

CS4641-Team81Site

Project Proposal: Modeling Potential Dating Suggestions Using Machine Learning

Introduction/Background:

Literature Review

Our goal is to create a recommendation system for dating profiles. By analyzing user data and machine learning algorithms, we can optimize compatible matches. In "Is Romantic Desire Predictable?" researchers used Random Forest models to forecast through user preferences without increased risk of overfitting/collinearity[1]. "Finding Love on a First Date: Matching Algorithms in Online Dating" highlights how machine learning and collaborative algorithms can solve the problem of choice overload, but poses risk of demographic bias[2]. "Matchmaking Under Fairness Constraints", utilizes speed dating databases and Knapsack /Tabu Search to promote preferential fairness. Logistic Regression performed the best with an F1 score of 0.2566 with the Knapsack technique. Accuracy decreased slightly, but fairness rating increased by 50% [3]. Balancing preference accuracy and user demographic distribution is key. With proper data preprocessing and re-ranking methods, we can produce accurate and fair matches without bias.

Dataset of Interest

As mentioned in the "Matchmaking Under Fairness Constraints", the Speed Dating Experiment dataset captures information about participants in speed dating events, including their demographic data and preferences. It also provides a binary match label of if a player matched with another. The dataset contains 8,000 instances of dating interactions. However the dataset is imbalanced, having 16% of interactions being matched, calling for cost-sensitive learning to address the issue.

Problem Definition:

Online dating users often feel overwhelmed by the sheer number of profiles, making it hard to efficiently find compatible matches. Machine learning can enhance this process by offering personalized dating profile recommendations. A model could be developed using datasets with attributes like demographics (age, gender, race), personal interests, and personality traits.

Methods

Online dating users often feel overwhelmed by the sheer number of profiles, making it hard to efficiently find compatible matches. Machine learning can enhance this process by offering personalized dating profile recommendations. A model could be developed using datasets with attributes like demographics (age, gender, race), personal interests, and personality traits.

Data Preprocessing Methods

Some methods we explored are handling missing data, data normalization, and text processing. Incomplete data can be addressed with scikit-learn's SimpleImputer, replacing missing values with the mean for numerical data and the mode for categorical data. Additionally, since features like age, height, and short-response lengths are scaled differently, data normalization ensures that feature disproportionality doesn't impact performance. This would be done using scikit-learn's MinMaxScaler scaling values (0-1). Text processing can be used to transform open-end responses into usable vectors with NLP techniques like TfidfVectorizer.

Implemented Data Preprocessing:

In developing our online dating profile recommendation system, we first focused on data preprocessing to ensure model accuracy and reliability. We selected key features related to age, self-rated attributes (such as attractiveness and intelligence), and partner-rated attributes. The dataset contained missing values, so we implemented imputation by filling missing numerical values with the mean and categorical values with the mode. This preprocessing step helped create a cleaner dataset, allowing for a more robust analysis. Additionally, categorical data were transformed to numeric codes, and new compatibility features were created by calculating the absolute differences between participants' ratings and their partners' ratings. These compatibility features aimed to capture essential alignment indicators between individuals, adding an informative layer for clustering.

Machine Learning Models/Algorithms:

K-Means Clustering

This unsupervised learning method helps group users with similar interests and characteristics, identifying compatible clusters. It can serve as a useful pre-processing step before applying supervised models, enhancing match recommendations within clusters.

Logistic Regression

Logistic regression can predict whether two users are likely to be a match based on their profile attributes. This model is effective for binary classification problems and offers interpretability by

showing the weight of each feature on the matching decision.

Gaussian Mixture Model (GMM)

The Gaussian Mixture Model provides a probabilistic clustering approach that can accommodate overlapping clusters, allowing for softer clustering of users. Unlike K-Means, GMM captures the variance and covariance in data, enabling the model to represent users with a probability of belonging to multiple clusters. This flexibility can improve recommendations by acknowledging users' diverse characteristics and shared traits across clusters.

Implemented Model 1: K-Means Clustering:

For the machine learning model, we used the K-Means clustering algorithm to group participants into distinct compatibility clusters. We standardized the data with StandardScaler to ensure uniform scaling, preventing features with larger ranges from dominating the clustering process. To reduce dimensionality and enhance interpretability, we applied Principal Component Analysis (PCA), retaining components that captured 95% of the data's variance. K-Means clustering was then performed on the transformed data, assigning participants to five distinct clusters. This choice of an unsupervised method allowed us to explore natural groupings in the data without predefined labels, providing valuable insights into dating preferences and compatibility patterns. Visualizing the clusters on the principal components further highlighted distinct groups, confirming the model's effectiveness in identifying compatibility trends across user profiles.

Implemented Model 2: Logistic Regression:

For our Logistic Regression implementation, we processed the Speed Dating Experiment dataset to ensure it was clean and balanced for accurate modeling. Key numerical features like self-rated and partner-rated attributes were scaled using a StandardScaler, while categorical features such as gender and race were transformed using one-hot encoding. To address class imbalance, we applied SMOTE to oversample the minority class, creating a more balanced dataset. Additionally, we calculated and incorporated class weights to further enhance model performance on imbalanced data. The Logistic Regression model provided interpretable coefficients for feature importance and was evaluated using metrics such as precision, recall, F1-score, and support. These evaluations highlighted the model's ability to predict matches while maintaining fairness across different demographic groups.

Implemented Model 3: Gaussian Mixture Model (GMM):

For our Gaussian Mixture Model (GMM) implementation, we focused on clustering participants based on their self-rated attributes like attractiveness, sincerity, intelligence, fun, and ambition. The data was standardized using a StandardScaler to ensure all features were on a similar scale, enabling the GMM to identify meaningful clusters. We set the number of components to three and

trained the GMM to assign participants to probabilistic clusters. Visualizing the clusters in two dimensions showed distinct groupings, which offered insights into compatibility patterns.

Results and Discussion

To measure the performance of the model, we plan on tracking the accuracy through labeled matches. The dataset we plan on using has a set of characteristics for each person and boolean match variable. Once the model makes its decision and suggests a person, the main candidate and the suggestion will be read to see if it is a true or false match. This data will be tracked and graphed to determine the accuracy of the model.

The expected goal for our model will be around 75%, with the current level of research and the dataset. Throughout the project, we aim to increase the accuracy to above 85%.

Post-Midterm:

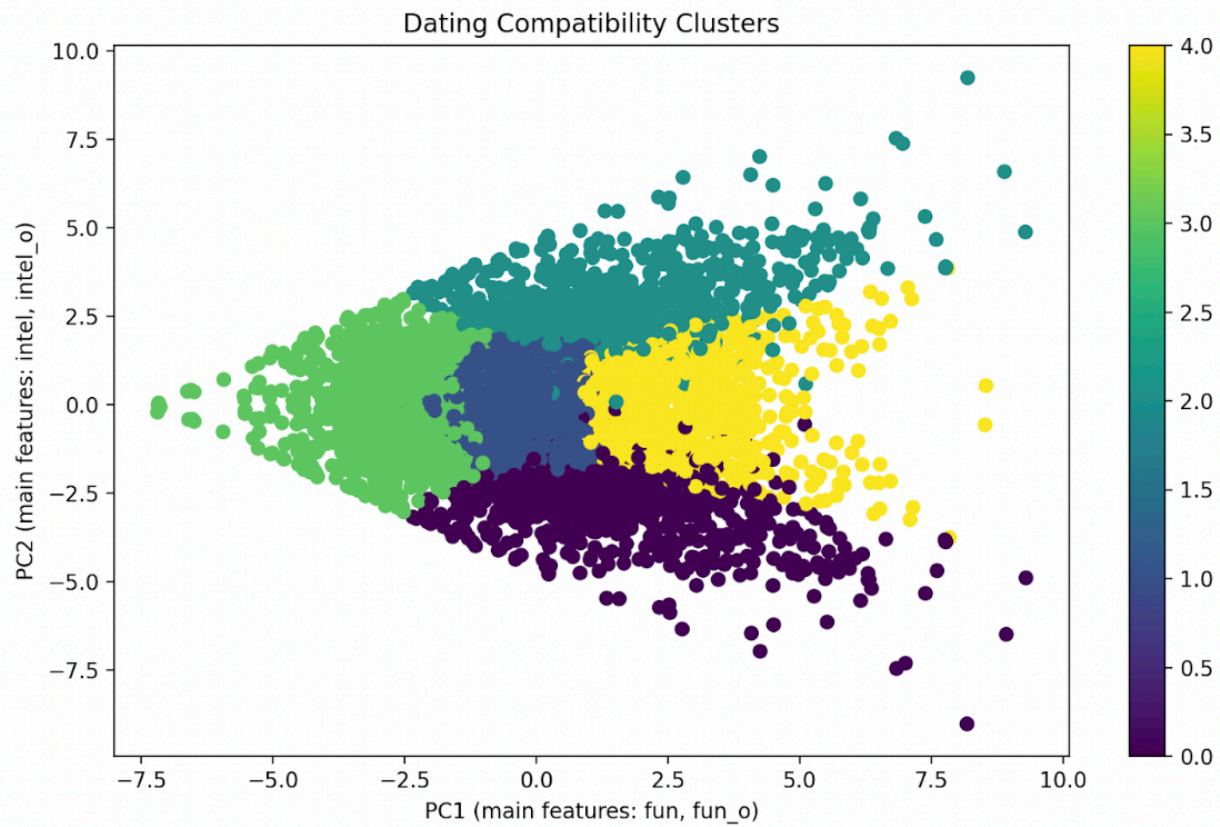


Figure 1: Visualization of K-Means clustering with 5 clusters based on the 2 most significant features, the preferred fun rating and the preferred intelligence rating.


```

Model Evaluation Metrics:
Silhouette Score: 0.066
Calinski-Harabasz Score: 758.628
Inertia: 129779.782

Cluster Sizes:
1      3046
3      2020
4      1268
2      1023
0      1021
Name: count, dtype: int64

Inter-cluster Distances:
[3.93206409 5.34726617 4.78080449 4.00273533 3.93632194 2.38535402
 2.81218201 4.77358042 4.01667319 5.09049813]

Within-cluster Variances:
Cluster 0: 22.157
Cluster 1: 12.199
Cluster 2: 22.134
Cluster 3: 13.438
Cluster 4: 15.938

```

Figure 2: *Quantitative Performance Metrics of K-Means clustering model.*

K-Means Clustering:

Visualization Analysis:

From Figure 1, we observe five distinct clusters, representing five groups of dating compatibility. The first principal component suggests the most variance, and thus the primary factor in dating compatibility. In addition, reasonable distribution of points in the chart demonstrates good representation of profile types across our data. However, one issue seen within the K-Means model performance is displayed through the overlapping cluster boundaries, which indicate that some profiles may share multiple features and could be part of multiple groups.

Quantitative Metrics Analysis:

Figure 2 displays several different types of metrics we calculated to evaluate model performance. We utilized the Silhouette Score to measure how well defined clusters are and how distinct they are from each other. For each point, the silhouette score is calculated as: $S = \frac{b-a}{\max(a,b)}$ where a is the mean distance between an object and all other points in the same class and b is the mean distance between an object and all other points in the nearest neighboring clusters. As a result, the Silhouette Score measures both how similar a point is to its own cluster as well as how

different a point is to other clusters. In addition, it also could serve as an indicator of cluster assignment accuracy.

The Silhouette score has a range of $[-1, 1]$, where higher values indicate higher levels of separation and lower values indicate poor cluster separation. Optimal performance targets higher values as that indicates good cohesion(inter-cluster similarity) and good separation (clusters are well-defined from each other). Our score of 0.066 suggests that we have moderate cohesion and separation, where there is average similarity between points within clusters and little separation between clusters. Another metric we utilized was the Calinski Harabasz Score, which measures the ratio of inter-cluster dispersion and between-cluster dispersion. Similar to the Silhouette Score, the C-H Score measures cohesion and separation, but rather than focusing on point-specific fit like the Silhouette Score, the Calinski Harabasz score utilizes cluster density and separation. The C-H score we calculated from our model was 758.632, which is a moderately high score. However, the Calinski Harabasz score is highly dependent on factors such as cluster size and variance. Therefore, it must be interpreted alongside these variables. We observe that our cluster sizes are highly uneven with one cluster(3046) being almost 3x the size of the other clusters. In addition, our inter-cluster distances are relatively small indicating low separation between clusters. Uneven cluster size distribution, high within-cluster variance, and low-inter-cluster distance all are variables that can impact the Calinski Harabasz Score. The final metric we calculated was the inertia score which measures the distance between a point and its centroid. Optimal clustering can be seen through compact clusters, meaning lower inertia scores indicate better model performance. The inertia score of our model was 129779.2, which is relatively high. This indicates that our clusters are not very compact.

Model Analysis:

Overall, our model performed moderately well, however, the clustering could be improved. The model does not create well-defined clusters and there is inconsistency in the size of the clusters. KMeans clustering operates on the assumption of feature independence and linear relationships between variables. However, compatibility features are interconnected and clear and separated clusters are difficult to define into natural clusters.

Next Steps:

The next steps we will take for this project are to implement Logistic Regression and a Gaussian Mixture Model. Logistic Regression will be more accurate in predicting whether two individuals are a match. In addition, Logistic Regression will be able to represent combinations of compatibility features and how this leads to the binary choice decision(Whether or not the matched couple wants to see each other again). GMM will be useful to handle users that have traits belonging to multiple clusters, by taking advantage of those shared characteristics.

Post-Final:

Logistic Regression:

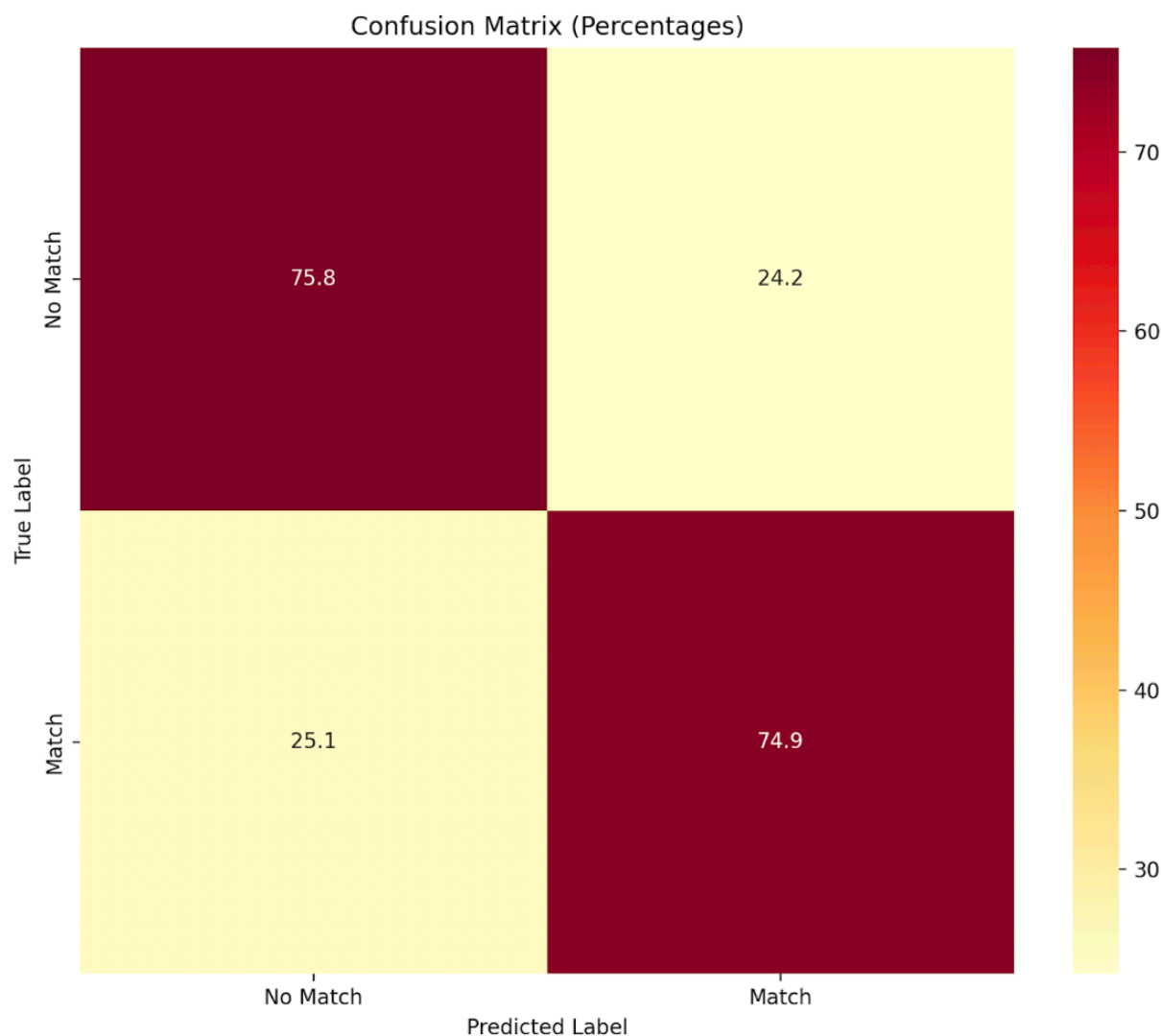
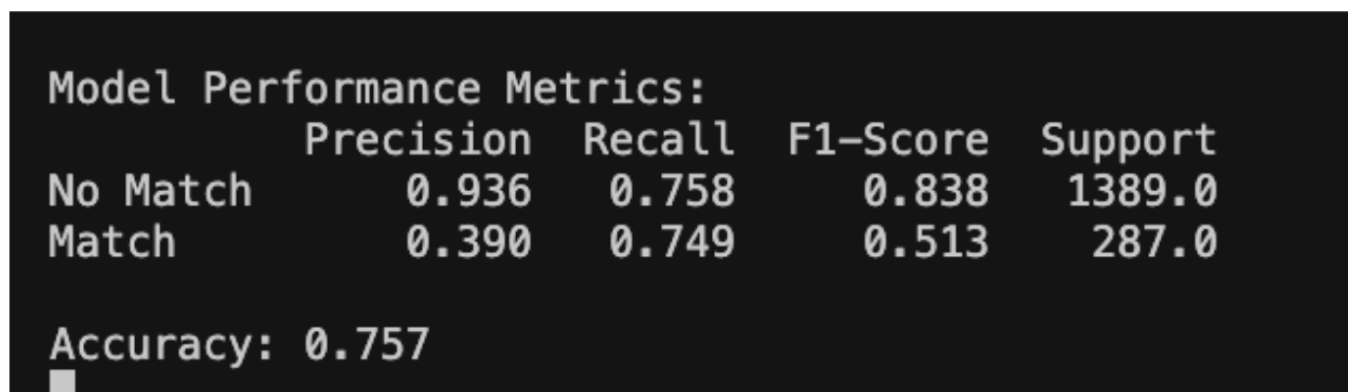


Figure 3: Visualization of a Logistic Regression Confusion Matrix showing the proportion of accurate predictions for the “Match” and “No Match” classifications.

Visualization Analysis:

In Figure 3 we can see the recall, or the percentage of correct and incorrect classifications for the “Match” and “No Match” classes. 75.8% of “No Match” predictions and 74.9% of “Match” predictions were accurate. The non-match score represents decently strong levels of non-match detection, and had slightly lower levels of Match identification. This means that the model is most accurate at predicting non-matches. It is still reasonably able to predict matches, but the difference in accuracy may be due to an imbalance in the dataset where there are more non-matches than matches. Higher levels of recall are important for preventing false positives and false negatives for our dataset. For our dataset, a higher non-match recall rate (False Positive) identified 75.8% of incompatible pairs which prevents high rates of potentially unsuccessful matched pairs. Similarly, our relatively strong “Match” (False Negative) score identified 74.9% of

"Match" predictions, successfully optimizing relatively high rates of compatible matches. Unfortunately, for both recall rates, this also means that 24.2% of "Non-match" pairings were incorrectly predicted as compatible as well as 25.1% "Match" pairings were incorrectly identified as incompatible. However, in the context of romantic compatibility, these scores are optimally high for recall rates.



Model Performance Metrics:				
	Precision	Recall	F1-Score	Support
No Match	0.936	0.758	0.838	1389.0
Match	0.390	0.749	0.513	287.0
Accuracy: 0.757				

Figure 4: *Quantitative Performance Metrics of the Logistic Regression model.*

Quantitative Metrics Analysis:

Figure 4 shows some of the Performance Metrics we can use to assess the performance of the model. The precision of correctly predicting a "No Match" was 93.7%, while the precision of correctly predicting a "Match" was 39.0%. This indicates that our model is more likely to inaccurately classify a match, since the precision is so low. We also utilized the F-1 score, which is the harmonic mean of precision and recall, to evaluate the accuracy of the model. A higher F-1 score suggests the model has a better performance, with the highest score being 1. The F-1 score for "No Match" was 0.838 and for "Match" was 0.513. This indicated the model performed much better while predicting the nonmatches than predicting matches. Another metric we used was Support, which counts the number of times each class appears in the dataset. This helps detect imbalance in the dataset, which can indicate the reliability of the other metrics. There are 1389 instances of nonmatches in the dataset, while there are only 287 instances of matches. This highlights a clear class imbalance, given that there are almost 5 times more occurrences on "No Match" classifications than "Match" classifications in the dataset. This imbalance explains why our other metrics are higher for "No Match" classification, because there is more of that data to train the model. The overall calculated accuracy of our model was 75.7%. This is a reasonable accuracy for our model, however it could definitely be improved by reducing the imbalance in our data.

Model Analysis:

Overall this model performed relatively well, however the accuracy could be improved by handling the class imbalance between "Match" and "No Match" data. The model was very precise when identifying pairs that were not a match. It was also moderately accurate when identifying pairs

that were a match, but there was a clear difference with the model's performance when handling matches and nonmatches. The main limitation of the model was the class imbalance in the data, since there was much more "No Match" data. This is why it was more accurate when handling nonmatch data. Logistic regression also mainly searches for a linear relationship between the features of the dataset, however this may not be the most optimal way to create matches dependent on multiple features.

GMM:

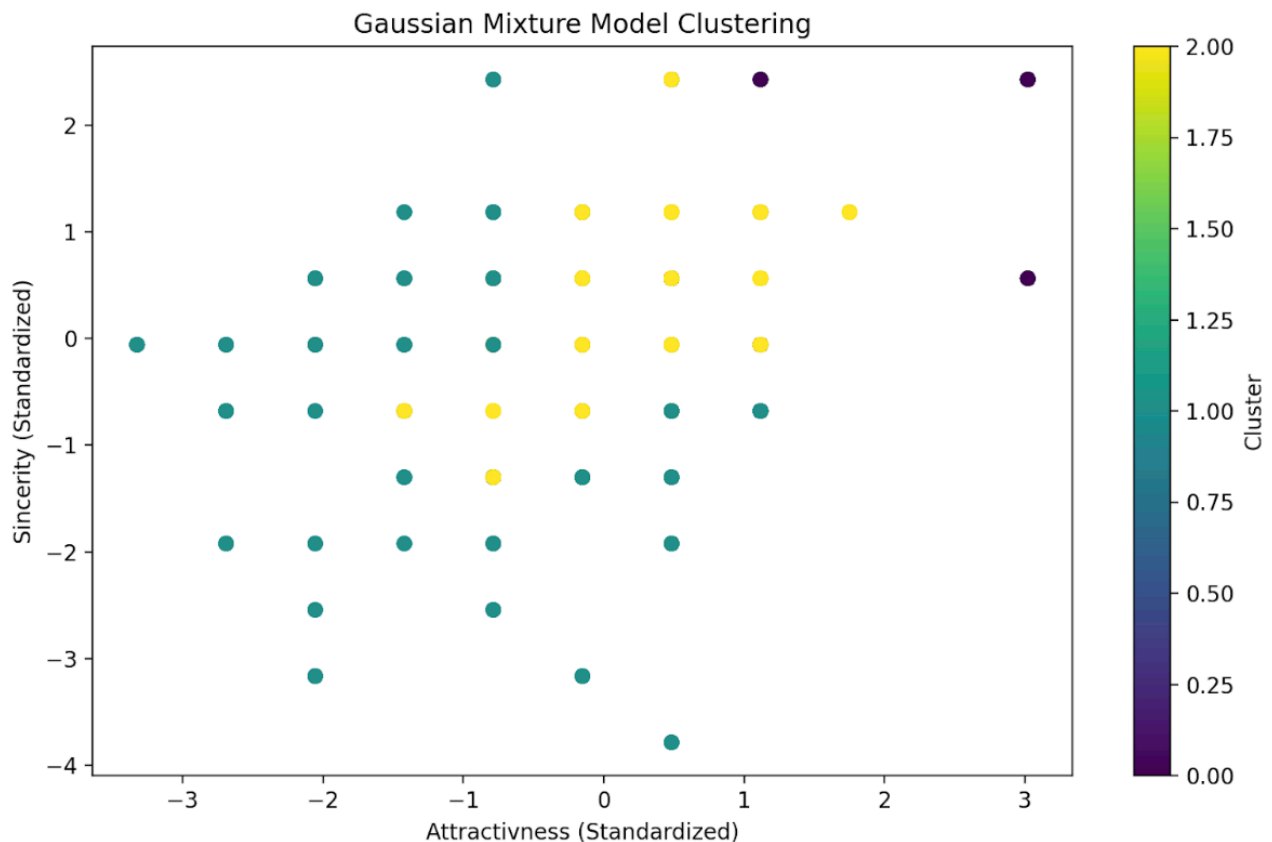


Figure 5: Visualization of Gaussian Mixture Model clustering with 3 distinct clusters based on the two most significant features, sincerity and attractiveness.

Visualization Analysis:

In Figure 5, we observe 3 distinct clusters, which represent the different groupings of dating preferences. The axes are defined to be the most significant attributes of Attractiveness & Sincerity, and the clusters grouped the sample data into how people determined compatibility based on these two factors. While we see distinct clustering, there is significant boundary overlap which demonstrates "borderline" cases where dating preferences are mixed. In our dataset, negative values on the attribute axes represent lower than average rating, 0 value represents average rating, and positive values represent higher than average ratings. Therefore we can define our clusters as teal(harsher raters), yellow(neutral raters), and purple(optimistic raters). In the visualization we see a pretty wide range of cluster sizes with the teal cluster being the largest

and densest and the purple and yellow clusters being more sparse. The teal cluster by far has the densest positioning, which represents stronger pattern identification. This means that the model is able to consistently identify more critical raters while having higher prediction accuracy for outcomes of critical raters. In addition, a larger teal cluster indicates more consistent classification for harsher raters. In comparison, sparser and smaller cluster sizes for the purple cluster (Optimistic Raters) potentially indicate smaller sample size, and thus smaller confidence for prediction and classification.

```
Model Evaluation Metrics:  
Silhouette Score: 0.249  
Calinski-Harabasz Score: 788.445  
Converged: True
```

Figure 6: *Quantitative Performance Metrics of the Gaussian Mixture Model*

Quantitative Metrics Analysis:

In Figure 7, we observe the different evaluation metrics we calculated to evaluate the performance of the model. We used the Silhouette Score to assess the definition of the clusters. The Silhouette Score of this model was 0.249, which indicates that the clusters are moderately well defined. However, there is some overlap within the clusters and little separation between the groups. We also calculated the Calinski-Harabasz Score to evaluate the cohesion and separation of the clusters. The Calinski-Harabasz Score was 788.445. This is a relatively high score, which suggests that there is not very much variance within each cluster, but moderately good variance between the different clusters. We can also see that the model successfully converged, meaning it reached a stable solution. This indicates the data is accurate and the results are meaningful.

Model Analysis:

Overall, this model performed moderately well, however, the clustering could be improved. The model does have well-defined clusters, and is able to identify 3 clear groups of people representing dating preferences in the sample data (harsh raters, moderate raters, and optimistic raters) for the sincerity and attractiveness attribute. However one weakness in our Gaussian Mixture Model is a variable amount of uncertainty between 2 clusters (harsh and moderate raters) as well as very high uncertainty for one of our clusters (optimistic raters). Gaussian Mixture Model allows for cluster overlap which can demonstrate more complex relationships/dating preferences as well as flexible clustering which can more accurately represent data points that can belong to more than one cluster.

Comparing the Models:

The three models implemented—K-Means Clustering, Logistic Regression, and Gaussian Mixture Model (GMM)—all offer unique strengths and address different aspects of the recommendation system for online dating profiles. K-Means Clustering acts as an unsupervised learning model to discover groupings within the data. By standardizing features and reducing dimensionality with PCA, K-Means highlights distinct compatibility clusters. This model helps us understand the data by grouping different outcomes of speed-date match-ups into defined attribute clusters. However, as seen in the silhouette score of 0.066, the compactness and separation between clusters are not optimal, suggesting that K-Means struggles to cleanly separate user groups, particularly in cases of overlapping preferences. Because it is the most simple, it is not able to represent overlapping user preferences.

Logistic Regression, on the other hand, is a supervised binary classification model focused on predicting matches. By addressing class imbalance through SMOTE and weighting, it provides interpretable insights into feature importance while maintaining fairness across demographic groups. For example, metrics like precision, recall, and F1-score across both matched and unmatched classes demonstrated the model's ability to handle imbalanced data while achieving a balanced evaluation of prediction outcomes. This model is particularly valuable for understanding individual attributes' contributions to match outcomes but lacks the flexibility to explore the nuances of shared traits or overlapping preferences among users.

The GMM is a midpoint between the clustering and classification models, providing probabilistic clustering by grouping users while accounting for feature variance and covariance. Compared to K-Means, GMM provides better insights into shared user traits, capturing overlapping and probabilistic relationships between clusters. However, this model sacrifices the simplicity of deterministic clustering for greater interpretability of shared traits. Metrics like the Calinski-Harabasz score of 788.445 and a Silhouette Score of 0.249 indicate well-separated clusters despite some overlapping characteristics. Both of these metrics are higher than that of K-Means, meaning it has better separation and higher intra-cluster density. Furthermore, the model converged successfully demonstrating its stability in optimizing cluster assignments.

Next Steps:

The next steps involve improving the Logistic Regression model by fine-tuning hyperparameters and adding compatibility features to enhance predictive accuracy. The Gaussian Mixture Model (GMM) will be optimized by testing different numbers of components and evaluating configurations using metrics like BIC and AIC. A comprehensive comparison of the three models—K-Means, Logistic Regression, and GMM—will be conducted to assess performance on metrics like accuracy, fairness, and interpretability. These efforts aim to refine each model and integrate their strengths into a robust recommendation system.

References

- [1]S. Joel, P. W. Eastwick, and E. J. Finkel, "Is Romantic Desire Predictable? Machine Learning Applied to Initial Romantic Attraction," *Psychological Science*, vol. 28, no. 10, pp. 1478–1489, Aug. 2017, doi: <https://doi.org/10.1177/0956797617714580>.
- [2]L. L. Sharabi, "Finding Love on a First Date: Matching Algorithms in Online Dating," *Harvard Data Science Review*, no. 4.1, Jan. 2022, doi: <https://doi.org/10.1162/99608f92.1b5c3b7b>.
- [3]D. Paraschakis and B. J. Nilsson, "Matchmaking Under Fairness Constraints: A Speed Dating Case Study," *Communications in Computer and Information Science*, pp. 43–57, 2020, doi: https://doi.org/10.1007/978-3-030-52485-2_5.

Gantt Chart

Contribution Table

Name	Proposal Contributions
Daniel	Introduction/ literature review / Video
Emily	Introduction/ literature review
Nikita	Problem Definition/ Methods / Slides
Aveek	Results & Discussion/ Video/ Github Repository & Page
Ria	Gantt Chart/ Video

Post-Midterm

Name	Proposal Contributions
Daniel	Model/Visual
Emily	Visuals Explanation
Nikita	Model Explanation
Aveek	Model/Visual
Ria	Visuals Explanation

Post-Final

Name	Proposal Contributions
Daniel	Model/Visual
Emily	Visuals Explanation
Nikita	Model Explanation
Aveek	Model/Visual
Ria	Visuals Explanation