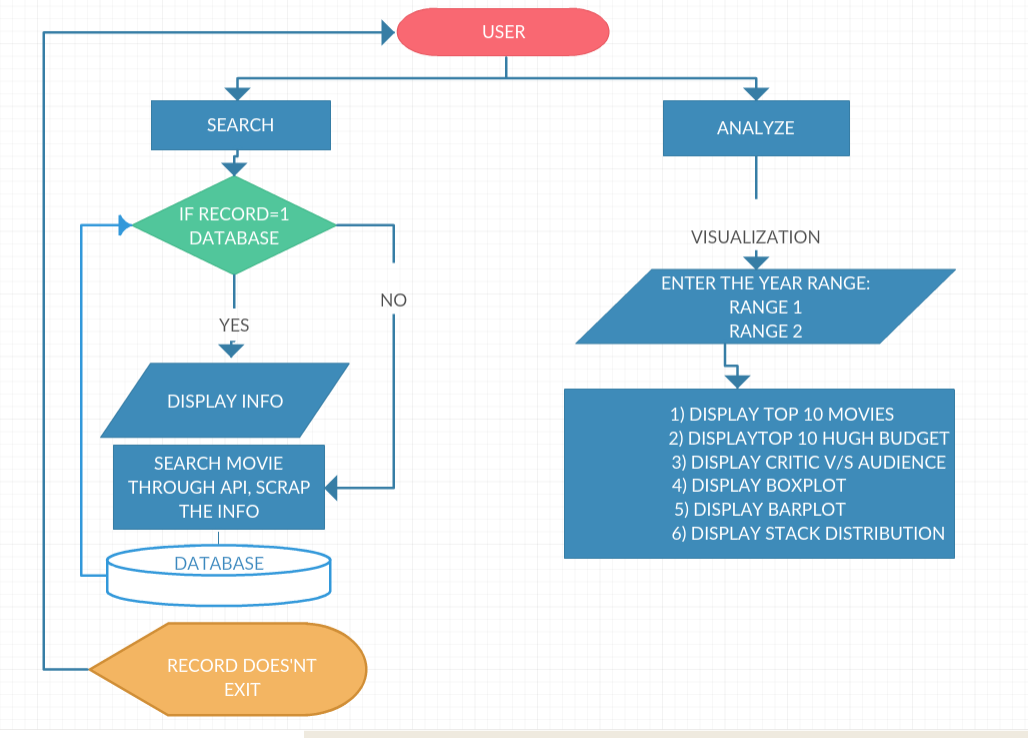
**IMDb Web Scraping and Data Analysis**

**Group Members:- Shehzada AlamSameer PophaliShreyas WankhedeSagar BhutadaShehnaaz Shareen**

**Background:**

* The Internet Movie Database (IMDb) is one of the world’s most popular sources for movie, TV and celebrity content with more than 100 million unique visitors per month.
* IMDb has huge collection of movies database that includes various details of movies along with different ratings and user reviews.
* This movie reviews affects everyone from audience, film critics to the production company.
* Idea of our project is to form a model to predict what are the sentiments of movies based on user reviews and Analyse the Data according to different categories.
* Dataset: 3500+ records and has 10 columns.

**Design:**



**Solution Details:**

* Extracting all the details of movies like IMDBID,Title,Genre, Year, Audience rating, Critics rating, Budget and Reviews using 'OMDB API' and webscrapping.
* Established the Database Connection as we are storing this movie data in MySQL.
* While searching the movie detail if the entry is not present then fetch the detail from imdb through webscraping and with API, insert the record in database and display result back to the user.
* For analysis extract the data from database into dataframe and visualize the data to get some insights
* Sentiment Analysis based on User Reviews and created a new column(polarity) which includes this Labels (Positive and Negative).
* “Sentiment analysis is an important research area that identifies the people’s sentiments, opinions and emotions underlying a text.”
* To extract this sentiments we have computed the polarity of the given reviews (whether the text is positive or negative).
* We have used “Unsupervised lexicon based method”, which are dictionaries or vocabularies of polar words specially constructed for sentiment classification task.
* The system uses VADER (Valence Aware Dictionary and sEntiment Reasoner) lexicon-based sentiment analysis tool that not only tells about the positivity, negativity, neutral and compound score between -1 to +1 but gives positive or negative sentiment of reviews based on this score.

**Model Building:**

* In order to make sense to our machine learning algorithm we have converted each review to a numeric representation which is called 'Vectorization'.
* The system uses TF-IDF Vectorizer (Term Frequency-Inverse document frequency) that transforms a count matrix to a normalized frequency representation in float.
* Splitting the movie data into Train and Test set (80-20 ratio).
* SVM: The objective of a Linear SVC (Support Vector Classifier) is to fit the data, returning a best fit hyperplane that divides or classifies the data.
* Accuracy= 79.61%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **f1-score** | **Support** |
| Negative | 0.77 | 0.82 | 0.79 | 371 |
| Positive | 0.83 | 0.77 | 0.80 | 404 |
| Total | 0.80 | 0.80 | 0.80 | 775 |

**Implementation:**

**Libraries Used:**

* pandas
* sklearn
* numpy
* BeautifulSoup
* vaderSentiment
* SentimentIntensityAnalyzer
* seaborn

**Functions Created:**

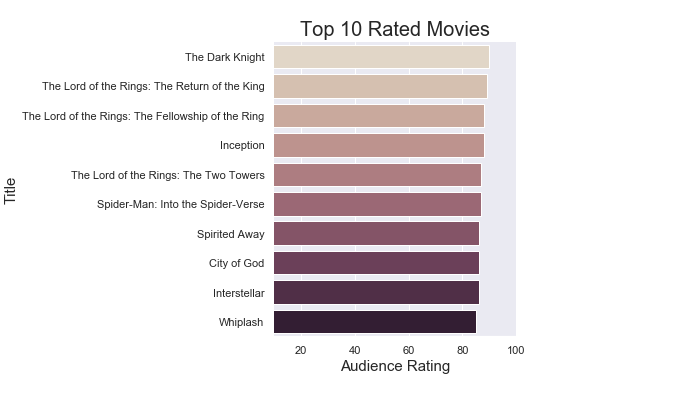
* sentiment\_lexicon():Using Vader lexicon to get the polarity based on threshold value.
* UserReview\_SentimentAnalyzer():Sentiment analysis based on user reviews.
* Predict\_Sentiment():This function will analyse user review and predict the polarity.
* Extract\_Budget\_UserReview():To extract Budget and Reviews from IMDB website.
* getMovieData():Extracting movie details from API.
* DatabaseConnection():To establish connection with MySQL.
* DataIntoDatabase():This Function will put the data extracted from API and from webscraping into movie database.
* getDataFromDatabase():This function will fetch the data from database from the title provided by the user.
* DisplayMovieInfo():This is use to display info about the movie title provided by the user, at the same time if the movie title doesn't exist the it will make an entry into the database.
* getDataByYear():This Function will fetch the data by year.
* DataAnalysis():This fucntion will fetch the data from the database & process it while formatting the data in long format
* DisplayTheDetails():This function will be called after the object is created and its link with mulitple functions.

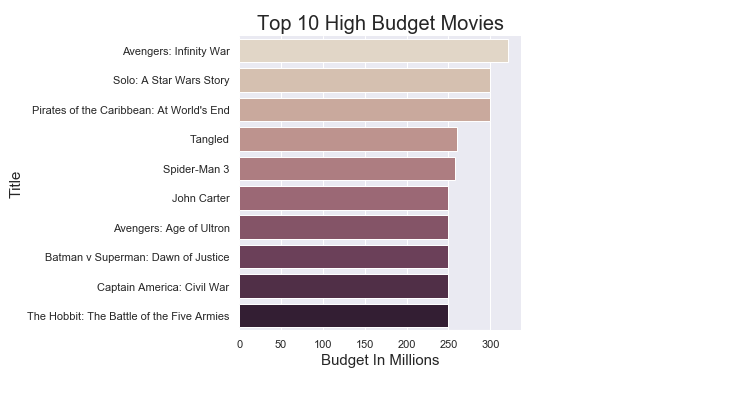
**Prerequisite:**

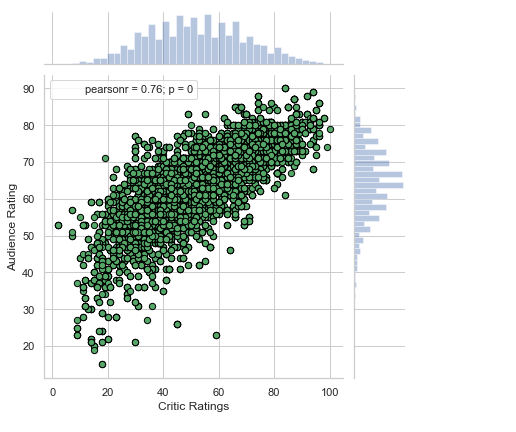
* MySQL
* nltk.download(‘vader\_lexicon’)
* Django Framework
* API Key (omdb)
* Access to imdb website.

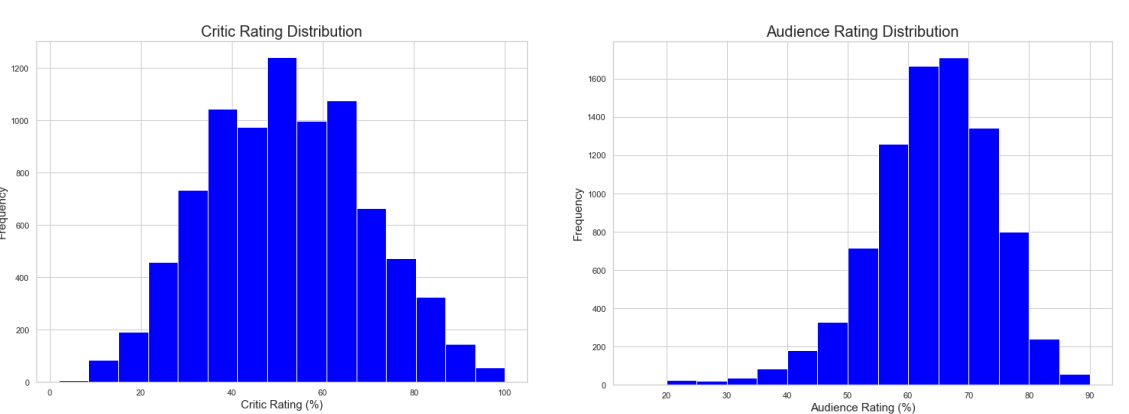
**Challenges:**

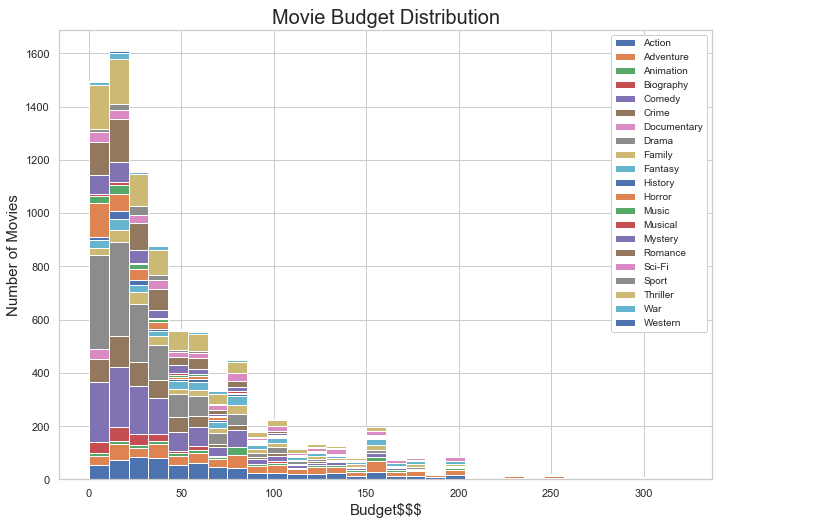
* Using omdb API we were not able to fetch Budget and User Reviews hence we have scrapped the data from imdb website using imdb id.
* While fetching 'Budget' from imdb website the amount were present in different currency format so we have converted the currencies in USD by using CurrencyConverter package.
* After getting polarity for all the user reviews, this reviews was needed to be converted in numerical representation to fit the classification model.
* While rendering seaborn graphs on Django Framework we were getting several response errors and were not able to display the plots.

**Data Analysis & Visualization:1) Top 10 Rated Movies:** 

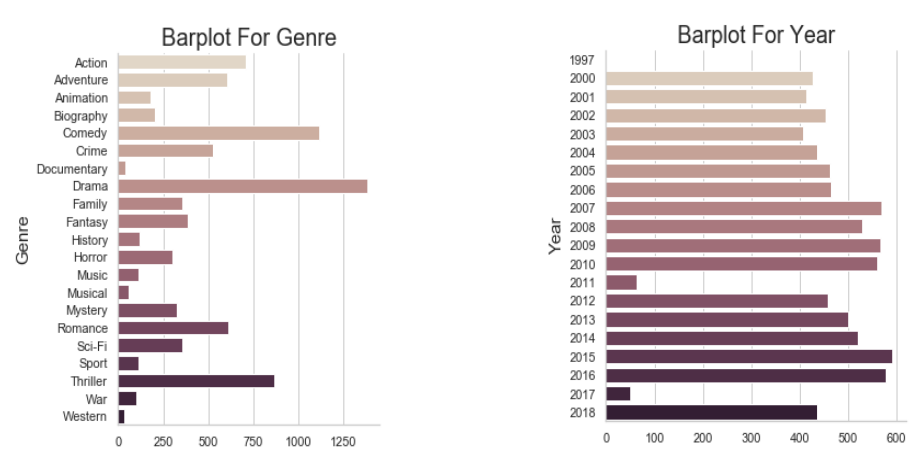
**2) Top 10 High Budget Movies:** 

**3) Critic Vs Audience Rating:** 

**4) Distribution of Critic or Audience Rating:**

**5) Stack Distribution of Movie Budget:** 

**6) BARPLOT:**



**7) BOXPLOT:**

