Final Project - K Night Stand

Sasha Cui (nc228), Yang Li (yl967), Matthew Rui (mxr3), David Zhao (dsz10)

Duke K-Night-Stand Pioneers Enhance Dating Dynamics with Predictive NLP-Driven Matchmaking

Durham, NC – April 26, 2024 – Speaking in front of the architecturally renowned Levine Science Research Center, Duke students Sasha Cui, Yang Li, Matthew Rui, and David Zhao announced “K Night Stand,” an application that aims to disrupt “big dating,” with its novel NLP-driven matchmaking algorithm sending shivers down the spines of the likes of Bumble, Hinge, and Tinder.

The students spoke about the various motivations for this soon-to-be unicorn. “Dating in Durham is incredibly difficult, the majority of students are more concerned about their Machine Learning final projects than finding a suitable match,” Zhao lamented, who is simultaneously pursuing a Master’s Degree in Computer Science. Rui, who is pursuing his Bachelor’s Degree in Computer Science, noted that since fraternities that threw parties that were “the historically most efficient way of finding one’s soulmate,” were kicked off campus, “we noticed a chronic need for infrastructure to fill in this matchmaking gap.”

The developers, backed by Y-Combinator, Andreessen Horowitz, and Sam Bankman-Fried, over the past 10 weeks have developed a novel, proprietary matching model, which introduces potential couples based on a novel "compatibility" score. Unlike other dating apps, which primarily match users based on similar profiles, K-Night-Stand also utilizes a proprietary NLP model to extract personality and compatibility characteristics from one’s past conversations in pairing potential matches.

A recent study conducted with 5000 students at Duke University and nearby academic dumpster fire University of North Carolina, Chapel Hill found an impressive 98.7% of users preferred K Night Stand over other traditional dating apps. One student, who requested anonymity, boasted about entering into four separate relationships within a week of beta testing. The data also supports the success of K Night Stand: users were reportedly 2.3x more likely to schedule a first date through this novel platform compared to existing dating apps.

Cui, pursuing her Master’s Degree in Computer Science, announced that K Night Stand would be available in App Stores on May 1st, 2024. Users will be able to choose between a free and paid tier. Two hours following this announcement, over 200,000 pre-orders have already been placed across the Apple App Store and Google Play Store. The public reaction can be best described as “jubilant,” with celebrities such as Kanye West and Elon Musk indicating their excitement on X (formerly Twitter).

Li, also pursuing a Master’s Degree in Computer Science, wrapped up the announcement, promising to “foster a platform that unites human experiences, that both pushes the frontier of technology and embraces the love and emotions that define our humanity.”

FAQs

Q1: What are the final deliverables?

A1. We've produced the algorithms/methods for extracting sentiment/emotion scores from conversation data and predicting high-probability matches given user-profiles and extracted NLP data. Using user metadata (like age, gender, gender preferences, etc) along with our extracted NLP features, we performed extensive feature engineering and implemented/optimized various machine learning models to predict and give hopeless singles better quality connections. Going forward, given two-sided conversation data, we also plan to implement models to extract the success/failure of a match given a conversation's metadata. Combining these models, our goal is to produce an end-to-end model continuously updated with data from new messages to produce new high-likelihood match recommendations. As more users engage with our platform, our models will be further optimized.

More details can be found in the technical details/results below.

Q2: What data is collected from users?

A2. Like normal dating apps, users will upload information about themselves like name, age, height, occupation, location, photos, sexuality, and partner preferences (like political affiliations, religion, etc). As users have conversations with other people on the application, our NLP model will calculate sentiment scores of the conversation and continually update this data for the current user and will leverage powerful predictive tools to probabilistically score the chances of success with other users.

Q3: How do you guys make money?

A3. All users will have access to the free version of the application which does not give users the option to see "high likelihood" matches. The paid version, starting at $69.69 a month, will allow users to leverage our proprietary machine-learning technology to see these "high likelihood" matches. Users are incentivized to do so because the potential matches that they see are also recommended to the user as a high-likelihood match.

Q4: How do you know when previous matches are successful?

A4. At the end of a conversation with another person, the user can indicate whether this match was a "good" match or not. Our model will also probabilistically give a likelihood that the match was good or not based on metadata collected (like the length of the conversation, sentiment scores of the conversation, if phone numbers/other social media was exchanged).

Q5: How do you calculate "scores"/sentiment analysis of given user conversations?

A5. In our dataset, each user has sent out text messages, grouped by different matches. We consider that all conversations sent out by the specific user would effectively reflect how the person talks/interacts with other people in general, and we have used two different metrics for this evaluation purpose. First, we utilized the SentimentAnalyzer in NLTK, which is based on VADER (Valence Aware Dictionary and sEntiment Reasoner), a lexicon-based and rule-based model that gives each sentence four sentiment metrics from these analyses: neg for negative, neu for neutral, pos for positive, and a compound score which is a normalized aggregate of the other scores. For each user, we averaged out the four VADER scores across sentences to get a grasp of the conversation style of the individual. Second, to capture more specific emotions embedded between words, we adopted a BERT-based

transformer model for emotion classifications. Bidirectional Encoder Architecture trained on MLM(Mask Language Modeling) objective is good at capturing the conditional relationships before and after each token in the sentence, with little effect of overfitting. As a result, for each sentence, we obtained a probability distribution across 5 classes: sadness, joy, love, anger, fear, and surprise. Similarly, the aggregated distribution among all text sent out by the user would give the matching algorithm later a good idea of the individual's personality.

Q6: How long will you retain the user data?

A6. We understand the importance of managing and using data responsibly. To balance the need for accurate match predictions and privacy concerns, we have adopted a data retention strategy that mirrors best practices for handling sensitive user information. We will retain user data for one year. This duration allows us to utilize recent and relevant data to enhance our matchmaking algorithms while ensuring our model reflects current user preferences and trends. After this period, user data will be anonymized and aggregated for ongoing statistical analysis to help improve the app, without compromising individual privacy. We will selectively retain smaller, anonymized subsets of user data for long-term analysis to identify broader usage patterns and trends. All other personally identifiable information will be securely erased from our systems to ensure user privacy further.

Q7: What if users stop using the system, what's your procedure for handling their data? How will you predict patterns from there?

A7. Upon a user's discontinuation of the service, all personally identifiable information is securely deleted from our systems according to our data retention policy, which complies with legal standards and best practices. However, we retain anonymized data from these accounts, stripped of any identifiers, to contribute to our aggregate data pool.

To predict patterns from users who have stopped using the system, we rely on the anonymized historical data combined with current active user data, which allows us to detect and analyze long-term trends and shifts in user behavior without compromising the privacy of any individual.

Q8: What if people don't want their chat history being stored and analyzed?

A8. We will provide users with several options to ensure their experience is compliant with their privacy preferences:

- Opt-Out Feature: Users can opt out of chat history storage and analysis at any time through their settings panel. Users can continue to use the app without having their conversations saved or included in data analysis processes. Their previous data will be discarded effectively immediately.
- Not Participate: Choosing not to participate in data storage and analysis will not affect the usability of the app. Users may choose not to participate in the data analysis process and have full access to the app's features and functionalities.

Of course, users who opt out of data-collection features will be unable to utilize our novel NLP-driven match-making algorithms.

Q9: What if users use code words? Is your system able to analyze the traits within the messages?

A9. That's a great question! The use of code words or slang by users presents a unique challenge in text analysis, particularly in the context of natural language processing. Our system employs state-of-the-art NLP models that are trained on vast and diverse datasets,

which include slang, idioms, and coded language. This training enables the system to better recognize and interpret the variety of expressions used by different demographics. Beyond individual words, our system analyzes the context in which words are used. This contextual understanding allows the system to infer meanings and sentiments more accurately, even when unconventional or coded language is used.

Technical details/results

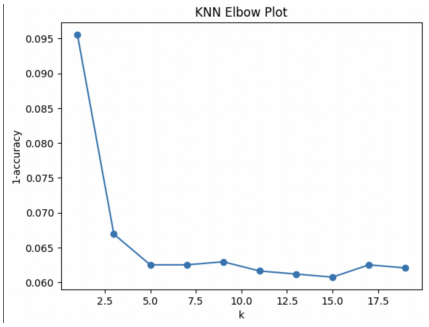
In our analytical process, the initial phase of data processing involved sourcing a dataset consisting of 1209 entries. Each entry corresponds to a user's interaction encapsulated within JSON files (note: we only have access to one-sided conversation data due to privacy constraints). This structured format allows for the systematic extraction of relevant conversational data specific to individual users. We begin our data preparation process by parsing the consolidated and raw JSON files to isolate and organize user-specific data.

To analyze the sentiment and emotion of text data extracted from JSON files, we employ advanced NLP sentiment analysis techniques. This process utilizes two different models:

- NLTK's VADER-based SentimentIntensityAnalyzer: Each user's message is scored on four key metrics: negative, neutral, positive, and compound sentiments. This step evaluates the general tone conveyed in the messages.
- Hugging Face's bert-base-uncased-emotion: Concurrently, the emotion classification model processes the text to identify specific emotional responses, including sadness, joy, love, anger, fear, and surprise. This classification helps in understanding the nuanced emotional layers present in the text messages.

Once we had the raw scores, we proceeded to normalize them to ensure comparability across different users and conversations. For VADER scores, we calculated averages to smooth out the variability inherent in individual messages. For BERT scores, we normalized the totals to create a proportional representation of each emotion relative to the overall emotional content expressed in the messages. These normalized scores were then integrated back into the original dataset, enhancing each user's profile with rich, quantitative insights into their typical emotional and sentiment expressions.

We implement KNN on our dataset to classify whether our current match should predict a successful match or not by looking at the k-closest matches in Euclidean space, and classify based on the majority vote



We use random forests to predict high-likelihood matches given user-profiles and extracted characteristics. We're interested in maximizing the precision of match-making (i.e. the probability that a predicted match is actually a match) over the general accuracy of our model. The random forest is tuned using K-fold cross-validation with outcome classes weighted inversely proportional to frequency to create a balanced dataset (since we anticipate the number of non-matches to far outnumber the number of matches).

To train our model, since we don't have availability of data for two-sided conversations and actual outcomes between previous matches. We attempt to randomly assign match success based on a deterministic rule. For $user_a(attr_{a1}, attr_{a2},...,attr_{an})$ and $user_b(attr_{b1}, attr_{b2},...,attr_{bn})$, we combine to create a new dataset with n features instead of 2n features, where each feature encodes $|attr_{ai}-attr_{bi}|$ (to quantify how similar the two users are to each other). We then set a threshold such that if the sum of the n features of this match is less than our threshold, then we assign match success with probability 0.95. If the sum of the features is greater than our threshold, we assign match success with probability 0.05.

Users are fed suitable matches

If both users match, conversation begins! Model is continually updated with new conversation data

After conversation ends, users input whether the match was good or bad

Algorithm 1 K Night Stand

User u logs into K-night stand

- Given user's preferences (age, gender, political party, etc.), filter out matches that don't fit at least one of their preferences. Provide list of initial matches M
- Initialize empty set M^*, user's that are target/optimal matches for current user
- **for** m in M **do**
  - If prediction(u,m) = True, add m to M^*
- **end for**
- If user has paid status, show M^* to user
- At end of interaction (u, m'), u and m' have option to determine if the match was successful was not.
- If both u and m' provide responses, result of $(u, m') = response(u)$ AND $response(m')$
- If only u or m' provide responses, result of $(u, m') = response(u)$ OR $response(m')$
- If neither provide responses, result of (u, m') determined by scanning conversation (ie length of conversation, sentiment score of conversation, whether phone numbers/social media exchanged)
- Interaction between u and m' used for future training.
- The more conversations user has on platform, scores of friendliness, response time, humor, etc continually updated.

Algorithm 2 Streaming Updates to Model

- Store tuples (day, sum, count) for each extracted sentiment/emotion statistic for each user each day
- For every sent message, calculate sentiment scores and update relevant tuples. After each day, prune old data (e.g. data from conversations from over a year ago) and then calculate average sentiment/emotion statistics with avg. $avg(emotion_i) = \frac{sum(emotion_i tuple[1])}{emotion_i tuple[2]}$
- For every day, run matching algorithm to find optimal matches, filter out previously suggested matches that either user has already seen in the past 1 year.

Acknowledgments

Sasha Cui: Searched and requested dating app data, and explored and selected different NLP packages for sentiment analysis and emotion detection
Yang Li: Performed necessary data preprocessing and utilized NLP techniques to prepare the data for further analysis
Matthew Rui: Clean and prepare the user profile and extracted NLP features for matching, Logistic Regression, Random Forest, Cross-Validation
David Zhao: Exploratory Data Analysis, feature engineering, KNN, Random Forest, cross-validation

We all jointly contributed to the writing of this report.

We'd also like to thank Kristian Elset Bø (https://www.linkedin.com/in/kristianeboe/) for generously providing us with the data for this project. We would like to thank Dr. Laber and Miles Martinez as well as all the other TAs for their assistance and support throughout this project.

References

https://towardsdatascience.com/i-analyzed-hundreds-of-users-tinder-data-including-messages-so-you-dont-have-to-14c6dc4a5fdd