Style Guidelines for Final Year Project ReportsSpotting Spammer using Group Spammer Behaviour Analysis

Final Year Project

Session 2016-2020

A 4th Year Student

A project submitted in partial fulfilment of the

COMSATS University Islamabad (CUI) Degree

of

BSc. (Hons.)BS in Computer Science (BSCS)



Department of Computer Science

COMSATS University Islamabad, Lahore Campus

20 December 2019

**Project Registration**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project ID (for office use) | | |  | | | | |
| Type (Nature of project) | | | [ ] **D**evelopment [ ] **R**esearch [ ] **R**&**D** | | | | |
| Area of specialization | | | Data Mining | | | | |
| **Project Group Members** | | | | | | | |
| Sr.# | Reg. # | Student Name | | CGPA | Email ID | Phone # | Signature |
| (i) | CUI/SP16-BCS-187/LHR | Kafeel Ahmad Butt | | 3.30 | [kafeelbutt45@gmail.com](mailto:kafeelbutt45@gmail.com) | +923044426671 |  |
| (ii) | CUI/SP16-BCS-019/LHR | Sheikh Muhammad Qasim | | 3.06 | q.shaikh14@gmail.com | +923054129775 |  |
| (iii) | CUI/SP16-BCS-083/LHR | Omer Tariq | | 3.22 | Om3r.Tariq97@gmail.com | +923157578588 |  |

# \*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others.

# Plagiarism Free Certificate

This is to certify that, I am **Kafeel Ahmad Butt** S/o **Naseer Ahmad Butt**, group leader of FYP under registration no CUI/SP16-BCS-187/LHR at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is **9%** that is less than 20%, an acceptable limit by HEC. The report is attached herewith as Appendix A.

Date: **16-Dec-2019** Name of Group Leader: **Kafeel Ahmad** Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Supervisor: Dr. Hamid Turab Mirza Co-Supervisor (if any): \_\_\_\_\_\_\_\_\_\_\_\_\_

Designation: Assistant Professor Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HoD: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

Online shopping has become a metamorphic phenomenon in the IT world. Online reviews are considered the best source of customer’s opinion about a product, and an asset for customers, and organizations for making important buying decisions. Unfortunately, in order to increase the number of profits, promotion or even to demote a rival product, deceptive reviews (Spam) mislead prospective customers to buy the best product and organizations in decision making. Works have been proposed on detecting individual spammer reviews. But group review spamming, which includes a group of swindlers working together to post fake online reviews for promoting or demoting a product/s, has become more damaging. More the size of the group more difficult it is to differentiate them as fake reviews. At first, the work uses hints from behavioral data (timestamp, rating) and interpersonal data (network) to construct a suspicious reviewer graph. Then, it breaks the whole suspicious reviewer graph into k-clique clusters, and we consider such k-clique clusters as highly suspicious candidate group spammers. Finally, it ranks candidate groups by group based and individual-based spam indicators. Count Vectorizer and TF-IDF Vectorizer were used for results evaluation. TF-IDF Vectorizer gave the best results we have seen by providing us with the most reliable curve and greater AUC better than any other test cases. Random Forest gave us the best model with TF-IDF Vectorizer using k=7 and Unigram + Bigram + Trigram with Precision, Recall, and AUROC being 1, 1, and 0.74 respectively. **Acknowledgement**

We would like to thank Allah Almighty to help us through this project and thank all the people who helped us and guided us to complete this project. A special thanks to our final year project supervisor,

“*Dr. Hamid Turab Mirza*”

who has helped and contributed in motivating us and providing us the suggestions and proper encouragement to coordinate our project, especially in writing this report.

Moreover, I would also like to thank with my heart the important role of Mr. Imran Raza. He gave us the permission to use all the required resources such as server, lab and other necessary resources to complete the computation of dataset.  Last but not least, much acknowledgement goes to the head of the project, Mr. Naveed Hussain who has invested his full effort in guiding us in achieving the goal, without him this would not have been possible. I must appreciate the suggestions given by other teachers from our university that has improved our project, thanks to their comments and advices.

Table of Contents

[1 Introduction 13](#_Toc30434115)

[1.1 Main Goals and Objectives 16](#_Toc30434116)

[1.2 Problem Statement 17](#_Toc30434117)

[1.3 Assumptions & Constraints 17](#_Toc30434118)

[1.3.1 Constraints 17](#_Toc30434119)

[1.3.2 Assumptions 17](#_Toc30434120)

[1.4 Project Scope 18](#_Toc30434121)

[1.5 Chapter Summary 19](#_Toc30434122)

[2 Requirements Analysis 20](#_Toc30434123)

[2.1 Literature Review / Existing System Study 20](#_Toc30434124)

[2.2 Stakeholders List (Actors) 20](#_Toc30434125)

[2.3 Requirements Elicitation 20](#_Toc30434126)

[2.3.1 Functional Requirements 20](#_Toc30434127)

[2.3.2 Non-Functional Requirements 21](#_Toc30434128)

[2.3.3 Requirements Traceability Matric 21](#_Toc30434129)

[2.4 Use Case Descriptions 21](#_Toc30434130)

[2.5 Software Development Life Cycle Model 22](#_Toc30434131)

[2.5.1 Requirements 22](#_Toc30434132)

[2.5.2 Planning 23](#_Toc30434133)

[2.5.3 Develop 23](#_Toc30434134)

[2.5.4 Release 24](#_Toc30434135)

[2.5.5 Track & Monitor 24](#_Toc30434136)

[2.6 Summary 24](#_Toc30434137)

[3 Dataset 25](#_Toc30434138)

[3.1 Dataset YELP 25](#_Toc30434139)

[3.2 Dataset Amazon 26](#_Toc30434140)

[4 System Design 27](#_Toc30434141)

[4.1 Work Breakdown Structure (WBS) 27](#_Toc30434142)

[4.2 Activity Diagram 28](#_Toc30434143)

[4.3 Sequence Diagram 29](#_Toc30434144)

[4.4 System Architecture 30](#_Toc30434145)

[4.5 Class Diagram 31](#_Toc30434146)

[4.6 Database Diagram 32](#_Toc30434147)

[4.7 Network Diagram 33](#_Toc30434148)

[4.8 Collaboration Diagram 33](#_Toc30434149)

[4.9 Proposed Method 34](#_Toc30434150)

[4.9.1 Algorithm: 34](#_Toc30434151)

[4.9.2 Constructing Reviewer Graph 35](#_Toc30434152)

[4.9.3 Collecting metadata 36](#_Toc30434153)

[4.9.4 Constructing Suspicious Reviewer Graph 36](#_Toc30434154)

[4.9.5 Finding k-groups 36](#_Toc30434155)

[4.9.6 Construct group-group overlap matrix 37](#_Toc30434156)

[4.9.7 Construct Threshold matrix 37](#_Toc30434157)

[4.9.8 Finding k-clusters 38](#_Toc30434158)

[4.9.9 Calculate Behavioural Features 38](#_Toc30434159)

[4.9.10 Rank groups 39](#_Toc30434160)

[4.10 Front End 39](#_Toc30434161)

[4.10.1 First Look and Features: 40](#_Toc30434162)

[4.11 Linguistic Feature Analysis 46](#_Toc30434165)

[4.11.1 Data Pre-Processing 47](#_Toc30434166)

[4.11.2 Imbalanced Datasets 47](#_Toc30434167)

[4.11.3 Handling Imbalanced Datasets 47](#_Toc30434168)

[4.11.4 Bivariate Analysis 48](#_Toc30434169)

[4.11.5 Classification Algorithms 51](#_Toc30434170)

[4.12 Behavioural Feature Analysis 54](#_Toc30434171)

[4.12.1 Average Rating Deviation 54](#_Toc30434172)

[4.12.2 Penalty L(g) 54](#_Toc30434173)

[4.12.3 Group Deviation 54](#_Toc30434174)

[4.12.4 Review Burst Ratio 55](#_Toc30434175)

[4.12.5 Group Size 55](#_Toc30434176)

[4.12.6 Group Size Ratio 55](#_Toc30434177)

[4.12.7 Review Tightness 56](#_Toc30434178)

[4.12.8 Group Support Count 56](#_Toc30434179)

[4.12.9 Group Content Similarity 56](#_Toc30434181)

[5 System Testing and Experimental Results 58](#_Toc30434182)

[5.1 Testing of Linguistic Model 58](#_Toc30434183)

[5.1.1 Performance Metrices 58](#_Toc30434184)

[ Confusion matrix 58](#_Toc30434185)

[ Precision 58](#_Toc30434186)

[ Recall 59](#_Toc30434187)

[ F-Measure 59](#_Toc30434188)

[ Accuracy 59](#_Toc30434189)

[ AUROC 59](#_Toc30434190)

[ Brier Score (Mean Square Error) 60](#_Toc30434191)

[5.1.2 Graphs 66](#_Toc30434195)

[5.1.3 Conclusion 95](#_Toc30434198)

[5.2 Results of Behavioural Model 96](#_Toc30434199)

[6 Conclusion 104](#_Toc30434200)

[6.1 Problems Faced 104](#_Toc30434201)

[6.2 Project Summary 105](#_Toc30434202)

[6.3 Future Work 105](#_Toc30434203)

[7 References 106](#_Toc30434204)

**List of Tables**

[Table 1 - Fake Reviewer Groups in Consumer Reviews [8] 16](#_Toc28058872)

[Table 2 - Requirements Traceability Matric 23](#_Toc28058873)

[Table 3- Three YELP datasets Table 27](#_Toc28058874)

[Table 4 - Dataset Amazon Statistics 28](#_Toc28058875)

[Table 5 - Key for Behavioural Features Formulae 56](#_Toc28058876)

[Table 6 - Dataset Statistics used for Testing 59](#_Toc28058877)

[Table 7 - Confusion Matrix 59](#_Toc28058878)

[Table 8 - Performance Measures using Count Vectorizer and k=3 and a=1 63](#_Toc28058879)

[Table 9 - Performance Measures using Count Vectorizer and k=7 and a=10 64](#_Toc28058880)

[Table 10 - Performance Measures using TF-IDF Vectorizer and k=3 and a=10 66](#_Toc28058881)

[Table 11 - Performance Measures using TF-IDF Vectorizer and k=7 and a =10 67](#_Toc28058882)

**List of Figures**

[Figure 1 - Sample Review 17](#_Toc30434205)

[Figure 2 – Use Case Design 20](#_Toc30434206)

[Figure 3 - SDLC Agile method 21](#_Toc30434207)

[Figure 4 – Amazon Dataset Summary 25](#_Toc30434208)

[Figure 5 - Amazon Dataset Pie Chart 25](file:///C:\Users\qasim\Desktop\New%20folder%20(3)\FYP-II%20Report.docx#_Toc30434209)

[Figure 6 – Work Breakdown Structure (WBS) 26](#_Toc30434210)

[Figure 7 - Activity Diagram 27](#_Toc30434211)

[Figure 8 - Sequence Diagram 28](#_Toc30434212)

[Figure 9 - System Architecture 29](#_Toc30434213)

[Figure 10 – Class Diagram 30](#_Toc30434214)

[Figure 11 – Database Diagram 31](#_Toc30434215)

[Figure 12 – Network Diagram 32](file:///C:\Users\qasim\Desktop\New%20folder%20(3)\FYP-II%20Report.docx#_Toc30434216)

[Figure 13 – Collaboration Diagram 33](#_Toc30434217)

[Figure 14 - Example of k clusters. 34](#_Toc30434218)

[Figure 15 – Finding k-groups 36](#_Toc30434219)

[Figure 16 – Group-Group Overlap Matrix 36](#_Toc30434220)

[Figure 17 – Threshold Matrix 37](#_Toc30434221)

[Figure 18 – Finding k-clusters 37](#_Toc30434222)

[Figure 19 – Feature Learning of groups 38](#_Toc30434223)

[Figure 20 – Rank Groups 38](#_Toc30434224)

[Figure 21 – Homepage 1 39](#_Toc30434225)

[Figure 22 – Homepage 2 40](#_Toc30434226)

[Figure 23 – Footer on every page 40](#_Toc30434227)

[Figure 24 – Store 41](#_Toc30434228)

[Figure 25 – Product Details 41](#_Toc30434229)

[Figure 26 – Similar Products 42](#_Toc30434230)

[Figure 27 – Review Section 42](#_Toc30434231)

[Figure 28 – Review Results 43](#_Toc30434232)

[Figure 29 – Register 43](#_Toc30434233)

[Figure 30 – Login 44](#_Toc30434234)

[Figure 31 – About Us 44](#_Toc30434235)

[Figure 32 – Contact Us 1 45](#_Toc30434236)

[Figure 33 – Contact Us 2 45](#_Toc30434237)

[Figure 34 – Imbalanced Dataset 46](#_Toc30434238)

[Figure 35 – Handling Imbalanced Dataset 47](#_Toc30434239)

[Figure 36 – Bivariate Analysis 47](#_Toc30434240)

[Figure 37 – Feature Selection 48](#_Toc30434241)

[Figure 38 - Accuracy Chart with Count Vectorizer and k=3 and a =10 66](#_Toc30434242)

[Figure 39- Accuracy Chart for Count Vectorizer with k=7 and a=10 66](#_Toc30434243)

[Figure 40- AUROC Chart with Count Vectorizer and k=3 and a =10 67](#_Toc30434244)

[Figure 41 - AUROC Chart with Count Vectorizer and k=7 and a=10 67](#_Toc30434245)

[Figure 42 - Brier Loss Chart for Count Vectorizer and k=3 and a=10 68](#_Toc30434246)

[Figure 43 - Brier Loss Chart for Count Vectorizer with k=7 and a=10 68](#_Toc30434247)

[Figure 44 - Comparison of Calibration of Classifiers using Count Vectorizer, with k = 3 and a=10 and Uni-gram 69](#_Toc30434248)

[Figure 45 - Probability Calibration Curve of Naïve Bayes & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni 69](#_Toc30434249)

[Figure 46 - Probability Calibration Curve of Random Forest & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni 70](#_Toc30434250)

[Figure 47 - Probability Calibration Curve of XGBoost & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni 70](#_Toc30434251)

[Figure 48 - Probability Calibration Curve of ANN & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni 71](#_Toc30434252)

[Figure 49 - Probability Calibration Curve of LSTM & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni 71](#_Toc30434253)

[Figure 50 - Probability Calibration Curve of KNN & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni 72](#_Toc30434254)

[Figure 51 - Comparison of Calibration of Classifiers using Count Vectorizer, with k = 3 and a=10 and Uni-gram + Bi gram 73](#_Toc30434255)

[Figure 52 - Comparison of Calibration of Classifiers using Vectorizer, k=3 and a =10 and Uni + Bi +Tri 74](#_Toc30434256)

[Figure 53 - ROC curve with AUC Score k=3 and a=10 and Unigram 74](#_Toc30434257)

[Figure 54 - ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram 75](#_Toc30434258)

[Figure 55 - ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram + Trigram 75](#_Toc30434259)

[Figure 56 - Comparison of Calibration of Classifiers using Count Vectorizer, with k=7 and a=10 and Unigram 76](#_Toc30434260)

[Figure 57 - Comparison of Calibration of Classifiers using Count Vectorizer, with k=7 and a=10 and Uni + Bigram 77](#_Toc30434261)

[Figure 58 - Comparison of Calibration of Classifiers using Count Vectorizer, with k=7 and a=10 and Uni + Bi + Tri 78](#_Toc30434262)

[Figure 59 - ROC curve with AUC Score, with k=7 and a=10 and Unigram 78](#_Toc30434263)

[Figure 60 - ROC curve with AUC Score, with k=7 and a=10 and Uni + Bigram 79](#_Toc30434264)

[Figure 61 - ROC curve with AUC Score, with k=7 and a=10 and Uni + Bigram + Trigram 79](#_Toc30434265)

[Figure 62- Accuracy Chart for TF-IDF Vectorizer with k=3 and a=10 80](#_Toc30434266)

[Figure 63 - Accuracy Chart for TF-IDF Vectorizer with k=7 and a=10 80](#_Toc30434267)

[Figure 64 - AUROC Chart with TF-IDF Vectorizer with k=3 and a=10 81](#_Toc30434268)

[Figure 65 - AUROC Chart with TF-IDF Vectorizer with k=7 and a=10 81](#_Toc30434269)

[Figure 66 - Brier Loss Chart with TF-IDF Vectorizer with k=3 and a=10 82](#_Toc30434270)

[Figure 67 - Brier Loss Chart with TF-IDF Vectorizer with k=7 and a=10 82](#_Toc30434271)

[Figure 68 - ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=3 and a=10 and Uni-gram 83](#_Toc30434272)

[Figure 69 - Probability Calibration Curve of Naïve Bayes & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni 83](#_Toc30434273)

[Figure 70 - Probability Calibration Curve of Random Forest & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni 84](#_Toc30434274)

[Figure 71 - Probability Calibration Curve of XGBoost & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni 84](#_Toc30434275)

[Figure 72 - Probability Calibration Curve of ANN & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni 85](#_Toc30434276)

[Figure 73 - Probability Calibration Curve of LSTM & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni 85](#_Toc30434277)

[Figure 74 - Probability Calibration Curve of KNN & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni 86](#_Toc30434278)

[Figure 75 - ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k =3 and a=10 and Uni + Bi 87](#_Toc30434279)

[Figure 76 - ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=3 and a=10 and Uni + Bi + Tri 88](#_Toc30434280)

[Figure 77 - ROC curve using TF-IDF Vectorizer, with k=3 and a=10 and Uni 89](#_Toc30434281)

[Figure 78 - ROC curve using TF-IDF Vectorizer, with k=3 and a=10 and Uni + bigram 89](#_Toc30434282)

[Figure 79 - ROC curve using TF-IDF Vectorizer, with k=3 and a=10 and Uni+Bi+Tri Gram 90](#_Toc30434283)

[Figure 80 - ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=7 and a=10 and Unigram 90](#_Toc30434284)

[Figure 81 - ROC curve with AUC Score k=3 and a=10 and Unigram 91](#_Toc30434285)

[Figure 82 - ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram 91](#_Toc30434286)

[Figure 83 - ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram + Trigram 92](#_Toc30434287)

[Figure 84 - ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=7 and a=10 and Uni + Bi 92](#_Toc30434288)

[Figure 85 - ROC curve & Mean Predicted values using TF-IDF Vectorizer with k=7 and a=10 and Uni + Bi + Tri 93](#_Toc30434289)

[Figure 86 - ROC curve using TF-IDF Vectorizer, with k=7 and a=10 and Uni 93](#_Toc30434290)

[Figure 87 - ROC curve using TF-IDF Vectorizer, with k=7 and a=10 and Uni + Bi Gram 94](#_Toc30434291)

[Figure 88 - ROC curve using TF-IDF Vectorizer, with k=7 and a=10 and Uni+Bi+Tri Gram 94](#_Toc30434292)

[Figure 89 – Results of Behavioural Features with k=3 95](#_Toc30434293)

[Figure 90 – Results of Behavioural Features with k=7 95](#_Toc30434294)

[Figure 91 – PDF of Average Rating Deviation with k=3 96](#_Toc30434295)

[Figure 92 - PDF of Average Rating Deviation (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 96](#_Toc30434296)

[Figure 93 - PDF of Group Deviation (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3 97](#_Toc30434297)

[Figure 94 - PDF of Group Deviation (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 97](#_Toc30434298)

[Figure 95 - PDF of Review Burst Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3 98](#_Toc30434299)

[Figure 96 - PDF of Burst Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 98](#_Toc30434300)

[Figure 97 - PDF of Group Size (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3 99](#_Toc30434301)

[Figure 98 - PDF of Group Size (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 99](#_Toc30434302)

[Figure 99 - PDF of Group Size Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3 100](#_Toc30434303)

[Figure 100 - PDF of Group Size Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 100](#_Toc30434304)

[Figure 101 - PDF of Review Tightness (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3 101](#_Toc30434305)

[Figure 102 - PDF of Review Tightness (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 101](#_Toc30434306)

[Figure 103 - PDF of Group Support (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3 101](#_Toc30434307)

[Figure 104 - PDF of Group Support (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 102](#_Toc30434308)

[Figure 105 - PDF of Group Content Similarity (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3 102](#_Toc30434309)

[Figure 106 - PDF of Group Content Similarity (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7 102](#_Toc30434310)

# Introduction

People these days, buy products online rather than going out in a real market and wasting their time. As we all know a well-known store is often full of people and people put their trust in them. That’s all because the customers are satisfied and they recommend other people to visit the store. Similarly, if someone wants to buy a product online, they would like to read the reviews about that product before buying it. Many opinion-sharing websites are open just for the sole purpose of reviewing different products. Also, there is a proper channel provided by almost every business online to provide your feedback through reviewing system. So, if someone visits such sites and all they read is positive feedback about the desired product they will most probably buy it in no time. If the feedback is negative, they would definitely go out for other options. But what is the guarantee that whether those reviews are genuine? Or they are spams? We can’t say for sure until a spam detection system is provided to such opinion-sharing websites. Reviews or opinions have become an asset not only for the customers to purchase a better product but also for organizations to handle important decisions regarding the betterment of the business.

In order to increase the number of profits or get some fame or even demote a particular rival, unfortunately fraudulent reviews (Spam) mislead potential customers to reach a conclusion and buy the best product and organizations into recalculating different decisions to reach a conclusion. Similar works have been proposed to find such spam reviews and they concentrated on detecting spam reviews and discrete or individual spammer reviews. As the internet is accessible by almost every person today, group review spamming, which includes a group of swindlers working together who post fake online reviews for the purpose of promoting or demoting a product or products, has become more damaging, as they become capable of controlling the sentiment of a product.

Recent works like those in [1-7] to find fake reviews and spammers have made this a matter to be concerned about. People are getting paid to do so and it has become a business, it has become a major part of the decision making for both the customers and organizations, which has made the detection of such fake reviews a deliberate and a sensitive issue.

But when it comes to Group Spam, the amount of work done is not significant and only limited numbers of people have done some research on this. Previous works state detecting group spam using review content feature [4] or after indicating the strange behaviours of lone swindlers [5] because the large the number of the members of a group the more different styles and fewer chances of abnormalities. The more the size of a group the difficult it is to differentiate them as fake reviews and the more control they get over the product’s worth. [8] Keep that in mind, by group reviewers this work means a collection of profiles or IDs. The groups can contain Sock puppets (Lots of profiles of a single person), lots of people, or both. This work is not differentiating them, rather the system is mainly based on the detection of such spam groups.

Group spammers work together to promote or demote a product and it becomes very difficult to differentiate between them. Through the content of reviews, they might be detected if they are copying the reviews. If so then the prior techniques can be used. [1,3] Otherwise, it’s too difficult to do so. [4-7] Instead of waiting for mixed reviews, people nowadays buy a bunch of people, pay them a little amount of money and use them for positive feedback for promotion and negative feedback to their rivals’ products for the demotion of that product in the market.

The group spam and how it works is perfectly portrayed in the following example;

|  |  |  |
| --- | --- | --- |
| 1 of 1 people found this review helpful:  **Practically FREE music**, December 4, 2004  This review is from: **Audio Xtract (CD-ROM)**  I can’t believe for $10 (after rebate) I got a program that gets me free unlimited music. I was hoping it did half what was …. | 2 of 2 people found this review helpful:  **Like a tape recorder,** December 8, 2004  This review is from: **Audio Xtract (CD-ROM)**  This software really rocks. I can set the program to record music all day long and just let it go I come home and my …. | 3 of 3 people found this review helpful:  **Wow, internet music!**, December 4, 2004  This review is from: **Audio Xtract (CD-ROM)**  I looked forever for away to record intrnet music. My way took a long time and many steps (frustrating). Then I found this with more than 3000 songs downloaded in …. |
| 3 of 8 people found this review helpful:  **Yes - it really works**, December 4, 2004  This review is from: **Audio Xtract (CD-ROM)**  See my review for Audio Xtract – this PRO is even better. This is the solution I’ve been looking for. After buying iTunes …. | 3 of 10 people found this review helpful:  **This is even better than…** , December 8, 2004  This review is from: **Audio Xtract (CD-ROM)**  Let me tell you, this has to be one of the coolest products ever on the market. Record 8 internet radio stations at once …. | 2 of 9 people found this review helpful:  **Best music just get…,**, December 4, 2004  This review is from: **Audio Xtract (CD-ROM)**  The other day I upgraded to this TOP NOTCH product. Everyone who loves music needs to get it from internet …. |
| 5 of 5 people found this review helpful:  **My kids love it**, December 4, 2004  This review is from: **Pool Aquarium 3D Deluxe Edition**  This was a bargain at $20 - better than the other ones that have no above water scenes. My kids get a back out of the …. | 5 of 5 people found this review helpful:  **For the price you….** , December 8, 2004  This review is from: **Pool Aquarium 3D Deluxe Edition**  This is one of the coolest screensavers. I have ever seen, the fish move realistically, the environments look real, and the …. | 3 of 3 people found this review helpful:  **Cool, looks great**, December 4, 2004  This review is from: **Pool Aquarium 3D Deluxe Edition**  We have this set up on the PC at home and it looks GREAT. The fish and the scenes are really neat. Friends and family …. |

Table 1 - *Fake Reviewer Groups in Consumer Reviews [8]*

In Table 1, researchers found the following patterns that lead them to believe that this work is suspicious and has been done by a group.

1. Five-star rating for all three products through all three profiles.
2. The small-time window of four days i.e. Dec-8 -2014 when they posted the reviews.
3. All of them reviewed the same products.
4. They are the first to review the products as soon as they were launched.

If you read the reviews individually, they do not match with each other at all and standing alone they look like genuine reviews, and they have clearly taken control of the worth of the product through their positive feedback and anyone whose most likely to read these are going to buy the product. Therefore, group spam is very powerful when it comes to control on the sentiment of the product and this work dealt with and pointed out.

In the proposed project we worked on the detection of group spam reviews and tried to produce the most efficient system to come across fake review groups. The system uses a dataset extracting method to find a set of such groups. When collecting the data, we had to be very careful about the phenomenon i.e. the difference between group spam and a user who has similar product taste. For example, if a person plays Clash Royale and he loves it, he is likely to play similar games like Clash of Clans and rate them five stars as well. Hence it is not spam. After collecting the dataset and metadata out of it. The proposed work produces a suspicious reviewer graph on the basis of common products and co review similarity. After that the work produces a group to group overlap matrix and then a threshold matrix and after that it collects behavioural features of the candidate groups and then rank them groups in a decreasing way. The work then identifies a list of group spammers.

Although labelling group spammers as compared to individual spammers seemed very difficult but surprisingly labelling group spammers was much easier. The proposed techniques depart from traditional works and provides a new approach to detect spam review groups and make the classic approach less effective.

## Main Goals and Objectives

Propose a detection system that helps all the customers and businesses in getting rid of the fraudulent reviews and get the original feedback and opinions of genuine peoples.

The Main goals of our project are:

1. Develop a group spam review detection system which provides a medium for a customer to get the original opinion and the truth about a particular product.
2. Develop a group spam review detection system which provides guidance for businesses for the betterment of their products through genuine feedback.
3. Produce a product that has never been offered in the market.
4. Provide a Web and Android Portal
5. Develop the system with the latest technology and trends.

The Objectives of our project are:

1. Improve the overall Online Review System E-Commerce in Pakistan and all over the world.
2. Provide a bridge for the consumer and businesses to communicate with each other through genuine feedback or reviews.
3. Earn money by providing a new product for the market and selling it out to different customers.
4. Improve the review system all over the globe.

## Problem Statement

Buying and selling of goods online have become a revolutionary phenomenon in the IT world. Online reviews are considered the best source of customer’s opinion about a product, and an asset for customers, and organizations in the decision-making process. Woefully, in order to increase the number of profits, promotion or even to demote a rival product, deceptive reviews (Spam) mislead prospective customers to buy the best product and organizations in decision making. Works have been proposed on detecting individual spammer reviews.

But group review spamming, which includes a group of swindlers working together to post fake online reviews for promoting or demoting a product/s, has become more damaging. More the size of the group more difficult it is to differentiate them as fake reviews. In the proposed project we are going to prepare a system to detect group spam reviews. Getting to know all these facts we had to develop a system that deals with it.

Our aim is to make e-commerce websites reliable for costumers. A platform where people gets a truthful buying experience. To overcome this problem, we’d studied this domain and work on linguistic and behavioural approach to detect spam review. We have developed a system that deals with this problem of group spammers and it will not only benefit customers on e-commerce websites but also researchers who are working on this small yet broad topic.

## Assumptions & Constraints

### Constraints

* **Absence of labelled dataset:** For training of our model labelled dataset will be required but authenticity of labelled dataset might be inappropriate.
* **Limited competence and precision:** The efficiency and correctness of our model will be limited because of limited dataset.
* **Limited computation power:** As, we all know that computing big data required huge computation power and we as students were limited by this.

### Assumptions

* We assume that, the required review text contains alphanumeric characters.
* We will collect metadata from review set (how frequently that reviewer post reviews, rating deviation, timestamp, and their targeted products).
* We assume that all the reviews in English language.
* All reviews will follow the below structure

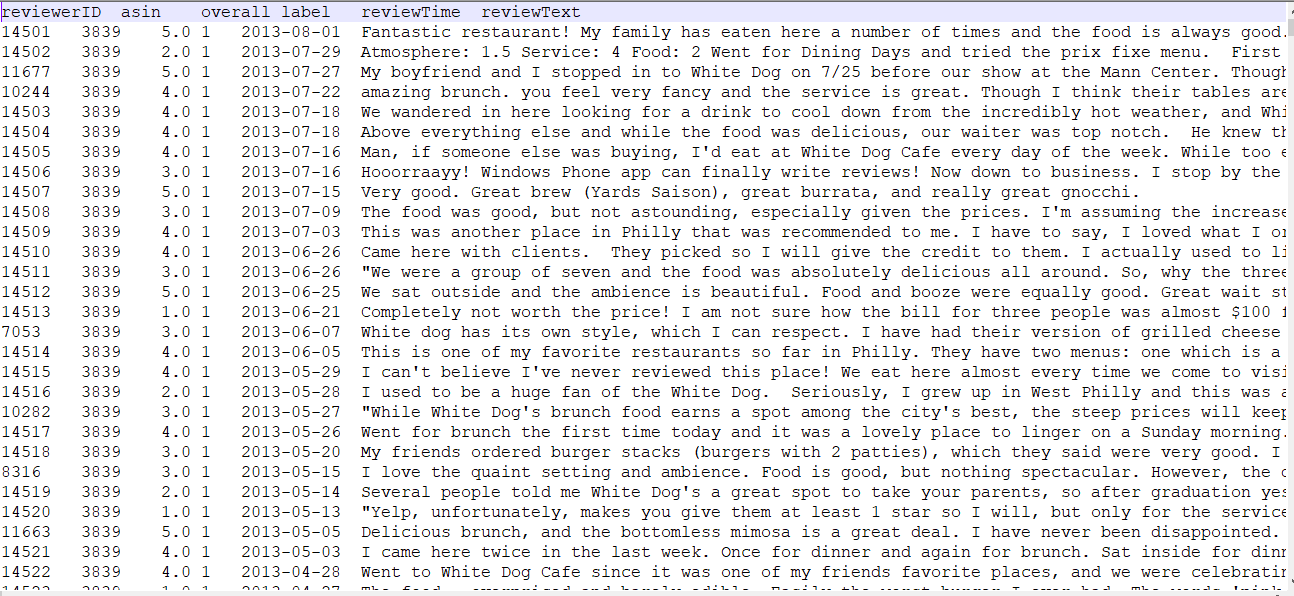


Figure 1 *- Sample Review*

Where

* **reviewerID** - ID of the reviewer
* **asin** - ID of the product
* **overall** - rating of the product
* **label** - label of the review (spam or not spam)
* **reviewTime** - time of the review (unixtime)
* **reviewText** - text of the review

## Project Scope

Technological change has been the significant main push for an increase in buying products online and improvement in the online shopping business. In emerging countries like Pakistan, people know a lot about the emerging IT sector and its prolific innovations and its use every day but are not taking advantage of that to increase their business’s profitability and make their work simpler.

Online Review Systems is considered the best possible way to buy a better product or get the proper feedback and group spammers take that away from both the customers and businesses. Work has been done but there is always space in the IT world. The main reason is this, that no system is available that is smart enough, to resolve problems of consumers and business holders on its own. In the age of artificial intelligence and machine learning the need for such systems is not just profitable but the need of the hour. Group Spammers are hard to find but our system will provide a platform where all fraudsters can be detected through different techniques and customers will be guaranteed a 100% genuine review system.

We believe businesses will use our system if they expect to get a boost in their business and promote their product by giving a feature of 100% original feedback through our system. Also, they can use it for the betterment of their product. No work has been done and we can use this as an advantage by producing a product that has never been offered and it is going to help us with money as well as getting good jobs.

## Chapter Summary

In this chapter, we have talked over the initial phase of our project. To conclude this the project is going to focus on:

1. Fetching datasets containing and produce a labelled dataset.
2. Crawl Review Dataset.
3. The data was pre-processed.
4. Suspicious reviewer graph.
5. K-cliques.
6. Matrices on the basis of k-cliques.
7. Calculated the normalized value of each attribute.
8. Checked whether it’s spam or not spam and let it work on each review.

# Requirements Analysis

## Literature Review / Existing System Study

In recent years there have been many people that have researched in this area. Most of the work done on this topic is quite respectable. Some of the work done in this field are stated as follows:

The most popular works done on group spammers was done by Mukherjee *et al.* [8] They used FIM (frequent pattern mining) to find candidate groups. Ahsan *et al.* [10] uses active learning to detect review spamming. It used the TFIDF feature of review content. Furthermore, Hai, Zhen *et al.* [11] prepared a multi-task learning method that in detail describes0 logistic regression (MTL-LR). To leverage unlabelled data, they used Laplacian regularizer then they introduced a semi-supervised multi-task learning method (SMTL-LLR) to further increase the performance of spam detection. Likewise, Adike, M. R., and Vivekanand Reddy [12] compellingly recognize untruthful reviews that are given by users that have semantic content based on sentiment analysis. They use J48 classifier and generate ARFF from distinct features. Zhuo Wang *et al*. [13] which is one of the most recent works on group spam review detection systems, proposed a system named “GSDLA” which claimed to work in two phases: Increasing the group size (GS) and Latent Dirichlet Allocation (LDA) which bounds the closely connected spammers into small clusters. Then their SCAN algorithm extracts highly sceptical spammer groups from these group spam behaviour features. They did experiments on three real-world datasets [13] and their work outperforms the many powerful standards. GSCPM a CPM based model to find group spammers Xu, G., Hu *et al*. [14] They detect spam problem as a network classification task Shebuti *et al*. [15]

## Stakeholders List (Actors)

The stakeholders list for this project is as follow:

* Researchers
* Developers
* Supervisors
* Ecommerce Industry
* E-store Owners
* Fake Reviewers / Spammers
* Project team

## Requirements Elicitation

Requirement description is the most significant step in a software engineering lifecycle. Inadequate requirements can be an important cause of product failure whereas clear and precise requirement gathering provide good results. In this phase, all the requirements both functional and non-functional will be collected and listed.

### Functional Requirements

* The system should receive a review as an input.
* The system should save the model in a pickel file.
* The system should generate output after proper work.

### Non-Functional Requirements

* The accuracy of the system must be high.
* System should be swift and user should not face time lag issues.
* The system should be protected and secure.
* The system must have the power to protect itself from outer attacks.

### Requirements Traceability Matric

|  |  |  |  |
| --- | --- | --- | --- |
| BR # | Module Name | Applicable Roles | Description |
| B1 | New Review Posted | Reviewer | System should be up and able to receive all reviews submitted on the front end for further processing. |
| B2 | Behavioural Model | System | The model should be able to receive all incoming reviews and return the probability of its being spam or not. |
| B3 | Output | System | The system should return the final value about whether the submitted review was spam or not |
| B4 | Prediction Posted on Front End | System | The system should return a detailed result of the output on the front end. |

*Table 2 - Requirements Traceability Matric*

## Use Case Descriptions

Once the review is posted by the reviewer, it will be processed by the model and analysed to make prediction about the review. The prediction at the back end will be based on the model using behavioural model for the prediction.

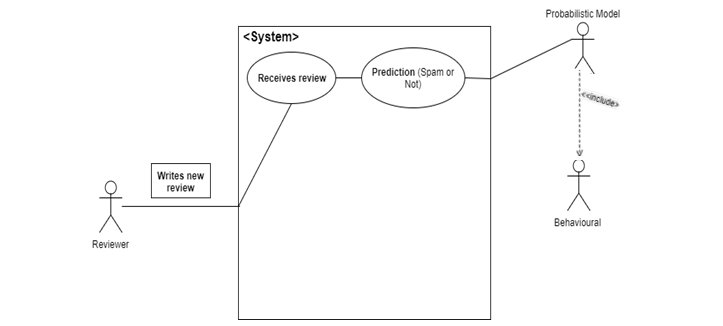


Figure 2 *– Use Case Design*

## Software Development Life Cycle Model

Software solution developing firms faces a lot of difficulties in choosing the appropriate software development life cycle (**SDLC**). [8] Agile development is the most effective SDLC not just because of its numerous benefits over other SE models, but because it will help us improve the project in future and update it whenever we want. This is because of rapidness of the change in the field of IT.

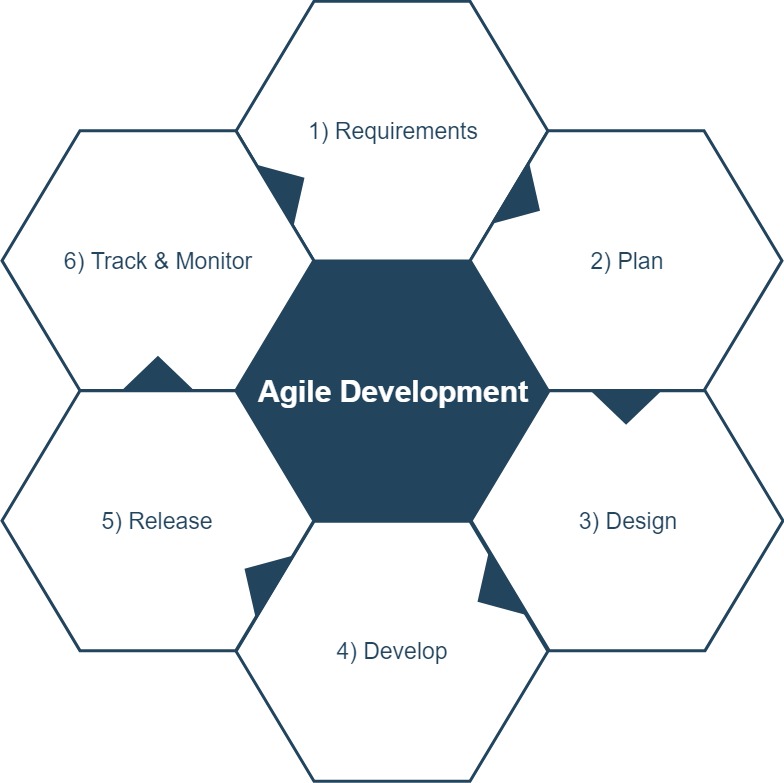


Figure 3 *- SDLC Agile method*

### Requirements

In this phase, all requirements will be gathered and listed. It is known as the most important phase in SDLC. In the proposed project, the initial requirement was to find large dataset of reviews which acquired from Amazon and Yelp. Some behavioural features were also gathered that will be applied on reviews and a probabilistic model to provide us with the desired results.

### Planning

In this phase we discuss different approaches and carry out a mechanism or plan to get the desired results. The proposed projects will use behavioural features to achieve this, each group will have to go through the undermine features. The study will cross check these behavioural features against the groups from the Spamicity that’ll will decide whether it’s a group spam or not.

### Develop

After the completion of all the initial phases this phase is to implement a prototype of the proposed system and we will be doing that using Python and different data sciences techniques.

### Release

After developing the prototype it’ll be exposed to testing and once the testing is done it will be released in the market.

### Track & Monitor

In this phase the tracking and monitoring of the system is done. The main purpose of the system is to learn with time and produce more accurate outputs. Hence after the release it will be monitored for any real time bugs, misclassifications or any other new change, so that a new and better version can be proposed.

## Summary

In this chapter, the detailed discussion about requirement analysis phase has been made. The chapter starts after the ending of chapter one at project’s scope, In the beginning of this chapter we discussed about the literature that have been reviewed for this project which includes analysing different algorithms and research techniques. We have done a detailed discussion on behavioural features and modelling of reviews based on these behaviours. The list of stakeholders was listed who might be interested in this research. Elicited requirements were also discussed in terms of functional and non-functional requirements. In the end, SDLC was proposed with detailed discussion on its each step-in relation to this project. Now, after completion of the requirement analysis this project will move to the next chapter i.e. System Design, where the design of our system will be discussed in detail.

# Dataset

Dataset selection is the major problem in Spam review detection. Amazon and Yelp are two big names in reviews. Amazon doesn’t give their reviews and Yelp gives a limited number of reviews. Some researches [16] The work uses the category of reviews on restaurants from the Yelp.com.

## Dataset YELP

In this work we use Yelp’s dataset. We had three datasets from Yelp.com to verify the performance of our proposed method. Yelp is considered as the biggest review website in the US and has widely used spam filtering algorithms. The three datasets were YelpChi, YelpNYC and YelpZip which are shown in table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **#Reviews (filtered %)** | **#Reviewers (spammer %)** | **#Product** | **Time span** |
| **YelpChi** | 67,395 (13.23 %) | 38,063 (20.33 %) | 201 | 2004. 10- 2012. 10 |
| **YelpNYC** | 359,052 (10.27 %) | 160,225 (17.79 %) | 923 | 2004. 10- 2015. 01 |
| **YelpZip** | 608,598 (13.22 %) | 260,277 (23.91 %) | 5044 | 2004. 10- 2015. 01 |

*Table 3- Three YELP datasets Table*

The work uses a sample of more than 52000 reviews from the YelpZip dataset. YelpZip is a dataset from the restaurants of different areas of US with continuous zip codes from all over the US.

## Dataset Amazon

Figure 4 illustrates the summary of dataset Amazon we used in the start of our project to understand and find out a way to deal the problem we made the basis of our project (group spam), whereas Table 4 explains the Statistics of the dataset and Figure 5 is a pie chart that describes the amount of products reviewed:

Figure 4 *– Amazon Dataset Summary*

|  |  |
| --- | --- |
| **Total Records** | 1524234 |
| **Number of Reviewers** | 1402714 |
| **Number of Products** | 826010 |
|  |  |
| **Category** | **Number of Reviews** |
| - Cell Phones and Accessories | 260551 |
| - Clothing, Shoes and Jewellery | 243929 |
| - Electronics | 246596 |
| - Home and Kitchen | 264241 |
| - Sports and Outdoor | 254282 |
| - Toys and Games | 254635 |

Figure 5 - Amazon Dataset Pie Chart

*Table 4 - Dataset Amazon Statistics*

# System Design

## Work Breakdown Structure (WBS)

After the completion of requirement analysis, the next step is to start designing the system. Diagrams are being used to explain the system, including WBS, Activity diagram, System Architecture, Network diagram.

Figure 6 elaborates work breakdown structure for this project, as this is a R&D project so the project will mainly breakdown into two major phases: The research phase, in which study all previous domains new proposed works current or ongoing works all domain specific knowledge conducted on this topic to use all that knowledge in this project, and the development phase in which the actual implementation and development of the proposed project was done.

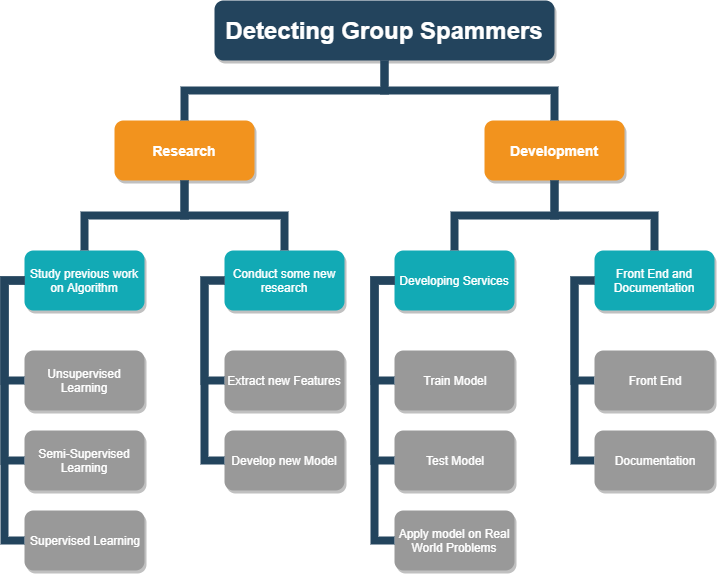


Figure 6 *– Work Breakdown Structure (WBS)*

## Activity Diagram

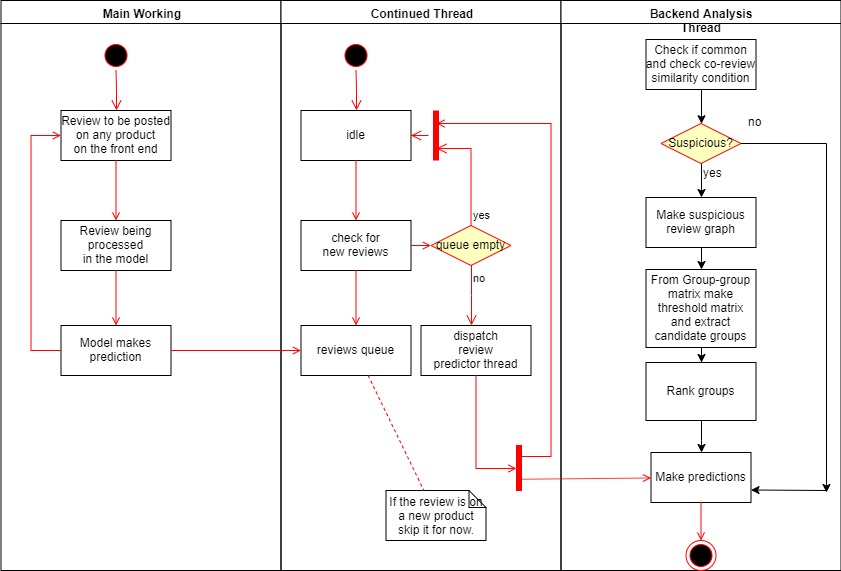


Figure 7 *- Activity Diagram*

## Sequence Diagram

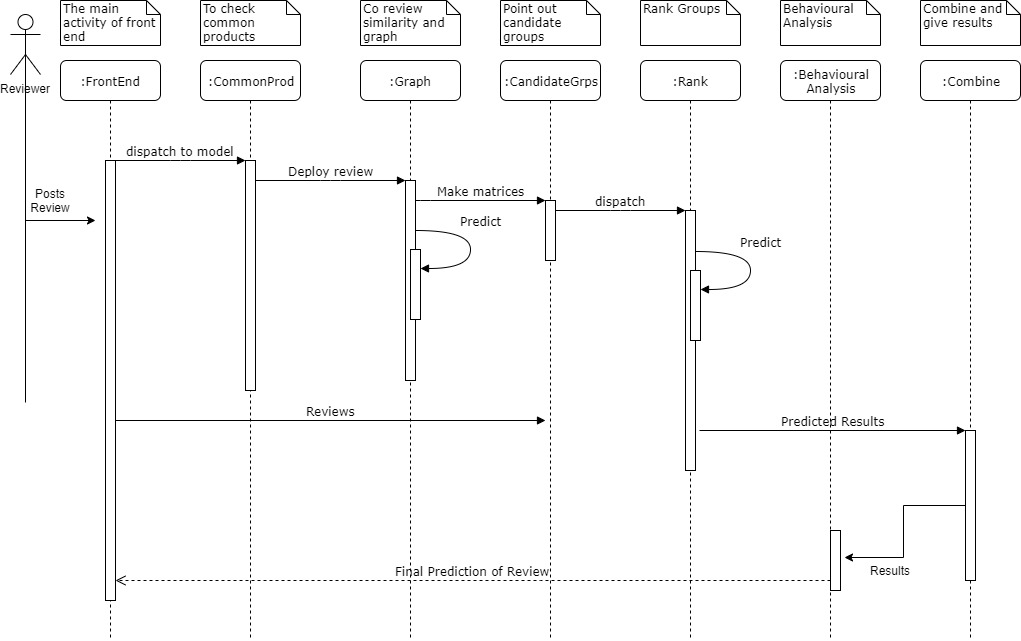


Figure 8 - Sequence Diagram

## System Architecture

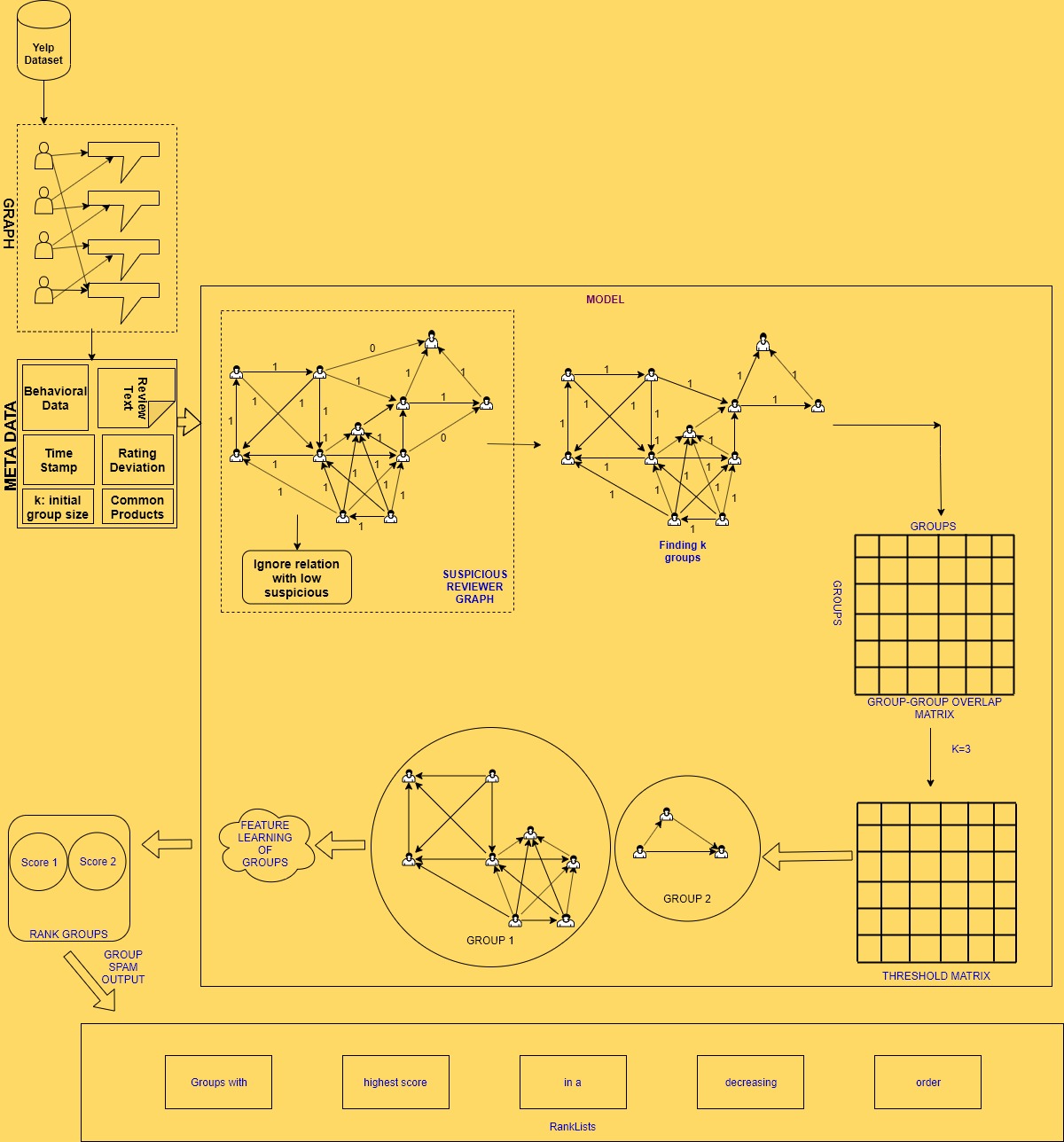


Figure 9 *- System Architecture*

## Class Diagram

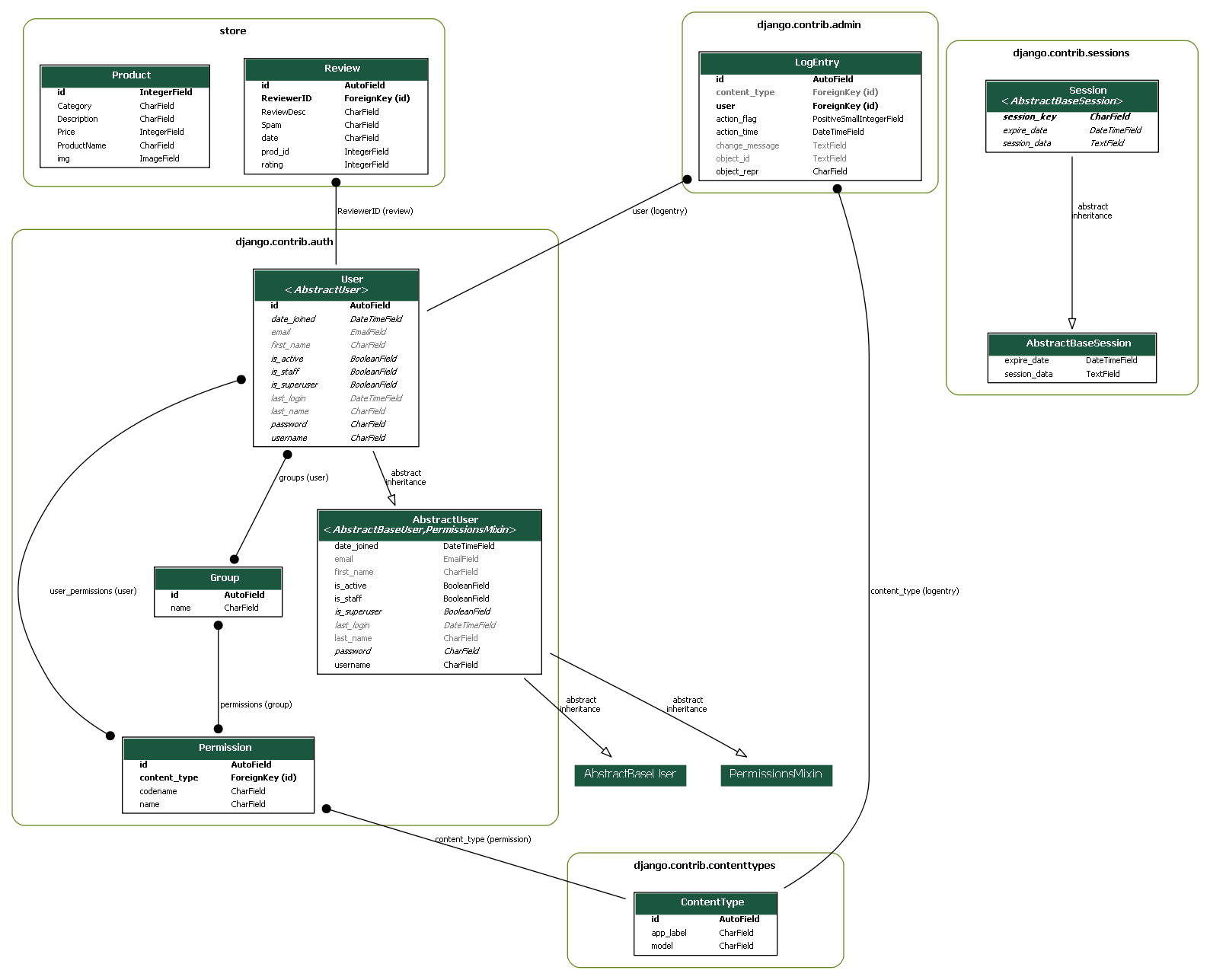


Figure 10 – Class Diagram

## Database Diagram

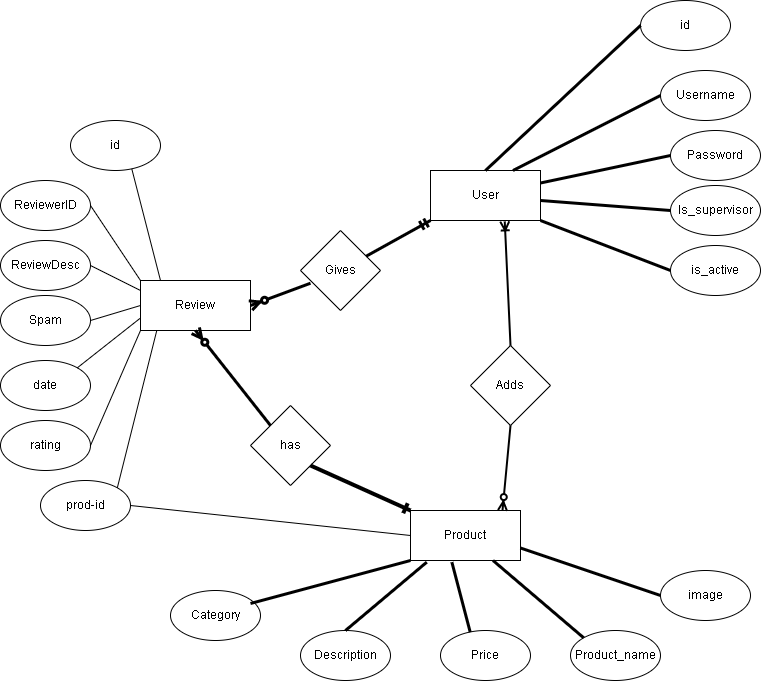


Figure 11 – Database Diagram

## Network Diagram

Figure 12 – Network Diagram

## Collaboration Diagram

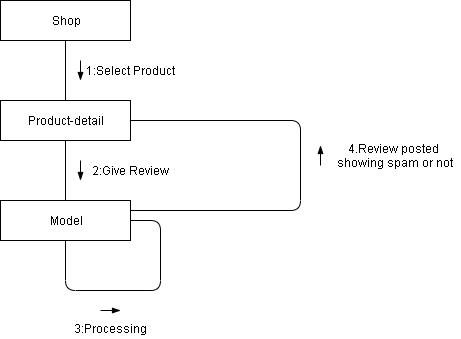


Figure 13 – Collaboration Diagram

## Proposed Method

The method has a functionality of finding group spammers in a completely unsupervised way. It can find all k-groups in a graph. A k-group or a clique is a complete graph with k-nodes.[5] K-group comprises of adjacent k-cliques, two groups are adjacent if and only if they have one less common node than ‘k’ i.e. (k-1 common nodes). The collaboration results in closeness of group spammers than the normal reviewers. The input to the algorithm is a newly arrived review from the front end which gives the output of whether the review is spam and of some group:

### Algorithm:

0: k = value \\ any value of group size

1: R = review // New review

2: D = dataset.load() // Load trained dataset

3: for i = dataset.length()

4: if (R.Product ∩ dataset.reviewer.product() && R.RS() == 1) // Reviewers Product matches other reviewers and co-review similarity is equal to 1

5: s = 1; // Suspicious

6: Construct k-clique

7: Construct Group-Group Overlap Matrix

8: if (matrix value < k-1)

9: Threshold Matrix value = 1

10: otherwise 0

11: Clique made when off diagonal element is 1

12: Calculate value of behavioural features

13: Suspicious Score = Sum.allBehaviouralFeatures()/8 // 8 Behavioural features

14: Higher the score more suspicious, hence group spam

15: Give group’s text as input to the best classifier // in our case RF Classifier

16: Extract Linguistic Features

17: Output = Features

18: From TF-IDF factor determine spam or not spam

19: For loop end

20: Otherwise Not Spam (0) // For loop breaks and system ends

In this method, following steps are followed:

**Step 1:** Construct a reviewer graph.

**Step 2:** Collect Metadata from reviewer graph and give value of k.



Figure 14- Example of k clusters.

**Step 3:** Construct Suspicious Reviewer Graph.

**Step 4:** Finding k-groups.

**Step 5:** Construct Group-Group overlap matrix.

**Step 6:** Construct Threshold matrix from Group-Group overlap matrix.

**Step 7:** Find k-clusters.

**Step 8:** Calculate Behavioural features.

**Step 9:** Ranking the candidate groups.

### Constructing Reviewer Graph

Construct a reviewer graph from the dataset. Assuming G = (V,E) where V are vertices and E are edges, and there is an edge between two reviewers if both of them have at least reviewed a one common product.

### Collecting metadata

Collect metadata from the reviewer graph timestamps, common products, behavioural data, review text, rating deviation and most important give the value of k.

### Constructing Suspicious Reviewer Graph

Once the reviewer graph is made add weights to the edges on the basis of the following formula:

*Equation 1 [14]*

Where Pi is the product reviewed by reviewer i and Pj is the product reviewed by reviewer j. RS(i,j,p) is the co review similarity. It will be 0 if and only if the co-review similarity is 0 and there is atleast one common product otherwise 1, and 1 represents highly suspicious and 0 represents not suspicious.

The formula for co-review similarity is :

Equation 2 [14]

Where tpi is the review time on product p by reviewer i and tpj is the review time on product p by reviewer j. α is a specified user time threshold. rpi is the rating of reviewer i on product p and rpj is the rating of reviewer j on product p.

The reviewer is considered highly suspicious if they meet both conditions. α is an important component and will have a direct impact on precision of the model.

### Finding k-groups

After ignoring and filter out the reviews with low suspiciousness. Find all k-groups in the suspicious reviewer graph.

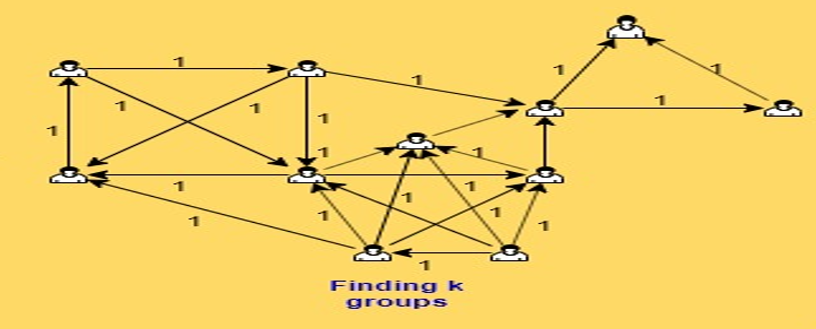


Figure 15 – Finding k-groups

### Construct group-group overlap matrix

Construct a group-group overlap matrix. It is a symmetric matrix and each row and column represents a group and the elements are the number of common nodes between the corresponding two groups and diagonal entries are equal to the size of the group. [14]

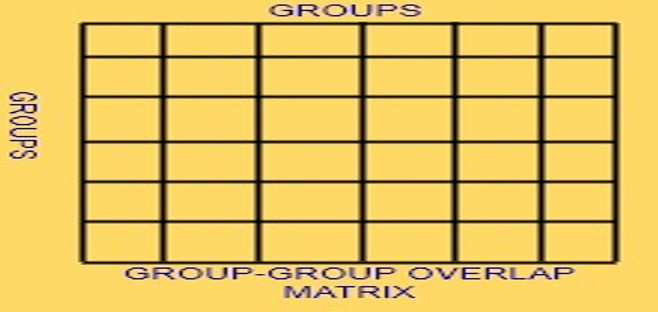


Figure 16– Group-Group Overlap Matrix

### Construct Threshold matrix

All the off-diagonal elements and the diagonal elements that are less than k-1 are set to 0 and others are set to 1. K is given to the model before constructing suspicious reviewer graph.



Figure 17 – Threshold Matrix

### Finding k-clusters

The work assumes that there exists a group when the corresponding diagonal element entry is 1, and two groups are adjacent when corresponding off diagonal element is 1.

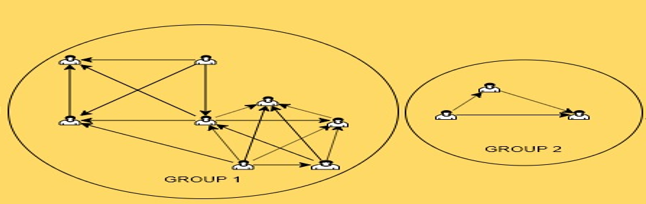


Figure 18– Finding k-clusters

### Calculate Behavioural Features

Now the work gives groups so calculate the values of behavioural features discussed in section 4.7. Then the average of all the features is calculated.



Figure 19 – Feature Learning of groups

### Rank groups

Now the work ranks the groups on the basis of the average of the values of behavioural features. The higher the score the more suspicious the group is.



Figure 20– Rank Groups

## Front End

Deciding which tool to use for the front-end was somewhat a challenging task for us. Challenging in a sense because we spent a lot of time deciding between Django and Django Oscar. First, we were opting for Django Oscar because it had everything built in. One would just have to use some code and a whole e-commerce site would be ready and waiting for you to use but due to we used code to install Django Oscar in a project, it had no local .html files. We spent a lot of time figuring out to find out where we could find it but then we decided to just use Django and started developing front-end on it. With relentless effort we were able to build up the structure of our front-end and then different features were added later on as decided. Following steps were followed to make the front-end. First of all, python must be installed on the system for all of this to work.

Starting off, a virtual-env-wrapper was installed, well as the name suggests to create a virtual environment so that we don’t mess up with other different configuration of Django for different projects. Thereon, Django was installed in that virtual environment wrapper. After navigating to the desired directory, we deployed our project and started working on it. At this point we needed an IDE to work on our project as guided by [17] Telusko. Any other IDE such as Notepad++ or sublime could have been used but we went for Visual Studio and it was quite fun to have used this tool. Having everything setup we just had to start the app and change the settings so that Django would know where all the html files would be. We named the folder templates and gave its path to Django DIRS in settings.py file.

For database we chose PostgreSQL from various options available online. Thereafter, to connect database with Django psycopg2 was used. Knowing Django has amazing feature of ORM (Object-Relational Mapper) which creates tables for you by just making a class, we used it (ORM) to make tables in database by just making classes. Some changes were made in settings.py file so the Django would know which database to connect to. After creating classes, we migrated the classes to tables in database. Hereafter, we began working on the actual front-end after having everything set up.

### First Look and Features:

In this section, front-end is shown with its features explained. Figure 21 and Figure 22 shows the homepage, we went for pretty standard look and feel of an e-commerce site with a “join us” button which takes you to the registration page.



Figure 21 – Homepage 1

Some of the categories are shown which user could click and on and it would take them to shop page with products of specific category shown.

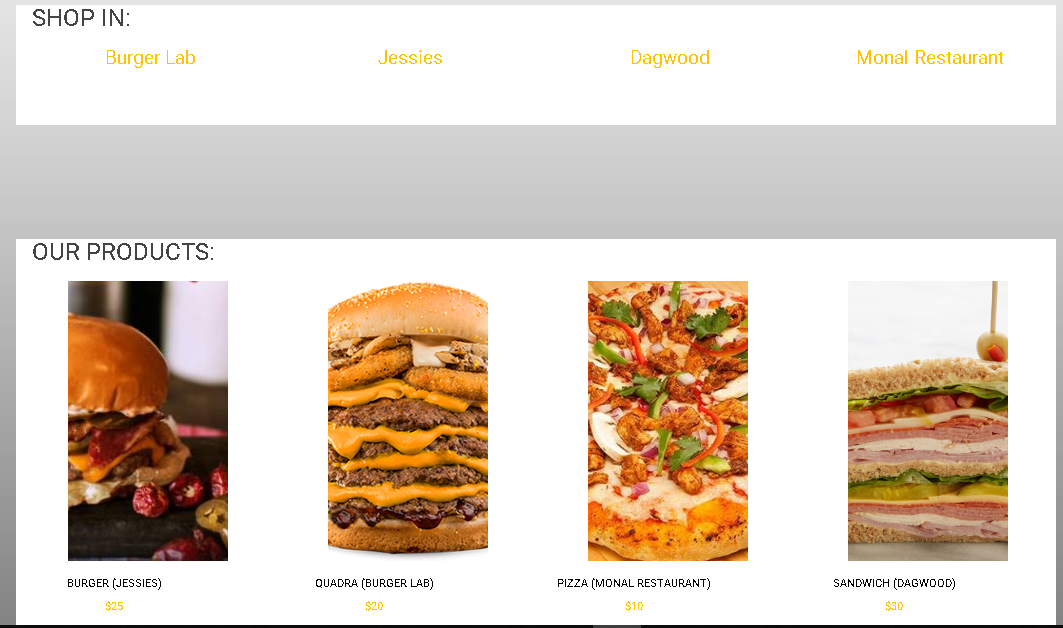


Figure 22 – Homepage 2

Figure 23 shows a basic footer which we used for every page.

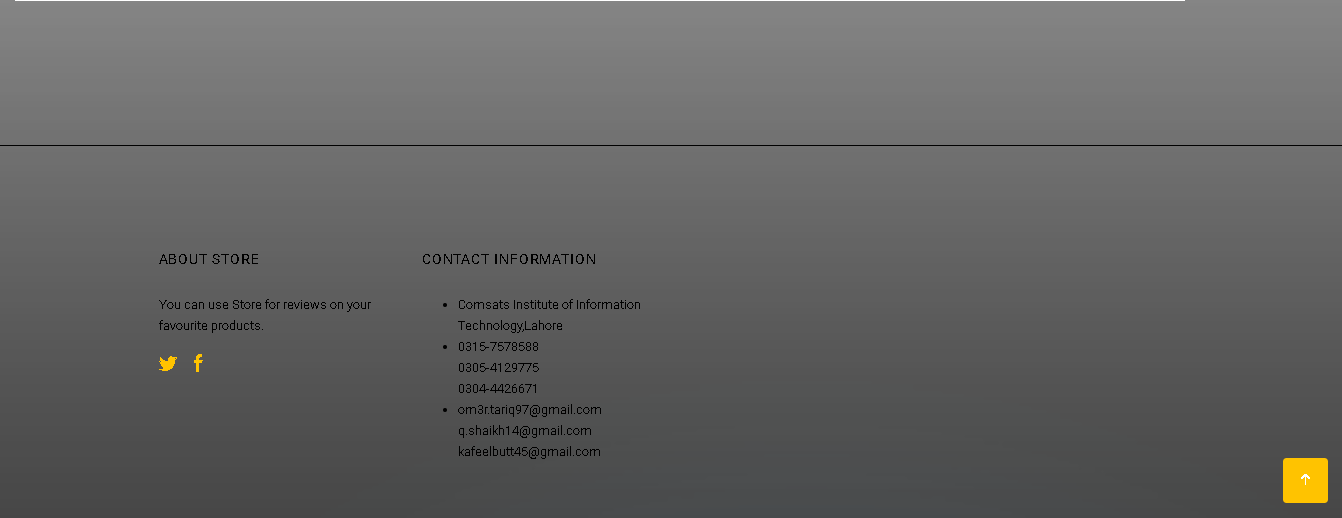


Figure 23– Footer on every page

This is the shop page which has all the products. You could select certain category and price range to have only those products shown which meet the criteria.

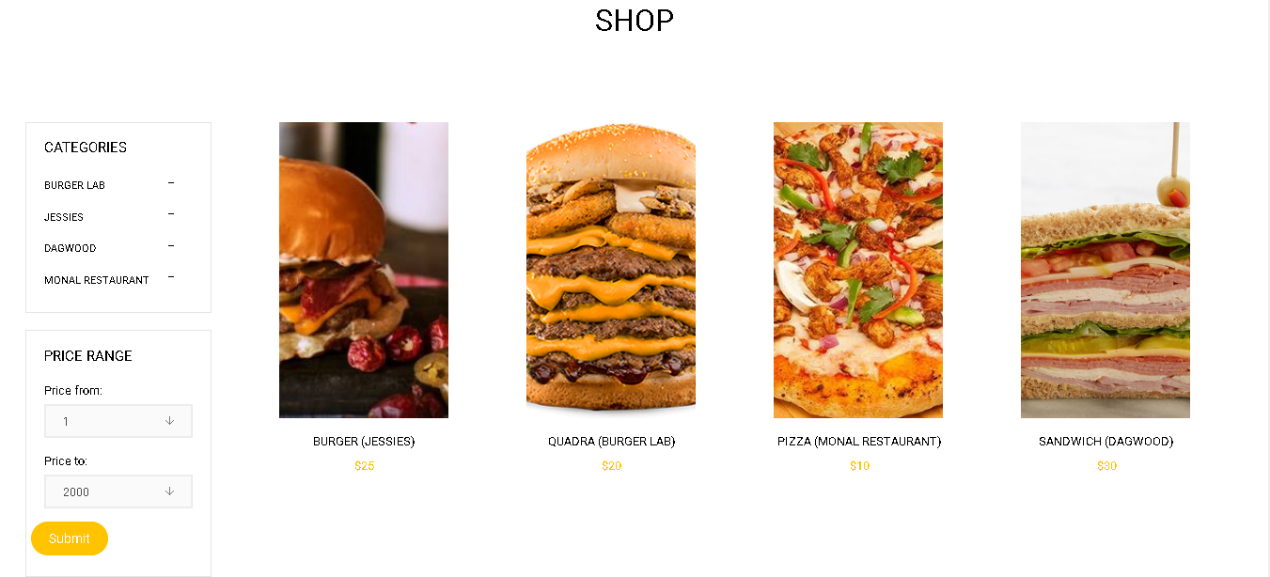


Figure 24 – Store

After selecting a product from shop page, you would be redirected to this product-detail page which shows you some of its details.

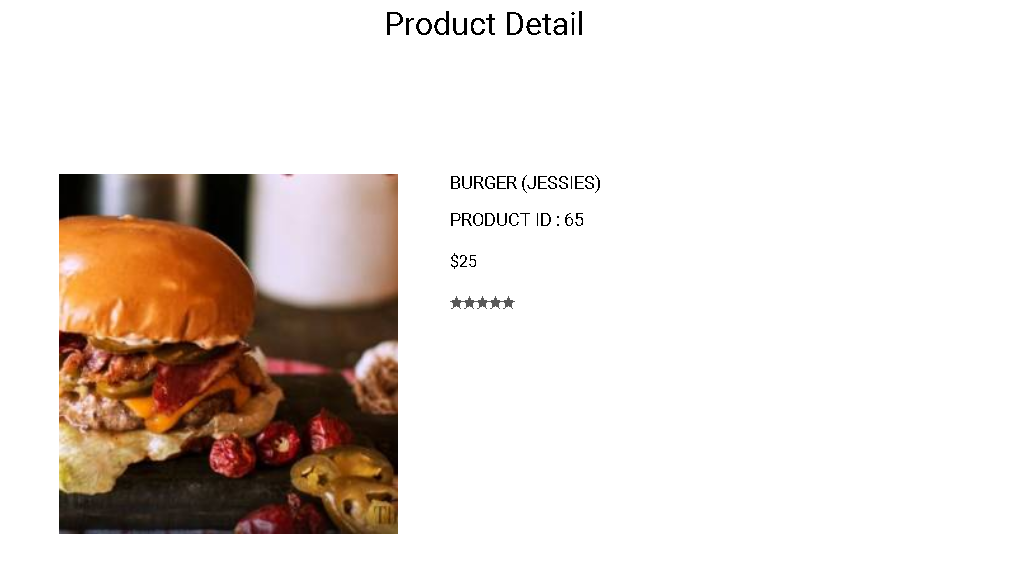


Figure 25 – Product Details

Description of the product is given in detail and similar products are shown if any.

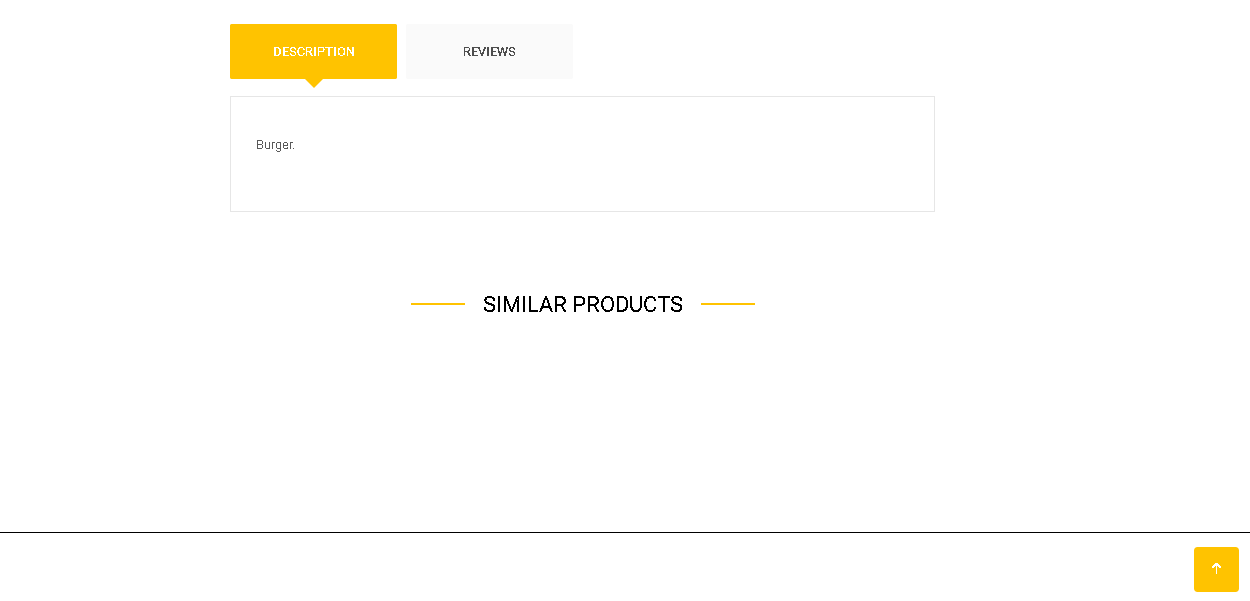


Figure 26– Similar Products

If the user is not logged in, he will be shown: “Give a Review” text which redirects him to login to the site to give review. The reviews are shown to the viewer.

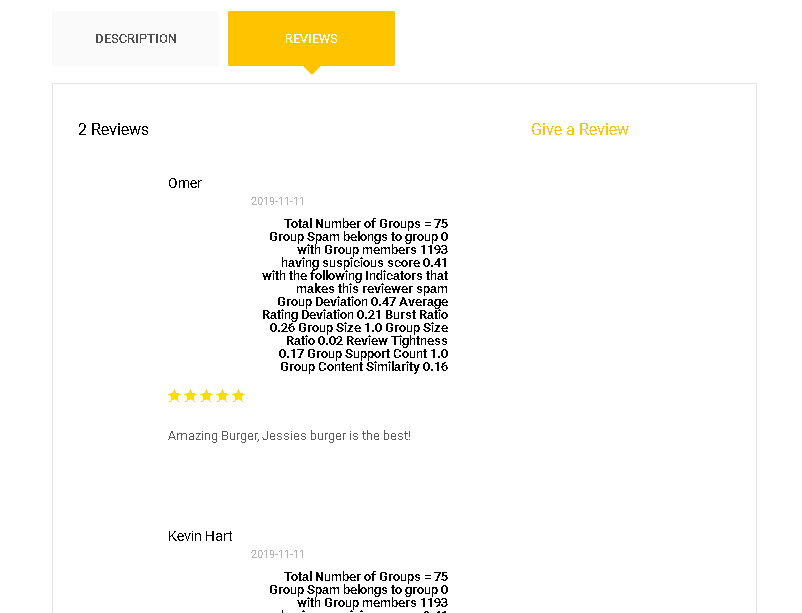


Figure 27– Review Section

After logging in, the user can give the review.

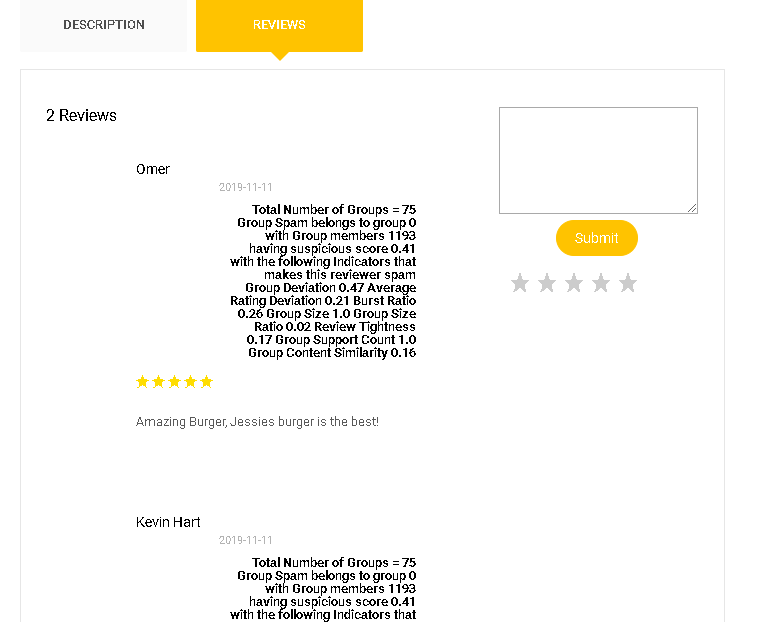


Figure 28 – Review Results

Figure 29 shows the Register page and Figure 30 shows the Login page, the user could login and would be redirected to homepage.

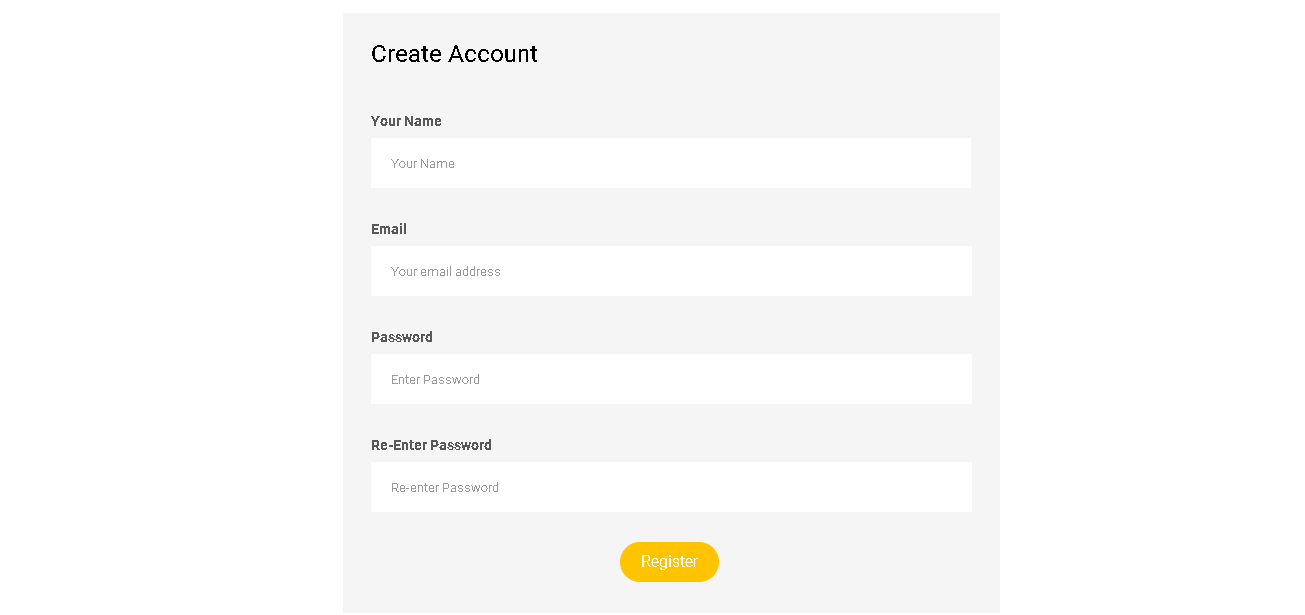


Figure 29 – Register

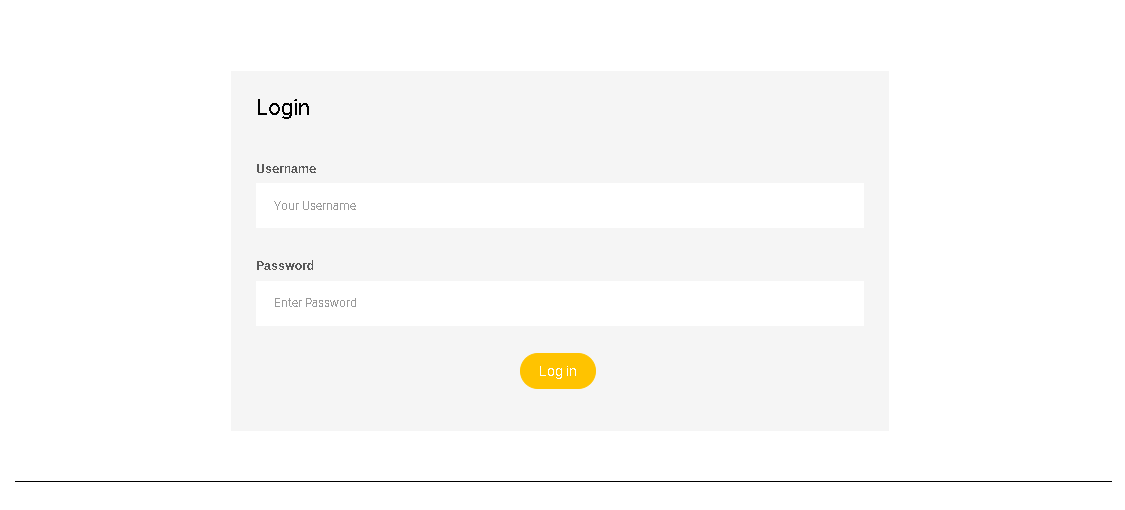


Figure 30– Login

This page gives a brief introduction about us.

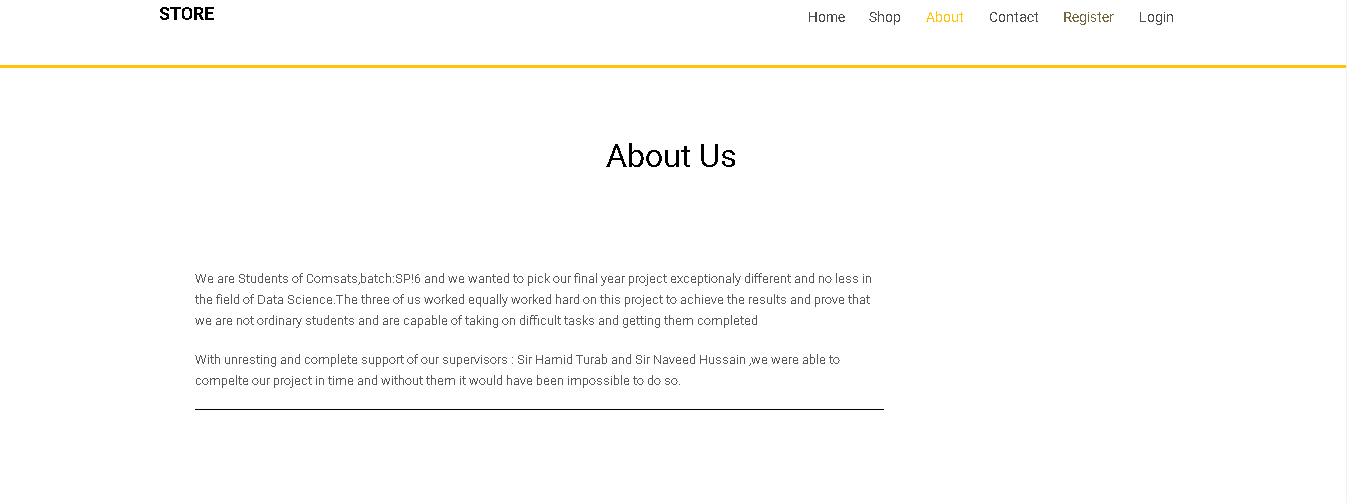


Figure 31– About Us

Figure 31 and 32 shows Contact information about the developers.

### Contact1

Figure 32– Contact Us 1

### Contact2

Figure 33 – Contact Us 2

## Linguistic Feature Analysis

This phase is used to identify spam reviews by considering the text data of reviews. It is used widely in detecting spam review. Most spammers use similar content in their reviews. Linguistic approach benefits from this to detect spammers from genuine reviewers.

A trained and tested Machine Learning model is accountable for this task. When a new review is passed to trained model, it refers whether it lies in group spam or not. It has two main steps

* Data Pre-Processing
* Classification Algorithm

### Data Pre-Processing

Machine learning model only works on numeric data which should also be continuous and discrete. Hence, it cannot be applied on text data. For this the conversion of text into numeric data is required. For this we have to pre-process our data and create a document term matrix. This project has also used pre-processing and the following techniques were followed:

### Imbalanced Datasets

The work has selected 50000 reviews sample from a dataset of millions and it has a problem called imbalanced dataset. This problem is due to number of imbalanced ratio of class variables as you see in this figure we have 25964 records of spam and 4036 records of not spam. This problem leads the classifier to biasness problem which will result in the dominant classifier as the resultant.

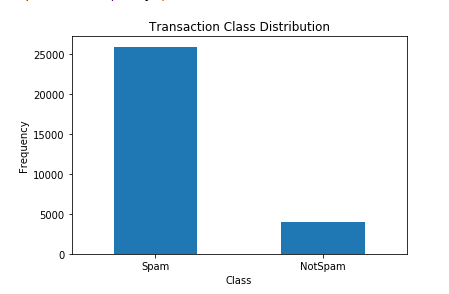


Figure 34 – Imbalanced Dataset

### Handling Imbalanced Datasets

To handle imbalanced dataset, the work used a technique called as sampling for given datasets. This technique copies some random points of minority class and hence the size becomes equal to majority class. To solve this issue Random Over Sampler function was used which is present in imblearn class in scikit-learn.

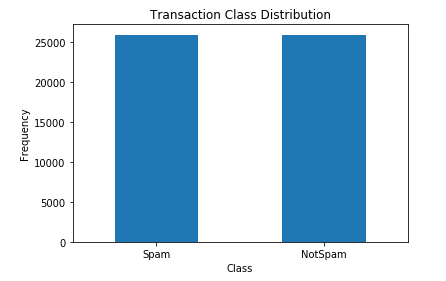


Figure 35 – Handling Imbalanced Dataset

### Bivariate Analysis

The work also did some exploratory data analysis on these features i.e. rating and time and concluded that these features cannot be separated by some linear function. By visualizing these features, it can be observed which one works better.

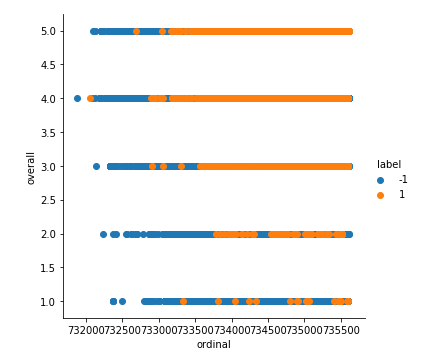


Figure 36 – Bivariate Analysis

#### Lemmatization

It is a process of combining different forms of a word together to be considered as a single item. It is similar to Stemming but it adds the context to the words and connects words with similar meaning to one word. Lemmatization is preferred over stemming because it does structural analysis of words.

#### Removing Stop Words or Punctuation

(is, am, are etc.) are the words that would not help us in detecting spam reviews so it is considered good to remove them before tokenizing the document. Do this to avoid un necessary token.

**For example**:

If the review is “This is a nice couple"

it will look like this after removing stop words or punctuation

“nice couple”

#### Feature Selection

Feature selection is an important method to remove needless words from a document as well. The work checks which features are correlated to the output. So, it is concluded that these features have very low value of correlation with respect to the output variable and so these features are not beneficial for our model to predict output hence we will rely only on linguistic features.

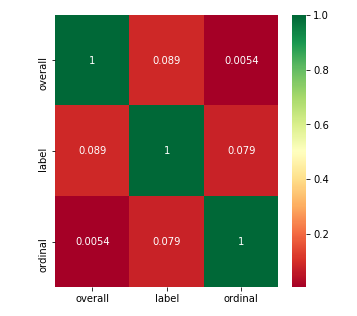


Figure 37– Feature Selection

#### Tokenizing

It is an important preprocessing technique. Text is split into individual words or sequences of words (n-grams). It is a complex and very difficult process regarding decision taking techniques.

**For example**:

The review (processed) “nice couple”

Unigram: ["nice", "couple"]

Bi-gram: ["nice couple"]

Uni + Bi-gram: [“nice”, “couple”, “nice couple”]

#### Document Term-Matrix

Machine learning models does not work well with textual data and we convert it into numeric data. After tokenizing we create a document term matrix based on them. Document term matrix is defined as:

DTM is a matrix containing mostly zero values or in other words a sparse matrix which is used to elaborate the tokens that occur in documents. Rows represent documents and columns represents tokens or terms. The work uses two simple techniques of DTM.

* **Simple Count:**

This technique is very simple and the no of times a term occurs is the value of term in matrix e.g. if a token appears twice its value is 2 otherwise 0 if it is not present at all.

* **Term Frequency and Inverse document frequency (TF-IDF):**

Term frequency and inverse document frequency is used to define how significant a token is to a document. The value of TF-IDF is proportional to the number of times a token appears and also it mostly decreases by the frequency of word in the corpus. TF-IDF is the most used and popular term-weighting scheme. It gives better results than simple count. TF-IDF can be mathematically represented as,

Equation 3

where,

* is frequency of term t in document d.
* is no of terms in document d.

Equation 4

where,

* N is total No of documents.
* is no of document with Term t in them.

Suppose a review contains 200 words in which the word couple appears 5 times. The term frequency (i.e., tf) for couple is then

(5/200) = 0.025.

Suppose now that we have 10 million reviews and the word couple appears in a thousand of them. Then, the inverse document frequency (i.e. IDF) is calculated as

ln (10,000,000 / 1,000) = 4

Thus, the weight TF-IDF is the product of these quantities:

0.025 \* 4 = 0.1.

**Conclusion:**

Data pre-processing is one of the most significant part in text classification. Choosing preprocessing techniques sensibly because all techniques can produce different result and choosing the one that best suits you are the core steps. TF-IDF has proved better in testing in our case.

### Classification Algorithms

The work uses a totally unsupervised machine learning approach and selected some important classification and regression algorithms which was best for the work. The following algorithms were used.

#### Naïve Bayes Classifiers

The Bayes’ theorem based Naïve Bayes classifiers are linear classifiers. They are simple and efficient. It follows the assumption that the features in a dataset are mutually independent.

**Mathematical representation:**

Equation 5

Where:

* *P* (*c | x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
* *P*(*c*) is the prior probability of *class*.
* *P* (x *| c*) is the likelihood which is the probability of *predictor* given *class*.
* *P*(*x*) is the prior probability of *predictor*.

**Advantages**

* Independence assumption is often violated, but Naïve Bayes classifiers performs efficiently under unrealistic assumptions [18].
* Unlike discriminative models such as logistic regression Naïve Bayes classifier will converge quicker even if the assumption doesn’t hold it still works fine [19].

**Disadvantages**

* Its main disadvantage is that it can’t learn interactions between features.

#### Multinomial Naïve Bayes:

The work uses the Multinomial Naïve Bayes. It is used for discrete counts. In a text classification problem, we do a count on how many times a word occurs in the document rather than how many times word occurs.

Equation 6

#### Random Forest

A random forest works as a meta estimator which fits numerous decision tree classifiers on several sub-samples of the dataset and use normalizing to improve the predictive accuracy and control over-fitting. It is used because of its simplicity and is widely used.

**Advantages**

* Easy to understand and elaborate.
* Feature interactions handling is easy through this and since they’re non-parametric, no need to worry about outliers or whether the data is linearly separable.
* Fast and scalable
* No need of tuning

**Disadvantages**

* Re-learning is not available.
* Overfits easily.

#### XGBoost:

XGBoost is derived from gradient boosted decision trees for speed and accuracy. It is a dominating algorithm on the structured or tabular data. It stands for extreme gradient boosting. It is made to boost the execution speed and model performance.

#### ANN:

ANN stands for Artificial Neural Networks. It is a process of learning to move samples in different classes by pointing out the common features between those samples of known classes. It is a set of connected I/O networks associated with weights on each connection. It consists of one input layer, one or more hidden or intermediate layers and one output layer. It is being used in pattern matching, medical fields etc.

#### LSTM:

LSTM stands for Long Short-Term Memory, and it is a type of recurrent neural network and it has the ability of learning order dependence in sequence prediction problems. It is used in complex problems like machine translation, and speech recognition.

#### KNN:

K-nearest neighbours is the simplest and most widely used algorithm in machine learning for classification. It uses data and find new data points on measures such as distant function. Finally, the classification is done by a majority vote of its neighbours.

#### Conclusion:

The main and core building any model or designing it is the selection of an algorithm. Choosing the one that best suits your model and gives the best results otherwise it will result in an overfitting and wrong results. It can be chosen by proper testing and training and the comparison of the values. In our case RFC (Random Forest Classifier) outperformed all others.

## Behavioural Feature Analysis

Behavioural features analysis emphases on behaviours of reviewers and content of reviews. We had worked on total eight group spam features.

|  |  |
| --- | --- |
| **Variables** | **Description** |
|  | Product ‘p’ |
|  | Review ‘r’ |
| rpr | Rating by reviewer ‘r’ on product p |
| Rg | Set of all reviewers of a group ‘g’ |
|  | Time of current review ‘r’ |
| rp | Average rating |

Table 5 *- Key for Behavioural Features Formulae*

### Average Rating Deviation

The average rating deviation a group g is given by:

Equation 7 [14]

Where Pr is the set that contains products reviewed by r, rpr is the rating score by reviewer r on product p and the modulus of rpr is the average rating score for product p. Maximum rating score is mostly 5, so maximum rating deviation is 4, which is used for the normalization.

### Penalty L(g)

To reduce the impact of groups of small size, penalty function is used in the proposed technique:

Equation 8 [14]

Where |Rg| represents the number of reviewers in group g, |Pg| represents the number of products in group g.

### Group Deviation

When the ratings of a group diverge from genuine reviewers. The worseness of the group depends on the larger the divergence. Group deviation demonstrates this behaviour on a 5-star rating measure. 4 is considered the maximum possible deviation:

Equation 9 [8]

Equation 10 [8]

where rp,g and rg,p are the average ratings for product p given by members of group g and by other reviewers not in g respectively. D(g, p) is the deviation of the group on a single product p. If there are no other reviewers who have reviewed the product p, rp,g = 0.

### Review Burst Ratio

Spammers in order to acquire maximum ratings and impact on a product’s sentiment, review target products in a short time interval. The review burst ratio by a reviewer r is given by:

Equation 11 [14]

### Group Size

The group size of group g is defined as:

Equation 12 [14]

where |Rg| represents the number of reviewers in g.

### Group Size Ratio

The total number of reviewers in a group for a particular product can also determine spam activity. Suppose (worst case), a whole group are the only ones to review the product and controlling its value. On the other hand, if reviewers are very large in number on that particular product, then the dominance of the group is negligible.

Equation 13 [8]

Equation 14 [8]

where GSRp(g, p) is the ratio of group size to Mp (the set of all reviewers of product p) for product p.

### Review Tightness

The review tightness of group g is defined as:

Equation 15 [14]

### Group Support Count

### The total number of products that a group has reviewed and worked on is refered to as the support count of that group. Higher the support count more chance of being a spam group because the possibility of random people reviewing many products together is very rare. max(|Pgi|) is the maximum support count of all the groups, it is used to normalize it to [0,1].

Equation 16 [8]

### Group Content Similarity

Group involvement can be showed by content similarity. It is a simple technique in which duplication of reviews can be found easily. If a group copy reviews among themselves they are easier to identify through this technique. In this technique, cosine similarity formula is used. GCS is easy to make. The proposed projected normalize its value to [0,1].

Equation 17 [8]

Equation 18 [8]

If the value is approximately equals to 1 then group is definitely spam as all the members have copied their reviews on different products in P~~g~~ CSM(g, m) represents the average pairwise content similarity of member m ∈ g over all products in Pg. [8]

# System Testing and Experimental Results

## Testing of Linguistic Model

To find the most appropriate out of all pre-processing techniques, some portion of dataset is selected for testing. The value of k is set to 3 and 7 and a = 10. Vectorizing techniques discussed above with all possible combination of N-grams. Following is the information about dataset used in the testing.

|  |  |
| --- | --- |
| Label | Count |
| Not-Spam | 528142 |
| Spam | 80456 |
| Total Reviews | 608,598 |
| Number of Reviews to be used for Training | 50000 |
| Number of Reviews to be used for Testing | 10386 |

Table 6 *- Dataset Statistics used for Testing*

### Performance Metrices

### Confusion matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted Class | | |
| Actual Class |  | **Not Spam** | **Spam** |
| **Not Spam** | **True Negative**  *Not spam and predicted spam* | **False Positive**  *Not spam, but predicted spam* |
| **Spam** | **False Negative**  *Spam, but predicted not* | **True Positive**  *Spam and predicted spam* |

Table 7 *- Confusion Matrix*

### Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It simply tells that out of all the reviews labelled as spam, what fraction were perfect.

Equation 19

### Recall

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answer is that out of all the spam reviews in the dataset, what fraction did the classifier pick up.

Equation 20

### F-Measure

F1 Score is the weighted average of Precision and Recall. It takes both false positives and false negatives into account Also known as F1-Score, it conveys the balance between precision and recall.

Equation 21

### Accuracy

Accuracy is the most intuitive performance measure; it is simply a ratio of correctly predicted observation to the total observations. But that’s does not mean that high accuracy promises best model. Accuracy is a great measure only when values of false positive and false negatives are almost same.

Equation 22

### AUROC

AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example,

Equation 23

Area Under the receiver operating characteristics (AUROC) curve is the most commonly used way to visualize the performance of a binary classifier. It is created by plotting true positive rate to false positive rate. AUC can be the best way to evaluate the performance of classifier in a single number.

### Brier Score (Mean Square Error)

It is the average difference (mean squared difference) between predicted probabilities and actual outcomes. A good classifier will have minimum brier score. Mean square error or Brier score is also a good measure to evaluate a classifier [20].

Equation 24

### Conclusion

All above defines measures have their own significances along with their pros and cons, to decide whether a trained model is good or bad we cannot just rely on a single measure, to decide whether our trained model is performing well or not, we need to evaluate our model against all above defined measures.

Using Count Vectorizer

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **K=3 , a=10** | | **Confusion Matrix** | **BL** | **P** | **R** | **F1** | **A** | **AUROC** |
| **Unigram** | **KNN** | [[9538 73]  [ 443 332]] | 0.05 | 0.94 | 0.95 | 0.94 | 0.95 | 0.70 |
| **Multinomial NB** | [[9028 583]  [ 451 324]] | 0.1 | 0.91 | 0.90 | 0.90 | 0.90 | 0.68 |
| **XG Boost** | [[9585 26]  [ 446 329]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.71 |
| **ANN** | [[9593 18]  [ 736 39]] | 0.07 | 0.93 | 0.93 | 0.89 | 0.93 | 0.51 |
| **LSTM** | [[9611 0]  [ 775 0]] | 0.07 | 0.86 | 0.93 | 0.89 | 0.93 | 0.50 |
| **RF** | [[9600 11]  [ 450 325]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.71 |
| **Unigram + Bigram** | **KNN** | [[9543 68]  [ 446 329]] | 0.05 | 0.94 | 0.95 | 0.94 | 0.95 | 0.71 |
| **Multinomial NB** | [[8876 735]  [ 399 376]] | 0.11 | 0.91 | 0.89 | 0.90 | 0.89 | 0.70 |
| **XG Boost** | [[9589 22]  [ 440 335]] | 0.05 | 0.95 | 0.95 | 0.95 | 0.95 | 0.71 |
| **ANN** | [[9583 28]  [ 732 43]] | 0.07 | 0.91 | 0.93 | 0.90 | 0.93 | 0.53 |
| **LSTM** | [[9611 0]  [ 775 0]] | 0.07 | 0.86 | 0.93 | 0.89 | 0.93 | 0.50 |
| **RF** | [[9601 10]  [ 448 327]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.71 |
| **Unigram +**  **Bigram +**  **Trigram** | **KNN** | [[9531 80]  [ 444 331]] | 0.05 | 0.94 | 0.95 | 0.94 | 0.95 | 0.71 |
| **Multinomial NB** | [[8853 758]  [ 385 390]] | 0.11 | 0.91 | 0.89 | 0.90 | 0.89 | 0.71 |
| **XG Boost** | [[9588 23]  [ 442 333]] | 0..04 | 0.95 | 0.96 | 0.95 | 0.96 | 0.71 |
| **ANN** | [[9583 28]  [ 741 34]] | 0.07 | 0.90 | 0.93 | 0.90 | 0.93 | 0.52 |
| **LSTM** | [[9611 0]  [ 775 0]] | 0.07 | 0.86 | 0.93 | 0.89 | 0.93 | 0.50 |
| **RF** | [[9602 9]  [ 450 325]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.71 |

Table 8 *- Performance Measures using Count Vectorizer and k=3 and a=1*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **K=7, a=10** | | **Confusion Matrix** | **BL** | **P** | **R** | **F1** | **A** | **AUROC** |
| **Unigram** | **KNN** | [[10304 8]  [ 38 36]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **Multinomial NB** | [[10277 35]  [ 58 16]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.61 |
| **XG Boost** | [[10310 2]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **ANN** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **LSTM** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **RF** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **Unigram + Bigram** | **KNN** | [[10305 7]  [ 37 37]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.75 |
| **Multinomial NB** | [[10224 88]  [ 39 35]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.73 |
| **XG Boost** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **ANN** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **LSTM** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **RF** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **Unigram +**  **Bigram +**  **Trigram** | **KNN** | [[10306 6]  [ 40 34]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.73 |
| **Multinomial NB** | [[10224 88]  [ 39 35]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.73 |
| **XG Boost** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **ANN** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **LSTM** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **RF** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |

Table 9 *- Performance Measures using Count Vectorizer and k=7 and a=10*

Using TF-IDF Vectorizer

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **K=3 , a=10** | | **Confusion Matrix** | **BL** | **P** | **R** | **F1** | **A** | **AUROC** |
| **Unigram** | **KNN** | [[9544 67]  [ 451 324]] | 0.05 | 0.95 | 0.95 | 0.94 | 0.95 | 0.71 |
| **Multinomial NB** | [[9603 8]  [ 748 27]] | 0.07 | 0.92 | 0.93 | 0.90 | 0.93 | 0.52 |
| **XG Boost** | [[9600 11]  [ 455 320]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.71 |
| **ANN** | [[9610 1]  [ 760 15]] | 0.07 | 0.93 | 0.93 | 0.89 | 0.93 | 0.51 |
| **LSTM** | [[9611 0]  [ 775 0]] | 0.07 | 0.86 | 0.93 | 0.89 | 0.93 | 0.50 |
| **RF** | [[9607 4]  [ 456 319]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.71 |
| **Unigram + Bigram** | **KNN** | [[9537 74]  [ 449 ] 326]] | 0.05 | 0.94 | 0.95 | 0.94 | 0.95 | 0.71 |
| **Multinomial NB** | [[9601 10]  [ 736 39]] | 0.07 | 0.92 | 0.93 | 0.90 | 0.93 | 0.52 |
| **XG Boost** | [[9577 34]  [ 432 343]] | 0.04 | 0.95 | 0.96 | 0.95 | 0.96 | 0.72 |
| **ANN** | [[9600 11]  [ 754 21]] | 0.07 | 0.91 | 0.93 | 0.89 | 0.93 | 0.51 |
| **LSTM** | [[9611 0]  [ 775 0]] | 0.07 | 0.86 | 0.93 | 0.89 | 0.93 | 0.50 |
| **RF** | [[9591 20]  [ 437 338]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.72 |
| **Unigram +**  **Bigram +**  **Trigram** | **KNN** | [[9526 85]  [ 437 338]] | 0.05 | 0.94 | 0.95 | 0.94 | 0.95 | 0.71 |
| **Multinomial NB** | [[9601 10]  [ 736 39]] | 0.07 | 0.92 | 0.93 | 0.90 | 0.93 | 0.52 |
| **XG Boost** | [[9572 39]  [ 428 347]] | 0..04 | 0.95 | 0.96 | 0.95 | 0.96 | 0.72 |
| **ANN** | [[9603 8]  [ 759 16]] | 0.07 | 0.91 | 0.93 | 0.89 | 0.93 | 0.51 |
| **LSTM** | [[9611 0]  [ 775 0]] | 0.07 | 0.86 | 0.93 | 0.89 | 0.93 | 0.50 |
| **RF** | [[9587 24]  [ 433 342]] | 0.04 | 0.96 | 0.96 | 0.95 | 0.96 | 0.72 |

Table 10 *- Performance Measures using TF-IDF Vectorizer and k=3 and a=10*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **K=7 , a=10** | | **Confusion Matrix** | **BL** | **P** | **R** | **F1** | **A** | **AUROC** |
| **Unigram** | **KNN** | [[10307 5]  [ 36 38]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.76 |
| **Multinomial NB** | [[10308 4]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **XG Boost** | [[10310 2]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **ANN** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **LSTM** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **RF** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **Unigram + Bigram** | **KNN** | [[10305 7]  [ 40 34]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.73 |
| **Multinomial NB** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **XG Boost** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **ANN** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **LSTM** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **RF** | [[10305 7]  [ 40 34]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **Unigram +**  **Bigram +**  **Trigram** | **KNN** | [[10306 6]  [ 38 36]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **Multinomial NB** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **XG Boost** | [[10310 2]  [ 39 35]] | 0..0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |
| **ANN** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **LSTM** | [[10312 0]  [ 74 0]] | 0.01 | 0.99 | 0.99 | 0.99 | 0.99 | 0.50 |
| **RF** | [[10311 1]  [ 39 35]] | 0.0 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 |

Table 11 - Performance Measures using TF-IDF Vectorizer and k=7 and a =10

**Findings**

In Table 8, from the Confusion matrix, Accuracy and AUROC we can say that Random Forest and XGBoost and KNN have performed better than all others. In Table 9, results are relatively increased than the results of table 7. All models performed better as we changed the value of k = 7. In Table 10, according to the results we can see that the XGBoost has performed better than all others even better than what we had in Table 8. In Table 11, results are comparatively higher than the result of table 9. Increasing k=7 has increased the performance of the models.

### Graphs

* **Accuracy Chart**

A bar chart showing accuracies of all models. In X-axis, we have values for accuracy and in Y-axis we have our trained models Longer bar means higher accuracy

* **AUROC Chart**

A column chart showing AUROC values all models. In X-axis, we have all the models whereas in Y-axis we have values for AUROC

* **Brier Loss Graph**

A simple bar chart showing Brier Loss or Mean square error for all models. In X-axis, we have all models and in Y-axis we have values of Brier Loss. As we are referring to the error values, so the lower value or minimum bar is better.

* **ROC Curves**

ROC (Receiver operating characteristics) curves are curves which are obtained by plotting true positive rate against the false positive. The curve having larger AUC (Area under the curve) will represent the best model. The graph also shows Brier loss values for each model right next to their name in graph legend area

* **Mean Predicted Values**

It’s a graph plotted between the No of records which were predicted and mean values of their predicted outcome. It helps us to analyse what our model has predicted the most or less, and can also be used to define customer threshold values for the probabilistic model.

* **Calibration Curves**

It’s a graph plotted for the primary purpose of learning the quality of the hypothesis through calibration curves and showing the precise measures of different classifiers.

Using Count Vectorizer

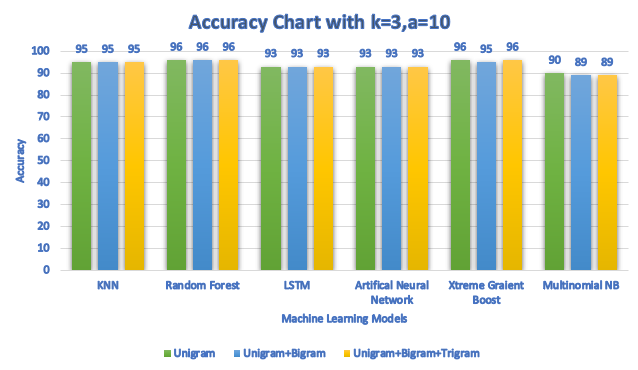


Figure 38 *- Accuracy Chart with Count Vectorizer and k=3 and a =10*

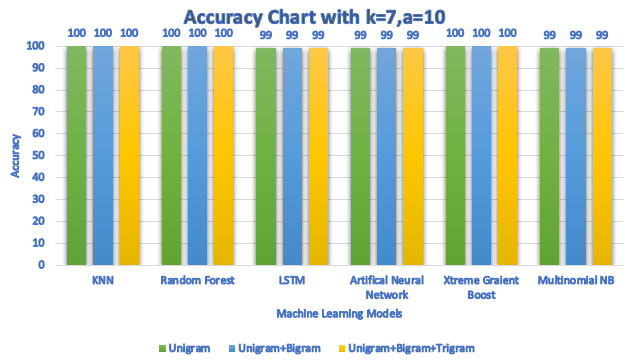


Figure 39*- Accuracy Chart for Count Vectorizer with k=7 and a=10*

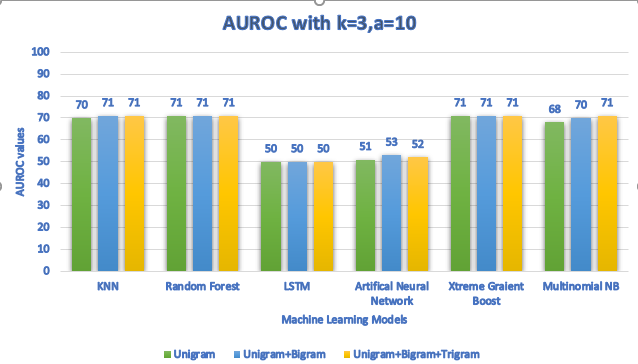


Figure 40*- AUROC Chart with Count Vectorizer and k=3 and a =10*

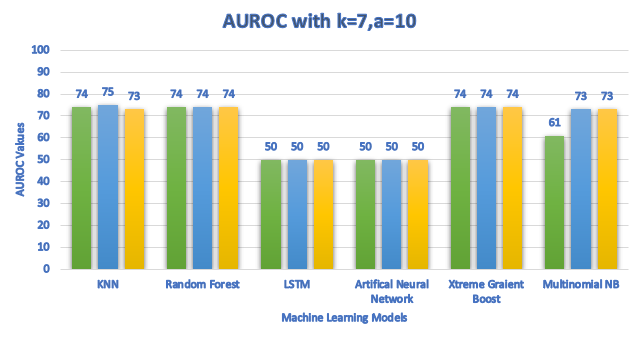


Figure 41 *- AUROC Chart with Count Vectorizer and k=7 and a=10*

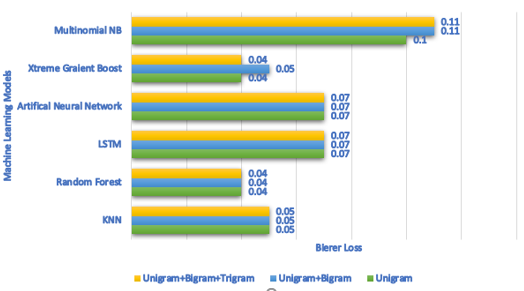


Figure 42 *- Brier Loss Chart for Count Vectorizer and k=3 and a=10*

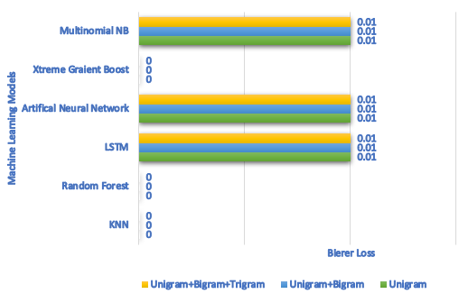


Figure 43 *- Brier Loss Chart for Count Vectorizer with k=7 and a=10*

**Findings**

In Figure 38, Random Forest has performed better providing 96 Accuracy better then all others. After RF Xtreme Gradient Boost and KNN are the top scorers. In Figure 39,RF, KNN, XGBoost, has scored extremely better respectively. In Figure 40,RF and XGBoost has performed better than all scoring 71 and 71 respectively KNN is also not far behind with 70.6 AUROC. In Figure 41, RF, KNN and XGBoost has scored the highest AUROC score 71, 71 and 71 respectively on average. In Figure 42, In Brier Loss, lowest value is better and the lowest of them all is 0.04, which is the Brier loss value for RF,all n-gram approaches have approximately same values i.e. 0 and Multinomial Naive Bayes and ANN and LSTM has the value 0.01.

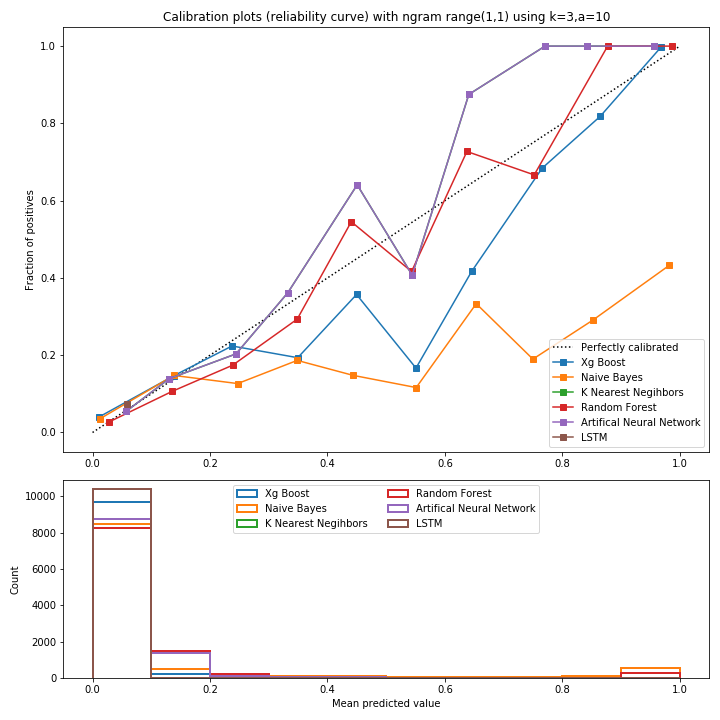


Figure 44 *- Comparison of Calibration of Classifiers using Count Vectorizer, with k = 3 and a=10 and Uni-gram*

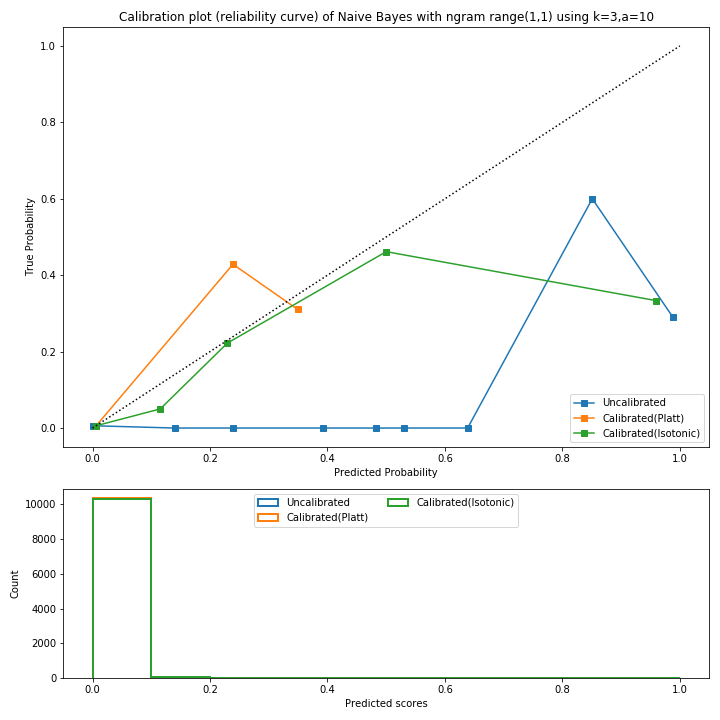


Figure 45 - *Probability Calibration Curve of Naïve Bayes & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni*

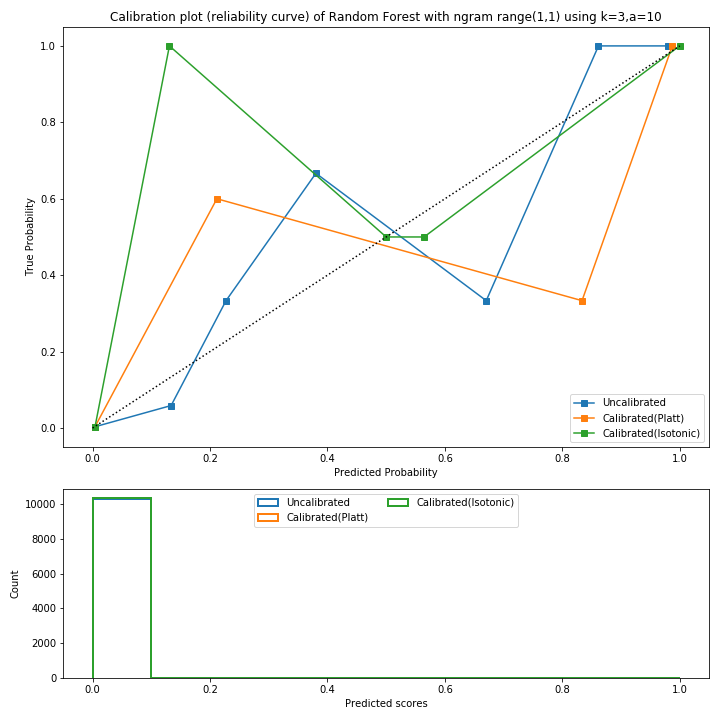


Figure 46 - *Probability Calibration Curve of Random Forest & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni*

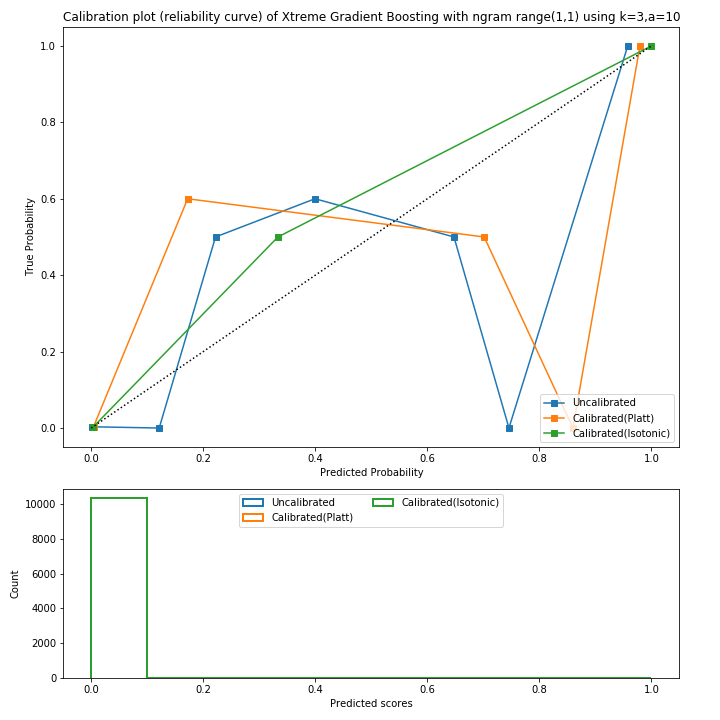


Figure 47 - *Probability Calibration Curve of XGBoost & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni*

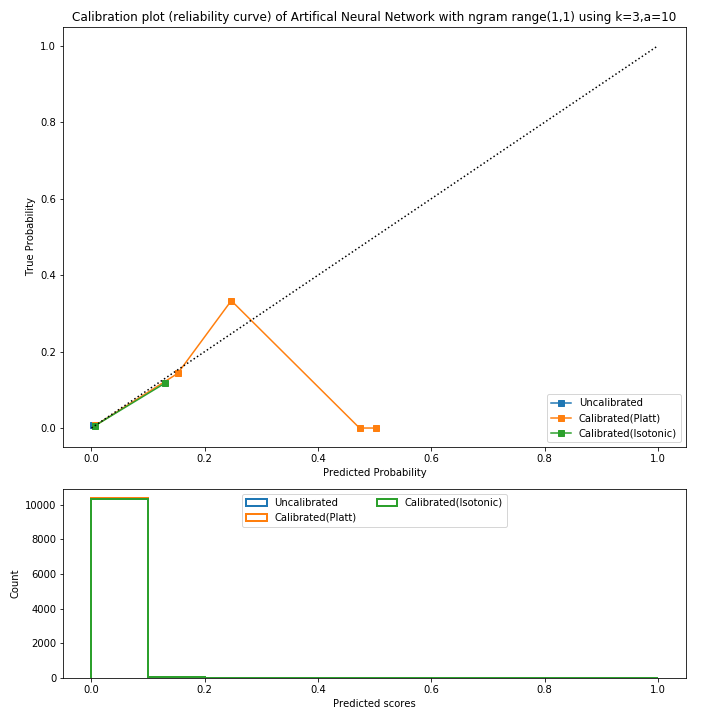


Figure 48 - *Probability Calibration Curve of ANN & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni*

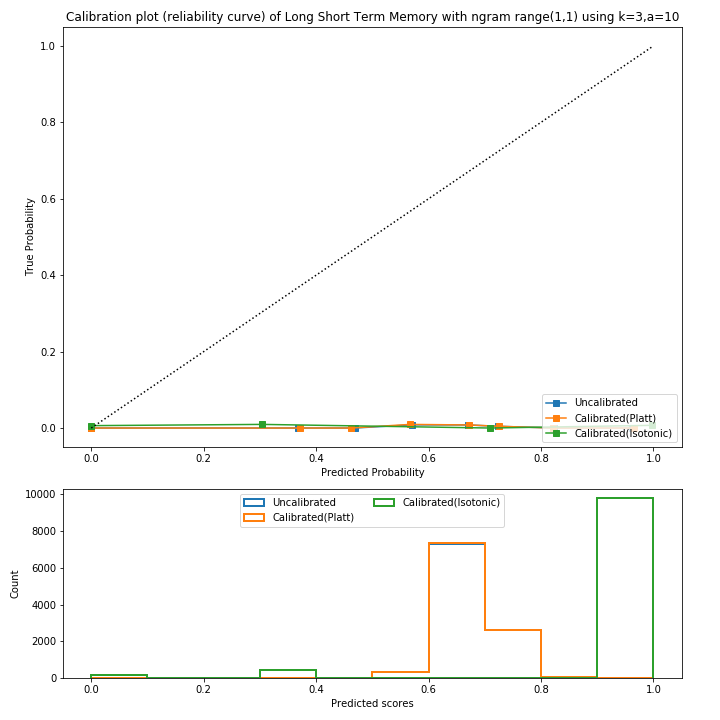


Figure 49 - *Probability Calibration Curve of LSTM & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni*

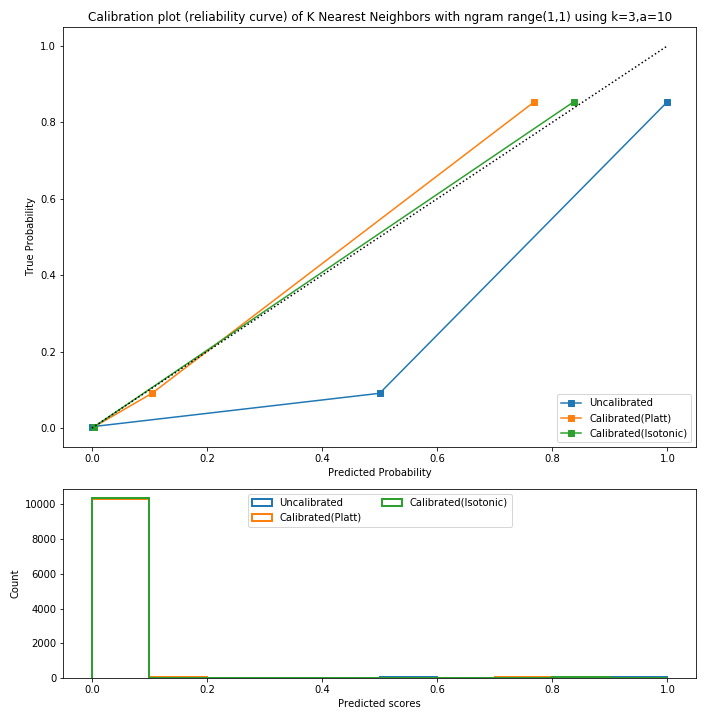


Figure 50 - *Probability Calibration Curve of KNN & Mean Predicted values using Count Vectorizer, with k=3 and a=10 and Uni*

**ROC Curve**

As we can see from the curves XGBoost, ANN and Random forest was better than the others. Whereas RF has shown slightly better results than the others.

**Mean Square Value**

Naïve Bayes has done most of its prediction between 0.7-0.8, whereas Random Forest has predicted mostly 0.3. Other models have moderate tendency towards different prediction values.

**Calibration Curve of Classifier**

Results were taken on unigram, uni+bigram and uni+trigram when k=3 and a =10 using Count Vectorizer and the results were almost same. In fact, it was still similar when k was changed to 7. In Figure 45, Figure 46, Figure 47, Figure 48, Figure 49 and Figure 50 Probability Calibration curves of all the classifier used in our project when k=3 and uni gram, are shown and here is a summary of the results in general:

* 1. Naïve Bayes is perfectly calibrated through isotonic until 0.4. But drops after that.
  2. Random Forest worked the most prominent as it is predicting values very close to perfectly calibrated through isotonic calibration.
  3. XGBoost shows the opposite of Random Forest.
  4. ANN has proven to be the least efficient.
  5. LSTM’s isotonic regression is the low.
  6. KNN is perfectly calibrated until 0.8 and works fine but gets uncalibrated after that.

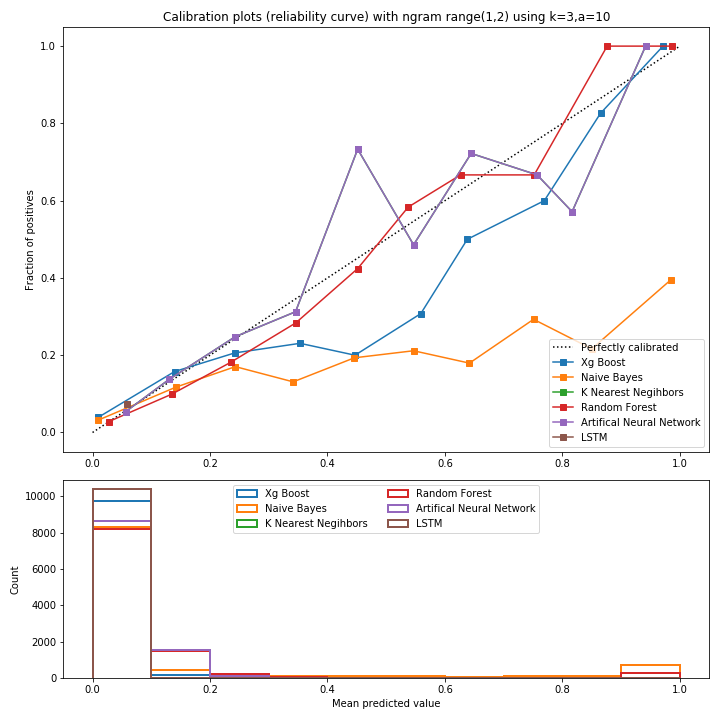


Figure 51 *- Comparison of Calibration of Classifiers using Count Vectorizer, with k = 3 and a=10 and Uni-gram + Bi gram*

**ROC Curve**

As we can see from the curves Random Forest, XGBoost and LSTM, are better than the other models showing more reliable curves. Whereas Random Forest has shown slightly better results than the others.

**Mean Square Value**

Naïve Bayes has done most of its prediction between 0.5-0.6, whereas Random Forest has predicted mostly 0.3. Others have predicted values between 0-0.1 and at random places.

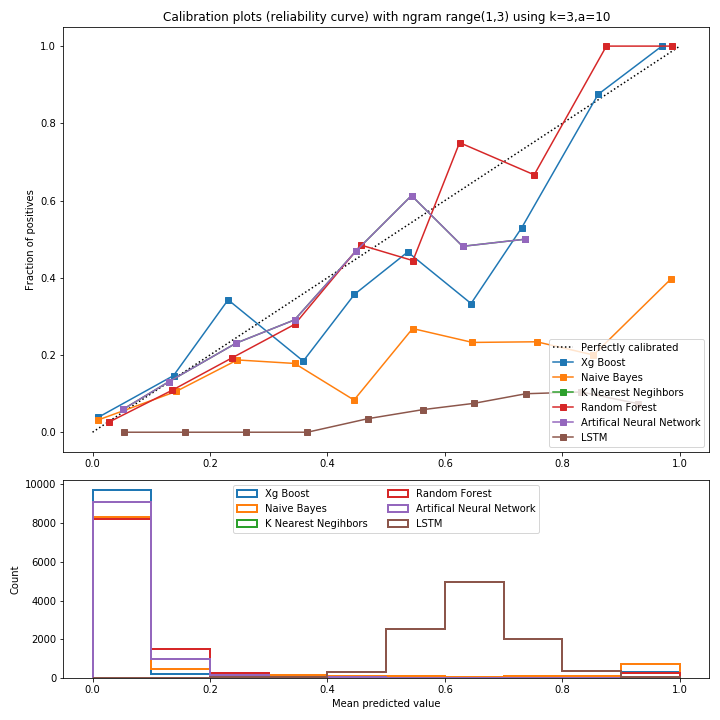


Figure 52 *- Comparison of Calibration of Classifiers using Vectorizer, k=3 and a =10 and Uni + Bi +Tri*

**ROC Curve**

As we can see from the curves Naïve Bayes has provided us with the bigger AUC out of others but its curve isn’t reliable more tends towards not spam. RF, LSTM and XGBoost are also better having comparatively more reliable curve.

**Mean Square Value**

SVM has done most of its prediction between 0.5-0.6, whereas Random Forest has predicted mostly 0.3. Bernoulli NB, Multinomial NB, LR and Random Forest respectively have. Predicted values between 0-0.1

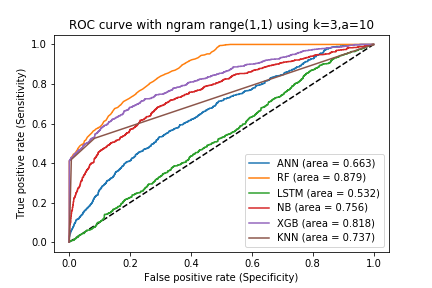


Figure 53 *- ROC curve with AUC Score k=3 and a=10 and Unigram*

When k = 3 and a = 10 and uni gram using Count Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

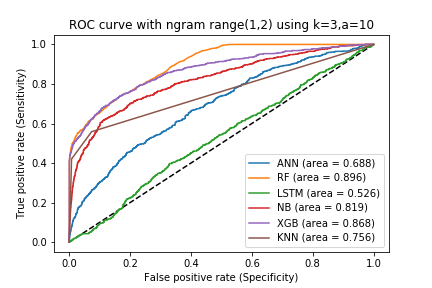


Figure 54- ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram

When k = 3 and a = 10 and uni + bigram using Count Vectorizer RF and XGB have the highest area in comparison with others. Then comes NB. Then KNN is also good whereas LSTM and ANN are the least in terms of area.

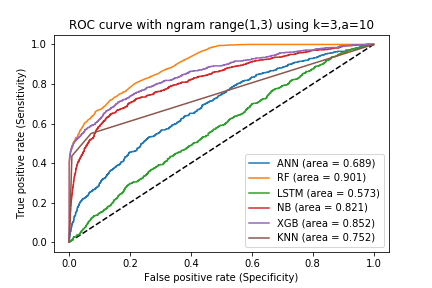


Figure 55 - ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram + Trigram

When k = 3 and a = 10 and uni+bi+trigram using Count Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

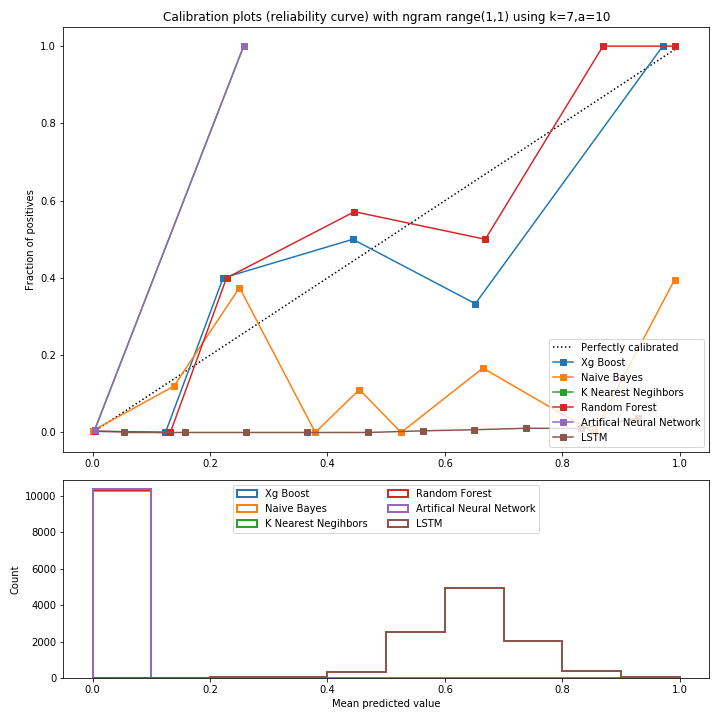


Figure 56 *- Comparison of Calibration of Classifiers using Count Vectorizer, with k=7 and a=10 and Unigram*

**ROC Curve**

The curves are not so reliable as these were when we have used k=7. Here LSTM, RF and XGBoost are better comparing to other models.

**Mean Square Value**

All models have predicted most of their values on different scales. With LSTM, and ANN predicting most values between 0.2-0.9 and ANN between 0.0-0.1. RF is near 0.1. Prediction of other values are comparatively less by all models.

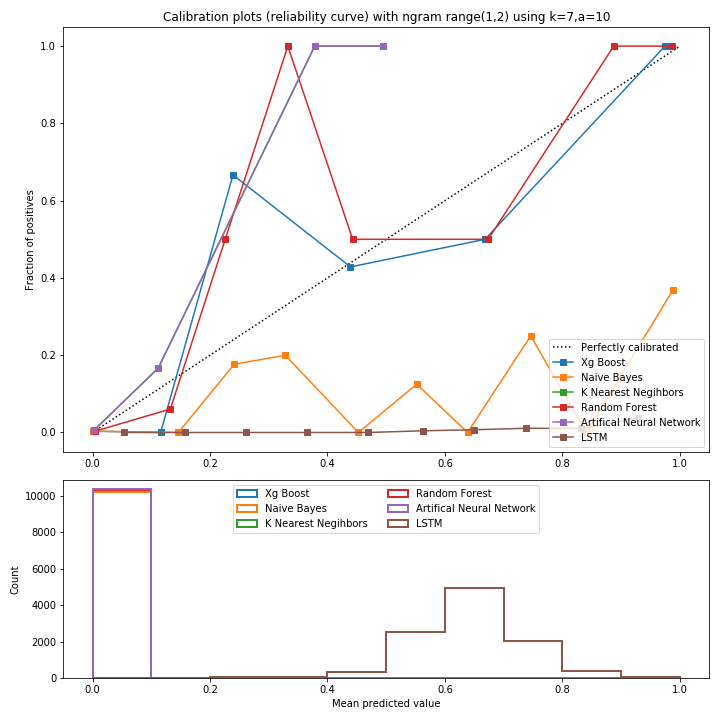


Figure 57 *- Comparison of Calibration of Classifiers using Count Vectorizer, with k=7 and a=10 and Uni + Bigram*

**ROC Curve**

The curves are also not so reliable here. Here RF, LSTM and XGBoost are better comparing to other models showing comparatively reliable curve.

**Mean Square Value**

All models have predicted most of their values between 0.2-0.5. With XGBoost, ANN and Random Forest predicting most values between 0.2-0.4 and LSTM between 0.4-0.9. Prediction of other values are comparatively less by all models.

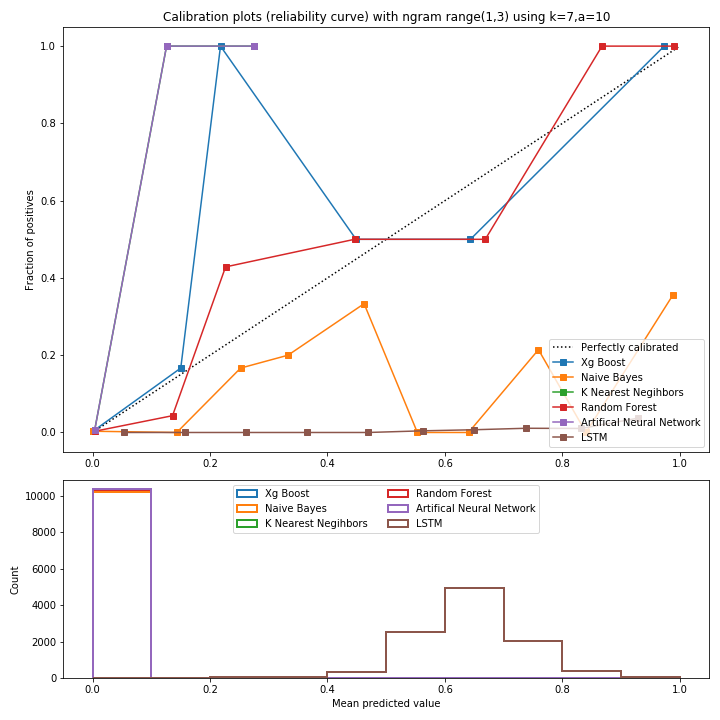


Figure 58 *- Comparison of Calibration of Classifiers using Count Vectorizer, with k=7 and a=10 and Uni + Bi + Tri*

**ROC Curve**

XGBoost and RF are better comparing to other models showing comparatively reliable curve. NB on the other hand has larger AU C but not so reliable curve.

**Mean Square Value**

All models have predicted most of their values between 0.1-0.3. With RF, NB and LSTM predicting most values between 0.2-0.4 and LSTM between 0.4-0.9. Prediction of other values are comparatively less by all models.

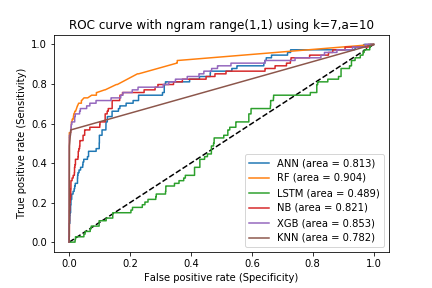


Figure 59 - ROC curve with AUC Score, with k=7 and a=10 and Unigram

When k = 7 and a = 10 and uni using Count Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

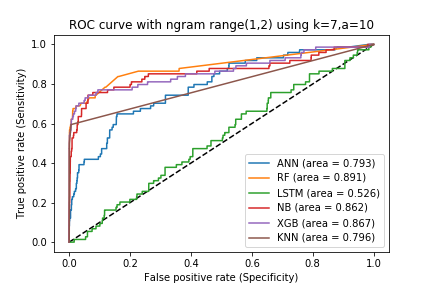


Figure 60 - ROC curve with AUC Score, with k=7 and a=10 and Uni + Bigram

When k = 7 and a = 10 and uni + bigram using Count Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

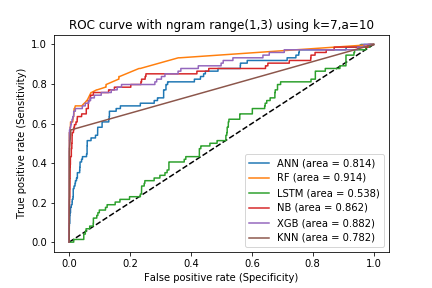


Figure 61- ROC curve with AUC Score, with k=7 and a=10 and Uni + Bigram + Trigram

When k = 7 and a = 10 and uni + bi + tri using Count Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

**Conclusion**

As we have seen from the above graph setting k = 3 gave us less reliable curves and not so good classifier models then before when we have used k=7 of all tokens using a=10. From our testing, we have seen that raising the value of k gives more reliable curves and models.

Using TF-IDF Vectorizer

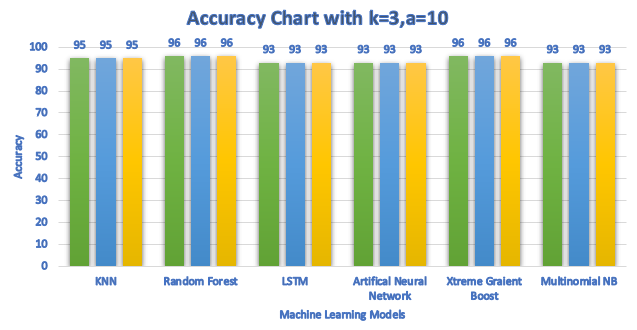


Figure 62*- Accuracy Chart for TF-IDF Vectorizer with k=3 and a=10*

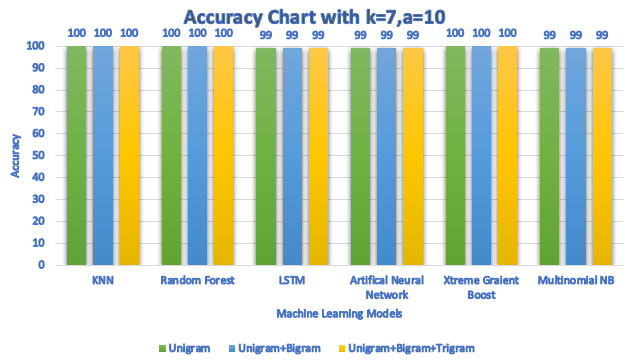


Figure 63 *- Accuracy Chart for TF-IDF Vectorizer with k=7 and a=10*

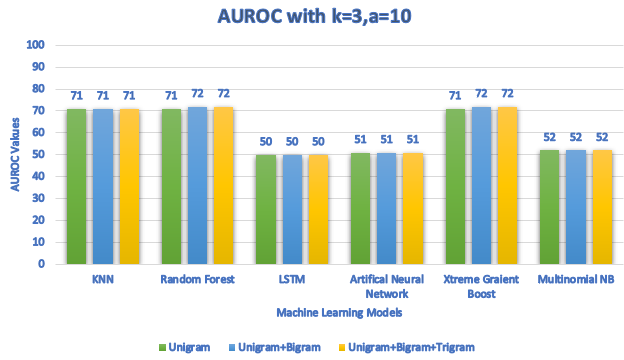


Figure 64 *- AUROC Chart with TF-IDF Vectorizer with k=3 and a=10*



Figure 65 *- AUROC Chart with TF-IDF Vectorizer with k=7 and a=10*

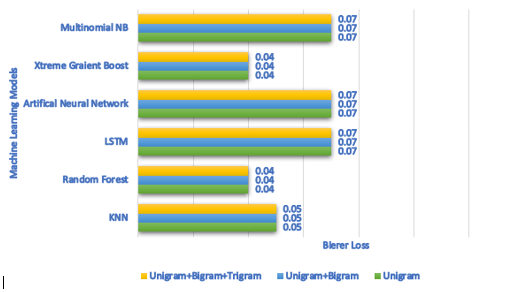


Figure 66 *- Brier Loss Chart with TF-IDF Vectorizer with k=3 and a=10*

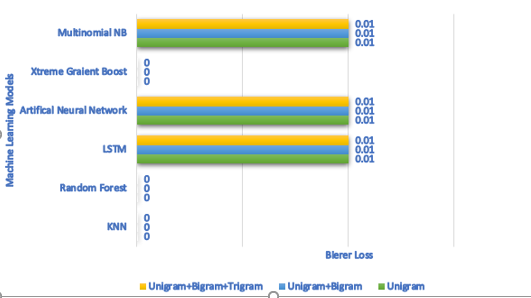


Figure 67 *- Brier Loss Chart with TF-IDF Vectorizer with k=7 and a=10*

**Findings**

In Figure 62, Accuracy Chart for TF-IDF Vectorizer Using k=3**:** Random Forest and XGBoost has performed better providing 96 Accuracy better then all others. After them KNN is the top scorer. In Figure 63,KNN, RF and XGBoost all scored equal. LSTM, Multinomial NB and ANN scored equal. In Figure 64,RF and XGBoost have provided the highest score of them all with 71.66 AUROC. After them, KNNhas performed better. In Figure 65,KNN has scored the highest and equal AUROC score 74.33. RF amd XGBoost has scored equal i.e. 74W. In Figure 66, we have Minimum Brier loss for XGBoost and RF which is 0.004. In Figure 67, all approaches of n-gram have provided us with the same result. With the lowest value of 0 for XGBoost, RF and KNN. All others have the value 0.001.

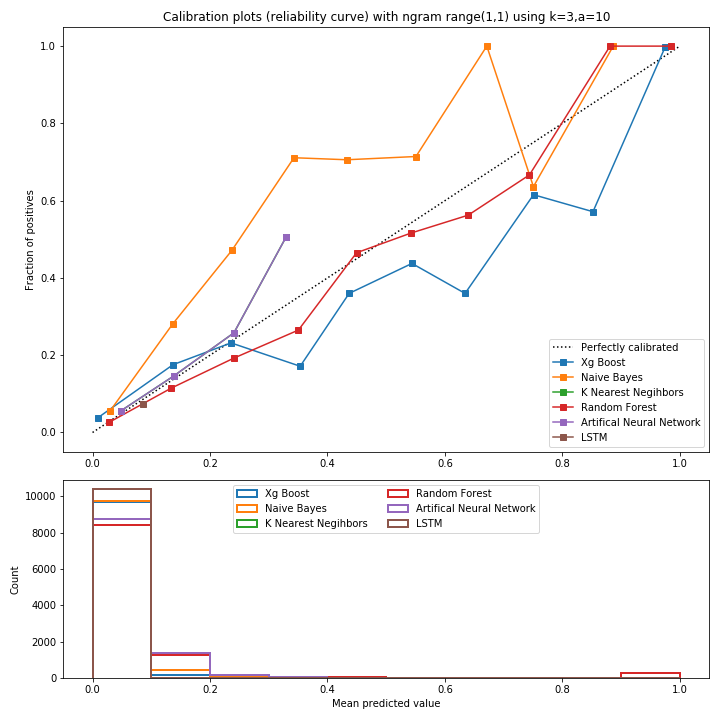


Figure 68 *- ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=3 and a=10 and Uni-gram*

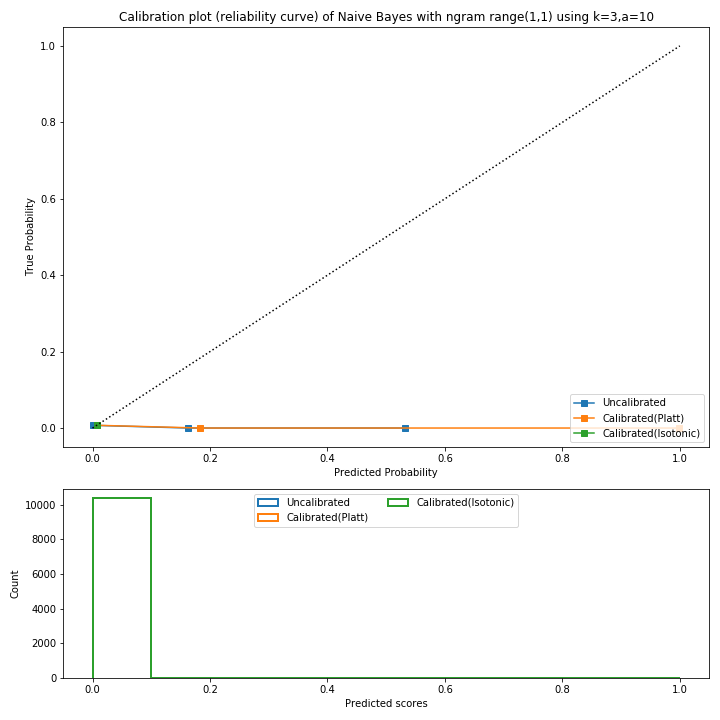


Figure 69 - *Probability Calibration Curve of Naïve Bayes & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni*

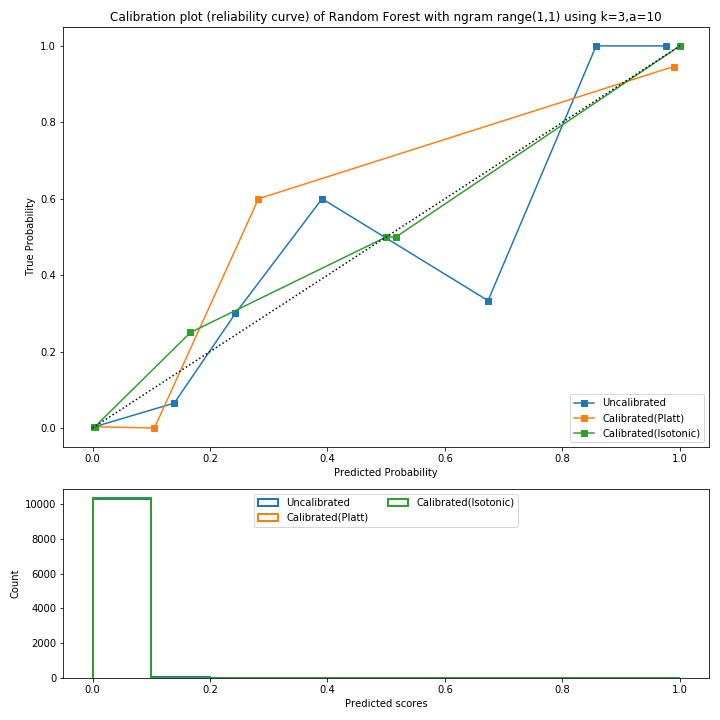


Figure 70 - *Probability Calibration Curve of Random Forest & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni*

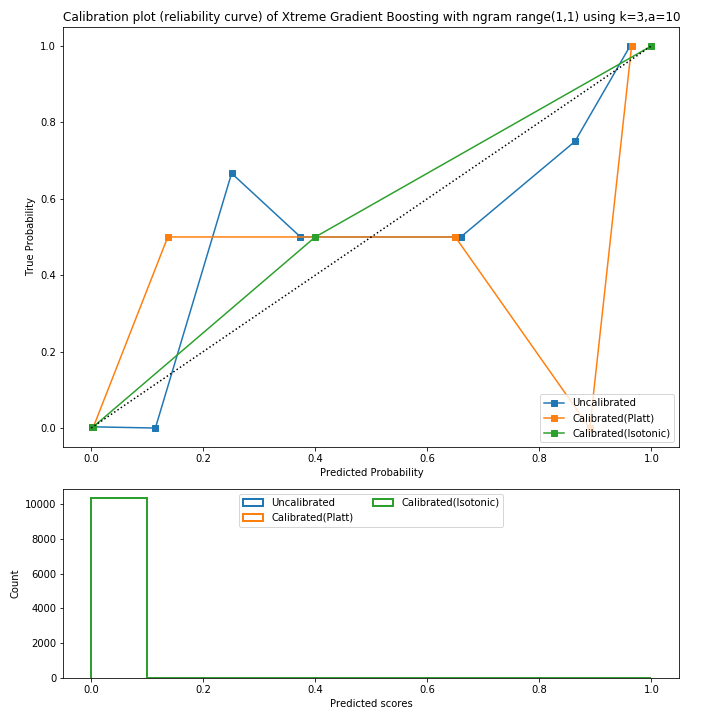


Figure 71 - *Probability Calibration Curve of XGBoost & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni*

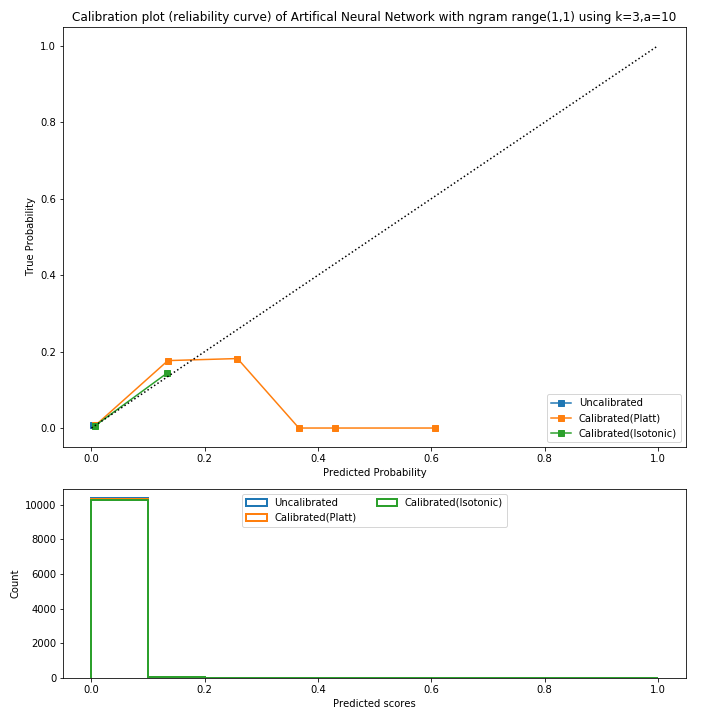


Figure 72 - *Probability Calibration Curve of ANN & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni*

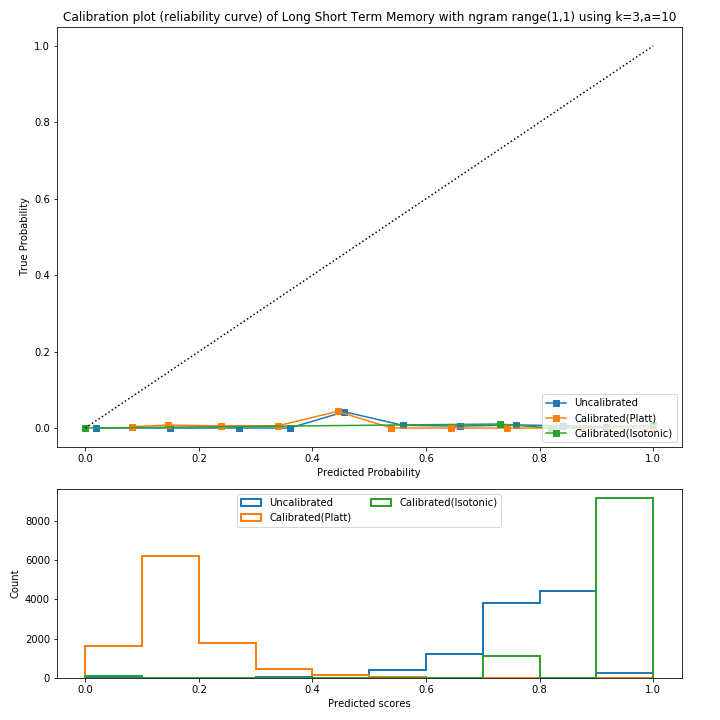


Figure 73 - *Probability Calibration Curve of LSTM & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni*

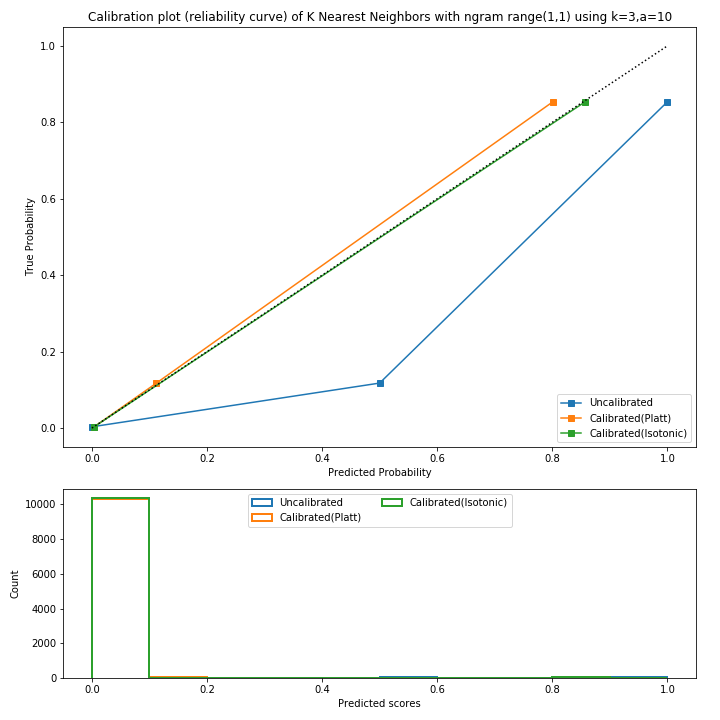


Figure 74 - *Probability Calibration Curve of KNN & Mean Predicted values using TF-IDF, with k=3 and a=10 and Uni*

**ROC Curve**

As we can see from the curves XGBoost, ANN and Random forest was better than the others. Whereas RF has shown slightly better results than the others.

**Mean Square Value**

Naïve Bayes has done most of its prediction between 0.7-0.8, whereas Random Forest has predicted mostly 0.3. Other models have moderate tendency towards different prediction values.

**Calibration Curve of Classifier**

Results were taken on unigram, uni+bigram and uni+trigram when k=3 and a =10 using TF-IDF Vectorizer and the results were almost same. In fact, it was still similar when k was changed to 7. In Figure 69, Figure 70, Figure 71, Figure 72, Figure 73 and Figure 74 Probability Calibration curves of all the classifier used in our project when k=3 and uni gram, are shown and here is a summary of the results in general:

* 1. Naïve Bayes is the worst as it shows a flat line on all points.
  2. Random Forest worked the most prominent as it is predicting values very close to perfectly calibrated through isotonic calibration.
  3. XGBoost shows the opposite of Random Forest.
  4. ANN has proven to be the least efficient.
  5. LSTM’s isotonic regression is the second lowest.
  6. KNN is perfectly calibrated until 0.8 and works fine but gets uncalibrated after that.

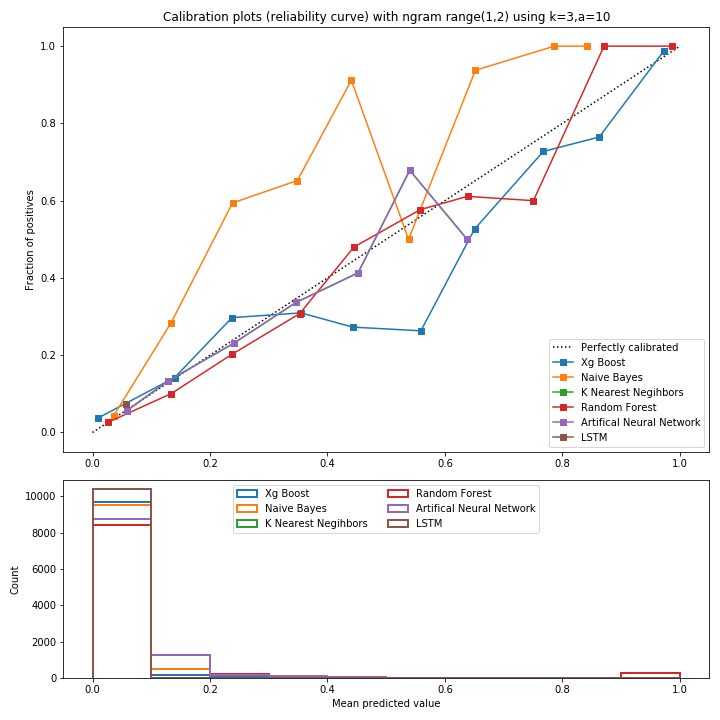


Figure 75 *- ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k =3 and a=10 and Uni + Bi*

**ROC Curve**

As we can see from the curves Random Forest, XGBoost and LSTM, are better than the other models showing more reliable curves. Whereas Random Forest has shown slightly better results than the others.

**Mean Square Value**

Naïve Bayes has done most of its prediction between 0.5-0.6, whereas Random Forest has predicted mostly 0.3. Others have predicted values between 0-0.1 and at random places.

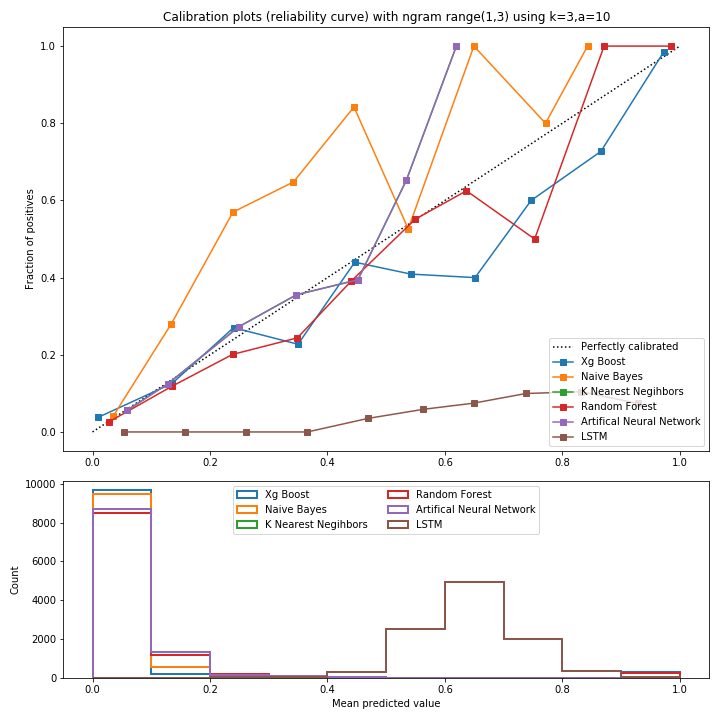


Figure 76 *- ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=3 and a=10 and Uni + Bi + Tri*

**ROC Curve**

As we can see from the curves Naïve Bayes has provided us with the bigger AUC out of others but its curve isn’t reliable more tends towards not spam. RF, LSTM and XGBoost are also better having comparatively more reliable curve.

**Mean Square Value**

SVM has done most of its prediction between 0.5-0.6, whereas Random Forest has predicted mostly 0.3. Bernoulli NB, Multinomial NB, LR and Random Forest respectively have. Predicted values between 0-0.1.

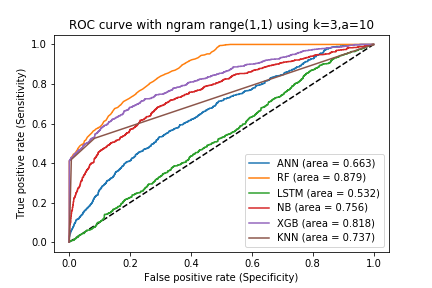


Figure 77- ROC curve using TF-IDF Vectorizer, with k=3 and a=10 and Uni

When k = 3 and a = 10 and uni gram using TF-IDF Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

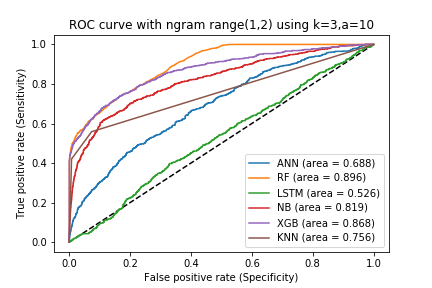


Figure 78 - ROC curve using TF-IDF Vectorizer, with k=3 and a=10 and Uni + bigram

When k = 3 and a = 10 and uni + bigram using TF-IDF Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

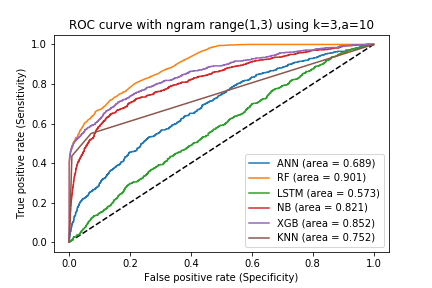


Figure 79- ROC curve using TF-IDF Vectorizer, with k=3 and a=10 and Uni+Bi+Tri Gram

When k = 3 and a = 10 and uni + bigram + trigram using TF-IDF Vectorizer RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

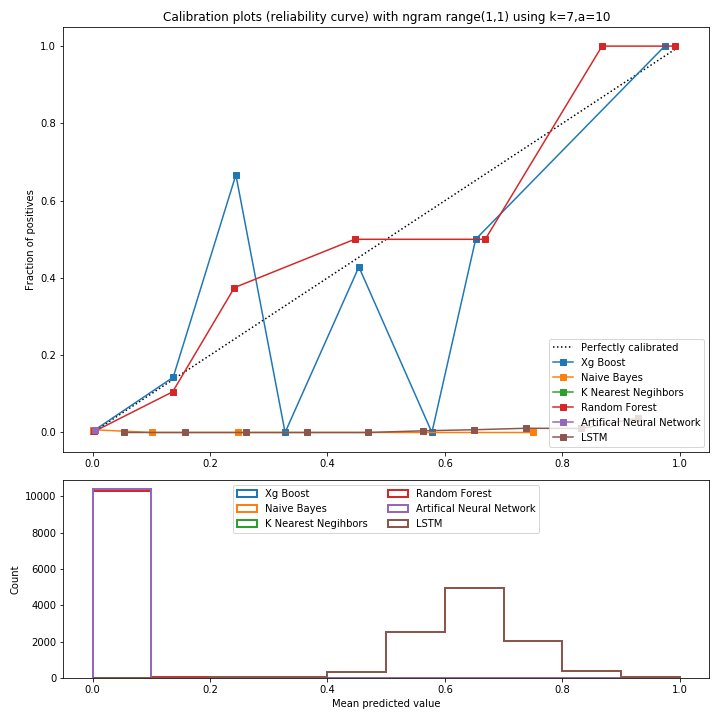


Figure 80 *- ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=7 and a=10 and Unigram*

**ROC Curve**

The curves are not so reliable as these were when we have used k=7. Here LSTM, RF and XGBoost are better comparing to other models.

**Mean Square Value**

All models have predicted most of their values on different scales. With LSTM, and ANN predicting most values between 0.2-0.9 and ANN between 0.0-0.1. RF is near 0.1. Prediction of other values are comparatively less by all models.

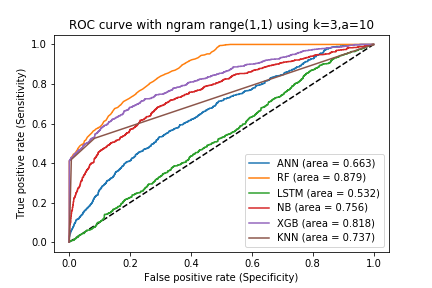


Figure 81 *- ROC curve with AUC Score k=3 and a=10 and Unigram*

When k = 3 and a = 10 and uni gram using TF-IDF RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

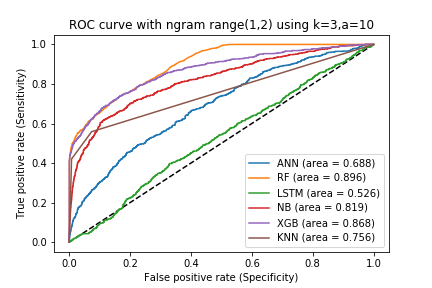


Figure 82- ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram

When k = 3 and a = 10 and uni gram + bi gram using TF-IDF RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

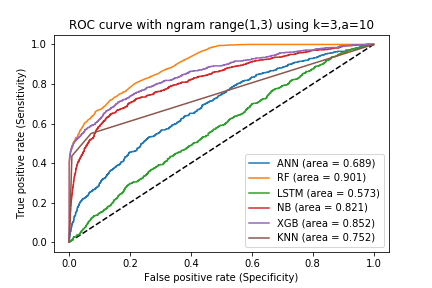


Figure 83 - ROC curve with AUC Score k=3 and a=10 and Unigram + Bigram + Trigram

When k = 3 and a = 10 and uni gram + bi gram + tri gram using TF-IDF RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

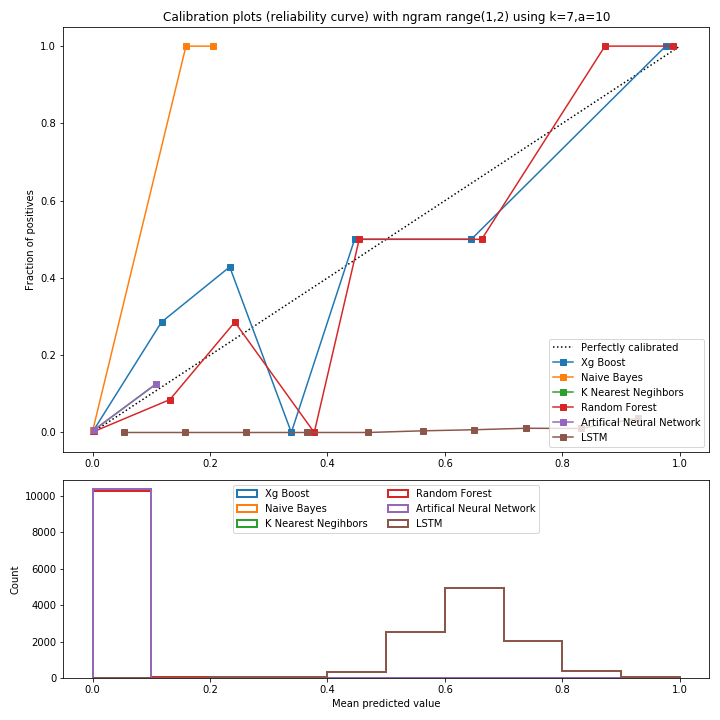


Figure 84 *- ROC curve & Mean Predicted values using TF-IDF Vectorizer, with k=7 and a=10 and Uni + Bi*

**ROC Curve**

Here as we can see that LR and SVM have performed comparatively better

**Mean Square Value**

Most predictions by all models are between the range of 0.2 - 0.6. Other values are less predicted values.

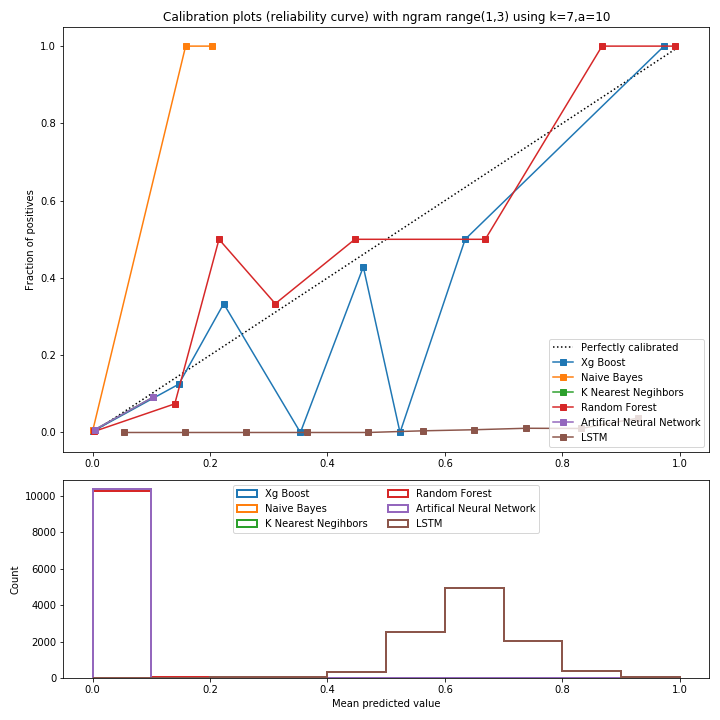


Figure 85 *- ROC curve & Mean Predicted values using TF-IDF Vectorizer with k=7 and a=10 and Uni + Bi + Tri*

**ROC Curve**

The curves are also not so reliable here. Here RF, LSTM and XGBoost are better comparing to other models showing comparatively reliable curve.

**Mean Square Value**

All models have predicted most of their values between 0.2-0.5. With XGBoost, ANN and Random Forest predicting most values between 0.2-0.4 and LSTM between 0.4-0.9. Prediction of other values are comparatively less by all models.

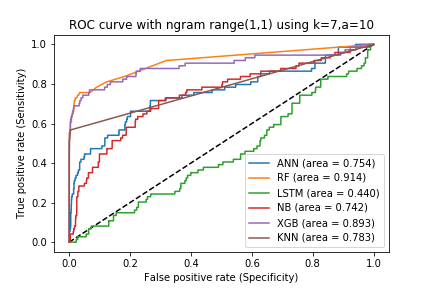


Figure 86 - ROC curve using TF-IDF Vectorizer, with k=7 and a=10 and Uni

When k = 7 and a = 10 and uni gram using TF-IDF RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

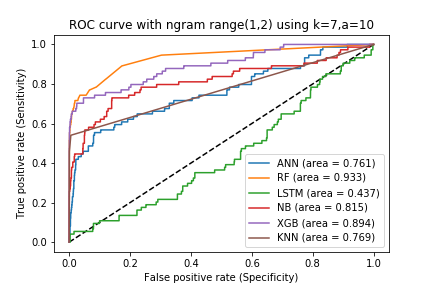


Figure 87 - ROC curve using TF-IDF Vectorizer, with k=7 and a=10 and Uni + Bi Gram

When k = 7 and a = 10 and uni gram using TF-IDF RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

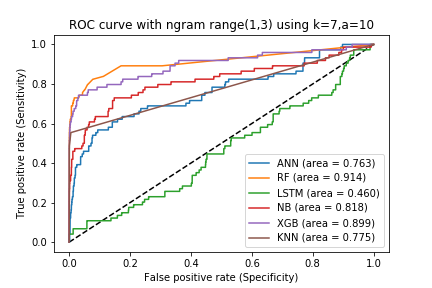


Figure 88 - ROC curve using TF-IDF Vectorizer, with k=7 and a=10 and Uni+Bi+Tri Gram

When k = 7 and a = 10 and uni gram using TF-IDF RF and XGB have the highest area in comparison with others. KNN and NB are also good whereas LSTM and ANN are the least in terms of area.

### Conclusion

As we can see from the performance metrices and graphs above both vectorising techniques has provided us with different results which we can summarize as

* Using Count Vectorizer the performance of the models was a bit tricky to understand sometimes RF was better than XGBoost and vice versa, sometimes it was KNN or LSTM.
* Using TF-IDF Vectorizer has shown us the best results we have seen by providing us with the most reliable curve and greater AUC better than any other test cases. Other than that, in all test cases performance of RF was better than all and after that XGBoost was comparatively better than other models in every test case.
* The best model obtained from all test results was Random Forest Classifier with TF-IDF Vectorizer using k=7 and Unigram + Bigram + Trigram.

## Results of Behavioural Model

The work ranks the candidate groups on the basis of behavioural features. Higher the suspicious score more chances of the group being group spam (*On x-axis there are behavioural feature values and on y-axis their suspicious score*). Below are the results we got:

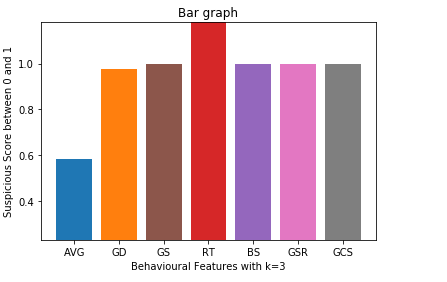


Figure 89– Results of Behavioural Features with k=3

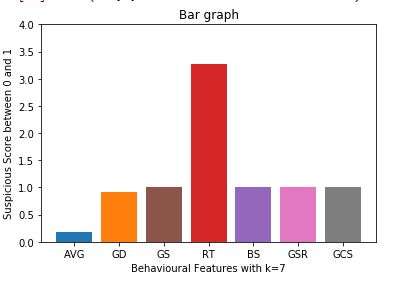


Figure 90– Results of Behavioural Features with k=7

#### Average Rating Deviation

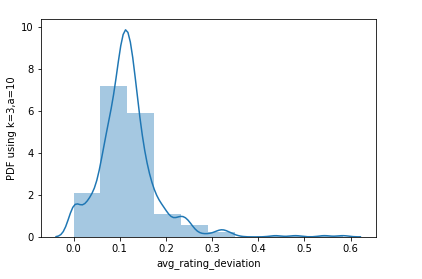


Figure 91 – PDF of Average Rating Deviation with k=3

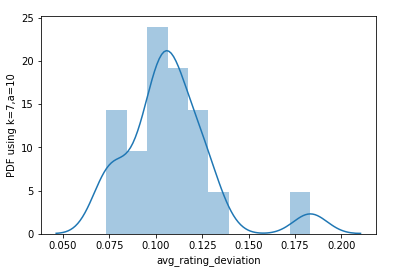


Figure 92 *- PDF of Average Rating Deviation (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7*

We can smoothen the histogram by PDF and in y-axis there is probability of distribution instead of count/frequency. This is from 0 to 1. We can say that 20 percentage of the total distribution present over here. Most of the values are dominant in range of 0.100 to 0.125. Most of the groups are deviate from their rating and it has highest value in 0.100 to 0.125.

#### Group Deviation

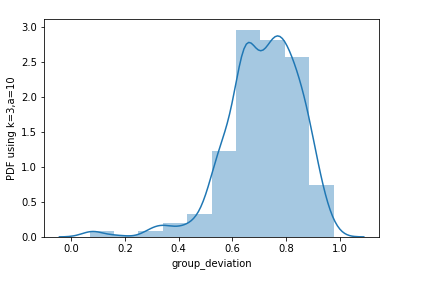


Figure 93 - PDF of Group Deviation (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3

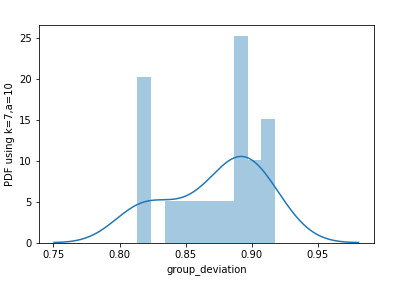


Figure 94 - PDF of Group Deviation (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7

10% of the total groups have highest group deviation. We can clearly see that 0 to 5 percent the group deviation are 0.75 to 0.80.

#### Review Burst Ratio

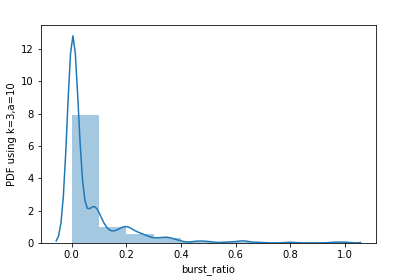


Figure 95 *- PDF of Review Burst Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3*

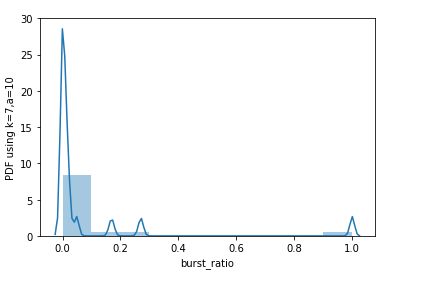


Figure 96 *- PDF of Burst Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7*

Most of the groups give their reviews between 0 and 1 so most of the spammer groups have 80% percent chance to do that.

#### Group Size

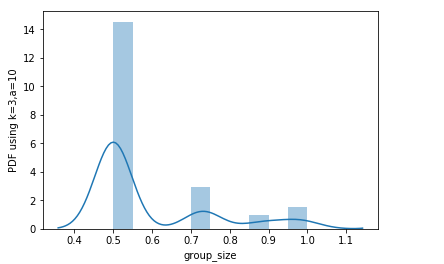


Figure 97 - PDF of Group Size (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3

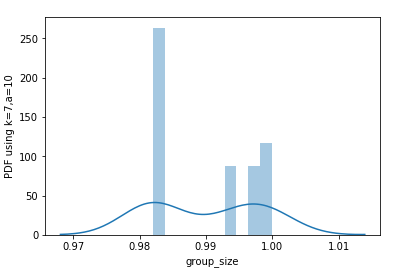


Figure 98 *- PDF of Group Size (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7*

Most of the groups having group size between range 0.98 to 1.00 because there is a highest probability for that range. 50 percent of the spammer groups have highest size and that range is from 0.98 to 1.00.

#### Group Size Ratio

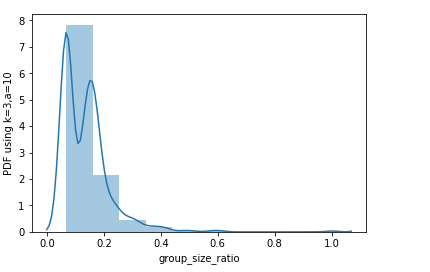


Figure 99 *- PDF of Group Size Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3*

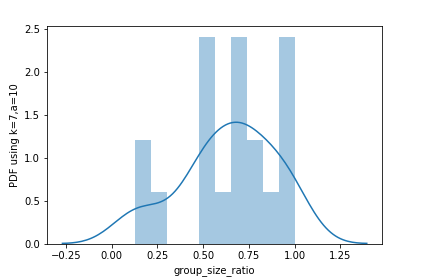


Figure 100 *- PDF of Group Size Ratio (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7*

The ratio of group size to the total number of reviewers for a product can also indicate spamming. At one extreme (worst case), the group members are the only reviewers of the product completely controlling the sentiment on.

#### Review Tightness

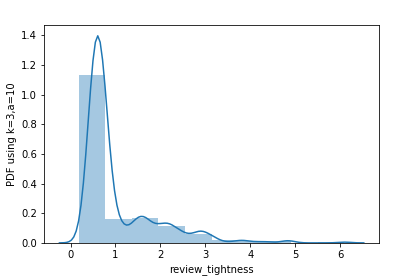


Figure 101 *- PDF of Review Tightness (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3*

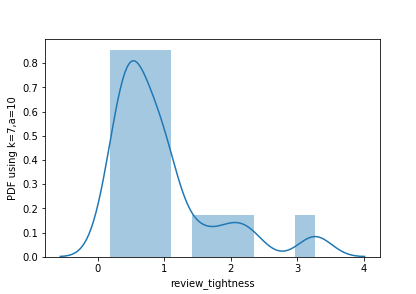


Figure 102 *- PDF of Review Tightness (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7*

Most of the groups give their reviews between 0 and 1 so most of the spammer groups have 80% percent chance to do that.

#### Group Support Count

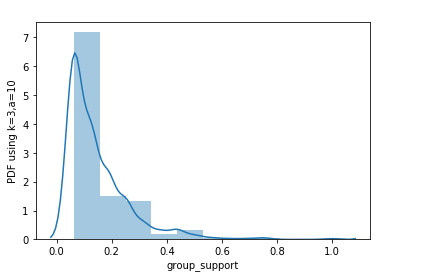


Figure 103 *- PDF of Group Support (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3*

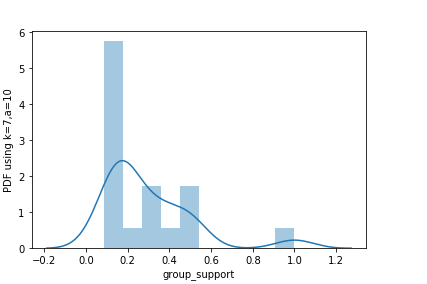


Figure 104 *- PDF of Group Support (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7*

Total number of products towards which the group has worked together. So, in this graph 20% chance having values between 0.2 to 0.4. In this range the spammer groups worked together.

#### Group Content Similarity

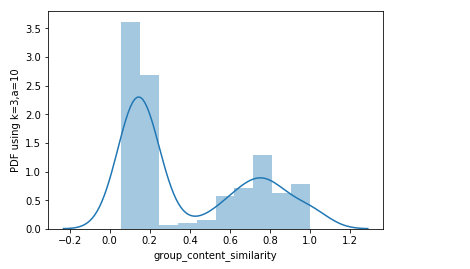


Figure 105 *- PDF of Group Content Similarity (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=3*

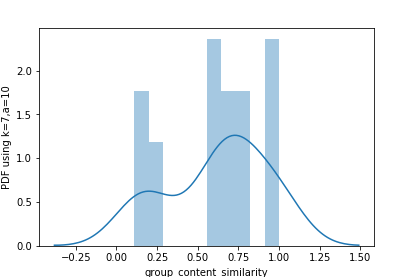


Figure 106- PDF of Group Content Similarity (X-Axis show value of average rating deviation and Y-Axis show cumulative percentages) with k=7

Spammer having 1 probability are more likely having similar content and this range is between 0.50 to 0.75.

# Conclusion

## Problems Faced

Group projects works just fine for R&D projects and generates effective results but, it is not easy to combine all the ideas in a single project. Defining the responsibilities of the group members and coordinating with the supervisors on the given time is very essential and one thing that is learned here is to work good and more in less time. Some of the problems we faced are discuss below:

* We had to read a lot of research papers to understand the problem we were dealing with.
* Then we had to collect the appropriate data out of those numerous research papers that was related to our project.
* After understanding the project and making a beginner layout of our project we had some problems in understanding different variables written in the research papers.
* When we couldn’t find a solution for that we had to read more papers.
* After finally picking out the desired content from research papers we started working on it, and at first things were difficult.
* It took almost 3 and a half months to complete the back-end coding of the project. But there were some issues with the computational time of it.
* There were some issues in the front-end as Django is still new to us.

Even though we faced many problems to achieve these outcomes and made several mistakes but we have also learned a lot of things. We learned to manage our time properly. How to speed things up? And similar things like that. We have learned data extracting techniques and how to play with the data through this project. We learned the significance of Software development lifecycle models. We learned the techniques to gather requirements and how to work on them. We learned new languages like Django, Python briefly.

## Project Summary

Fake reviews are a great risk to the genuineness of the reviewer community. Working in a group to control the sentiment of a product is the new go. This research helps eliminate this issue. This project spots spammer analysis using a largescale real- life dataset with high accuracy fake review labels and find a group of them working together. It classifies the reviews into two categories, i.e. group spam and not- group-spam. Firstly, a reviewer’s common products are checked with the set of reviewers available in the dataset and the co-review similarity to find whether the reviewer is suspicious or not. Then a suspicious reviewer graph is made of the reviewers then the group to group overlap matrix is made and on the basis of “k” given in the start the threshold matrix is prepared. From the threshold matrix candidate groups are extracted and behavioural features are applied on the review, then a suspicious score is prepared and the reviewers are ranked to get the final output. Count Vectorizer and TF-IDF Vectorizer were used for results evaluation. TF-IDF Vectorizer gave the best results we have seen by providing us with the most reliable curve and greater AUC better than any other test cases. Random Forest gave us the best model with TF-IDF Vectorizer using k=7 and Unigram + Bigram + Trigram with Precision, Recall, and AUROC being 1, 1, and 0.74 respectively.

## Future Work

Currently, the proposed project is limited by its scope. However, our study can be helpful to researchers for further work and to build a well-organized prototype. When the system comes in proper running form, we can save user’s data like location, time, IP Addresses to make our model more precise. This project has worked on group spammers detection and it can also identify the individual spammers with some minor changes. So, it can give the work two fully operational systems.

# References

[1] Jindal, N., & Liu, B. (2008, February). Opinion spam and analysis. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 219-230). ACM

[2] Jindal, N., Liu, B., & Lim, E. P. (2010, October). Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 1549-1552). ACM.

[3] Li, F., Huang, M., Yang, Y., & Zhu, X. (2011, July). Learning to identify review spam. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence* (Vol. 22, No. 3, p. 2488-2493).

[4] Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011, June). Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 309-319). Association for Computational Linguistics.

[5] Lim, E. P., Nguyen, V. A., Jindal, N., Liu, B., & Lauw, H. W. (2010, October). Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 939-948). ACM.

[6] Wang, G., Xie, S., Liu, B., & Philip, S. Y. (2011, December). Review graph based online store review spammer detection. In *Data mining (icdm), 2011 IEEE 11th international conference on*(pp. 1242-1247). IEEE

[7] Wu, G., Greene, D., Smyth, B., & Cunningham, P. (2010, July). Distortion as a validation criterion in the identification of suspicious reviews. In *Proceedings of the First Workshop on Social Media Analytics* (pp. 10-13). ACM.

[8]Mukherjee, A., Liu, B., & Glance, N. (2012, April). Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web* (pp. 191-200). ACM.

[9] Agarwal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference* (pp. 487-499).

|  |
| --- |
|  |
| [10] Ahsan, M. I., Nahian, T., Kafi, A. A., Hossain, M. I., & Shah, F. M. (2016, October). Review spam detection using active learning. In *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual* (pp. 1-7). IEEE. |

[11] Hai, Z., Zhao, P., Cheng, P., Yang, P., Li, X. L., & Li, G. (2016). Deceptive review spam detection via exploiting task relatedness and unlabeled data. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1817-1826).

[12] Adike, M. R., & Reddy, V. (2016). Detection of fake review and brand spam using data mining. *Int J Recent Trends Eng Res*, *2*(7), 251-256

[13] Wang, Z., Gu, S., & Xu, X. (2018). GSLDA: LDA-based group spamming detection in product reviews. *Applied Intelligence*, *48*(9), 3094-3107.

[14] Xu, G., Hu, M., Ma, C., & Daneshmand, M. (2019, May). GSCPM: CPM-Based Group Spamming Detection in Online Product Reviews. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.

[15] Rayana, S., & Akoglu, L. (2015, August). Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 985-994). ACM.

[16] J. Huang and C. X. Ling, “Using AUC and accuracy in evaluating learning algorithms,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, 2005.

[17] Django Tutorial for Beginners - Full Course(Jun 6, 2019) Youtube Video, Addedby Telusko [Online] Available at <https://www.youtube.com/channel/UC59K-uG2A5ogwIrHw4bmlEg> [Accessed 15 December 2019]

[18] S. Rosset, “Model selection via the AUC,” in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 89.

[19] F. Provost and P. Domingos, “Tree induction for probability-based ranking,” *Mach. Learn.*, vol. 52, no. 3, pp. 199–215, 2003.

[20] C. Ferri, J. Hernández-Orallo, and R. Modroiu, “An experimental comparison of performance measures for classification,” *Pattern Recognit. Lett.*, vol. 30, no. 1, pp. 27–38, 2009.

Appendix A

