CAREER EPISODE 2

SPAM DETECTION AND FILTERING ALGORITHM

2.1 INTRODUCTION

College	Leelaben Dashrathbhai Ramdas Patel (L.D.R.P.) Institute of Technology & Research
Location	Gandhinagar, India
Sub/Sem	Project– II, 7th
Chronology	August 2020 - Oct 2020
Supervisor	Sandip Modha
Role	Group Leader

2.2 FRAMEWORK

2.2.1

Spamming remains a pervasive issue in digital communications, with unwanted messages inundating inboxes and affecting system performance. One effective countermeasure to this nuisance is the k-Nearest Neighbors (KNN) algorithm, a renowned machine learning technique. This algorithm classifies data points based on their similarity to existing samples, making it a potent tool for distinguishing spam from legitimate messages. However, the effectiveness of KNN largely hinges on the quality and balance of the dataset used for training. Imbalanced datasets, where one class of data greatly outnumbers another, can skew results and undermine the accuracy of predictions. Resampling emerges as a pivotal solution in this scenario, allowing for the adjustment of data class distributions either by augmenting the minority class (oversampling) or reducing the majority class (under sampling). This ensures that the KNN algorithm operates on a balanced dataset, leading to more reliable and precise spam detection outcomes.

The endeavor is focused on the detection, separation, and removal of potential spam emails. It is intended to analyze the use of the filtering algorithm for spam detection and filtering of the detected spam. The study delved into the pressing issue of spam in electronic communication, exploring various filtering methods. The key tool of the initiative was the implementation of the k-Nearest Neighbors (KNN) algorithm, valued for its no-training requirement and ability to implement adaptive thresholds. Evaluations were conducted on datasets containing up to 1,000 emails, with a balance between spam and legitimate emails. Determining the ideal k value in the KNN algorithm and the optimal dataset size were essential steps. Using stratified validation, the study ensured balanced class labels. Results revealed the necessity for periodic feedback, with the ideal rate being 20. This feedback system was integrated with a real-time spam filtering process, refining the attributes after every 20-email sequence. The final evaluation showed improvements in the True Positive Rate (TPR), but the F-measure remained unchanged.

OBJECTIVES

The exploration of the use of the filtering algorithm for spam detection, separation, and removal was the major intention.

Also, other aims were;

- To reduce the number of spam emails obtained from emails
- To enhance feedback rates and achieve enhanced TPR outcomes.

2.2.2 WORK'S NATURE

I conducted various technical and managerial task as the team frontrunner during the project. I explored the challenges of electronic spam and then assessed the filtering algorithm's effectiveness as a filter. I selected relevant emails, applied filtering method, and gauged the filter's efficiency across different datasets. I pinpointed the k's best value in the KNN, tweaked dataset configurations, and executed a cross-validation of 10-fold. I analyzed the results from the experiments, assessed the update component, and incorporated a feedback system. Finally, I refined feedback frequencies, merged all insights, and emulated a real-time spam filtering protocol, aiming for improved TPR outcomes. Furthermore, my managerial duties involved controlling the seamless integration of individual steps and ensuring that the team stayed aligned with the project's purposes. Additionally, I communicated with other departments to guarantee cross-functional synergy as well as effective communication throughout the project's duration.

2.2.3 ORGANOGRAM

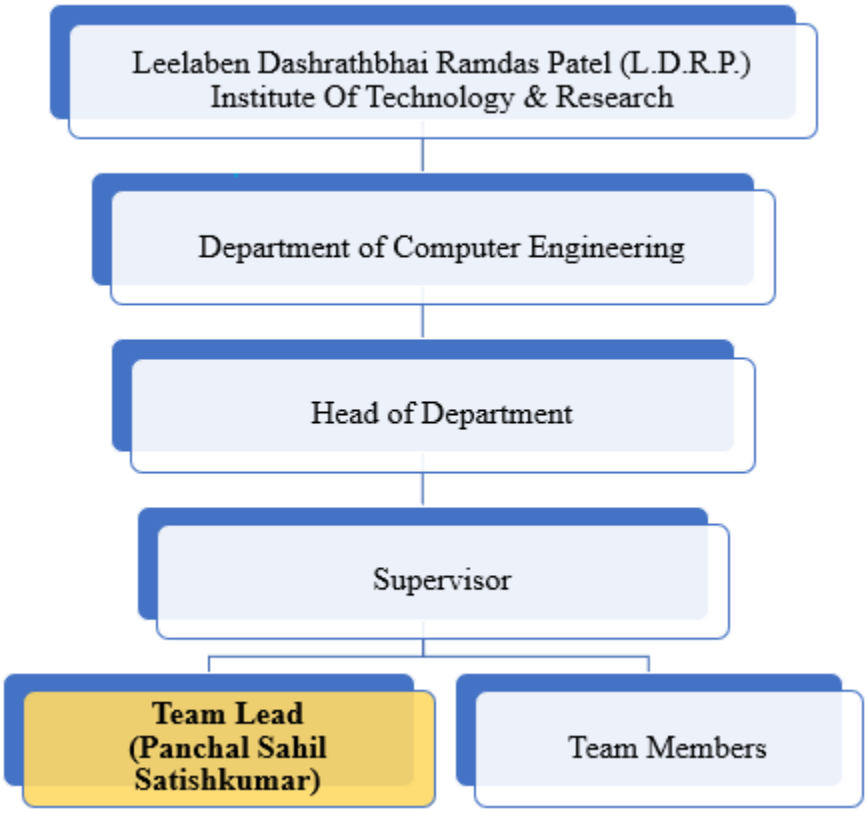


Fig1: Organogram

2.2.4 TASKS

- To examine the issue of electronic spam and explore the efficacy of filtering algorithms.
- To select appropriate emails, utilize the KNN algorithm, and evaluate the filter's performance using varied datasets.
- To govern the optimal value for k in KNN, adjust dataset parameters, and perform a 10-fold cross-validation.
- To review experimental outcomes, test the update component, and integrate a feedback mechanism.
- To optimize feedback rates, consolidate findings, and simulate a real-time spam filtering process, ensuring the enhanced TPR outcome.

2.3 PEAs

2.3.1

I examined the pervasive problem of spam in electronic communication, recognizing its effects on both particular users and more extensive systems. When I looked into several ways to address this problem, I noticed that filters which serve as barriers to stop or reduce the number of unsolicited emails and messages that make it to an inbox was particularly appealing. I researched the k-Nearest

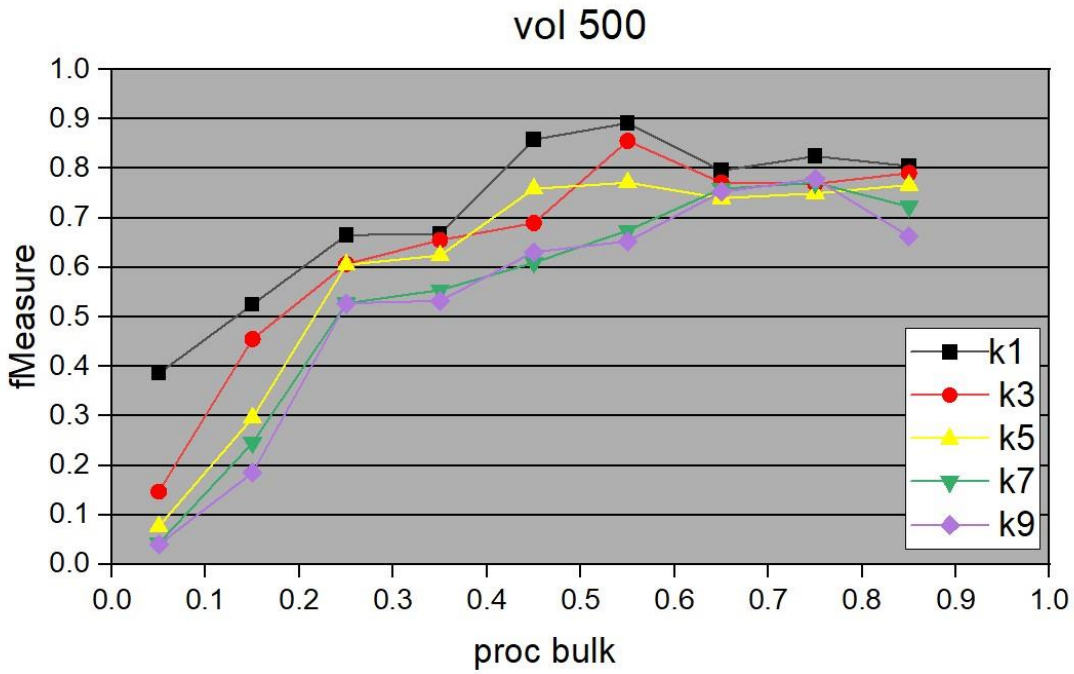
Neighbors (KNN) algorithm, a crucial machine learning method frequently used to categorize emails and other data based on resemblance to existing samples. I understood that this technique was effective against spam because it considered how close new data points are to current data points. In order to maintain a balanced dataset for training, I realized how important resampling approaches were. I found that this in-depth analysis improved my expertise in spam detection and prevention.

2.3.2

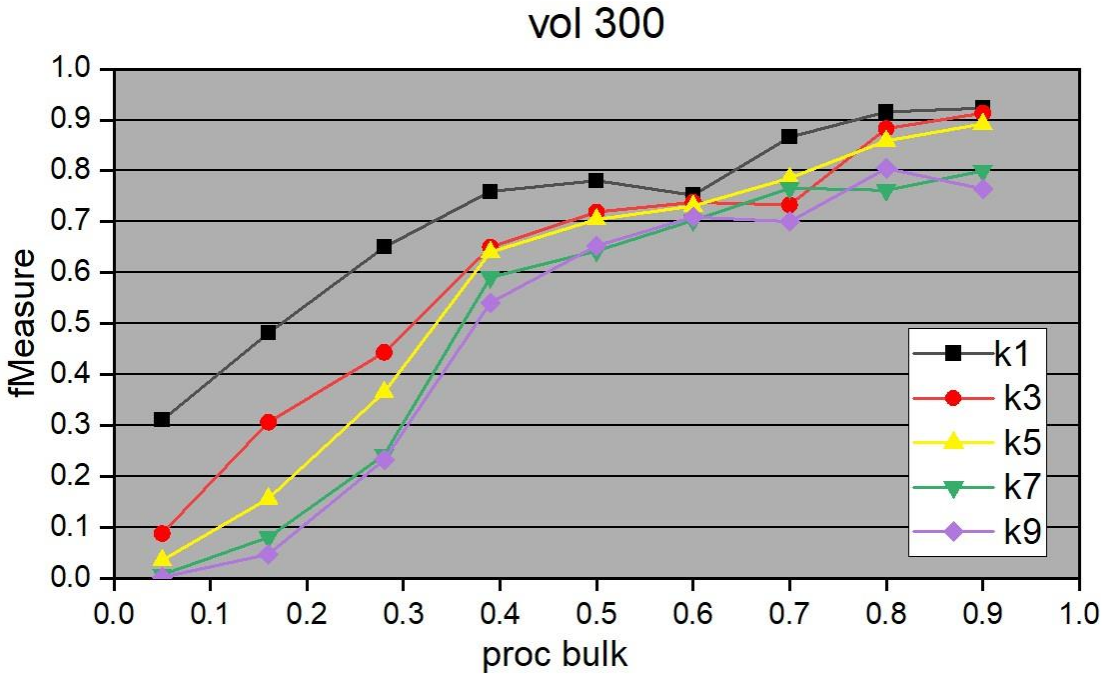
I began with the selection of emails as well as essential algorithms to gather data for analysis. I selected I chose the k-Nearest Neighbors (KNN) algorithm primarily because it didn't need any training step, making the apprising of the common words swift. I chose the spam detection filters and feedback mechanisms. I chose to measure a few parameters such as TPR F-measure, and pf. I also selected update rates to make sure the system remained updated by adapting to new emails or new possible scams. I selected its capability to include an adaptive threshold, which determined that a message could be deemed legitimate even if its likelihood of being spam exceeded 50%. I found this feature especially beneficial when prioritizing filter sensitivity. I decided to execute multiple assessments of the filter's effectiveness on both static and dynamic datasets. I further opted to execute evaluation on a dataset comprising 1000 instances, from which I picked the necessary samples. I made use of 500 spam as well as 500 legitimate emails.

2.3.3

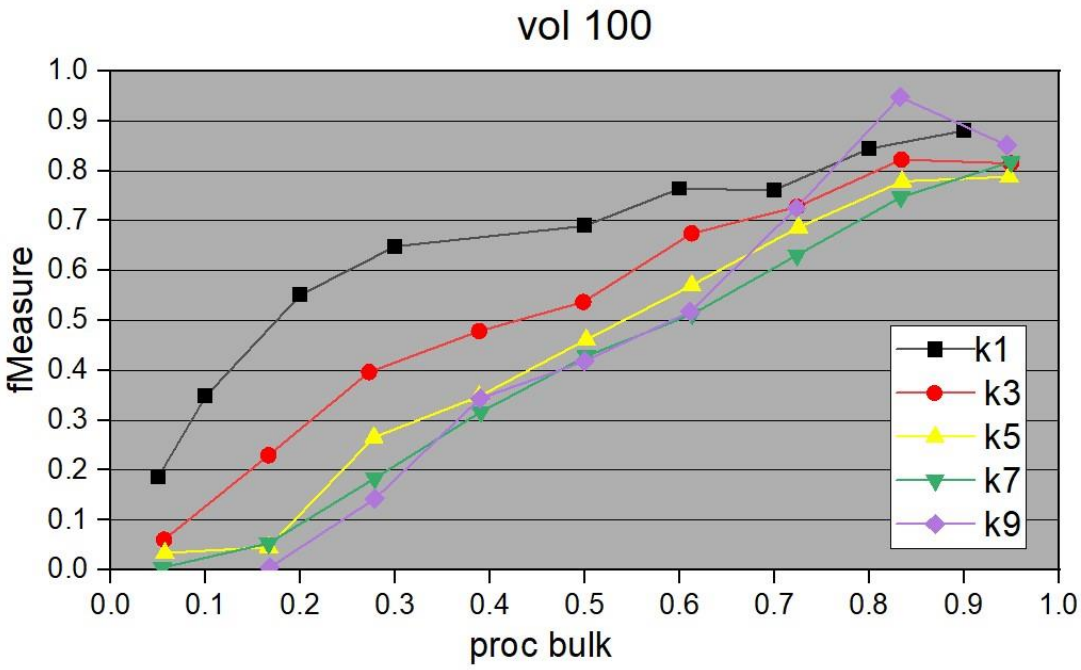
I conducted the initial experiment to pinpoint the utmost suitable k value in the KNN algorithm. I varied the legitimate bulk percentage amid 0.1 and 0.9 for email volumes ranging from 100 to 500, and generally, the curve for k=1 provided the finest fit. I then proceeded to determine the ideal dataset size and the distribution of the optimistic class in emails. I adjusted the dataset size from 100 to 500 emails then altered the percentage of legitimate emails between 0.5 and 0.9. For every combination of dataset size and optimistic class distribution, I performed a cross-validation of 10-fold with KNN. I favored this method over recurrent random sub-sampling since it ensured every observation was utilized for both training as well as validation, with individual observation validated precisely once. I employed stratified validation to keep consistent proportions of both class labels in every fold. The figures I analyzed were the average outcomes.


vol 500
fMeasure
proc bulk
k1
k3
k5
k7
k9

(a)


vol 300
fMeasure
proc bulk
k1
k3
k5
k7
k9

(b)



(c)

Fig2: (a)(b)(c) Determination of k value

2.3.4

After the experiment, I reviewed the outcomes that were achieved from the experimentation. I reviewed the F-measure evolution as the distribution function of data set for every size. I utilized 5 data sets with the size ranging from 100, 200, 300, 400, and 500. I noted that the F -measure of data set with size 200 to 400 displayed similar outcomes with the optimum value for data set with size 500. I also witnessed that as the distribution increased, the F-measure also increased. In the subsequent phase, I experimented the update component. I assessed the filter using a test set comprising 100 emails with a 0.5 distribution, as well as a training set of varying sizes (from 100 to 500) and distributions (from 0.5 to 0.9). I observed that the update component underperformed in this context compared to the process without retraining. However, I noted that the two outcomes weren't entirely comparable because the initial tests were executed in conjunction with a cross-validation of 10-fold, whereas the latter was based on a secure test set. I also stated that a misclassified message once added to the training set, could perpetuate the error, affecting the classification of subsequent messages. This led me to consider enhancing the filter by incorporating a feedback mechanism to rectify these mistakes. I determined that this mechanism required user intervention and judgment. To simulate this feedback, I revisited the previous evaluation, and after classifying each email, I adjusted the prediction before integrating the new instance into the training set.

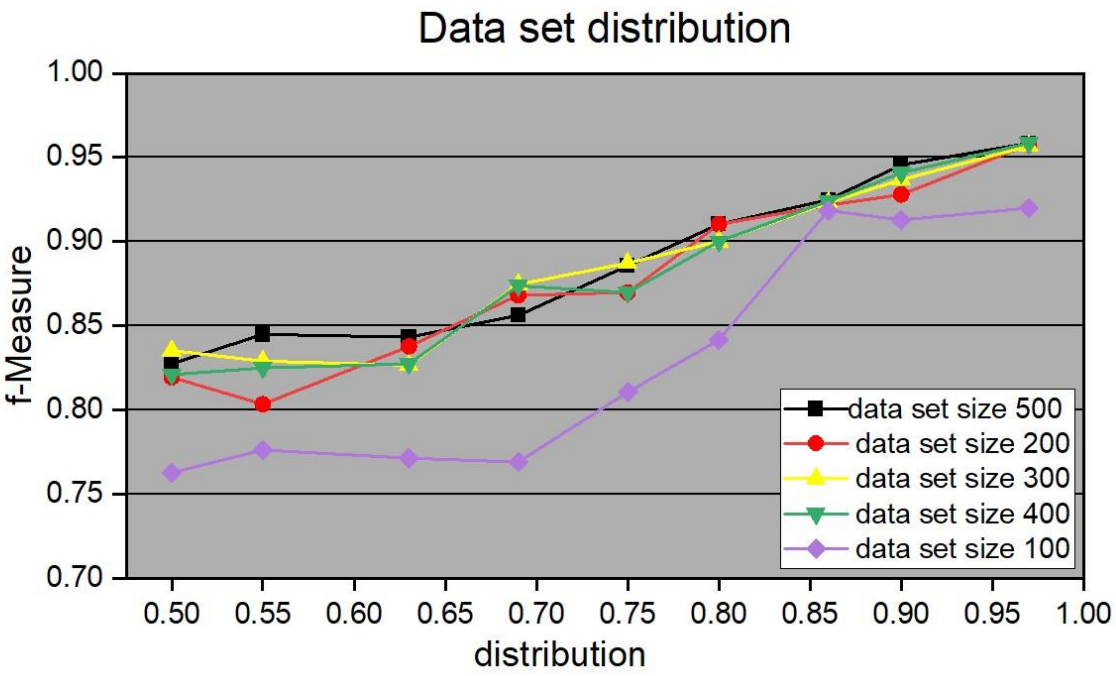


Fig3: F-measure after a 10-fold cross validation with KNN

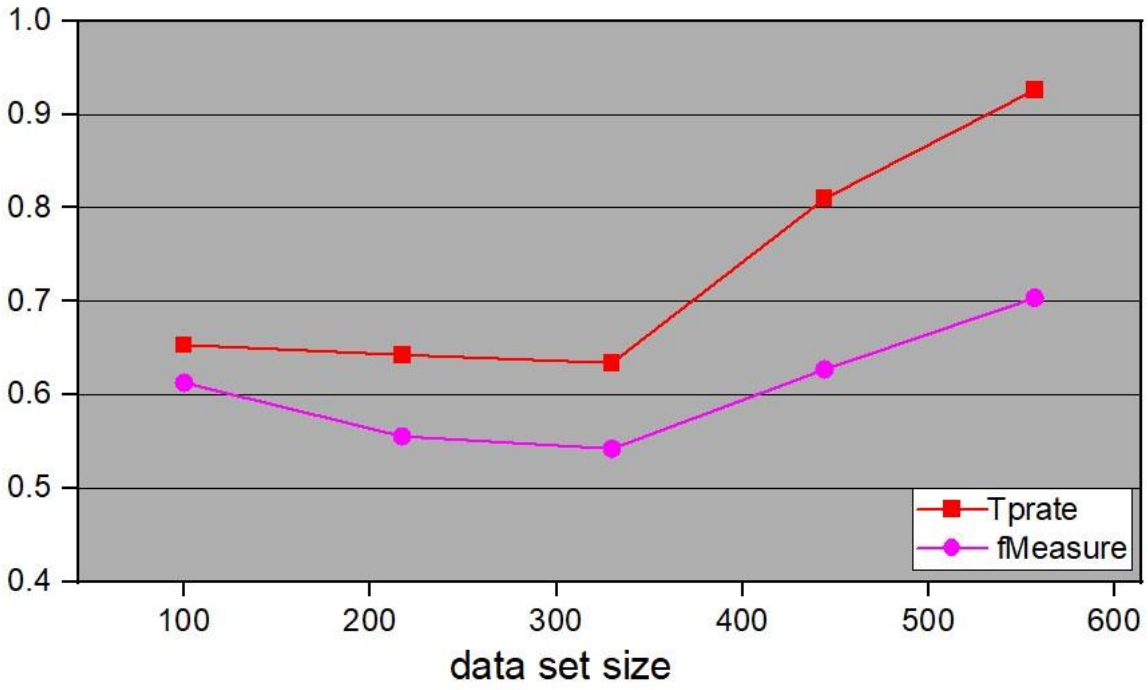
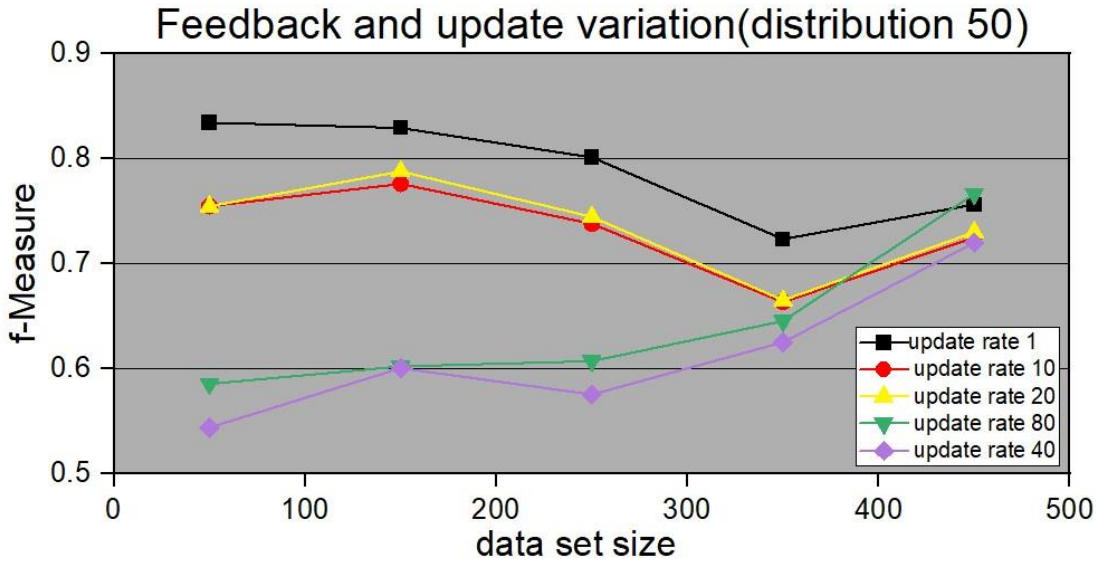


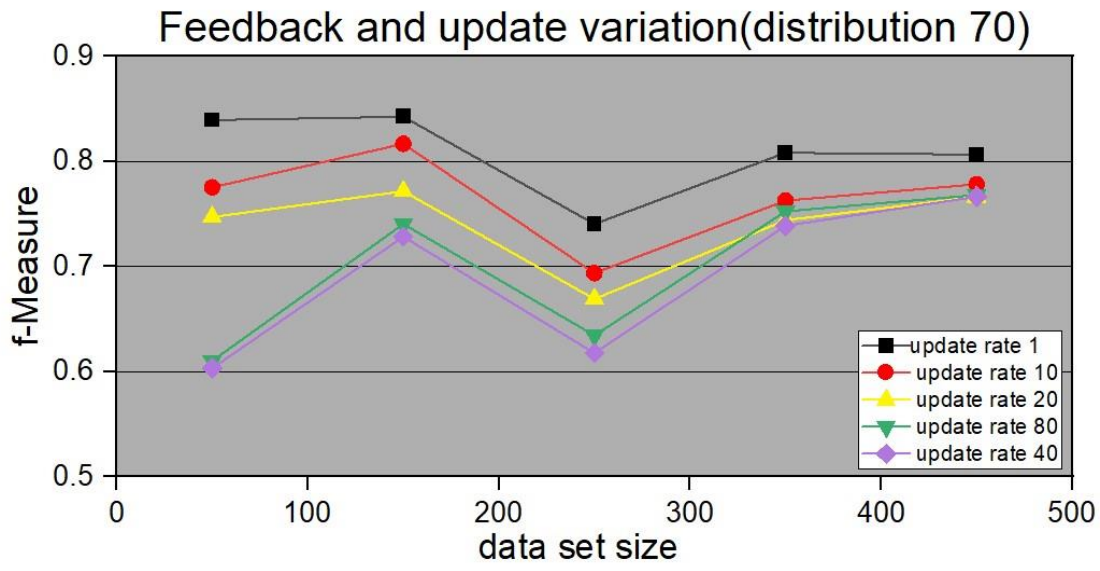
Fig4: Performance pf Simple update component

2.3.5

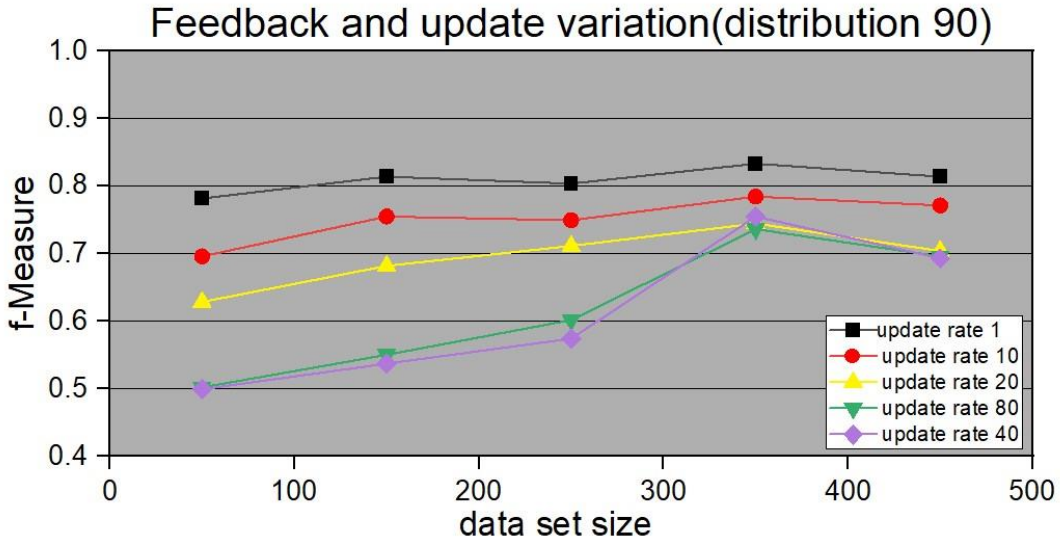
I recognized that real-time feedback after each email might be impractical, given it relies on user actions. As a result, I experimented with various feedback rates (1, 10, 20, 40, 80) to pinpoint the most effective frequency for updating across different dataset sizes. I found that the optimal performance was achieved at a rate of 1, but it gradually declined as the rate increased. In our scenario, a rate of 20 appeared to be the most suitable for feedback, since it yielded nearly the same results as rate=10 but was more cost-efficient. For my final assessment, I amalgamated all the optimal values determined previously (k=1, 500 resample size, 0.9 resample distribution, 20 feedback/update rate) to emulate a real-time spam filtering procedure. I reviewed and categorized each email, recording its characteristics. After every 20-email batch, I updated these features, reconstructing and resampling the training set to 500 emails with a distribution 0.9 for the optimistic class. I concluded that the final outcome displayed the enhancement for TPR but not the F-measure.



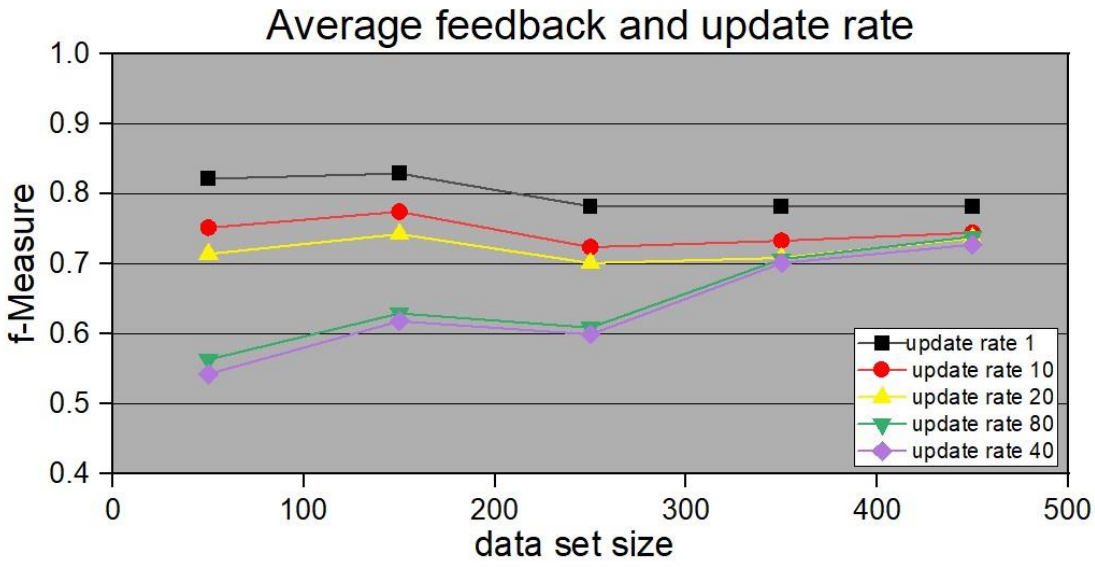
(a)


Feedback and update variation(distribution 70)
f-Measure
data set size
0.9
0.8
0.7
0.6
0.5
0
100
200
300
400
500
update rate 1
update rate 10
update rate 20
update rate 80
update rate 40

(b)


Feedback and update variation(distribution 90)
f-Measure
data set size
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
100
200
300
400
500
update rate 1
update rate 10
update rate 20
update rate 80
update rate 40

(c)



(d)

Fig5: (a)(b)(c)(d) Different feedback rates analysis

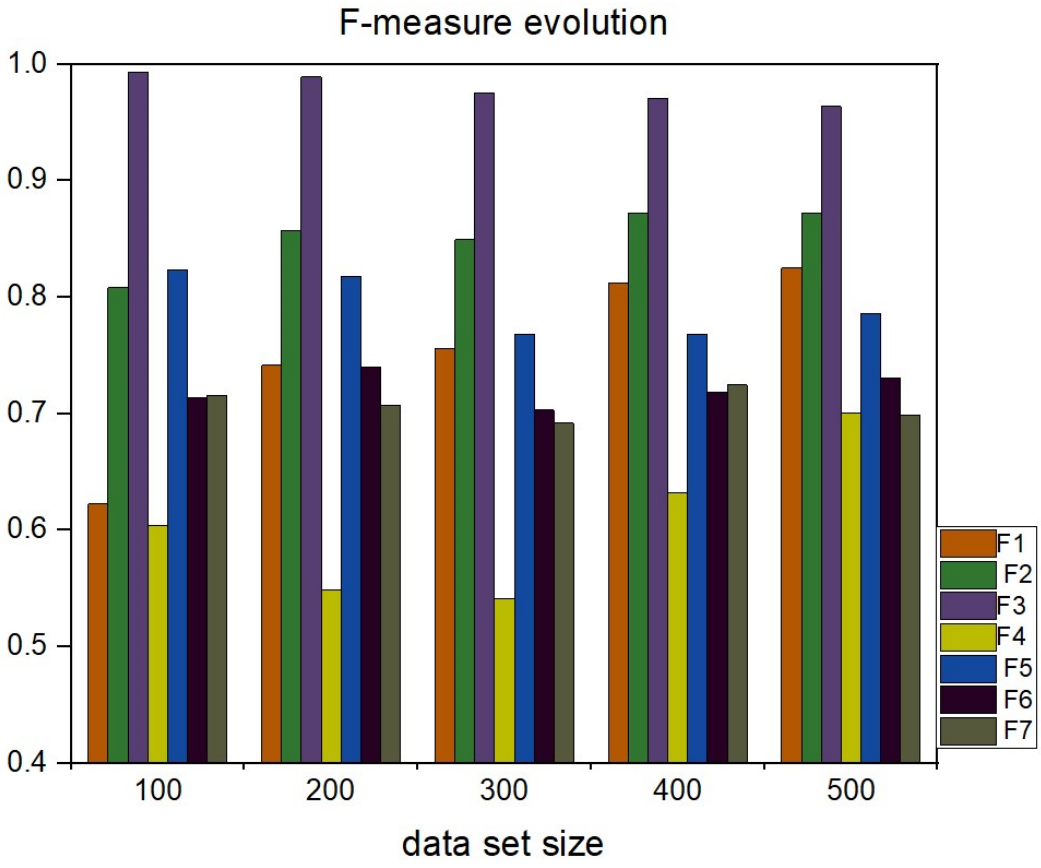


Fig6: Performance evolution for F-measure

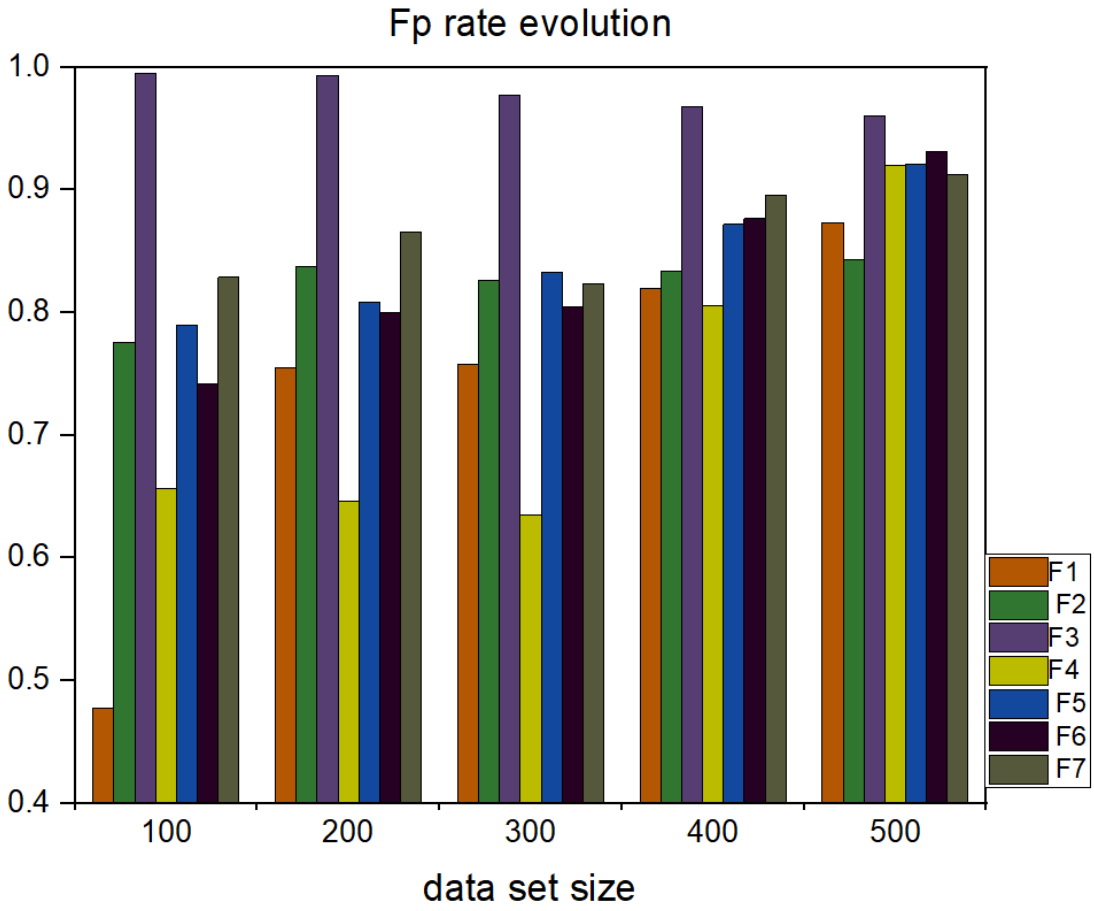


Fig7: Performance evolution of TPrate

2.4 ERROR AND RESOLUTIONS

Throughout the project, I ran into a scalability issue with the network, which was a significant technological hurdle for me. I discovered the problem when the system started to slow down and became noticeably less responsive. I noticed that the system's overall performance was greatly harmed by the inability to immediately recognize and classify spam emails. To address this issue, I started a number of in-depth investigations to identify its underlying causes. I learned a lot from the team members and subject matter experts over the numerous meetings and discussions we had. I asked our supervisor for guidance so that I could better understand the situation. I thoroughly considered the situation and came to the conclusion that the reason for the sluggish performance was the large volume of incoming emails that needed to be processed. I deliberately chose to address the issue by limiting the dataset. I made the decision to focus on just 1,000 emails for the survey as opposed to the entire collection of data. By reducing the number of emails, the system had to handle, I was able to restore its previous responsiveness and speed. As a result, the system regained its speed and responsiveness and was able to classify spam emails with astonishing precision. I made sure that this tactical choice significantly increased the system's capacity for scaling, ensuring that it would continue to operate smoothly even in the face of high email loads.

2.5 CREATIVE WORK

I cleverly created adaptive algorithms in the project that dynamically adjusted to email patterns in order to increase the precision of spam identification. To facilitate user interactions, I also developed a straightforward user interface. I prevented the email overflow slowness by limiting the dataset to 1000 emails, which immediately improved system responsiveness and speed.

2.6 TEAM MANAGEMENT

Upon adopting the position of team head, I immediately took the helm of operations and task allocation. Duties were systematically assigned to team members, ensuring each task matched their expertise. Through consistent meetings and discussions with both the supervisor and the team, I tackled arising challenges and brainstormed solutions collectively. Keeping track of everyone's progress was essential, equipping them with the necessary resources and direction for effective results. As the project approached its conclusion, I proactively compiled a detailed report and forwarded it to the relevant department and supervisor for evaluation. My leadership approach was marked by decisive actions and meticulous attention, both of which were instrumental in the project's smooth realization.

2.7 CODES

I heeded the rules and regulations set by the college while preparing the project. I made sure all the codes of conduct were followed by me and the team.

2.8 SUMMARY

2.8.1

The scheme sought to explore the application of the KNN algorithm in addressing electronic spam. It probed the challenges posed by unwanted emails and examined various filtering techniques. The k-Nearest Neighbors (KNN) algorithm stood out due to its ease of implementation without training and its adaptive threshold capabilities. Assessment was done on datasets of up to 1,000 emails, maintaining a balance of spam and genuine messages. Critical to the research was the identification of the best k value for the KNN and the right dataset size. Stratified validation was employed to ensure consistency in class labels. The findings highlighted the need for regular feedback, ideally after every 20 emails. Integrating this feedback improved real-time spam filtering. While there was a notable increase in the True Positive Rate, the F-measure remained consistent. The targeted aims of removing spam efficiently were thus attained on time. The spam was efficiently reduced and removed from the incoming emails and higher feedback rates were achieved.

2.8.2

The venture helped me to understand spam, its detection as well as the filtering algorithm. My skill to lead and handle the team enhanced. I also became proficient in resolving the issues in a short time with group discussion and detailed study.