Predicting Relationship Quality from Relationship Attributes in 2022

Jasjot Parmar, Eugene Tse, Jade Chen, and Johnson Leung

Summary

In this project, we used a dataset on based on survey responses to common relationship questions to develop a logistic regression model to classify relationships into one of five relationship quality statuses: excellent, good, fair, poor, or very poor. The model developed below uses common relationship-based features such as whether the subject is married or not and how many children the subject hsa in the relationship.

Introduction

Relationship quality classification is an important topic for couples to be aware of, particularly when trying to maximize the amount of satisfaction each partner recieves from the relationship. Accurate relationship quality classification allows couples in relationships to assess the quality of their relationship to develop better targeted strategies to improve or maintain that relationship quality. It is often difficult for couples to estimate the perceived quality of their relationship, so we investigate below if our machine learning model can correctly classify the quality of a relationshup, based on common relationship attributes. The below analysis tries to answer: How well do relationship characteristics such as age, income category, marital status, relationship duration, and number of children predict relationship quality?

The dataset contains 1293 survey responses of common relationship characteristics such as the income category of the respondent and their estimated relationship quality (our target to predict). This dataset (Diverse Data Hub (n.d.)) is originally based on the How Couples Meet and Stay Together survey (Rosenfeld, Thomas, and Hausen (2023)). We accessed the CSV version of this dataset directly from CRAN (R Core Team (n.d.)). The README.md file contains seperate conda lock files that explain how to run the environment if you want to run the code. We used principles from the Reproducible and Trustworthy Workflows for Data Science textbook (Timbers et al. (n.d.)) on conda lock files to manage enviornments.

In our analysis below, we investigate whether a logistic regression model can correctly classify relationship attributes into one of five relationship quality statuses: excellent, good, fair, poor, or very poor.

EDA

We start by inittially confirming our data was read in to a pandas DataFrame object, as shown in Table 1.

	subject_age	subject_education	subject_sex	subject_ethnicity	subject_income_category	subject_
0	53.0	high_school_grad	female	white	35k_40k	working_
1	72.0	some_college	female	white	75k_85k	working_
2	43.0	associate_degree	male	white	75k_85k	working_
3	64.0	some_college	male	white	75k_85k	working_
4	60.0	high_school_grad	female	black	75k_85k	working_

We can see from the distribution of the Relationship Quality categorical variable, that the dataset contains imbalanced classes, with a very large number of respondents reporting excellent or good relationship quality and a much lower number of respondents reporting fair, poor, and very poor relationship quality.

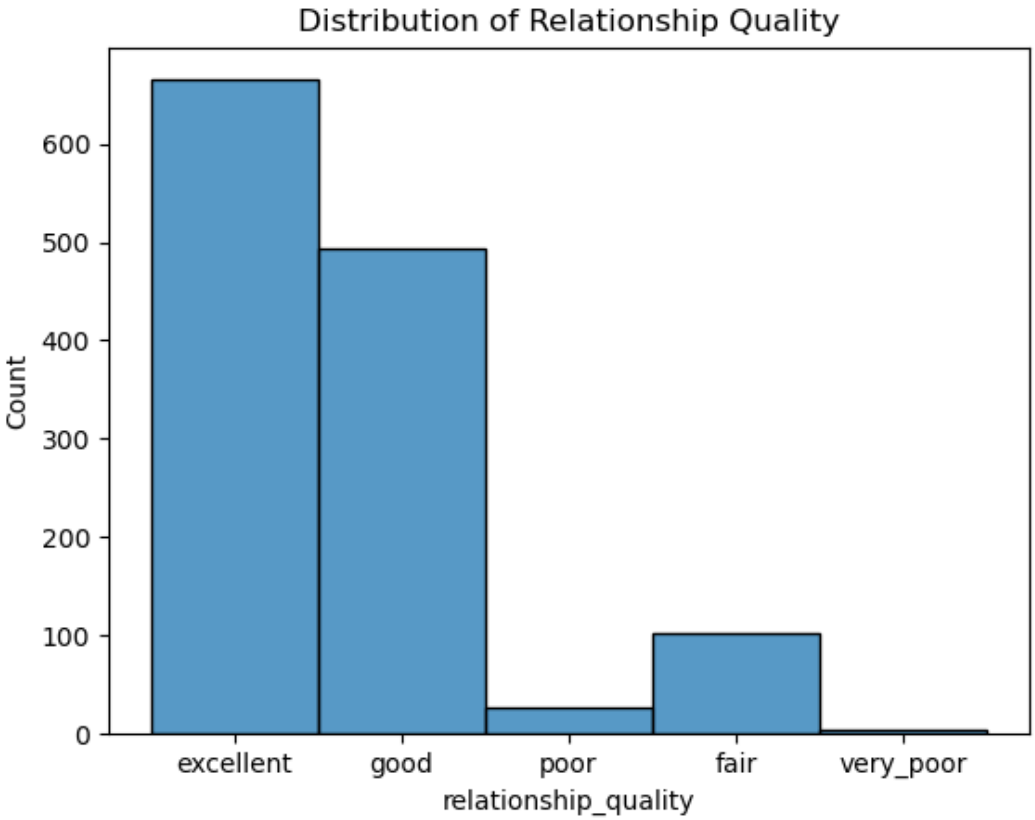


Figure 1: Relationship Quality Score Distributiom

Our numeric predictor / input features show a high correlation between subject age and relationship duration with a ρ value of np.float64(0.736), which means that as the subject's age increases, their relationship duration increases. Subject age and (number of) children show a weak negative correlation of np.float64(-0.326), indicating that as subject age increases, the reported number of children slightly decreases.

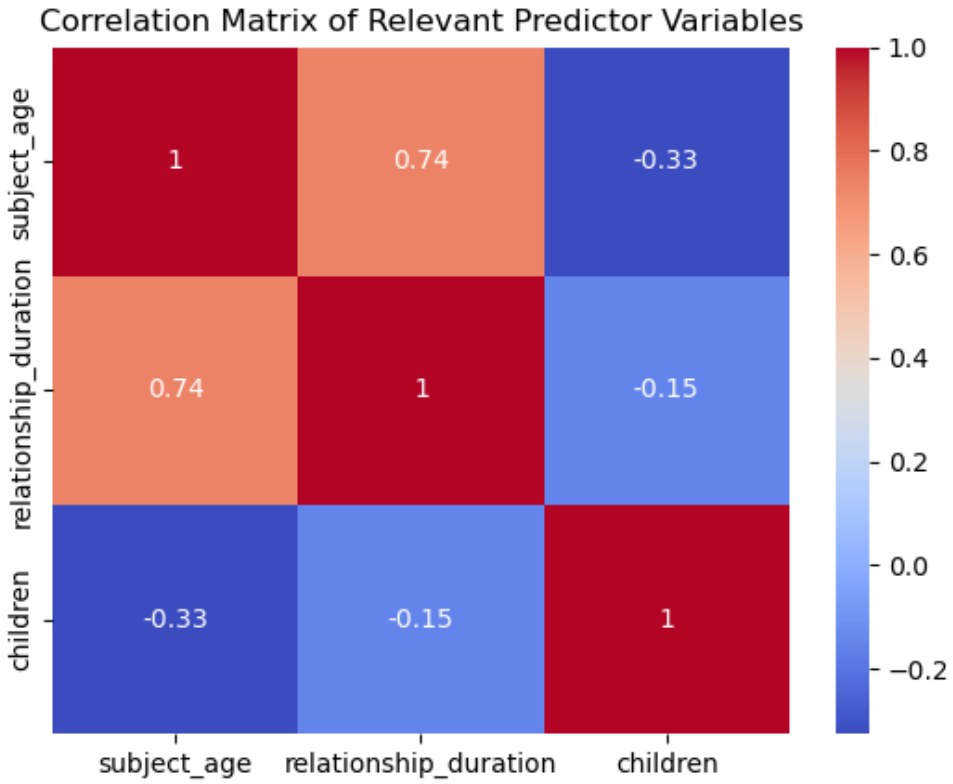


Figure 2: Correlation Heatmap

The distribution of income categories show that we have a left skewed distribution, with most respondents making over 50k per year. For respondents who earn >= 50k / year, income distribution between 50k and 250k+ seems to be roughly uniformly distributed, showing that there is a pretty even spread of incomes between respondents as income passes 50k.

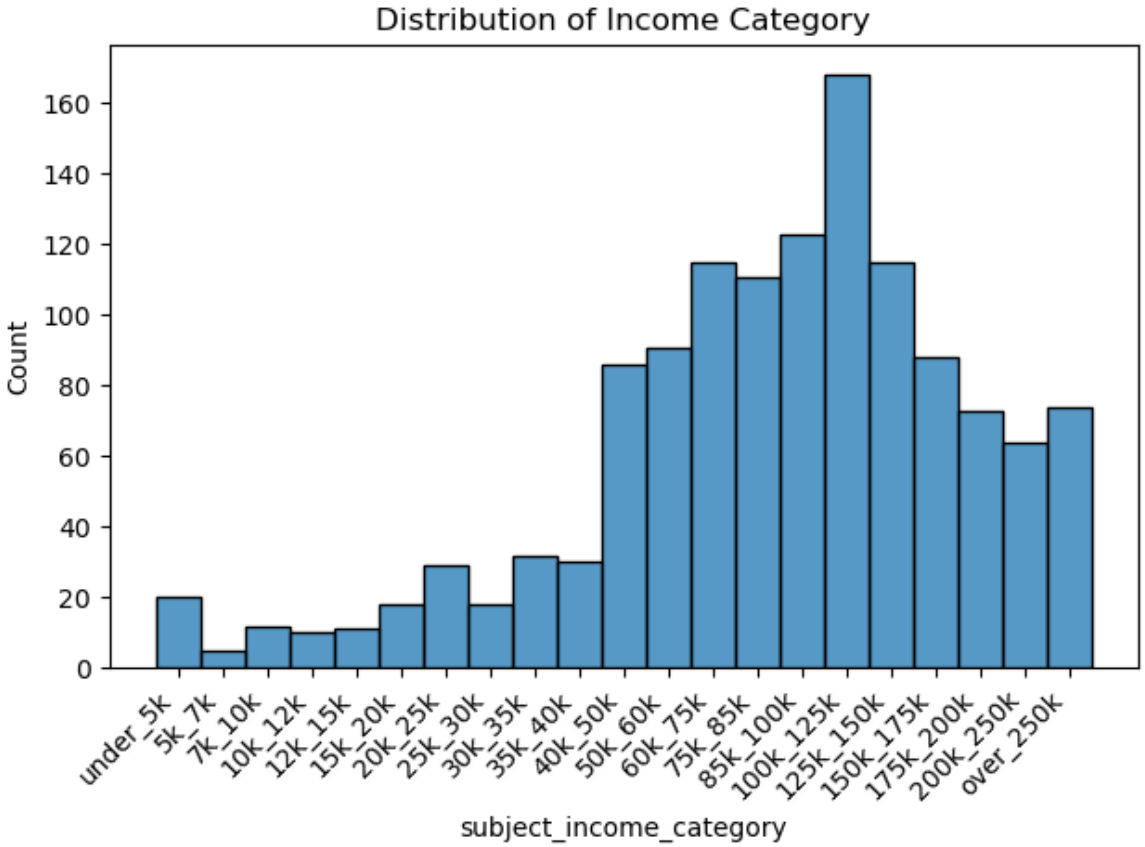


Figure 3: Income Category Distribution

We then split up the data into the relationship features that we want to predict relationship quality with. Input features include: Subject Age, Subject Income Category, Martial Status, Relationship Duration, and Number of Children, before splitting the data into train and test splits. We then conduct simple data cleaning through changing relevant numeric features such as age and number of children into integers, before reordering the income category feature to be ordered in ascending order by income.

Numeric features have different scales, with age having much larger values than relationship duration and number of children. Therefore, Standard Scaler is applied to numeric features so all numeric features contribute equally to the logistic regression model. Ordinal features such as subject income category are converted to ordinal categories, as their categories have an order based on the income of the subject. Categorical features such as marital status are one hot encoded, resulting in one column indicating martial status or not (0 / 1). Each transformation is wrapped in a column transformer.

A scikit-learn pipeline is used to preprocess and train the model on the training data in one step.

The pipeline first applies the preprocessor above to the training set to standardize numeric features and one hot encodes categorical features, before training the Logistic Regression model. The Logistic Regression model addresses our above issue regarding the class imbalance in relationship quality by giving the minority class a bigger penalty, so the model pays more attention to that observation

steps	[('columntransformer', ...), ('logisticregression', ...)]
transform_input	None
memory	None
verbose	False

transformers	[('standardscaler', ...), ('onehotencoder', ...), ...]
remainder	'drop'
sparse_threshold	0.3
n_jobs	None
transformer_weights	None
verbose	False
verbose_feature_names_out	True
force_int_remainder_cols	'deprecated'

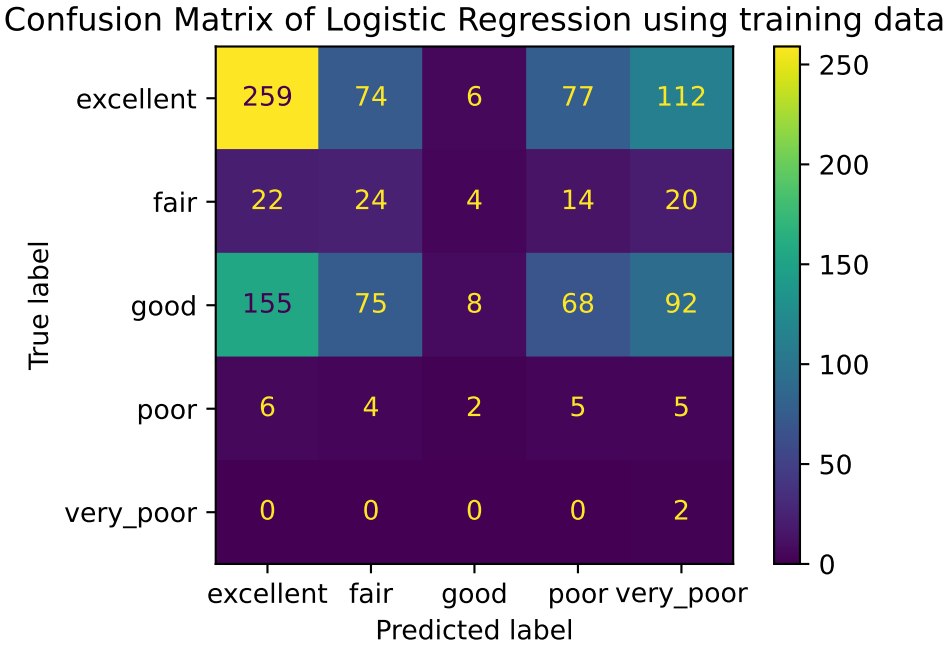
copy	True
with_mean	True
with_std	True

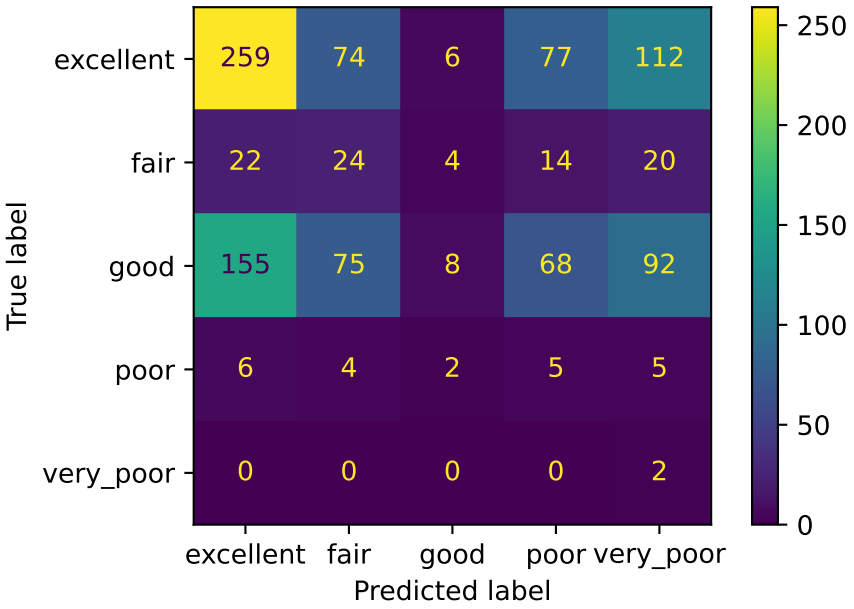
categories	'auto'
drop	'if_binary'
sparse_output	True
dtype	<class 'numpy.float64'>
handle_unknown	'error'
min_frequency	None
max_categories	None
feature_name_combiner	'concat'

categories	'auto'
dtype	<class 'numpy.float64'>
handle_unknown	'error'
unknown_value	None
encoded_missing_value	nan

min_frequency	None
max_categories	None

penalty	'l2'
dual	False
tol	0.0001
C	1.0
fit_intercept	True
intercept_scaling	1
class_weight	'balanced'
random_state	None
solver	'lbfgs'
max_iter	1000
multi_class	'deprecated'
verbose	0
warm_start	False
n_jobs	None
l1_ratio	None





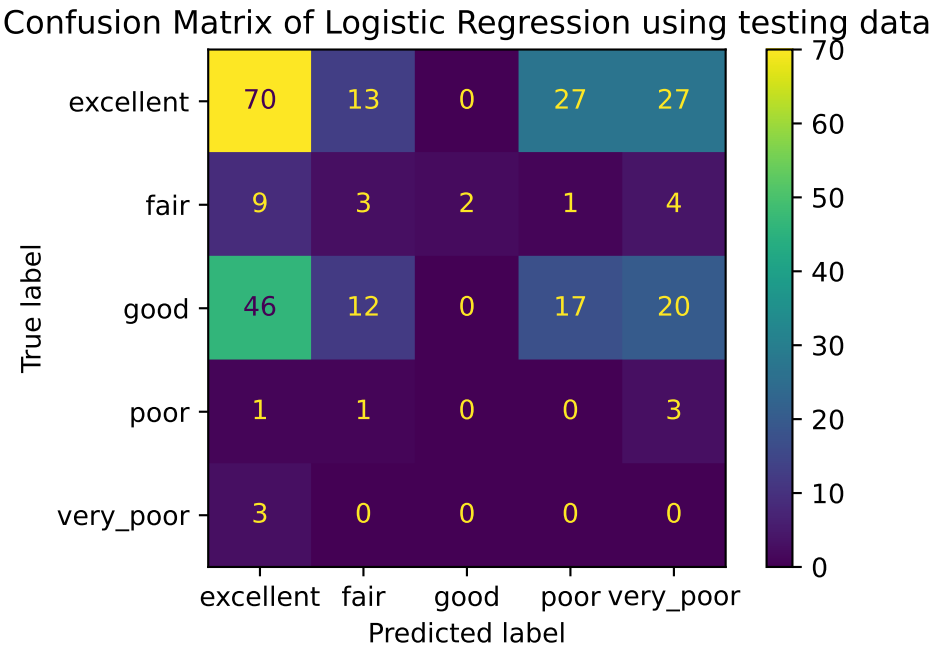
To see how our model did on predicting relationship quality based on the training data, we plot a confusion matrix. We see that for respondents with excellent relationship quality, the model only correctly predicts np.float64(49.1)% of relationships that have excellent relationship quality correctly (Recall = np.float64(0.49053030303030304)). Of all relationships that are predicted to have excellent relationship quality,the model correctly predicts np.float64(58.6)% of them (Precision = np.float64(0.586)).

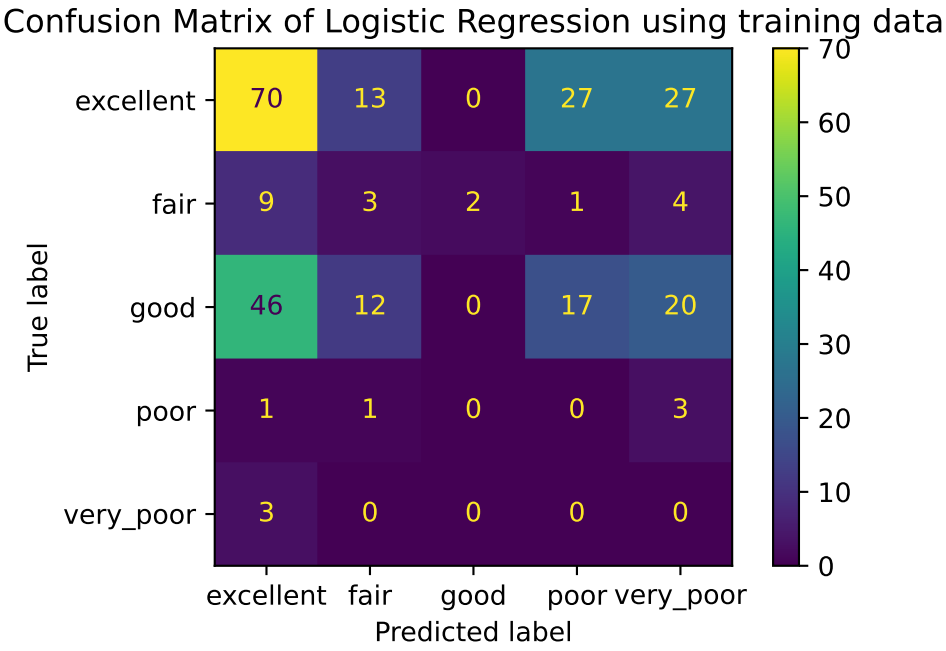
For respondents with Good relationship quality, the model only correctly predicts np.float64(2.0)% of them. For all relationships that are predicted to have good relationship quality, the model correctly predicts np.float64(40.0)% of them.

For respondents with Fair relationship quality, the model only correctly predicts np.float64(28.6)% of them. Of all relationships that are predicted to have Fair relationship quality, the model correctly predicts np.float64(13.6)% of them.

For respondents with Poor relationship quality, the model correctly predicts np.float64(22.7)% of them. For all relationships that are predicted to have Poor relationship quality, the model only correctly predicts np.float64(3.0)% of them.

For respondents with Very Poor relationship quality, the model correctly predicts np.float64(100.0)% of them (there are very few observations with very poor relationship quality so this prediction should be used carefully). Out of all relationships that are predicted to have very poor relationship quality, the model correctly predicts only np.float64(0.9)% of them.





To see how our model did on predicting relationship quality based on the testing set, we plot a confusion matrix for predictions on the test set. We see that for respondents with excellent relationship quality, the model only correctly predicts np.float64(51.1)% of relationships that have excellent relationship quality correctly (Recall = np.float64(0.511)). Of all rela-

tionships that are predicted to have excellent relationship quality,the model correctly predicts np.float64(54.3)% of them (Precision = np.float64(0.543)).

For respondents with Good relationship quality, the model only correctly predicts np.float64(0.0)% of them. For all relationships that are predicted to have good relationship quality, the model correctly predicts np.float64(0.0)% of them.

For respondents with Fair relationship quality, the model only correctly predicts np.float64(15.8)% of them. Of all relationships that are predicted to have Fair relationship quality, the model correctly predicts np.float64(10.3)% of them.

For respondents with Poor relationship quality, the model correctly predicts np.float64(0.0)% of them, as there is only 1 poor relationship quality observation in the test set. For all relationships that are predicted to have Poor relationship quality, the model only correctly predicts np.float64(0.0)% of them, since the model predicted an observation to have Poor relationship quality 0 times.

For respondents with Very Poor relationship quality, the model correctly predicts np.float64(0.0)% of them. Out of all relationships that are predicted to have very poor relationship quality, the model only correctly predicts np.float64(0.0)% of them.

```
Micro-averaged One-vs-Rest ROC AUC score:
0.62
```

Our micro-averaged ROC curve above shows a AUC of 0.621. This means that our model is not the strongest at correctly predicting relationship quality based on the input relationship features we specified above. The ROC curve shows that our model is only slightly better than randomly guessing the relationship quality class, meaning that our model's accuracy is pretty weak.

Discussion:

Our findings above indicate a poor overall accuracies across each class on the testing set, with an overall test accuracy from the confusion matrix of np.float64(28.2)% meaning that the model is poor at predicting the correct relationship quality based on features such as age, income_category, marital status, relationship duration, and number of children. We also see that the precision and recall of each relationship quality class is quite low in the training and testing set. Our poor accuracy, precision, and recall for all relationship quality classes tells us that with a Logistic Regression model, the features we included (age, income category, marital status, relationship duration, and number of children) do not predict relationship quality well.

This is generally what we expected to find because the features we chose are mostly external or demographic traits about the relationship that one can argue, do not define the emotional or personal status of a relationship. Since our features do not include deeper characteristics that could matter more for relationship quality compared to demographic features like age, it makes sense our Logistic Regression model is performing poorly.

These findings could change how people in relationships and researchers think about what defines relationship quality. Since our above results show that demographic or external features like age, number of children, income category, and relationship duration were not good predictors of relationship quality with our Logistic Regression model, people could place less focus on these relationship features when gauging the relationship quality of their own relationship. The results could lead to people in relationships placing more importance on emotional or behavioural metrics in relationships instead like how often partners openly communicate about problems. These emotional and behavioural relationship features could be much better predictors of relationship quality. These results could ultimately impact how relationship quality is assessed, by changing the focus to deeper personal relationship dynamics instead of surface level demographic features.

The future questions our above results could lead to are:

- What emotional or personal relationship features (that relate to both partners) best predict relationship quality?
- Could using a dataset that contains data for relationship matrics from both partners in the relationship improve accuracy of our above Logisitc Regression model?
- How much better (or worse) would non-linear models such as decision trees perform on the same above dataset?
- Which relationship features that we used above contribute most to predicting relationship quality?

References

Diverse Data Hub. n.d. "How Couples Meet and Stay Together." https://diverse-data-hub.github.io/website_files/description_pages/hcmst.html.

R Core Team. n.d. "Hcmst.csv [Data Set]." CRAN. https://cran.r-project.org/incoming/UL/diversedata/data-clean/hcmst.csv.

Rosenfeld, Michael J., Reuben J. Thomas, and Sonia Hausen. 2023. "How Couples Meet and Stay Together 2017–2020–2022 Combined Dataset [Data Set]." Stanford University Libraries. https://data.stanford.edu/hcmst2017.

Timbers, Tiffany A., Joel Ostblom, Florencia D'Andrea, Rodolfo Lourenzutti, and Daniel Chen. n.d. "Conda Lock: Reproducible Lock Files for Conda Environments." In *Reproducible and Trustworthy Workflows for Data Science*. UBC Master of Data Science. https://ubc-dsci.github.io/reproducible-and-trustworthy-workflows-for-data-science/lectures/090-conda-lock.html.