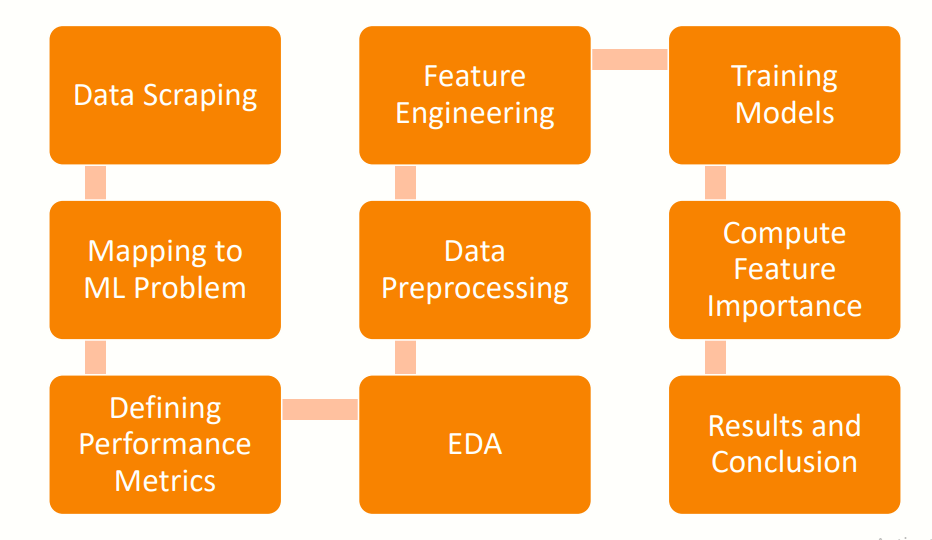
**Web Scraping and Sentiment Analysis of the Movie Reviews**

**Packages used:**

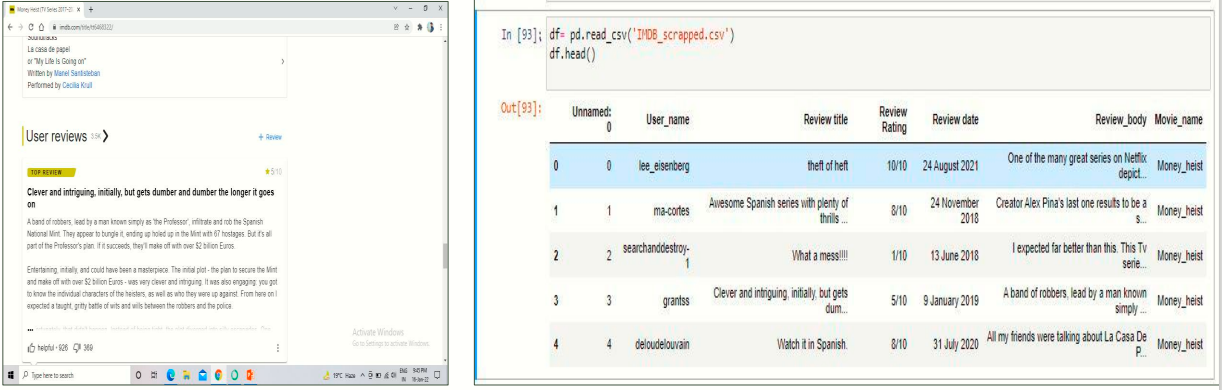
Web Scraping: Selenium and BeautifulSoup

Other: pandas, sklearn, numpy, xgboost etc.

**Approach:**



**1) Data Scraping:** Selected TV series: Money Heist Data regarding the review text, review ratings etc. are scrapped from the IMDB website. For this scraping libraries like Selenium and BeautifulSoup are used. Scrapped data is stored to 'IMDB\_scrapped.csv‘ csv file.



IMDB website reviews Scrapped Data

**2) Mapping to ML Problem:**

• It is multiclass classification problem with 3 classes: Positive,

Neutral, Negative.

• Scrapped data contains text features: review\_title,

review\_text and review\_rating in the range 1 to 10.

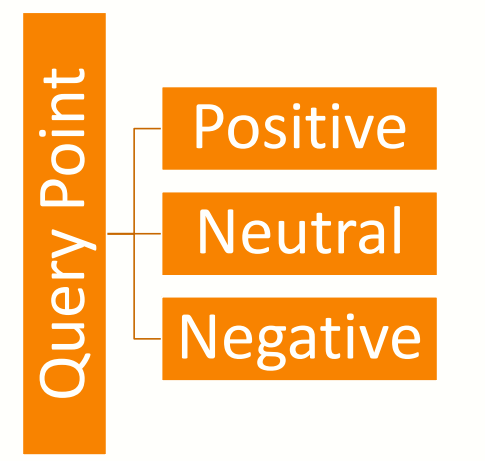
• We have not given the class labels to the data point yet.

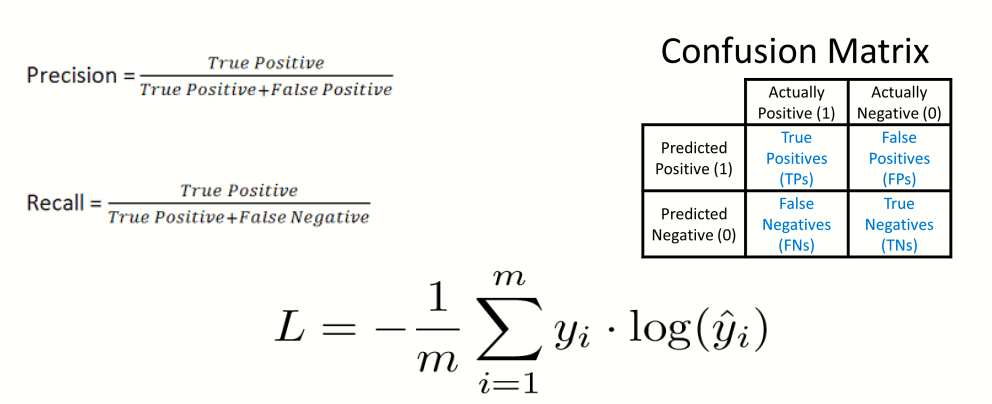
First we read some of the reviews and then decide the

review’s polarity whether Positive, Neutral or Negative.

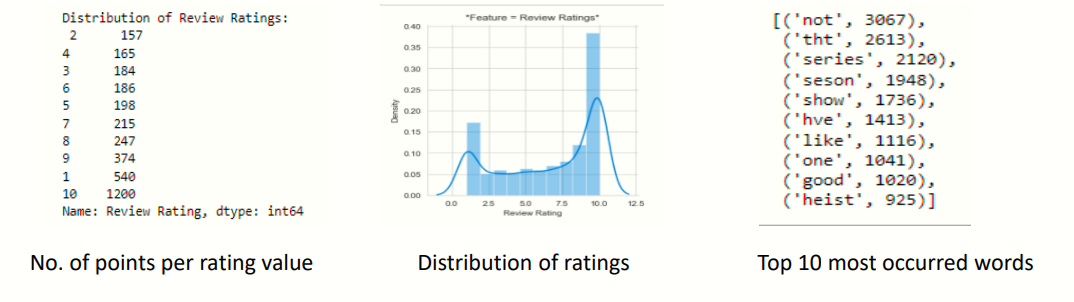
• We have to build model which could solve a multiclass

classification. So given a Query point it has to give output class from [Positive, Neutral, Negative]



**3) Defining Performance Metrics:** Type of problem: Multiclass Classification Considering it is multiclass classification task metrics like: f1\_micro score, Precision, Recall, Multiclass Logloss (Cross Entropy) and Confusion Matrix are taken for measuring performance of the model. 

**4) EDA:**



After reading some text reviews we will be assigning polarity class labels as:

**Negative:** The reviews with rating 3 and less will be considered as the Negative review for this analysis task. There are 881 Negative reviews.

**Neutral:** The reviews with rating between 4 and 7 will be considered as the Neutral review for this analysis task. There are 764 Neutral reviews.

**Positive:** The reviews with rating 8 and above will be considered as the Positive review for this analysis task. There are 1821 Positive reviews.

**5) Data Preprocessing:**

Using some string operations and regex expressions we processed the text data. Also class labels are assigned as Negative: 0, Neutral: 1 and Positive: 2.

1) Removed stop words excluding words like ‘no’, ‘nor’, ‘not’.

2) Removed extra space, new line, tab space

3) Removed special characters

4) Removed all digits

5) Removed words have length<2 and >15.

6) All the words transformed to lower case

7) All NaN strings are replaced with word ‘Money Heist’

8) Preprocessed data stored to csv file

**6) Feature Engineering:**

1) Train – Test split is performed. Assigned 20% datapoints as Test\_data.

2) Tf-idf vectorization performed on ‘review\_title’ feature.

3) Tf-idf vectorization performed on ‘review\_text’ feature.

4) Tf-idf weighted w2v vectorization performed on ‘review\_title’ feature using glove vectors.

5) Tf-idf weighted w2v vectorization performed on ‘review\_text’ feature using glove vectors.

6) Sentiment Scores are calculated for both train and test data.

7) Concatenated all the data to prepare two different sets as:

set (1): tf-idf Vectorized

set (2): tf-idf w2v Vectorized

**7) Training Models:**

• Model\_1: SVC

a) Hyperparameter tuning on set (1) : parameters='kernel‘ and ‘C’ :-

Found best parameters as : C= 0.5 and Kernel= linear

b) Hyperparameter tuning on set(2) : parameters='kernel‘ and ‘C’ :-

Found best parameters as : C= 1 and Kernel= rbf

• Model\_2: XGBoost

a) Hyperparameter tuning on set(1) : parameters= ‘n\_estimators’

Found best parameters as : n\_estimators = 100

b) Hyperparameter tuning on set(2) : parameters= ‘n\_estimators’

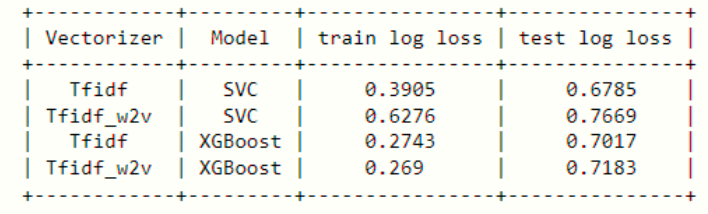
Found best parameters as : n\_estimators = 500

**8) Compute Feature Importance:** Feature importance is computed for each model which shows the words that are most important in deciding the class



**9) Results and Conclusion:**

Result Table:



1) On the acquired data vectorization using tf-idf and SVC with linear kernel gives less test log loss. Also number of misclassified points are also less.

2) The words like 'series', 'chrctr', 'seson', 'show' etc. have high importance as shown in the World Cloud.

3) Even though some words in the text don't have particular meaning, they are contributing in defining the class