Trey Hibbard 5/8/2025

Blind Dating: Final Report

Background:

Back in spring of 2024, my friend Tyler held a blind dating night which hosted 11 dates (22 participants) all together at the same restaurant, at the same time. Data regarding the participants and their personalities were gathered via a Google Form and qualitative questions. There were mixed results from the dates, with most people parting ways, but there was one pairing who became close friends and another who is still together today! In fall 2024, the same event was held at the same place, although with much more participation. With 60 dates (120 participants), the task of manually pairing up as many dates as possible became a daunting one. With plans to do another round of blind dates this spring 2025, the number of submissions was expected to grow yet again. After I saw a video on YouTube, now titled "What is the Dot Product and How is it Used? Applied Linear Algebra (part 2)", the idea of using cosine similarity to determine correlation became an apparent solution for the future of this project. Together with another friend Garrett, the three of us split up our involvement in the project as follows: I would mainly focus on the mathematics, Tyler focused on the outreach and coordination, and Garrett made the code.

Motivation:

The most tedious part of the blind dates had been individually pairing individuals based on personal judgement. As 22 participants became 120, it also becomes a challenge to see which two people may be a good fit for each other if their rows are far apart in a spreadsheet. Cosine similarity was seen as a solution to this. It would provide each pairing a score between -1 and 1 to determine compatibility based on numerous personality-based questions. After developing a code to compare every person against each other with cosine similarity the code would return a list of each person with a ranking of others based on their cosine similarity pairing.

Cosine similarity transforms our 60 questions, or indicators, into one metric that can be used in determining pairings. Questions focused on 6 categories: emotion, conflict, extraversion, lifestyle, communication, and partner interaction. In addition to the 60 questions, dealbreaker questions such as one's sexuality and age limits were used to eliminate conflicted pairings. The resulting ranking simplifies the final pairing process. There were several options we considered using to make the final pairings, but in the end, we ended up using a maximum weight matching to meet the needs of everybody, those needs being consideration for queer preference and dealbreaker questions. A maximum weight matching works by finding the maximum number of edges in a network such that no edges share a vertex, and adds it to the maximum weight matching if it would increase the total weight.

Through our survey we gathered 192 submissions. That is 11,520 datapoints (60*192) but does not include dealbreaker questions and several qualitative questions asked for manual

investigation. We had about a 44% retention rate by the end as a consequence of short notice and people dropping out. A proposed solution to these problems is made below.

Methodology:

Without any vital quantitative data and results from the first two rounds of blind dates, this time served as round zero for what we intend to be an iterative process going forward. The process is as follows:

1) Determine a list of questions
2) Adjust the algorithm to work with these questions
3) Send the preliminary survey
4) Create a list of matchings
5) Have the blind dates
6) Send the post-date survey
7) Perform statistical analysis comparing the results of the post-date survey to the answers from the preliminary survey.

The beginning of the process again would use results from the past statistical analysis to determine which questions are the most important or should be emphasized more. There is certainly a large amount of personal judgement as to what makes a "good question" on our part, but we hope that by analyzing our results we can improve which questions we ask in the future.

Cosine Similarity

To begin deciding which questions should be asked and the best way to ask them, we sought out questions that would fit our cosine similarity model. To understand how to best fit the model, it's important to understand how it works.

$$\cos(\Theta) = \frac{A \bullet B}{|A|\,|B|}$$

$\cos(\Theta)$ tells us the cosine of the angle between two vectors A and B. For vectors in the space $\mathbb{R}^n$, where n is the dimension and furthermore the number of questions, the vector may be made up of positive and negative values. The dividend is a strictly positive value, so negative scores come directly from the dot product of A and B. This requires us to build questions where inputs in A and B when multiplied are positive if there is correlation and negative if there is not. To do this, we aimed to ask questions that could be answered on a seven-point scale for the values -3 to 3. An example of a question would be:

"You are often drained when at a big party."

The seven-point scaled ranged from “very unlike me” to “very like me” with a score in the middle being assumed as indifference, or mathematically 0. If person A answers this question with a score of -3 and person B answers with a score of -1, then those answers would be placed in the same index of their respective vectors and multiplied in the dot product, adding a positive score of +3 to the sum of A • B. This increases the total cosine similarity score, though it is important to note that -3 * -2 = +6 would’ve resulted in an even higher score. If they differed on the question, however, then -3 * 2 = -6 would decrease the cosine similarity score. This is an assumption we made that compatible individuals would be interested in attending similarly sized events. Statistical analysis after the dates will ideally allow us to see which questions were good indicators or not.

The question provided above is an example of a “reflexive” question as we called it. It assumes that people want the same thing. However, often people want somebody else to be different than themselves. We call these “dual” questions. It also presents the need for each person to have two different vectors. One to describe themselves personally, and one to describe what they are looking for in somebody else. Here is an example of a dual question:

Self: “You tend to be the center of attention.”

Want: “You are more attracted to a quieter personality.”

If person A answered 3 for the “Self” question and person B answered -2 for the “Want” question, then we would calculate the similarity as 3 * -(-2) = 6. The extra negative sign is specific to certain questions and addresses something we referred to as polarity: when one of the responses to a dual question needs its sign flipped to accurately show correlation in the dot product. The two different types of vectors, simplified, are shown below.

ASelf = […]

AWant = […]

BSelf = […]

BWant = […]

To determine person A’s compatibility towards every other person, for example person B, we would apply cosine similarity between AWant and BSelf. Compatibility of B towards A would apply cosine similarity between BWant and ASelf. These two calculations would give different scores and answers the obvious problem that someone may have greater interest than somebody else. The number we used to determine mutual compatibility was the minimum of the two, since if the minimum was high, we assumed the match would be good.

The following images show excerpts of our process handling csv files and python code. The first image shows examples of user responses to the question “You tend to be more attracted to people who are very outgoing.” The response was received as a number from 1-7 and then shifted by four to fit our scale between -3 and 3. The second image shows an example from our code where we consider one our dealbreakers regarding alcohol usage. The third image provides an excerpt (with relabeling for privacy) of the beginning of a list of Person A’s ranking of others based on their score.

	You tend to be more attracted to people who are very outgoing. [Extraversion]
1	
2	3
3	3
4	5
5	7
6	5
7	6
8	4
9	5
10	4
11	6

```
300         # Alcohol and substances check
301         if person1.dealbreakers["alcohol"][1] != person2.dealbreakers["alcohol"][0]:
302             print(person1.dealbreakers["alcohol"][1])
303             print(person2.dealbreakers["alcohol"][0])
304             print(f"Alcohol issue {person1.name}--> {person2.name}")
305             return False, common_days
```

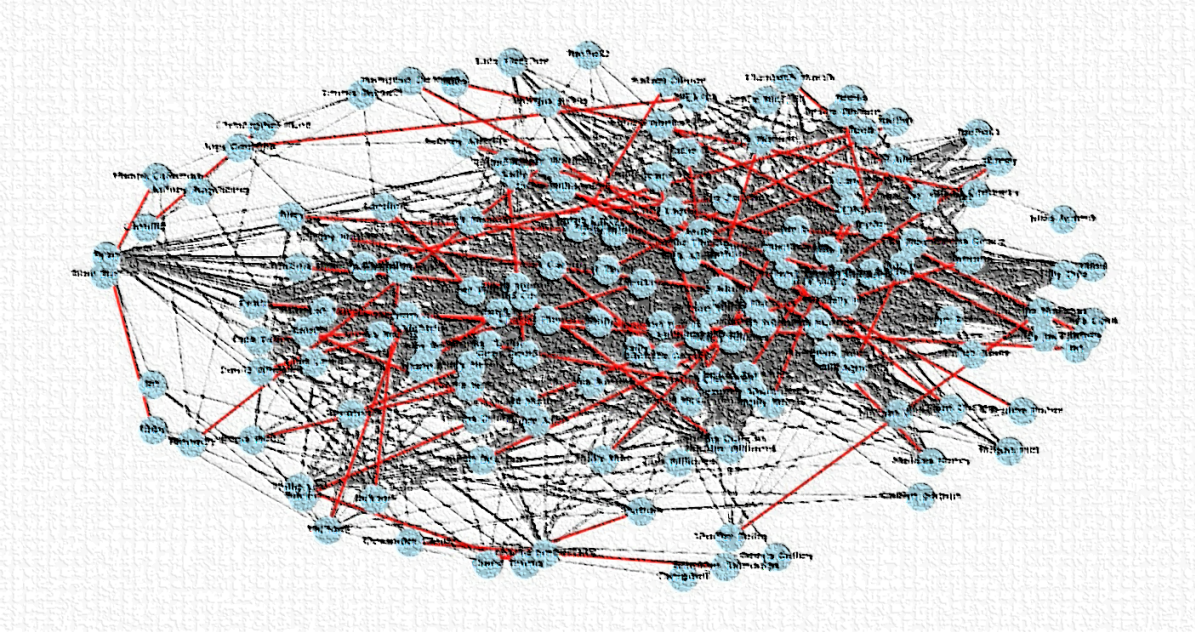
427	**Person A**	Person B	0.8347
428	**Person A**	Person C	0.7407
429	**Person A**	Person D	0.7224
430	**Person A**	Person E	0.6911
431	**Person A**	Person F	0.6803
432	**Person A**	Person G	0.6661

Maximum Weight Matching

After receiving every cosine similarity score of each person towards another, it was a separate question of how we would use that information to pair people. As mentioned in the "Motivation" section of this paper, the method we ended up using was a maximum weight matching. The NetworkX python package provides a simple way to incorporate this.

We set each person's name as a node, and the minimum cosine similarity between two people as a weighted edge. If there was a dealbreaker between the two people, no edge was assigned. The maximum weight matching algorithm then found the pairing list with the largest score sum from the weighted edges. It is important to note that this algorithm is not only maximal, but maximum, which means that there is not a greater combination of positive edge weights that could have been made. In other words, you wouldn't be able to manually spot another edge that had a positive score which nodes weren't already paired elsewhere.

The following graph represents a visual for the matching network. Red lines indicate pairs. The graph has been blurred to maintain privacy.



The pairs gathered from this algorithm were the primary ones used, with manual changes being made when people dropped out after confirming their availability for the date. Manual changes have been noted and will be considered carefully or excluded in the statistical analysis. Compatibility scores exist for all pairings however and still provide a practical way to apply linear regression.

Statistical Analysis:

A post-date survey was sent about a week after the event to everyone who went on a blind date. Questions from this survey gathered both regression and classification information. Here is a summary of the questions asked:

Regression Questions (asked on a 7-point scale from 1-7):

- Did you enjoy your date?
- Did your personalities match?
- Was there romantic tension?
- Did you find them physically attractive?

Classification Questions (Yes or No)

- Did you want a second date?
- Did you have a second date / is one lined up?
- Do you expect to be friends with this person

While a huge aim of the blind dates was to introduce people to a stranger, they may have romantic compatibility with, we understood that our questions also served as a sort of compatibility test overall and could be spun as a friend-finding survey. We aimed to ask questions about both the platonic and romantic mood of the date.

Our hope for our statistical analysis was to find trends in the questionnaire that we could use to improve or change our survey. As will be discussed alongside the models used to look for these trends, the post-survey data we captured was a small sample size of only n=42. With 60 questions / features and a small sample size, overfitting was immediately a clear problem for the analysis process. However, I have used different statistical models nevertheless to look for trends. This analysis may serve more as a "what we could use to analyze the data and why", but there are some upshots to be had from this trouble. Heading into another round of blind dates, re-evaluating the data-collection process and how many questions we should ask is of utmost consideration.

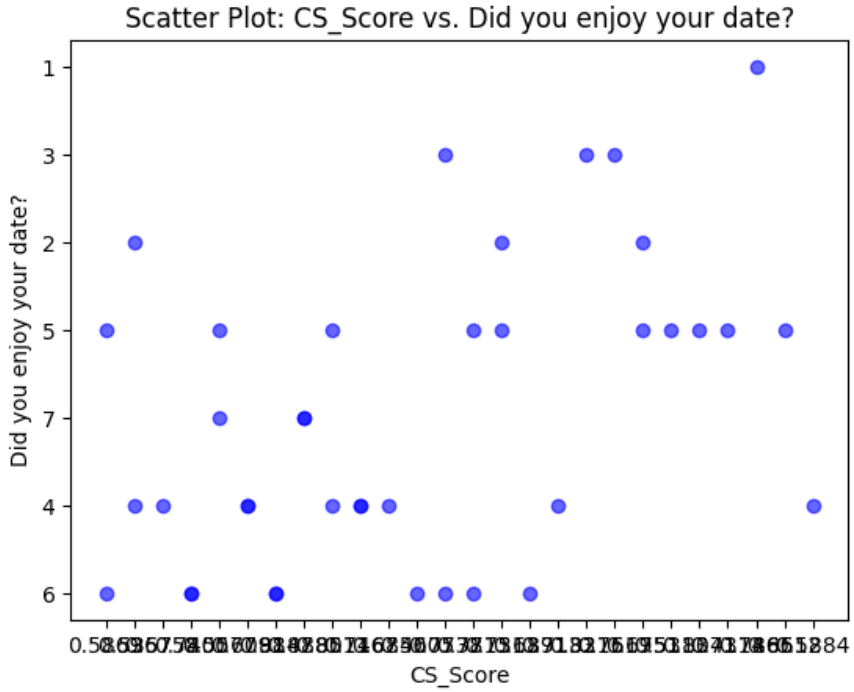
Regression Models

Let's first look at the regression models we used for our analysis. The regression models used were simple linear regression, multivariate linear regression, ridge regression, and support vector regression. All models used an 80/20 training/test split and were standardized.

<u>Simple Linear Regression</u>

The main metric used in the first half of this project was the cosine similarity score. Its main purpose was to provide a way to create a ranking of dating interests between people. Though it proved useful then, the score itself contains so much information that it doesn't really explain anything about "what makes a good date". I applied regression for that feature against

one outcome just to see a trend. The line seemed impossible to draw, but a scatterplot of the data along with the slope and intercept is provided below. The x-axis is from 0.58 to 0.84.



Slope = 0.869, Intercept = 4.133

A mean square testing error of 4.67 and R^2 score of -0.0513. The negative R^2 indicates that the linear model is already overfitting, a problem that will be continuously seen. The MSE tells us that the model is on average off by about sqrt (4.67) = 2.16 units from the target. With questions asked on a 7-point scale, this is a very wide gap and concerning for our model. However, as mentioned before, the lack of data received is the biggest contributing factor into this error.

Multivariate Linear Regression

For the rest of the regression problems, all 60 questions were used, with later models searching to shrink unnecessary questions. First though, a multivariate linear model was used to consider everything. This and most of the other models are too high-dimensional to visualize, so our understanding of the concepts and the numbers will have to suffice.

The multivariate model with no adjustments found a MSE of 11.65 and R^2 of -2.47. Once again, it would've been better to just assume the average of everything. This aside, the five most positively correlated, the five most negatively correlated, and the five least correlated questions according to the feature coefficients were:

Positively correlated

Q31	0.246
Q12	0.184
Q40	0.181
Q55	0.151
Q29	0.132

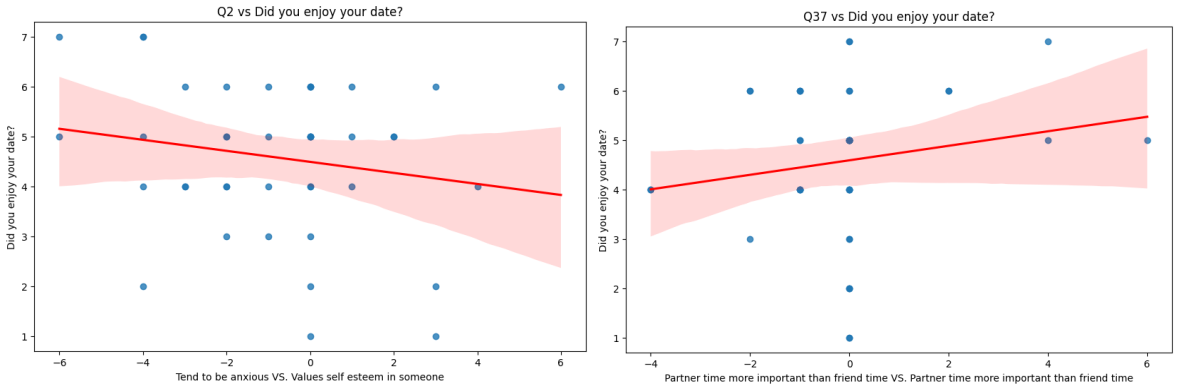
Negatively correlated

Q2	-0.327
Q1	-0.189
Q19	-0.167
Q21	-0.140
Q32	-0.128

Lest Correlated

Q37	-0.000881
Q6	0.001580
Q5	-0.002086
Q59	-0.004382
Q3	-0.006955

A brief inspection of the linear model for each of these against the general question “Did you enjoy your date?” was made to see what the data we had looked like at its extremes.



Just visually, even if slightly, Q31 and Q2 fit their respective correlation and have some directional trend. Q37 seems uncorrelated; most responses to the question of if "partner time is more important than friend time" landed on the average, or the indifferent response of 0. Aside from outliers, it is clear than for some questions, people across the board felt very down the middle. The goal with asking questions for us was to find questions that provided variation in response, leading to graph like this prior two.

It is worth mentioning again how these questions are evaluated as well. Each person's response for how they see themselves as fit to a question is multiplied by the score of what somebody else wants in a complementary question. It is a bit convoluted, and certainly worth revisiting in another round of blind dates, but it's important to keep in mind that every "question" is two questions scores multiplied together. The polarity of each question is listed out in the related code to this paper.

Ridge Regression

Shrinking the least useful variables was clearly crucial for the problem of high dimensionality. Additionally, LOOCV cross validation was used to compensate for the small size sample size, and the model size was limited to 5 using stepwise forward selection based on negative MSE. For the question "Did you enjoy your date?", ridge regression made huge improvements to the poor MSE and R^2 results from multivariate linear regression (without ridge), though not enough to call them useful. The training MSE was 1.049, and the test results were an MSE of 3.08 and an R^2 of -0.066. So still, assuming the average would've been a better guess according to R^2, but clearly the strategic model selection helped drop MSE from 11.65 to 3.08 in a multivariable model. The most significant question by coefficient magnitude, in order, were Q4, Q21, Q38, Q53, and Q8. The first two questions had negative correlation to the results, as is true of most of the questions found in all analysis.

For all of the outcomes tested:

"Did you enjoy your date?"

Training MSE	Test MSE	Test R^2
1.049	3.082	-0.066

"Did your personalities match?"

Training MSE	Test MSE	Test R^2
1.084	4.414	-1.260

"Was there romantic tension?"

Training MSE	Test MSE	Test R^2
0.804	3.596	-1.510

"Did you find them physically attractive?"

Training MSE	Test MSE	Test R^2
2.483	5.464	-0.185

SVM Regressor

To account for nonlinear relationships in the data, I wanted to analyze it with an SVR model using an RBF (radial basis function) kernel. It's a good all-around good kernel, and without a great understanding on the shape of the data I had, using something that worked good in general was the reasonable decision. No feature selection was done, but huge improvements were made to the test MSE for some questions, but then not others. All R^2 scores were still below 0. Unfortunately, there doesn't seem to be enough data to make truly good predictions. For the "Did you enjoy your date?" question again, the SVM returned that Q2, Q23, Q53, Q6, and Q11 were the most important questions. Unfortunately, that doesn't align with the other models well at all, so there is no consistency with the analysis. Tables specifying the scores for different questions are below.

"Did you enjoy your date?"

Test MSE	Test R^2
0.411	-0.665

"Did your personalities match?"

Test MSE	Test R^2
1.979	-0.635

“Was there romantic tension?”

Test MSE	Test R^2
3.137	-0.815

“Did you find them physically attractive?”

Test MSE	Test R^2
6.051	-0.400

Regression Models

The classification models proved to have their own troubles but were a little bit better. The models used were linear discriminant analysis, logistic regression, k-means, and random forest classifiers. All models used an 80/20 training/test split and were standardized.

Linear Discriminant Analysis

As seen with the other linear models, the data from the post-survey doesn’t bode well to linear approximation. However, again to compensate for the small dataset, LOOCV cross validation was used and for interpretation and to stay below the test sample size of nine, feature selection was limited to 5 in the forward stepwise selection of LDA models. The decision for model selection was based on accuracy.

The predictor variable corresponding to each is the following.

- Top left: “Did you want a second date?”
- Top right: “Did you have a second date / is one lined up?”
- Bottom: “Do you expect to be friends with this person?”

Classification Report:

	precision	recall	f1-score	support
0	0.60	0.50	0.55	6
1	0.25	0.33	0.29	3
accuracy			0.44	9
macro avg	0.42	0.42	0.42	9
weighted avg	0.48	0.44	0.46	9

Classification Report:

	precision	recall	f1-score	support
0	0.89	1.00	0.94	8
1	0.00	0.00	0.00	1
accuracy			0.89	9
macro avg	0.44	0.50	0.47	9
weighted avg	0.79	0.89	0.84	9

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.50	0.67	6
1	0.50	1.00	0.67	3
accuracy			0.67	9
macro avg	0.75	0.75	0.67	9
weighted avg	0.83	0.67	0.67	9

Accuracy was over 50% for two of the classification problems, which is a big step up from the regression accuracy. However, the size of each test sample size brings to question the luck behind every pick. For example, the top right outcome heavily skewed everything to the 8/9ths dominating outcome. The dominating questions for the top left were: Q40, Q34, Q14, and Q21.

Logistic Regression

LOOCV was once again used for cross validation of the small set and accuracy was used to determine the outcome in the 5 feature forward stepwise selection model. Logistic regression would hopefully be step one in identifying any nonlinear trends.

The predictor variable corresponding to each is the following.

- Top left: “Did you want a second date?”
- Top right: “Did you have a second date / is one lined up?”
- Bottom: “Do you expect to be friends with this person?”

Classification Report:

	precision	recall	f1-score	support
0	0.75	0.50	0.60	6
1	0.40	0.67	0.50	3
accuracy			0.56	9
macro avg	0.57	0.58	0.55	9
weighted avg	0.63	0.56	0.57	9

Classification Report:

	precision	recall	f1-score	support
0	0.89	1.00	0.94	8
1	0.00	0.00	0.00	1
accuracy			0.89	9
macro avg	0.44	0.50	0.47	9
weighted avg	0.79	0.89	0.84	9

Classification Report:

	precision	recall	f1-score	support
0	0.75	0.50	0.60	6
1	0.40	0.67	0.50	3
accuracy			0.56	9
macro avg	0.57	0.58	0.55	9
weighted avg	0.63	0.56	0.57	9

It seemed that there were some nonlinear trends since Logistic Regression did improve to having all classifications above 50%. Two also happened to be the exact same, but that is by coincidence with the small sample size. The dominating questions for the top left were: Q17, Q20, and Q11. So, no trends in the questions being asked.

K-Means with PCA

. K-Means didn’t show improvement from the logistic regression test, but with more data it could be expected to perform potentially as one of the best. Q15 and Q20 contributed the most to the PCA components.

The predictor variable corresponding to each is the following.

- Top left: “Did you want a second date?”
- Top right: “Did you have a second date / is one lined up?”
- Bottom: “Do you expect to be friends with this person?”

Classification Report:

	precision	recall	f1-score	support
0	0.67	1.00	0.80	6
1	0.00	0.00	0.00	3
accuracy			0.67	9
macro avg	0.33	0.50	0.40	9
weighted avg	0.44	0.67	0.53	9

Classification Report:

	precision	recall	f1-score	support
0	0.89	1.00	0.94	8
1	0.00	0.00	0.00	1
accuracy			0.89	9
macro avg	0.44	0.50	0.47	9
weighted avg	0.79	0.89	0.84	9

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.17	0.29	6
1	0.38	1.00	0.55	3
accuracy			0.44	9
macro avg	0.69	0.58	0.42	9
weighted avg	0.79	0.44	0.37	9

Random Forest Classification

Random forest has the highest potential out of all the classification models to identify nonlinear patterns, however, its complexity got in the way of the classification report for the question “Did you want a second date?” with only a 22% accuracy for example. An example of the first tree in the random forest is provided on the next page.

The predictor variable corresponding to each is the following.

- Top left: “Did you want a second date?”
- Top right: “Did you have a second date / is one lined up?”
- Bottom: “Do you expect to be friends with this person?”

Classification Report:

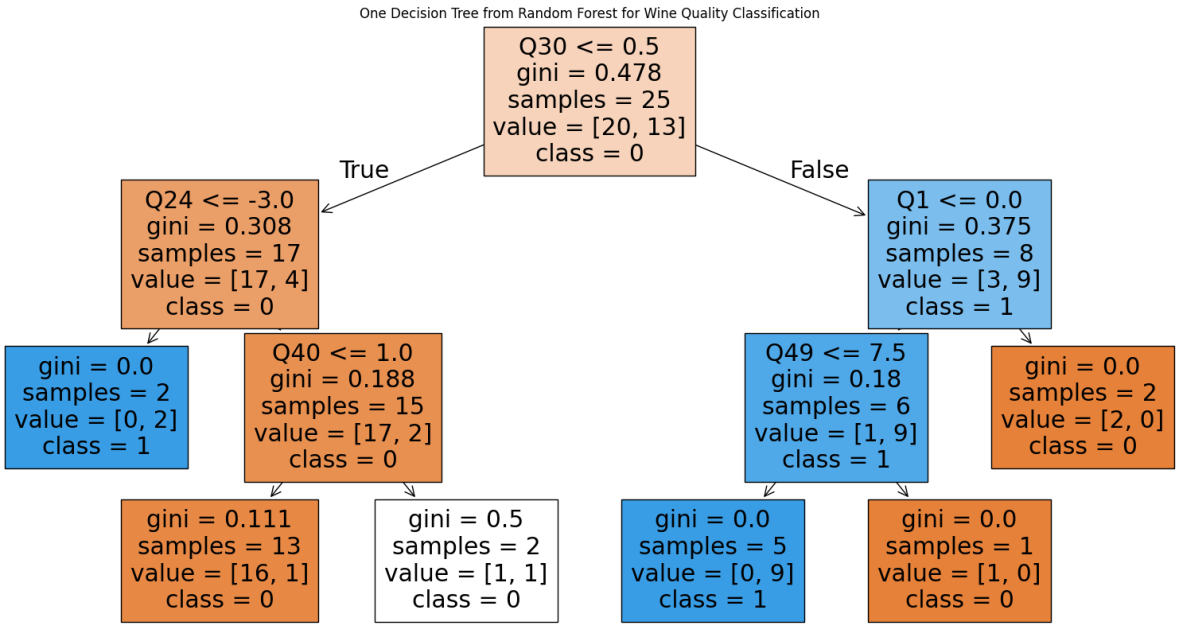
	precision	recall	f1-score	support
0	0.33	0.40	0.36	5
1	0.00	0.00	0.00	4
accuracy			0.22	9
macro avg	0.17	0.20	0.18	9
weighted avg	0.19	0.22	0.20	9

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	8
1	1.00	1.00	1.00	1
accuracy			1.00	9
macro avg	1.00	1.00	1.00	9
weighted avg	1.00	1.00	1.00	9

Classification Report:

	precision	recall	f1-score	support
0	0.57	1.00	0.73	4
1	1.00	0.40	0.57	5
accuracy			0.67	9
macro avg	0.79	0.70	0.65	9
weighted avg	0.81	0.67	0.64	9



Conclusions and Takeaways:

Clearly, the data available for the blind dating project is too small for proper analysis, however, there are still improvements to make the bad a little less bad with the right model. I believe that if this event is to happen again, I would seek out nonlinear models to analyze the data, as well as collect more meaningful, yet smaller amounts of data. There isn't a takeaway for the most common Questions like I was hoping for. Surveying people beforehand as to what is most important to them may be the most viable way to find out what those questions are though.

Logistic Hurdles:

Many of the challenges we had during the project pertained to a quick timeline, short notices sent to participants, and poor data collection techniques that made the code tedious and impractical.

Luckily, since our new methods have already been discussed and investigated, such as cosine similarity and maximum weight matching, timing is expected to be less of an issue. However, we do plan to have our code written before a survey is sent out to participants next time. This would also solve the issue of short notices and people dropping out.

Regarding data collection, our use of a Google Form created several problems. When changing questions for word choice, we learned it would create an entirely new column in the resulting Google Sheet, which led to an unnecessarily large spreadsheet with empty columns. Also, when making questions, there wasn't a convenient way to pair dual questions to each other aside from directly indexing them. This is understandably poor practice and solutions such a key

identifier in front of the question (such as [A1] and [A2] for dual question A) have already been proposed to solve this problem and make survey expansion easier.

Additionally, to promote having more blind dates, we felt that we could remove some dealbreakers. For example, we could likely remove a category such as height, while offering an alternative to women who desire a tall man by simply asking if their partner should be taller than them or not. At the end of the day, this is a blind dating service.

Sources:
https://www.youtube.com/watch?v=BKwKRIUKv64