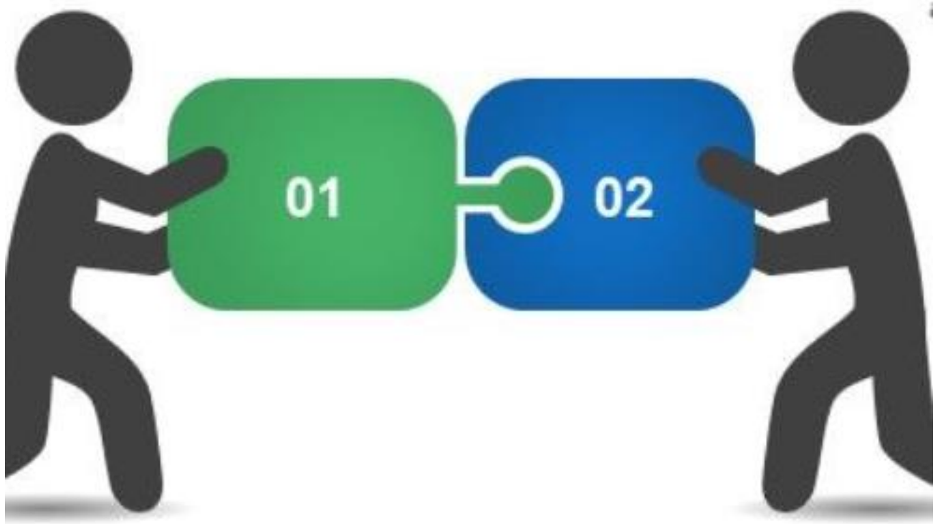
Matching Question pairs with similar intent : An NLP based approach

GROUP 03

1. Priyala Verma (261261)
2. Ankesh Anupam (261387)
University of Stavanger (UiS)

Universitetet i Stavanger

Contribution



Group 03

Team Members

- Ankesh Anupam (261387)
  - Feature Engineering
  - Word Embedding: GloVe and Spacy
  - Machine Learning – Logistic Regression, SVM
  - Deep Learning: CNN model
- Priyala Verma(261261)
  - Exploratory Data Analysis and Text Preprocessing
  - Machine Learning – RF , XGBoost
  - Deep Learning: Bi LSTM

Outline

1. **Problem Definition**
2. **Exploratory Data Analysis**
3. **Feature Engineering**
4. **Classical Machine Learning**
5. **Deep Learning Approach**
6. **Conclusion**

Problem Definition

Identify question pairs which have the same intent or are duplicates
These question pairs have the same answer

Quora released its first ever dataset publicly on 24th Jan, 2017 This dataset consists of question pairs which are either duplicate or not -based on the intent of the question

The data consisted of 404K question pairs with 255K negative samples (non-duplicates) and 149K positive samples (duplicates)

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

Examples

Duplicates

How come Trump win the Presidency?

How did Donald Trump win the 2016 Presidential Election?

What practical applications might evolve from the discovery of the Higgs Boson?

What are some practical benefits of discovery of the Higgs Boson?

Non-Duplicates

Who should I address my cover letter to if I'm applying for a big company like Apple?

Which car is better from safety view – Toyota or Volvo?

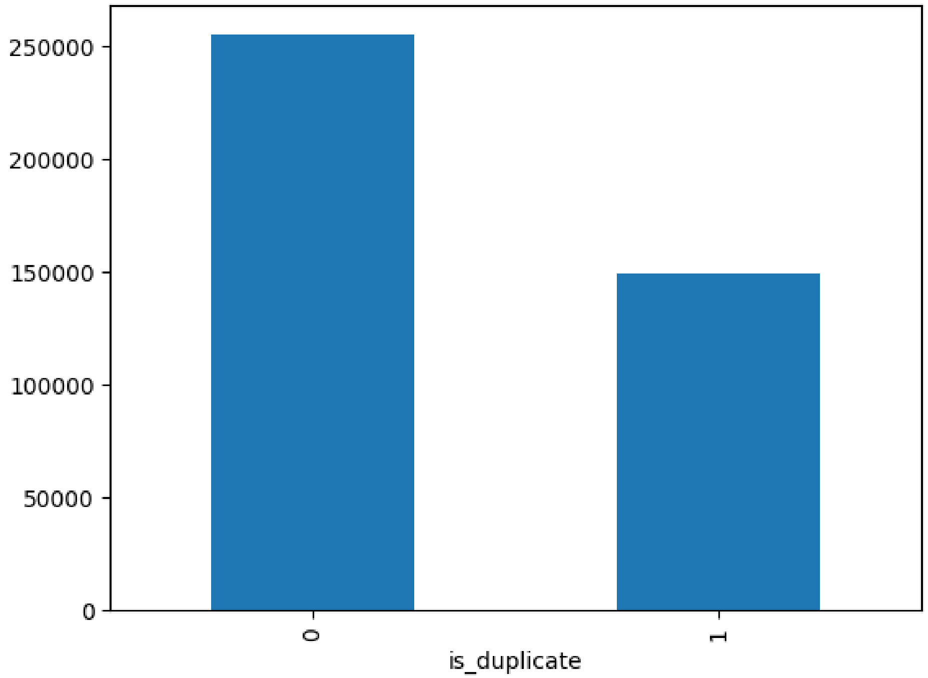
How can I start an online shopping (e-commerce) website?

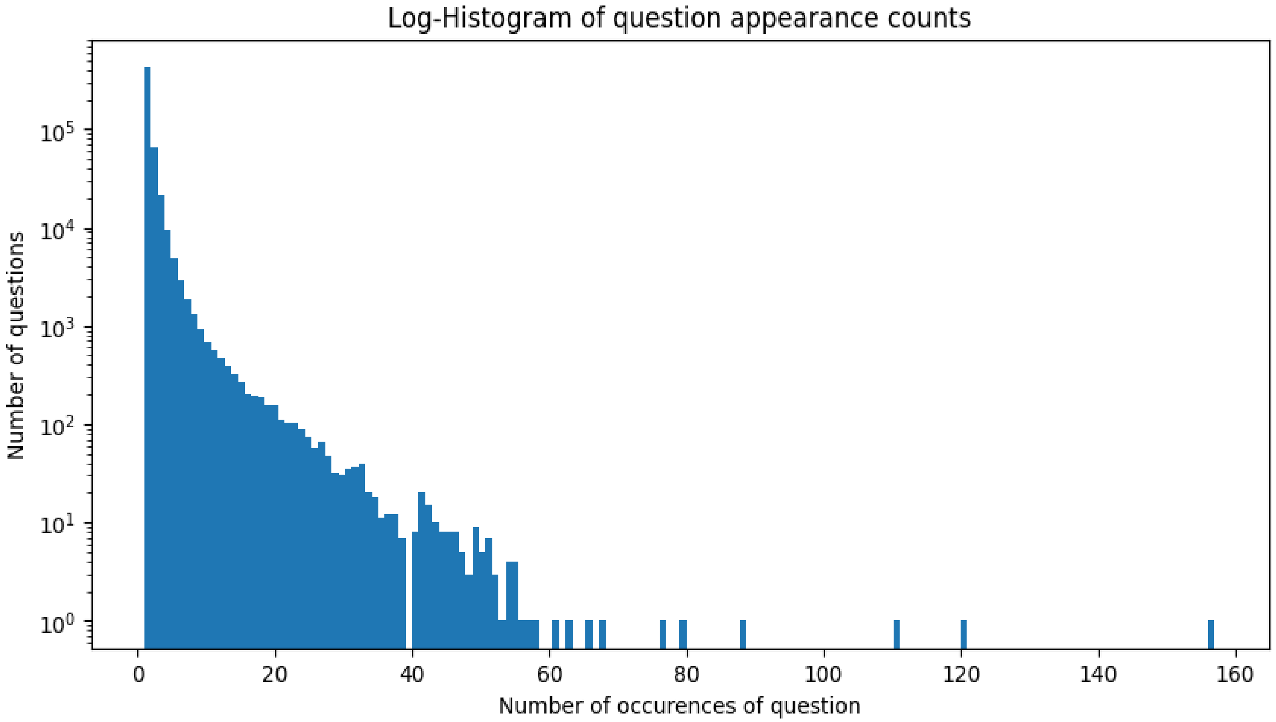
Which web technology is best suitable for building a big E-Commerce website?

Data Exploration

Remove duplicates and NaN values

(1) Total number of question pairs for training:- 404290
(2) Question pairs are not similar (is_duplicate= 0) in percentage:- 63.08%
(3) Question pairs are similar (is_duplicate= 1) in percentage:- 36.92%





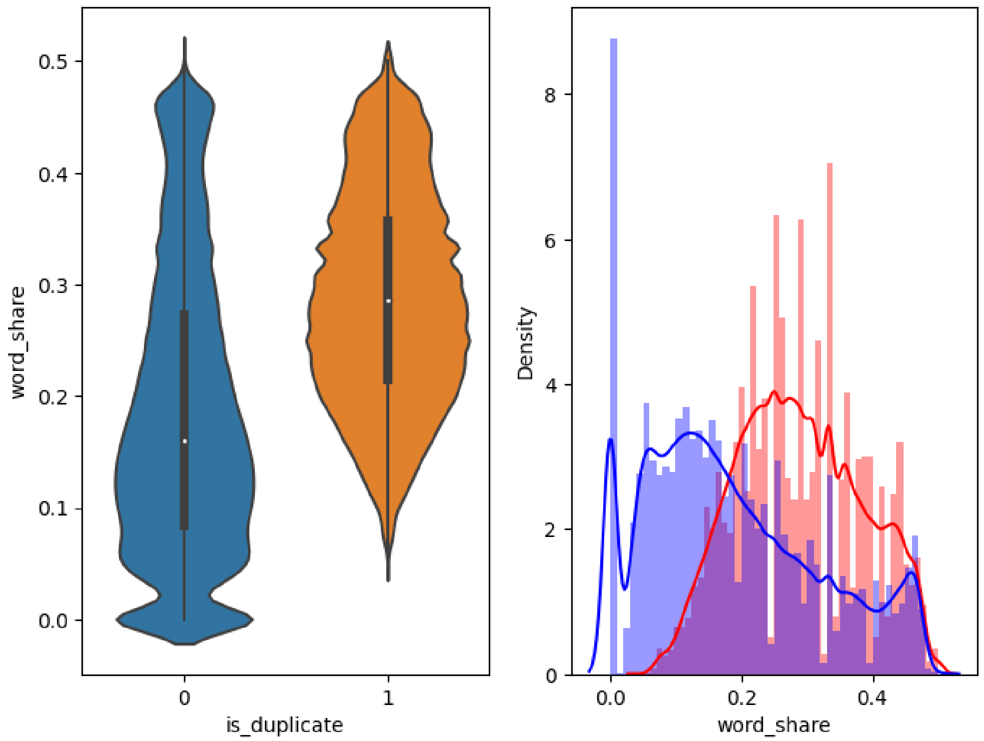
(1) Total number of Unique Questions are:- 537933
(2) Number of unique questions that appear more than one time:- 111780 (20.7%)
(3) Max number of times a single question is repeated:- 157

Feature Engineering

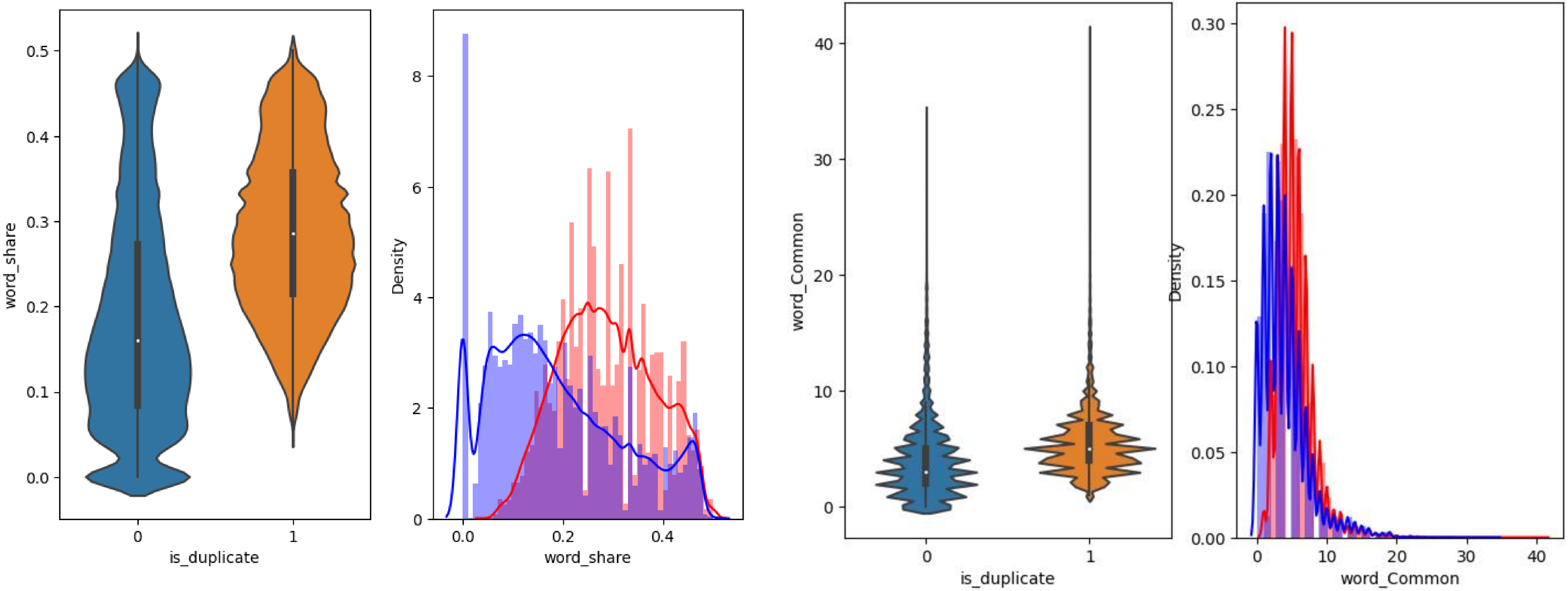
Extract useful features which can be used as input to machine learning algorithms

Basic Features

1. freq_qid1 = Frequency of Question1
2. freq_qid2 = Frequency of Question2
3. q1len = Length of Question1
4. q2len = Length of Question2
5. q1_n_words = Number of words in Question 1
6. q2_n_words = Number of words in Question 2
7. word_Common = Number of common unique words in Q1 and Q2
8. word_Total = Total num of words in Q1 + Total num of words in Q2
9. word_share = (word_common)/(word_Total)
10. freq_q1+freq_q2 = Sum total of frequency of Q1 and Q2
11. freq_q1-freq_q2 = Absolute difference of frequency of Q1 and Q2



Basic Features



Advanced NLP Features

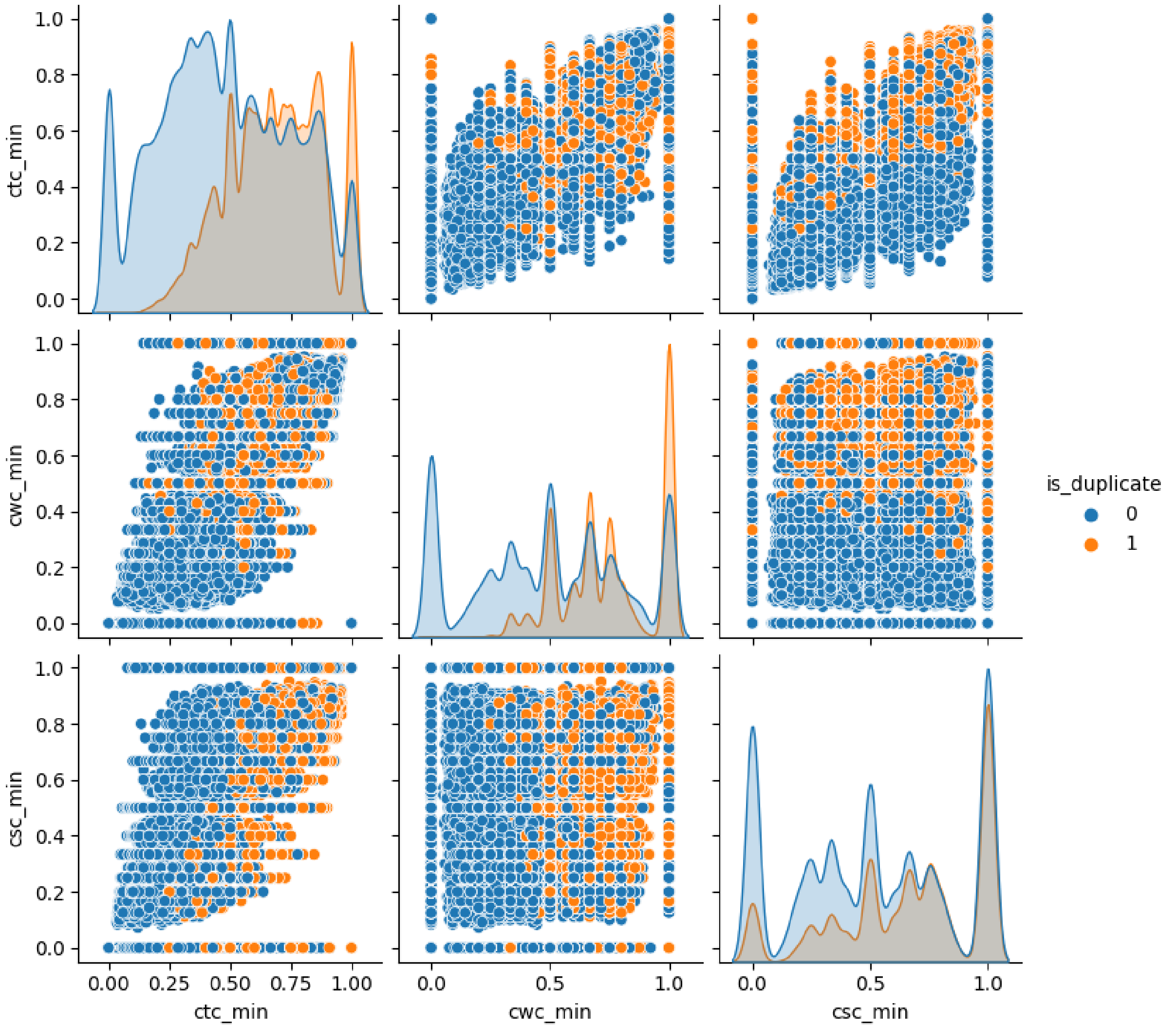
Before extracting advanced features, **Preprocessing** is done:

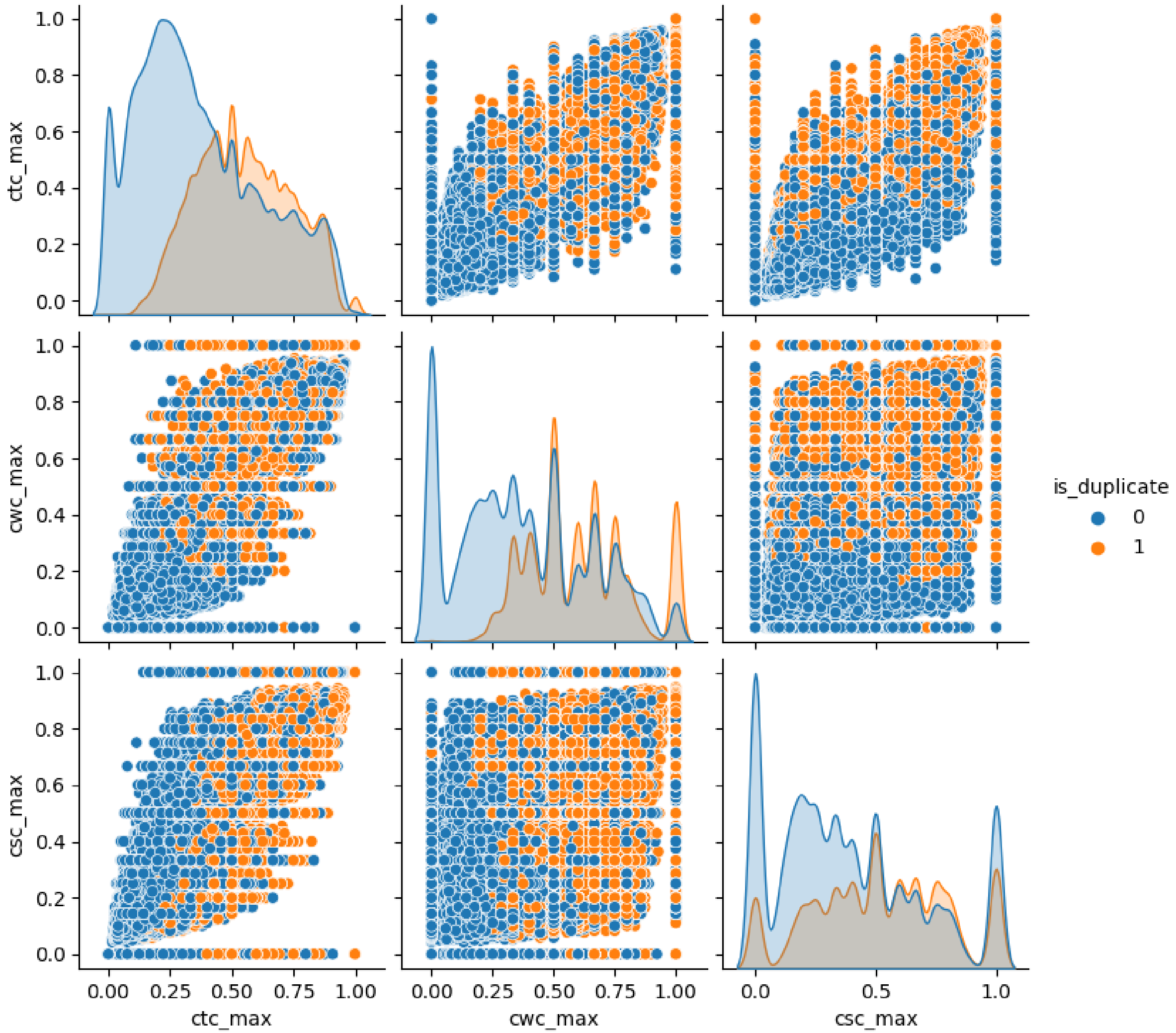
- Removing Html tags
- Removing Punctuations
- Performing stemming
- Removing Stopwords
- Expanding contractions

Advanced Features (10 Features)

- cwc_min : Ratio of common_word_count to min length of word count of Q1 and Q2
- cwc_max : Ratio of common_word_count to max length of word count of Q1 and Q2
- csc_min : Ratio of common_stop_count to min length of stop count of Q1 and Q2
- csc_max : Ratio of common_stop_count to max length of stop count of Q1 and Q2
- ctc_min : Ratio of common_token_count to min length of token count of Q1 and Q2
- ctc_max : Ratio of common_token_count to max length of token count of Q1 and Q2
- last_word_eq : Check if last word of both questions is equal or not
- first_word_eq : Check if first word of both questions is equal or not
- abs_len_diff : Abs length difference
- mean_len : Average Token Length of both Questions

Advanced NLP Features



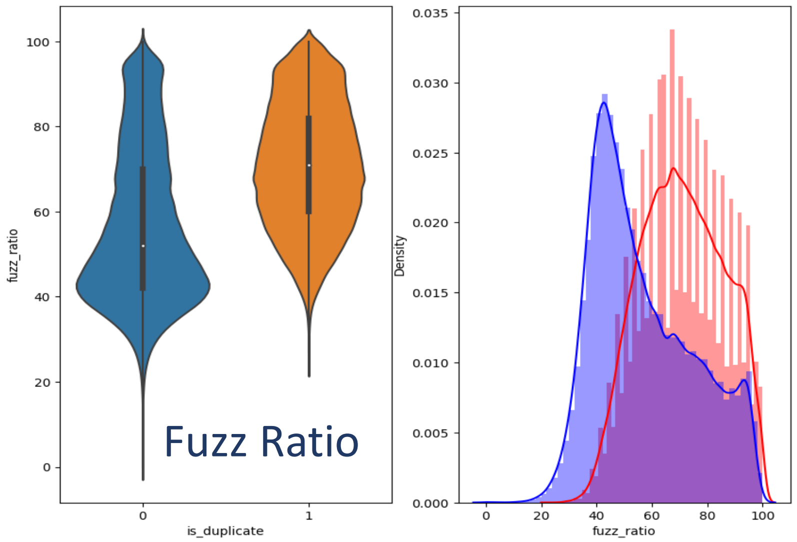


Fuzzy Features

FuzzyWuzzy package in Python allows matching strings using similarity index using pattern matching (Levenshtein Distance)

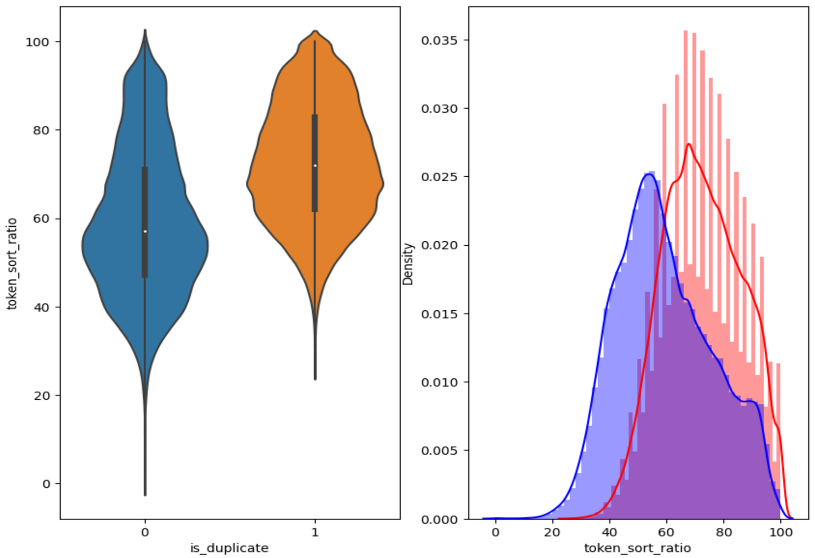
- Fuzz Ratio - Calculates the edit distance based on the ordering of both input strings
- Fuzz Partial Ratio - Calculates the similarity by taking the shortest string
- Token Sort Ratio - Ignore the ordering of the words in the strings but still determine how similar they are
- Token Set Ratio - It takes out common tokens before calculating how similar the strings are This is extremely helpful when the strings are significantly different in length
- Longest Sub string Ratio

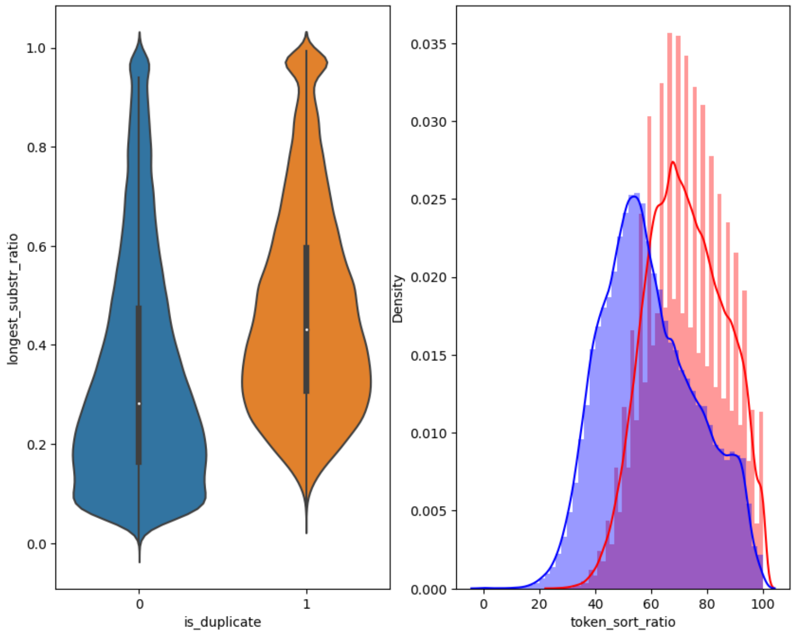
Fuzzy Features



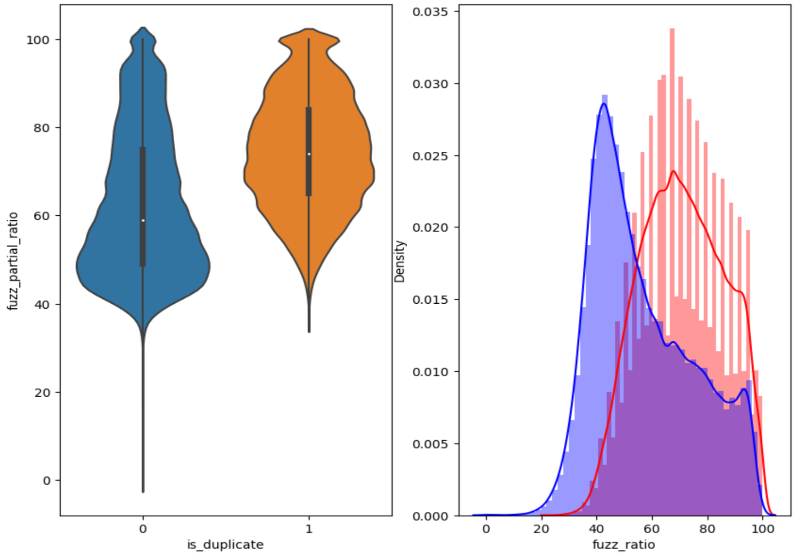
Fuzz Ratio

Token Sort Ratio

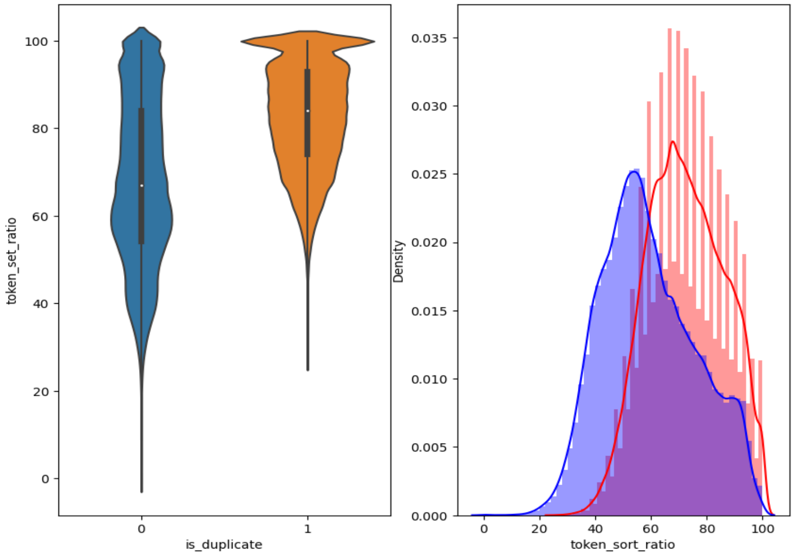




Longest Sub string Ratio



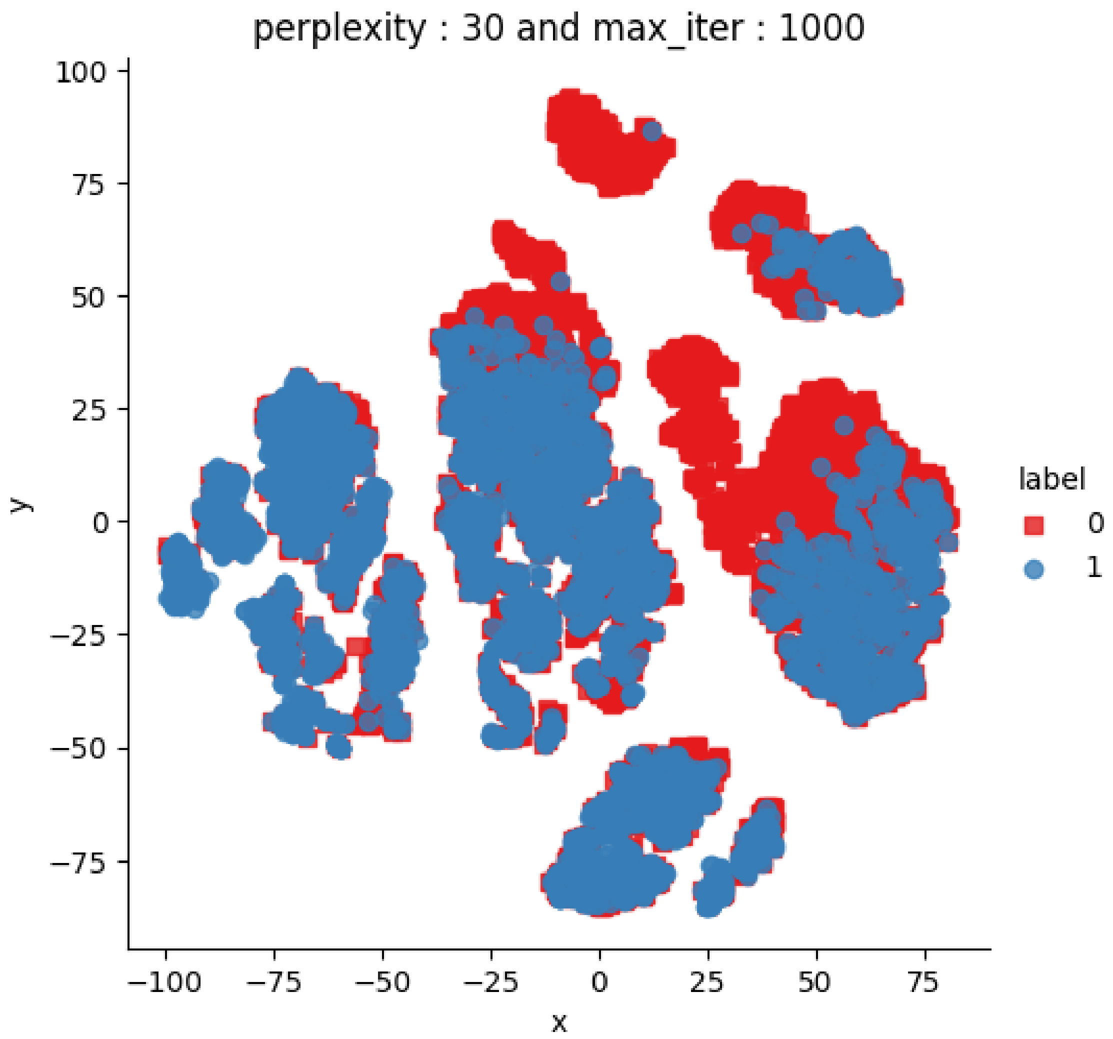
Fuzz Partial Ratio

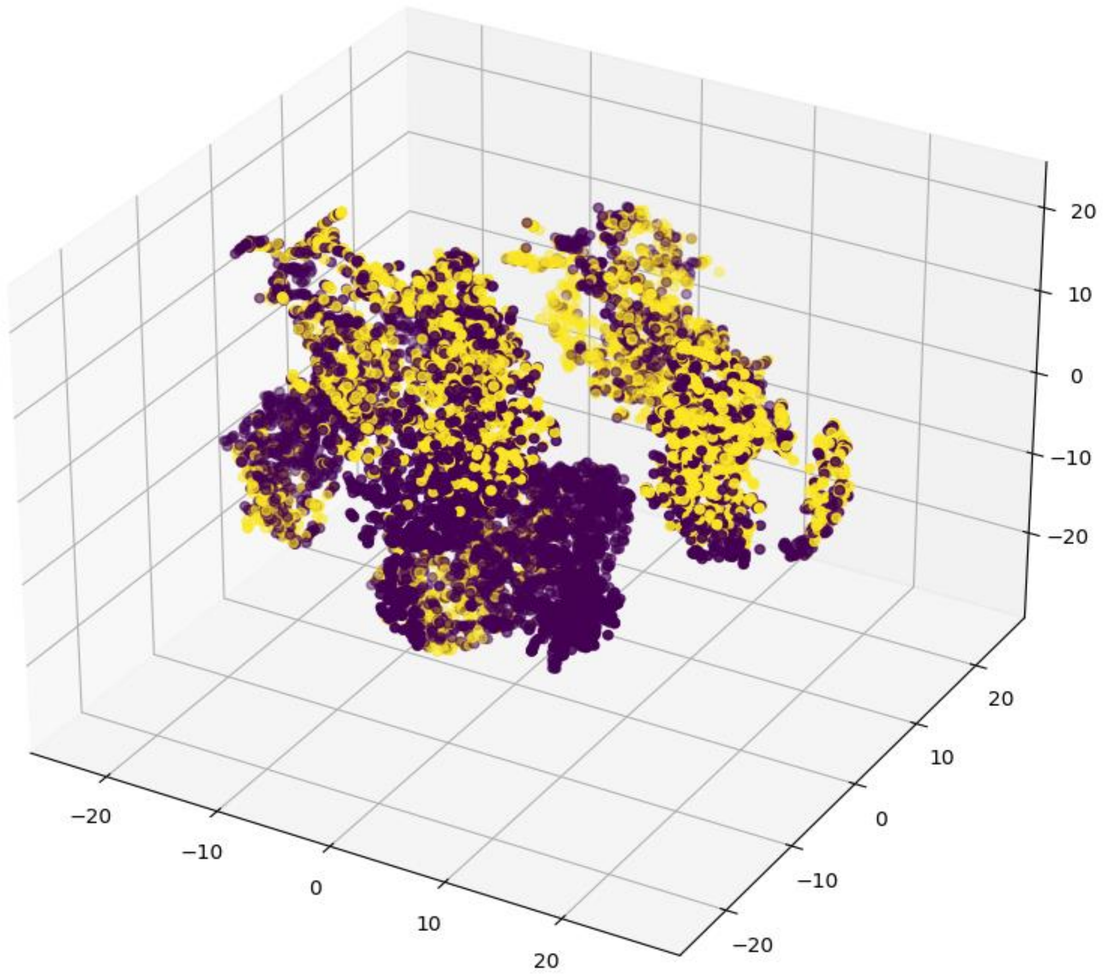


Token Set Ratio

Features Visualization

TSNE Dimensionality Reduction & Visualization on Advanced Feature Extraction (NLP and Fuzzy Features) features





More NLP Features (TF-IDF + Word Embeddings)

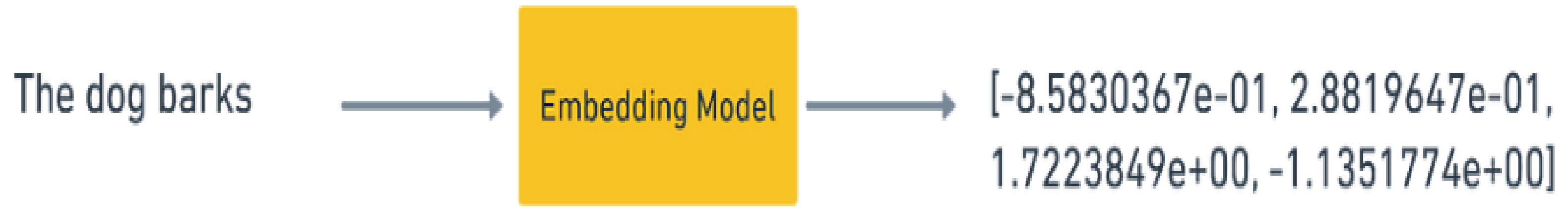
TFIDF : TF-IDF (Term Frequency - Inverse Document Frequency) is an algorithm that uses the frequency of words to determine how relevant those words are to a given document Importance of a term is high when it occurs a lot in a given document and rarely in others

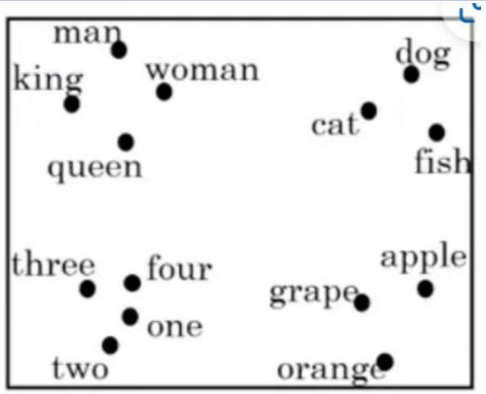
$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}})$$

$$TF\text{-}IDF = TF * IDF$$

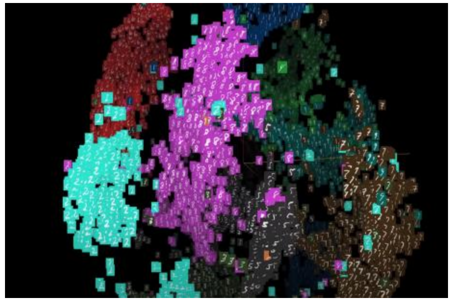
Embeddings transform human-readable text to a multi-dimensional vector in such a way that similar words are spatially near to each other This allows the contextual information from text to be captured in a dense numerical vector





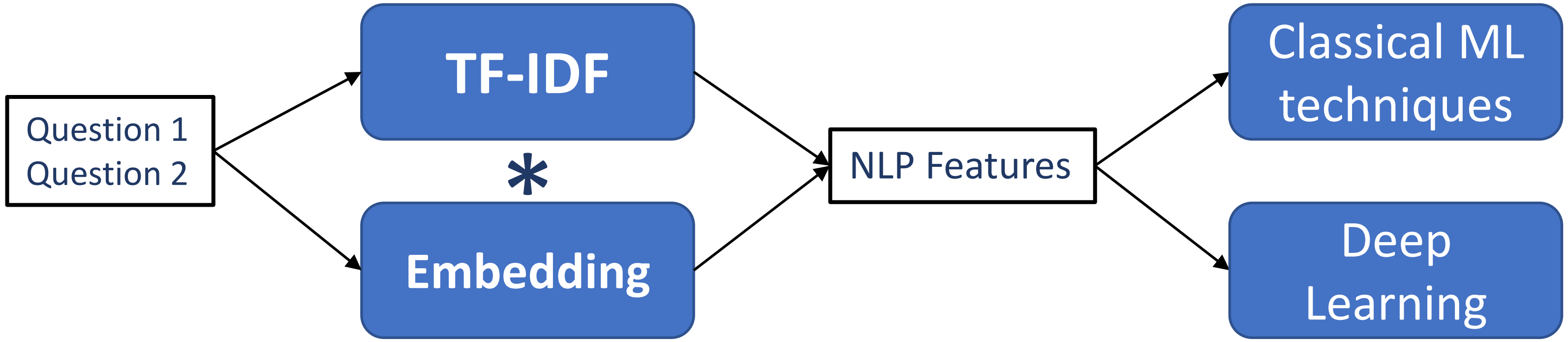
Pre-Calculated embeddings: Word2Vec , GloVe , Spacy etc

More NLP Features – Project Workflow



Project Workflow

- TF-IDF calculated for the text corpus (Question 1 and Question 2)
- Word Embedding generated for each word in a Q1 and Q2
- For each question weighted sum of embedding vector is calculated (Weighted by TFIDF)
- This is normalized by the number of words in the question



Embeddings Used

GloVe (Global Vectors)
- glove_42B_300d
- glove_6B_300d
- glove_6B_100d

SPACY Python Package:
- en_core_web_sm (Small 96 dimensional)
- en_core_web_md (Medium 300 dimn)
- en_core_web_lg (Large 300 dimn)

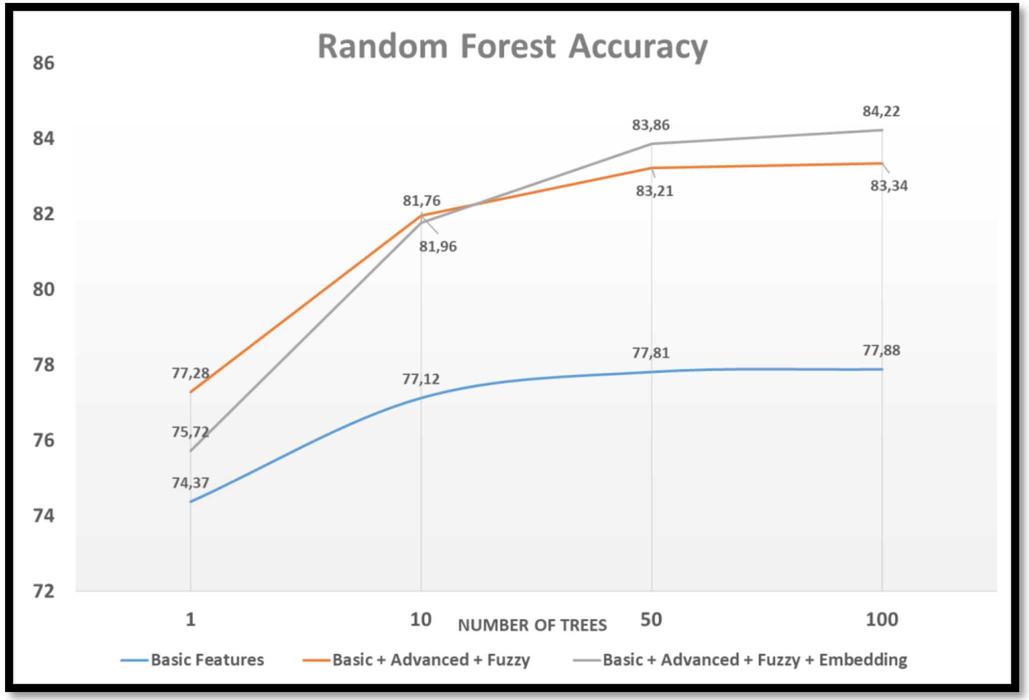
Classical Classification Algorithms

- ❑ Random Forest (Number of Trees: 1, 10 , 50 , 100)
- ❑ Logistic Regression
- ❑ Linear Support Vector Machine
- ❑ XG Boost

Three different version of features used

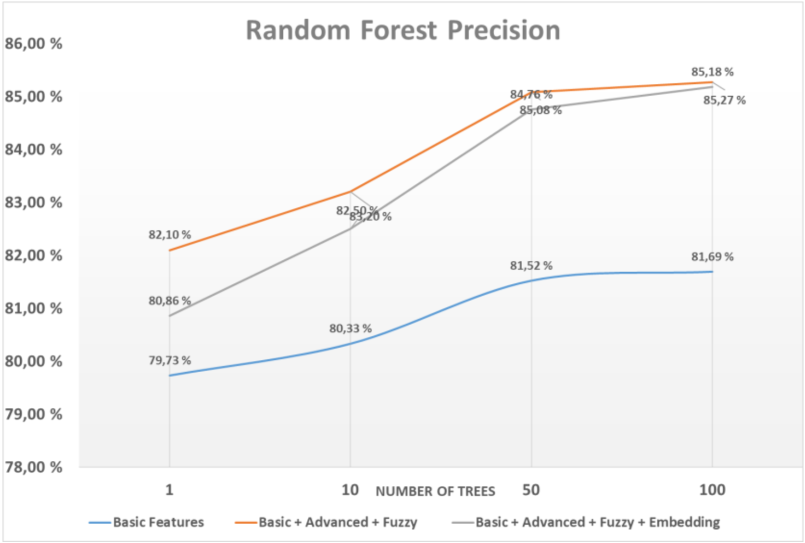
- ❑ Basic Features (11 features)
- ❑ Basic + Advanced + Fuzzy Features (11 + 10 + 5 = 26 features)
- ❑ Basic + Advanced + Fuzzy Features + Embeddings (11 + 10 + 5 + Em(Q1) +Em(Q2))

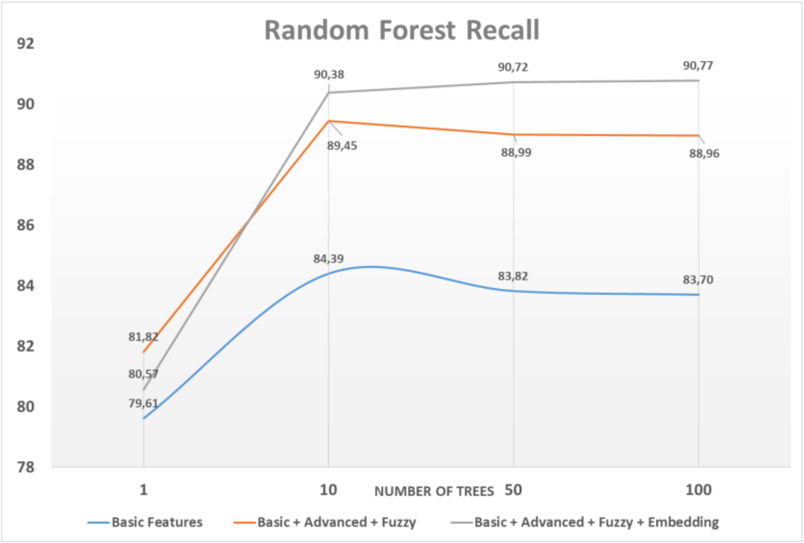
Random Forest

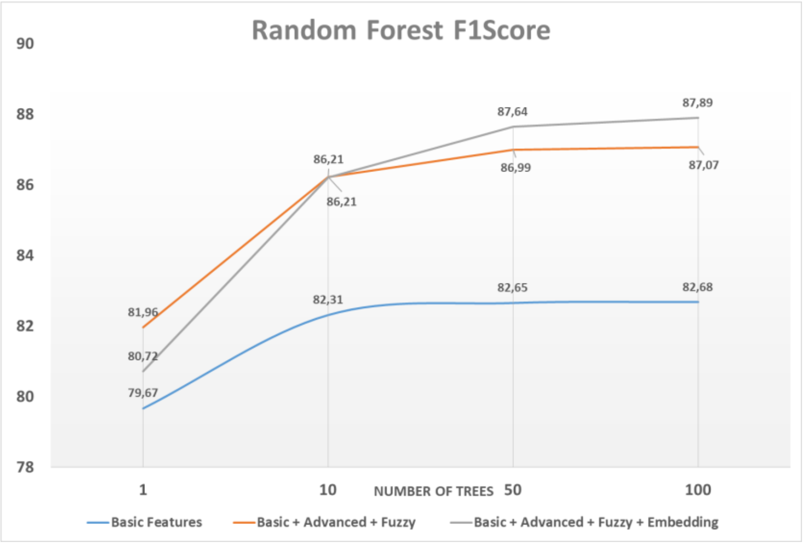


- Accuracy 84.22%

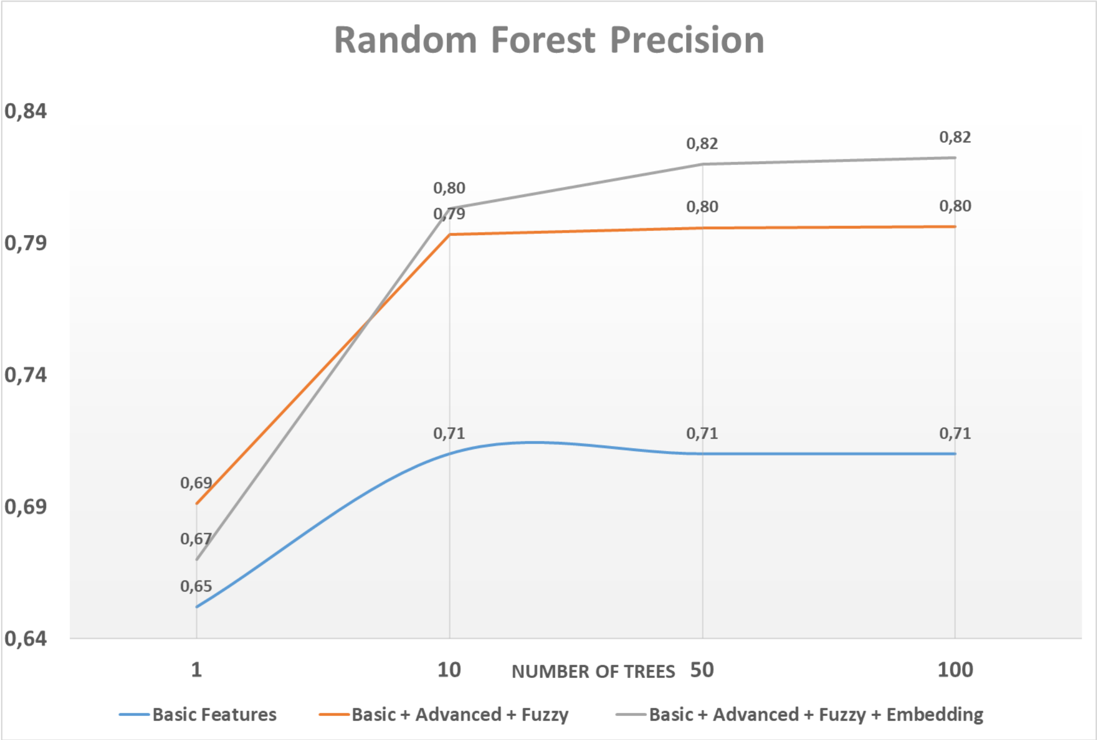
Non -Duplicate

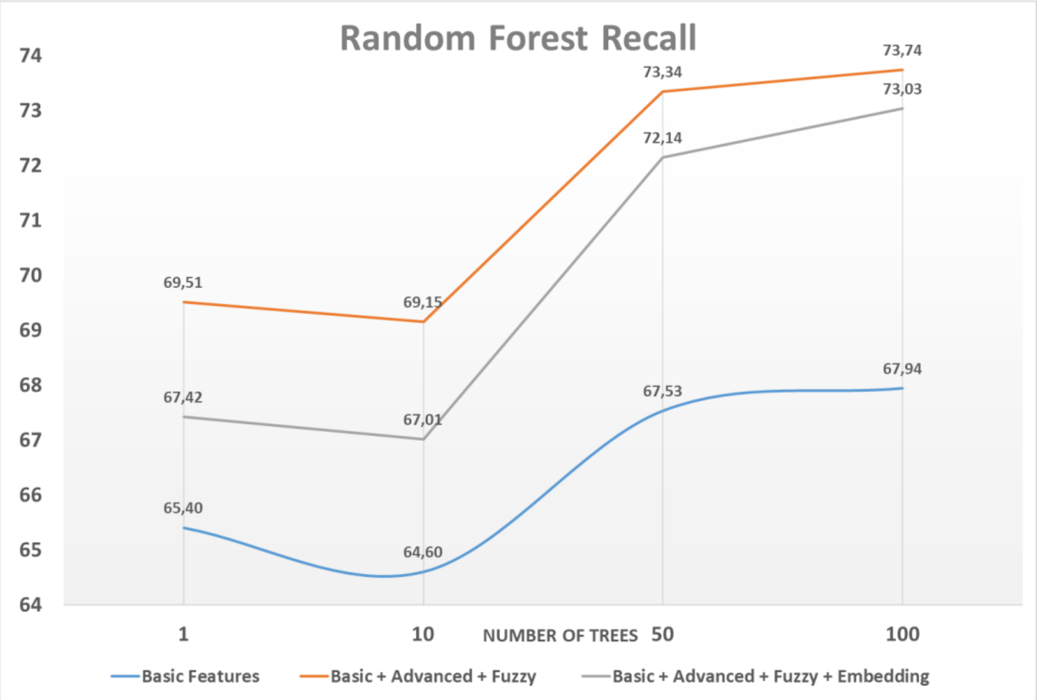


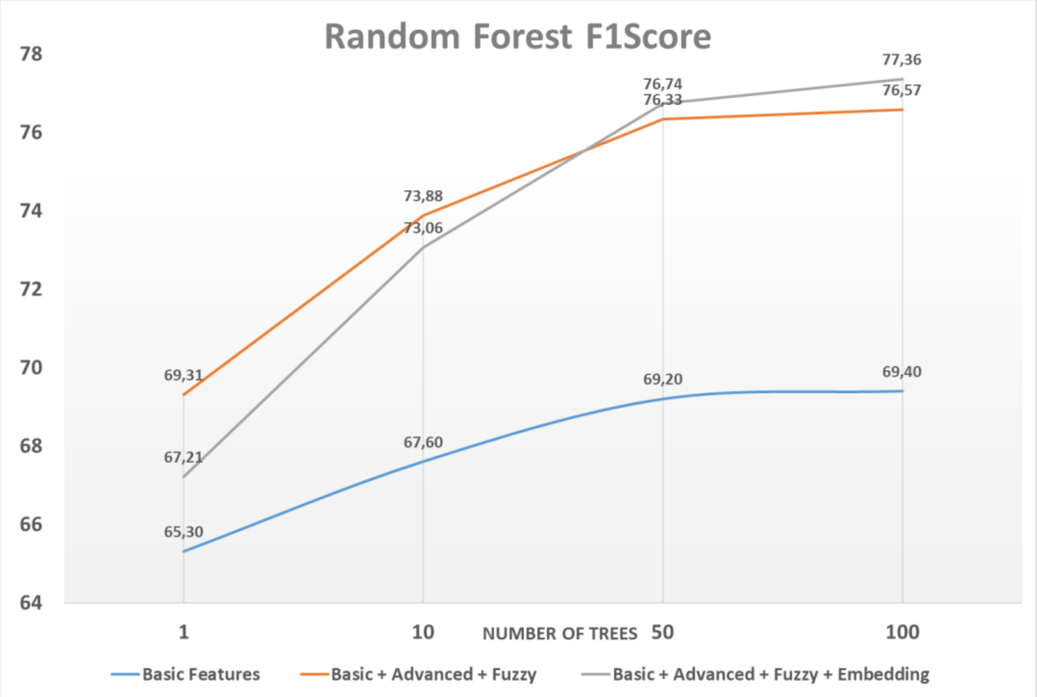




Random Forest : Duplicates

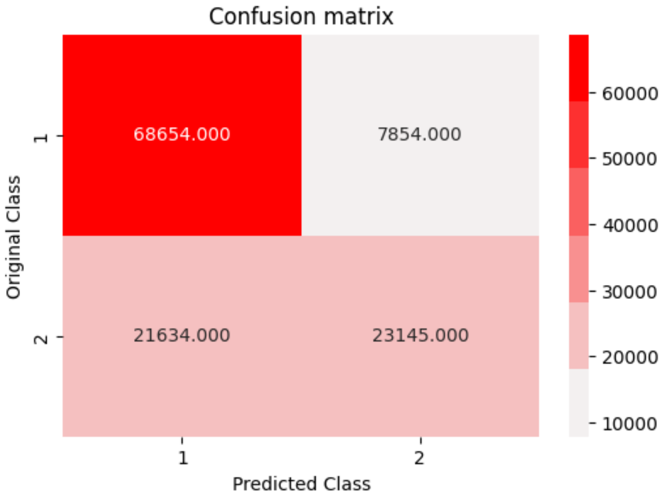






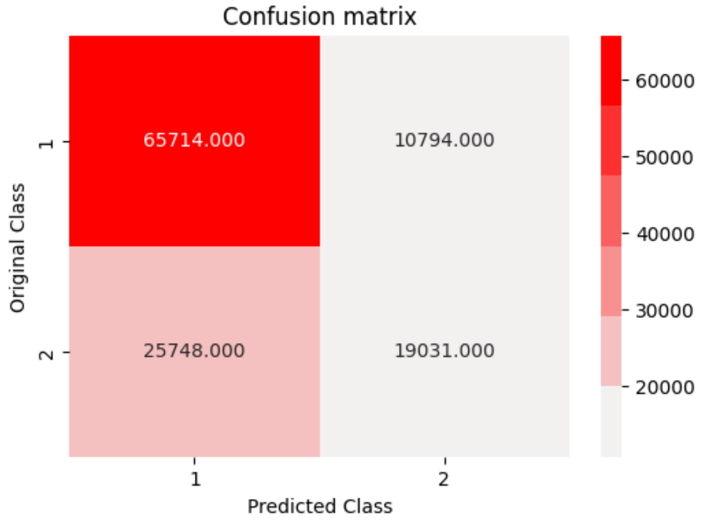
Logistic Regression

Basic Features



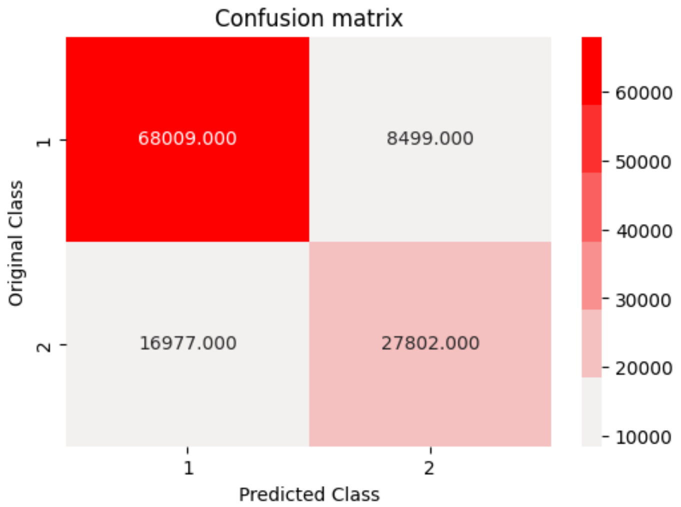
	Duplicate	Non-Duplicate
Accuracy	75,60 %	74,66 %
Precision	76,00 %	74,70 %
Recall	51,70 %	89,70 %
F1 Score	30,77 %	40,76 %

Basic + Advanced + Fuzzy + Embedding Features



	Duplicate	Non-Duplicate
Accuracy	63.9%	61,0 %
Precision	63,8 %	71,8 %
Recall	42,5 %	85,9 %
F1 Score	25,5 %	39,1 %

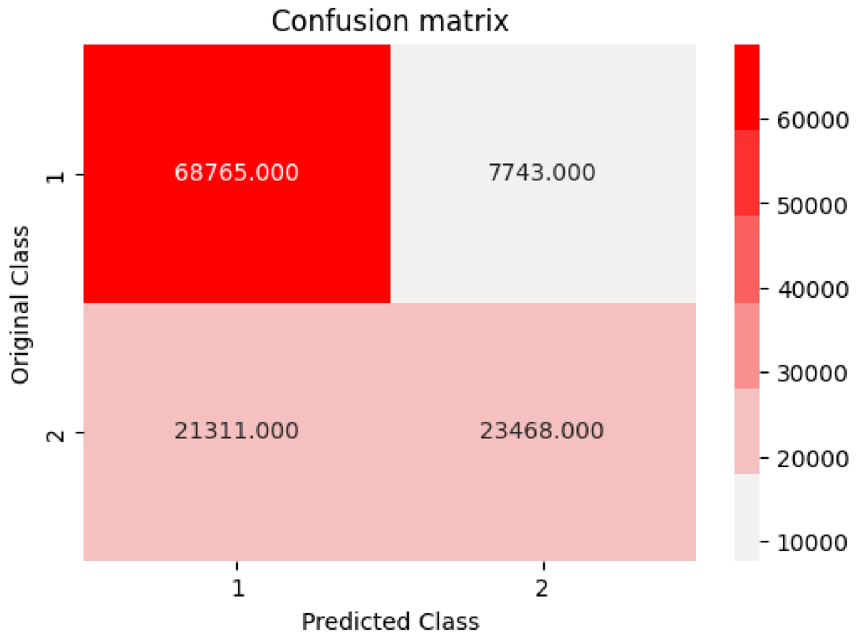
Basic + Advanced + Fuzzy Features



	Duplicate	Non-Duplicate
Accuracy	79,0 %	76,6 %
Precision	76,6 %	80,0 %
Recall	62,1 %	88,9 %
F1 Score	34,3 %	42,1 %

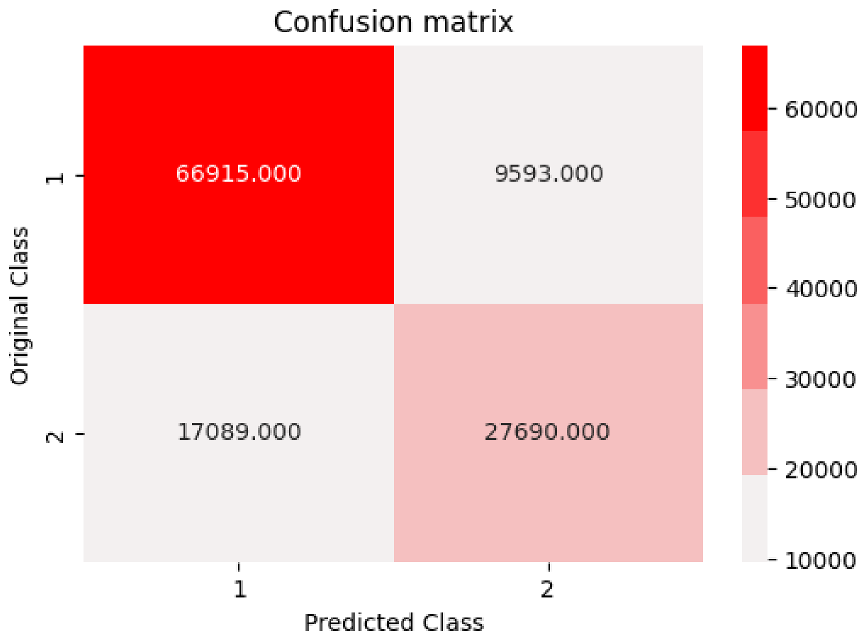
Linear Support Vector Machine

Basic Features



	Duplicate	Non-Duplicate
Accuracy	76.1%	76,04 %
Precision	75,20 %	76,30 %
Recall	52,40 %	89,90 %
F1 Score	30,88 %	41,27 %

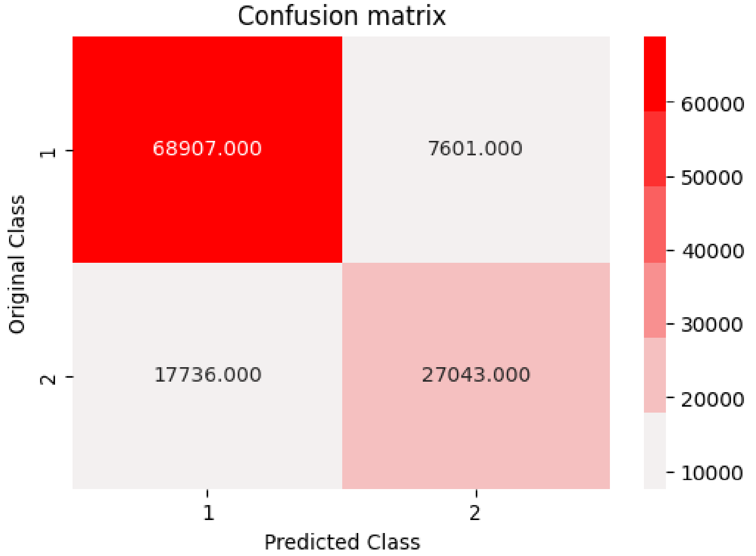
Basic + Advanced + Fuzzy Features



	Duplicate	Non-Duplicate
Accuracy	79.5%	78,0 %
Precision	74,3 %	79,7 %
Recall	61,8 %	87,5 %
F1 Score	33,7 %	41,7 %

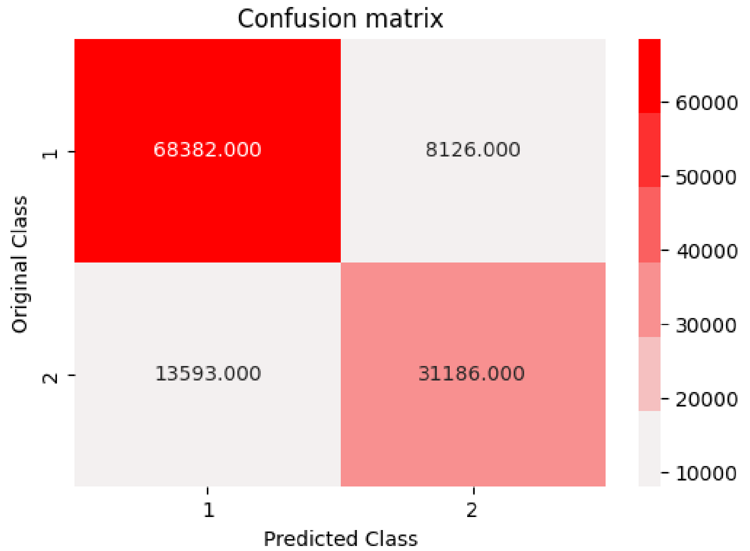
XG Boost

Basic Features



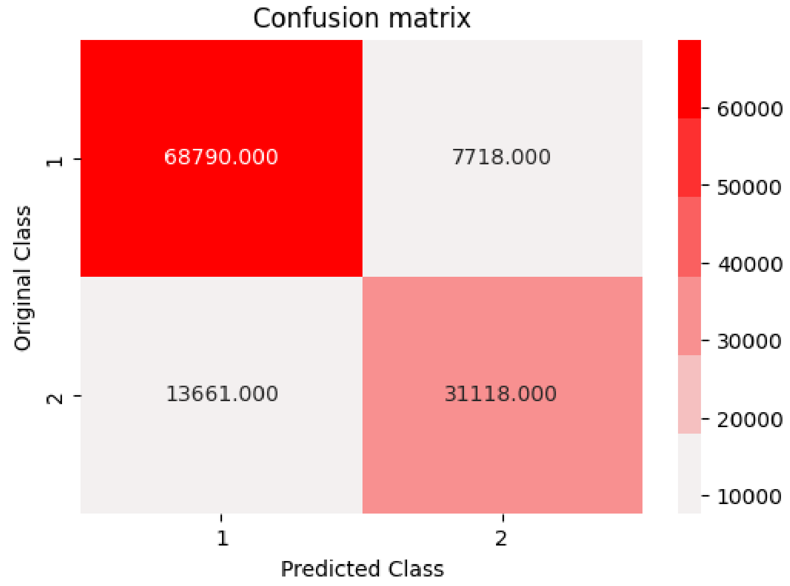
	Duplicate	Non-Duplicate
Accuracy	79%	79.1%
Precision	78.1%	79.5%
Recall	60.4%	90.1%
F1 Score	34.1%	42.2%

Basic + Advanced + Fuzzy Features



	Duplicate	Non-Duplicate
Accuracy	81	82.1%
Precision	79.3%	83.4%
Recall	69.6%	89.4%
F1 Score	37.1%	43.1%

Basic + Advanced + Fuzzy + Embedding Features

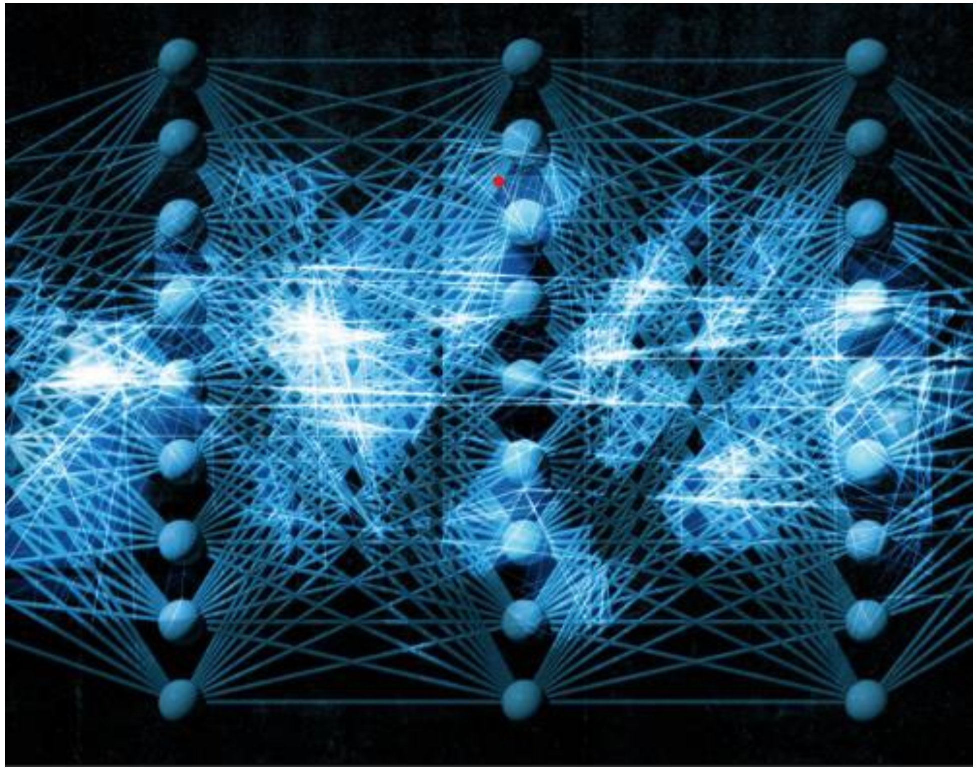


	Duplicate	Non-Duplicate
Accuracy	82.1%	82.37%
Precision	80.1%	83.4%
Recall	69.5%	89.9%
F1 Score	37.2%	43.2%

DEEP NEURAL NETWORK

Two different Deep learning models are used

- ❑ **Convolutional Neural Network (CNN)**
  - **1 CNN Layer**
  - **4 CNN Layer**
- ❑ **2 layer Bi-directional Long Short-Term Memory (BiLSTM)**



- Both of them use GloVe 300 dimensional word embedding
- Parallel Siamese network is used for training where Q1 and Q2 share the same weights.

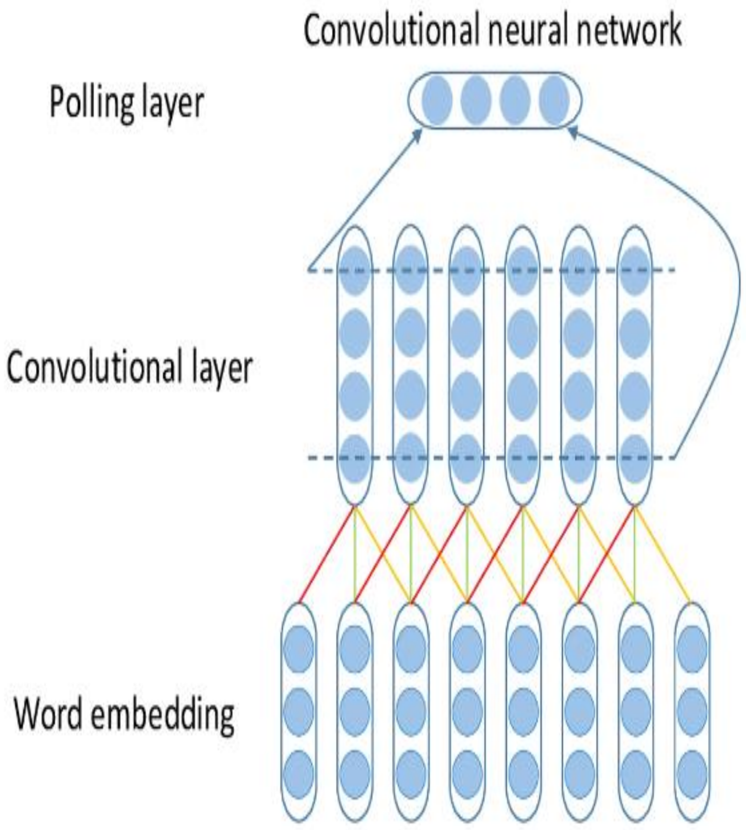
CONVULATIONAL NEURAL NETWORK

CNNs are a powerful type of neural network used for image recognition and classification tasks

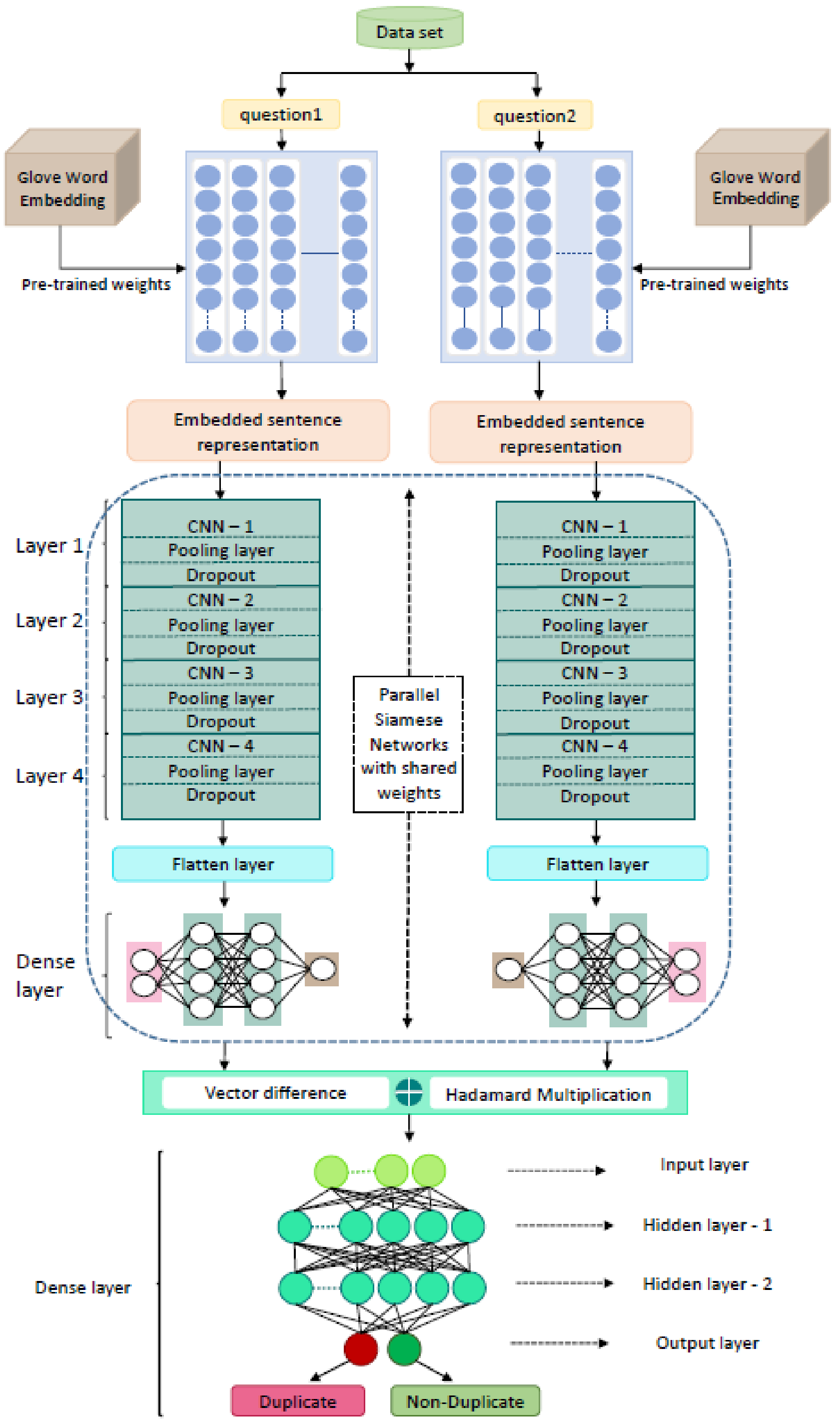
How CNNs work:

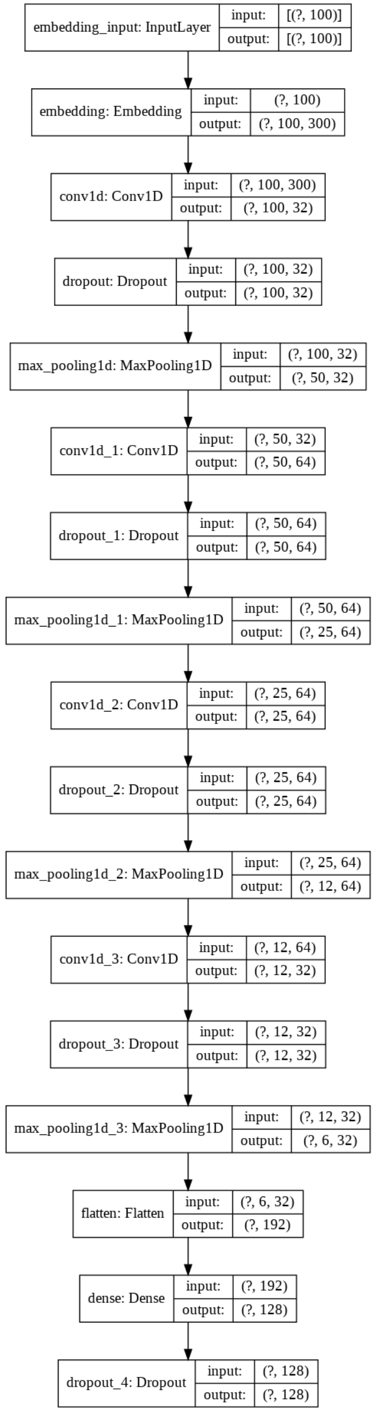
Convolutional layers: apply filters to the input to detect specific features

- Pooling layers: Down sample the output of the convolutional layers to reduce the number of parameters
- Fully connected layers: interpret the features learned by the convolutional layers and produce a classification output



CNN Architecture

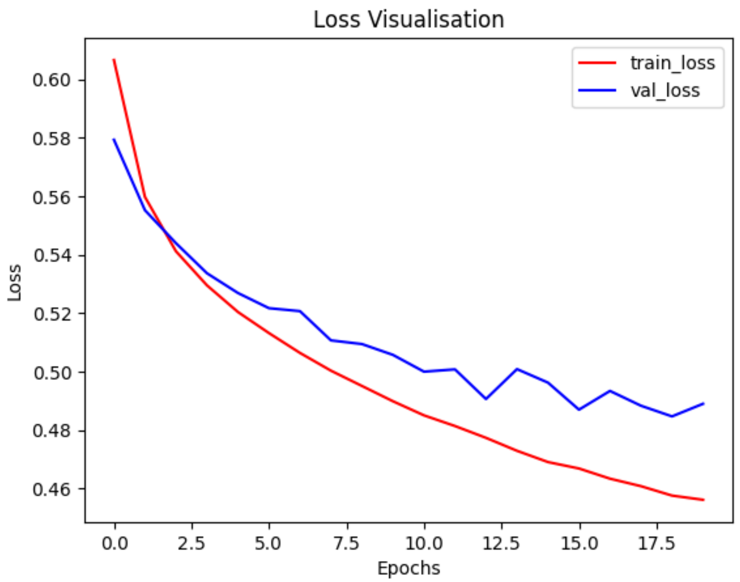


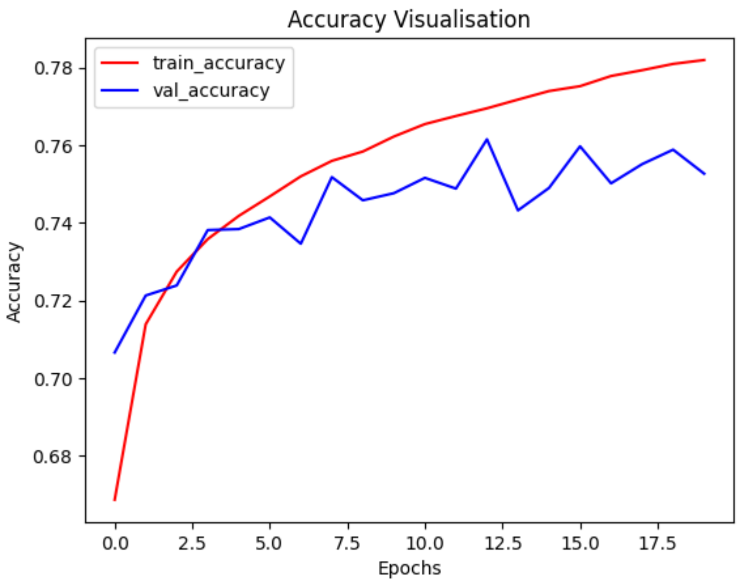


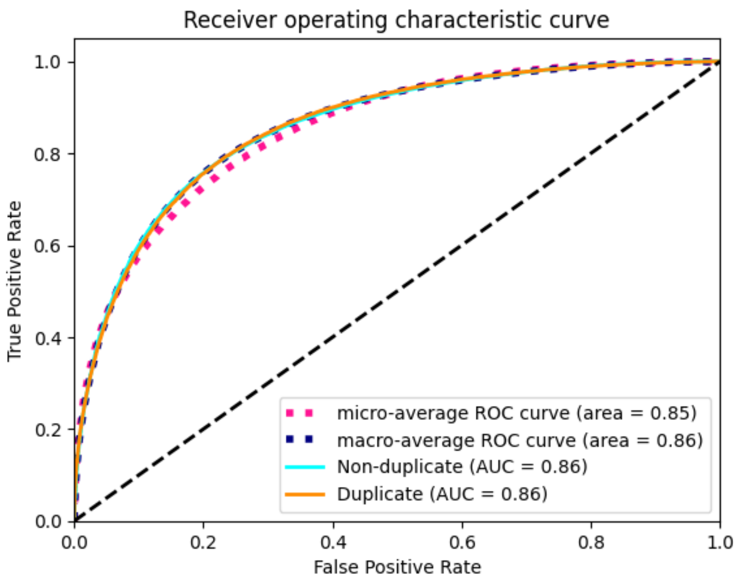

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 300)          28948500
_________________________________________________________________
conv1d (Conv1D)              (None, 100, 32)           28832
_________________________________________________________________
dropout (Dropout)            (None, 100, 32)           0
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 50, 32)            0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 50, 64)            10304
_________________________________________________________________
dropout_1 (Dropout)          (None, 50, 64)            0
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 25, 64)            0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 25, 64)            20544
_________________________________________________________________
dropout_2 (Dropout)          (None, 25, 64)            0
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 12, 64)            0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 12, 32)            6176
_________________________________________________________________
dropout_3 (Dropout)          (None, 12, 32)            0
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 6, 32)             0
_________________________________________________________________
flatten (Flatten)            (None, 192)               0
_________________________________________________________________
dense (Dense)                (None, 128)               24704
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
=================================================================
Total params: 29,039,060
Trainable params: 90,560
Non-trainable params: 28,948,500
```

Parallel Siamese network CNN 1-layer Results

- Accuracy Training 78.2%
- Accuracy Test 75.3%

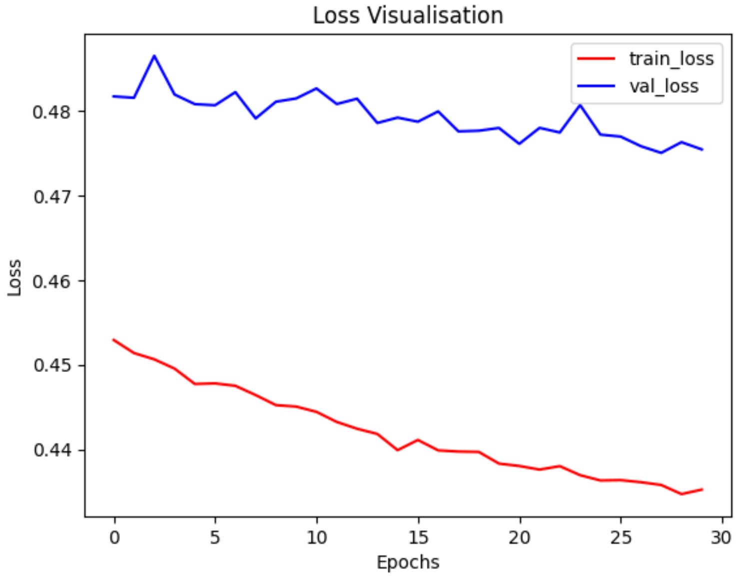


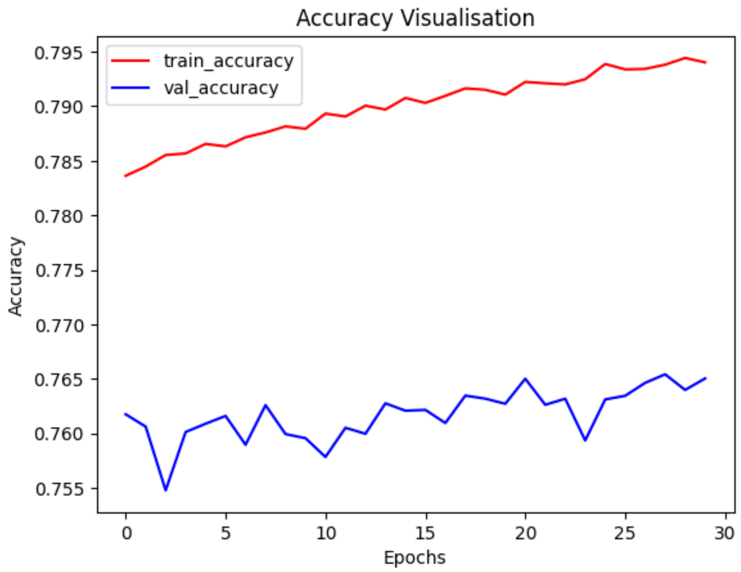


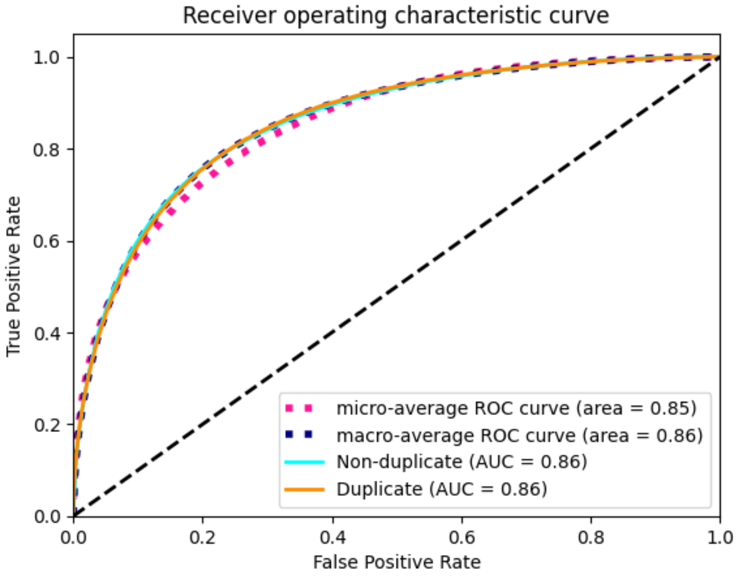


Parallel Siamese network CNN 4-layer Results

- Accuracy Training 79.4%
- Accuracy Test 76.5%







2-layer Bi-directional LSTM

A biLSTM is a sequence processing model that utilizes two LSTMs to increase the context available to the algorithm

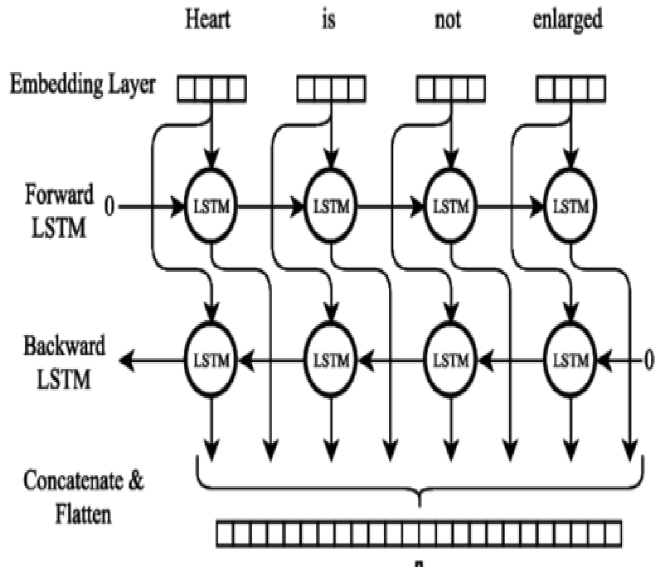
How biLSTMs work:

- Two LSTMs are used, one for processing the input sequence in a forward direction, and the other for processing it in a backwards direction
- The outputs of both LSTMs are then combined to produce the final output
- This approach effectively increases the amount of information available to the network, improving the context available to the algorithm

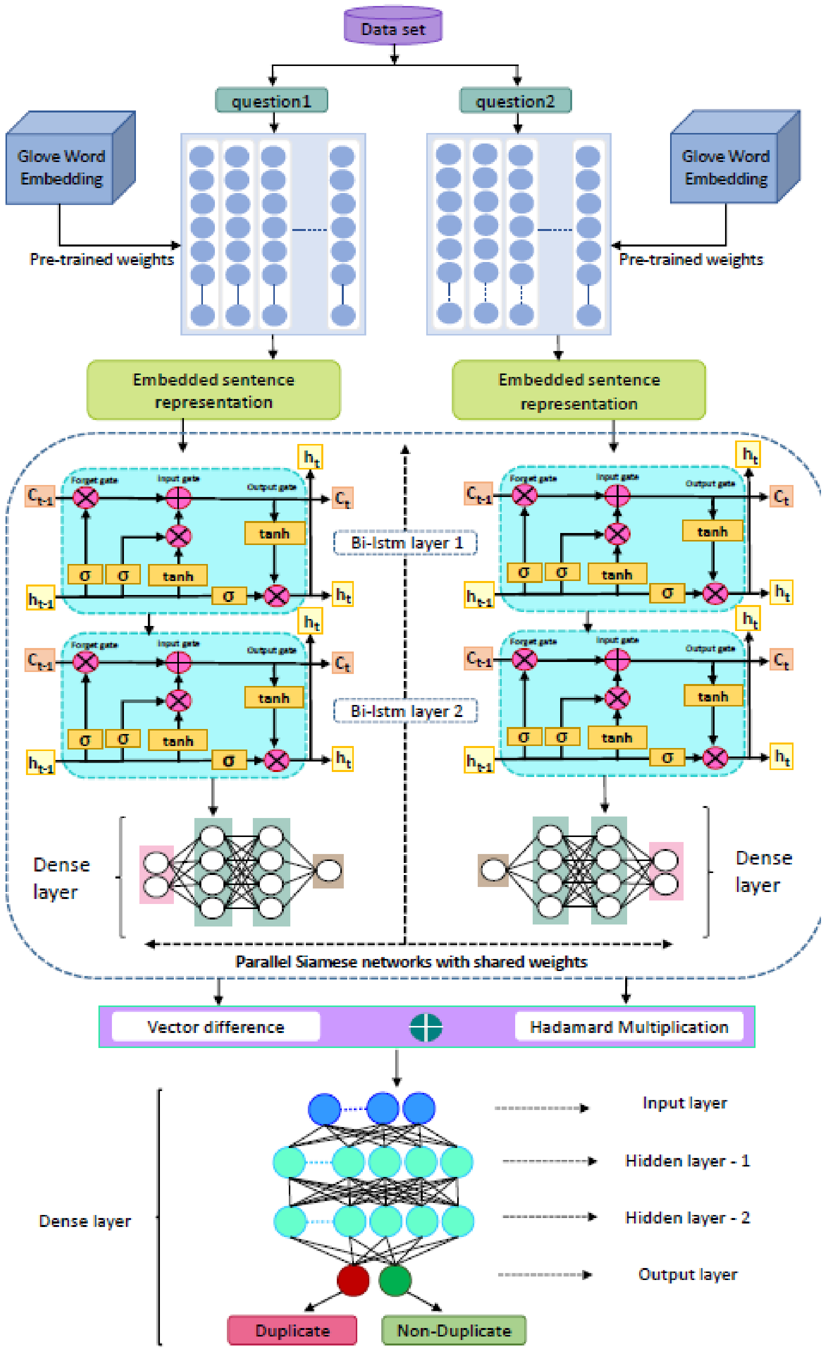
Benefits of using biLSTMs:

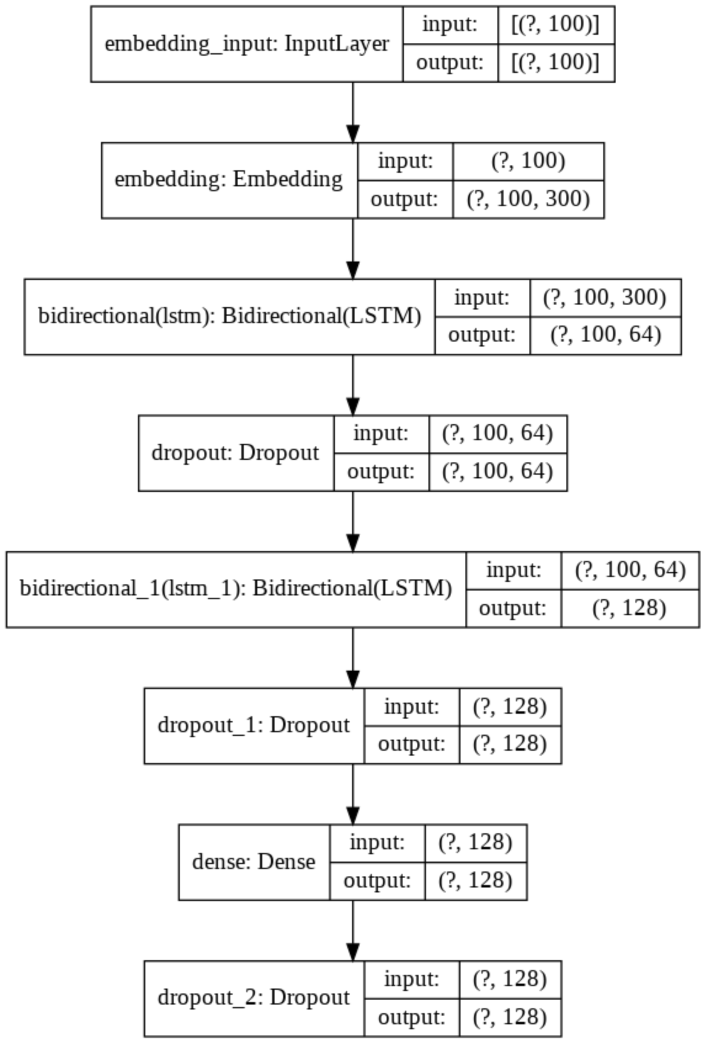
- Can capture both past and future information about a sequence

- Better able to understand the context of words in a sentence



BiLSTM Architecture

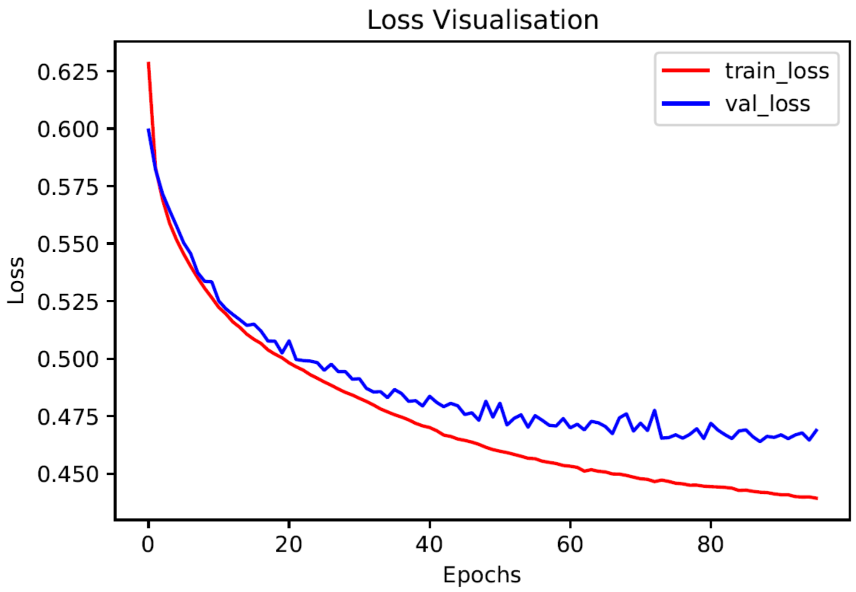


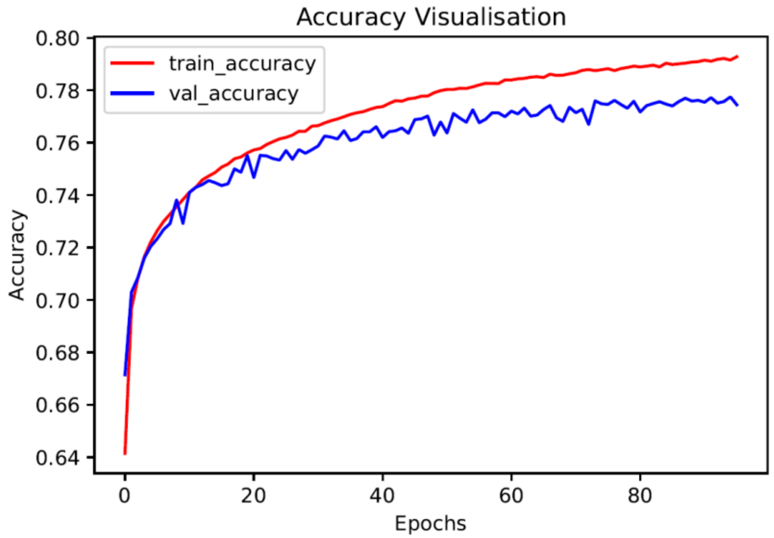


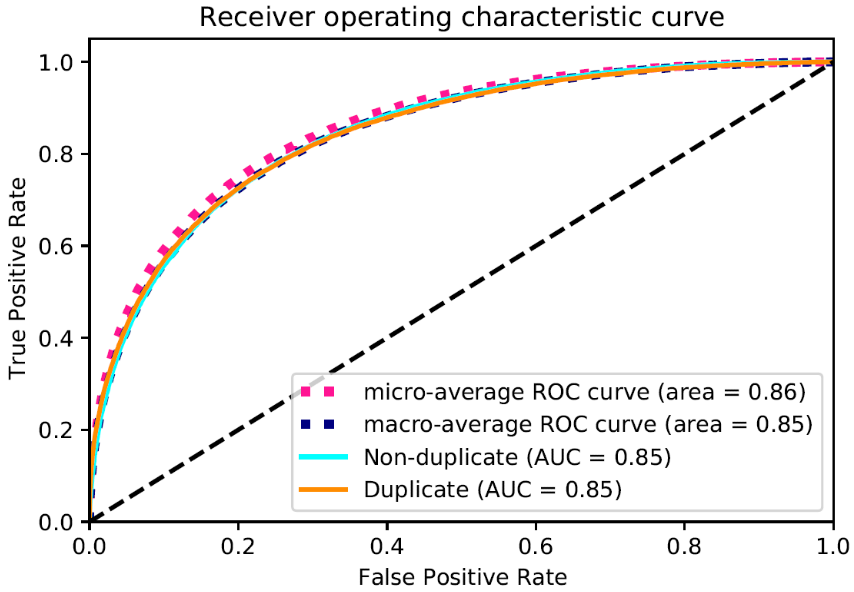

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 300)          26995200
_________________________________________________________________
bidirectional (Bidirectional (None, 100, 64)           85248
_________________________________________________________________
dropout (Dropout)            (None, 100, 64)           0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 128)               66048
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 128)               16512
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
=================================================================
Total params: 27,163,008
Trainable params: 167,808
Non-trainable params: 26,995,200
```

Parallel Siamese network Bi-LSTM Results

- Accuracy Training 79.0%
- Accuracy Test 76.8%







Conclusion

- Random Forest with 100 trees gives the best accuracy closely followed by XGBoost.
- Addition of advanced, fuzzy, and embedding features improves Random Forest performance, while increasing the number of trees further enhances it.
- Parallel Siamese network with CNN and Bi-LSTM has been used. The results are inferior to the classical ML techniques.
- GloVe word embeddings is exclusively used for deep learning.