NLP Sentiment Analysis on IMDB Dataset

Capstone Project , 8th October 2021

Ramasubramanian Sairaman

Table of Contents

- Engagement Background
- NLP – Quick Overview
- Data Science Process
- Capture Business Problem
- Understanding the Data
- NLP sentiment Analysis – Solution Flow
- Data Loading & Pre-Processing
- Feature Selection & Visualization
- Model Training & Evaluation
- Model Recommendation
- Next Steps

Engagement Background

NLP is an emerging area with a huge focus. The buzz of NLP in the market is growing in an exponential manner which is expected to touch the **mark of $ 16 billion by 2021** with the compound growth rate of **16 % annually**.

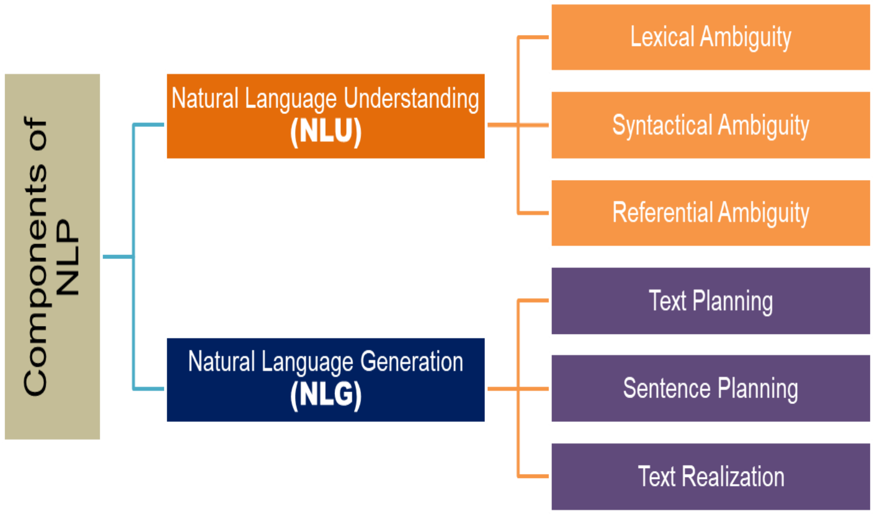
A Leading **AI & ML Organization** is expanding its footprint in the area of **NLP** with specific focus on **Sentiment analysis** and has an aggressive roadmap of 3 years to capture a significant market share.

NLP – Quick Overview

What is NLP?

The field of study that focuses on the interactions between human language and computers is called Natural Language Processing or NLP for short. It sits at the intersection of computer science, artificial intelligence, and computational linguistics (Wikipedia).

Components of NLP



Components of NLP

Natural Language Understanding (NLU)

Lexical Ambiguity

Syntactical Ambiguity

Referential Ambiguity

Natural Language Generation (NLG)

Text Planning

Sentence Planning

Text Realization

Natural language understanding (NLU) involves transforming human language into a machine-readable format.

Natural Language Generation (NLG) involves

Text planning − It includes retrieving the relevant content from knowledge base.

Sentence planning − It includes choosing required words, forming meaningful phrases, setting tone of the sentence.

Text Realization − It is mapping sentence plan into sentence structure.

NLP – Few Use Cases

Automatic Summarization

Intelligently shortening long pieces of text

Named entity recognition

Locate and classify named entities pre-defined categories such as the organizations; person names; locations etc.

Sentiment analysis

To identify, for instance, positive, negative and neutral opinion form text or speech widely used to gain insights from social media comments, forums or survey responses

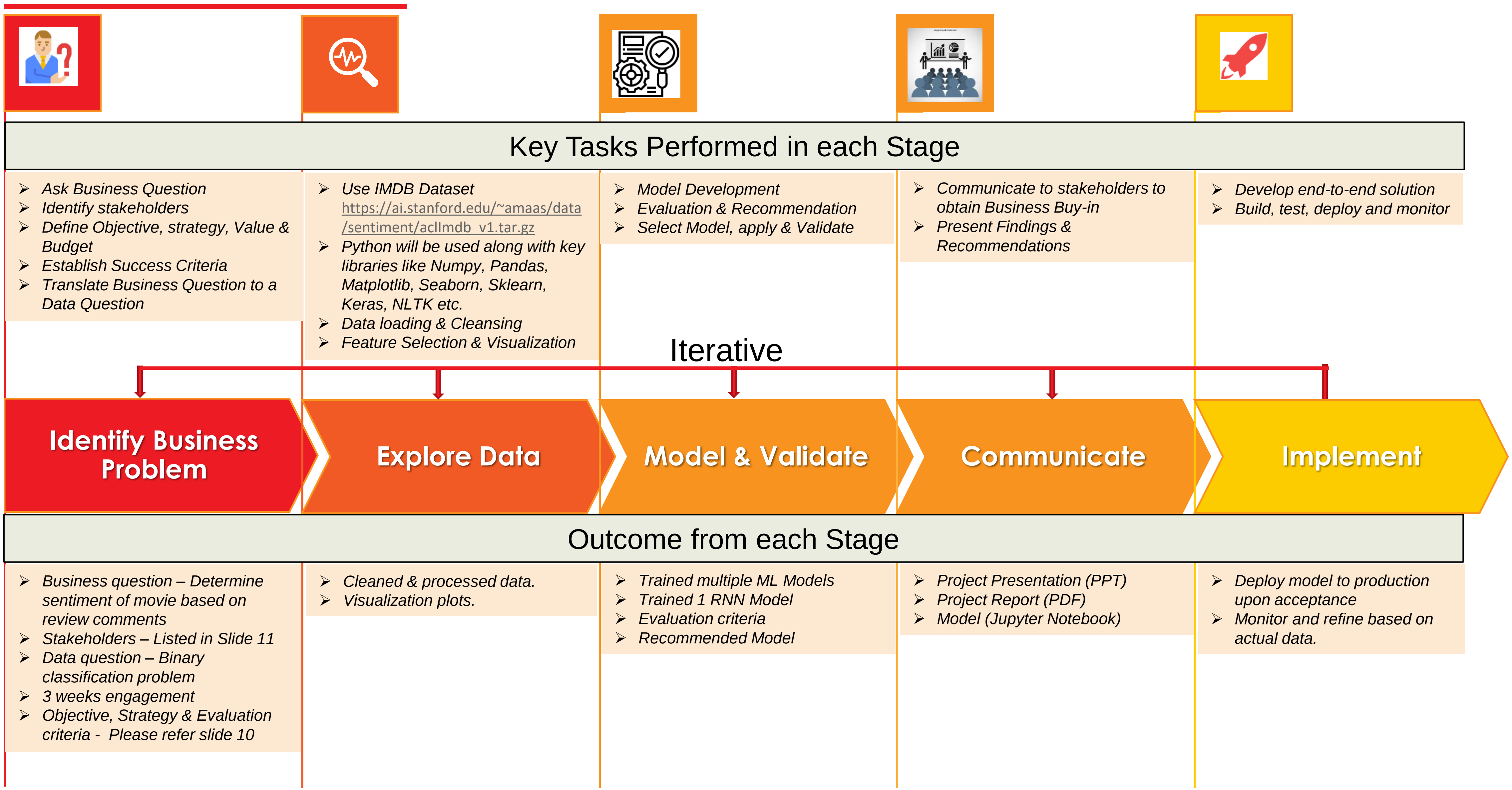
Speech recognition

Enables computers to recognize and transform spoken language into text – dictation – and, if programmed, act upon that recognition – e.g. in case of assistants like Google Assistant Cortana or Apple's Siri

Topic segmentation

Automatically divides written texts, speech or recordings into shorter, topically coherent segments and is used in improving information retrieval or speech recognition

Data Science Process



Key Tasks Performed in each Stage

Identify Business Problem

- *Ask Business Question*
- *Identify stakeholders*
- *Define Objective, strategy, Value & Budget*
- *Establish Success Criteria*
- *Translate Business Question to a Data Question*

Explore Data

- *Use IMDB Dataset https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz*
- *Python will be used along with key libraries like Numpy, Pandas, Matplotlib, Seaborn, Sklearn, Keras, NLTK etc.*
- *Data loading & Cleansing*
- *Feature Selection & Visualization*

Model & Validate

- *Model Development*
- *Evaluation & Recommendation*
- *Select Model, apply & Validate*

Communicate

- *Communicate to stakeholders to obtain Business Buy-in*
- *Present Findings & Recommendations*

Implement

- *Develop end-to-end solution*
- *Build, test, deploy and monitor*

Iterative

Identify Business Problem → Explore Data → Model & Validate → Communicate → Implement

Outcome from each Stage

Identify Business Problem

- *Business question – Determine sentiment of movie based on review comments*
- *Stakeholders – Listed in Slide 11*
- *Data question – Binary classification problem*
- *3 weeks engagement*
- *Objective, Strategy & Evaluation criteria - Please refer slide 10*

Explore Data

- *Cleaned & processed data.*
- *Visualization plots.*

Model & Validate

- *Trained multiple ML Models*
- *Trained 1 RNN Model*
- *Evaluation criteria*
- *Recommended Model*

Communicate

- *Project Presentation (PPT)*
- *Project Report (PDF)*
- *Model (Jupyter Notebook)*

Implement

- *Deploy model to production upon acceptance*
- *Monitor and refine based on actual data.*

Capture Business Problem

- A **Leading AI & ML Organization** is expanding its footprint in the area of **NLP Sentiment analysis**

- The Organization has engaged the services of Sai Science Pte Ltd and is looking for a **robust sentiment analysis model** to be built as part of the pilot phase of the program using dataset pertaining to movies review.

Corresponding Data Problem/Question

- **Binary classification problem**
  - Predict whether the sentiment of the movie review is positive or negative….

- Target variable : Label (Sentiment)

Capture Business Problem – Summary of Activities & Deliverables

The **following activities** will be performed as part of the engagement

✓ Data Loading & Pre-processing to clean data

✓ Vectorization, Feature selection & visualization

✓ Model training & Evaluation

✓ Recommending the best model& testing couple of reviews.

Deliverables & Tools used: -

Development is done using Python. As part of the engagement, the following will be delivered.

- **Project Presentation** – MS Power Point
- **Project Report** – MS Word
- **Codebase** – Jupyter Notebook. 3 Notebooks have been created.
    1. Data loading, Cleansing & Feature Selection
    2. ML Models including ensemble techniques
    3. RNN Model
- **Key Libraries Used** – Numpy, Pandas, Seaborn, Matplotlib, NLTK, Keras, sklearn, pickle.

Capture Business Problem – Objective, Strategy & Success Criteria

Objective: - To build a Model to predict Sentiment based on the Movie review comments.

Strategy & Evaluation Criteria:-

- To Evaluate multiple models using the evaluation parameters identified and recommend the one with the best overall score.
- Accuracy, Precision, Recall and ROC_AUC scores are the evaluation parameters used for the various models that will be trained and tested.

Establish Success Criteria:

- To validate the performance metrics of the different models using the agreed evaluation parameters
- The recommended model should have an overall score of at least 85%.

Capture Business Problem – Key Stakeholders

Key Client Stakeholders	Vendor Stakeholders
Client Engagement Director	Engagement Director
Client Project Manager	Project Manager
Client BA / SME	Lead Data Scientist
Business Sponsor – Head of New Business	Data Architect
Technology Sponsor – Head of Technology	Developers

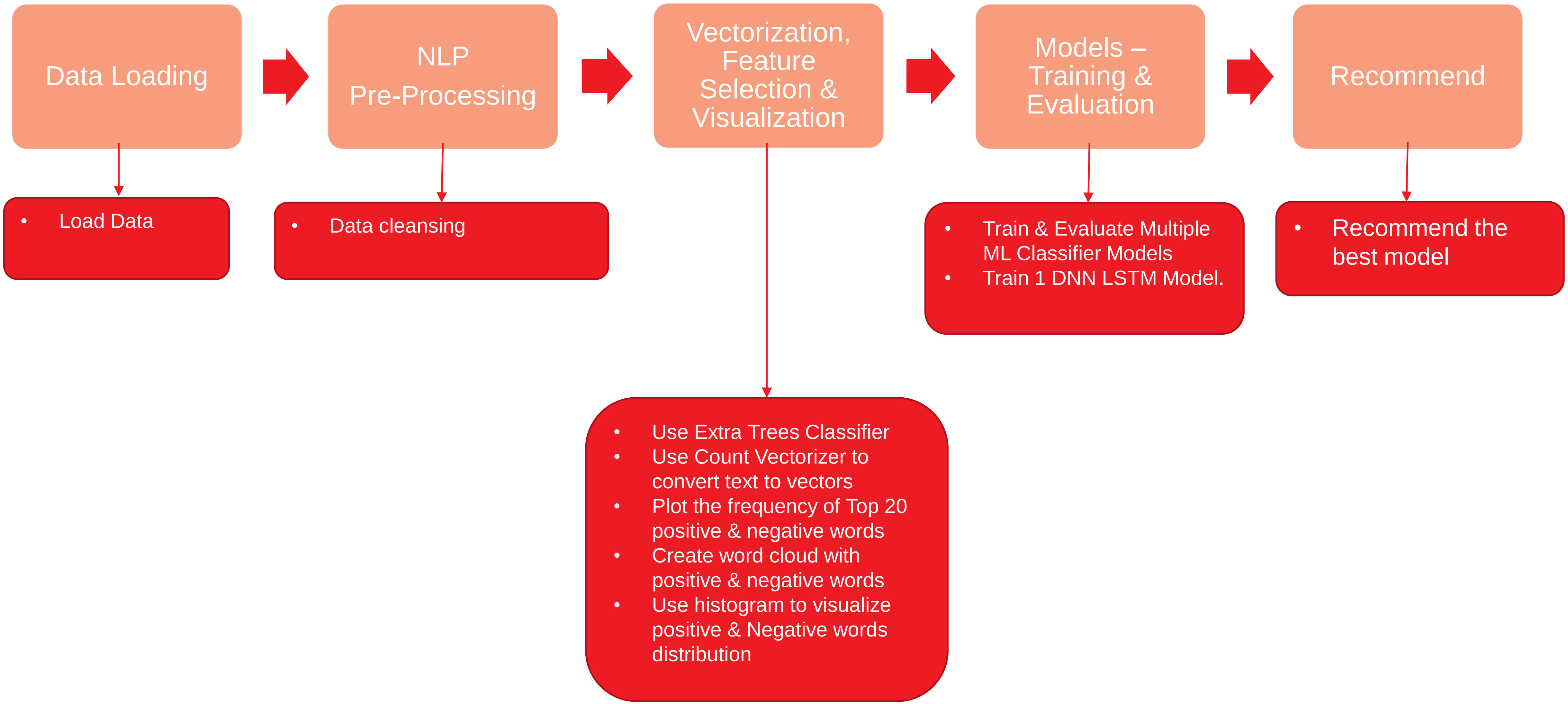
Capture Business Problem - Key Assumptions

- The client team would make themselves available to clarify any questions on the data set. (2 sessions of 2 hours each have been planned to tackle such questions)
- The scope of the project covers only supervised data.
- As discussed, and agreed upfront, this is a **3-weeks** engagement
- The data set is complete and a significant representation.
- The output will be the codebase (Jupyter Notebook), Power point presentation and a project report in MS word.

Explore Data - Understanding the data

- ✓ **Data Source (IMDB Data) –** https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz This dataset contains highly polar movie reviews split in to
- ✓ **Training dataset** – Stored as individual files in training folder
- ✓ **Testing dataset** – Stored as individual files in testing folder
- ✓ **# of records**
  - ✓ **Training Data set – 25000**
  - ✓ **Testing Data set - 25000**
- ✓ **Type of Data – Movie Sentiment Analysis –** To predict whether the sentiment is positive or negative based on Movie review.

NLP Sentiment Analysis – End-to-End solution Flow



Data Loading

- Load Data

NLP Pre-Processing

- Data cleansing

Vectorization, Feature Selection & Visualization

- Use Extra Trees Classifier
- Use Count Vectorizer to convert text to vectors
- Plot the frequency of Top 20 positive & negative words
- Create word cloud with positive & negative words
- Use histogram to visualize positive & Negative words distribution

Models – Training & Evaluation

- Train & Evaluate Multiple ML Classifier Models
- Train 1 DNN LSTM Model.

Recommend

- Recommend the best model

Data Loading & Pre-Processing

Data Loading

Data is loaded from the Training & Testing data sets using Python.

Training data

\train

\train\pos --- Positive review – 12500 records

\train\neg --- Negative reviews – 12500 records

Testing data

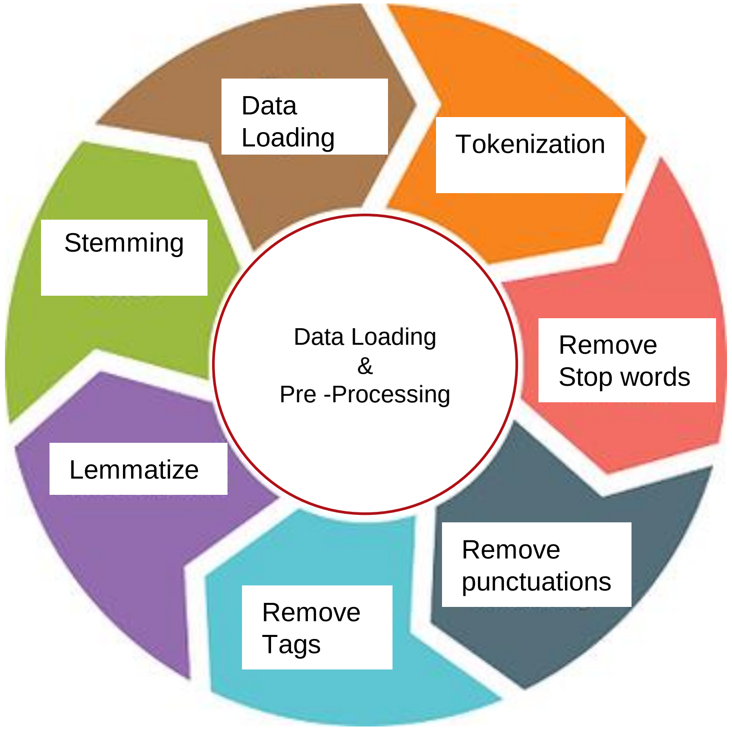
\test

\test\pos --- Positive review – 12500 records

\test\neg --- Negative reviews – 12500 records

NLP Pre-Processing

- Data cleansing done to
  - ✓ Remove Stop Words
  - ✓ Remove Punctuations
  - ✓ Remove Tags
  - ✓ Lemmatization
  - ✓ Stemming



Vectorization, Feature selection & Visualization

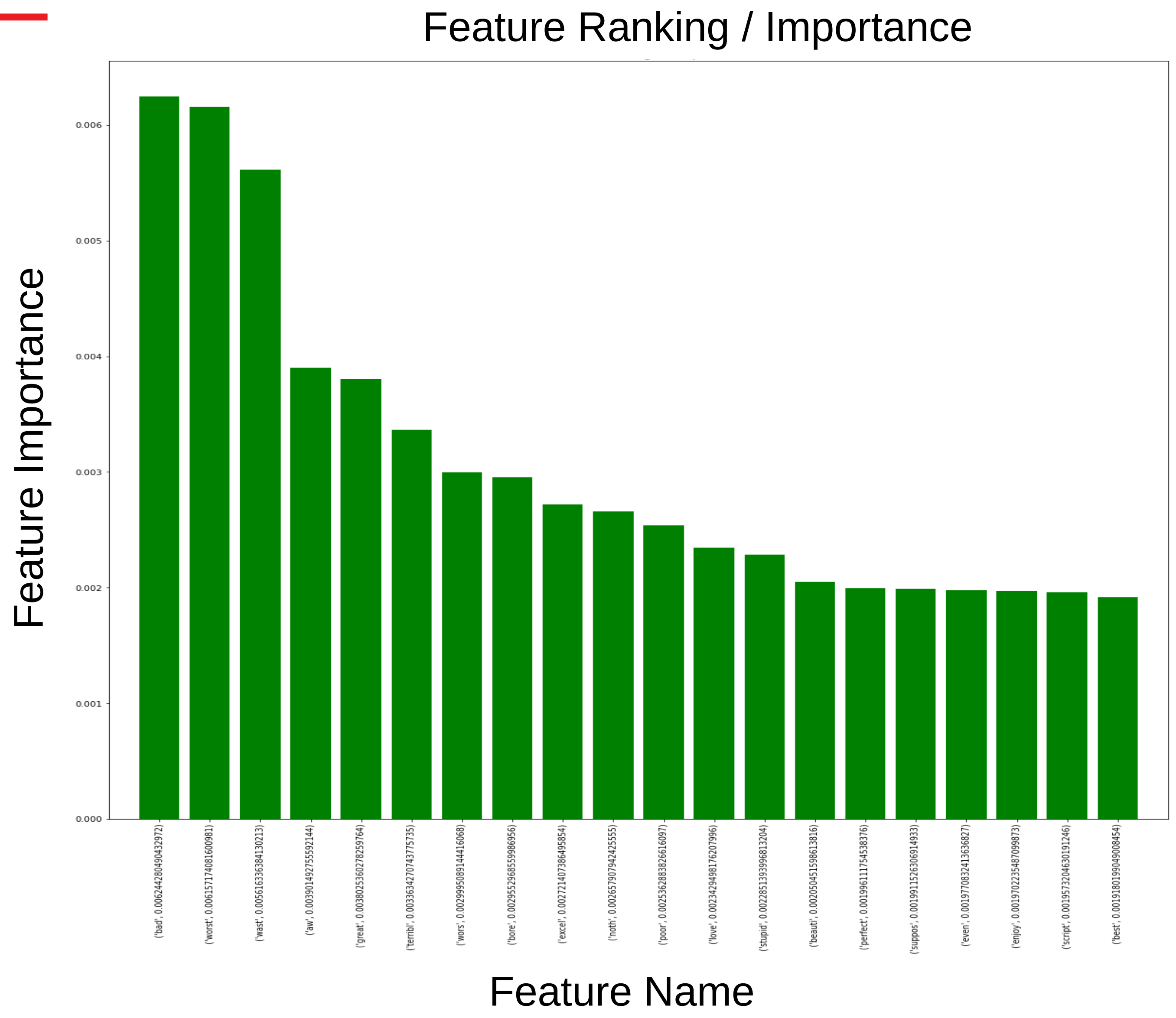
Feature Selection

Vectorization is the process of converting text to vector of numbers.

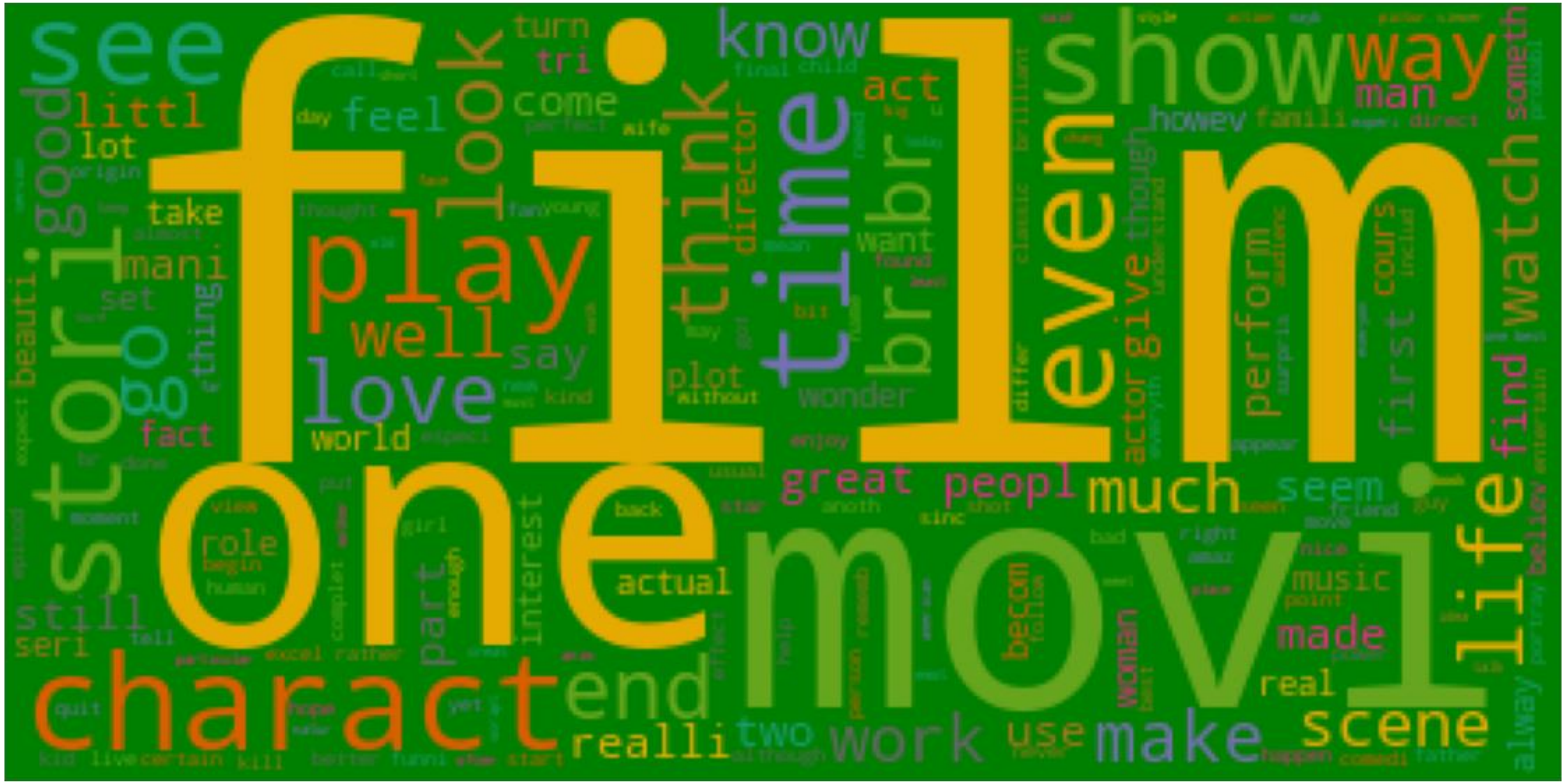
Feature selection is a process where we automatically select those features in the data that contribute most to the prediction variable or output of interest.

Method Chosen - An extra-trees classifier. This class implements a meta estimator that fits several randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting

Feature Ranking / Importance – Top 20

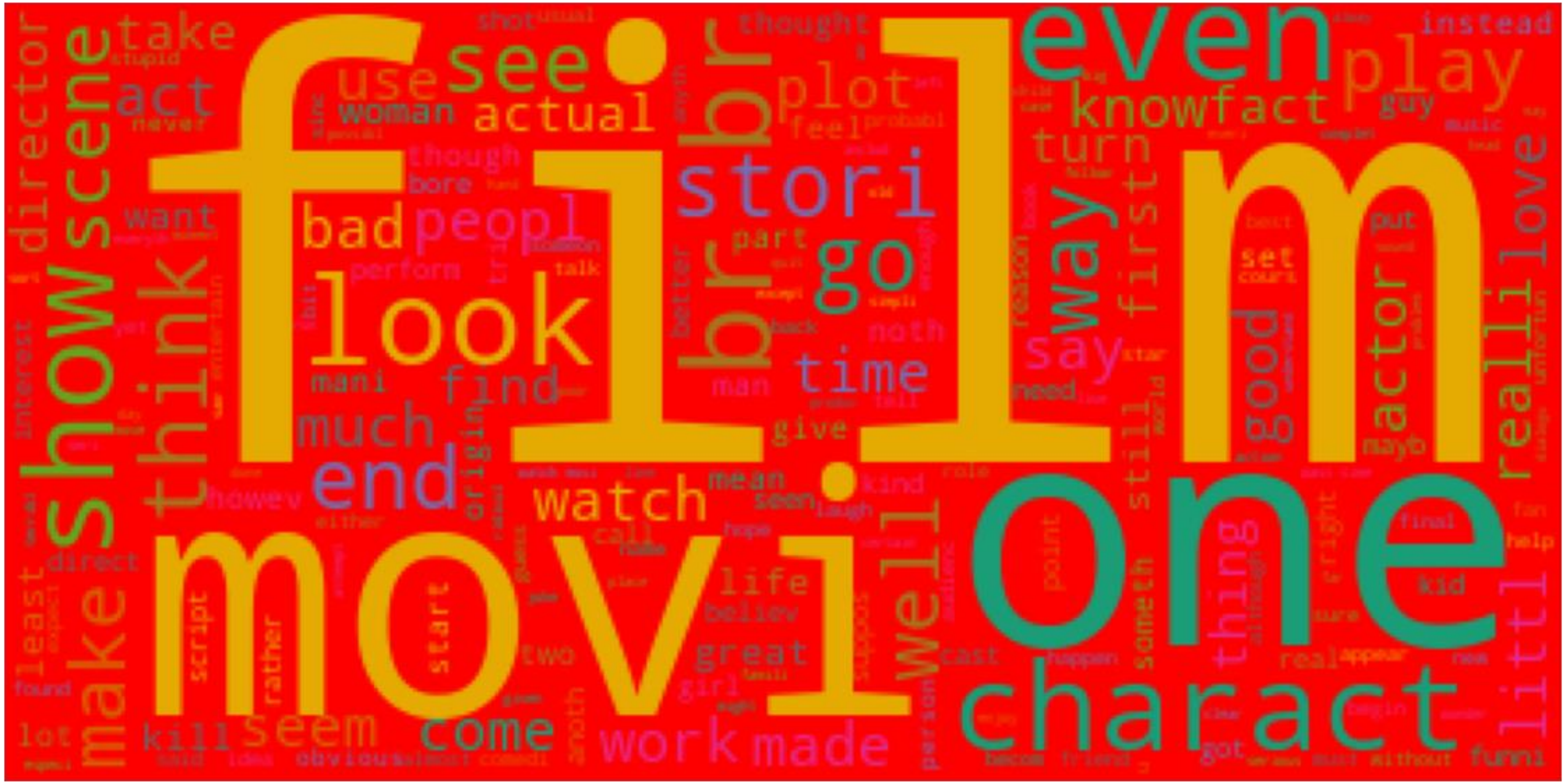


Word Cloud – Positive words



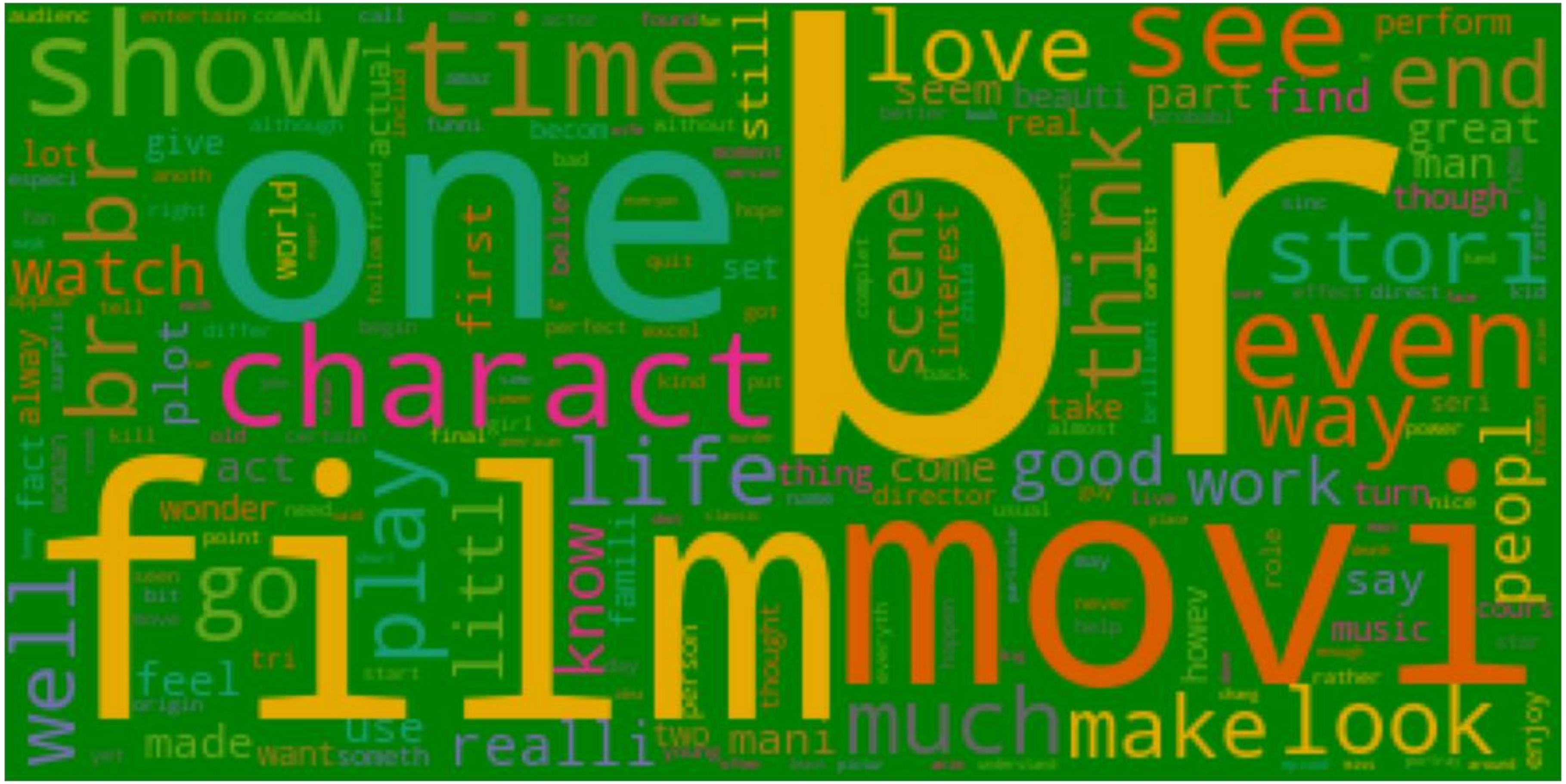
Word cloud is a technique for visualising frequent words in a text where the size of the words represents their frequency.

Word Cloud – Negative words



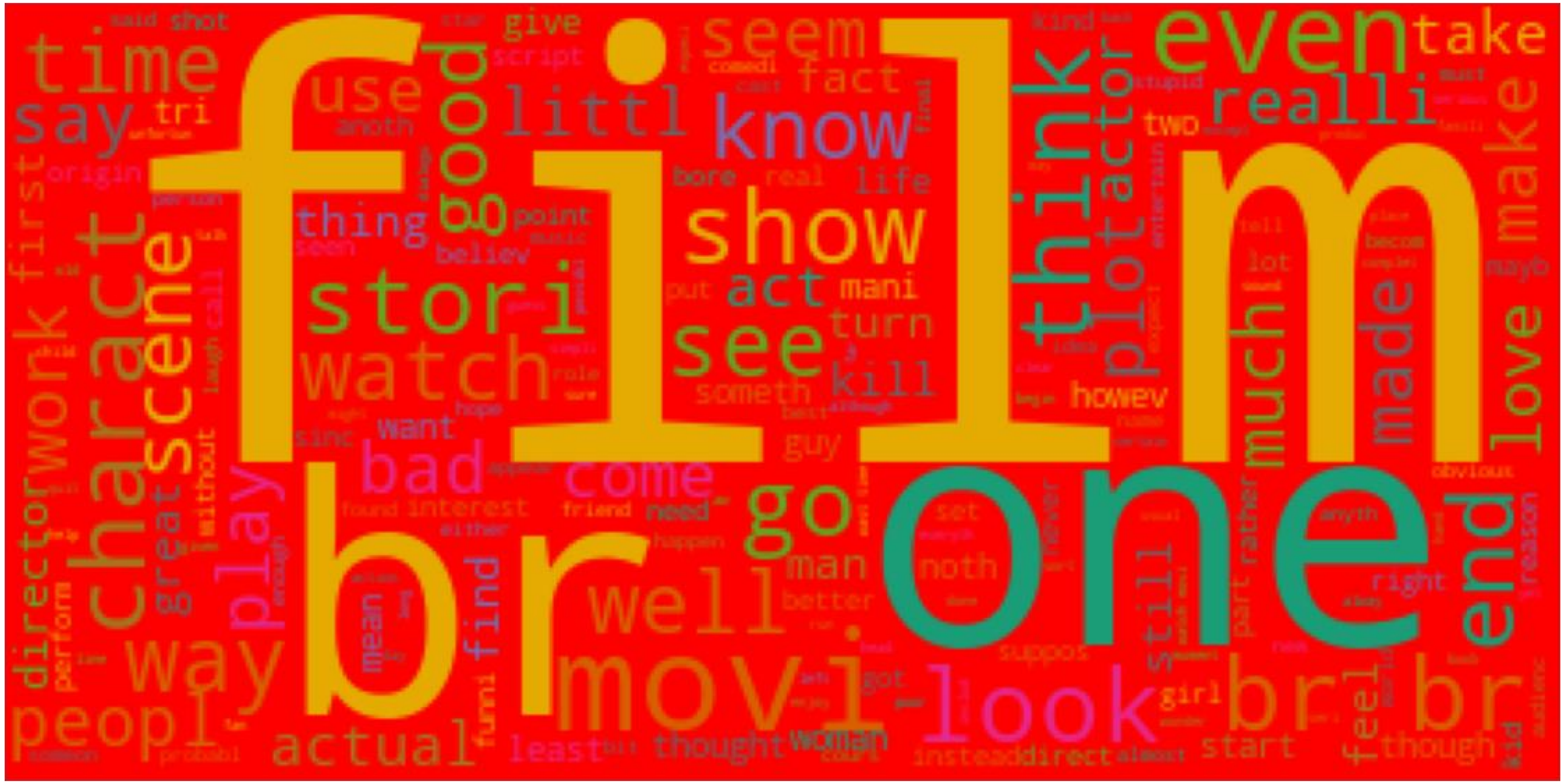
Word cloud is a technique for visualising frequent words in a text where the size of the words represents their frequency.

Word Cloud – Positive words post feature selection



Word cloud is a technique for visualising frequent words in a text where the size of the words represents their frequency.

Word Cloud – Negative words post feature selection



Word cloud is a technique for visualising frequent words in a text where the size of the words represents their frequency.

Models Training & Evaluation

Model Approach

- 6 ML Classifiers including 2 ensembles and 1 LSTM Model trained & evaluated
- Results stored in a data frame.
- Pipeline approach used for Building ML classifiers

Pipeline is a utility that provides a way to automate a machine learning workflow. It works by allowing several transformers to be chained together.

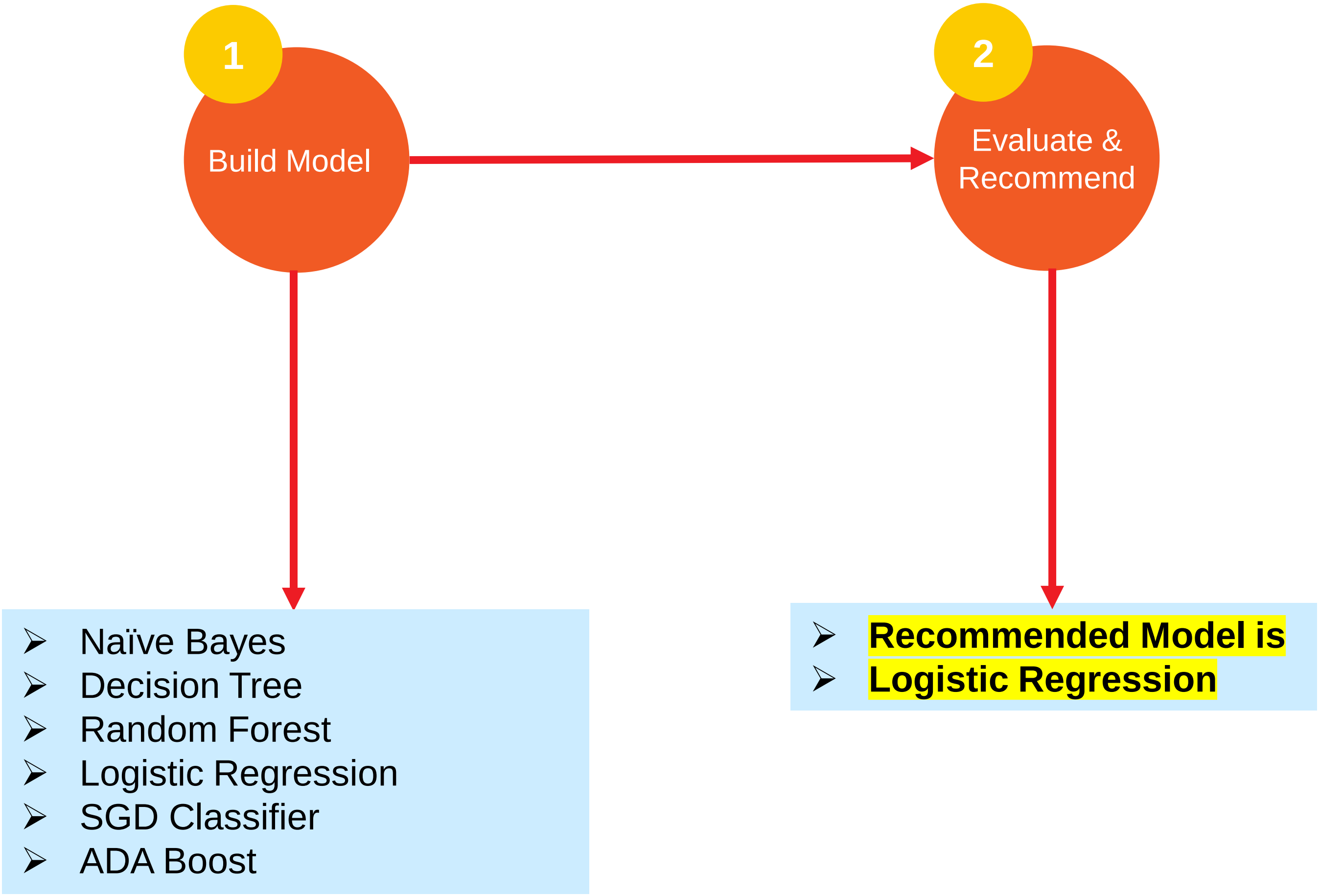
One can also add an estimator at the end of the pipeline. Data flows from the start of the pipeline to its end, and each time it is transformed and fed to the next component.

A Pipeline object has two main methods:

fit_transform: this same method is called for each transformer and each time the result is fed into the next transformer

fit_predict: if your pipeline ends with an estimator, then as before the data is transformed until it arrives at the last step, where it is fed into the estimator and fit_predict is called on the estimator.

Model Training & Evaluation – ML Models



1

Build Model

- Naïve Bayes
- Decision Tree
- Random Forest
- Logistic Regression
- SGD Classifier
- ADA Boost

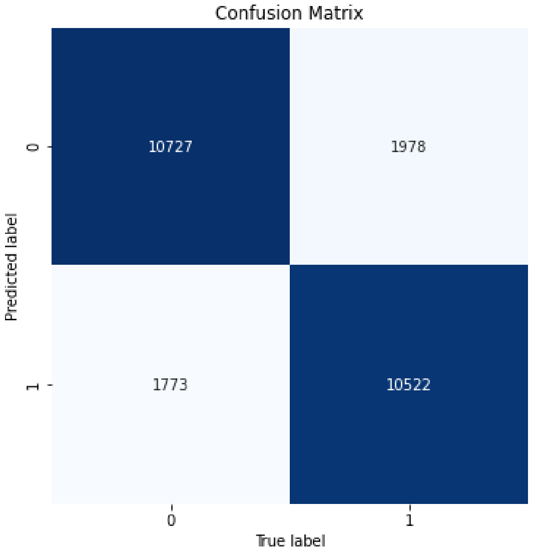
2

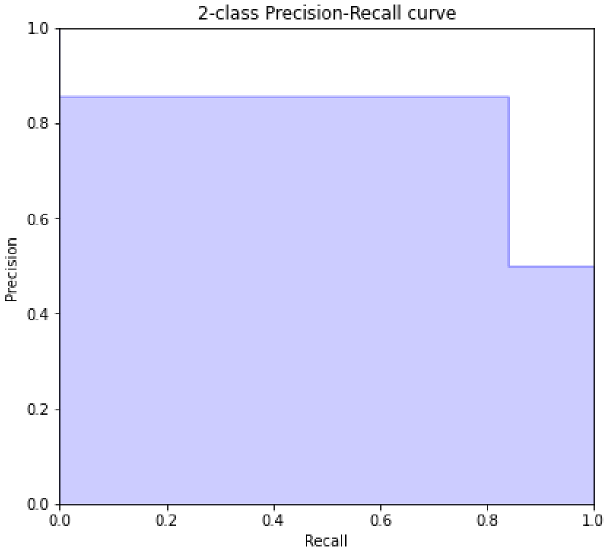
Evaluate & Recommend

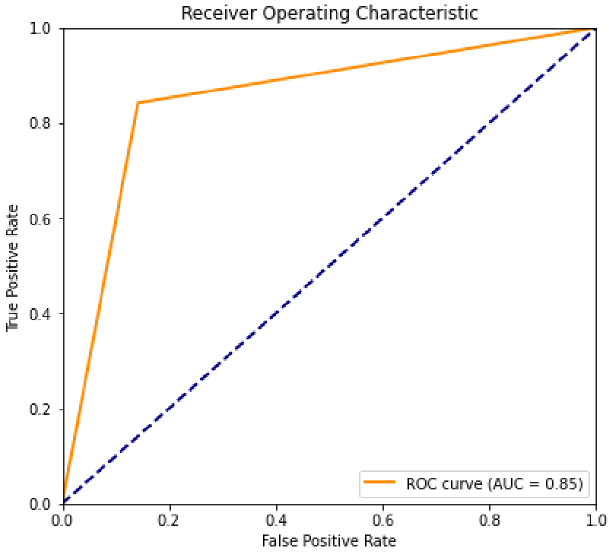
- **Recommended Model is**
- **Logistic Regression**

Naïve Bayes – Details of scores

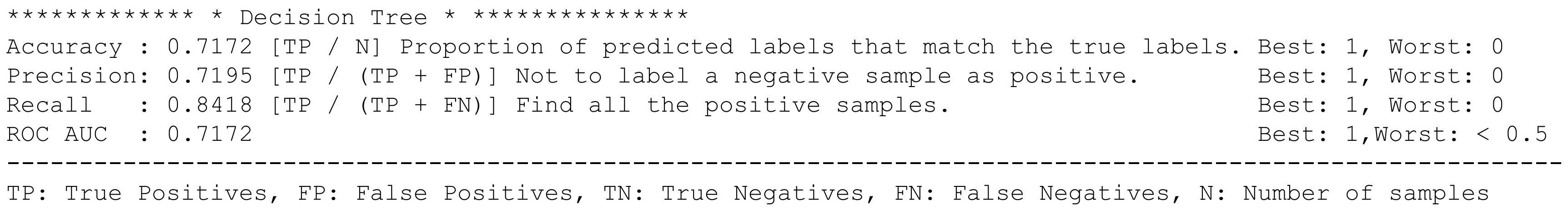
```
************* * Naive Bayes * ***************
Accuracy : 0.8500 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.8558 [TP / (TP + FP)] Not to label a negative sample as positive.       Best: 1, Worst: 0
Recall   : 0.8418 [TP / (TP + FN)] Find all the positive samples.                    Best: 1, Worst: 0
ROC AUC  : 0.8500                                                                    Best: 1,Worst: < 0.5
----------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```



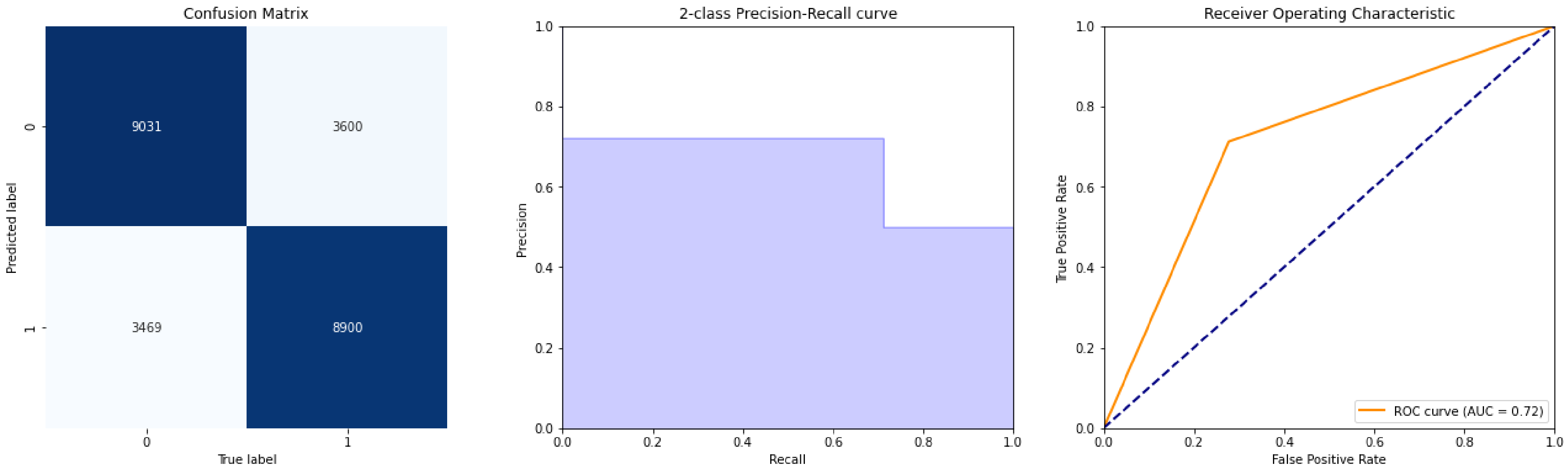




Decision Tree – Details of scores

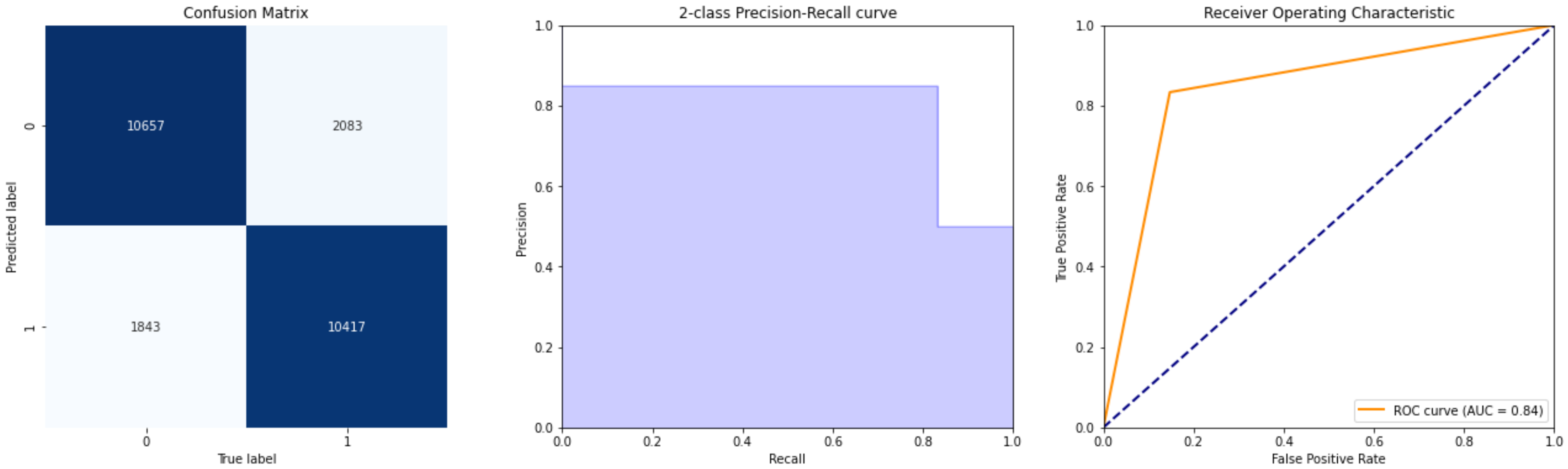


```
************* * Decision Tree * ***************
Accuracy : 0.7172 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.7195 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall    : 0.8418 [TP / (TP + FN)] Find all the positive samples.                    Best: 1, Worst: 0
ROC AUC   : 0.7172                                                                    Best: 1,Worst: < 0.5
-----------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```

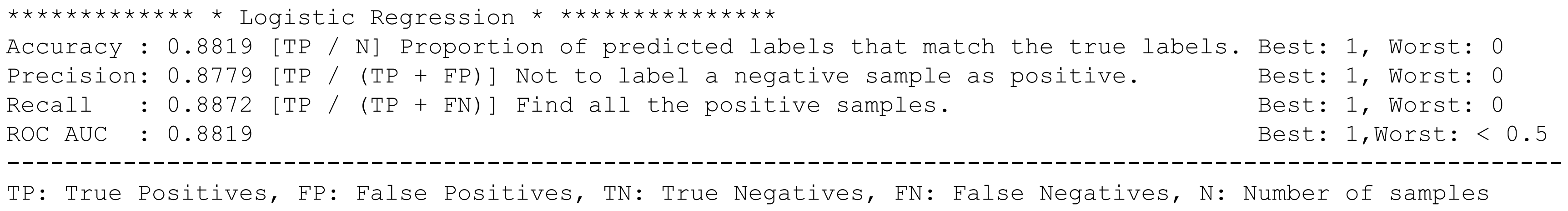


Random Forest – Details of scores

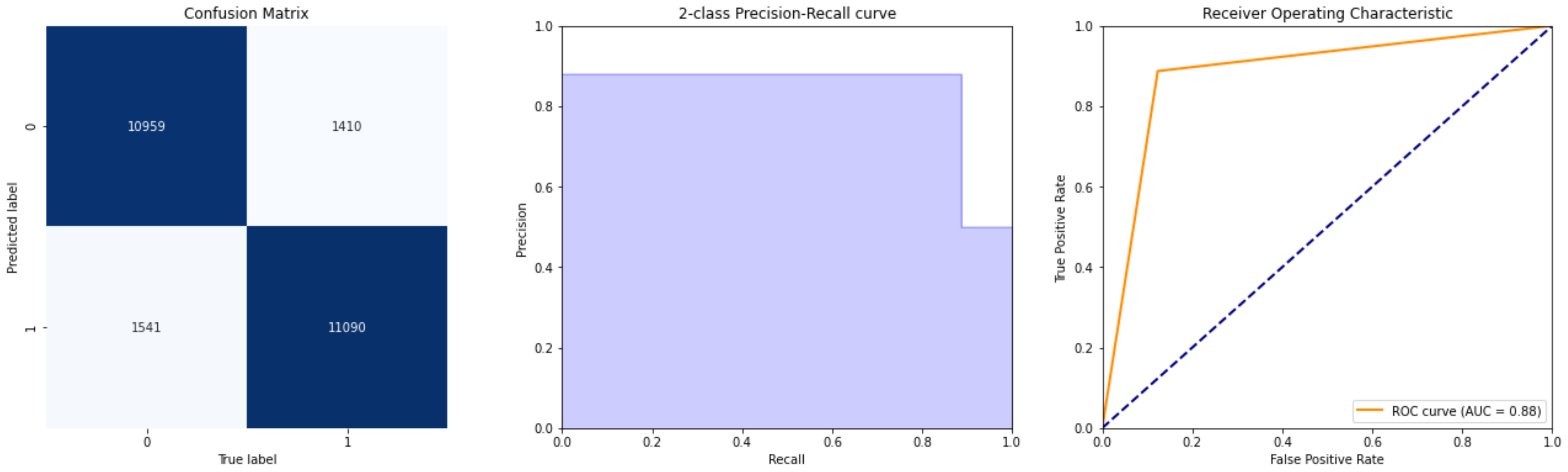
```
************* * Decision Tree * ***************
Accuracy : 0.8429 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.8496 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.8333 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.8429                                                                      Best: 1,Worst: < 0.5
-----------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```



Logistic Regression – Details of scores

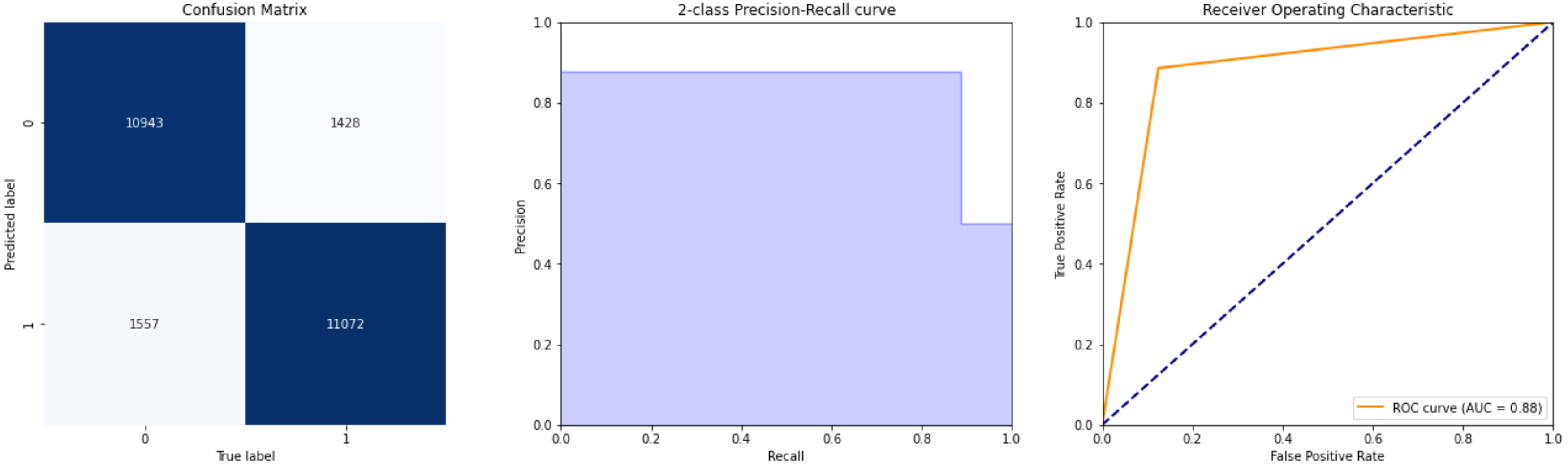


```
************* * Logistic Regression * ***************
Accuracy : 0.8819 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.8779 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.8872 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.8819                                                                     Best: 1,Worst: < 0.5
-----------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```



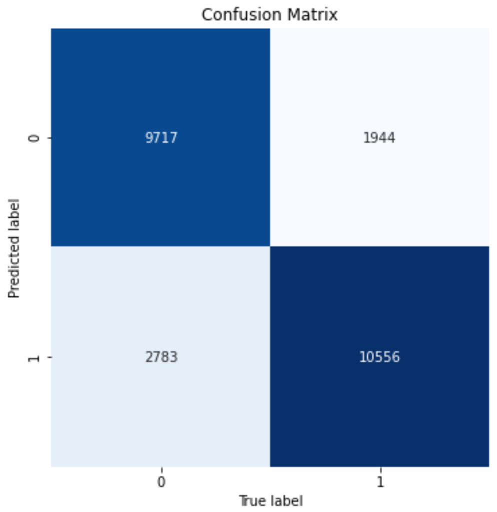
SGD Classifier – Details of scores

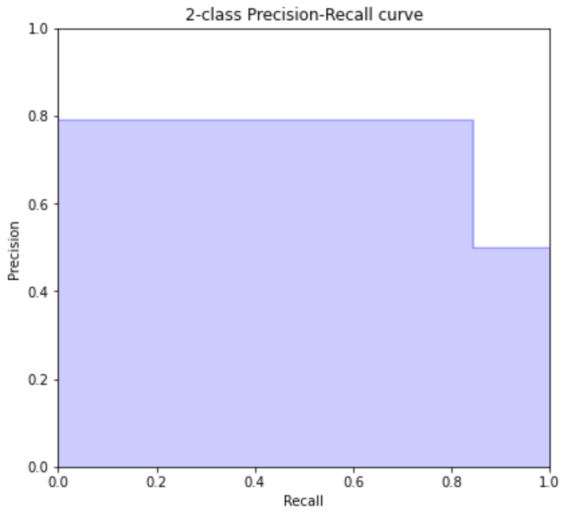
```
************* * SGD Classifier * ***************
Accuracy : 0.8806 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.8767 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.8857 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.8806                                                                      Best: 1,Worst: < 0.5
-----------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```

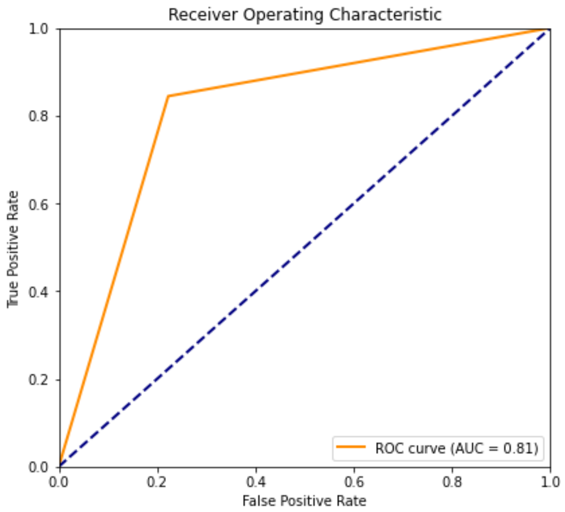


ADA Boost – Details of scores

```
************* * ADA Boost * ***************
Accuracy : 0.8109 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.7914 [TP / (TP + FP)] Not to label a negative sample as positive.        Best: 1, Worst: 0
Recall   : 0.8445 [TP / (TP + FN)] Find all the positive samples.                     Best: 1, Worst: 0
ROC AUC  : 0.8109                                                                     Best: 1,Worst: < 0.5
-------------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```

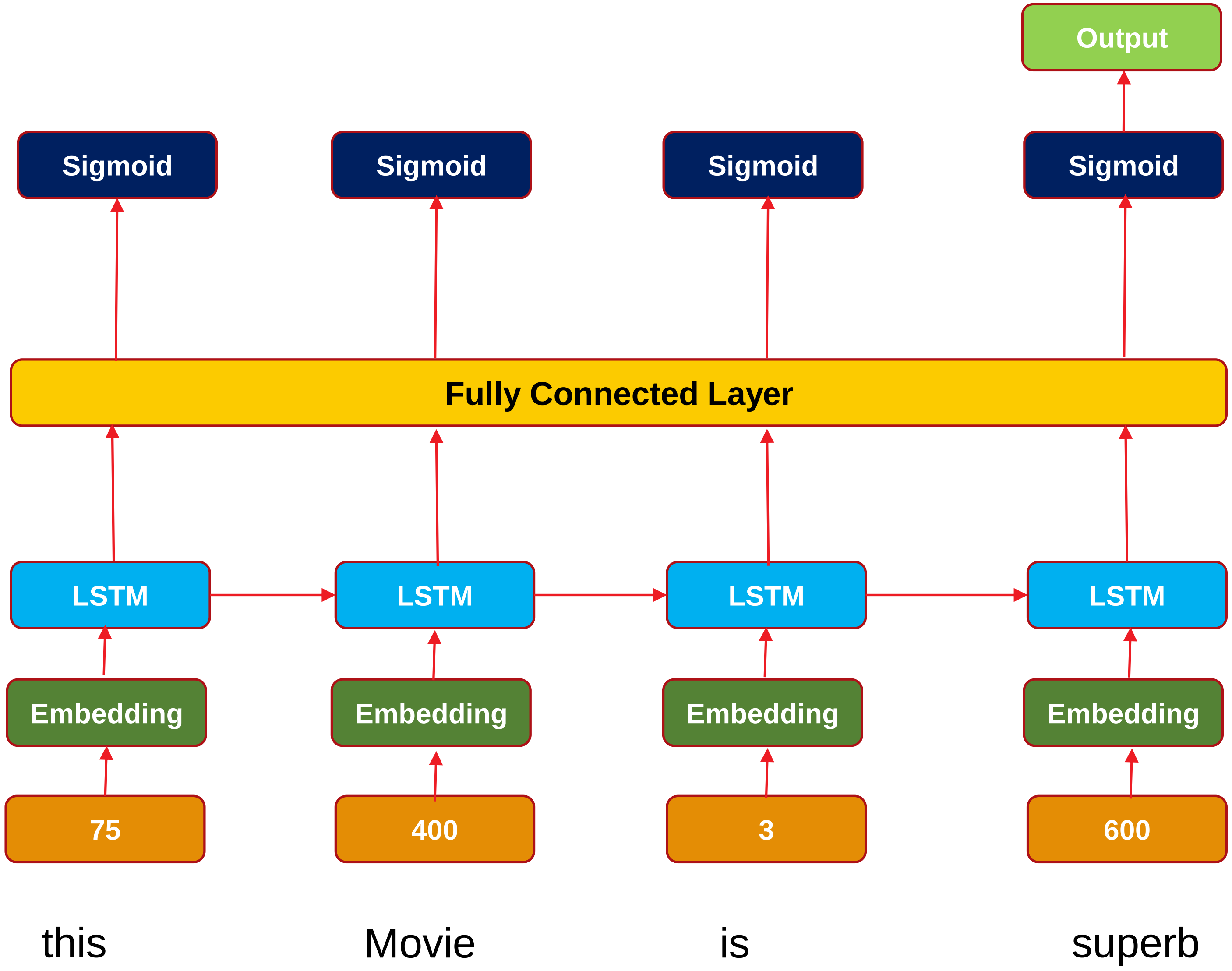






RNN Model Training & Evaluation

LSTM Architecture for Sentiment Analysis



Output

Sigmoid Sigmoid Sigmoid Sigmoid

Fully Connected Layer

LSTM LSTM LSTM LSTM

Embedding Embedding Embedding Embedding

75 400 3 600

this Movie is superb

Probability Score

Last Timestep

Tokenize

LSTM Model Architecture - Current Model

```
EMBED_DIM = 32
model = Sequential()
model.add(Embedding(total_words, EMBED_DIM, input_length=max_length))
model.add((LSTM(32, return_sequences = True)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(250, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
['accuracy'])
print(model.summary())
```

LSTM Model - Total Params

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 106, 32)           96032
_________________________________________________________________
lstm (LSTM)                  (None, 106, 32)           8320
_________________________________________________________________
dropout_1 (Dropout)          (None, 106, 32)           0
_________________________________________________________________
flatten_1 (Flatten)          (None, 3392)              0
_________________________________________________________________
dense_2 (Dense)              (None, 250)               848250
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 251
=================================================================
Total params: 952,853
Trainable params: 952,853
Non-trainable params: 0
```

Accuracy is 82%

Model Recommendation

ML Model Evaluation Basis

- **Accuracy Score** – Model Accuracy
- **Precision** – represents the model’s ability to correctly predict the positives out of all the positive prediction it made. **[TP / (TP + FP)]**
- **Recall** – quantifies the number of correct positive predictions made from all positive predictions that could have been made. **[TP / (TP + FN)]**
- **ROC_AUC** – AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

	Model	Accuracy	Precision	Recall	ROC_AUC
0	Naive Bayes	0.84996	0.855795	0.84176	0.84996
1	Decision Tree	0.71724	0.719541	0.71200	0.71724
2	Random Forest	0.84296	0.849674	0.83336	0.84296
3	Logistic Regression	0.88196	0.877999	0.88720	0.88196
4	SGD Classifier	0.88060	0.876712	0.88576	0.88060
5	ADA Boost	0.81092	0.79136	0.84448	0.81092

- *Based on all the scoring parameters, **Logistic Regression** emerges as the recommended model from ML category*

Next Steps

- Given the massive focus and investment in NLP, need to further tune models to achieve better accuracy
- Need to fine tune the ensemble models with different base estimators & hyper parameters.
- Work on tuning of LSTM Models to achieve higher accuracies.

References

- Natural Language Processing (NLP) Simplified : A Step-by-step Guide (datascience.foundation)
- Sentiment Analysis — A how-to guide with movie reviews | by Shiao-li Green | Towards Data Science
- https://towardsdatascience.com/sentiment-analysis-a-how-to-guide-with-movie-reviews-9ae335e6bcb2

Thank You