**Predicting Movie Revenue based on IMDb Data**

**Abstract**

The propose of this system is to predict the movie box office gross level in U.S. for movie investors before filming. They can use this system to adjust budget, cast and movie genre to optimize the movie box office gross.

**1.Introduction:**

Cinema is a multi-billion-dollar industry where even individual films earn over a billion dollars. Large production houses control most of the film industry, with billions of dollars spent on advertisements alone. Advertising campaigns contribute heavily to the total budget of the movies. Sometimes the investment results in heavy losses to the producers. Success or failure of a movie can depend on many factors: star-power, release date, budget, MPAA (Motion Picture Association of America) rating, plot and the highly unpredictable human reactions. The enormity of the number of exogenous variables makes manual revenue prediction process extremely difficult. However, in the era of computer and data sciences, volumes of data can be efficiently processed and modeled. Hence the tough job of predicting gross revenue of a movie can be simplified with the help of modern computing

power and the historical data available as movie databases.

Given the information known about a movie in the week of its release, can we predict the total gross revenue for that movie? Such information would be useful to producers, marketers, theater operators, and others in the movie industry, but it is a hard problem, even for human beings. We found that, given a set of numeric, textbased IMDb, logistic regression outperforms class-based linear regression at predicting gross revenue. In this project, we attempt to use IMDb data to predict the gross revenue of the movies.

The objective of our project is to predict the movie revenue based on attributes such as budget, actors involved, directors, year in which they were released, Facebook likes, user rating, number of votes, total revenue generated by movie, and recording the votes or rating, the geographical areas where movie was released, any other influences such as political movements, ongoing trends, and so on.

**2. DATASET COLLECTION AND PRE-PROCESSING**

Searched for available datasets to support our idea and thoroughly scrutinized them, to get the most suitable dataset for our idea. Shortlisted few datasets, we picked the most suitable dataset for our project. The initial dataset to be used will be collected from IMDb. It will consist of movies that were released from 2000 to 2016.

**2.1 Data Description**

1. movie\_title -Title of the Movie
2. duration - Duration in minutes
3. director\_name - Name of the Director of the Movie
4. director\_facebook\_likes - Number of likes of the Director on his Facebook Page
5. actor\_1\_name - Primary actor starring in the movie
6. actor\_1\_facebook\_likes - Number of likes of the Actor\_1 on his/her Facebook Page
7. actor\_2\_name - Other actor starring in the movie
8. actor\_2\_facebook\_likes - Number of likes of the Actor\_2 on his/her Facebook Page
9. actor\_3\_name - Other actor starring in the movie
10. actor\_3\_facebook\_likes - Number of likes of the Actor\_3 on his/her Facebook Page
11. num\_user\_for\_reviews - Number of users who gave a review
12. num\_critic\_for\_reviews - Number of critical reviews on imdb
13. num\_voted\_users - Number of people who voted for the movie
14. cast\_total\_facebook\_likes - Total number of facebook likes of the entire cast of the movie
15. movie\_facebook\_likes - Number of Facebook likes in the movie page
16. plot\_keywords - Keywords describing the movie plot
17. facenumber\_in\_poster - Number of the actor who featured in the movie poster.
18. title\_year - The year in which the movie is released
19. language - English, Arabic, Chinese, French, German, Danish, Italian, Japanese etc
20. country - Country where the movie is produced
21. content\_rating - Content rating of the movie
22. aspect\_ratio - Aspect ratio the movie was made in
23. movie\_imdb\_link - IMDB link of the movie
24. budget - Budget of the movie in Dollars
25. imdb\_score - IMDB Score of the movie on IMDB

**2.2 Data Preprocessing**

The data we obtained are highly susceptible to noisy, missing and inconsistent data due to the huge size and their likely origin from multiple, heterogeneous sources. We mainly used IMDb. The main problem with datasets was missing ﬁelds. To overcome this missing ﬁeld problem we adopted a method which uses a measure of central tendency for the attribute. We used both mean and median as central tendency. Then removed duplicate items.

**2.3 Data Transformation and Features**

In this step, data is transformed so that the regression process may be more efﬁcient and easier. Dataset is mixed with both nominal and numeric attributes, but for a regression process, we need all attributes to be numerical. We used a measure of central tendency of Box ofﬁce revenue to convert corresponding nominal attributes to numerical. Training to test ratio is kept as 3:1

|  |  |
| --- | --- |
| Type | Feature |
| Numeric | Actor facebook likes, num voted users, cast total facebook likes, num user for reviews, budget, director facebook likes,imdb score,  Movie facebook likes. |
| Categorical | Actor name, language, country |

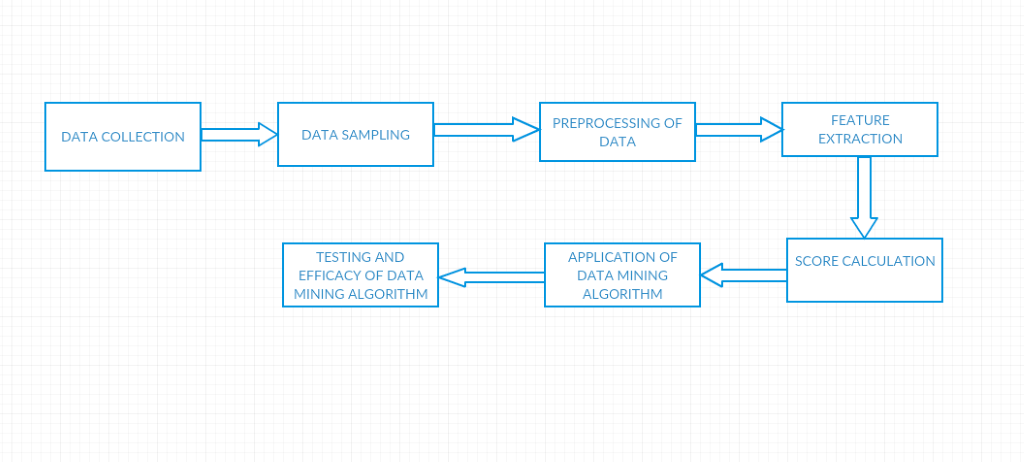
Numerical Features Preprocessing:

Numerical data has been scaled to 0 – 1 by using fitting in Standard scaler. As data had wide range of values, using 0-1 scaling was very helpful.

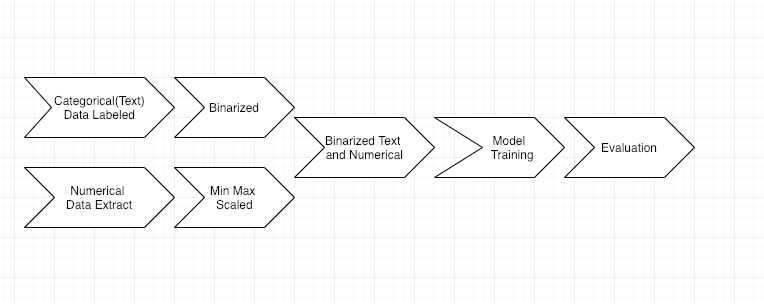
Textual Features Preprocessing:

Textual data has been labeled for each column separately. Each column was assigned label for each distinct feature. Each textual data column has been labeled and then they have been transformed to binary form.

**3. Data Flow Diagram**



**4. Architecture**



**5 REGRESSION MODEL**

Supervised learning technique is adopted for this project. We are using four models to predict the revenue and we will compare the performance of the different methods.

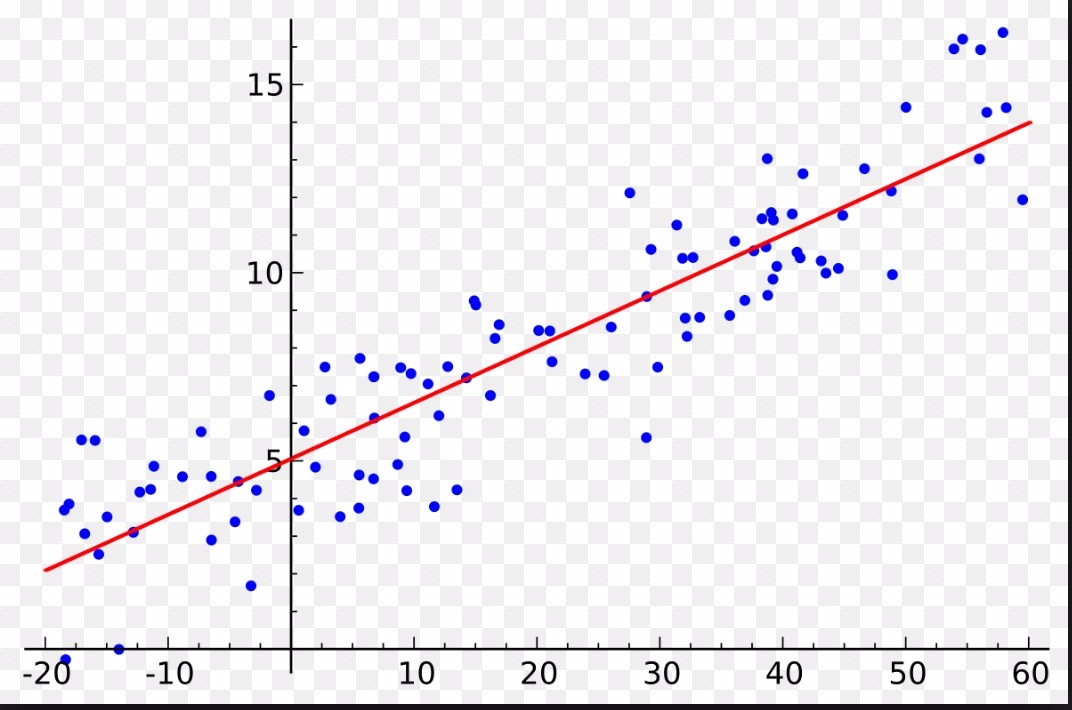
The machine learning algorithms we have used are Linear Regression, Logistic Regression, SVM, Linear SVM,

**5.1 Linear Regression Model**

In our ﬁrst model, we use standard least-squares linear regression. To do this, we intend to use stochastic gradient descent. Once we have trained a set of feature weights, we could then generate gross revenue predictions as follows:

Gross = θ0+θ1\* F1+θ2\* F2+ ... + θn\* Fn

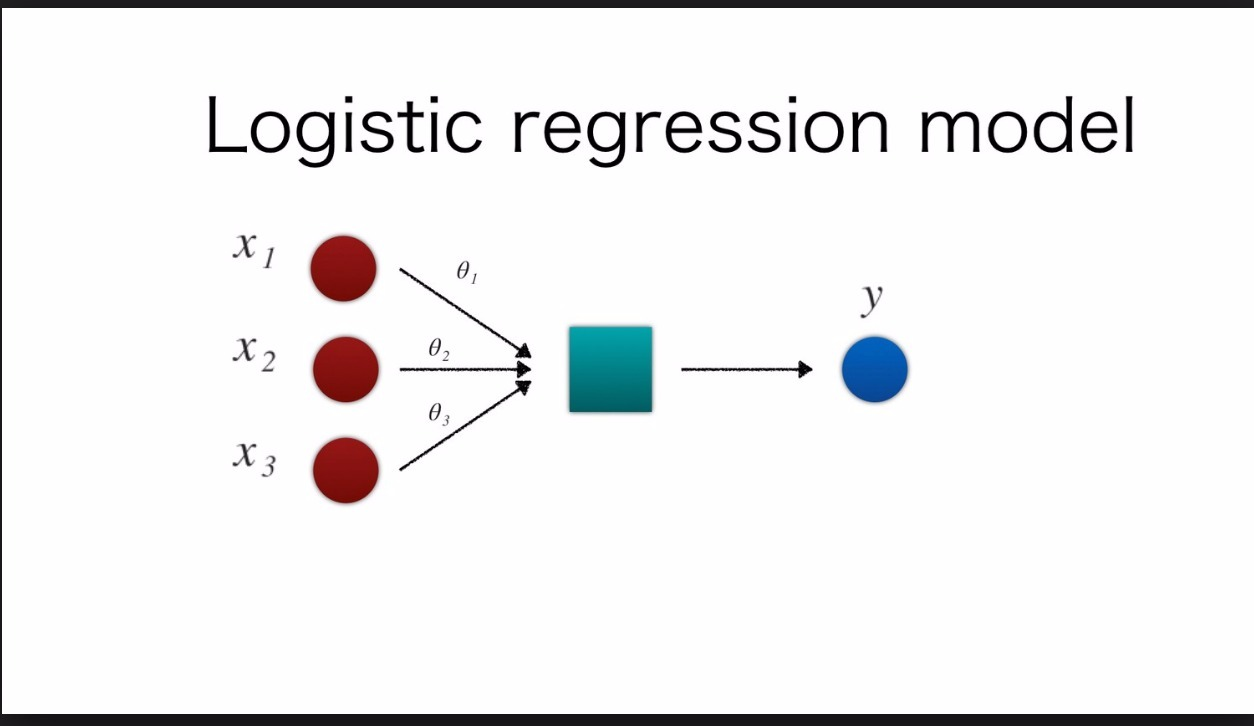
where θi are the weights, Fi are the features, and n is the number of features.



**5.2 Logistic Regression model**

We used logistic regression as our second prediction model. Since logistic regression can build a multi-class model having Linear weights. And we can compare these to the feature weight obtained from the linear regression model. We needed to formulate the regression problem as classiﬁcation problem to apply the logical regression. By splitting the range of the target variable into a two buckets of equal size, we constructed it as a classiﬁcation problem we considered the threshold value to 50000000 and value above this is bucket 1 and below are

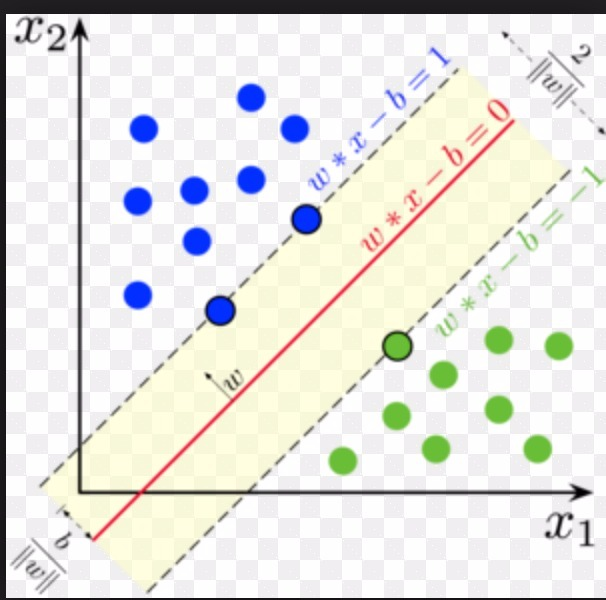
bucket 2. For determining our classes, with the help of histogram we drew from movie revenues, we were able to create different buckets for prediction which were continuous ranges of movie revenues. We created buckets such that it covers the entire sample space



**5.3 Support Vector Machine (SVM)**

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges However, it is mostly used in classification problems. SVM is like a sharp knife – it works on smaller datasets, but on them, it can be much more stronger and powerful in building models.

SVMs can also be applied to regression problem. SVM Regression tries to ﬁnd a function f(x) that has at most deviation from the actually obtained targets y for all the training data, and at the same time is as ﬂat as possible. SVM Regression does not care about errors as long as they are less than ε. We used a linear kernel function to map the data into a high dimensional feature space where linear regression performed. Since Training data is not large as compared to the number of features, we used a linear kernel function Hyperparameter C optimization was done using grid search. A grid search is an exhaustive search through a manually speciﬁed subset of the hyperparameter space.

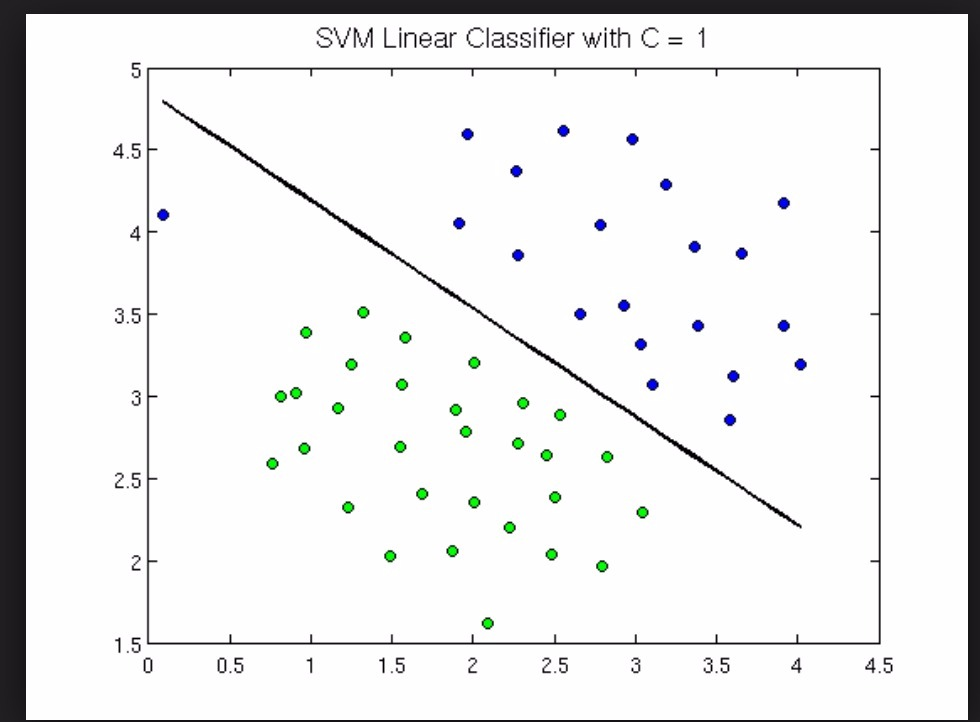


**5.4 Linear Support Vector Machine (SVM):**

Linear SVM is the newest extremely fast machine learning (data mining) algorithm for solving multiclass classification problems from ultra large data sets that implements an original proprietary version of a cutting plane algorithm for designing a linear support vector machine. Linear SVM is a linearly scalable routine meaning that it creates an SVM model in a CPU time which scales linearly with the size of the training data set. Our comparisons with other known SVM models clearly show its superior performance when high accuracy is required.

Linear Support Vector Classification.

Similar to SVC with parameter kernel=’linear’, but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.



**6. Conclusion:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alogorithm | AccuracyScore | Precision | recall | F1 score |
| Logistic | 0.87 | 0.85 | 0.96 | 0.90 |
| SVM | 0.86 | 0.87 | 0.96 | 0.91 |
| Linear SVM | 0.84 | 0.85 | 0.96 | 0.90 |

For this dataset logistic, SVM,Linear SVM with binning mechanism gives better accuracy score when compare to linear.The Accuracy score for Linear is 0.50