Insult Detection in Social Commentary

Chirag Khurana **MT17010**
Pallavi S. rawat **MT17034**
Shubham Goyal **MT17055**

Introduction

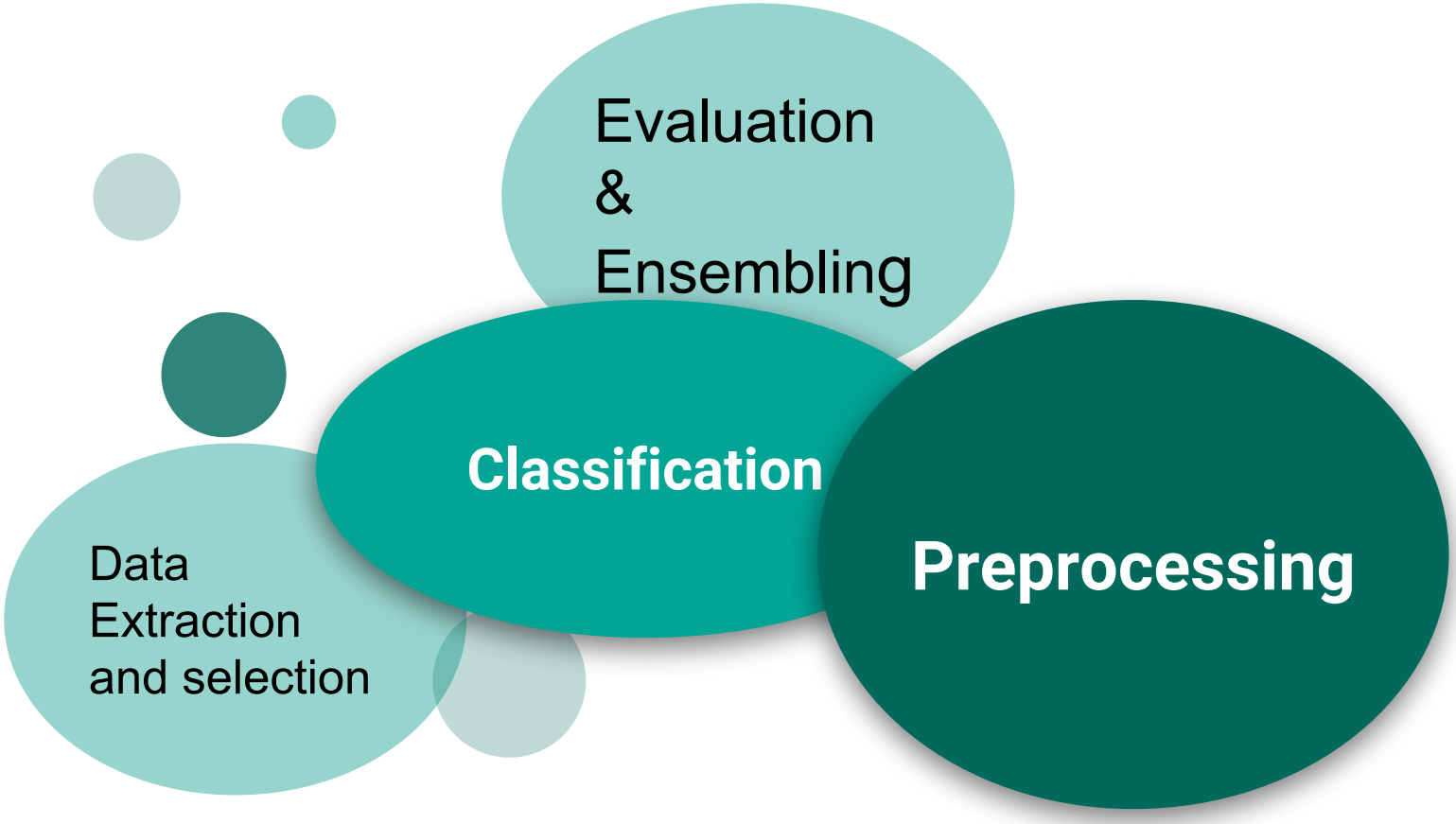
- The main area of focus in our project is to **detect if a comment** or a remark in a post or in a conversation **is insulting or not.**
- The goal is to **create a classifier** which could operate on a variable test set to do classification with good performance.

Application & future scope

- ➔ It can be put to use in **blacklisting** abusive users, **child-safe web crawling**, **filtering** in websites or webpages.
- ➔ This application can further be projected in Detection of **graphic insults** (I.e. **meme insults**, or mentioning someone in comments of a meme)

DATASET:

- Twitter comments
- Diverse variety
- The dataset has 5000+ comments.


Evaluation
&
Ensembling
Classification
Preprocessing
Data
Extraction
and selection

Resources added:

sortedBadWords_for_checking.txt

Sorted collection of bad words and phrases. (self-compiled)

Train_features.csv

It is a train.csv + features file that has the computed value of the features.

(code- computed)

ConvertedBadWords.txt

Word Collection to clean file of misspelling escapes.

Pre Processing

- removing extra spaces(\n,\s etc.),
- url removal,(Eg https://www……)
- html tags removal (Eg.
 <H> etc)
- word correction-level 1(coool to cool),
- word Expansion, (Eg . u -> you)
- Special Character tagging (Eg. *$@# -> TOKEN)
- bad Word Correction -level 2 (Gaarbagee -> garbage)

Result (without preprocessing):

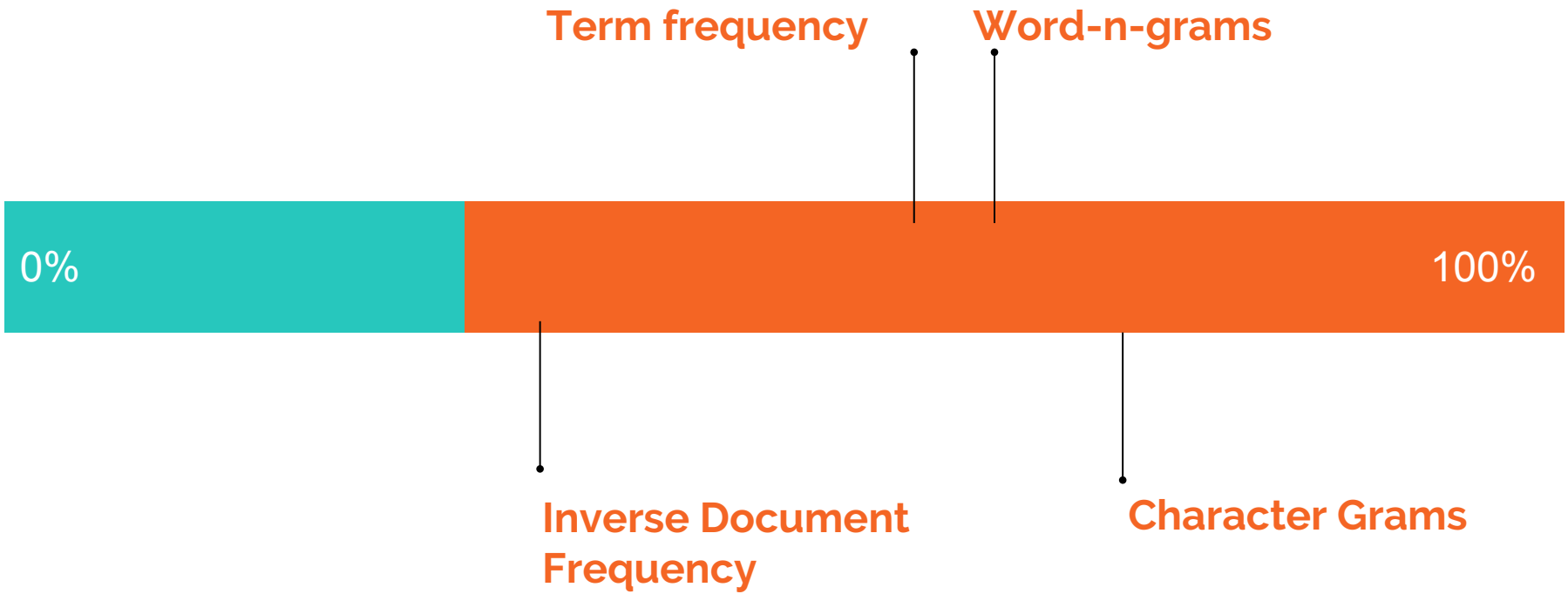
Result (after preprocessing)-

Feature Extraction

- ➔ Term Frequency- Inverse Document frequency
- ➔ word-grams
- ➔ Character n-grams

- ➔ Capital Word ratio sentence length
- ➔ Curse word ratio sentence length
- ➔ Curse word ratio positive word
- ➔ Short sentence ratio total sentences
- ➔ Lexical score -Distance
- ➔ Lexical Score -Dependency

Performance with addition of Features:



Classification

We had tried

various classifiers, simple neural Net ,even Multilevel classification using neural Nets with single hidden layer for classification task.

Models Used :

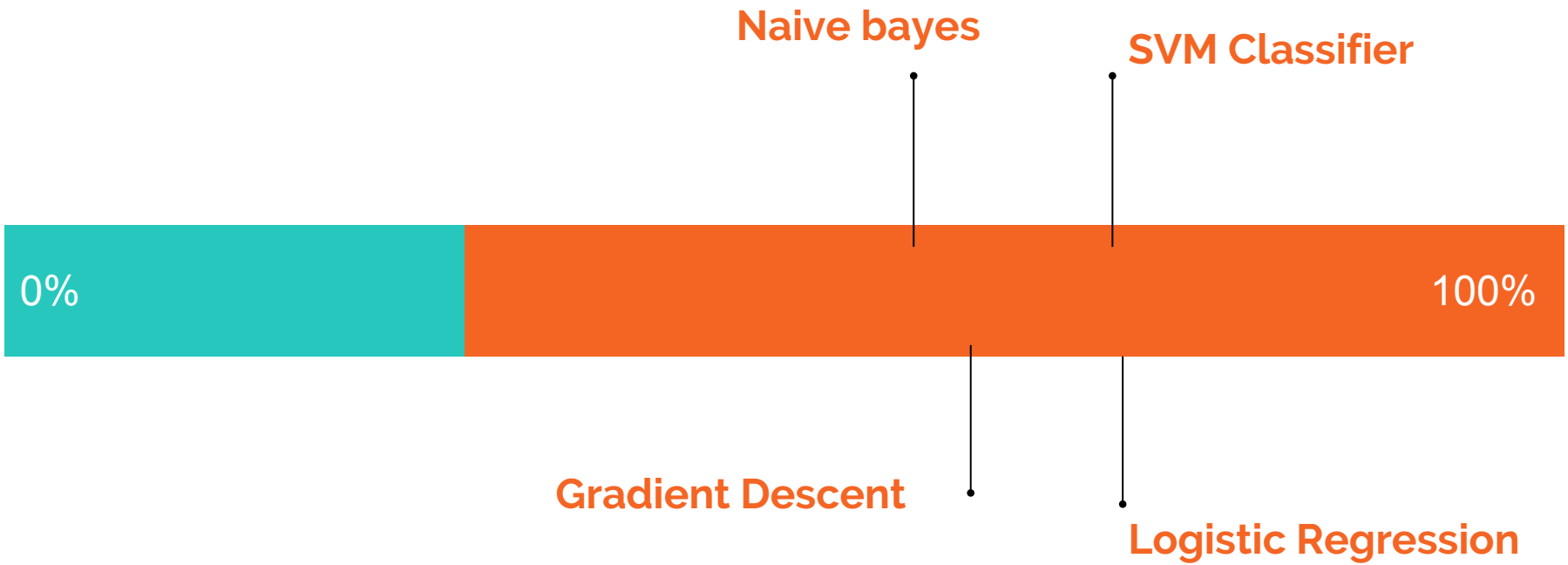
- Multinomial Naive Bayes
- Stochastic Gradient Descent
- SVM Classifier
- Logistic Regression
- MLP classifier(Neural Network)

Finally used:

SVM Classifier

Logistic Regression

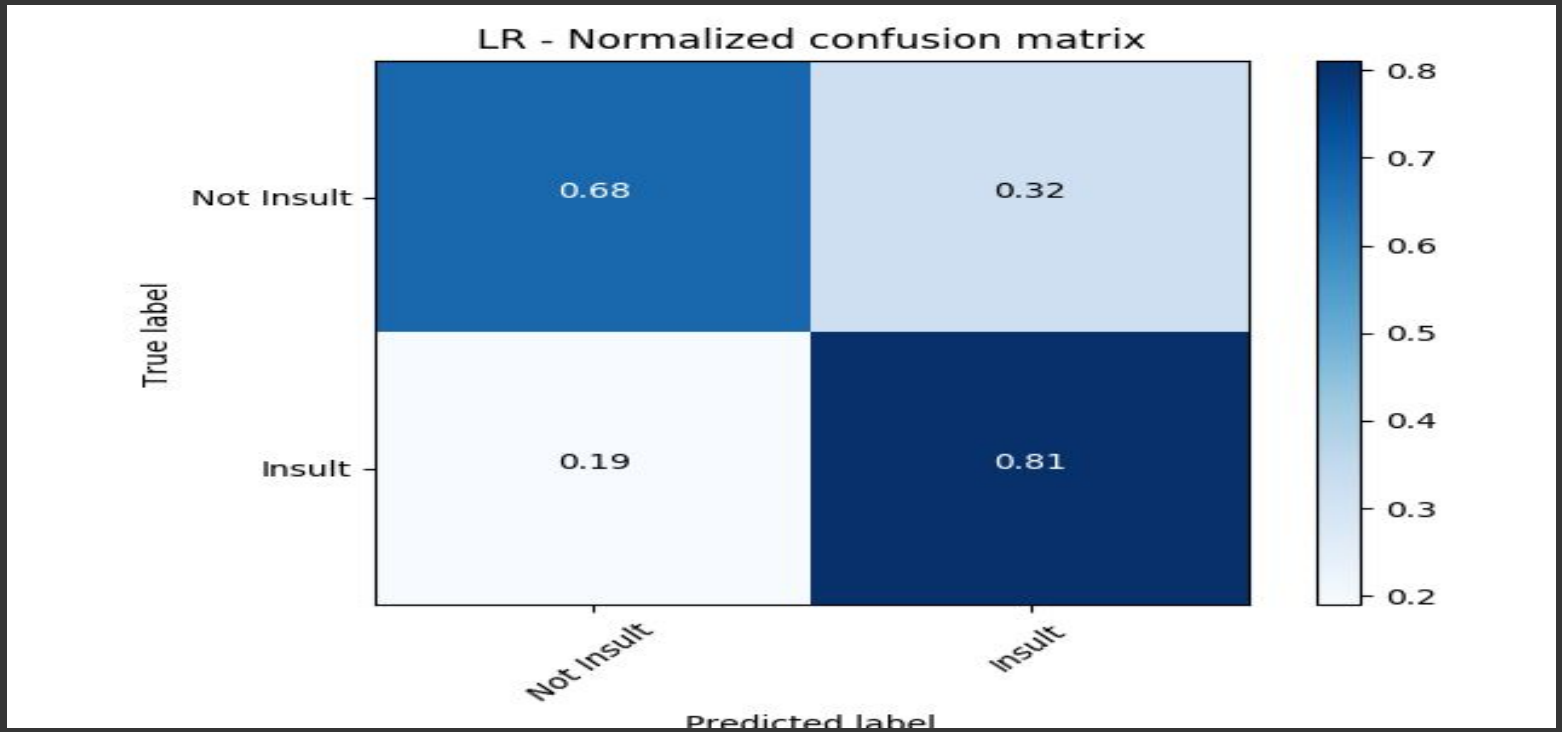
Variations in results with different classifiers:



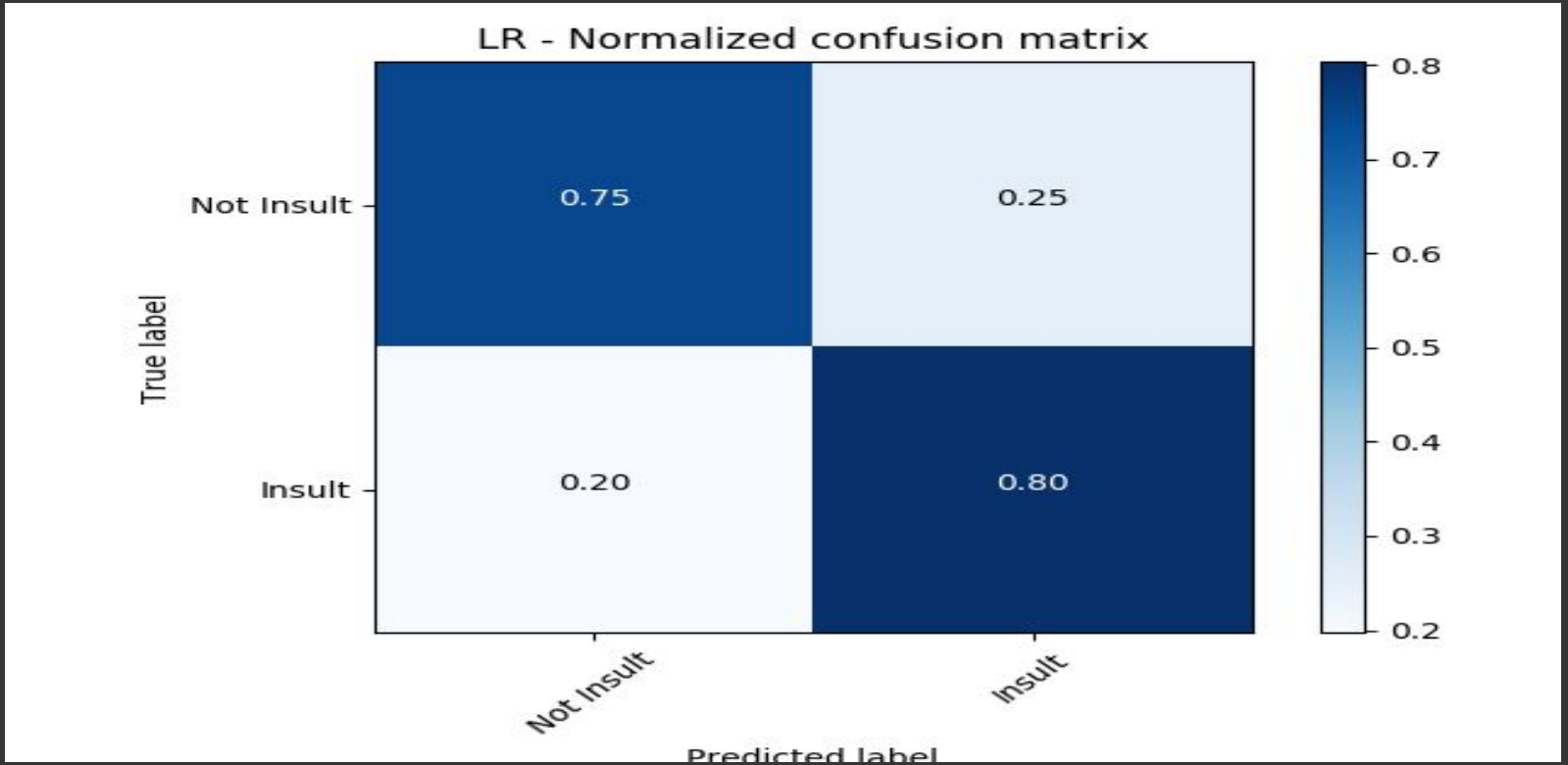
Evaluation

- Accuracy
- Confusion Matrix
- Area Under the curve

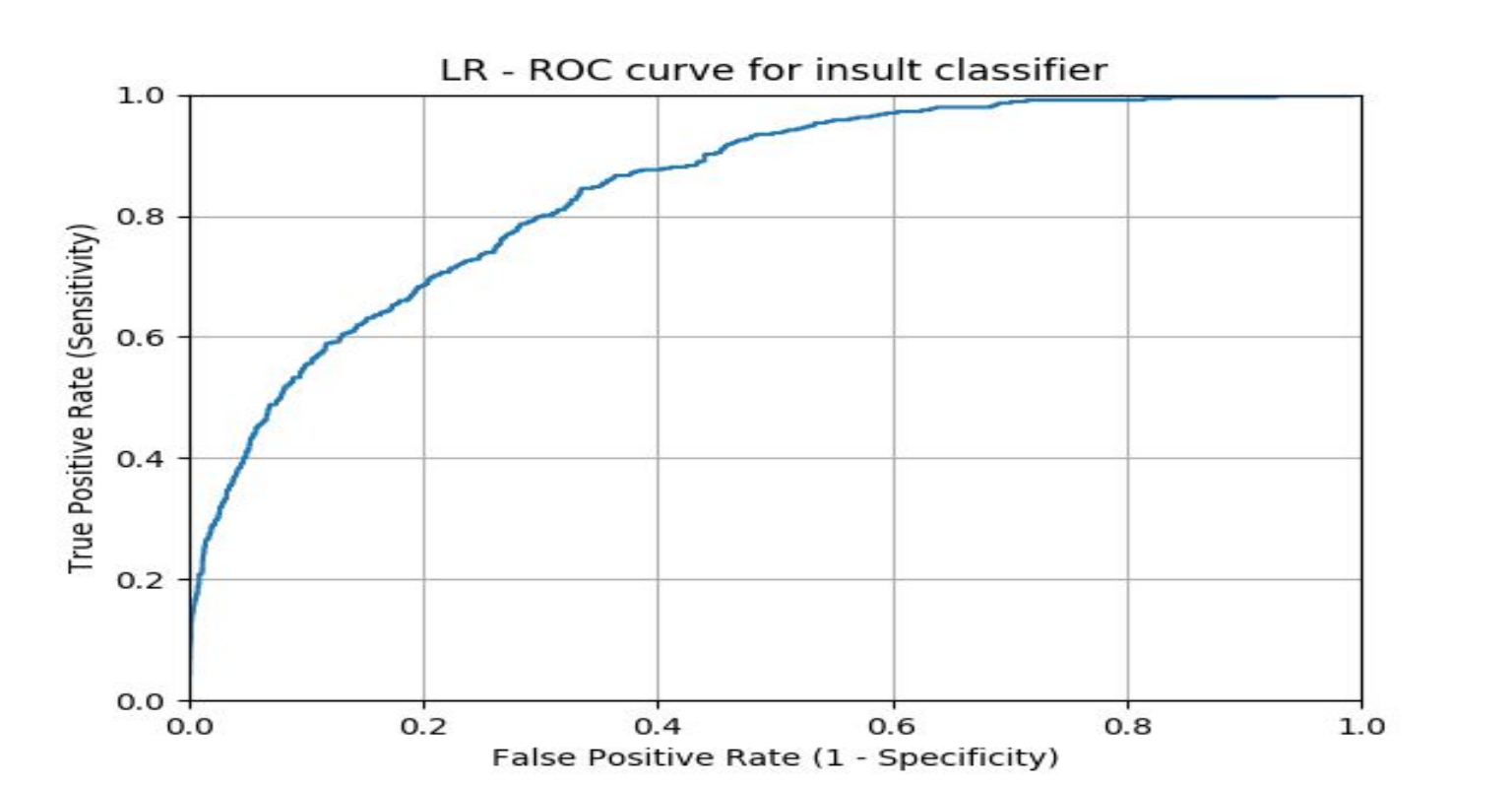
Confusion Matrix- LOGISTIC REGRESSION



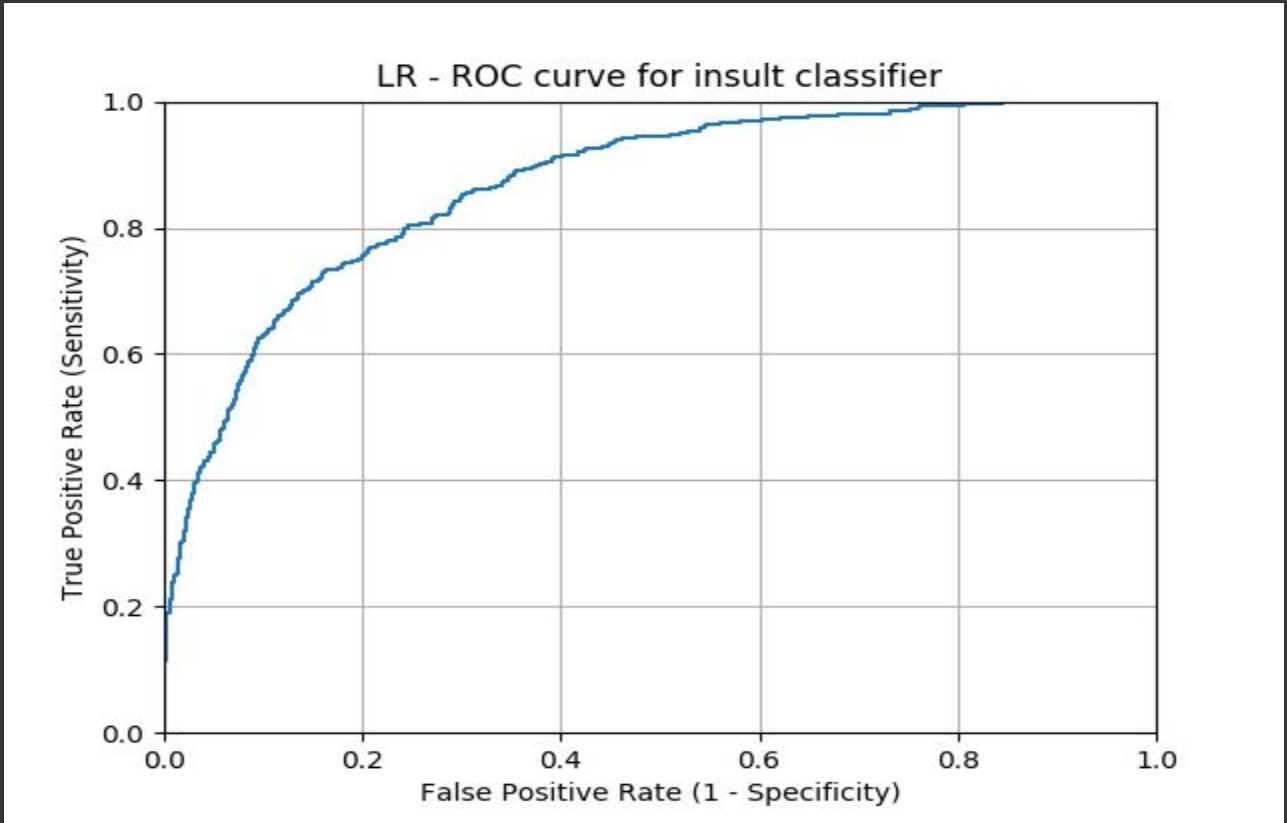
Confusion Matrix- LOGISTIC REGRESSION 2



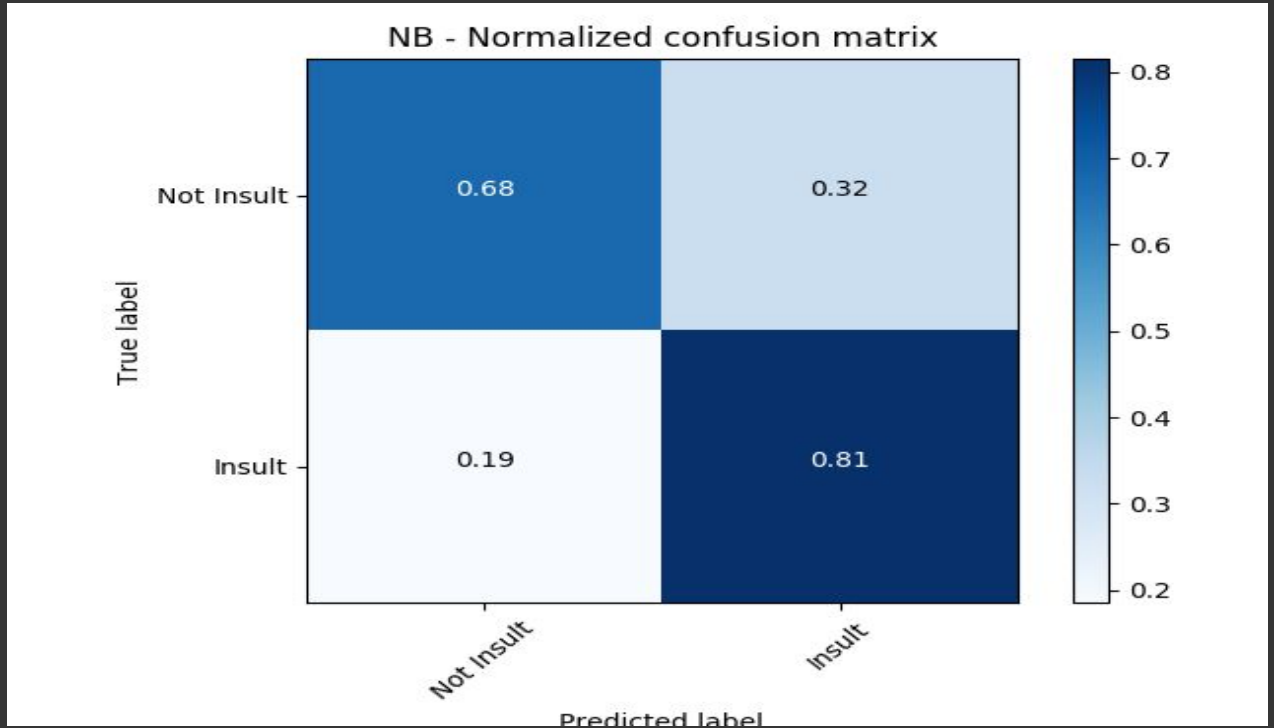
AUC - LOGISTIC REGRESSION



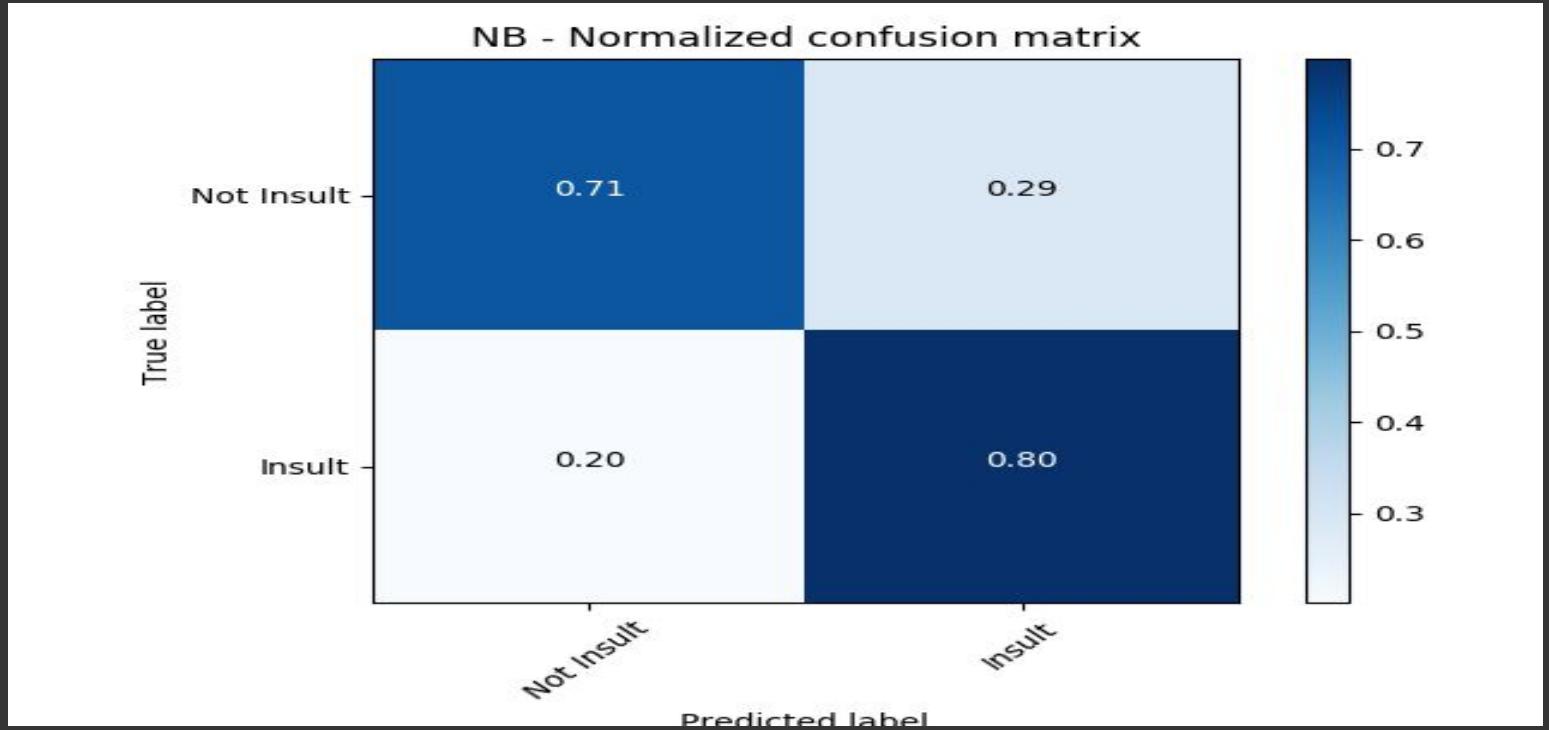
AUC - LOGISTIC REGRESSION -2



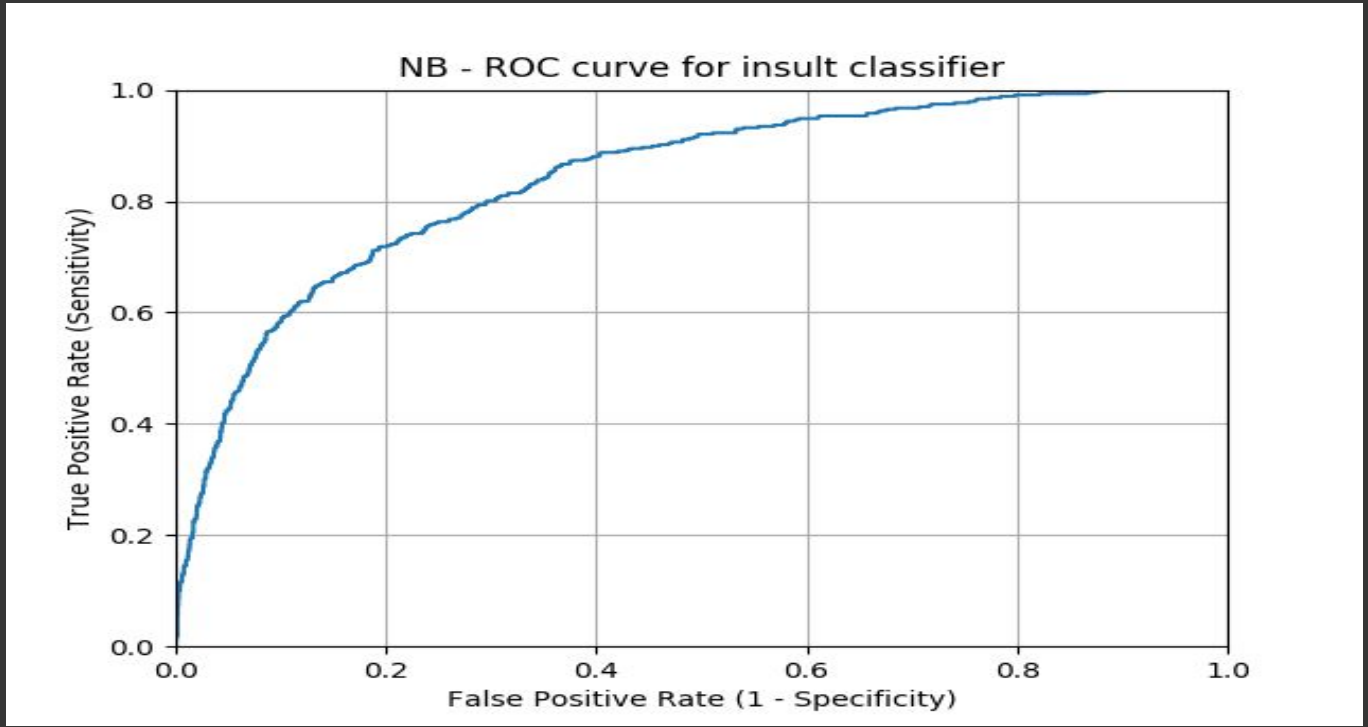
Confusion Matrix- Naive Baiyes



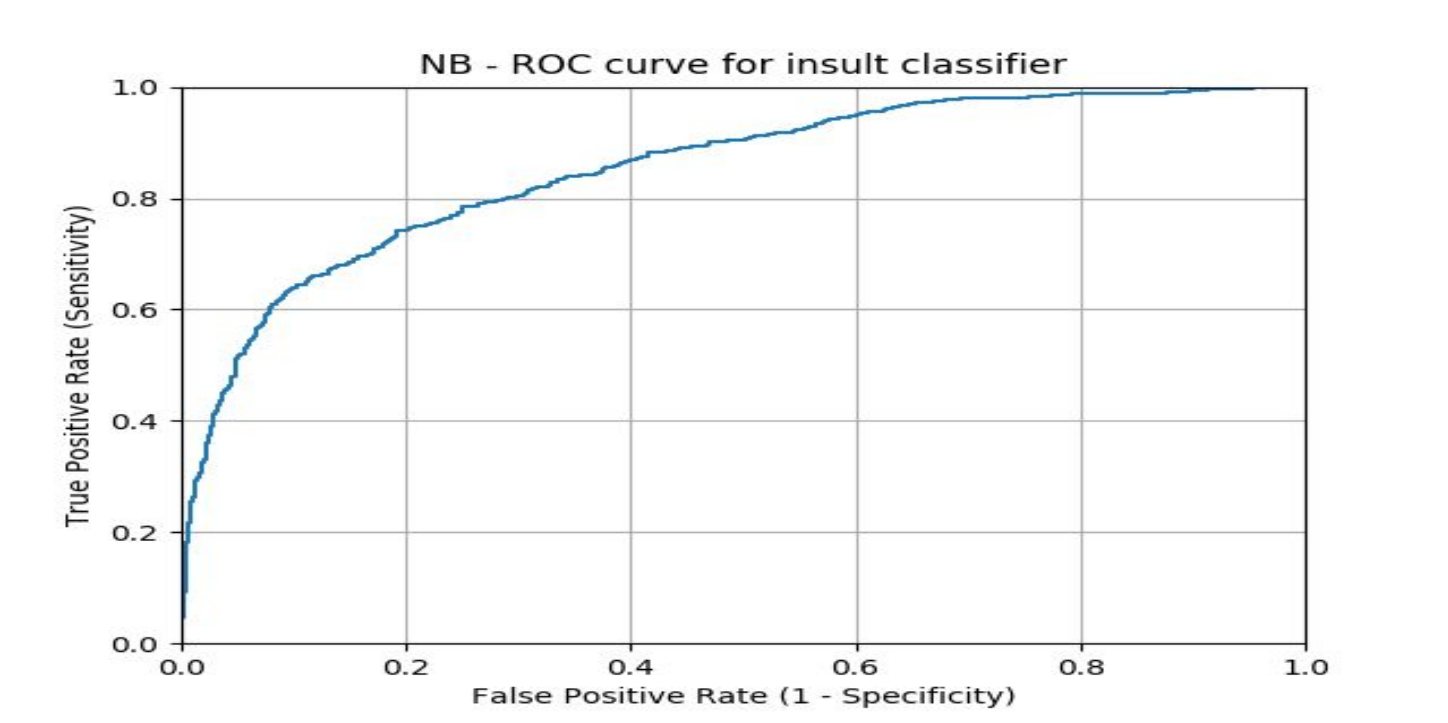
Confusion Matrix- Naive Baiyes 2



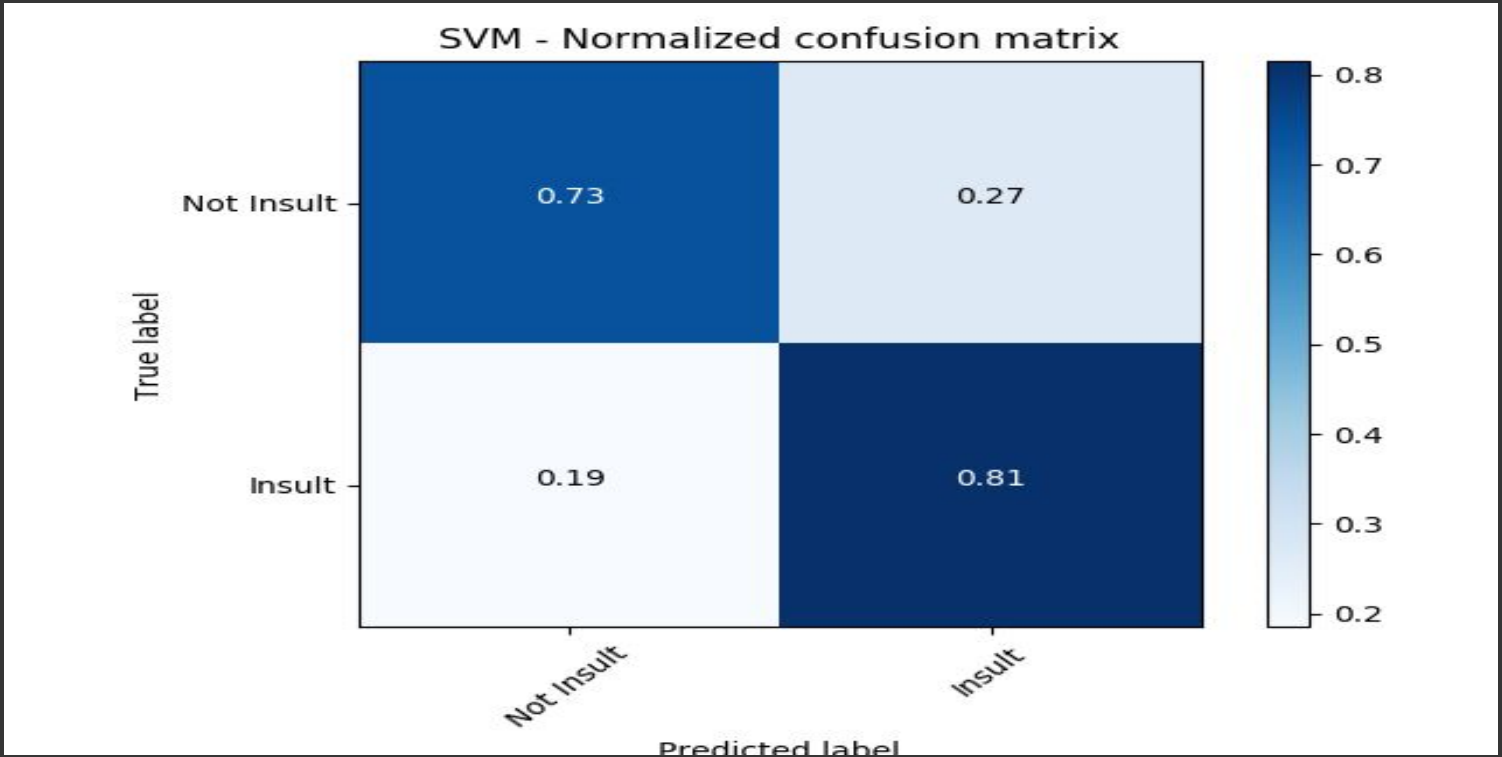
AUC - Naive Baiyes



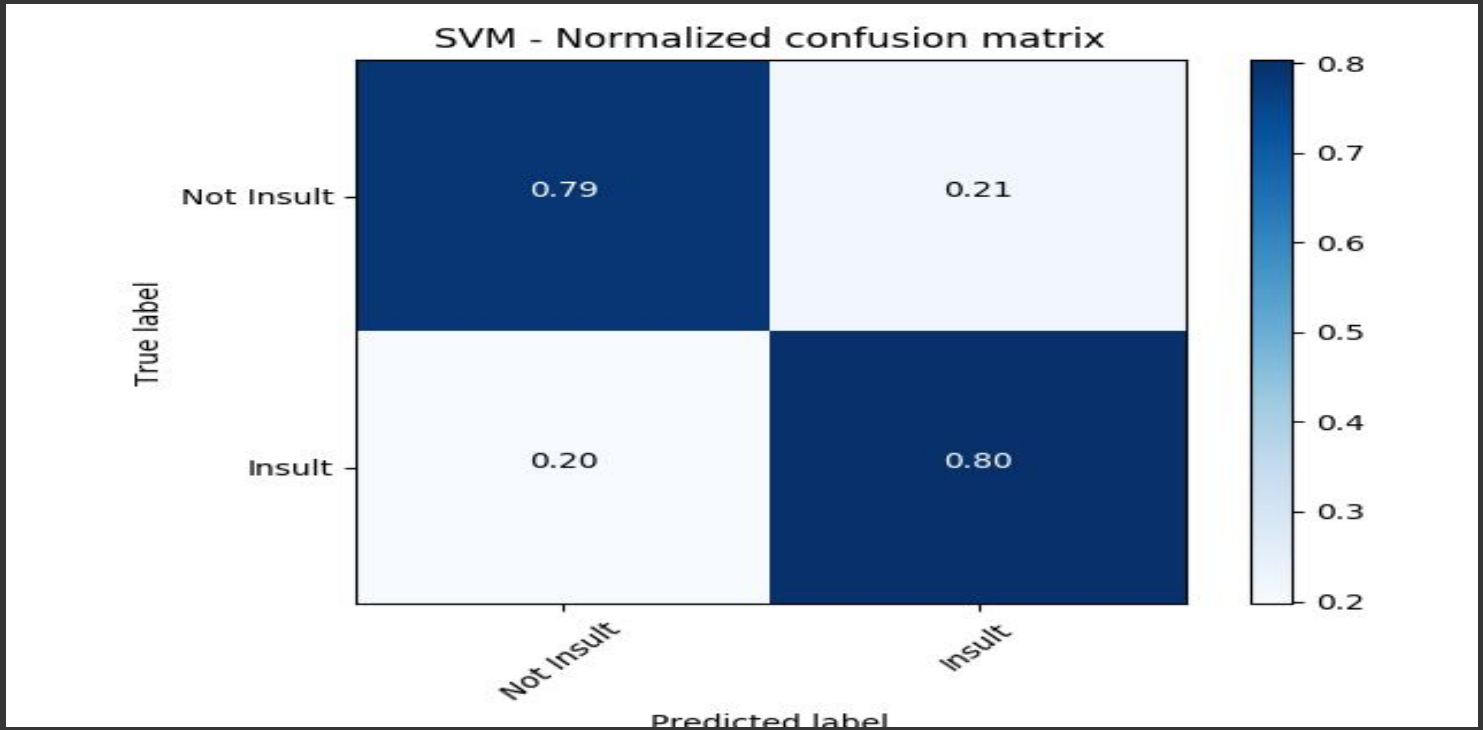
AUC - Naive Baiyes -2



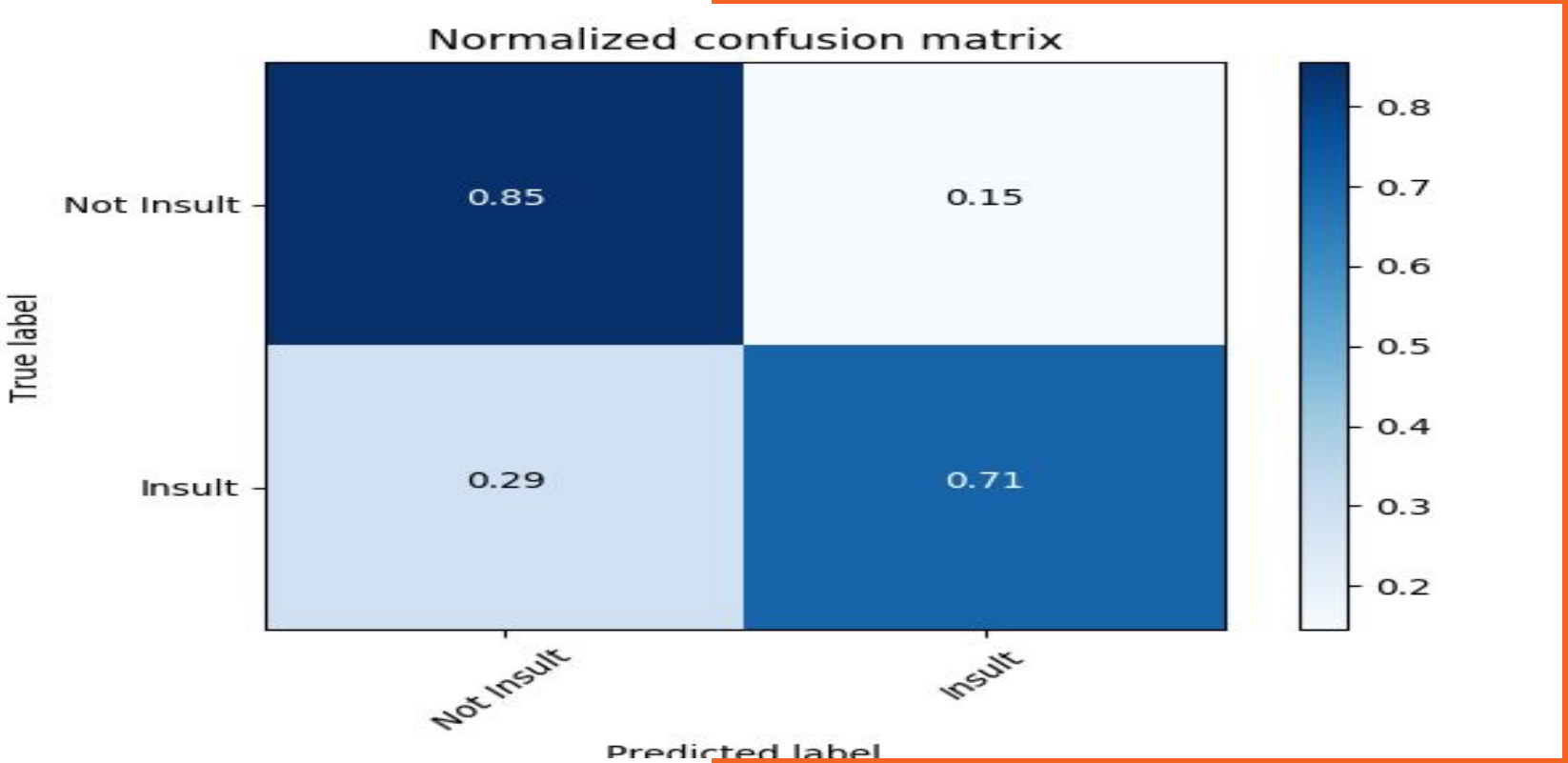
SVM- Confusion Matrix



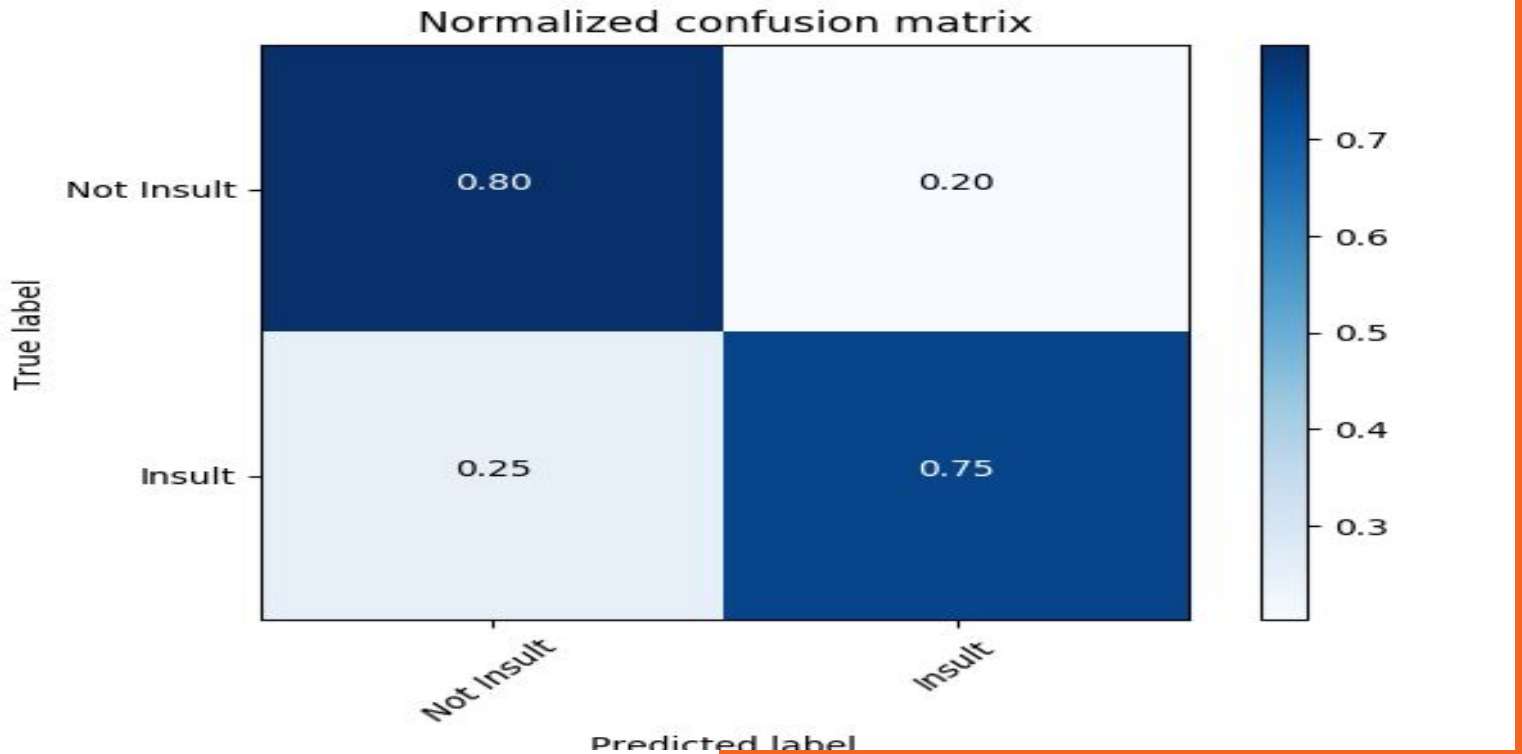
SVM- Confusion Matrix-2



ENSEMBLING:


Normalized confusion matrix
Not Insult
Insult
True label
0.85
0.15
0.29
0.71
Not Insult
Insult
Predicted label
0.8
0.7
0.6
0.5
0.4
0.3
0.2

ENSEMBLING 2:



Takeaways & Learning Outcomes:

We have not used neural networks in our project , for two reasons, firstly , for 1000,400,700 neurons and 1-3 hidden layers the accuracy touched 79% .Secondly, memory contraints restricted us from exploring it further.

Finally,

A **working** Model

http://labs.chiragkhurana.com/iiitd/nlp/insult-detection

References:

Multi-level classifier for the detection of insults in social media

https://goo.gl/snmwMa

Detecting Offensive Language in Social Media to Protect Adolescent Online Safety

https://goo.gl/CVT9Ci

THANK YOU !