1. Describe your Model

What approach did you take?

I download the data from https://osf.io/wx7ck/, which is recommended by Professor Cowan on “final project discussion board”. For training and interpolate this dataset, I took linear regression, knn, and natural language processing method.

What design choices did you make, and why?

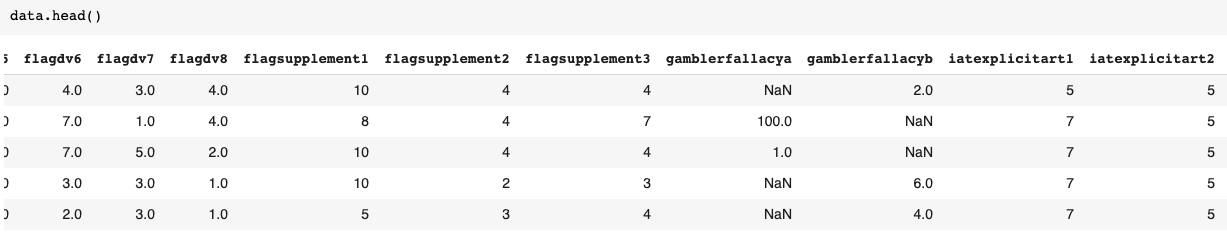
I planned to convert all the data in my dataset into numerical types, and use some different methods to build the model, and get the predicted value in the missing parts.

The design choices I made are mainly linear regression. The linear regression is a linear approach to modelling the relationship between a scaler response and one or more explanatory variables (dependent and independent variables) The overall model is simply: y = wx +b.

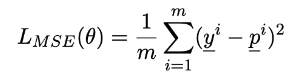
And what I need to do is to minimize the MSE. KNN is a simple algorithm that is often used for classification. It can classify the data point by analyzing how its neighbor is classified.

How did you represent the data? Did you make an effort to identify and exclude irrelevant variables?

I dropped empty columns which are columns that only have a title but have no data. I dropped date columns which are columns that have a lot of repetitive date and time. I dropped useless url columns which are columns that contains only some irrelevant links. I also dropped 'session\_created\_by', 'study\_url','study\_name','text','session\_id', 'user\_id', 'previous\_session\_id', 'expcomments' because after my review of all the string columns, these columns have nothing to do with the questionnaire. They are irrelevant variables.

Then since there are many types of data here, I want to convert all the types to a number type: float. For existing float and int data, it is already done. For string data, I used natural language processing method to convert them to float type. I used a function which is like some kinds of encoder to use int or float to represent different string data. For columns that describes the extent of some choices, I use numbers to represent the extent. For example, in “iatexplicitmath1” column with strings, I use '1' if x == 'Very bad', '2' if x == 'Moderately bad', ‘3' if x == 'Slightly bad'. For columns that describes different stuff, I use increasing numbers to represent each one. For example, in “exprace” column, I use '11' if x == 'brazilwhite', '12' if x == 'brazilblack', ‘13' if x == 'brazilbrown', '14' if x == 'chinese', '15' if x == 'malay', '16' if x == 'dutch’. The resulting dataset is like this: 

How can you evaluate your model for goodness of fit?

To evaluate the model, I need to first fit the training data to the model and get the weights and bias. Then I will use the w and b I get to predict y, and compare the difference between real y and my predicted y. I used MSE and RMSE  

How did you handle missing data?

For missing data, I think my task is to fill in these missing data by predicting using my model.

2. Describe your Training Algorithm

Given your model, how did you fit it to the data?

I planned to first fit in the columns with less null elements as it is easier to predict. Then there should be more data that I could use to do the prediction. The efficiency and accuracy would increase if more features are known when predicting our target feature.

Were you able to train over all features?

I am able to train over all numerical features and most string features that can be converted to numerical features by using NPL method. For unrelated data, I just ignore them. I also give up some string features that cannot be represented by numbers. I think training over the existed numerical features is easier and more accurate.

What kinds of computational tradeoffs did you face, and how did you settle them?

Mostly the computational tradeoffs I faced was the parameter in a function. For example, I was struggling on choosing lambda parameter in my ridge and lasso regression functions. If I set the alpha too big, the weights are too small to observe. If the alpha is too small, the linear regression is not solvable. I settled them by trying to implement different lambda values and compare the error of each value like we did in the last homework. I also combine the test results with my results from the last homework and get the proper lambda value for my linear regression functions.

3. Describe your Model Validation

How did you try to avoid overfitting the data?

I used ridge regression to avoid overfitting the data. By using ridge regression, we introduced penalty which can decrease the variance, which affect the final function I want to fit. When I tried to avoid overfitting, actually I wanted to get rid of the noise, and get a “smooth” data. “Smooth” here needs the process of fitting to reduce those points where there is a large difference between the fitting data and the original trend of the data. I tried to get rid of these data, which somehow avoid overfitting data.

How did you handle the modest (in ML terms) size of the data set?

I have about 6000-7000 rows of data, although a few of them contain only a little useful information, I will still use them as long as I can. Less is better than nothing.

4. Evaluate your Model

Where is your model particularly successful, where does it lack?

My model successes in generating the data to the missing places. Although the accuracy is not high, but it is apparently higher than a filling in a possible random number in the place of a missing data. For example, …….

It lacks more complicated algorithms to fill up the model with. So, the accuracy of the model is not very high. I think another problem is that I did not do a normalization of the data, which means that the range of data in a column is sometimes too big and sometimes small. For example, in column “anchoring2b”, the range of the data can be 1 billion. However, in columns that are multiple choice questions with numbers 1-6, the range is at most 5. I think the if the range of a column is not normalized, the stability and accuracy of the weight cannot be guaranteed.

Does it need a certain number of features in order to interpolate well?

I think I need more data that are not null in order to interpolate well. Of cause, increase the number of features can also interpolate better. But I can only deal with limited string features and numerical features. So maybe some features in the dataset are wasted.

Are there some features it is really good at predicting and some it is really poor at predicting?

I decide whether some features are good at predicting by looking at the weight they have on predicting one other feature in the dataset.

The orange line here means the weight of each feature. I used the column “quotea” as the predicted column because this column represents the test takers’ attitude toward a certain statement. Since other columns have collected and analyzed the characteristic of the test takers, the “quotea” column is considered a good feature for predicting. During the prediction process, we can see that the weight of the 35-38th columns is relatively large, there are also some features that have a relatively large weight as we can see in the plot. These features are good at predicting. In the graph, we can also see many features have a weight of 0 or near 0. These features are poor at predicting. There are more analyses of data in the next question.

5. Analyze the Data

What features were particularly valuable in predicting/interpolating?

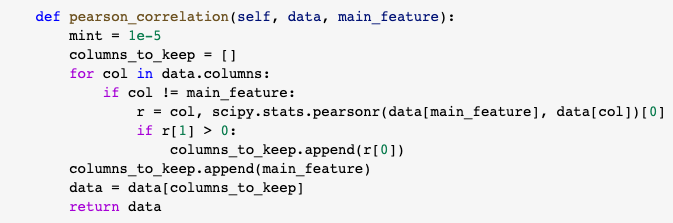
The features with a larger weight as I analyzed above were particularly valuable in predicting and interpolating. For example, these kinds of features in my dataset includes “moneyage” which represent the age of test takers, “gamblerfallacya” and “gamblerfallacyb”. These two features contain the attitude of test takers toward the problem. After I check the codebook, I know that these two tests were asking “Imagine that you are in a casino and you happen to pass a man rolling dice. You observe him roll three dice and all three come up 6's. Based on your imagined scenario, how many times do you think the man had rolled the dice before you walked by?” and “Imagine that you are in a casino and you happen to pass a man rolling dice. You observe him roll three dice and one come up 3, and two come up 6's. Based on your imagined scenario, how many times do you think the man had rolled the dice before you” I think the answer of these kinds of questions can represent the thoughts and characters of the people. This is relevant with our y column, which was also testing the test taker’s thoughts and character. Since the main purpose of the whole questionnaire was to investigate the thoughts and characters of people, I think these features should also be valuable in predicting and interpolating any other columns in the dataset.

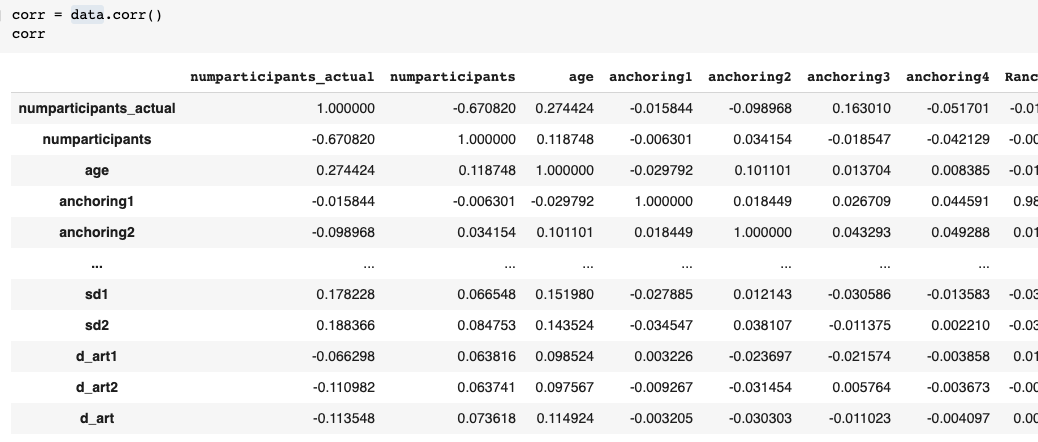
What features weren’t particularly useful at all?

In my analysis, the features with a small or 0 weight on predicting the target feature were not particularly useful at all. For example, I found the “omdimc3rt” and “omdimctrt” features to be not very useful since they got a weight of 0. I don’t know what do they mean in English and in the questionnaire, but I found that the values under these features were over 10 thousand, which were much larger than most of values under other features, which were mostly 1-10. I think maybe it is not because that these two features are not important at all. It is because of the large value of these features, a small change on the weight could cause a relatively very large impact on the predicting feature. Anyway, in my model, features like these were not very useful.

Were there any features about the researcher/experimental environment that were particularly relevant to predicting answers - and if so, what conclusions can you draw about the replicability of those effects?

In order to see the replicability of features, we need to consider the correlation between each pair of features. The correlation of the features with a similar name were more likely to be larger than features with no relations on names. For example, in the table of a small part of correlation between features, we can see that the correlation between “numparticipants\_actual” and “numparticipants” are big. They are indeed similar in names and maybe have similar function in the questionnaire. I think we could delete some features that have a large correlation with other features since they have almost the same effect on predicting. This may increase our speed when running the model. I think as long as the correlation is not near 1, the features can remain especially when we don’t have so many features to use.





6. Generate Data

Use your system to try to generate realistic data, and compare your generated data to the real data. How good does it look?

The system can generate realistic data. I just generate data inside the original dataset. I choose the columns that were full originally, and then I delete features to make it look like a missing data. After training and predicting using the system, the missing row is filled with the predicted data.

What does it mean for it to ‘look good’?

I think the first step to ensure it to “looking good” is to see if the predicted data “look like” the right one. For example, if the data of a feature is between integer 1-6. A “Looking good” data must between 1-6 too. If we need much data to look good, then we need to analyze the accuracy of the data. I think a good way to determine if some data looks good is to calculate the ratio of the correct data to the number of data I implemented. This means that for predicted data, there should be as much cells as possible that are the same as the real data.