Background

Customer satisfaction is an extremely important metric for businesses, and especially so in sensitive domains such as healthcare. Akira by TELUS health is an on-demand virtual healthcare system. When they wish, individuals are able to connect through an app and have a video consultation with a clinician on the other end. This efficiency, ease of use and availability are what make this platform unique. Currently, feedback for clinicians comes in the form of ratings out of 5 as well as more detailed text reviews. As of now the platform uses a simple yet random matching technique which connects the next available patient in the virtual waiting room to the next available doctor. This intuitive approach maximizes the efficiency and availability of the platform, however user satisfaction may be reduced because of this. The user has minimal ability to consult with many different doctors before finding the right match for them.

Considered in this project was an improved patient matching system; how could machine learning ensure a higher probability that a user will be happy with their clinician and maximize satisfaction? In which ways could metrics and behavioural descriptors of doctors be utilized to allow algorithms to match doctors with patients based on a patient's desired behaviour? Wanting a clinician who is quite humorous could be one patient's most important trait, but the next person might have a matching desire for a very serious, no-nonsense approach. Analyzing and sorting the doctors into behavioural bins using patient reviews would allow for this level of granularity to take place. Each review would be analyzed and tagged with a behavioural bin, a model would be created with this data and be used to select ideal clinicians for a user. Behavioural tendencies would be determined to give a doctor their class label and bin them appropriately. Traits can then be accounted for and used to provide a patient with a clinician that is most similar to what they want. To determine exactly ‘what they want’ this implementation has users write a ‘dream review’ about a doctor that they hypothetically have. They would leave the feedback that would describe their experience with their ideal doctor. This review would then be run through the model and compared to other reviews in the search for the best fitting doctor. Best fit could be determined in a number of ways, in this project it is determined as being labeled in the same bin as the ‘dream review’.

Data collection

As with any analytical model, data is the key to accuracy, robustness, and usefulness. Collection of the data for this project was extremely difficult for a number of reasons. Initially, the plan was to gain access to the private doctor feedback reviews on the Akira end. This would be the most logical as it could easily be implemented on their end if successful and would allow for a very clear measurement of effectiveness. Using real feedback from real Akira users regarding their real clinicians would prove the efficacy of this approach. Unfortunately, many restrictions regarding health privacy acted as a blockade in this free flow of information. Patient Health Information (PHI) is a very sensitive issue, covered by HIPAA in the United States of America and PIPEDA in Canada. These are federal statutes with criminal penalties if individuals are found in violation. These statutes ensure that all personal information, especially where sensitive as in the case of healthcare, remains private and secure; reviews in the Akira

database fall under this umbrella. As a private feedback form, there is an obligation on the end of Akira to ensure the privacy of any potential sensitive information here. If users get detailed in their health issues within a review and that information were to be relayed to another individual without consent Akira could be on the hook for large penalties. Difficulty in ensuring that all reviews were not in violation of this forced a different approach to usable data.

Scouring public review forms seemed the next best approach, simply find the reviews and scrape the data to use in the models. Achieving this meant that a webscraper would need to be used on a public review board. The websites found included ratemds.com and ontariodoctordirectory.ca. After investigating the robots.txt pages and the html makeup of the page it was clear that ontariodoctordirectory.ca would be the best choice. The page makeup was simple and scraping would be easy. The problem with this website however was in the quality of reviews. There were simply too few reviews (less than 5) for each doctor and many reviews that were there turned out to be nearly useless due to lack of information. This meant ratemds.com would be the scraped website. To scrape the website, a beautifulsoup implementation was attempted. After many problems with this and feedback from Professor Ding, a beautifulsoup and selenium approach was used. Selenium was needed to evade certain robot limitations implemented by the website.

Public health information is very valuable and hard to get due to the aforementioned federal privacy laws. Due to this, forms that have this data try to protect it as best they can, with many barriers in the way for automated scraping of the website. This meant that the final iteration of the webscraper was still running into problems with captcha. Each page navigation resulted in captchas being required, neutralizing any effectiveness of the scraper. With this new problem in hand it was finally time to accept that to have any model at all there needs to be at least some data. Thus the process of hand collecting the data began. By hand, doctors on the ratemds.com website were selected and reviews were collected from their public review pages. This information contained the users text review, a score out of 5 for staff, punctuality, helpfulness, knowledge, and a combined rating. 10 reviews for 12 different doctors were collected, giving a measly sample size of only 120 reviews. With this data the model implementation could almost begin.

Preprocessing

Preprocessing of the data was much easier than most data would be due to the handscraped nature. All reviews were ensured to be of reasonable quality and there was no empty data. The main problem with the processing was due to the dataset itself. Creating a new dataset meant that each sample had to be hand tagged with a label. This was time consuming as it was necessary to read each review in order to determine the most notable behaviour explained, and also subjective, with human determination for the label given after reading. The determination process was basic: the first trait mentioned in the review would take label1 unless another trait was mentioned multiple times; in which case the other trait would be given label1. To try and minimize this lumpiness a second label was added where possible to smooth the data, The second trait mentioned in the review would take label2, as would a trait mentioned first but not

mentioned as often as another. The labels included in this project were the following: thorough, patient, knowledgeable, compassionate, available, listener, funny, caring, humble, committed, kind, trustworthy, punctual, attentive, negative, helpful, approachable, communication, experienced, friendly, patient, and innovative. The distribution can be seen in figure 1.

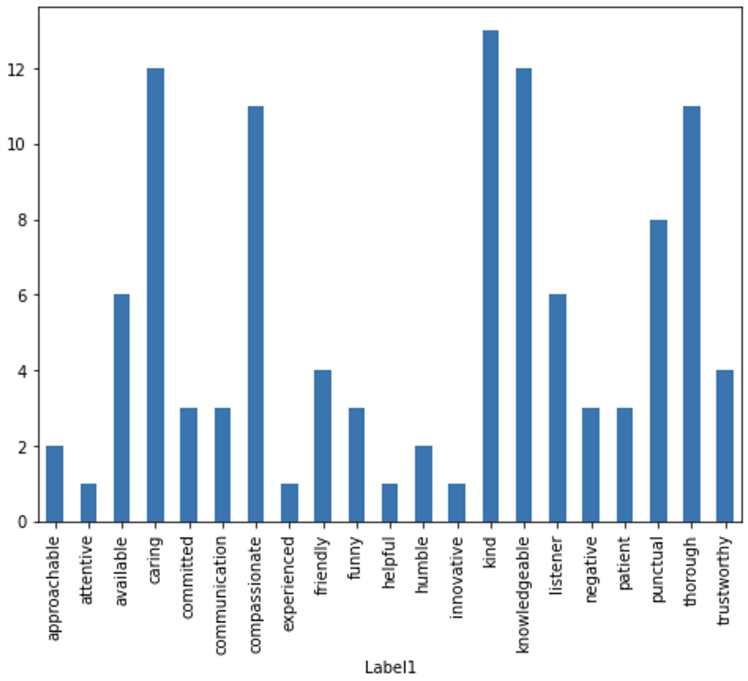

Figure 1: Review distribution of each behavioural bin

There was some hesitation in choosing traits that one would consider a requirement for putting your health in a doctors' hands such as knowledgeable and experienced because that would generally be a given, but were eventually included due to a notable amount of those words being used in descriptions of particular doctors. Another consideration was the number of labels that should be used. With data this small it would probably behoove the model if there were only a few traits. This would allow more data for each trait and encourage more accurate results. Unfortunately this was quite difficult to put into practice as reviews would often be quite specialized in a trait or be vague and able to fit into many bins. The solution to this was to use already made labels when possible and only add a trait when it was necessary. Otherwise this was fairly straightforward and allowed work to begin on model selection and implementation.

Model Selection

Deciding on text classification models was fortunately more straightforward than anticipated. Ideally there would have been an intense look and comparison between deep neural networks to gain fantastic accuracy and dynamic behaviour binning abilities but there was simply too little data to build a worthwhile model. Robustness would be poor, accuracy would be poor, it would be entirely useless with data of this size. Instead, research was done on models that perform best with small datasets.

These models included naive bayes, svc, randomforest, and as a standard comparison, a logisticregression. All approaches were run in this initial stage with default hyperparameters from scikit-learn. These models would be compared using multi-class text classification to place a patient review into a behavioural bin, this would then be compared to the label given. In this implementation there were only unigram and bigram implementations. Higher n-gram counts would even more easily fall into an overfitting trap due to the size of the data and the model would be even more inaccurate with the testing set. With the box and whisker plot shown in figure 1, it was clear that the randomforest performed the best. This test was run a few times to ensure minimal randomness and each time the results were the same - randomforest coming out as the clear winner. Continuing to use the default randomforest parameters was the most practical way forward as they tend to be the most robust. Any improvements from tuning would almost certainly result in overfitting the model and was not worth experimenting heavily with. This was the final model used for this approach, with notable parameters of n_estimators=100, criterion="gini", max_depth=None, min_samples_split = 2, min_samples_leaf=1. The remaining parameters were also default.

Results

Results from the initial model selection are shown below in figure 2.

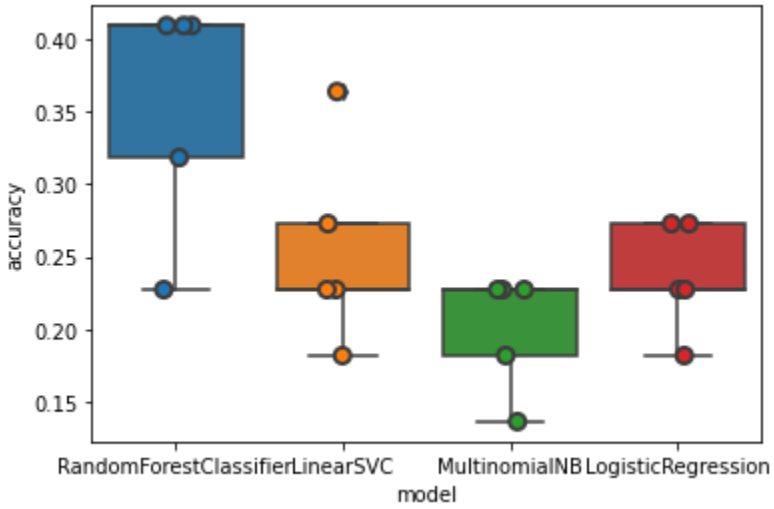

Figure 2: Box and whisker plot comparison of the four potential models

Here it is clear that the randomforest led the pack in accuracy with 0.35, though was middle of the pack in variability. Surprisingly, the multinomial naive bayes performed the worst with 0.20 accuracy. LogisticRegression and linear svc performed similarly, with accuracy scores of 0.24 and 0.25 respectively.

Results of the randomforest model were quite poor in practice. Interestingly however, the accuracy on the testing set was 0.34 to the training set of 0.35. This suggests a somewhat robust model that is neither under or overfitting. Table 1 demonstrates accuracies for each word and a weighted accuracy of the model. The heatmap in figure 3 shows what a problem lack of

data can have. The inaccurate predictions did not follow any sort of pattern and were extremely random and sporadic in which labels achieved accuracy and which models did not.

	precision	recall	f1-score	support
thorough	1.00	0.17	0.29	6
patient	0.00	0.00	0.00	1
knowledgeable	0.00	0.00	0.00	6
compassionate	0.40	0.50	0.44	4
available	0.00	0.00	0.00	3
listener	0.00	0.00	0.00	1
funny	0.00	0.00	0.00	0
caring	0.33	0.25	0.29	4
committed	0.00	0.00	0.00	2
kind	0.67	0.80	0.73	5
trustworthy	0.00	0.00	0.00	3
punctual	0.25	1.00	0.40	1
negative	0.00	0.00	0.00	0
helpful	0.00	0.00	0.00	0
communication	0.00	0.00	0.00	1
accuracy			0.24	37
macro avg	0.18	0.18	0.14	37
weighted avg	0.34	0.24	0.23	37

Table 1: Accuracy metrics for each label

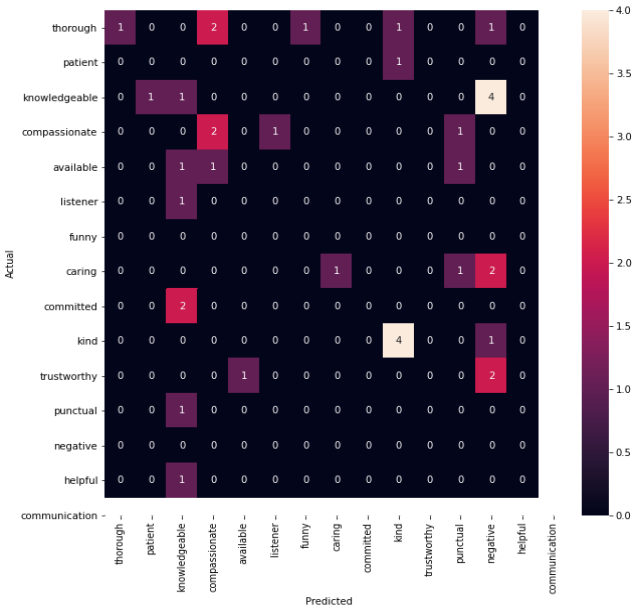

Figure 3: Heatmap of actual vs. predicted labels

There are a number of labels that were included in the training set yet never once guessed in the testing set. These labels were: humble, attentive, helpful, approachable, experienced, friendly, and innovative.

There is a list of the two most correlated unigrams and bigrams for each label, however only two are shown in table 2. This table demonstrates how some correlations are nonsense due to the lack of training data. Words like ‘dr weinstein’ should have no correlation to Listener, however he was often tagged as a listener and reviews tend to name their doctor making his name highly correlated to the label.

Label	**Most correlated unigrams**	**Most correlated bigrams**
Listener	‘fantastic’, ‘listens’	‘takes time’, ‘dr weinstein’
Patient	‘medical’, ‘old	‘years ago’, ‘feels like’

Table 2: Most correlated unigrams and bigrams for ‘Listener’ and ‘Patient’ labels

Table 3 shows examples of two very short ‘dream reviews’ that a user would submit, and the label that was selected from the pool of possibilities.

Review Input	Result output
‘Makes me laugh’	‘kind’
‘Always there for me’	‘punctual’

Table 3: Two short ‘dream reviews’ and their resulting label

Discussion

There is plenty to unpack in this project. We will begin by breaking down the insights gained. It is extremely clear how data impacts model performance and usability. Without enough data these models find patterns that only exist due to unrelated correlations. The example with the listener label responding to ‘dr weinstein’ as an indicator is a clear example of this. Data is the backbone of a good model and it takes data collection to gather useful, relevant data. Due to this inability to collect enough data it is inconclusive if this approach would work in practice. An interesting concern about reported accuracy that wasn’t looked into was the accuracy calculation. With multi-class classification, especially with neural networks, it is not uncommon to use cross-entropy loss to determine the true accuracy of a model. With how inaccurate this model was it seemed unnecessary to look into how the metrics would change, though this would be something to consider in further research with this approach.
It is also unknown if a simple rules-based approach would work for this. With keywords acknowledged through named entity recognition and classifying based on the highest count. That type of approach ignores context of the sentence, whether the user says “they are a bad listener” or “they are a good listener” is ignored, but it is unknown how much of a difference that would make.

Improvements could potentially be found in many different ways and are inherently tied into future applications. The most interesting aspect that could see improvement in this model is from different algorithm approaches. Possibilities of convolutional neural networks or KNN algorithms doing text and similarity analysis could allow a model to compare a ‘dream review’ quickly and accurately to all previous reviews, and find the most similar reviews. A model would be able to recognize traits in the review while also being able to compare the writer to other users. This comparison could be used as a reference to group a collection of users into a bin. Much like advertisers would use these sorts of bins for facebook users this could be used to streamline the matching process or allow it to work in reverse, with doctors being able to select the patient traits they wish to work with.

There are practically limitless possibilities for these insights, with enhanced granularity being one; with more subtle traits being accounted for. There could be levels for each personality trait and users could have a slider style interface where much like a video game character they could place points into each possible trait to find their best match.

Any of these applications could become increasingly important for Akira as they move more into the mental wellness side of healthcare. Being able to give worthwhile descriptors of

psychologists and psychotherapists to encourage the trust and comfort a patient has with them. It is already overwhelming to seek help for mental health, and moving those interactions online only makes it more difficult. It is quite important for Akira to focus on ensuring users can still feel comfortable and willing to share everything they need to, and hopefully an approach like this would encourage that.

References

Li, Susan. “Multi-Class Text Classification with Scikit-Learn.” *Medium*, Towards Data Science, 20 Feb. 2018, towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f.

Deutschman, Zuzanna. “Multi-Label Text Classification.” *Medium*, Towards Data Science, 4 Dec. 2019, towardsdatascience.com/multi-label-text-classification-5c505fdedca8.

Saldanha, Rodolfo. “Multi-Label Text Classification with Scikit-Learn and Tensorflow.” *Medium*, The Startup, 8 May 2020, medium.com/swlh/multi-label-text-classification-with-scikit-learn-and-tensorflow-257f9ee30536.

“Multi-Label Classification: Overview & How to Build A Model.” *MonkeyLearn Blog*, 8 June 2020, monkeylearn.com/blog/multi-label-classification/.

“8 Qualities to Look for in a Doctor - Live Better.” *Revere Health*, 16 May 2019, reverehealth.com/live-better/8-qualities-look-doctor/.

Ardazi, Shai. “Web Scraping with Python - A to Z.” *Medium*, Towards Data Science, 6 Apr. 2019, towardsdatascience.com/web-scraping-with-python-a-to-copy-z-277a445d64c7.