Internship Assignment

Name: **Akshay Bhasme**
Mail ID: **akshaybhasme30@gmail.com**

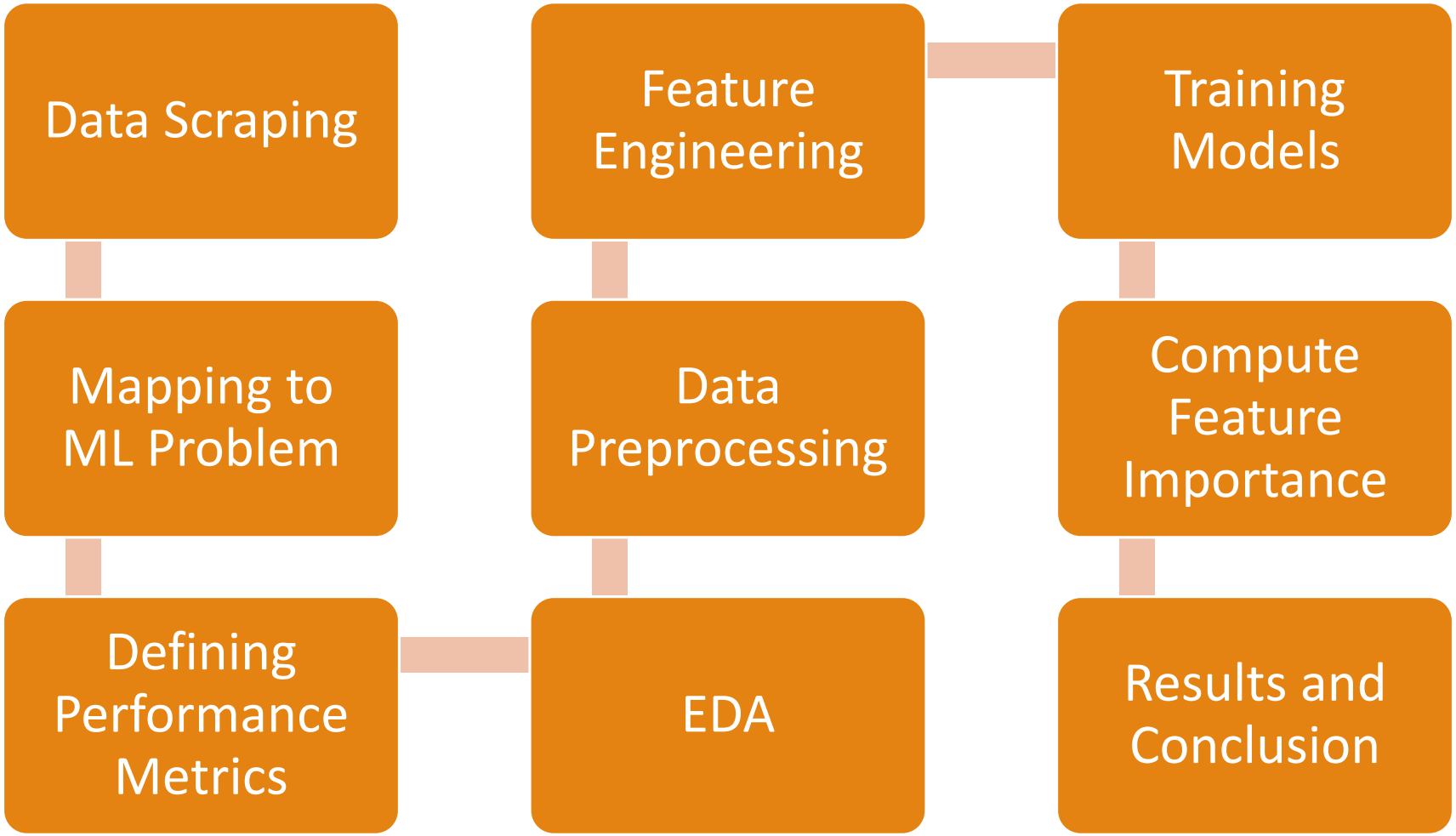
Problem statement –

Create a data story of the online/social media response of a movie or a TV series.

Instructions:

1. Select one recently launched movie or a TV show from Hotstar or Netflix.
2. Extract reviews, tweets, or any relevant text data from social media platforms/websites like Twitter, Facebook, Google etc.
3. Clean the data and create an appropriate schema to store it in a table format(s).
4. Perform EDA and apply relevant ML algorithms if required.
5. Highlight insights/relevant stats and conclude whether the movie/TV series has received a positive/negative or neutral response from the online community.
6. Record your outputs as a presentation or a dashboard.
7. Share the following outputs
    a. PDF file of your presentation OR a dashboard link to an online public library (Example Tableau public or Power BI gallery).
    b. Supporting documents in PDF format – code, data, approach etc.

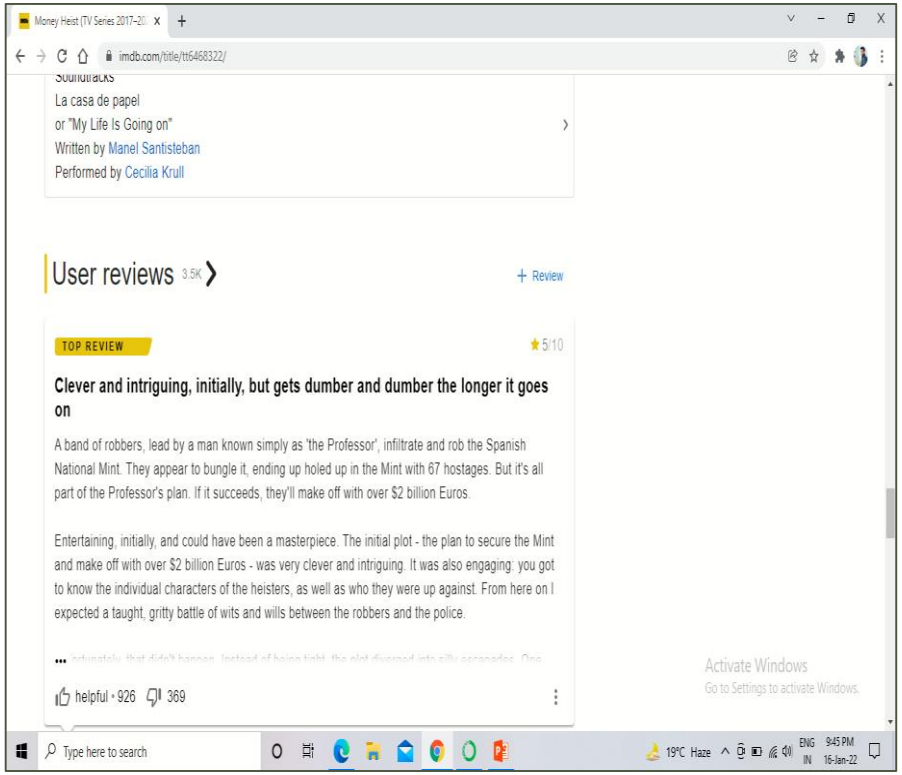
Approach to solve the assignment



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

```
In [93]: df= pd.read_csv('IMDB_scrapped.csv')
         df.head()
```

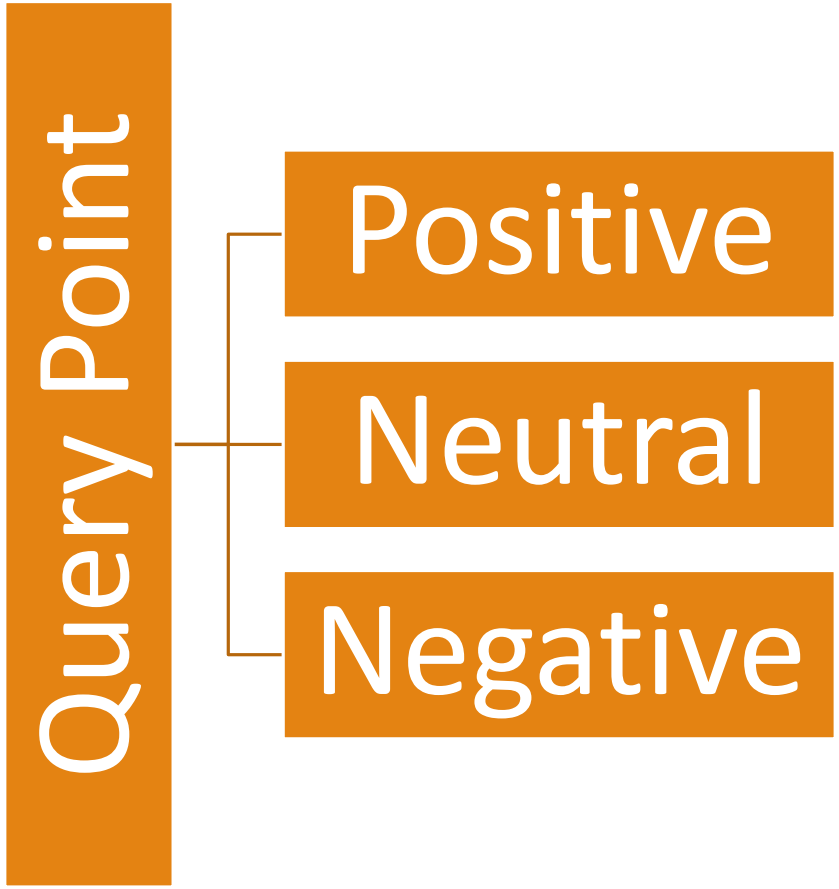
Out[93]:

	Unnamed: 0	User_name	Review title	Review Rating	Review date	Review_body	Movie_name
0	0	lee_eisenberg	theft of heft	10/10	24 August 2021	One of the many great series on Netflix depict...	Money_heist
1	1	ma-cortes	Awesome Spanish series with plenty of thrills ...	8/10	24 November 2018	Creator Alex Pina's last one results to be a s...	Money_heist
2	2	searchanddestroy-1	What a mess!!!!	1/10	13 June 2018	I expected far better than this. This Tv serie...	Money_heist
3	3	grantss	Clever and intriguing, initially, but gets dum...	5/10	9 January 2019	A band of robbers, lead by a man known simply ...	Money_heist
4	4	deloudelouvain	Watch it in Spanish.	8/10	31 July 2020	All my friends were talking about La Casa De P...	Money_heist

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.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

4) EDA:

```
Distribution of Review Ratings:
 2      157
4       165
3       184
6       186
5       198
7       215
8       247
9       374
1       540
10     1200
Name: Review Rating, dtype: int64
```

No. of points per rating value

Distribution of ratings

```
[('not', 3067),
 ('tht', 2613),
 ('series', 2120),
 ('seson', 1948),
 ('show', 1736),
 ('hve', 1413),
 ('like', 1116),
 ('one', 1041),
 ('good', 1020),
 ('heist', 925)]
```

Top 10 most occurred words

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

SVC set(1)
important features

SVC set(2)
important features

XGBoost set(1)
important features

XGBoost set(2)
important features

9) Results and Conclusion:

Result Table:

Vectorizer	Model	train log loss	test log loss
Tfidf	SVC	0.3905	0.6785
Tfidf_w2v	SVC	0.6276	0.7669
Tfidf	XGBoost	0.2743	0.7017
Tfidf_w2v	XGBoost	0.269	0.7183

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.