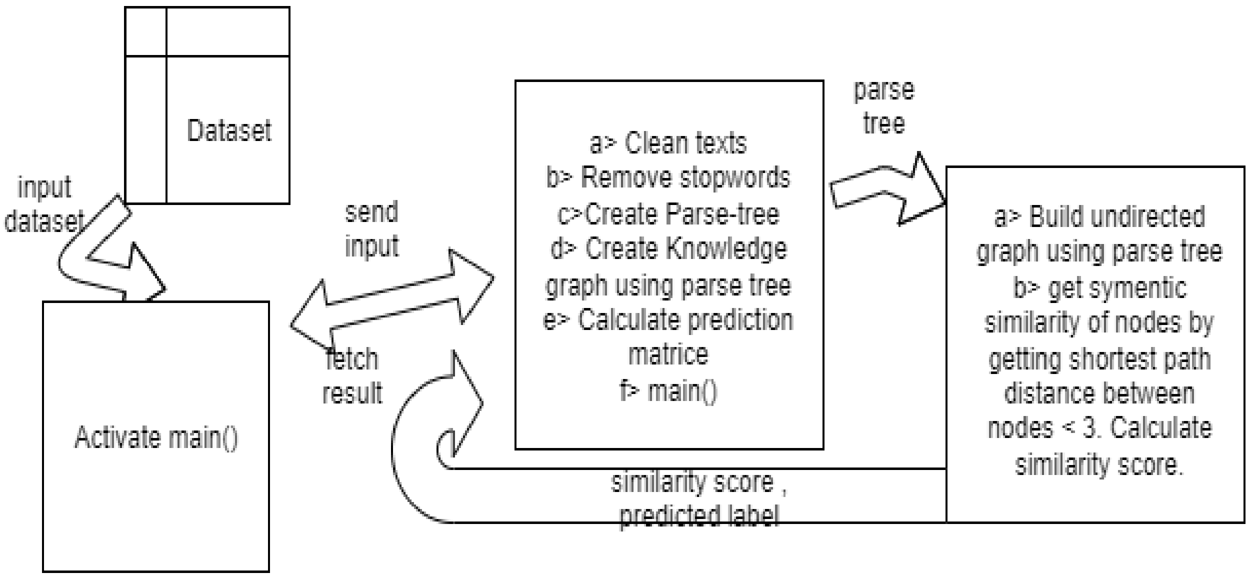
Experiment Report

1.Evaluation Dataset Details::

The dataset used in this experiment consists of three features or columns i.e. text, reason, label. Features 'text' and 'reason' are phrases in english which are either semantic or non-semantic. 'label' column represents the binary representation for semantic and non semantic classifications. Dataset consists of 9000 non-null values with mean 0.33 and standard deviation 0.471. The minimum value is 0 and maximum value is 1. 25th percentile of object in labels are 0, 50th percentile are 0 and 75th percentile is 1.

```
             label
count  9000.000000
mean      0.333444
std       0.471470
min       0.000000
25%       0.000000
50%       0.000000
75%       1.000000
max       1.000000
```

2.Model Design::


Dataset
input dataset
Activate main()
send input
fetch result
a> Clean texts
b> Remove stopwords
c>Create Parse-tree
d> Create Knowledge graph using parse tree
e> Calculate prediction matrice
f> main()
parse tree
a> Build undirected graph using parse tree
b> get symentic similarity of nodes by getting shortest path distance between nodes < 3. Calculate similarity score.
similarity score , predicted label

The reason for opting for a knowledge graph based model is because the dataset is having imbalance data. To apply ML/AI model we will be needing Dataset with balanced data i.e. well mixed of records of labels. In this Situation it would result in overfitting which will result in improper Prediction of labels for unseen dataset i.e. test dataset. If we closely See train dataset, major occurance of label 1 has been found. Also, graph based approach will be helpfull in understanding the model and the relation between nodes resulting in proper prediction capability in this case.

3.Data Preprocess::

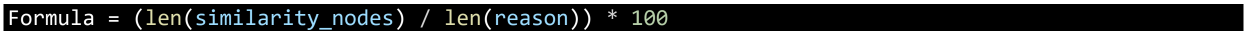
Dataset consists of english phrases. Data preprocessing will remove all unwanted characters from phrases eg. ‘ ’, ‘.’, etc. . Also, the unwanted Stop words are cleaned from the phrases to get only the meaningfull words Which can led us to make better predictions.

4.Creating Parse Tree::

To build a knowledge graph we need a tree that will break down the words in a phrase into a tree form with pos-tags. Pos-tags are one of the important feature to reach a conclusion if phrases or words are symentically similar or not. In this experiment Nouns, Adjectives,Verbs and Adverbs has been considered as Noun represents objects, adjectives and adverbs represents object properties and verbs represents object behavior. From cleaned phrases all the above were extracted and a parse tree for each text and reason has been created.

5.Building Knowledge Graph::

Using the parse tree undirected graph has been created as a knowledge graph. One can also create undirected weighted graph as knowledge graph. Intension was to find the distance from reason phrase node to text phrase node. The shortest distance has been considered as symentically similar.To find shortest distance between two nodes Dijkstras’s Algo is used. In this experiment in between source and destination node only <= 2 nodes has been allowed and only these nodes will be considered. Source and destination nodes must consist of same pos-tag in parse tree. To calculate score



```
Formula = (len(similarity_nodes) / len(reason)) * 100
```

the above equation has been used which will present a percentage of symentical similarity. If the percentage is greater than 60% then the reason phrase shall be awarded a label of 1 or else 0.

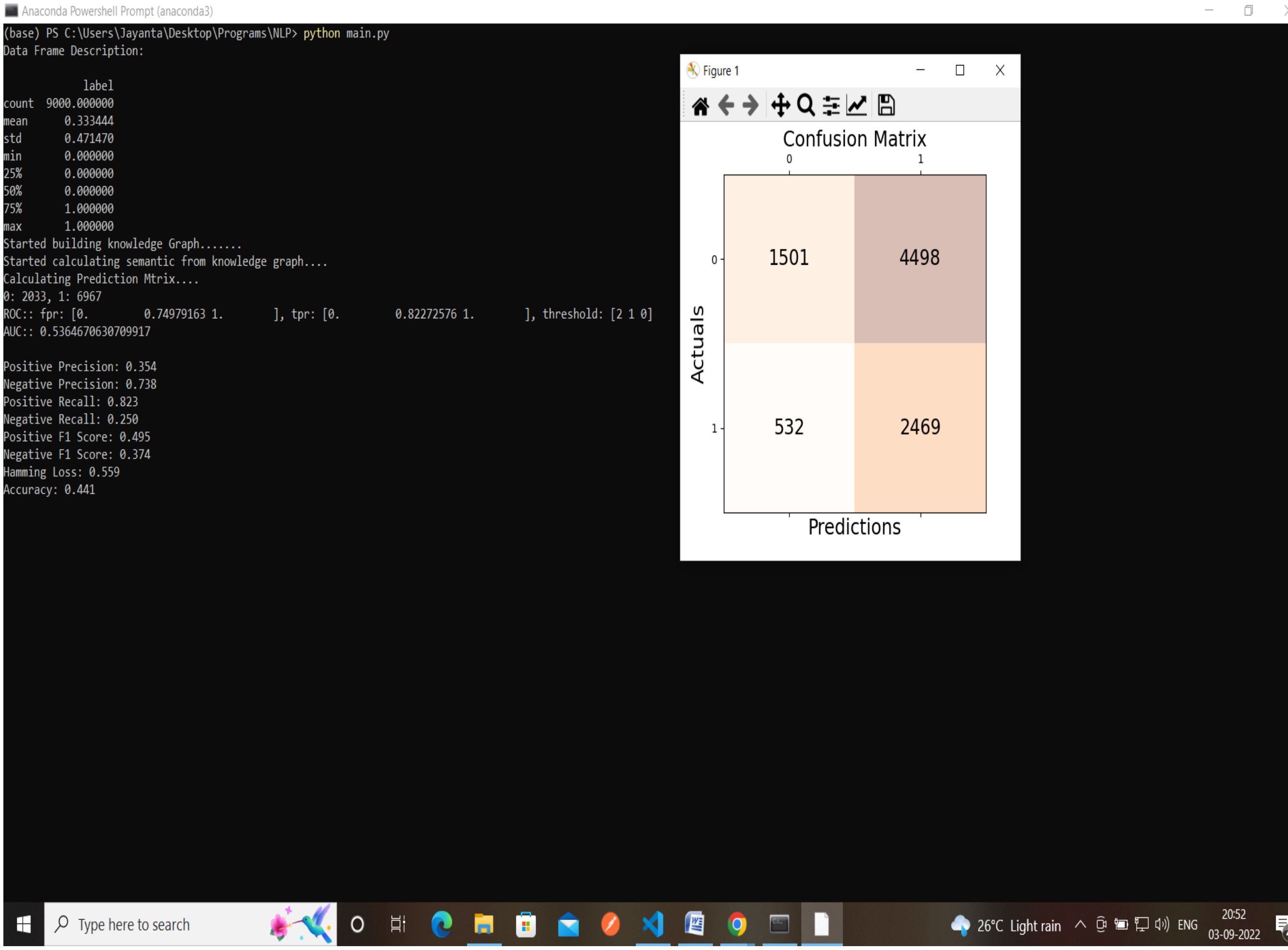
6.Calculating Evaluation Parameters::

To evaluate the performance of the model in this experiment AUC score, Precision score, Recall score, F1 score, Confusion matrix and Hamming loss score has been calculated.

Evaluation Parameter Analysis::

- **AUC score:** 0.5 < AUC <= 1 is considered. 0 beign worst, 1 beign best. Also AUC = 0.5 is also not considered good model. This model has scored AUC = 0.537 which satisfies condition 0.5 < AUC <= 1. Though model can be further be improved by creating a knowledge graph with more English words and experiment with different techniques using pos-tags.
- **Precision:** precision of negative class is intuitively the ability of the classifier not to label as positive a sample that is negative. The precision of positive class is intuitively the ability of the classifier not to label as negative a sample that is positive. 0 <= Precision_score <= 1. In this model with test data the precision for label 1 is 0.354 where precision for label 0 is 0.738.

- **Recall:** Recall of positive class is also termed sensitivity and is defined as the ratio of the True Positive to the number of actual positive cases. Recall of negative class is also termed specificity and is defined as the ratio of the True Negative to the number of actual negative cases. 0 <= Recall_score <= 1. In this experiment TPR(True Positive Rate) is 0.823 where TNR(True Negative Rate) is 0.250
- **Accuracy:** ratio of the number of correctly classified cases to the total of cases under evaluation. 0 <= Accuracy <= 1. In this experiment accuracy recorded is 0.441
- **F1 Score:** weighted average of recall and precision of the respective class. 0 <= F1 <= 1. In this experiment F1 score for model for label 1 is 0.495 where for label 0 is 0.374
- **Loss:** Hamming loss is the fraction of targets that are misclassified. 0 <= Loss <= 1. In this experiment loss recorded as 0.559.
- **Confusion Matrix:** In the evaluation data set the distribution of label is highly imbalanced as there are 5999 records having label as 0 whereas 3001 records has labels as 1. In the train dataset there are 0 records with label 0 whereas 2061 records with label 1 which is highly imbalanced dataset.



Anaconda Powershell Prompt (anaconda3)

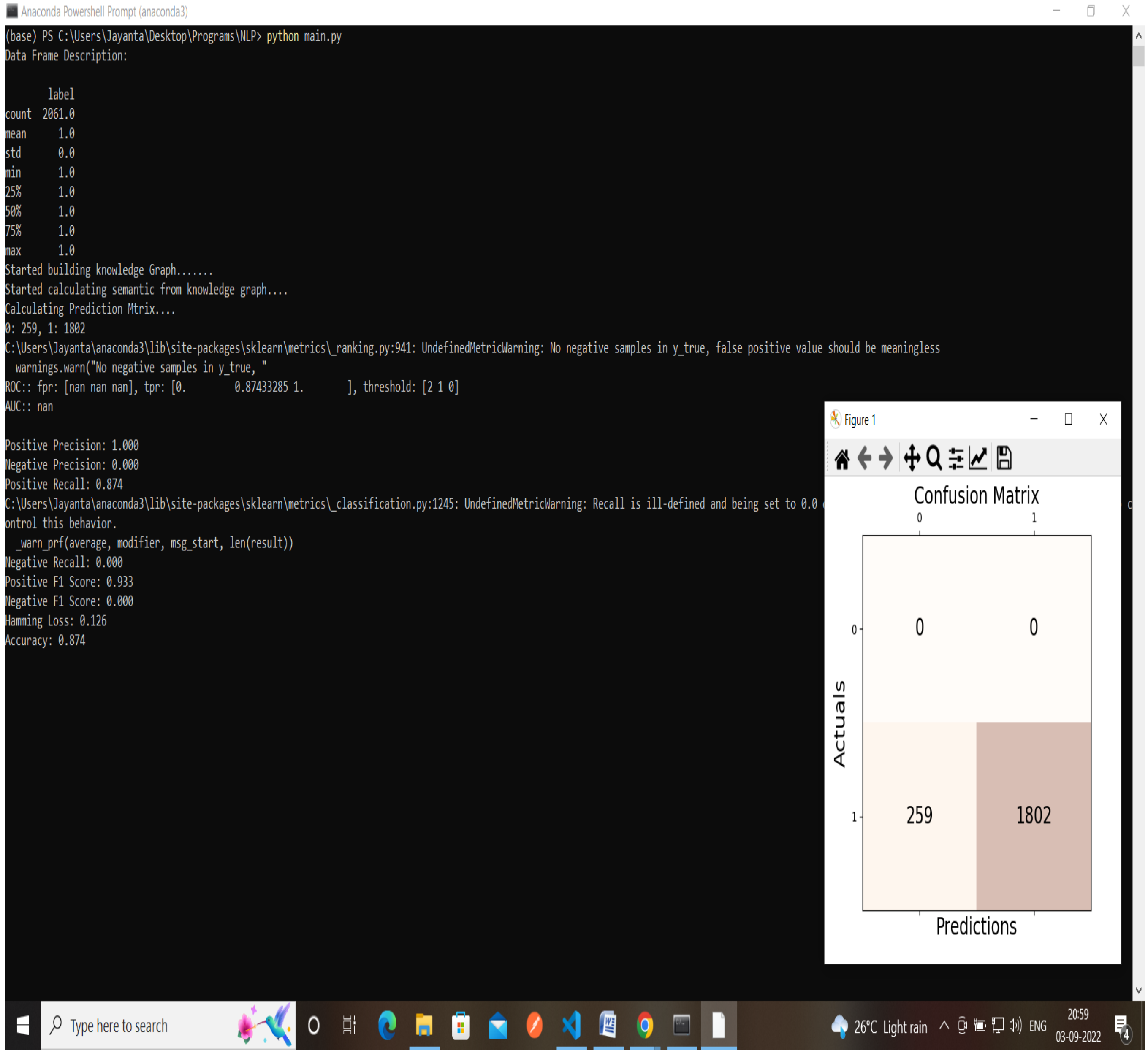
```
(base) PS C:\Users\Jayanta\Desktop\Programs\NLP> python main.py
Data Frame Description:

             label
count  9000.000000
mean      0.333444
std       0.471470
min       0.000000
25%       0.000000
50%       0.000000
75%       1.000000
max       1.000000
Started building knowledge Graph.......
Started calculating semantic from knowledge graph....
Calculating Prediction Mtrix....
0: 2033, 1: 6967
ROC:: fpr: [0.         0.74979163 1.        ], tpr: [0.         0.82272576 1.        ], threshold: [2 1 0]
AUC:: 0.5364670630709917

Positive Precision: 0.354
Negative Precision: 0.738
Positive Recall: 0.823
Negative Recall: 0.250
Positive F1 Score: 0.495
Negative F1 Score: 0.374
Hamming Loss: 0.559
Accuracy: 0.441
```

NOTE:

If we replace the evaluation dataset with train dataset and try to find semantic similarity of phrases then the parameters we get is in below image:



```
(base) PS C:\Users\Jayanta\Desktop\Programs\NLP> python main.py
Data Frame Description:

       label
count  2061.0
mean      1.0
std       0.0
min       1.0
25%       1.0
50%       1.0
75%       1.0
max       1.0
Started building knowledge Graph.......
Started calculating semantic from knowledge graph....
Calculating Prediction Mtrix....
0: 259, 1: 1802
C:\Users\Jayanta\anaconda3\lib\site-packages\sklearn\metrics\_ranking.py:941: UndefinedMetricWarning: No negative samples in y_true, false positive value should be meaningless
  warnings.warn("No negative samples in y_true, "
ROC:: fpr: [nan nan nan], tpr: [0.         0.87433285 1.        ], threshold: [2 1 0]
AUC:: nan

Positive Precision: 1.000
Negative Precision: 0.000
Positive Recall: 0.874
C:\Users\Jayanta\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1245: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0
ontrol this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
Negative Recall: 0.000
Positive F1 Score: 0.933
Negative F1 Score: 0.000
Hamming Loss: 0.126
Accuracy: 0.874
```

	0	1
0	0	0
1	259	1802