MOVIE SUCCESS RATE FORECAST SYSTEM

Mrs. Jayashree L K [1], S.Preethi [2], Shreya Shetty K[3], Sowmya P [4], Sushmitha G[5]

[1.]Assistant Professor, Department of CSE, Vemana Institute of Technology,Bangalore. [2, 3, 4, 5.]
UG student, Department of CSE, Vemana Institute of Technology, Bangalore.
jayashree26@gmail.com,spreethi286@gmail.com,shreya140397@gmail.com,sowmyajes@gmail.com,
sushmithag2222@gmail.com

Manuscript History
Number: IRJCS/RS/Vol.06/Issue06/JNCS10115
Received: 29, May 2019
Final Correction: 30, May 2019
Final Accepted: 02, June 2019
Published: June 2019
doi://10.26562/IRJCS.2019.JNCS10115
Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India

ABSTRACT - Movie success prediction plays a vital role in the movie industry as it involves huge amounts of investment. However, success rates of a movie cannot be predicted based on a single attribute. Hence, a model is built based on an interesting relationship between the attributes. Movie industry can take a help of this model to change the movie criteria for obtaining likelihood of blockbusters. Every criteria involved is given a weight and then the prediction is made based on these. For example, if a movies budget was below 5 million, the budget was given a lower weight. Based on the number of actors, directors and producers past successful movies, every category is given equal weight age. Suppose the movie is to be released on a weekend, it is given higher weight because the chances of success were greater. If with the release of a movie, there was another high success movie released, a lower weight was given to the release time indicating that the fate of movie being a success is low as a result of competition. The criteria were not limited just to the ones mentioned. There was being additional factors discussed in this work.

KEYWORDS – IMDb dataset, Random Forest and Feature selection.

I. INTRODUCTION

For this work, patterns and trends were extracted from the dataset that could be beneficial in predicting movies success. The data goes through cleaning and integration process after which the machine learning procedures are applied. The trend and patterns in the data can be identified by algorithms in machine learning. Machine learning approach is important since it can help to identify the hidden patterns and relationships among various variables by itself. These relationships can in turn help in identifying sequence of events, classification, clustering, and predicting future events. Some examples are profit prediction where lots of data are involved that makes use of patterns in the data, investment decision, weather forecast, simulations, visualization tools, and medicinal purposes. Movie success forecasting is important because it involved significant time and investment. For this reason, it is important for the shareholders to have less uncertainty involved. They can achieve this very well using machine learning techniques. Movie success predictions, trends and variable dependence can very well be determined using data mining. Due to huge investments involved in the movie industry, success forecast plays an important role. Production houses invest millions of dollars on advertising campaigns and movie promotions so, knowing the likelihood of the movie being success or flop could benefit them greatly. It will also help them to decide when it is most appropriate to release a movie by looking at the overall market. If the outcome is not forecasted by the model, uncertainty increases and success confidence is lowered. This is particularly risky for stakeholders who have invested their significant resources. The objective is to come up with a precise prediction using the model, subsequently, providing confidence to stakeholders in their investments.

A useful model is developed in this study which can lower chance of failure and can provide the stakeholders with confidence and a visible prediction of success. Variables like budget, actors, director, producer, set locations, story writer, movie release day, competing movie releases at the same time, music, and release date are considered for the prediction. In this proposed work, a forecasting model has been developed (using machine learning techniques) which predicts the success and failure of an upcoming movie depending on certain criteria's. The proposed work provides an advantage in that strong correlations were found between different criteria and movie success rating.

II. METHODOLOGY

This project has four phases, as listed below:

1. Collection of data and cleaning
2. Feature selection and extraction
3. Model Training
4. Test and Analysis

A. Collection of data and Cleaning

Primary data extracted from the online sources remains in the raw form of statements, digits and qualitative terms. There are errors, omissions and inconsistencies in the unclean data. It requires corrections. A huge volume of raw data collected through field survey needs to be grouped for similar details of individual responses. Machine learning requires two things to operate, data and models. The learning model can only be trained correctly if the extracted data has sufficient features. Data Preprocessing is a process converts the raw data into a clean data set. In other words, the data extracted from online sources cannot readily be used for analysis. Therefore, the following steps are executed to transform the data into pure data set. It includes –

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

Real time data is composed of the following:

- **Inaccurate data or missing data –** Incase of continuously collected, mistakes in data entry, technical problems with biometrics etc.
- **The presence of noisy or erroneous data and outliers -** The reasons could be a technical problem in gadget that gathers data, human mistakes during data entry.
- **Inconsistent data -** Due to the reasons such as existence of duplication within data, human data entry, containing mistakes in codes or names.

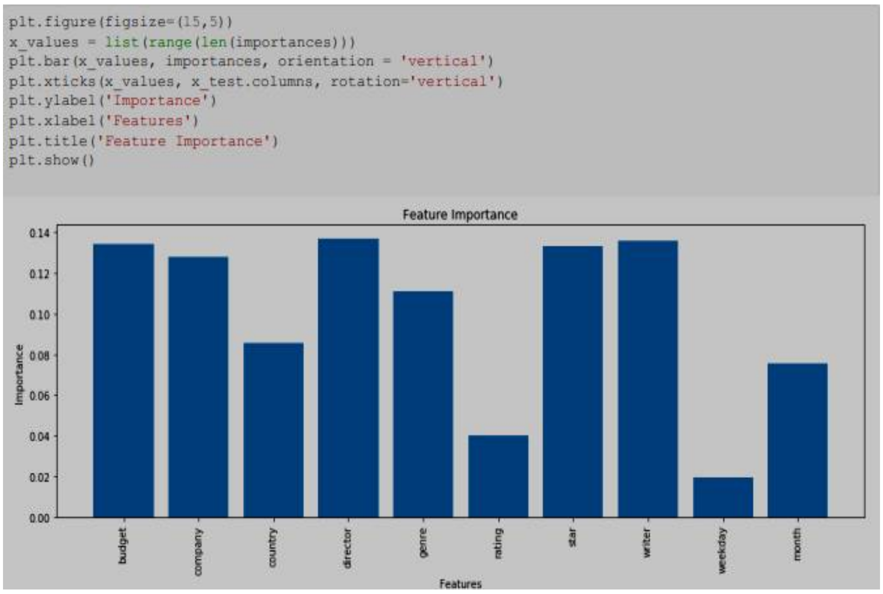
The dataset that is employed in training the model is collected from IMDb website. It consists of movies that were released from 1986 to 2016. Initially the dataset consisted of 9782 entries, after data preprocessing the dataset was reduced to 6825 entries.

The dataset primarily contains the following fields:

- Budget: Investment on the movie. Some movies don't have this, so it appears as 0.
- Company: The production company.
- Country: Country of origin.
- Director: The director of the movie.
- Genre: The main theme of the movie.
- Gross: Revenue from the movie.
- Name: Name of the movie.
- Rating: Rating declared to the new movie entry by censor board(R, U, UA etc)
- Released: Release date in YYYY-MM-DD format.
- Score: IMDb user ratings.
- Votes: User votes, includes both up votes as well as down votes.
- Star: The main actor/actress.
- Writer: The Writer of the movie.

B. Feature Selection and Extraction

Feature extraction begins from an initial set of measured data and then obtains derived features that are informative and non-redundant, subsequently learning and generalizing steps, and in few cases resulting in better human interpretations. Dimensionality reduction is achieved by feature extraction. A feature vector is computed when input data to an algorithm is too huge to be processed or if the data is redundant. Deducing a subset of features is called feature selection. This reduced representation instead of the complete initial data facilitates to carry out the desired operation. The features selected to train the model are budget, company, country, director, rating, genre, star, writer, weekday and month because they highly affect the prediction accuracy.



```
plt.figure(figsize=(15,5))
x_values = list(range(len(importances)))
plt.bar(x_values, importances, orientation = 'vertical')
plt.xticks(x_values, x_test.columns, rotation='vertical')
plt.ylabel('Importance')
plt.xlabel('Features')
plt.title('Feature Importance')
plt.show()
```

C. Model Training

The process of training a machine learning model involves providing a machine learning algorithm with training data to learn from. The training data will contain the correct answers for the historical dataset. The machine will learn how to predict the answers for respective data sets given through the help of training data. When the trained model is given with the new set of test data, it will be able to predict the answers. Model can be trained using Random Forest Algorithm.

Random Forest Algorithm

Random Forest Algorithm is flexible and easy to use. It gives required accuracy most of the times without hyper-parameter tuning. It is very easy to use because of its simplicity. We can use Random Forest for both classification and regression tasks. In Random forest algorithm, many decision trees are built at the beginning, in the final step all the outcomes are merged considering the most frequent outcome as the result.

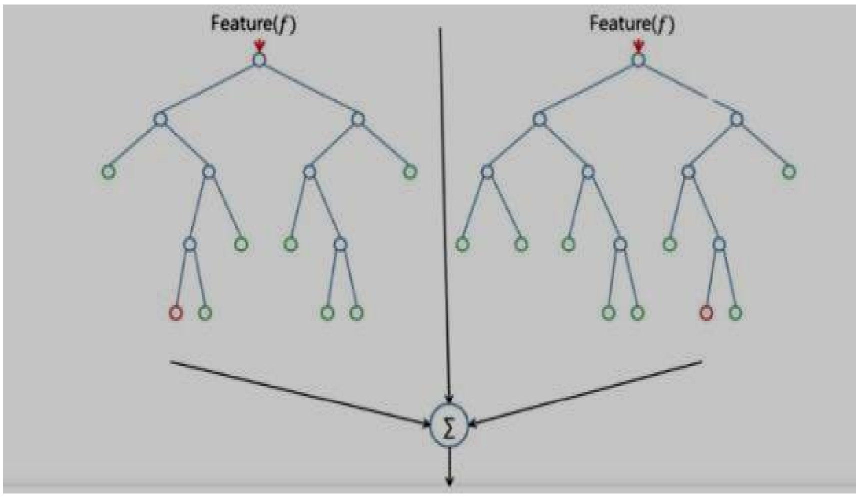


Fig.2: Illustration of random forest algorithm

Random Forest Algorithm randomly creates decision trees. Each decision tree will be trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result. Using Random Forest, we can easily predict the importance of each independent attribute on the dependent attribute. To increase the predictive power of the model or to make the model faster Hyper-parameters are used. The advantages of random forest algorithm are as follows:

- As hyper-parameters produce a good prediction result, it is easy to use.
- Hyper-parameters easy to understand and are very less in number.

The following figure shows the working of random forest algorithm

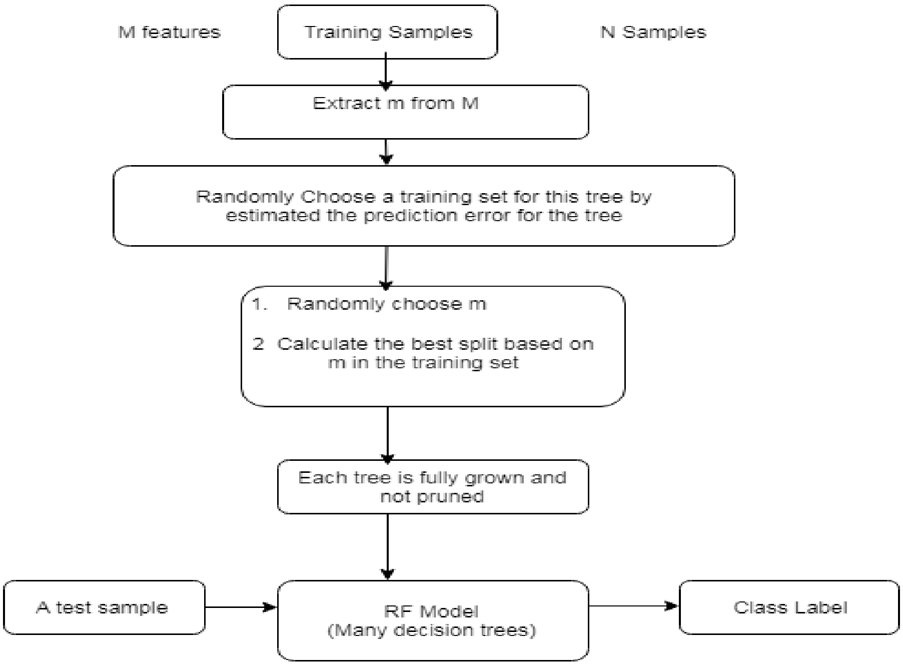
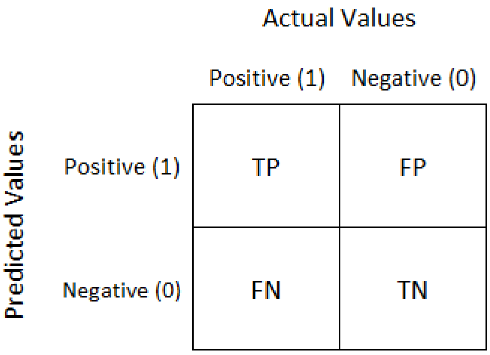

M features
Training Samples
N Samples
Extract m from M
Randomly Choose a training set for this tree by estimated the prediction error for the tree
1. Randomly choose m
2 Calculate the best split based on m in the training set
Each tree is fully grown and not pruned
A test sample
RF Model (Many decision trees)
Class Label

Fig.3: Working of random forest algorithm



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Fig.4: A general confusion matrix

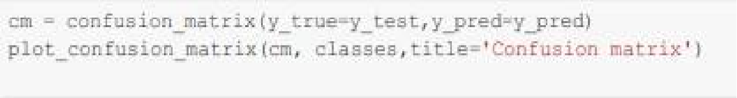
Definition of terms:

- **True Positive (TP):** Observation and its prediction are positive.
- **False Negative (FN):** Though the observation is positive, its prediction is negative.
- **True Negative (TN):** Both observation and prediction are negative.
- **False Positive (FP):** Though Observation is negative, it’s predicted positive.

Prediction accuracy can be calculated from confusion matrix as the sum of positives by total sum of positives and negatives. The equation is described below:

D. Test and Analysis

There are two different datasets that are used to build the model, namely training and test datasets. Dataset is split into train and test datasets during the initial phases. In testing phase the model uses test data set. The objective the machine learning model is to train the model perform well. Once the build model is tested then we will pass real time data for the prediction. Once prediction is done the output is analyzed to find out the crucial information. Performance of the trained model is evaluated by using accuracy as a measure of effectiveness of model. After model building, knowing the potential of model prediction on a new instance, is very important. Once a model is developed using the historical data, one would be curious as to how the model performs on the data that it has not seen during the model building process. The performance of the predictor is measured using, common performance metrics, such as accuracy, recall etc. Accuracy of prediction model can be measured through confusion matrix. Confusion matrix is a table that is describes the performance of prediction model on test data with known target values. Below figure represents a general confusion matrix: Confusion matrix used in this project is as shown below:



```
cm = confusion_matrix(y_true=y_test,y_pred=y_pred)
plot_confusion_matrix(cm, classes,title='Confusion matrix')
```

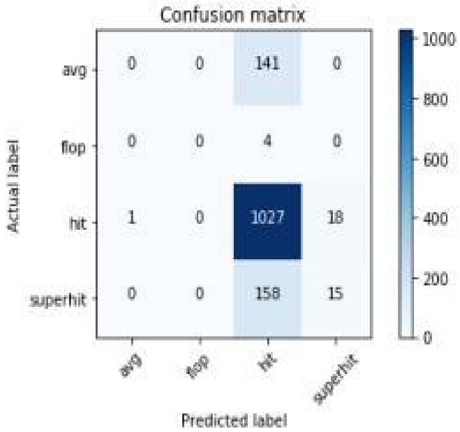


Fig.5: Confusion matrix

III.SYSTEM ARCHITECTURE

System Architecture design identifies the entire structure of the model

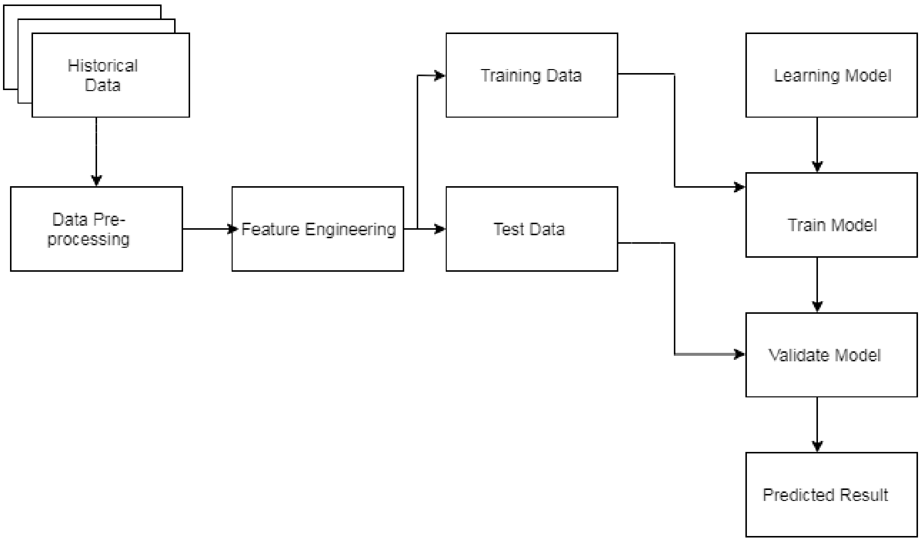


Fig. 6: System Architecture

Architecture design is tied to the goals established for a model and the content to be presented. Content architecture represents the way in which content objects are structured for presentation and navigation. Model architecture deals with how the application is structured to handle user interaction, internal processing tasks, effect navigation, and present content. Model architecture is described with respect to the context of the development environment where the application is to be implemented.

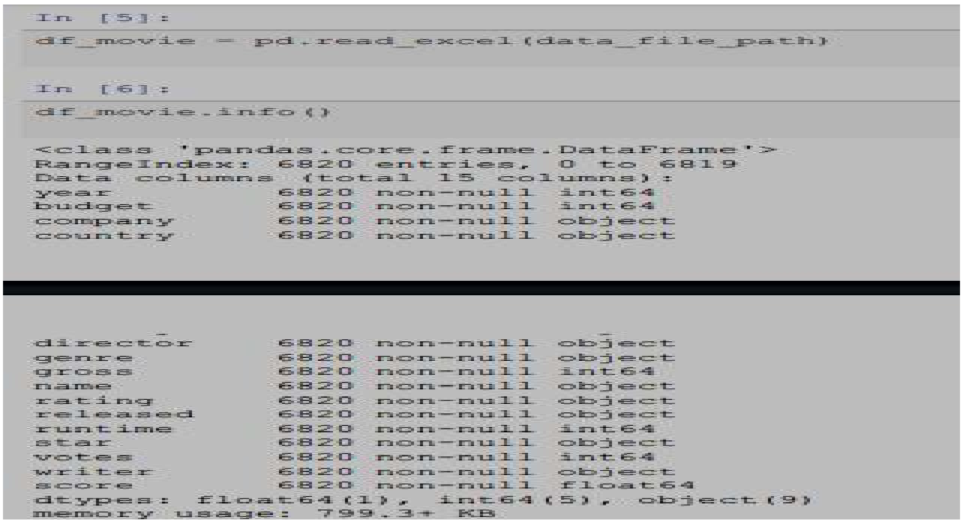
System Architecture design identifies the complete structure of success rate prediction. Firstly, the historical data is extracted from the IMDb data set and the collected data is preprocessed and feature engineering is done for building additional features out of the existing data, then data is split into Train and Test data. The training data is trained using random forest algorithm. The test data is validated. The validation process is done with list out expected outcomes. Finally the model gives the Predicted result.

IV. IMPLEMENTATION

Humungous amounts of numerical data will be dealt by various machine learning techniques. The data that is downloaded from IMDb contains data in a format that can be understood by humans. This data will be transformed into numeric format in the data preprocessing stage.. While already numeric information like budget, date of release, gross, runtime, score and votes retain in their original format. The other features that are in the string format need to be converted. There is a list representation of the data obtained from IMDb. Indexing in python language begins with zero. The string data included in the list will be encoded based on the index by taking their first occurrence into consideration. This encoded movie data is then utilized for processing. Not all the features present in the dataset influence the prediction. Hence the feature that highly influences the prediction must be discovered. This can be done by computing the feature importance, after which attributes that affect the final movie forecasting by a very small amount will be eliminated.. Before the algorithm could be applied to train the model, the dataset is bifurcated as training and test data. The training dataset can be used in making the model learn different patterns necessary to identify success and failure.

The training section of the dataset will have the attribute values along with its target class. The model is then verified on the part of the dataset that involves testing which does not contain the target class values. Correct prediction of target values for test dataset determines the predictive accuracy of the model. It is possible to demine how accurate the model is b using the confusion matrix. The performance of the random forest classifier deployed is accurate by 76%. Decision trees are used for constructing the random forest. Every decision tree represents on how much each feature affects the final class attribute. Based on the decision made the final success rate is classified as to hit, flop or super hit. The bagging method of the random forest classifier is used. At each iteration of constructing the decision tree a bootstrap sample is referred. As the iteration continues several bootstrap samples containing subsets of IMDb is created alongside with their prediction. Now when a new train set is presented for the prediction the result from every bootstrap sample is taken into consideration. The new entry in the train data is compared with the attributes of bootstrap sample. The result with the highest vote will be considered and the final forecasting of the success rate will be decided

IV. RESULT



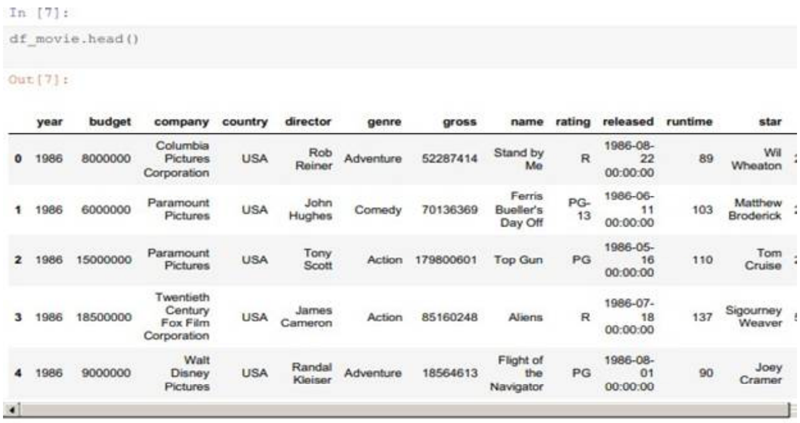
```
In [5]:
df_movie = pd.read_excel(data_file_path)

In [6]:
df_movie.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6820 entries, 0 to 6819
Data columns (total 15 columns):
year        6820 non-null int64
budget      6820 non-null int64
company     6820 non-null object
country     6820 non-null object

director    6820 non-null object
genre       6820 non-null object
gross       6820 non-null int64
name        6820 non-null object
rating      6820 non-null object
released    6820 non-null object
runtime     6820 non-null int64
star        6820 non-null object
votes       6820 non-null int64
writer      6820 non-null object
score       6820 non-null float64
dtypes: float64(1), int64(5), object(9)
memory usage: 799.3+ KB
```

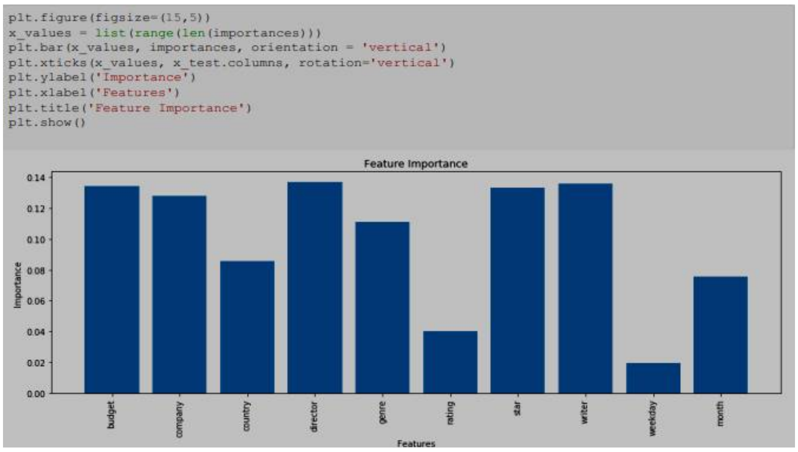
Fig.7: Displaying attributes of the dataset



```
In [7]:
df_movie.head()
Out[7]:
```

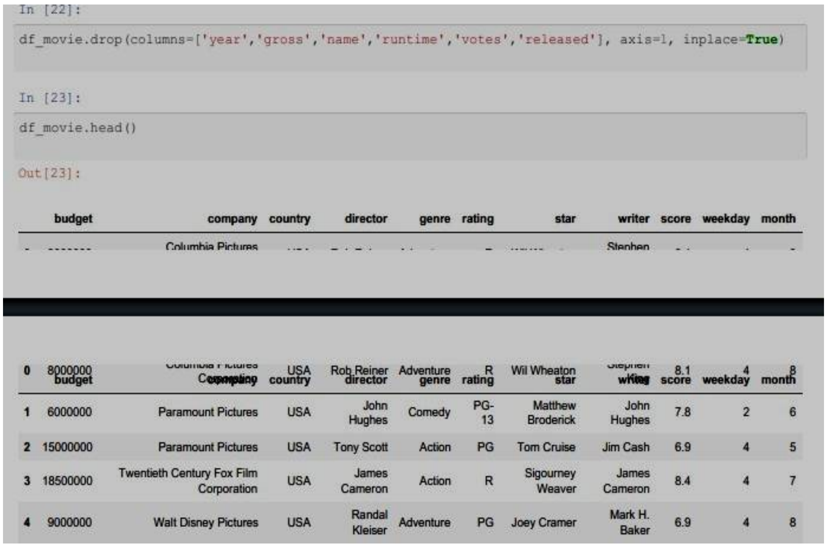
	year	budget	company	country	director	genre	gross	name	rating	released	runtime	star
0	1986	8000000	Columbia Pictures Corporation	USA	Rob Reiner	Adventure	52287414	Stand by Me	R	1986-08-22 00:00:00	89	Wil Wheaton
1	1986	6000000	Paramount Pictures	USA	John Hughes	Comedy	70136369	Ferris Bueller's Day Off	PG-13	1986-06-11 00:00:00	103	Matthew Broderick
2	1986	15000000	Paramount Pictures	USA	Tony Scott	Action	179800601	Top Gun	PG	1986-05-16 00:00:00	110	Tom Cruise
3	1986	18500000	Twentieth Century Fox Film Corporation	USA	James Cameron	Action	85160248	Aliens	R	1986-07-18 00:00:00	137	Sigourney Weaver
4	1986	9000000	Walt Disney Pictures	USA	Randal Kleiser	Adventure	18564613	Flight of the Navigator	PG	1986-08-01 00:00:00	90	Joey Cramer

Fig. 8: Displaying first 5 rows of the dataset



```
plt.figure(figsize=(15,5))
x_values = list(range(len(importances)))
plt.bar(x_values, importances, orientation = 'vertical')
plt.xticks(x_values, x_test.columns, rotation='vertical')
plt.ylabel('Importance')
plt.xlabel('Features')
plt.title('Feature Importance')
plt.show()
```

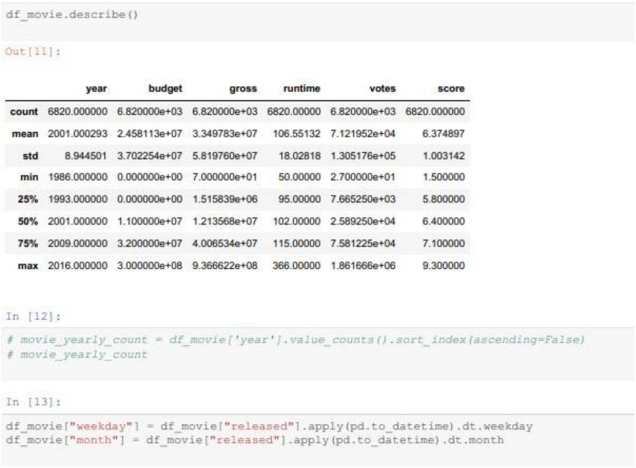
Fig. 12: Feature Importance



```
In [22]:
df_movie.drop(columns=['year','gross','name','runtime','votes','released'], axis=1, inplace=True)
In [23]:
df_movie.head()
Out[23]:
```

	budget	company	country	director	genre	rating	star	writer	score	weekday	month
0	8000000	Columbia Pictures	USA	Rob Reiner	Adventure	R	Wil Wheaton	Stephen King	8.1	4	8
1	6000000	Paramount Pictures	USA	John Hughes	Comedy	PG-13	Matthew Broderick	John Hughes	7.8	2	6
2	15000000	Paramount Pictures	USA	Tony Scott	Action	PG	Tom Cruise	Jim Cash	6.9	4	5
3	18500000	Twentieth Century Fox Film Corporation	USA	James Cameron	Action	R	Sigourney Weaver	James Cameron	8.4	4	7
4	9000000	Walt Disney Pictures	USA	Randal Kleiser	Adventure	PG	Joey Cramer	Mark H. Baker	6.9	4	8

Fig. 11: Displaying number of movies in each genre



```
df_movie.describe()
```

Out[11]:

	year	budget	gross	runtime	votes	score
count	6820.000000	6.820000e+03	6.820000e+03	6820.00000	6.820000e+03	6820.000000
mean	2001.000293	2.458113e+07	3.349783e+07	106.55132	7.121952e+04	6.374897
std	8.944501	3.702254e+07	5.819760e+07	18.02818	1.305176e+05	1.003142
min	1986.000000	0.000000e+00	7.000000e+01	50.00000	2.700000e+01	1.500000
25%	1993.000000	0.000000e+00	1.515839e+06	95.00000	7.665250e+03	5.800000
50%	2001.000000	1.100000e+07	1.213568e+07	102.00000	2.589250e+04	6.400000
75%	2009.000000	3.200000e+07	4.006534e+07	115.00000	7.581225e+04	7.100000
max	2016.000000	3.000000e+08	9.366622e+08	366.00000	1.861666e+06	9.300000

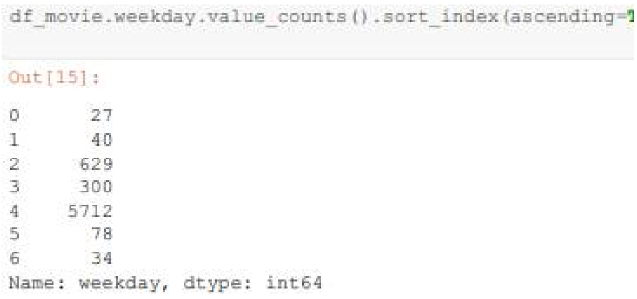
In [12]:

```
# movie_yearly_count = df_movie['year'].value_counts().sort_index(ascending=False)
# movie_yearly_count
```

In [13]:

```
df_movie["weekday"] = df_movie["released"].apply(pd.to_datetime).dt.weekday
df_movie["month"] = df_movie["released"].apply(pd.to_datetime).dt.month
```

Fig. 9: Displaying mean, min, max etc for numerical valued attributes

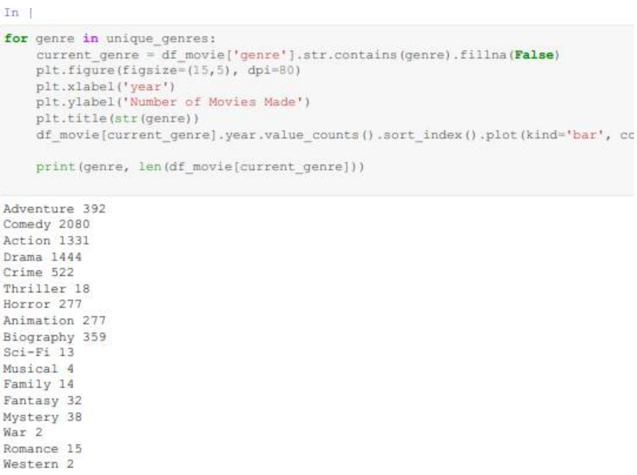


```
df_movie.weekday.value_counts().sort_index(ascending=1
```

Out[15]:

```
0      27
1      40
2     629
3     300
4    5712
5      78
6      34
Name: weekday, dtype: int64
```

Fig. 10: Displaying number of movies released on each day of a week



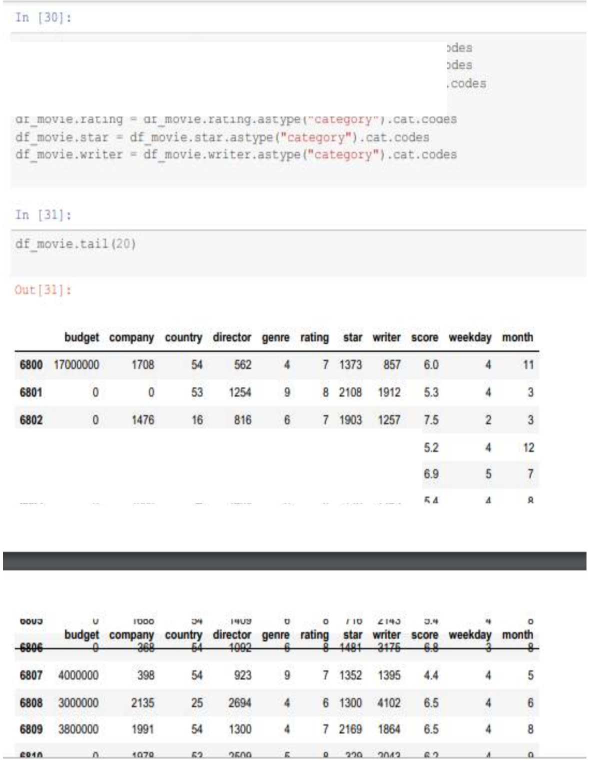
In |

```
for genre in unique_genres:
    current_genre = df_movie['genre'].str.contains(genre).fillna(False)
    plt.figure(figsize=(15,5), dpi=80)
    plt.xlabel('year')
    plt.ylabel('Number of Movies Made')
    plt.title(str(genre))
    df_movie[current_genre].year.value_counts().sort_index().plot(kind='bar', cc

    print(genre, len(df_movie[current_genre]))
```

```
Adventure 392
Comedy 2080
Action 1331
Drama 1444
Crime 522
Thriller 18
Horror 277
Animation 277
Biography 359
Sci-Fi 13
Musical 4
Family 14
Fantasy 32
Mystery 38
War 2
Romance 15
Western 2
```

Fig. 13: Feature Selection and extraction



In [30]:

```
                                                                  odes
                                                                  odes
                                                                  .codes
df_movie.rating = df_movie.rating.astype("category").cat.codes
df_movie.star = df_movie.star.astype("category").cat.codes
df_movie.writer = df_movie.writer.astype("category").cat.codes
```

In [31]:

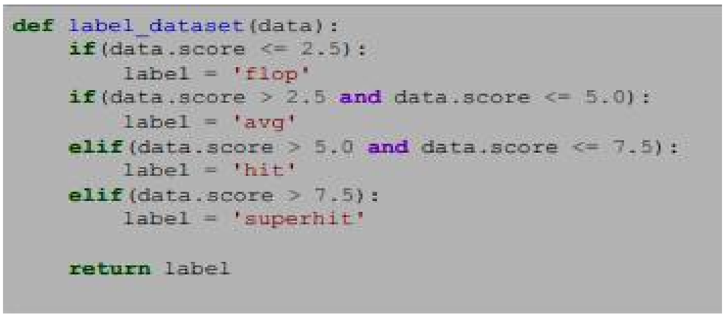
```
df_movie.tail(20)
```

Out[31]:

	budget	company	country	director	genre	rating	star	writer	score	weekday	month
6800	17000000	1708	54	562	4	7	1373	857	6.0	4	11
6801	0	0	53	1254	9	8	2108	1912	5.3	4	3
6802	0	1476	16	816	6	7	1903	1257	7.5	2	3
									5.2	4	12
									6.9	5	7
									5.4	4	8

	budget	company	country	director	genre	rating	star	writer	score	weekday	month
~~6806~~	~~0~~	~~368~~	~~54~~	~~1092~~	~~6~~	~~8~~	~~1481~~	~~3176~~	~~6.8~~	~~3~~	~~8~~
6807	4000000	398	54	923	9	7	1352	1395	4.4	4	5
6808	3000000	2135	25	2694	4	6	1300	4102	6.5	4	6
6809	3800000	1991	54	1300	4	7	2169	1864	6.5	4	8

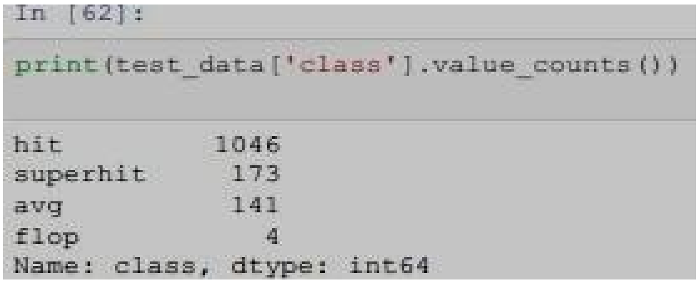
Fig. 14: Data Transformation



```
def label_dataset(data):
    if(data.score <= 2.5):
        label = 'flop'
    if(data.score > 2.5 and data.score <= 5.0):
        label = 'avg'
    elif(data.score > 5.0 and data.score <= 7.5):
        label = 'hit'
    elif(data.score > 7.5):
        label = 'superhit'

    return label
```

Fig. 15: Defining range for target class



```
In [62]:

print(test_data['class'].value_counts())

hit         1046
superhit     173
avg          141
flop           4
Name: class, dtype: int64
```

Fig. 16: Number of movies in each target class

CONCLUSION

Forecasting the fate of a movie even before its release forms the vital part of this model. With machine learning approach used in this experimentation this system is fitted as a go to model for investors of movies to have confidence on the amount that they invest and reduce the chances of risk. Forecasting the success of upcoming movies is an important task for the entertainment industry, and is inherently complex because to its extremely unpredictable nature. Predictions are made using data from IMDb.. Mining IMDb data is a tedious task there will be lots of features associated to a movie and each of them in different dimensions with huge amounts of missing fields and noisy data. In this work, random forest approach has been used to overcome the issues related to tweets. The proposed model aims to forecast movie success. The rate of forecasting is 76%.

REFERENCES

1. Darin Im, Minh Thao, Dang Nguyen, Predicting Movie Success in the U.S. market, Dept.Elect.Eng, Stanford Univ., California, December, 2011
2. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining Concepts and Techniques, 3rd ed. MA:Elsevier, 2011, pp. 83-117
3. Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, 2nd ed.NewYork: Wiley, 1973
4. Cohen, J., Cohen P., West, S.G., & Aiken, L.S. (2003). Applied multiple regression correlation analysis for the behavioral sciences. (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates
5. Christopher M.Bishop(2006),Pattern Recognition and Machine Learning, Springer, p. 205.
6. Cristianini, Nello; and Shawe-Taylor, John; An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000. ISBN 0-521-78019-5
7. W. Zhang and S. Skiena, Improving movie gross prediction through news analysis, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Milan, 2009
8. Sagar V. Mehta, Rose Marie Philip, AjuTalappillil Scaria, Predicting Movie Rating based on Text Reviews, Dept.Elect.Eng, Stanford Univ., California, December, 2011
9. Suhaas Prasad, Using Social Networks to improve Movie Ratings predictions, Dept. Elect.Eng, Stanford Univ., California, 2010