THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

COLLEGE OF INFORMATION SCIENCES AND TECHNOLOGY

Harnessing AI Techniques to Improve Accessibility of Healthcare for Pregnant Women in Kenya

PRERNA RANGANATHAN
SPRING 2022

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Computer Science
with honors in Information Sciences and Technology

Reviewed and approved* by the following:

Amulya Yadav
PNC Technologies Career Development Assistant Professor
Thesis Supervisor

Marc Friedenberg
Assistant Teaching Professor of Information Sciences and Technology
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

Through the use of artificial intelligence (AI) and natural language processing (NLP), an advanced health-agent system, Promoting Mums through Pregnancy & Postpartum Through SMS (PROMPTS), has been deployed in Kenya to help pregnant mothers, or mums, receive the help they need. Unfortunately, the health care system in Kenya is poor, which leads to challenges in mums getting the necessary care and results in many maternal and neonatal deaths. While the PROMPTS platform has helped to improve the health care that the mums are receiving, there is potential for it to be optimized further by considering the current challenges that the system faces. This paper focuses on the use of traditional classification models to improve classification performance. By analyzing three classification models, Adaptive Boosting (AdaBoost), Random Forest (RF), and k-Nearest Neighbors (k-NN), this work aims to create a new model that combines the three classifiers in an optimal manner. The models discussed in this paper are evaluated using three main performance metrics: precision, recall, and F1 score. With a model that effectively and accurately classifies information, PROMPTS will have a better overall performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Amulya Yadav for supervising my thesis and providing me with his guidance and support to partake in research.

I would like to thank Marc Friedenberg for supporting me as my thesis honors advisor and reviewing my thesis.

I would like to thank my research group, Wenbo Zhang, Hangzhi Guo, Nidhi Danayak, and Manan Gupta, for working on this project with me to make a real impact in the world.

Chapter 1

Introduction

Kenya lacks a high quality health care system, resulting in delays in care-seeking as well as inaccurate diagnoses. This limitation makes it hard for people to get the help they need in a timely manner. Women are often not able to receive the appropriate care during their pregnancies [1]. Many maternal and neonatal deaths occur in Kenya, with 6,300 women dying annually during pregnancy and childbirth as of 2015, and the maternal mortality ratio estimated to be 342 per 100,000 live births as of 2017 [1, 2]. In 2020, the infant and neonatal mortality rates were 31 and 20 per 1000 live births, respectively [3]. Approximately two-thirds of neonatal deaths and 80% of maternal deaths are due to delays in care-seeking [4]. These women do not have the information they need to make informed decisions about their pregnancy care, evidenced by the fact that as little as 8% of poor women in Kenya have access to minimal quality maternal health services [5]. Kenya failed to meet the 2015 Millennium Development Goals for maternal and child mortality rates, and it remains an important issue [2].

Jacaranda Health is an organization that aims to stop preventable deaths by developing affordable and sustainable solutions to improve the maternal care quality that these mums are receiving. The organization partners with government health systems to deliver and deploy these technology-based solutions to over 930 hospitals and health centers through working with the National Ministry of Health and 20 Kenyan County Governments [6]. Through better care and improved access to it, 80% of maternal deaths that happen in facilities as well as a large portion

of newborn deaths in Kenya can be prevented [7]. As a result, 50% of all maternal deaths and 1 million neonatal deaths can be avoided [8].

An advanced health-agent system, PROMPTS, developed by Jacaranda Health is widely used among pregnant women in Kenya with over 1 million users [9]. PROMPTS is a "digital health platform connecting mothers with lifesaving advice and referral to care" [10]. It enables mums to use an SMS messaging system to seek required online advice and doctor assistance at no cost to them and for a lifetime cost of only 0.74 USD per mother to those funding the platform [11]. Once a pregnant woman registers for PROMPTS, she will start receiving "prompts" through text messages providing tips and behaviors based on her current stage of pregnancy. The mum can also use the service to ask questions regarding her pregnancy by sending a text message to a trained Helpdesk agent, which will then decide the case emergency level based on the text. Simple questions that are not high risk can be answered by the Helpdesk service directly. If the case exhibits a potential danger sign that could mean it is high risk and needs to be urgently addressed, then an AI-based triage system will kick in. The system will use AI to assign a degree of urgency to the situation so that the mum can be screened by the Helpdesk agent. A "Triage Bot" categorizes the intent of various user questions using NLP. If the screening determines that the mum has an urgent issue that needs care, then the Helpdesk agent will refer her to the appropriate facility and share the required information with the healthcare providers in a digital file so that they can assess the situation and act immediately. The pregnant woman will successfully receive care through this process that uses AI and a Helpdesk agent in a much faster and more effective manner.

The PROMPTS platform leads to improved maternal outcomes and safer mothers and children. The service has already "driven key behavior changes such as improved breastfeeding,

family planning, and infant vaccination, as well as improve referral and triage of urgent cases, connecting mums and their babies facing life-threatening clinical conditions to the closest, well-equipped facility" [10].

While the PROMPTS service has been deployed by Jacaranda Health and is greatly improving the quality of care for pregnant women in Kenya, there are some challenges that need to be considered to optimize the system further. First, the text messages are in a code-mixed format, but the existing translation tools do not work well. By more effectively translating the messages, the emergency level prediction accuracy of the model could potentially be improved. The second challenge is how to improve the system so that the messages can be explained and evaluated correctly. Currently, the system is only looking at the latest, most recent incoming text message to classify the risk level, but it is important to consider how the categorization could be bettered to further improve the prediction accuracy. This paper will focus on the second challenge by looking at various traditional classification models that can be used in the system.

Based on these considerations, the research goal of the work is to propose an NLP-based system to process the code-mixed text messages and take into consideration the history of text messages from patients in order to improve the prediction accuracy on the emergency level of patients' health issues. The plan to achieve this goal can be broken down into three parts: transform the emergency level prediction problem into a multi-class text classification problem and optimize the classification performance, apply NLP techniques to propose new models or algorithms to deal with the code-mixed text in Swahili and English automatically, and further improve the prediction accuracy through considering the whole medical history of patients. Jacaranda Health's PROMPTS service has already shown promising results of improving the quality of care for pregnant women in Kenya. This paper will look at the first part of the goal to

further optimize the PROMPTS platform in order to improve the quality of care even more and give the mums the help they need.

Chapter 2

Related Work

The use of an SMS messaging system to improve the quality of maternal health care has been studied in prior work. This chapter will discuss the various perspectives through which the topic has been investigated.

Support Network

One aspect that prior work has focused on is how to improve communication between pregnant women and healthcare workers. Often, weak communication links can cause mums to not receive the care they need in a timely and effective manner. As a result, Khanum, De Souza, et al. [12] designed a pregnancy care network based on smartphone devices and communication systems. The network relies on an online server, which stores medical information uploaded by healthcare professionals and pregnant women. The healthcare workers and mums can use a web user interface and client application, respectively, to access and view, as well as upload, the data. Through this setup, pregnant women and healthcare staff will be better connected and have stronger communication. The system also educates women about and monitors pregnancy. Another paper explored the use of video on phones for communication between midwives and pregnant women in rural India to observe whether mobile phones would have a positive response and impact on the relationship between the two groups [13]. While these papers demonstrate the importance of strong communication links between healthcare workers and mums as well as the usefulness of mobile phones in creating these links, which are important conclusions for the work in this paper, the systems in such prior work still require manual work to be done and there

are no automated parts to the process. With an increasing demand for healthcare, healthcare professionals are not equipped to properly care for all the mums. Therefore, unlike these previous papers, this work will analyze the use of automation in the PROMPTS system to provide mums with the maternal care they need in a more time-efficient manner.

AI for Communication and Information Dissemination

A lot of work has been done on how AI can be used to talk to pregnant mothers and convey important information about pregnancy to them. For example, "MumCare" is an AI assistant that provides case-by-case services [14]. The assistant gives pregnant women easy access to maternal health-related information, sends daily reminders based on individual needs, allows women to make emergency calls, and includes a virtual chatbot that women can communicate with to feel like they are talking to their unborn child in order to improve their mental health. According to [15]-[17], Projecting Health, a public health project in rural India, works to spread maternal and neonatal pregnancy information to mums through mobile phones and videos using a community approach. While the findings in these works are helpful in distributing information that could help mums avoid pregnancy issues, they do not explore the possibility of women reaching out for help on an individual basis and being directed to the right resources. The MumCare application has a virtual chatbot, but it does not serve this purpose. Instead, it is only meant to give the illusion of talking to an unborn child. The work in this paper examines the idea of pregnant women asking questions and receiving the necessary information and care.

Automated Message Triage

The closest prior research found to the work being presented in this paper uses an automated message triage. Engelhard, Copley et al. [18] focus on using automation with a helpdesk to improve the quality of maternal health care in South Africa. Their work attempts to improve responsiveness to high-priority messages by automating the process of assigning labels to incoming messages similar to the PROMPTS system in this paper. Their paper, however, examines violence against and mistreatment of women as the main high-priority label that needs to be predicted correctly. The discussion of this research will expand and focus more broadly on the classification of incoming messages into several different intents. Additionally, Engelhard, Copley et al. analyze the use of the naïve Bayes classifier for labeling incoming messages, while this paper will study other classifiers. Although the overall ideas of the two papers are similar, the focus of this work is slightly different.

Chapter 3

Dataset

The dataset contains 107,714 total data points. An individual incoming text message from a patient is considered as one data point, and each text message is in Swahili or English. An 80% train-20% test split was used to separate the dataset into train and test data, so 86,171 data points were used for training while 21,543 data points were used for testing. Although the patients' phone numbers are provided in the dataset, they are not included in this analysis since only the current incoming text message is being considered at this point instead of the history of the patients' texts.

Text feature extraction had to be done on the input of text messages before the traditional classification models could be tested on the data. Two different methods were tried for text feature extraction: word frequency and Bidirectional Encoder Representations from Transformers (BERT). For the first technique based on word frequency, the input of text messages first needed to be preprocessed. The method used for preprocessing was the Snowball Stemmer algorithm, which processes words and simplifies them to the base word, otherwise known as the stem [19]. By reducing words to their stems, words that are similar and come from the same base word will be under one overarching stem. The text processing method of regular expressions was also tried, but the stemmer yielded better results. After preprocessing using the stemmer was complete, the term-frequency inverse document-frequency (TF-idf) method, which does both tokenization and occurrence counting was used for feature extraction [20]. The second method based on BERT implemented the pretrained BERT model, 'bert-based-uncased,' which has 12-layer, 768-hidden, 12-heads, 110M parameters and is trained on lowercase English [21]. The BERT model was tried with a maximum length of 100 and a batch size of 256. Text feature

extraction based on BERT performed better, so this method was chosen in the end. After text feature extraction, the input was ready to be fed into the traditional classification models.

To classify the data points into categories, 58 intents were used when testing the traditional classification models. The intents describe the topic of an incoming text, and each intent is associated with a risk level that allows the PROMPTS system to determine how urgent a patient's case is. The largest intent is "pregnancy_general," which contains 13,557 data points and holds all the data points related to pregnancy that do not fit under the other more specific pregnancy-related categories. There are 60 intents used in the system in total, but 2 of them have very few data points and have therefore been omitted from the evaluation in this paper. Each of the 58 intents is associated with a label between 0 and 57 using a label encoder, and each text message in the dataset is associated with a number as its true intent label. Figures 1 and 2 show the distribution by intent of training and testing data points, respectively. The two different sets have the same approximate distribution. Appendix A shows the relation between the intents and labels. Using the processed input of text messages and the intent labels represented as numbers, the traditional classification models were run on the data and each data point in the test data was assigned a number as the predicted intent label so that further analysis could be done by comparing the true and predicted labels.

Figure 1: Intent Distribution of Training Data Points

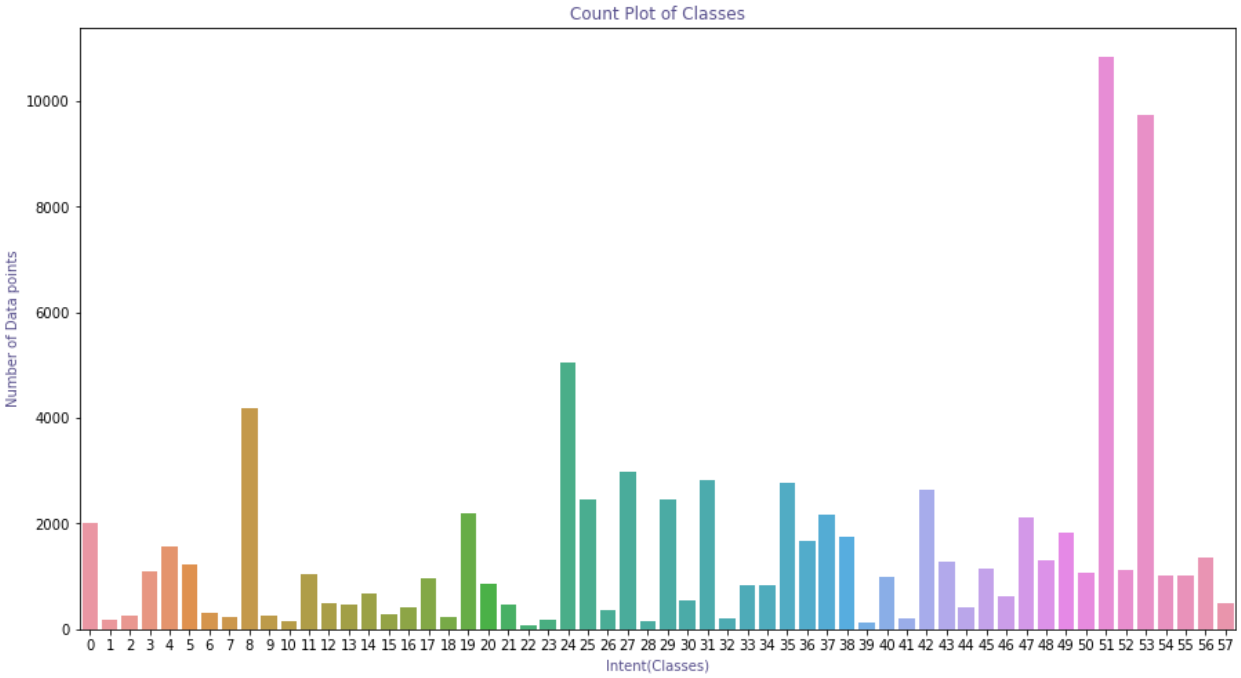
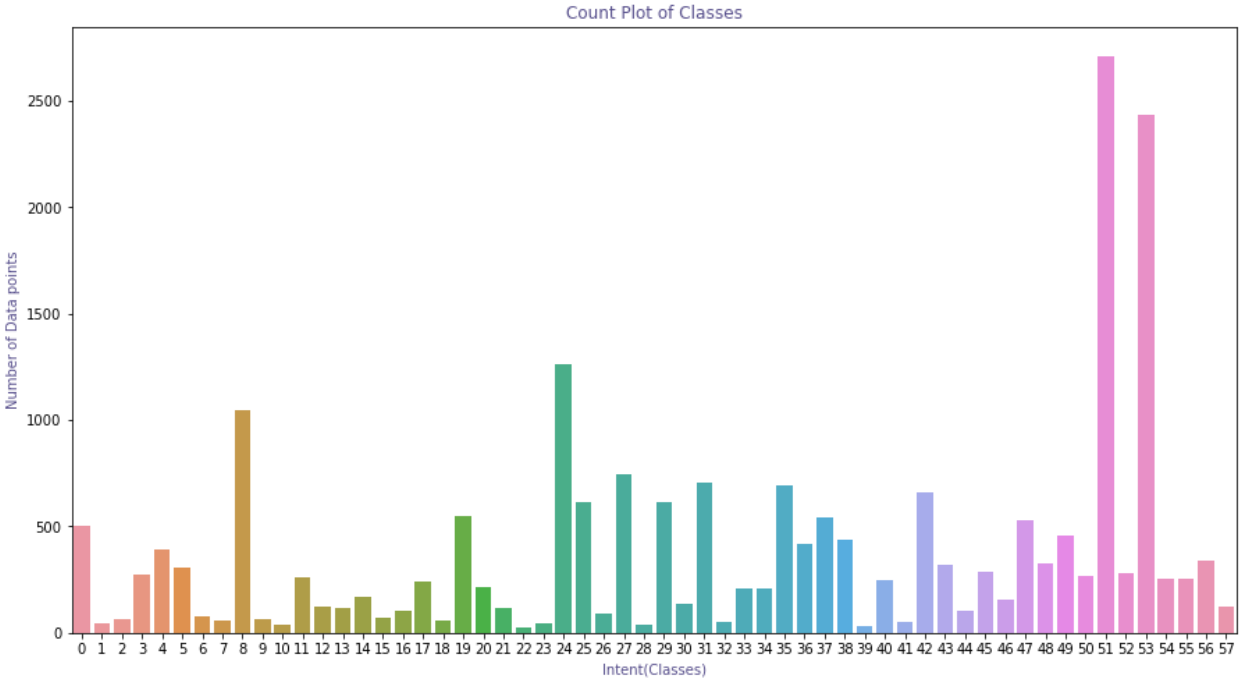


Figure 2: Intent Distribution of Testing Data Points



Chapter 4

Methods

Three traditional classification models were run on the dataset: AdaBoost, RF, and k-NN. Each classifier was tested with its default parameters to observe which intents it performed best on so that a new model could be created by optimally combining the three models.

For each classifier, an intent-level confusion matrix was constructed, as well as graphs for the precision, recall, and F1-scores. The intent-level confusion matrices for AdaBoost, RF, and k-NN are shown in Appendix B, Appendix C, and Appendix D, respectively. The main diagonal in each confusion matrix displays the accurate results (true positives and true negatives), while the rest of the confusion matrix shows the inaccurate results, where the rows show false positives and the columns show false negatives. The matrices are shown as heat maps for easy visualization of the results through the use of a color range. The lighter colors show where a larger portion of the data points fall, so the goal is to have more lighter colors along the main diagonal of the matrices. The numbers in each cell show the logarithm of the value calculated for that point in the confusion matrix. The logarithm was taken to help with visualization by making the colors more distinct since it allowed the colors to have a smaller range of values. The results for each model individually will be discussed in detail in each subsection followed by an overall comparison of the three models.

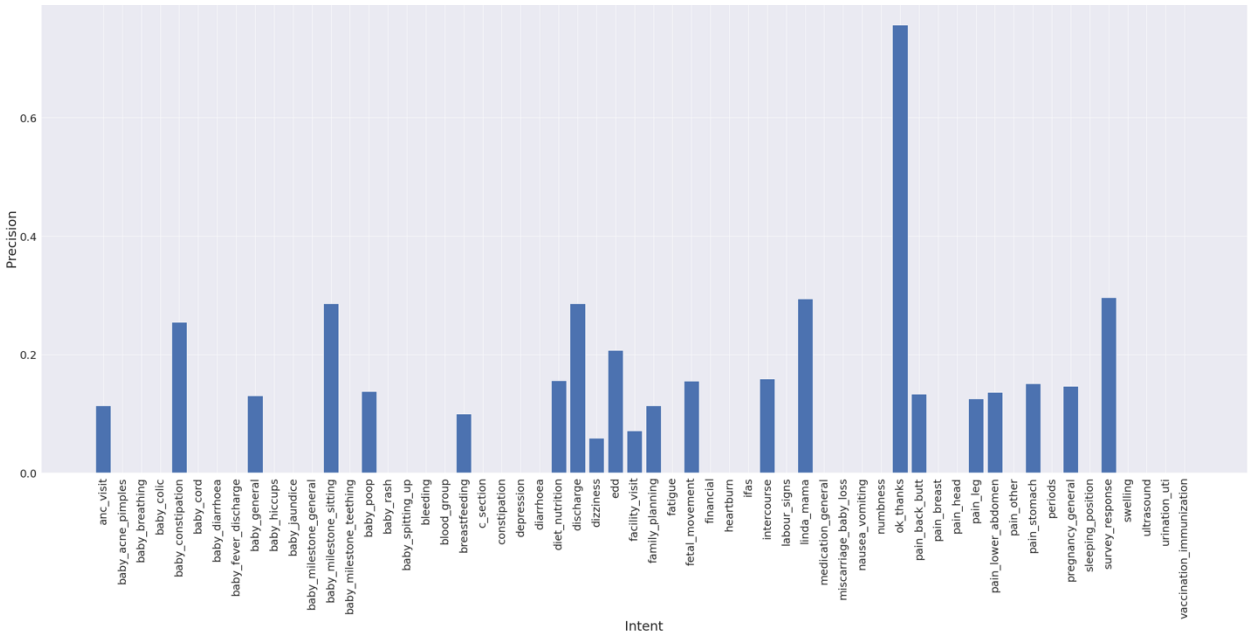
Adaptive Boosting

AdaBoost is a boosting technique, which works by converting weak learners into strong learners. The classifier creates decision trees called stumps, where each stump is a node with

only two leaves. During the training stage, the algorithm classifies the data points based on an initial model. The data points that are misclassified are assigned higher weights for the next iteration so that they have a higher classification probability. Another model will be created and tested, and once again, the misclassified points will be given a higher weight for the next model. This training process will continue until a final, stronger model is made from a combination of the weaker, individual models based on the results from each iteration. By passing through all the stumps that were created during training, the test data can then be classified based on the majority of the votes from all of the stumps [22].

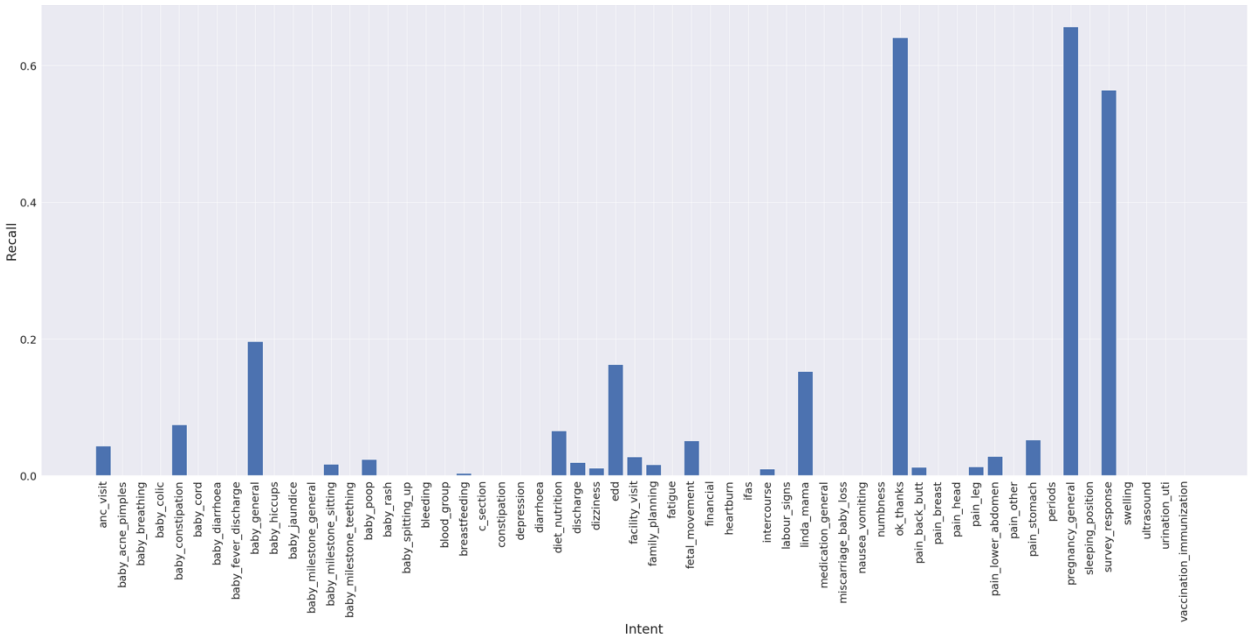
The precision for each intent is displayed in Figure 3. Precision is the ratio of the number of predicted true positives to the total number of predicted positives (both true and false) [23]. As can be seen in the graph, the precision for many of the intents is low, with many intents having a precision of 0.0 and the lowest nonzero precision being 0.058823529 for the “dizziness” intent. The intent “ok_thanks” has the highest precision with a value of 0.756708408, but this is the only intent that has a precision above 0.50. In fact, the precisions of all the other intents are below 0.30. The macro average for precision is 0.073576464, and the weighted average is 0.148979851. Even though the weighted average is a little over double the macro average, meaning that the precision performed better for intents that are more prevalent in the data, the average is still low for AdaBoost.

Figure 3: Precision for AdaBoost



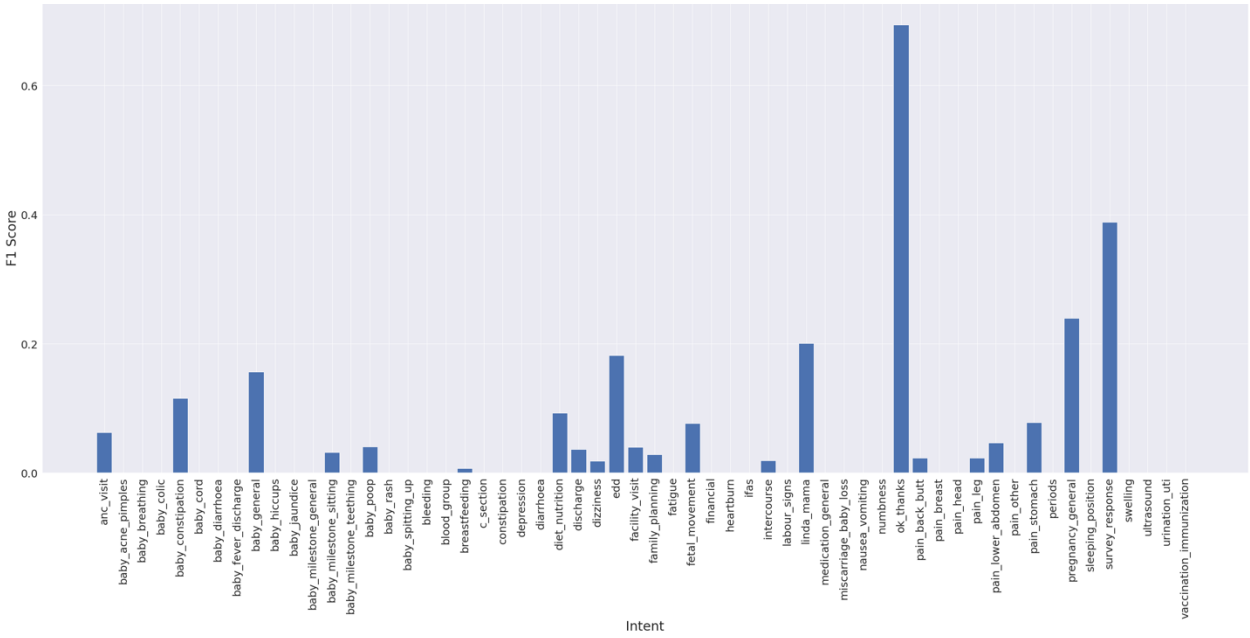
The recall for each intent using the AdaBoost classifier is shown in Figure 4. Recall is the ratio of the number of predicted true positives to the total number of actual positives (true positives and false negatives) [23]. Similar to the precision scores, many of the recall scores for AdaBoost are 0.0 or close to 0.0, with the lowest nonzero recall being 0.003656307 for "breastfeeding." There are a couple more recall scores above 0.50 than precision scores, with "pregnancy_general" having the highest recall score of 0.656710914 and "ok_thanks" being a close second with a recall score of 0.640909091. Other than the three intents with a recall above 0.50, however, the rest of the intents have recalls below 0.20. The macro average for recall is 0.049009902, and the weighted average is 0.19686209. Once again, the weighted average is higher than the macro average (almost 5 times greater), but the average is still low overall like the average for precision.

Figure 4: Recall for AdaBoost



The F1 scores for each intent are summarized in Figure 5. The F1 score is the weighted average of precision and recall [23]. Since the precision and recall scores for many of the intents are low, the F1 scores are also low for many of the intents with several of the scores once again being 0.0. The lowest nonzero F1 score is for “breastfeeding” with a score of 0.007054674. Since the “ok_thanks” intent has both high precision and recall scores, its F1 score is also high at 0.694011485, which is the highest F1 score across all the intents. Apart from this intent, the rest of the intents have F1 scores below 0.40. The macro average for F1 scores is 0.044836835, and the weighted average is 0.132315396. These averages follow the same trend as the precision and recall scores.

Figure 5: F1 Score for AdaBoost



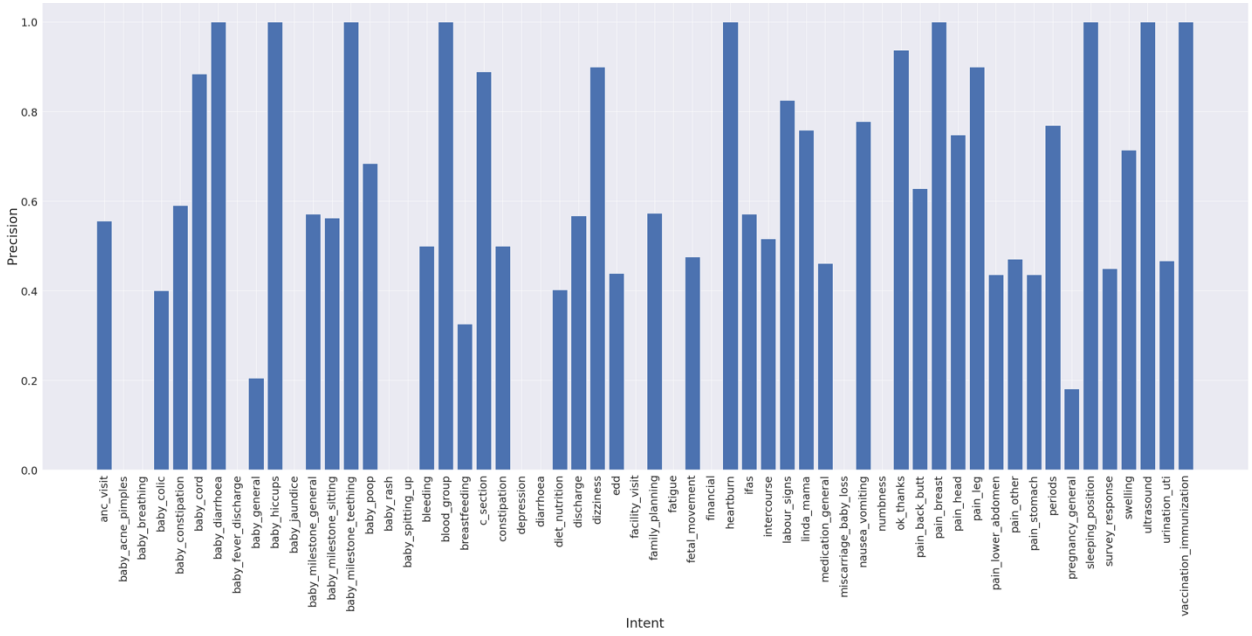
Overall, the AdaBoost classifier does not perform well in terms of precision, recall, and F1 score. All three metrics have low scores for the most part, signifying that this model is likely not the best fit to achieve the goal. The next traditional classification model that was tested was the RF classifier.

Random Forest

The RF classifier creates multiple decision trees based on subsets of the training data. Unlike AdaBoost, the trees in this model can have more than two leaves. Data points from the training set are selected randomly as subsets, and decision trees are created based on the data points. Depending on how many decision trees are necessary, data points will continue to be randomly chosen from the training data and more decision trees will be created. Once all the decision trees have been created, the testing data can be classified through voting, where the majority vote determines the categorization of the data point [24].

The precision for each intent is shown in Figure 6. From a quick scan of the graph, it is clear that the precision for the RF classifier is much better than that of AdaBoost. There are several intents that have a precision score of 1.0, including baby_diarrhoea, pain_breast, and ultrasound. While there are still some intents that have a precision of 0.0, there are very few intents that have such a low precision with the RF model, especially when compared to the precisions from AdaBoost. The lowest nonzero precision is 0.180218332 for "pregnancy_general," which is also much higher than the lowest nonzero precision score of the AdaBoost classifier. The macro average for precision is 0.488244827, and the weighted average is 0.489061435. In this case, the macro average and weighted average are close, meaning that the precision was about the same over all the data regardless of the distribution of the intents.

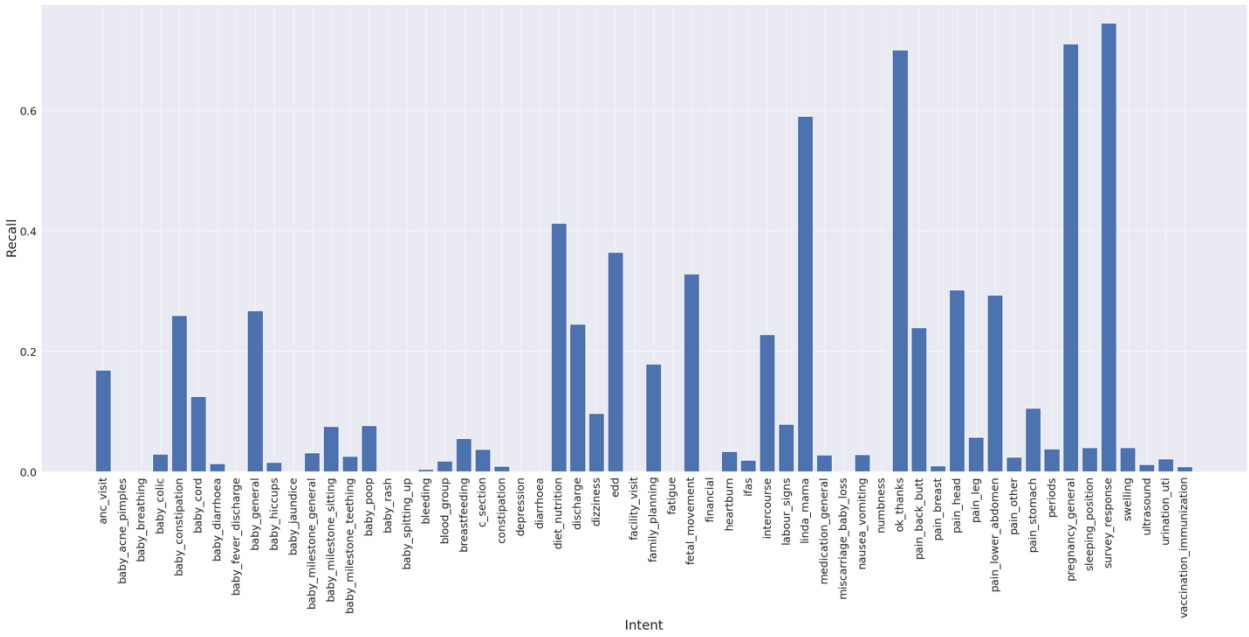
Figure 6: Precision for RF



The recall for each intent is summarized in Figure 7. While the recall scores overall appear lower than the precision scores for the RF model, they are still generally better than the recall scores from AdaBoost and there are less recall scores of 0.0. The highest recall of

0.737166324 is for “survey response,” and the lowest nonzero recall of 0.0041841 is for “bleeding,” both of which are higher than the highest and lowest recall scores, respectively, from AdaBoost. Even though “pregnancy_general” has a low precision score with the RF classifier, it has the second highest recall score of 0.706120944. The macro average for recall is 0.122889067, while the weighted average is 0.325117207. Similar to the averages from AdaBoost, the difference in the macro and weighted averages shows that the recall is better for intents with more data.

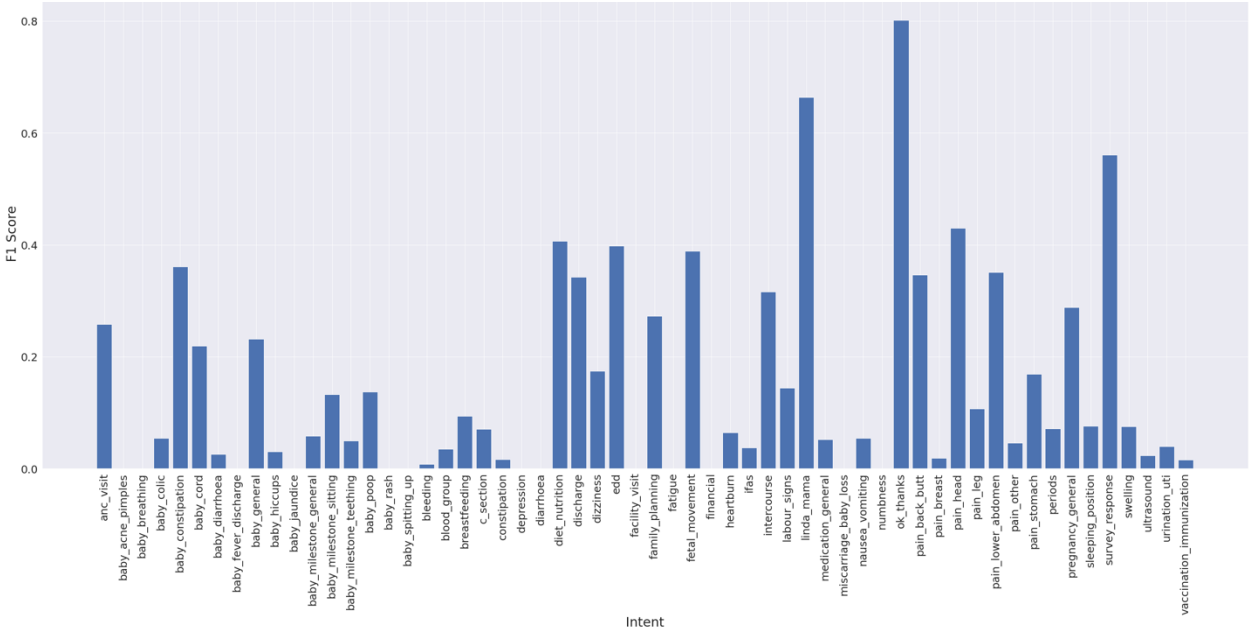
Figure 7: Recall for RF



The F1 scores for each intent are displayed in Figure 8. Since the precision and recall scores were overall better for the RF classifier than the AdaBoost classifier, the F1 scores are also generally better for the RF model. The highest F1 score is for “ok_thanks” with a score of 0.79202773, while the lowest nonzero F1 score is for “bleeding” with a score of 0.008333333. Similar to the precision and recall scores, the highest and lowest F1 scores are higher for the RF

model than the AdaBoost model. The macro average for F1 scores is 0.14481923, and the weighted average is 0.28513143, so the F1 scores are better for intents with a larger portion of data.

Figure 8: F1 Score for RF



Although a full comparison of the three models will be done after investigating the k-NN classifier, the RF model seems to have generally outperformed the AdaBoost classifier. In terms of all three metrics, the RF classifier achieved better results.

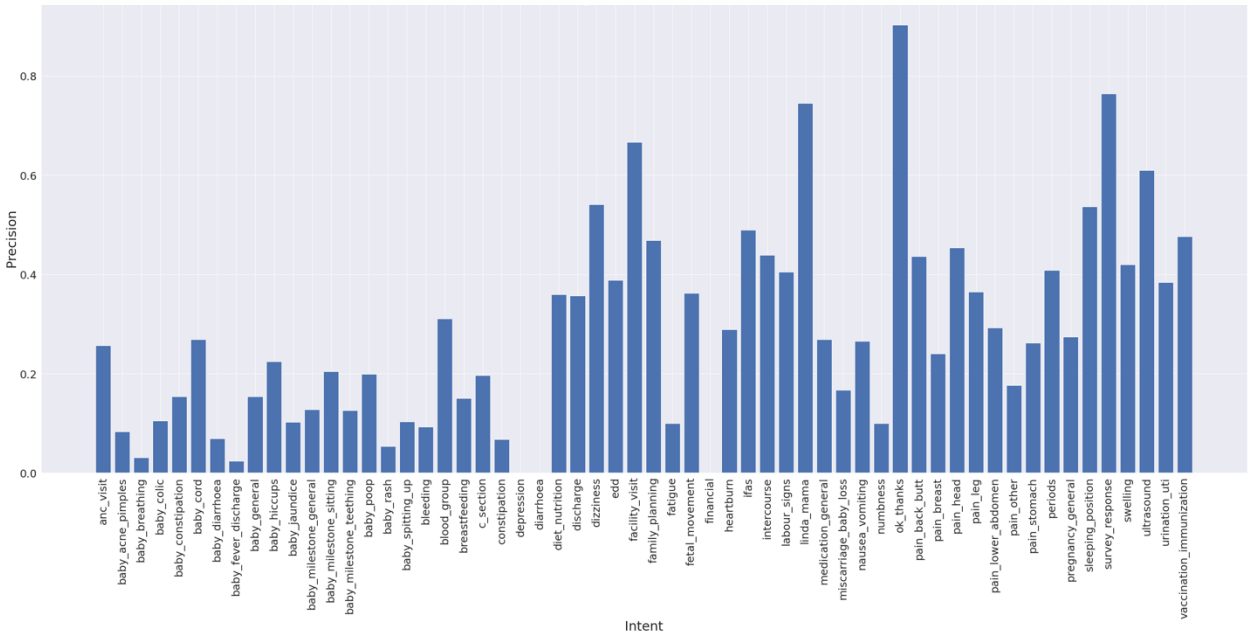
k-Nearest Neighbors

Contrary to the AdaBoost and RF classifiers, which are ‘eager learners,’ the k-NN classifier does not involve pretraining to create a model that will then be used to predict classifications of testing data. Instead, the k-NN classifier is a ‘lazy learner,’ so a model is not created using training data before predictions are made about the testing data. The training data is plotted in an n-dimensional space, where n is determined by the number of data attributes, and

each data point is labeled with its intent. Each point in the testing data is then plotted in the n-dimensional space, and the classification of a testing data point is determined by the labels of its k nearest neighbors, where k is specified as a parameter. For the model in this case, k is set to be the default of 5. The label with the majority vote based on the k nearest neighbors is assigned to each test data point [25].

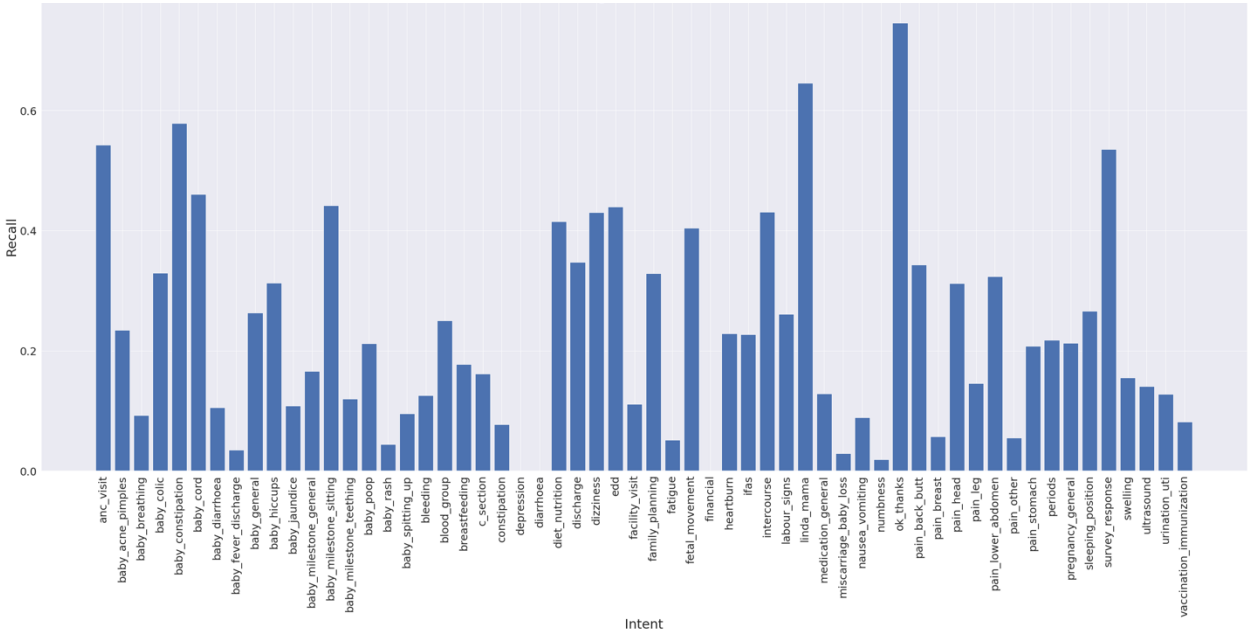
The precision for each intent is summarized in Figure 9. The number of precision scores that are 0.0 is the least for the k-NN classifier out of the three models, with only 3 intents having a precision of 0.0. The highest precision of 0.902752294 is for the "ok_thanks" intent, and the lowest nonzero precision of 0.023529412 is for "baby_fever_discharge." While the highest precision from the k-NN classifier is larger than that from AdaBoost, it is lower than the highest precision from the RF model. Also, the lowest precision from the k-NN model is smaller than the lowest precisions from both the AdaBoost and RF classifiers. The weighted average for precision of 0.380853557 is higher than the macro average of 0.284792197.

Figure 9: Precision for k-NN



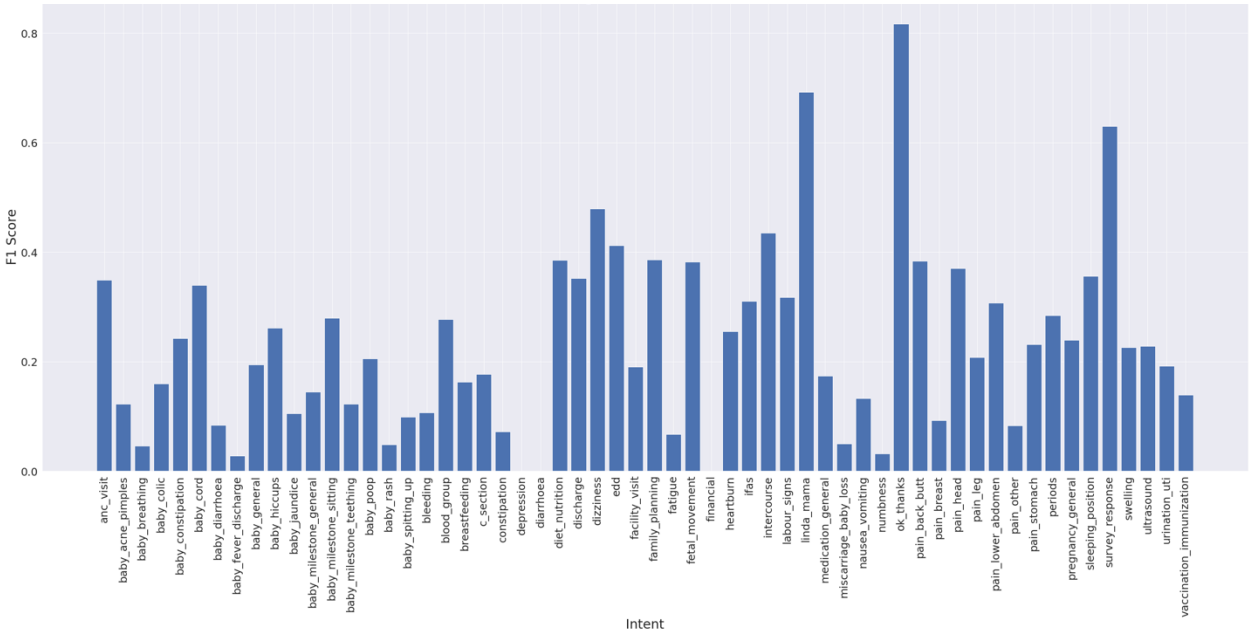
The recall for each intent is displayed in Figure 10. The highest recall score is for "ok_thanks" with a score of 0.745454545, which is the highest recall out of all three models. The lowest nonzero recall score is for "numbness" with a score of 0.018867925. This is the highest lower bound for the recall score out of the three models as well. The macro average for recall using k-NN is 0.231764762, while the weighted average is 0.328552198. Once again, the weighted average is higher than the macro average.

Figure 10: Recall for k-NN



The F1 scores for each intent are shown in Figure 11. Since the "ok_thanks" intent performed the best in terms of both precision and recall, it also has the highest F1 score of 0.81659751. In fact, this F1 score is the highest one out of all three models. The lowest F1 score of 0.028169014 for the "baby_fever_discharge" intent using the k-NN classifier is also the largest one out of the lowest F1 scores of the three models. The macro average of the F1 scores is 0.232216644, and the weighted average is 0.336308875. The difference in the macro and weighted averages for F1 scores is similar to the difference in averages for precision and recall.

Figure 11: F1 Score for k-NN



The k-NN model mostly performed better than the AdaBoost classifier, but when compared to the RF classifier, there were some metrics where each one performed better than the other. Since it is not as clear which model would be a better fit to achieve the overarching goal, the next part of this paper will go into a more detailed comparison of the three models.

Comparison of Three Traditional Classification Models

For each of the three classification models, the overall accuracy, which is a ratio of the total correct predictions (both true positives and true negatives) to the total number of predictions, is shown in Table 1 [23]. As can be seen, the RF and k-NN classifiers perform better than AdaBoost by approximately 13%. RF and k-NN, however, have very close accuracy scores, with k-NN performing better by less than 0.5%.

Table 1: Overall Accuracy for Traditional Classification Models

AdaBoost	0.19686209
RF	0.325117207
k-NN	0.328552198

The macro and weighted averages for the three performance metrics (precision, recall, and F1 score) for each of the classifiers are summarized in Tables 2 and 3. Similar to the accuracy scores, the RF and k-NN models performed better in every metric for both the macro and weighted averages than the AdaBoost classifier. The k-NN model yielded better recall and F1 score averages than RF, while the RF classifier resulted in better precision scores for both macro and weighted averages.

Table 2: Macro Average for Each Performance Metric

	Precision	Recall	F1 Score
AdaBoost	0.073576464	0.049009902	0.044836835
RF	0.488244827	0.122889067	0.14481923
k-NN	0.284792197	0.231764762	0.232216644

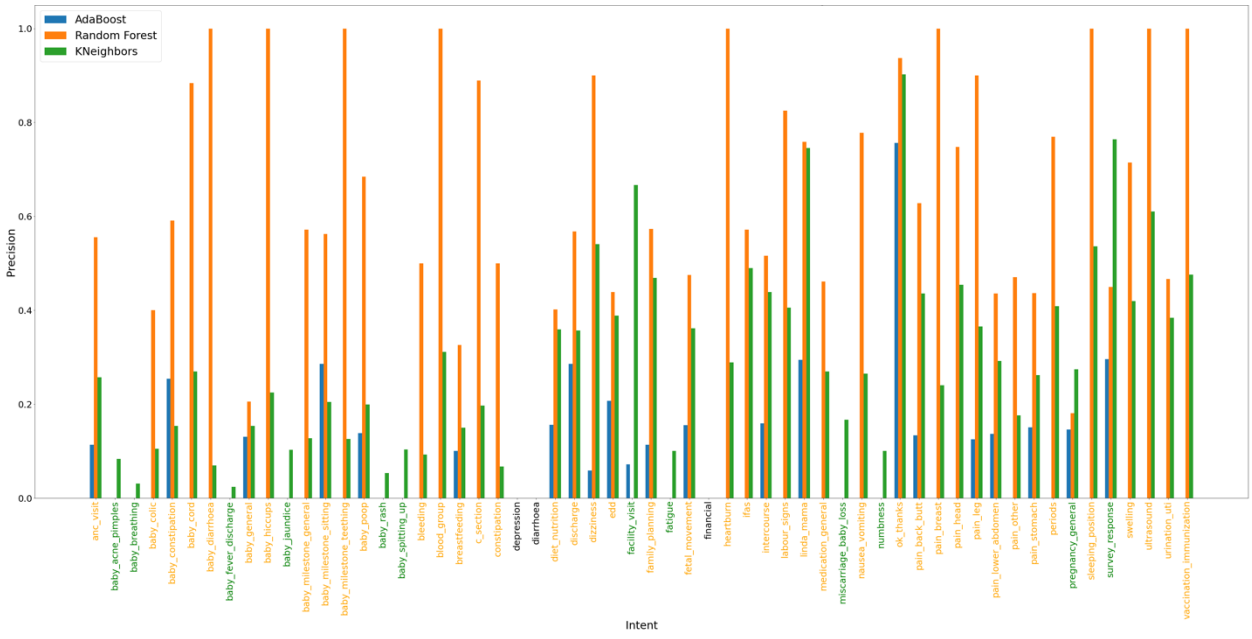
Table 3: Weighted Average for Each Performance Metric

	Precision	Recall	F1 Score
AdaBoost	0.148979851	0.19686209	0.132315396
RF	0.489061435	0.325117207	0.28513143
k-NN	0.380853557	0.328552198	0.336308875

While the averages in Tables 2 and 3 give an overview of performance metric comparisons for the three models, the graphs in Figures 12, 13 and 14 give a comparison of the precision, recall, and F1 scores, respectively, for each individual intent. In each graph, the blue, orange, and green bars represent the AdaBoost, RF, and k-NN classifiers, respectively. The intents on the horizontal axis are color-coded using the same color scheme as the bars based on which model performed the best for each specific intent. Intents that are black signify that all three models have the same score of 0.0 for these intents.

The graph in Figure 12 comparing the precision scores for each intent shows that the orange bars, which correspond to the RF classifier, are the highest for a majority of the intents, such as “baby_hiccups” and “fetal_movement.” When looking at each intent individually, many of the bars for the RF classifier are drastically higher than the bars for the k-NN and AdaBoost models. The RF classifier not only performs the best in precision when comparing the macro and weighted averages, but it also performs the best for individual intents. The k-NN classifier in green gives the highest precision for the rest of the intents that RF model does not cover, such as “baby jaundice” and “pregnancy_general.” The AdaBoost classifier does not have the highest precision for any of the intents.

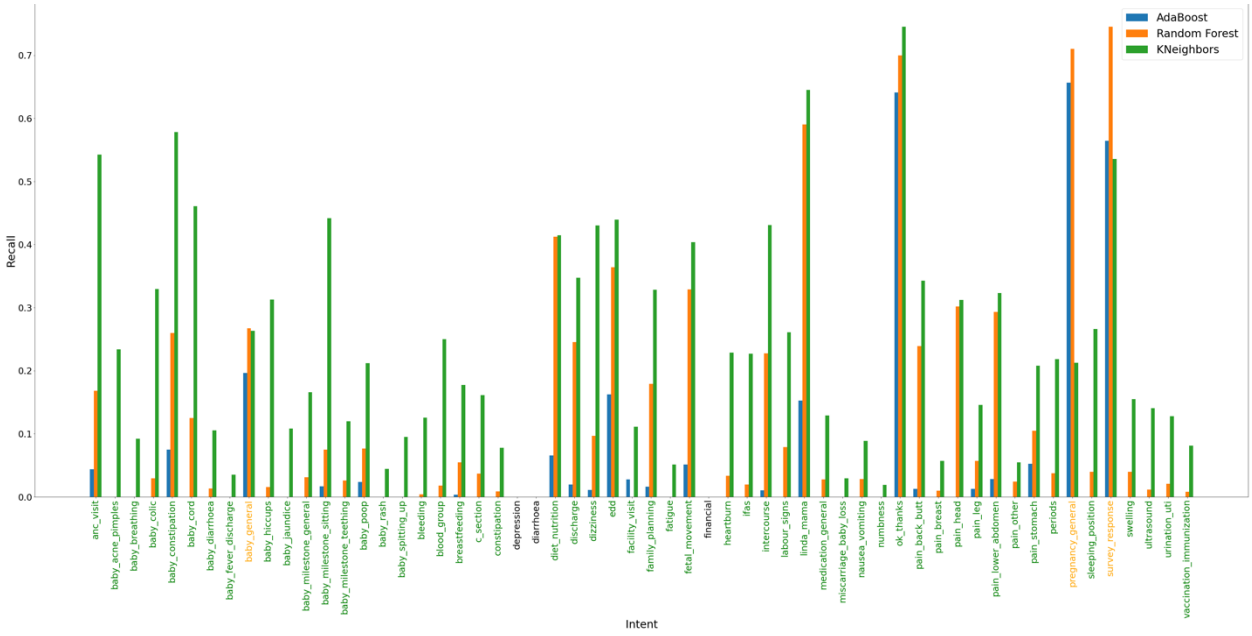
Figure 12: Precision Comparison of Traditional Classification Models



Unlike the precision comparison where the RF classifier performed the best, the recall comparison in Figure 13 shows that the green bars representing the k-NN model are the highest for most of the recall scores. Other than “baby_general,” “pregnancy_general,” and “survey_response” for which the RF model performed better, and “depression,” “diarhhoea,” and

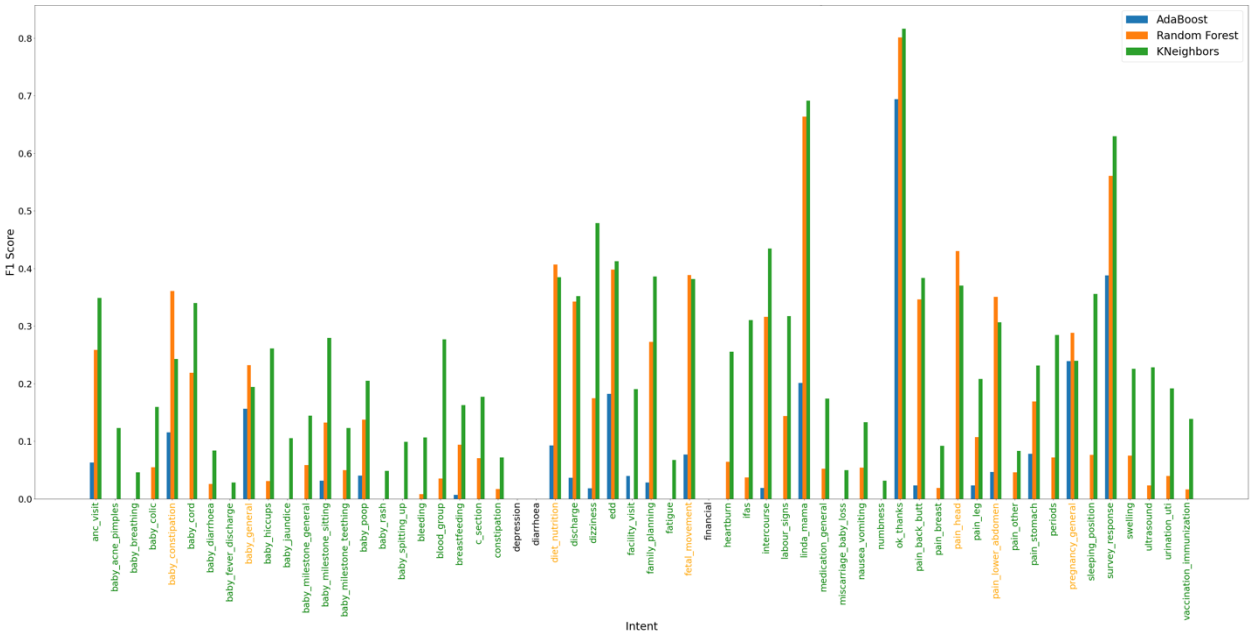
"financial," which had scores of 0.0, all the other intents had the highest recall with the k-NN classifier. Once again, AdaBoost did not perform best for any of the intents. For many of the intents, the recall score from the k-NN model was a substantial amount greater than that from the other two models. Based on observations from both the macro and weighted averages as well as the intent-level comparisons, the k-NN classifier performed best in terms of recall.

Figure 13: Recall Comparison of Traditional Classification Models



The comparison between the three models of F1 scores for each intent are shown in Figure 14. While some intents have the best F1 score with the RF model and others with k-NN, AdaBoost does not have the best F1 score for any of the intents. A majority of the intents have a better F1 score with k-NN in comparison to RF, with only 7 intents having a better F1 score with the RF classifier. The macro and weighted averages for F1 score with k-NN are also better than those with RF. Based on all of this information, k-NN seems to yield the best F1 score results out of the three models.

Figure 14: F1 Score Comparison of Traditional Classification Models



Both the k-NN and RF classifiers have potential to optimize classification performance, since they are comparable in the performance metrics and each classifier achieves better results in different ways. AdaBoost, on the other hand, had the poorest performance by a significant margin in all the performance metrics. Therefore, when creating a new model based on the results from the traditional classifiers, the focus will be on using k-NN and RF to construct an optimal model.

Chapter 5

Proposed Framework

Model

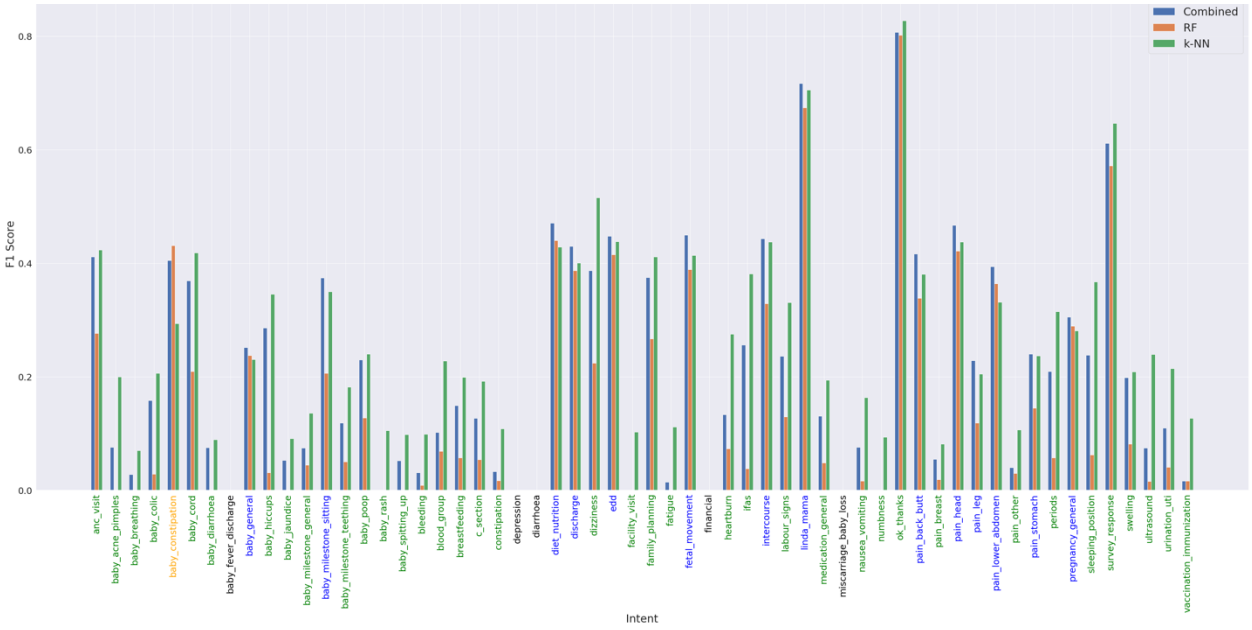
To create a new model using the k-NN and RF classifiers, the data was run through four versions of each type of classifier. The n_estimators parameter, which specifies the number of trees in the forest, was changed between 50, 100, 150, and 200 for the four runs of the RF classifier [25]. During the four runs of the k-NN classifier, the n_neighbors parameter, which represents how many of the closest neighbors of a data point will vote to determine the label of that data point, was changed between 5, 10, 15, and 20 [26]. Once the data was run through all eight versions of the classifiers, each test data point had a predicted label from each version.

The eight predicted labels were used to create three different models. In the first model, each data point was assigned a predicted label based on the intent that was chosen the maximum total number of times by all eight versions of the classifiers. The other two models were similar except that the second model was only based on the four RF classifiers, while the third model was only based on the four k-NN classifiers.

Since F1 scores take into consideration both precision and recall, the intent-level F1 scores of the three models were compared to each other as shown in Figure 15 to determine which model to complete further analysis on. Based on the key in the graph, “Combined,” “RF,” and “k-NN” represent the first, second and third models, respectively, as described in the previous paragraph. The intents on the horizontal axis are color-coded based on which model gave the best results. The model based on solely the RF classifier (orange) performs the best on only one intent, “baby_constipation.” The model based on both the RF and k-NN classifiers

(blue) does the best on 14 intents, while the model based on only the k-NN classifier (green) performs well on 38 intents. Not only does the model based on the k-NN classifier yield better results than the other two models for more intents, but it in fact performs the best on a majority of the intents (approximately 65.5%).

Figure 15: F1 Score Comparison of Three Proposed Frameworks



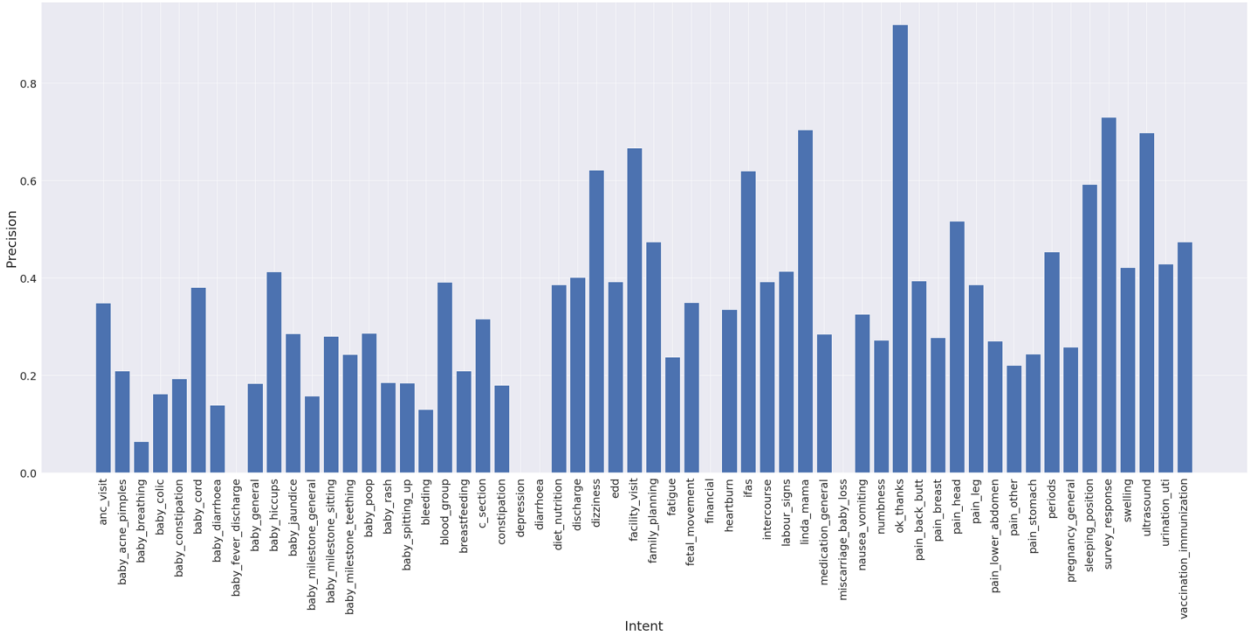
Since the model using only the four different versions of the k-NN classifier has the best performance based on the F1 score comparison between the three models, this model will be used for the rest of the evaluation in this section and will be referred to as “k-NN Combined.”

Experimental Evaluation and Results

The same results were generated for k-NN Combined as were previously generated for AdaBoost, RF, and k-NN in Chapter 4. The confusion matrix is provided in Appendix E. Figure 16 displays the intent-level precision for k-NN Combined, with the highest precision being 0.920222635 for the “ok_thanks” intent and the lowest nonzero precision being 0.064102564 for

"baby_breathing." The macro average for precision is 0.329432379, while the weighted average is 0.398154204. Since the weighted average is a little higher, the precision for intents with larger amounts of data is slightly better.

Figure 16: Precision for k-NN Combined



The recall scores for each intent are shown in Figure 17. The highest recall score of 0.751515152 is for "ok_thanks" similar to the highest precision score, while the lowest recall score of 0.047619048 is for "pain_breast." The macro and weighted averages for recall are 0.242063674 and 0.368240264, respectively. There is a larger difference between the averages for recall than those for precision, signifying that recall is more sensitive to the amount of data per intent.

Figure 17: Recall for k-NN Combined

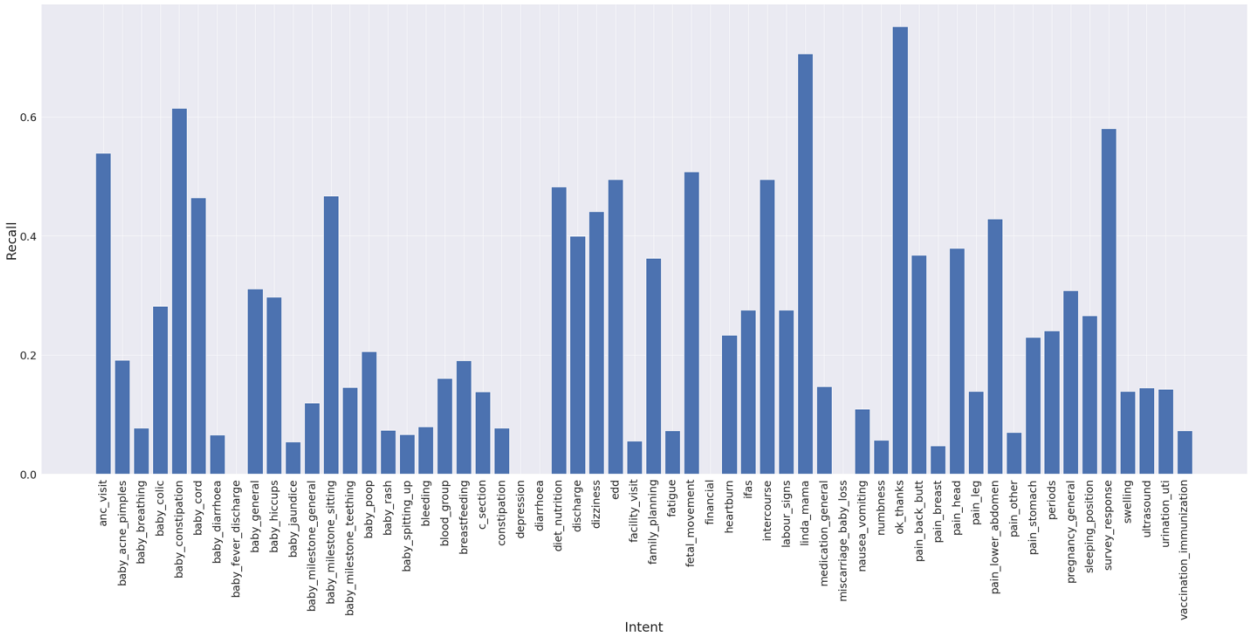
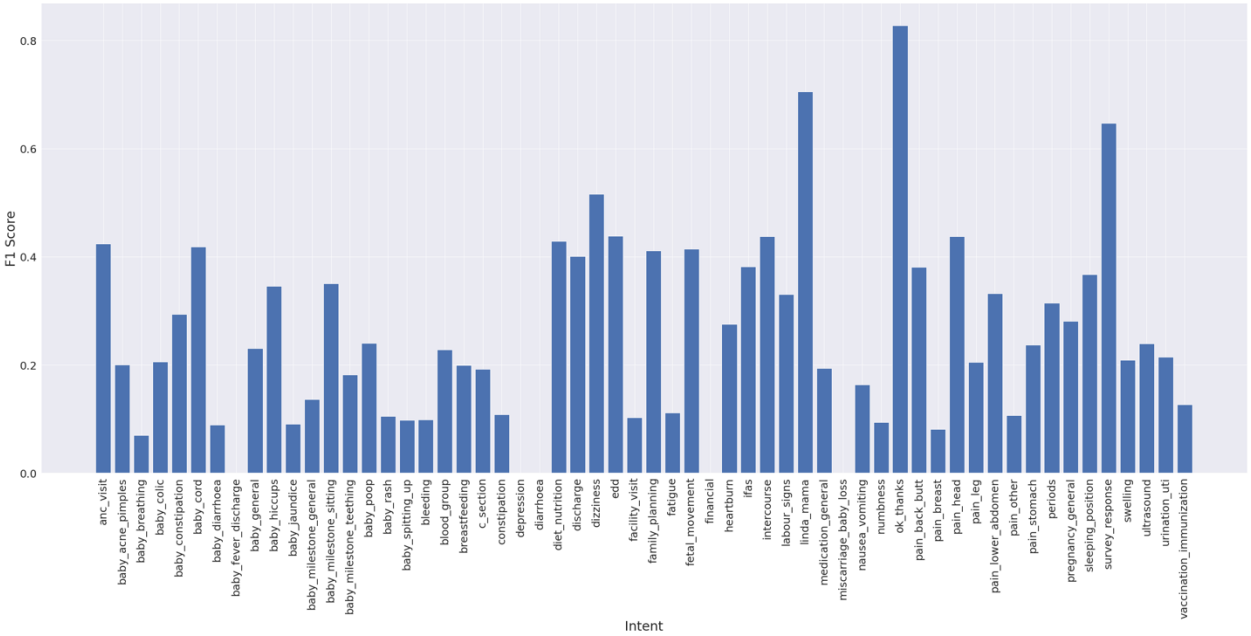


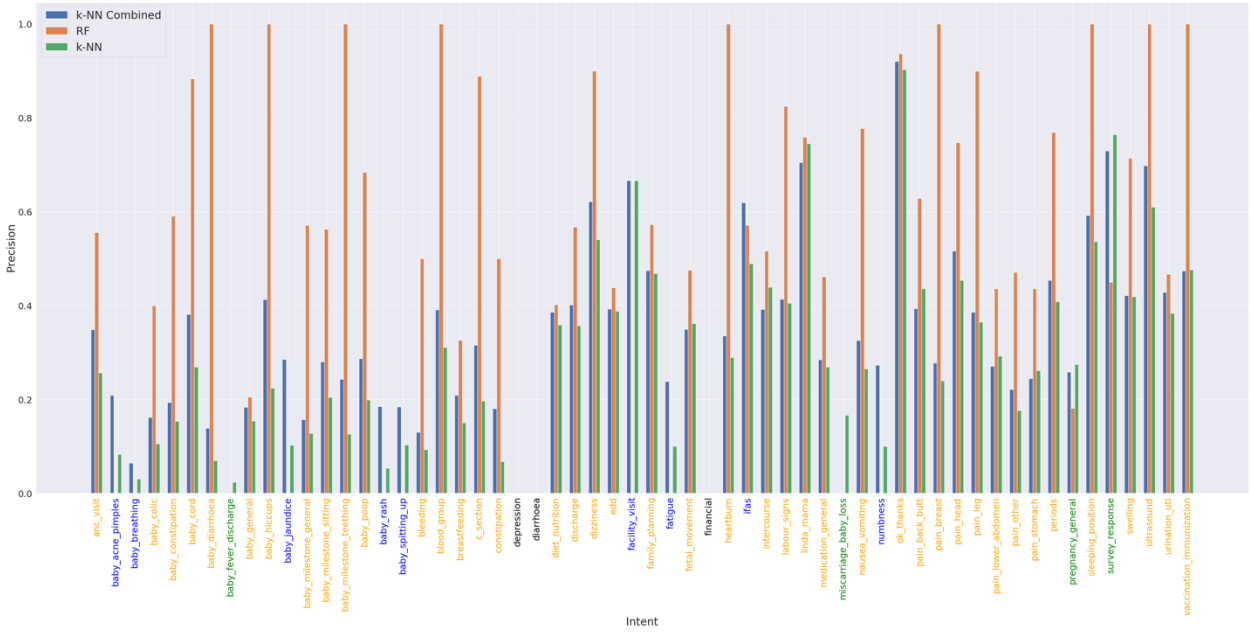
Figure 18 displays the F1 scores. Based on the highest precision and recall scores, the highest F1 score of 0.82735613 is for “ok_thanks.” The lowest F1 score of 0.06993007 is for “baby_breathing.” The macro and weighted averages are 0.253656704 and 0.364241046.

Figure 18: F1 Score for k-NN Combined



To further analyze the k-NN Combined model, the next few figures compare the model to the traditional classification models of RF and k-NN through the performance metrics of precision, recall, and F1 score. Figure 19 shows a comparison of precision between the three models. A majority of the intents still have better results with the RF classifier in terms of precision, but the k-NN Combined model performs better on more intents than the simpler k-NN classifier. Specifically, the k-NN Combined model has the best precision performance for 9 intents compared to 4 intents with k-NN. The macro and weighted averages for precision also support this conclusion, as the averages for k-NN Combined are worse than those for RF but better than those for k-NN.

Figure 19: Precision Comparison for k-NN Combined



The comparison of the recall scores is shown in Figure 20. In terms of recall, k-NN Combined performs the best out of the three models on 33 intents, which is a majority of them (56.9%). In comparison, k-NN yields the best results for 20 intents and RF does the best for 2

intents. The macro and weighted recall averages for k-NN Combined are also better than those for both k-NN and RF.

Figure 20: Recall Comparison for k-NN Combined

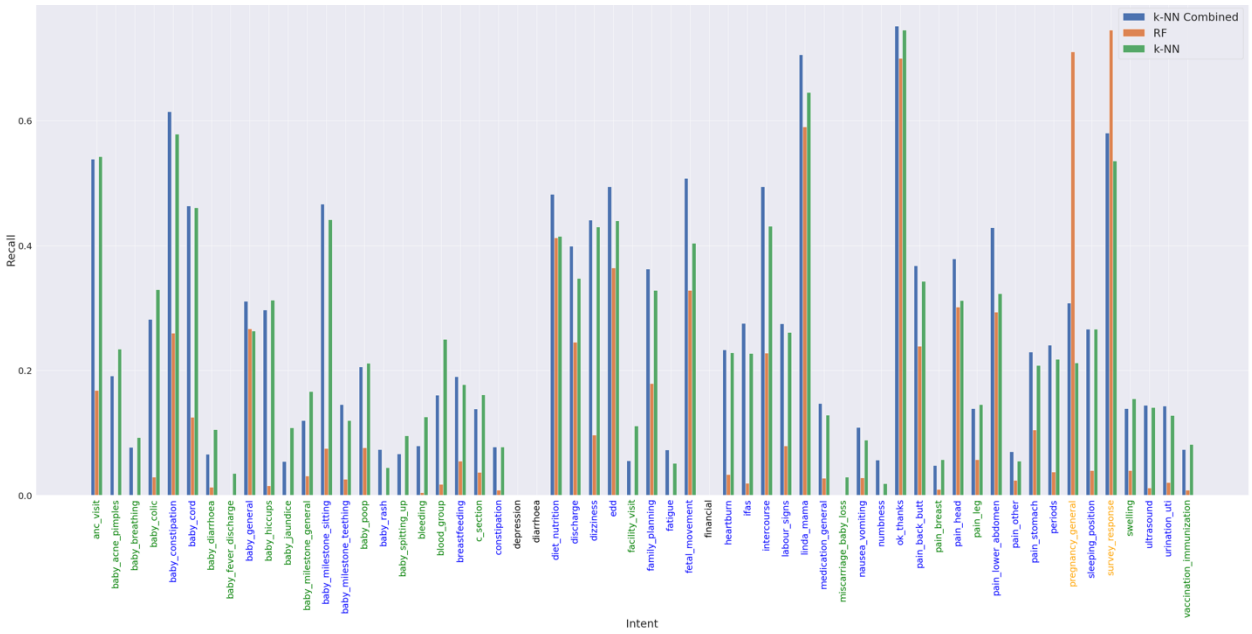
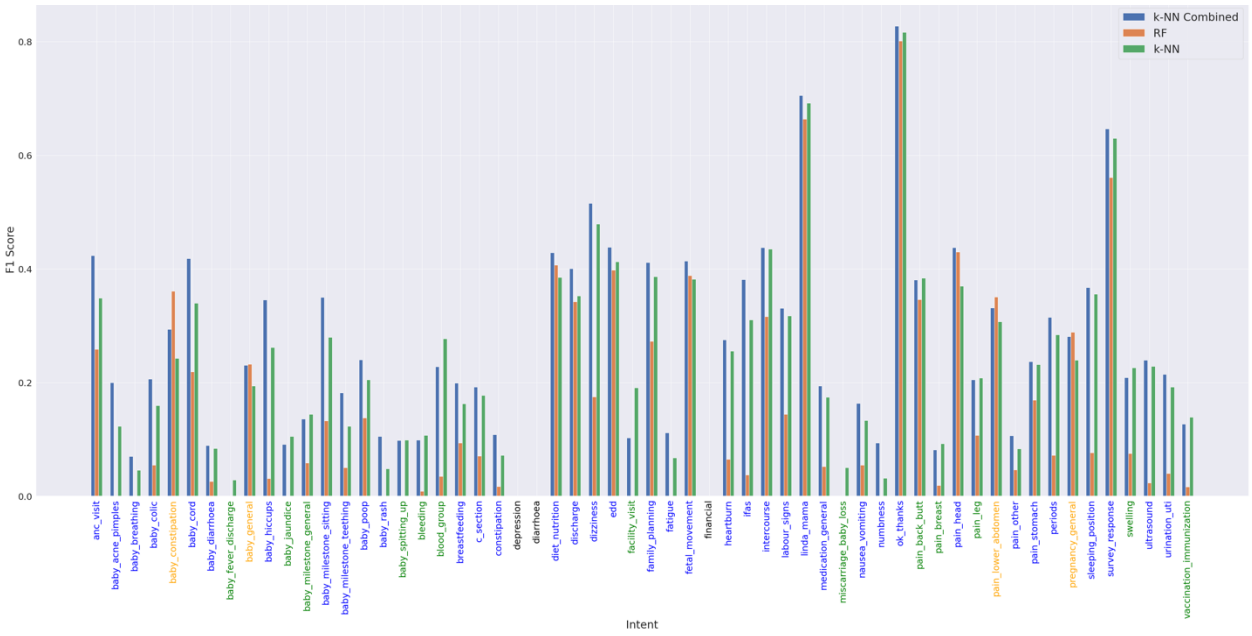


Figure 21 shows the comparison of the F1 scores. While RF and k-NN perform the best on 4 and 13 intents, respectively, k-NN Combined yields the best results for 38 intents (65.5%). Clearly, k-NN Combined has a much better performance in terms of F1 scores. The weighted and macro averages for k-NN Combined are also better than those for both RF and k-NN.

Figure 21: F1 Score Comparison for k-NN Combined



Even though RF mostly performs better than k-NN Combined in terms of precision, k-NN Combined performs better than both the traditional classifiers of RF and k-NN for every other performance metric. The overall accuracy of k-NN Combined, which is 0.36824026, is also better than the accuracies of RF and k-NN, which are both approximately 0.33 as shown in Table 1. Since F1 score is based on both precision and recall, and the k-NN Combined model performed much better than the traditional RF and k-NN classifiers in terms of the F1 score metric, it can be concluded that k-NN Combined has a better overall performance than either of the traditional classification models.

Chapter 6

Conclusion and Further Research

The discussion so far provides a baseline for improving prediction accuracy in the PROMPTS system by better classifying the incoming text messages into their appropriate intents and risk levels. The traditional classifiers of AdaBoost, RF and k-NN provided a good starting point to create a new model by combining the appropriate classification models based on their results. Through a performance analysis of the traditional classification models, it became apparent that AdaBoost did not yield great results but RF and k-NN had potential. As a result, three new models were created using the RF and k-NN classifiers. Out of the three newly created models, the k-NN Combined model, which is based on combining four different versions of only k-NN, was chosen as the final proposed framework since it had the best intent-level F1 scores. Upon further analysis of the k-NN Combined model through a comparison to the traditional RF and k-NN classifiers, it was determined that k-NN Combined had a better overall performance when considering all the performance metrics even though RF had better precision.

Apart from traditional classification models, deep learning frameworks can also be investigated to improve prediction accuracy. As mentioned in the introduction, there are three parts to achieving the overall goal of this research. Exploring deep learning frameworks would help to continue to work towards the first part of the plan. To address the second part of the plan, another aspect to investigate is NLP techniques that would better handle the code-mixed text in Swahili and English. Currently, the Google API is being used to deal with code-mixed text, but it might be possible to improve the accuracy of intent classification by creating a new model for translation based on NLP. The last part of the plan for achieving the goal can also lead to further research by analyzing if changing the system to look at the history of medical records and

messages from patients produces better and more accurate results than simply looking at the currently incoming text. The traditional classification models discussed in this paper have already shown promise, and these avenues of further research could potentially improve the system beyond the current results.

Appendix A

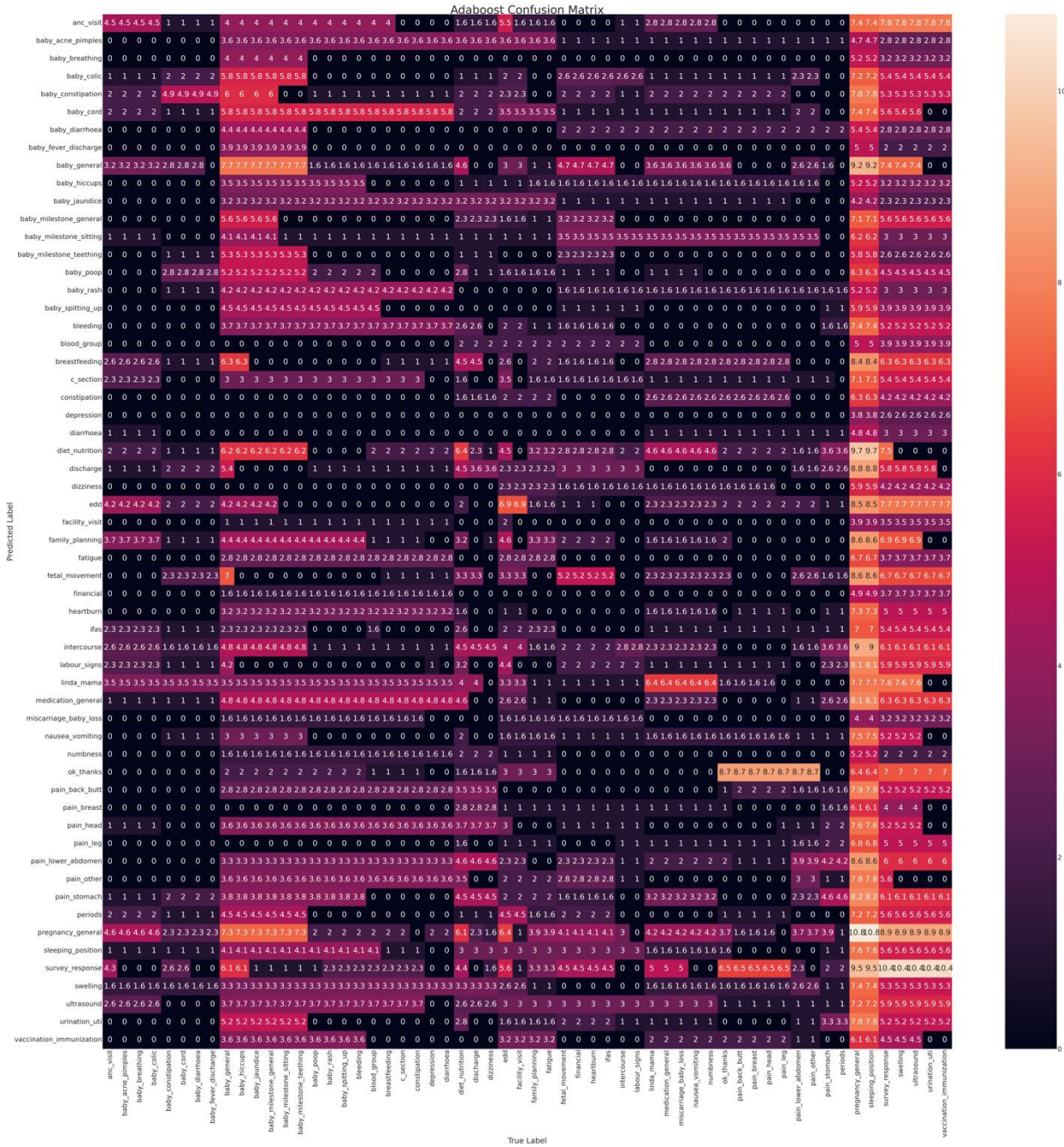
Intent-Label Relations

Intent	Corresponding Label
anc_visit	0
baby_acne_pimples	1
baby_breathing	2
baby_colic	3
baby_constipation	4
baby_cord	5
baby_diarrhoea	6
baby_fever_discharge	7
baby_general	8
baby_hiccups	9
baby_jaundice	10
baby_milestone_general	11
baby_milestone_sitting	12
baby_milestone_teething	13
baby_poop	14
baby_rash	15
baby_spitting_up	16
bleeding	17
blood_group	18
breastfeeding	19
c_section	20
constipation	21
depression	22
diarrhoea	23
diet_nutrition	24
discharge	25
dizziness	26
edd	27
facility_visit	28

Intent	Corresponding Label
family_planning	29
fatigue	30
fetal_movement	31
financial	32
heartburn	33
ifas	34
intercourse	35
labour_signs	36
linda_mama	37
medication_general	38
miscarriage_baby_loss	39
nausea_vomiting	40
numbness	41
ok_thanks	42
pain_back_butt	43
pain_breast	44
pain_head	45
pain_leg	46
pain_lower_abdomen	47
pain_other	48
pain_stomach	49
periods	50
pregnancy_general	51
sleeping_position	52
survey_response	53
swelling	54
ultrasound	55
urination_uti	56
vaccination_immunization	57

Appendix B

AdaBoost Confusion Matrix

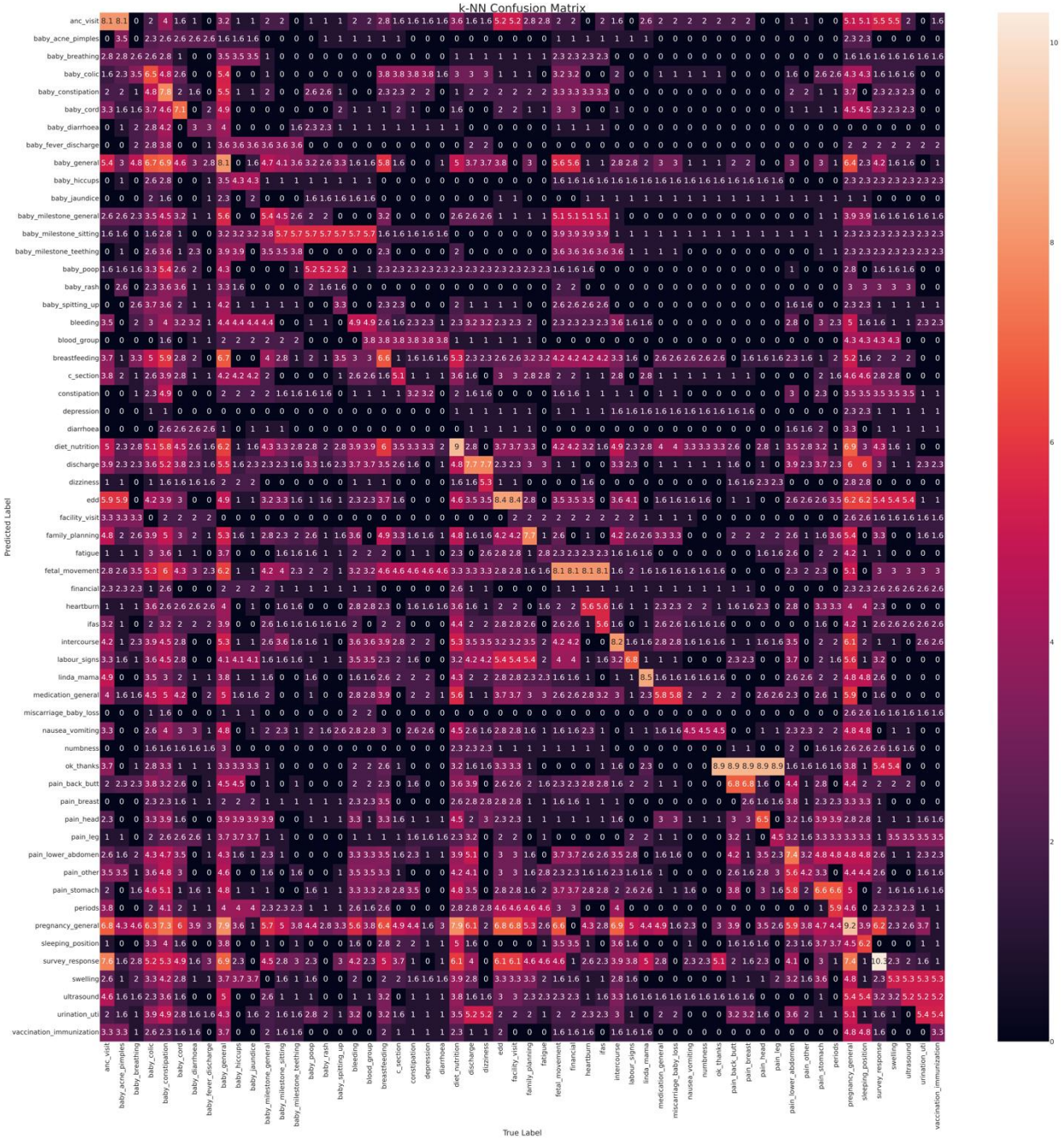


Appendix C

RF Confusion Matrix

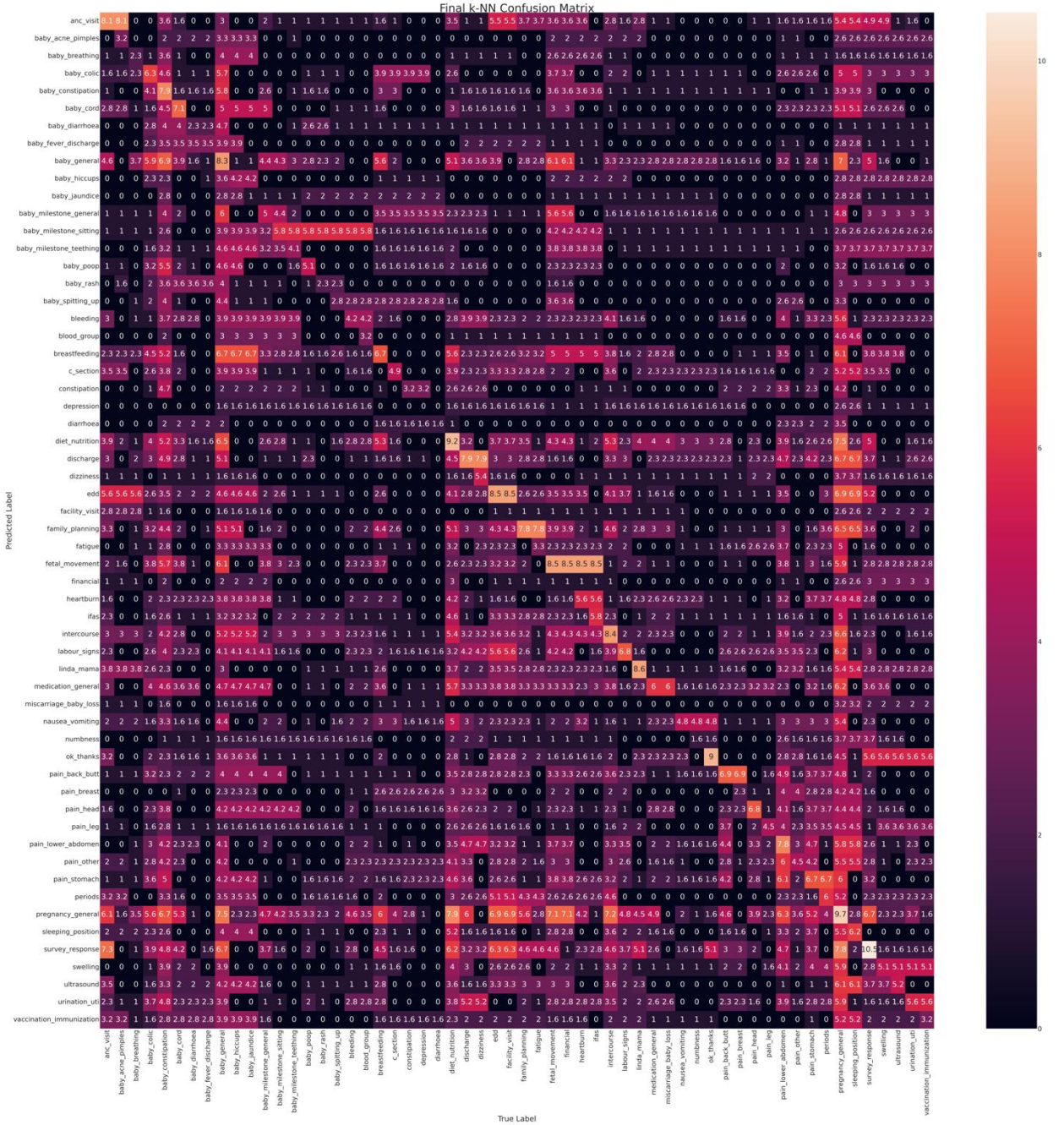
Appendix D

k-NN Confusion Matrix



Appendix E

k-NN Combined Confusion Matrix



BIBLIOGRAPHY

[1] B. B. Masaba and R. M. Mmusi-Phetoe, "Free Maternal Health Care Policy in Kenya; Level of Utilization and Barriers," *Int. J. Africa Nurs. Sci.*, vol. 13, p. 100234, Aug. 2020, doi: 10.1016/j.ijans.2020.100234.

[2] M. L. Scanlon *et al.*, "'It was hell in the community': a qualitative study of maternal and child health care during health care worker strikes in Kenya," *Int. J. Equity Health*, vol. 20, pp. 1–12, Sept. 2021, doi: 10.1186/s12939-021-01549-5.

[3] L. Y. Maldonado *et al.*, "Improving maternal, newborn and child health outcomes through a community-based women's health education program: a cluster randomised controlled trial in western Kenya," *BMJ Glob. Heal.*, vol. 5, no. 12, p. e003370, Dec. 2020, doi: 10.1136/bmjgh-2020-003370.

[4] M. Onono *et al.*, "Narratives of Women Using a 24-Hour Ride-Hailing Transport System to Increase Access and Utilization of Maternal and Newborn Health Services in Rural Western Kenya: A Qualitative Study," *Am. J. Trop. Med. Hyg.*, vol. 101, no. 5, pp. 1000–1008, Nov. 2019, doi: 10.4269/ajtmh.19-0132.

[5] J. Sharma, H. H. Leslie, F. Kundu, and M. E. Kruk, "Poor Quality for Poor Women? Inequities in the Quality of Antenatal and Delivery Care in Kenya," *PLoS One*, vol. 12, no. 1, p. e0171236., Jan. 2017, doi: 10.1371/journal.pone.0171236.

[6] Jacaranda Health, "Quarterly Impact Report October-December 2021." [Online]. Available: https://static1.squarespace.com/static/611674b43f5c6655c9a884a0/t/61b8ca5da0ee872a22522666/1639500417841/Q4+Quarterly+Impact+Report++%287%29.pdf

[7] Jacaranda Health, "Jacaranda Health." https://www.jacarandahealth.org/ (accessed Mar. 08, 2022).

[8] Jacaranda Health, "What We Do." https://www.jacarandahealth.org/what-we-do (accessed Mar. 08, 2022).

[9] Jacaranda Health, "Reaching 1 million mums through PROMPTS," 2022. https://www.jacarandahealth.org/blog/1millionmums (accessed Mar. 08, 2022).

[10] Jacaranda Health, "PROMPTS." https://www.jacarandahealth.org/prompts (accessed Mar. 08, 2022).

[11] Jacaranda Health, "2020-2021 Annual Report." [Online]. Available: https://static1.squarespace.com/static/611674b43f5c6655c9a884a0/t/61aa1b3cdfd3fb39c4f86957/1638538057883/Jacaranda+Health+Lite+2021+Annual+Report.pdf

[12] S. Khanum, M. de Souza, A. Sayyed, and N. Naz, "Designing a Pregnancy Care Network for Pregnant Women," *Technologies*, vol. 5, no. 4, p. 80, Dec. 2017, doi: 10.3390/technologies5040080.

[13] B. Fiore-Silfvast *et al.*, "Mobile Video for Patient Education: The Midwives' Perspective," in *Proceedings of the 3rd ACM Symposium on Computing for Development, DEV 2013*, 2013, pp. 1–10. doi: 10.1145/2442882.2442885.

[14] M. C. Maduwantha and V. N. Vithana, "'MumCare' An Artificial Intelligence Based Assistant," *IJECER*, vol. 1, no. 1, pp. 21–28, Jun. 2021, doi: 10.53375/ijecer.2021.25.

[15] N. Kumar and R. Anderson, "Mobile phones for maternal health in rural India," in *Conference on Human Factors in Computing Systems - Proceedings*, Apr. 2015, vol. 2015-April, pp. 427–436. doi: 10.1145/2702123.2702258.

[16] N. Kumar *et al.*, "Projecting Health: Community-Led Video Education for Maternal Health," in *ACM International Conference Proceeding Series*, May 2015, vol. 15, no. 17, pp. 1–10. doi: 10.1145/2737856.2738023.

[17] A. Vashistha, N. Kumar, A. Mishra, and R. Anderson, "Mobile video dissemination for community health," in *ACM International Conference Proceeding Series*, Jun. 2016, vol. 03-06-June, pp. 1–11. doi: 10.1145/2909609.2909655.

[18] M. Engelhard, C. Copley, J. Watson, Y. Pillay, P. Barron, and A. E. LeFevre, "Optimising mHealth helpdesk responsiveness in South Africa: Towards automated message triage," *BMJ Glob. Heal.*, vol. 3, no. Suppl 2, p. e000567, Apr. 2018, doi: 10.1136/bmjgh-2017-000567.

[19] S. Kotaiah, "Snowball Stemmer – NLP," *GeeksforGeeks*, Oct. 14, 2020. https://www.geeksforgeeks.org/snowball-stemmer-nlp/ (accessed Mar. 08, 2022).

[20] "6.2. Feature Extraction." https://scikit-learn.org/stable/modules/feature_extraction.html (accessed Mar. 08, 2022).

[21] "Pretrained Models," 2020. https://huggingface.co/transformers/v2.10.0/pretrained_models.html (accessed Mar. 08, 2022).

[22] A. Kumar, "The Ultimate Guide to AdaBoost Algorithm | What is AdaBoost Algorithm?," *Great Learning*, Jan. 9, 2022. https://www.mygreatlearning.com/blog/adaboost-algorithm/ (accessed Mar. 08, 2022).

[23] K. Markham, "Simple guide to confusion matrix terminology," *Data School*, Mar. 25, 2014. https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ (accessed Mar. 08, 2022).

[24] “Random Forest Algorithm.” https://www.javatpoint.com/machine-learning-random-forest-algorithm (accessed Mar. 08, 2022).

[25] A. Harnal, “A working example of K-d tree formation and K-Nearest Neighbor algorithms,” Jan. 22, 2015. https://ashokharnal.wordpress.com/2015/01/20/a-working-example-of-k-d-tree-formation-and-k-nearest-neighbor-algorithms/ (accessed Mar. 08, 2022).

[26] “sklearn.ensemble.RandomForestClassifier.” https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed Mar. 08, 2022).

[27] “sklearn.neighbors.KNeighborsClassifier.” https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (accessed Mar. 08, 2022).

ACADEMIC VITA

PRERNA RANGANATHAN

https://www.linkedin.com/in/prerna-ranganathan/

TECHNICAL SKILLS

- **Programming:** Python, Java, SQL, C, MATLAB, Verilog
- **Databases:** DynamoDB
- **Methodologies:** Agile/Scrum
- **Cloud Technologies:** AWS

WORK EXPERIENCE

Vanguard - *IT Intern - Application Development* *June 2021-Present*

- Added AWS lambda logic to validate DynamoDB data and batched lambda invocations to improve resource allocation
- Refactored logic, enhanced test cases, and improved exception handling to increase the resiliency of lambda functions
- Validated data shown on the Ungork UI used by the end client to manage work based on capacity and demand
- Collaborated with other interns to develop a roommate finder application with a focus on frontend work

Penn State - *Teaching Assistant* *August 2020-May 2021*

- Held weekly office hours to assist students, graded weekly blogs and helped professor find content for lectures

Kumon Math and Reading Center - *Teaching Assistant* *December 2015-May 2018*

- Graded 50+ students' papers weekly and organized papers for 100+ students' weekly for future classes

Pennsylvania Free Enterprise Week - *Student* *August 2017*

- Participated in a one-week program to learn about entrepreneurship, finance, business and collaboration skills

CAREER EXPLORATION

Appian First Year Forward Program *July 2020*

- Took part in a day-long virtual event to learn more about Appian and the software industry, and network with others

Liberty Mutual Women in Technology Summit *June 2020*

- Attended a virtual two-day program to learn about women pursuing careers in technology and network with other women

ACTIVITES

Association of Women in Computing – *Member / Volunteer for Girls Who Code* *September 2019-Present*

- Participated in tech talks, workshops, company info sessions, and networking events organized by the association
- Volunteered for Girls Who Code program to teach young girls programming skills

Python Learning Organization – *Member* *January 2019-May 2021*

- Trained in Python through projects to learn more advanced concepts

Future Business Leaders of America - *Vice President (2017) / Treasurer (2016)* *August 2015-June 2019*

- Competed at state and national leadership conferences
- Coordinated logistics at a career fair event for juniors

Federation of Galaxy Explorers (FOGE) - *Volunteer* *August 2018-May 2019*

- Led sessions for space "explorers" from grades 3 through 7 at the local FOGE chapter

Shakti Foundation - *Youth Group Volunteer* *September 2013-August 2018*

- Assisted at an underfunded organization for students with disabilities in India. Volunteered 100+ hours annually.

EDUCATION

The Pennsylvania State University, Schreyer Honors College **University Park, PA**
Bachelor of Science in Computer Science, Minor in Economics *Graduated: May 2022*

Downingtown STEM Academy **Downingtown, PA**
Received IB diploma *Graduated June 2019*

RELEVANT COURSES

- **CMPSC 132:** Simulated a game of blackjack, a basic calculator, a modified cache, etc. in Python using various data structures, such as stacks, queues, linked lists and hash tables
- **CMPSC 221:** Developed a GUI interface and database driven application for a room scheduler in Java and SQL
- **CMPSC 311:** Coded a virtualized device driver that mocked a file system with open, read, write and close functions in C
- **CMPEN 331:** Implemented a 5-stage pipelined CPU design in Verilog
- **CMPSC 431W:** Designed a library database using MySQL, PHP and HTML
- **CMPSC 442:** Explored various AI techniques to optimize a Pac-man agent's moves in a game of Pac-man in Python
- **CMPSC 465:** Used various algorithms in Python, including divide-and-conquer, graph, dynamic programming, greedy, network flow and linear programming
- **CMPSC 473:** Created a dynamic memory allocator in C