth = 20) only on movies that has positive revenue. The results it showed is provided in Table 4.1.

    - Plotting the results from the 10-fold cross validation for both the models –

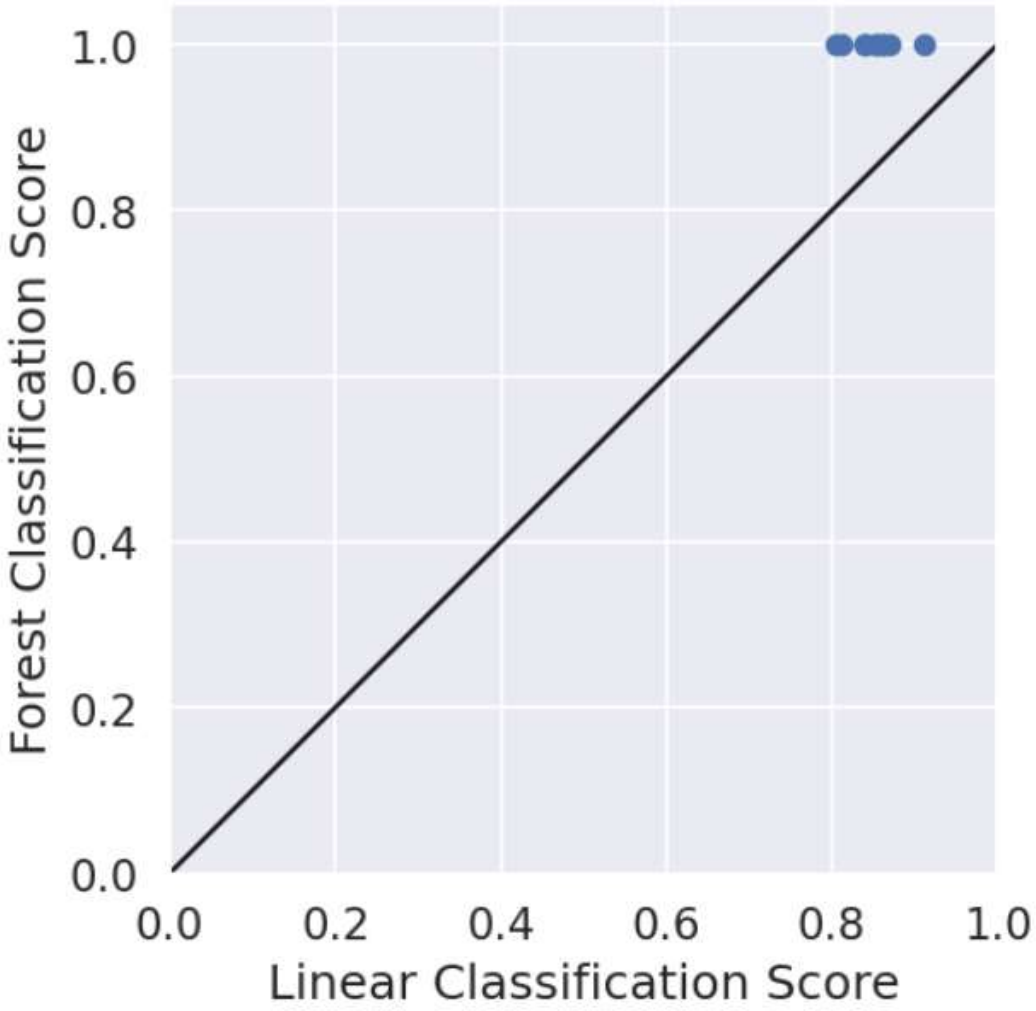

Fig 4.2.4 Optimized Linear Regression vs Random Forest Regression Scores

We observe that 10-fold cross-validation of -

- Optimized forest_classifier scores for positive revenue data are all perfect and classifies the profitability with 100% accuracy.
- logistic_regression scores for positive revenue data are still the same as it was during the baseline models, i.e., they range from 0.8 to 0.9 and didn't improve at all.

This again shows that the optimized model for forest_classifier works better than that of logistic_regression and considering only positive revenue movies has increased the score of forest_regressor considerably but the scores for logistic_regressor remains the same.

We also print the feature importance of all the features using RandomForest and found out that vote_count, popularity and budget are the most important of them all.

4.3 Summary

Through all these test and validations, we surmised that Tree-based Ensemble Models works the best for both Regression and Classification tasks of the Movie dataset we used.

5. FUTURE ENHANCEMENTS

5.1 Limitation / Constraints of our System

Although our pipeline resulted in a pretty high score with Random Forest Ensemble model, we still have quite a few limitations that needs to be attended:

- The Dataset is too small in comparison, so, fitting a model with such limited data might fetch us a good result for now but it won't be able to for larger datasets.
- The method we used for treating missing values was by dropping those rows. Although this would be good for datasets with less noisy data but one with a high amount will provide inaccurate results.
- While optimizing our predictive models, we found out that none of the models are able to properly train for those movies with revenue generated as 0 while that is not unnatural in the real world.

5.2 Future Enhancements

In accordance to the limitations mentioned above, there can be various enhancements to be made in building our model. Some of them are:

- Using a very large Dataset supposedly provided by either Netflix or any other large database where all TV Series and other shows will also be considered since they are becoming quite popular nowadays.
- Explore and use various other feature engineering techniques to come up with a way for countering missing data. Also, since it's a large dataset, outliers should also be accounted for and taken care of.
- Using a two-layer neural network to segregate movie revenue into different categories, where, revenue is modelled as a discrete variable instead of as a continuous one that we used in our case.

5.3 Conclusion

There are various ways to implement a Predictive Model Pipeline for predicting Movie revenue and classifying Movie profitability but here we compared the most popular Ensemble model with the basic Statistical Models and drew an analysis of how we can improve the analysis and prediction score by using various feature engineering techniques. I also demonstrated that Logistic regression performs quite well even with faulty data and that Random Forest can produce very good results in spite of noises and outliers present in the data.

REFERENCES

1. TMDb 5000 Movie Dataset, Metadata on ~5,000 movies from TMDb, Version 2, 2017 <https://www.kaggle.com/tmdb/tmdb-movie-metadata>

2. "The Numbers - Movie Market Summary 1995 to 2011." The Numbers - Movie Box Office Data, Film Stars, Idle Speculation. < https://www.the-numbers.com/market/ >

3. Simonoff, J. S. and Sparrow, I. R. "Predicting movie grosses: Winners and losers, blockbusters and sleepers", In Chance, 13(3), (Summer 2000). <http://pages.stern.nyu.edu/~jsimonof/movies/movies.pdf > <Google Scholar>

4. Chen, Andrew, "Forecasting Gross Revenues at the Movie Box Office", University of Washington, Seattle, WA, June 2002. <Google Scholar>

5. N. Quader, M. O. Gani, D. Chaki and M. H. Ali, "A machine learning approach to predict movie box-office success," *2017 20th International Conference of Computer and Information Technology (ICCIT)*, Dhaka, 2017, pp. 1-7, doi: 10.1109/ICCITECHN.2017.8281839.

6. Data Pre-processing, Wikipedia, October, 2020, < https://en.wikipedia.org/wiki/Data_pre-processing>

7. Exploratory data analysis, Wikipedia, October, 2020 <https://en.wikipedia.org/wiki/Exploratory_data_analysis>

8. Predictive modelling, Wikipedia, October, 2020 <https://en.wikipedia.org/wiki/Predictive_modelling>