Introduction

Here, we investigate the dataset *New York City Longitudinal Study of Well-Being* where 1182 variables were collected from 3908 different people between the years of 2015 and 2018. The variables contain financial factors and subjective factors, most collected by survey and many collected by imputation of other variables. We sifted through some of these variables' descriptions and singled out 11 variables we found to be of interest. Financial factors include medical debt (yes/no) (IMP_Q3MEDICALDEBT_D); education-related debt (yes/no) (IMP_Q3EDUCDEBT_D); spending on recreation, entertainment, or fitness (numeric) (Q5C20X_TC); monthly rent (numeric) (QA9X_TC). Non-financial factors include minutes to get from home to work (numeric) (Q6B27X_TC); age (numeric) (IMP_AGE_TC); subjective feeling of parks in their neighborhood (1-4) (Q2C2_11); access to internet on cellphone, tablet, or mobile device (yes/no) (QI4); quality of food eaten in household (1-4) (IMP_PFOOD1); and overall self-reported health (1-5) (IMP_HEALTH).

We are ultimately interested in the respondent's **subjective feeling of life in the past 12 months** (QD6). We sought out a diversity of variables to seek the most important factors contributing to a person's "happiness" or "well-being" as evaluated by themselves. US culture tends to emphasize *financial* well-being in the analysis of *overall* well-being, which is one reason we included these financial factors; we also sought non-financial factors, like age, quality of food, overall health, and minutes traveling to and from work in order to get a more holistic view of one's subjective feeling of life.

There are multiple things to note about these variables, specifically regarding their values in the dataset. The values are frequently associated with categorical responses because the survey required it. For example, for self-reported feelings of overall health, the values ranged from the integers 1-5 corresponding to the survey responses 1. Excellent, 2. Very Good, 3. Good, 4. Fair, and 5. Poor. This makes the ultimate interpretation of these variables

precarious, because whatever distinguishes a respondent's "Good" and "Very Good" (or some equivalent difference in response) can be very subjective and might not be best represented by integers with a spacing of 1. The actual variables' response values are in Appendix A. The code for our analysis is in Appendix B.

Methods

Before elaborating on the specific models and methods used in the statistical analysis of these data, we handled missingness by only taking the observations where the value of all variables mentioned above are present. This resulted in 657 observations, which is 16.8% of the original number of observations. We did not do an extensive analysis regarding *which* observations were eliminated; so there could be systematic bias of the dataset entering here, for instance if the respondents with missing values were more or less likely to rate themselves as more satisfied with life.

We created a classification variable from the 1-10 response by dividing the classes into (QD6 <= 5)→0 and (QD6 > 5)→1. Approximately this corresponds to a good/bad assessment of one's life, but at a very high level. We called this variable QD6_bin. We performed regression on the raw subjective feeling of life (QD6) and classification for a binarized subjective feeling of life (QD6_bin). To do this, we used least-squares linear regression for QD6, logistic regression for QD6_bin, and random forests for both regression and classification. The random forests we built used all 10 variables, and we tried a value of m=3 and m=4 (selected according to the heuristic a good m~$\sqrt{10}$). For all models we initially used all variables, but we tried smaller, more interpretable models using the "most important" variables as judged by low p-values and importance metrics from random forests.

To validate our models, we used a 70/30 train/test split, arrived at after trying multiple splits (50/50, 80/20, 90/10), which gave us a training set of 459 observations and a validation set of 198 observations. For the assessment of each model we performed 10-fold

cross-validation to get the MSE value for regression and the accuracy for binary classification, with an exception being the boosting methods, for which we used a 5-fold method.

Results

In almost all models there were recurring variables which displayed high predictive power and variables which were not useful at all and regularly hurt prediction accuracy. The variables which displayed a low p-value in our linear regressions were the quality of food eaten in household (IMP_PFOOD1), overall self-reported health (IMP_HEALTH), and yes/no medical debt (IMP_Q3MEDICALDEBT_D). The p-values for each of the variables in our full model is presented in Table 1. Further, when we built a random forest model and ran variable importance, we found the same variables rising to the top importance, which you can see in Figure 1.

Variable	**Parameter estimate**	**p-value**
IMP_Q3MEDICALDEBT_D	-3.908e-01	0.032087 *
IMP_Q3EDUCDEBT_D	1.040e-01	0.530373
Q5C20X_TC	2.677e-04	0.373494
QA9X_TC	2.508e-05	0.845859
Q6B27X_TC	-2.967e-03	0.394073
IMP_AGE_TC	4.122e-03	0.454543
Q2C2_11	-1.122e-01	0.229636
QI4	-3.760e-01	0.176703
IMP_PFOOD1	-7.342e-01	-6.305 6.93e-10 ***
IMP_HEALTH	-2.816e-01	0.000595 ***

Table 1: Full linear regression model summary

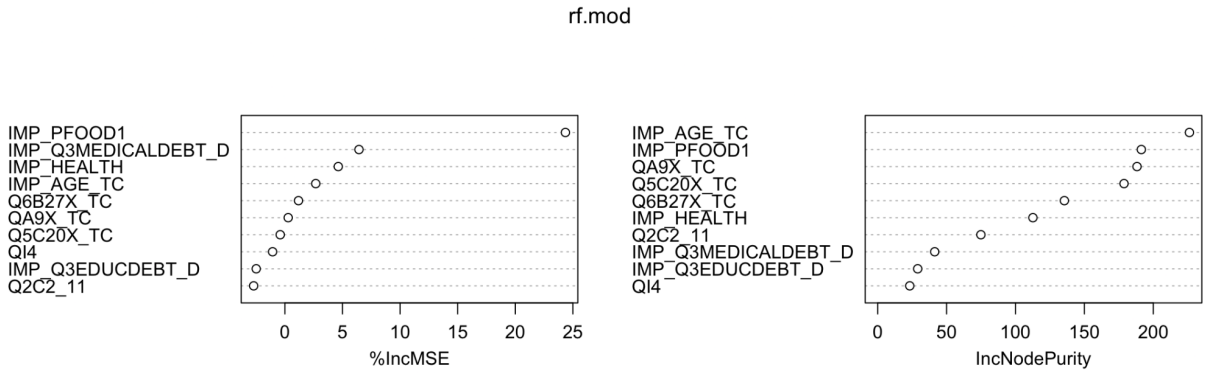

Figure 1: Variable importance from full random forest model (m=4); left: % increase in MSE for each variable; right: increase in node purity for each variable

When comparing these models, we needed to be skeptical of the MSE because of our small range (the values 1 through 10) and the distribution of observations across these values.

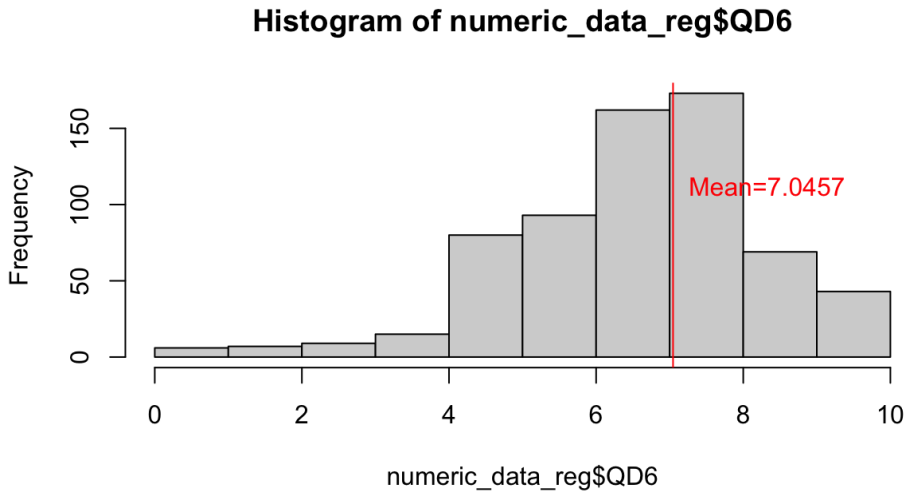

Figure 2: Distribution of observations over each value of the response variable. numeric_data_reg is the full, unsplit dataset

The distribution of response values is skewed and it leaves our binarized classes unbalanced. We want to compare our predictive models to naive strategies in order to get a better idea of our models' performance. In the case of naive regression, if we predict using the mean of the response variable QD6 from the training data, we get an MSE of 3.076. And if we

naively predict 1 for every classification, we get a classification rate of 0.8232. All models should be compared directly with these naive strategies. In the table below, a 70-30 split testing estimate is shown first, with a 10-fold cross-validated estimate being shown in parentheses.

	Regression (MSE)	Classification (Accuracy)
Naive	3.076	0.8232
Linear Regression (all variables)	2.769 (2.621)	N/A
Linear Regression (3 variables)	2.821 (2.612)	N/A
Logistic Regression (all variables)	N/A	0.8426 (0.825)
Random Forest (all variables) (m=3)	2.928	0.8173
Random Forest (all variables) (m=4)	2.960	0.8071
Random Forest (boosted, all variables) (λ=0.01, CV folds = 5)	(3.19)	(0.7715)

Table 2: Mean Squared Error/Accuracy for various models tested

Discussion

Our goal was to seek the impact of various factors on how a person rates their feeling of life in the past 12 months. In this pursuit we attempted to model a person's subjective feeling of life as a function of other variables given by the person or imputed from the person's survey responses. If our goal was to use these variables to *predict* a person's subjective feeling of life, we do not do much better than simply predicting 1 for all classifications or predicting the mean of our training data for regression. The primary insight of this analysis was the discovery of the "important" variables that lead to a person's subjective feeling of life.

It turns out that, if we look at the financial variables medical debt, education-related debt, spending on recreation, entertainment, or fitness, monthly rent, and non-financial variables minutes to get from home to work, age, subjective feeling of parks in their neighborhood, access

to internet on cellphone, tablet, or mobile device, quality of food eaten in household, and overall self-reported health, then the most important variables are *quality of food eaten in the household* and one's *overall (self-reported) health*. These variables seem intuitively associated with one's overall "feeling of life"; it reinforces the idea that taking care of your body and overall health is a significant factor in how you feel about your life overall. Secondarily but of repeat importance we found a higher *monthly rent* to be associated with a higher overall feeling of life. Many hypotheses might be generated from these data; for example one's home environment being slightly more expensive might be associated with a *safer* environment, leading to a higher feeling of well-being.

Appendix A: Variables

IMP_Q3MEDICALDEBT_D – Medical debt

- Values: Yes/No

IMP_Q3EDUCDEBT_D – Education-related debt

- Values: Yes/No

Q5C20X_TC – Spending on recreation, entertainment, or fitness

- Values: Numeric

QA9X_TC – Monthly rent

- Values: Numeric

Q6B27X_TC – Minutes to get from home to work

- Values: Numeric

IMP_AGE_TC – Age

- Values: Numeric

Q2C2_11 – Subjective feeling of parks in their neighborhood

- Values: 1. Excellent, 2. Good, 3. Fair, 4. Poor

QI4 – Access to the internet on cellphone, tablet, or mobile device?

- Values: Yes/No

IMP_PFOOD1 – Best describes food eaten in household

- Values: 1. Enough of the kinds of foods we want to eat, 2. Enough, but not always the kinds of foods we want, 3. Sometimes not enough to eat, 4. Often not enough to eat

IMP_HEALTH – Overall Health self-reported

- Values: 1. Excellent, 2. Very Good, 3. Good, 4. Fair, 5. Poor

QD6 – Subjective feeling of life in past 12 months

- Values: 1-10 (1 being worst, 10 being best)

Appendix B: Code

```
load("C:/Users/polit/OneDrive/Desktop/CS
288/DS0001/38062-0001-Data.rda")
library(randomForest)
library(gbm)
library(boot)
library(caret)
RNGversion('3.5.3')

# FINANCIAL #
#(financial, categorical)
da38062.0001$IMP_Q3MEDICALDEBT_D # Yes/No medical debt
da38062.0001$IMP_Q3EDUCDEBT_D # Education-related debt

# (financial, quantitative)
da38062.0001$Q5C20X_TC # Spending on recreation, entertainment, or
fitness
da38062.0001$QA9X_TC # Monthly rent

# NON-FINANCIAL #
#(non-financial, quantitative)
da38062.0001$Q6B27X_TC # Minutes to get from home to work
da38062.0001$IMP_AGE_TC # Age

# (non-financial, categorical)
```

```
da38062.0001$Q2C2_11 # Subjective feeling of parks in their
neighborhood
da38062.0001$QI4 # Access to internet on cellphone, tablet, or mobile
device?
da38062.0001$IMP_PFOOD1 # Best describes food eaten in household
da38062.0001$IMP_HEALTH # Overall Health self-reported

###### Complete Cases ########
# Extracts the observations in the dataset for which all fin/non-fin
variables are present
complete_cases = function(dataset, response) {
  financial = c('IMP_Q3MEDICALDEBT_D', 'IMP_Q3EDUCDEBT_D',
'Q5C20X_TC', 'QA9X_TC')
  non_financial = c('Q6B27X_TC', 'IMP_AGE_TC', 'Q2C2_11', 'QI4',
'IMP_PFOOD1', 'IMP_HEALTH')

  complete = complete.cases(dataset[c(financial, non_financial,
response)])
  return(dataset[complete, c(financial, non_financial, response)])
}
subsetted_data_reg = complete_cases(da38062.0001, response=c('QD6'))
subsetted_data_log = complete_cases(da38062.0001, response=c('QD6'))
subsetted_data_log$QD6_bin = as.integer(subsetted_data_reg$QD6 > 5)
subsetted_data_log = subsetted_data_log[, names(subsetted_data_log) !=
'QD6']

subsetted_data_log
subsetted_data_reg

###### Convert To Classes ########
binary_vars = c('IMP_Q3MEDICALDEBT_D', 'IMP_Q3EDUCDEBT_D', 'QI4')
classes_to_int = function(dataset, binary_vars) {
  for(var in names(dataset)) {
    dataset[var][,1] = as.numeric(dataset[var][,1])
    # Convert Yes/No values to 0/1 values
    if(var %in% binary_vars) {
      dataset[var] = rapply(dataset[var], function(x) ifelse(x==2, 0,
x), how="replace")
    }
  }
  return(dataset)
}

numeric_data = classes_to_int(subsetted_data_reg, binary_vars)
```

```
numeric_data_log = classes_to_int(subsetted_data_log, binary_vars)
numeric_data
numeric_data_log

#REGRESSION -----
set.seed(1)
#data split
train = sample(1:nrow(numeric_data), 460)
test = numeric_data[-train, "QD6"]

#linear regression model --- all variables
mod = lm(QD6 ~ ., data = numeric_data, subset = train)

yhat = predict(mod, newdata = numeric_data[-train,])
testMSE = mean((yhat - test)^2)
testMSE
yhat2 = predict(mod, newdata = numeric_data[train,])
trainMSE = mean((yhat2 - numeric_data[train,"QD6"])^2)
trainMSE

#bootstrapping
boot.fn = function(data, index){
  return(coef(lm(QD6 ~ ., data=data , subset=index)))
}
boot(numeric_data, boot.fn, 1000)
summary(mod)

#10-fold CV
set.seed(1)
glm.fit = glm(QD6 ~ ., data = numeric_data)
cv.error = cv.glm(numeric_data, glm.fit, K=10)
cv.error$delta

#linear regression model --- financial variables
mod = lm(QD6 ~ IMP_Q3MEDICALDEBT_D + IMP_Q3EDUCDEBT_D + Q5C20X_TC +
QA9X_TC,
         data = numeric_data,
         subset = train)
yhat = predict(mod, newdata = numeric_data[-train,])
testMSE = mean((yhat - test)^2)
testMSE
yhat2 = predict(mod, newdata = numeric_data[train,])
trainMSE = mean((yhat2 - numeric_data[train,"QD6"])^2)
trainMSE
```

```
#linear regression --- reduced model w/ k = 10 cv
set.seed(1)
glm.fit = glm(QD6 ~ IMP_PFOOD1 + IMP_HEALTH + IMP_Q3MEDICALDEBT_D,
data = numeric_data)
cv.error = cv.glm(numeric_data, glm.fit, K=10)
cv.error$delta

#random forest --- regression
rf.mod = randomForest(QD6 ~ ., data = numeric_data, subset = train,
                      mtry = 4, ntrees = 500, importance = TRUE)
yhat = predict(rf.mod, newdata = numeric_data[-train,])
testMSE = mean((yhat - test)^2)
testMSE
yhat2 = predict(rf.mod, newdata = numeric_data[train,])
trainMSE = mean((yhat2 - numeric_data[train,"QD6"])^2)
trainMSE

#boosting --- regression
set.seed(1)
boost.mod = gbm(QD6 ~ ., data = numeric_data[train,],
                    n.trees=5000, interaction.depth=3,
                    shrinkage = 0.01, cv.folds = 5)
yhat.boost = predict(boost.mod, newdata = numeric_data[-train ,],
                    n.trees=5000)

testMSE = mean((yhat.boost - test)^2)
testMSE
yhat2.boost = predict(boost.mod, newdata = numeric_data[train,],
                      n.trees=5000)
trainMSE = mean((yhat2.boost - numeric_data[train,"QD6"]))
trainMSE

#CLASSIFICATION -----
#logistic regression ---
set.seed(1)
test = numeric_data_log[-train, "QD6_bin"]
log.reg = glm(QD6_bin ~ ., data = numeric_data_log, subset = train,
family = "binomial")
yhat.log = predict(log.reg, type = "response", newdata =
numeric_data_log[-train,])
glm.pred = rep("0", 197)
glm.pred[yhat.log > .5]= "1"
table(glm.pred, test)
```

```
accuracy = (158 + 8) / (8 + 4 + 27 + 158)
accuracy
#10-fold cv
set.seed(1)
data.copy = numeric_data_log
data.copy$QD6_bin = factor(data.copy$QD6_bin, levels = c("1", "0"))
cvResults = train(QD6_bin ~ ., data = data.copy,
                  method = "glm",
                  family = "binomial",
                  trControl = trainControl(method = "cv", number =
10))
summary(cvResults)
cvResults

#random forest --- classification
set.seed(1)
rf.mod.class = randomForest(QD6_bin ~ ., data = data.copy[train,],
                    mtry = 4, ntrees = 500, importance = TRUE)
test = data.copy[-train, "QD6_bin"]
yhat = predict(rf.mod.class, type = "response", newdata =
data.copy[-train,])
yhat_num <- as.numeric(as.character(yhat))
rf.pred = rep("0", 197)
rf.pred[yhat_num > .5]= "1"
table(rf.pred, test)
accuracy = (156 + 3) / (6 + 3 + 156 + 32)
accuracy

#10-fold classification
#cvResults <- rfcv(x, y, cv.fold = 10, ntree = 500, step = 0.01, scale
= "log")

#boosting --- classification
set.seed(1)
test = data.copy[-train, "QD6_bin"]
boost.mod.class = gbm(QD6_bin ~ ., data = numeric_data_log[train,],
                n.trees=5000, interaction.depth=3,
                shrinkage = 0.01, cv.folds = 5)
yhat.boost = predict(boost.mod.class, type = "response", newdata =
numeric_data_log[-train ,],
                     n.trees=5000)
boost.yhat_num <- as.numeric(as.character(yhat.boost))
boost.pred = rep("0", 197)
boost.pred[boost.yhat_num > .5]= "1"
```

```
table(boost.pred, test)
accuracy = (146 + 6) / (16 + 6 + 29 + 146)
accuracy

#### importance analysis and plotting -----

# Missingness
financial = c('IMP_Q3MEDICALDEBT_D', 'IMP_Q3EDUCDEBT_D', 'Q5C20X_TC',
'QA9X_TC')
non_financial = c('Q6B27X_TC', 'IMP_AGE_TC', 'Q2C2_11', 'QI4',
'IMP_PFOOD1', 'IMP_HEALTH')
response = c('QD6')
df = data.frame(da38062.0001[c(financial, non_financial, response)])
sum(complete.cases(df)) # 456 complete cases
colSums(is.na(df))

nrow(da38062.0001)
sum(complete.cases(df)) / nrow(da38062.0001)
sum(complete.cases(df))

subsetted_data_reg = complete_cases(da38062.0001, response=c('QD6'))
subsetted_data_log = complete_cases(da38062.0001, response=c('QD6'))
subsetted_data_log$QD6_bin = as.integer(subsetted_data_reg$QD6 > 5)
subsetted_data_log = subsetted_data_log[, names(subsetted_data_log) !=
'QD6']
numeric_data_reg = classes_to_int(subsetted_data_reg, binary_vars)
numeric_data_log = classes_to_int(subsetted_data_log, binary_vars)
numeric_data_reg
numeric_data_log

sum(numeric_data_reg$QD6 > 5) / nrow(numeric_data_reg)


set.seed(1)
num_train = floor(0.7 * nrow(numeric_data_reg))
train_indices = sample(1:nrow(numeric_data_reg), num_train)

nrow(numeric_data_reg) - 459

train_reg = numeric_data_reg[train_indices, ]
test_reg = numeric_data_reg[-train_indices, ]

train_log = numeric_data_log[train_indices, ]
test_log = numeric_data_log[-train_indices, ]
```

```
# Random forest full model
rf.mod = randomForest(QD6 ~ ., data = train_reg,
                      mtry = 4, ntrees = 500, importance = TRUE)
importance(rf.mod)
varImpPlot(rf.mod)
yhat = predict(rf.mod, test_reg)
testMSE = mean((yhat - test_reg$QD6)^2)
testMSE

# Naive calculations
mse_reg_naivemean = mean((test_reg$QD6 - mean(train_reg$QD6))^2)
mse_reg_naivemean # 3.076
mse_log_naiveone = sum(test_log$QD6 == 1) / nrow(test_log)
mse_log_naiveone # 0.8232

full_mod = lm(QD6 ~ ., data=numeric_data_reg, subset=train_indices)
summary(full_mod)

log_mod = glm(QD6_bin ~ ., data=numeric_data, family=binomial)
summary(log_mod)

mod = lm(QD6 ~ ., data=numeric_data_reg, subset=train_indices)
summary(mod)
mod.pred = predict(mod, test_reg)

# Reduced LR model (3 variables)
mod_reduced = lm(QD6 ~ IMP_PFOOD1 + IMP_HEALTH + IMP_Q3MEDICALDEBT_D,
data=numeric_data_reg, subset=train_indices)
summary(mod_reduced)
mod.reduced.pred = predict(mod_reduced, test_reg)
test_mse_all = mean((test_reg$QD6 - mod.pred)^2)
test_mse_all
test_mse_reduced = mean((test_reg$QD6 - mod.reduced.pred)^2)
test_mse_reduced

plot(numeric_data$IMP_HEALTH, numeric_data$QD6)
plot(numeric_data$IMP_PFOOD1, numeric_data$QD6)
summary(mod2)

# Response variable distribution plot
hist(numeric_data_reg$QD6)
abline(v=mean(numeric_data_reg$QD6),col='red')
```

```
text(mean(numeric_data_reg$QD6), 100, "Mean=7.0457", adj = c(-0.1,
-0.5), col='red')
mean(numeric_data_reg$QD6)
```