CS M148 Project: What Traits Do Dating App Users Value Most in a Potential Partner?

Group ID: 20 (Original) / 18 (Presentation Order)

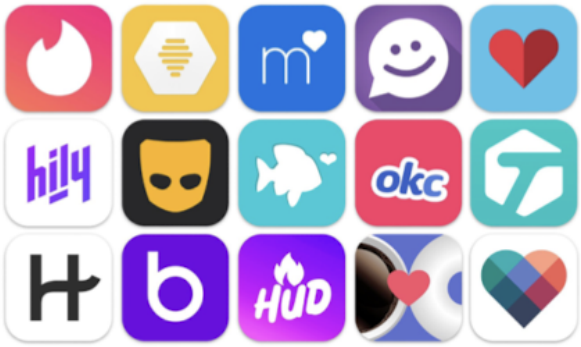
Nick Cuenca
University of California, Los Angeles
Los Angeles, California, USA
nwcuenca@g.ucla.edu

Ethan Diana
University of California, Los Angeles
Los Angeles, California, USA
ethandiana@g.ucla.edu

Grant Gilchrist
University of California, Los Angeles
Los Angeles, California, USA
gilchrist@g.ucla.edu

Samuel Levy
University of California, Los Angeles
Los Angeles, California, USA
sdlevy@g.ucla.edu

Ofri Mayer
University of California, Los Angeles
Los Angeles, California, USA
ofrimayer@g.ucla.edu



Abstract

The goal of our project is to predict the traits dating app users value most in a potential partner, examining the specific characteristics of gender, income, age, number of children, and attractiveness level, as well as the user's VIP status on the app. This topic is highly relevant because dating apps are widely used today, and their matching methodologies can have significant short and long term effects. These apps rely on effective algorithms to recommend suitable matches, which is very necessary if they want users to have a positive app experience, high satisfaction, and ultimately, successful matches. Our study uses a number of data processing techniques, as well as model development using Logistic Regression, Random Forest, SVM, and Neural Network algorithms. Our findings indicate that the Random Forest model outperforms the rest; this report details the complete methodology behind these findings and the effectiveness of our various models.

CS M148 Project, 2024, University of California, Los Angeles

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Nick Cuenca, Ethan Diana, Grant Gilchrist, Samuel Levy, and Ofri Mayer. 2024. CS M148 Project: What Traits Do Dating App Users Value Most in a Potential Partner?: Group ID: 20 (Original) / 18 (Presentation Order). In *Proceedings of CS M148 Project*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction of the Overall Goal and Background

The design of dating apps often encourages people to look for partners based on only a few basic factors. On apps such as the one which produced the data for this project, people take a look at another person's profile and can choose to 'match' with the person if they are interested in them. Profiles are often made up of a set of personal photos with descriptions as well as key information such as age, gender and general location. Due to confidentiality constraints, the only pieces of information available in our dataset were age, gender, income, number of children, attractiveness, and VIP status on the app (this combination of traits alone does not compromise the privacy of the users the data represents). While many dating apps allow users to input information relating to individual hobbies and activities, it makes sense to reason that in this fast paced environment where matches can be initiated in rapid succession, the most important/valued factors are the more overarching ones, like income and attractiveness. The objective of our project is to develop a predictive model that estimates the number

of matches a user might receive based solely on these overarching factors (age, gender, income, number of children, attractiveness, and VIP status on the app). Being able to understand the general trends in people's priorities in choosing partners online can likely provide many valuable insights that can help dating apps improve their recommendations. Understanding these trends could also be useful in studying the ways in which online dating has changed how people form relationships.

2 Problem Definition and Formalization

Our problem can be formalized as a regression task, where our target variable is the number of matches, and our predictors are the demographic and personal characteristics of the dating app users. We aim to identify the strength of each of these attributes in contributing to the number of successful matches.

3 Data Preparation and Preprocessing

Firstly, we load our dataset from the CSV file and split it into the features (X) and target variable (y). We then check for any missing values in the dataset, but fortunately, our dataset is shown to have no missing values; each person has complete data for all features. In order to focus only on "meaningful data," we also remove rows of the data where the "Matches" value is zero, as the distributor of the dataset stated that these people were usually inactive users. We also utilize bootstrapping to generate additional samples in our data, as the dataset doesn't give as large or robust of a dataset as we want. Bootstrapping is done by resampling the data with replacement, adding slight variations, and then taking our newly augmented datasets and combining them with the original for a larger and more robust dataset. We repeat this process 5 times. Next, we split the data into training, validation, and test sets; first, we split into a training set (70%) and a temporary set (30%). The temporary set is further split equally into validation and test sets (so 15% each). This is done to give us separate data for evaluating the model's performance during training, as well as to prevent overfitting.

For data preprocessing purposes, we define numeric features (Income, Children, Age, Attractiveness) and categorical features (PurchasedVIP, Gender). We create a column transformer that will apply different preprocessing steps to the features mentioned above – we want to normalize our quantitative features, and then one-hot encode our categorical features.

We normalize our quantitative features to the range [0,1], utilizing min-max scaling (Figure 1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Figure 1: Min-Max Scaling Formula

This formula is given X as the original value to be scaled, and X_min and X_max the minimum and maximum values of the feature, respectively. X' is our newly normalized value. We perform min-max scaling particularly to prepare the data for our models that are sensitive to differences in feature magnitude, such as Logistic Regression and Neural Networks. It's also important for our SVM model, because since it's distance based, we need to prevent large features from dominating the model. Min-Max scaling is very sensitive to outliers, but fortunately our data contains very few outliers, and none so severely large/small as to warrant treatment. For the handling of our two categorical features "PurchasedVIP" and "Gender", the features are essentially one-hot encoded for us already. For "PurchasedVIP", 0 represents not having bought a VIP subscription, and 1 represents having bought a VIP subscription. For "Gender", it's 0 for male, and 1 for female; we can think of this as a one-hot encoding of "is_female". Finally, we define the models we will use for experimentation of which traits are preferred: Logistic Regression, Random Forest Regression, Support Vector Machine (SVM), and Neural Network.

4 Methods Description (Detailed Steps)

Our project focused on building predictive models to estimate the number of matches a dating app user might receive based on their demographic and personal characteristics.

As detailed in the previous section, we began by collecting and preparing our dataset from Kaggle, which included critical user information (specifically their age, gender, income, number of children, attractiveness, and VIP status). We ensured the data was clean and meaningful by filtering and normalizing it appropriately. Specifically, we removed rows where the "Matches" value was zero to focus on the actual meaningful data. More information on this is provided in the previous section.

Afterwards, we selected four models for our experiments: Logistic Regression, Random Forest Regression, Support Vector Machine (SVM), and Neural Network. We chose these models due to their diverse strengths: Logistic Regression for its simplicity and interpretability, Random Forest for its robustness and ability to handle nonlinear relationships, SVM for its effectiveness in high-dimensional spaces, and Neural Network for its capability to capture complex patterns. We reasoned that since each of these models are known to perform very well for certain use cases, we could likely be successful if we tried all of them to see which one was best for this application.

To make sure we had a consistent and efficient workflow, we created a pipeline for each model. These pipelines integrated data preprocessing steps with the model training process. Preprocessing involved normalizing numeric features and encoding categorical features (as described in the data preparation section).

Each model was trained using the training dataset. During the training process, the models learned to predict the number of matches based on the input features.

We evaluated the performance of each model using Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), Accuracy, Recall, F1, and Precision, on the validation dataset. These metrics helped us determine how well the models were performing overall, in order to help us choose which model to eventually select.

For the best performing model, which was the Random Forest Regressor, we analyzed feature importance to understand the impact of different features on the predictions. This analysis brought many insights into which characteristics were most influential in predicting the number of matches.

We assessed the performance of the Random Forest model on the test using the same evaluation metrics (MSE, MAE, R^2, Accuracy, Recall, F1, Precision) to ensure that it generalizes well to unseen data. This step was extremely important for confirming the model's effectiveness in predicting new scenarios.

Throughout the process, we generated many visualizations to interpret the model's performance and the significance of different features (and to help communicate some of our findings to the audience for our presentation). Most of these visualizations were either feature importance charts or error distribution graphs.

5 Experiments Design and Evaluation

As we mentioned previously, we trained and evaluated several different models and compared their performance. Our train_evaluate_model function, as its name suggests, was responsible for training and evaluating our models. It combined preprocessing with model training by creating a pipeline. The pipeline first applied preprocessing steps to the data and then fitted the specified model. The function then fitted the pipeline to the training data, made predictions on the validation data, and calculated all the evaluation metrics (listed in the previous section). These metrics provided a solid overall view of our model performance, as we used a sufficient number of differing metrics to avoid individual metrics' oversights being weighted enough to encourage a poor model choice. Then, the code iterated over each model, trained and evaluated it, and stored the results, including the evaluation metrics and the trained pipeline, for each model. This was done to make sure the different models are compared on the same basis. To determine the best model, we selected the model with the best overall evaluation metrics. This ensured that the chosen model was the most effective at capturing the underlying patterns in the data. Feature importance was then analyzed to understand which features were most influential in the model's predictions.

6 Evaluating and Comparing Algorithms

Each algorithm was evaluated using the same 7 metrics: R^2, MAE, MSE, Accuracy, Precision, Recall, F1. The table below shows all of the following values with an additional table below containing bar plots of each metric ranked against every other model's metric. See Table 1 and Figures 2-8.

Table 1: Evaluation Metric Comparison

Model	R^2	MAE	Acc.	MSE	Precis.	Recall	F1
LR	-0.18	2.38	0.23	12.20	0.13	0.18	0.15
RF	0.96	0.52	0.51	0.39	0.51	0.50	0.50
SVM	0.16	1.75	0.34	8.61	0.31	0.39	0.30
NN	0.44	1.89	0.17	5.76	0.14	0.12	0.12

Based on all the metrics, the Random Forest performed the best. It boasted the lowest errors for both MSE and MAE. It also performed the best in terms of R^2, accuracy, precision, recall and F1. While it is indeed the best performing model, its very high R^2 score and other similarly strong evaluation metrics might indicate that either some overfitting is happening, or perhaps the dataset is somewhat

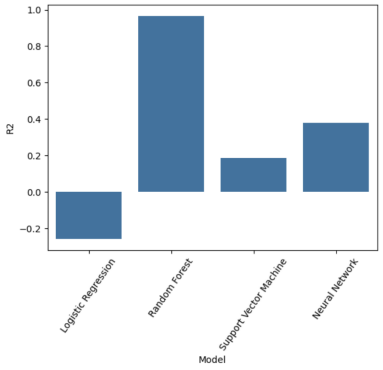


Figure 2: R^2 Performance

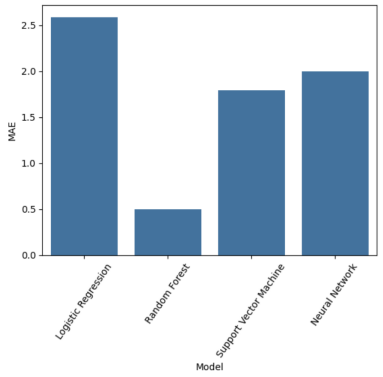


Figure 3: MAE Performance

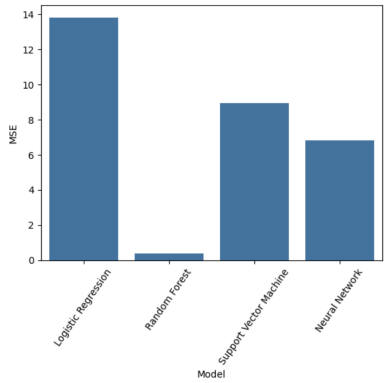


Figure 4: Accuracy Performance

limited and may not necessarily generalize to other dating apps well. Also, the decent (but not outstanding) precision and recall scores for Random Forest do suggest some imperfections in this model's performance.

The bar chart shown in Figure 9 displays the feature importance for the random forest model.

It is clear that the random forest heavily emphasizes a high level of attractiveness along with being female in predicting how

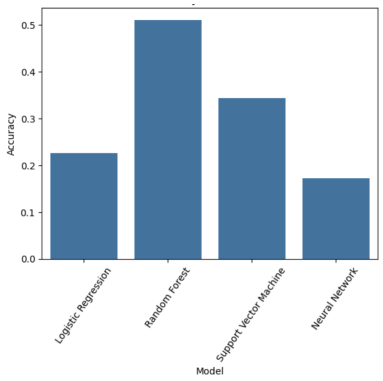


Figure 5: MSE Performance

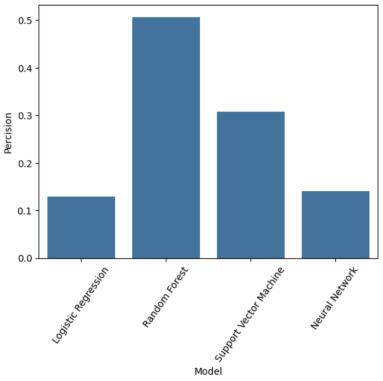


Figure 6: Precision Performance

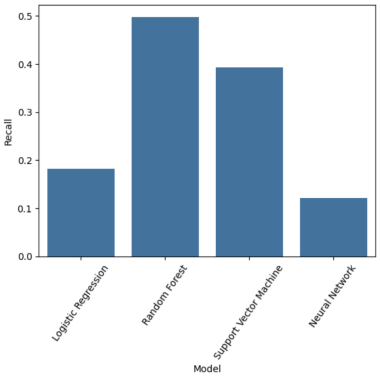


Figure 7: Recall Performance

many matches an individual receives. That is, the model accurately takes into account that female users on this app tended to receive significantly more matches than male users. With these two features being so heavily accented, the model takes into almost no account a person's age, number of children and VIP status on the app.

Because these results were so heavily dependent on gender, it was natural to want to further separate the data set into gendered spits, one with only women and one with only men, to see if their results

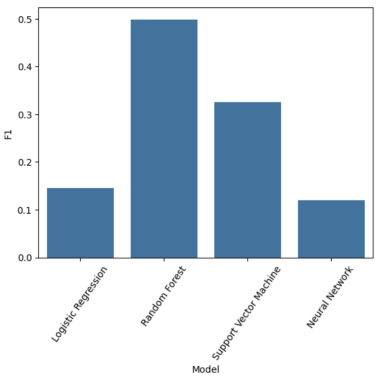


Figure 8: F1 Performance

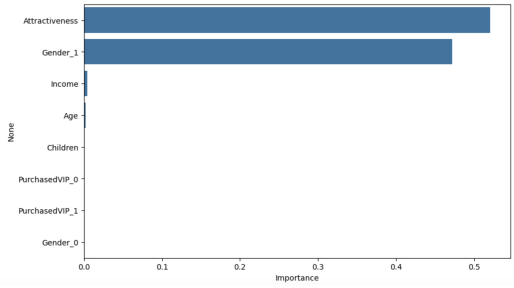


Figure 9: Feature Importance for Random Forest

would reveal any additional patterns. Each split went through the same process as the whole dataset, which produced the following feature importance graphs and tables of evaluation metrics for each gender seen in Figures 10-11 and Tables 2-3.

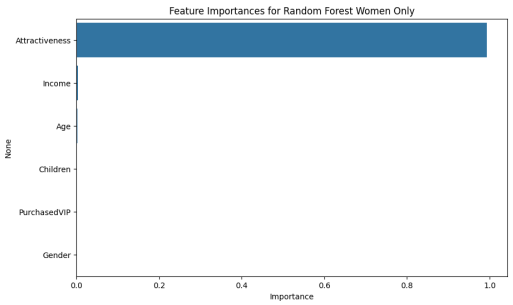


Figure 10: Feature Importance for Women Only

These new feature importance values indicate that male matches depend largely on income, age, attractiveness, and children, in that order from most to least important. For women, it was nearly entirely based on attractiveness, with all other features largely insignificant. Notably, we also see that the gender specific data for both genders was modeled best by the SVM model rather than Random Forest (although Logistic Regression and Random Forest perform so similarly, that it would be hard to decisively crown SVM as the obvious best). For the women-specific portion of the

Table 2: Evaluation Metric Comparison for Women

Model	R^2	MAE	Acc.	MSE	Precis.	Recall	F1
LR	0.96	0.27	0.73	0.28	0.68	0.68	0.67
RF	0.97	0.30	0.73	0.23	0.68	0.68	0.67
SVM	0.98	0.24	0.82	0.16	0.73	0.75	0.74
NN	0.77	0.97	0.33	1.57	0.25	0.24	0.24

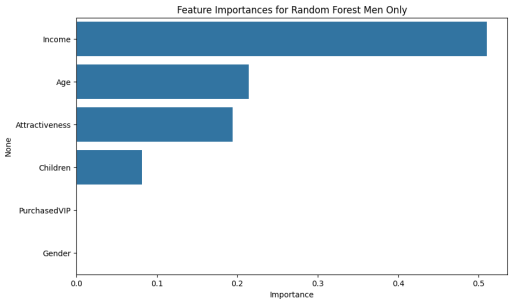


Figure 11: Feature Importance for Men Only

Table 3: Evaluation Metric Comparison for Men

Model	R^2	MAE	Acc.	MSE	Precis.	Recall	F1
LR	-0.16	0.14	0.86	0.14	0.43	0.50	0.46
RF	-0.61	0.26	0.75	0.19	0.52	0.52	0.52
SVM	-0.01	0.21	0.86	0.12	0.43	0.50	0.46
NN	-9.48	0.86	0.41	1.25	0.19	0.13	0.14

data, our models in general perform extremely strongly though for nearly all of the models tested. Logistic Regression, Random Forest, and SVM all scored above a 0.96 R^2, suggesting an extremely linear relationship between attractiveness and matches that's able to be modeled very well (and also suggesting that our dataset might indeed be fairly limited or app-specific). This is also indicated by our higher performance measures across the board compared to the mixed gender model, with Accuracy, Precision, and Recall all scoring 0.68 or above – outside of the inferior Neural Network model. Neural Network models generally excel in handling more complex data, but our dataset is relatively simple, which is likely the reason behind its poorer, but still somewhat sufficient results (0.77 R^2 score). For the men-specific data, we see that none of our models capture this data well, giving all negative R^2 scores across all models. However, accuracy was high for Logistic Regression, Random Forest, and SVM; this is likely due to the models correctly predicting the typical number of matches for many of the men in our data, but failing to predict variations in this number of matches correctly. This would cause the discrepancy we see between poor R^2 and high accuracy. Our results for men could suggest that gender itself was interacting with other features for men in order to influence predictions when the dataset was combined – an interaction that's lost once we split by gender.

The high performance for women and low performance for men for our data can also likely explain why our mixed-gender data gave average performance rather than high performance – our model was only successfully fitting one portion of the users as well as the other.

7 Conclusion

In this project, we set out to predict the number of matches a user may receive on a dating app based on demographic and personal features. This included age, gender, income, number of children, attractiveness (on a scale 1-10), and VIP subscription status on the app. Our goal was to identify which traits were most valued by users in a potential partner, as with this knowledge, the matching algorithms of these apps could likely be enhanced, providing better user experience and higher user satisfaction. After cleaning and augmenting our dataset, we experimented with four models to try and find the most accurate one: Logistic Regression, Random Forest Regression, Support Vector Machine (SVM), and Neural Network. Among these, we found that the Random Forest model emerged as the best performer, achieving the best R^2 score of 0.96 and lowest MSE and MAE scores of 0.39 and 0.52 respectively. Looking at our other models, we can see how they paled in comparison. None of the three other models scored an R^2 score of above 0.44 (Neural Network), an MSE of below 5.76 (Neural Network), or a MAE of below 1.75 (SVM).

When observing feature importance, our Random Forest model revealed attractiveness and gender to be the most influential predictors of the numbers of matches a user will receive. This tells us that a higher perceived attractiveness level and being female relates significantly to an increased likelihood of receiving more matches. Our other features all comparatively received much lower feature importance scores, showing that they have only a very minor impact on number of matches made by the app users.

We can likely attribute the major successes of our Random Forest model compared to the others we tried to a few key strengths. It was very good at handing both numerical and categorical data as we have, which is a general strength of RF and a general weak point for SVM. The ability for Random Forest to capture more complex relationships in data also likely let it greatly prevail over Logistic Regression (the model that seemed to struggle the most overall with the data).

Despite Random Forest being relatively successful, we also need to acknowledge that it isn't a model with extremely high performance. From some of its performance measures (accuracy, precision, and recall), we can see that there's still considerable room for improvement with our model. Also, given the relatively small amount of data we were working with (all of which came from one app), there may be many limitations as far as the scope of the conclusions we can draw from this model. In the future, we could attempt to incorporate more behavioral data or explore a dataset with more overall user features in order to improve our predictive power.

Because of our finding of a strong dependency of results based on gender, we also broadened the initial scope of our discussion to apply our models to each gender of users independently. Doing this, we observed nearly all of our models performing extremely well for the women-only data. Logistic Regression, Random Forest, and

SVM all achieved high R^2 scores and strong accuracy, precision, and recall performance metrics compared to the our mixed-gender data. We also found that for Women, attractiveness was nearly the sole indicator of matches; this combined with our high performance metrics suggested a very strong linear relationship between attractiveness and matches for women. For men, our model struggled much more; we can likely attribute this to more variation in the men's matches that our models are not fitting well, or potentially an interaction between gender itself and the other features.

Both men-only and women-only data had SVM as their highest performing model over Random Forest, although for both, Logistic Regression and Random Forest scored rather close to the same. As such, it's hard to say the true best model among the three for the gender-specific data.

We are quite satisfied and proud with our findings, as we uncovered valuable insights about the traits most valued by dating app users. Using user data in this way can hopefully enhance dating app match algorithms in the future, making users happier and matches more successful in the process.

8 Task Distribution Form

Task	People
Collecting and preprocessing data	Samuel, Ofri, Ethan
Implementing Models	Samuel
Implementing Evaluation Metrics	Samuel, Grant
Evaluating and Comparing Algorithms	Grant, Ethan
Writing Report	Everyone
Preparing slides	Everyone

9 References

Predict Online Dating Matches Dataset, *Kaggle*:
https://www.kaggle.com/datasets/rabieelkharoua/predict-online-dating-matches-dataset

Our Google Drive Folder Containing Our Code Files and Dataset (Note: the 3 links after this are for the files contained in this folder):
https://tinyurl.com/m148drive

Our Google Drive File Containing Our Code For the Main Experiment:
https://tinyurl.com/m148mainexperiment

Our Google Drive File Containing Our Code For the Additional Gender-Specific Experiments:
https://tinyurl.com/m148genderexperiment

Our Google Drive Folder Containing Our Dataset:
https://tinyurl.com/m148dataset