Elizabeth Sabiniano
862188616

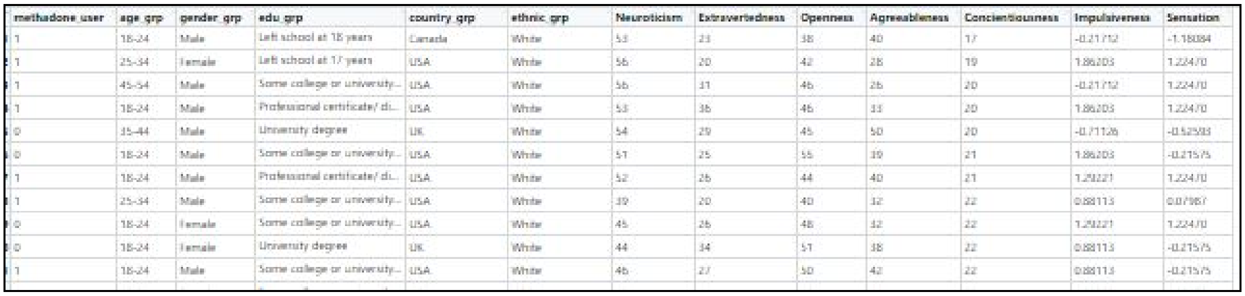
Background

From the UCI Machine Learning Repository is a data set from the study conducted by Fehrman, et al., which is *The Five Factor Model of personality and evaluation of drug consumption risk*. The goal of the study is to discern personality traits that poses the risk of drug consumption and misuse by an individual. The data was obtained from an anonymous online survey posted online, which was made available to English-speaking countries for people aged 18 and over from March 2011 to March 2012. It yielded 2051 respondents and was eventually narrowed down to 1885 respondents due to invalid answers and completeness.

The survey comprised of 12 input features with demographic information and personality attributes.[1] The response variable is the individual's use of legal and illegal drugs. The demographic information asked in the survey included the age, gender, country of residence, ethnicity, and level of education of the respondent. Personality measurements are scored based on the Revised NEO Five-Factor Inventory (NEO-FFI-R), the Baratt Impulsiveness Scale version 11 (BIS-11), and Impulsivity Sensation-Seeking scale (ImpSS). The NEO-FFI-R measures the participants tendencies for *Neuroticism* (N), *Extraversion* (E), *Openness to experience* (O), *Agreeableness* (A), and *Conscientiousness* (C) whereas BIS-11 measures the impulsiveness and ImpSS measures a person's tendency for sensation seeking. Higher personality scores imply that the respondent is more than likely to have the specific trait.

Of the 18 drugs classified using the 12 input features described in the preceding paragraph, I chose Methadone. Methadone is a synthetic opioid drug that is used to counteract addictions and withdrawal symptoms from other drugs, such as morphine, heroin, and other opioids.

In my job at the Workers' Compensation of Insurance Rating Bureau (WCIRB), we are currently looking into prescription opioid drugs within the workers' compensation industry to identify potentially problematic providers and methadone is one of the drugs we have been examining. I am curious to see whether the data can offer some perspective on to which a user's demographic and personality traits can put them at a risk of potential misuse.



methadone_user	age_grp	gender_grp	edu_grp	country_grp	ethnic_grp	Neuroticism	Extravertedness	Openness	Agreeableness	Concientiousness	Impulsiveness	Sensation
1	18-24	Male	Left school at 18 years	Canada	White	53	23	38	40	17	-0.21712	-1.18084
1	25-34	Female	Left school at 17 years	USA	White	56	20	42	28	19	1.86203	1.22470
1	45-54	Male	Some college or university…	USA	White	56	31	46	26	20	-0.21712	1.22470
1	18-24	Male	Professional certificate/ di…	USA	White	53	36	46	33	20	1.86203	1.22470
0	35-44	Male	University degree	UK	White	54	29	45	50	20	-0.71126	-0.52593
0	18-24	Male	Some college or university…	USA	White	51	25	55	39	21	1.86203	-0.21575
1	18-24	Male	Professional certificate/ di…	USA	White	52	26	44	40	21	1.29221	1.22470
1	25-34	Male	Some college or university…	USA	White	39	20	40	32	22	0.88113	0.07987
0	18-24	Female	Some college or university…	USA	White	45	26	48	32	22	1.29221	1.22470
0	18-24	Female	University degree	UK	White	44	34	51	38	22	0.88113	-0.21575
1	18-24	Male	Some college or university…	USA	White	46	27	50	42	22	0.88113	-0.21575

Figure 1. Snippet of the data with the categorical variables not transformed.

[1] Each categorical variables in the dataset has been converted to numerical values using polychoric correlation and nonlinear CatPCA (Categorical Principal Component Analysis)

Data Exploration

From looking at the data, there are 417 methadone users out of 1885 participants. Figure 2 displays the demographics of the methadone users v.s. non-users. We can see that most of the methadone users are between the ages of 18-24 and most are male. About 15% of the methadone users are from the USA and 20% are white without a degree or certificate.

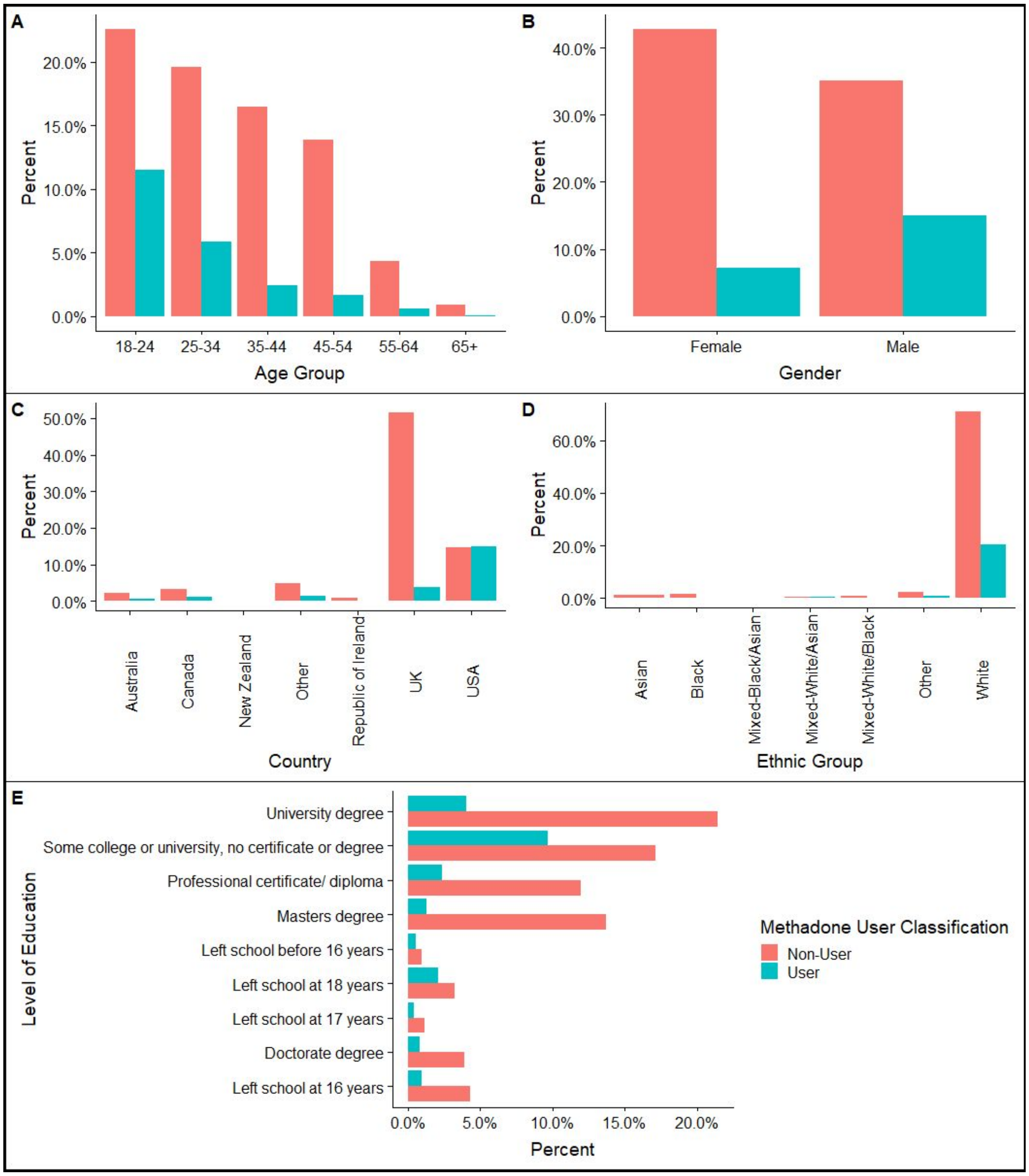


Figure 2. Demographics of Methadone users and non-users.

Figure 3 shows the distribution of the personality trait scores among methadone users and non-users. We can see that the methadone users have higher shares of Neuroticism than the other personal trait characteristics. Additionally, Fehrman, et al. describes that high N, low

C, and low A describe “negative urgency”, which is the tendency to act rashly when distressed and is high for users of illegal drugs.

The average scores for the impulsiveness and sensation-seeking traits are taken since the data is provided without the raw scores but with the scaled scores. We can see that methadone users have higher average scores for impulsivity and sensation-seeking. According to Fehrman, et al., higher tendencies for sensation-seeking correlates with substance-misuse.

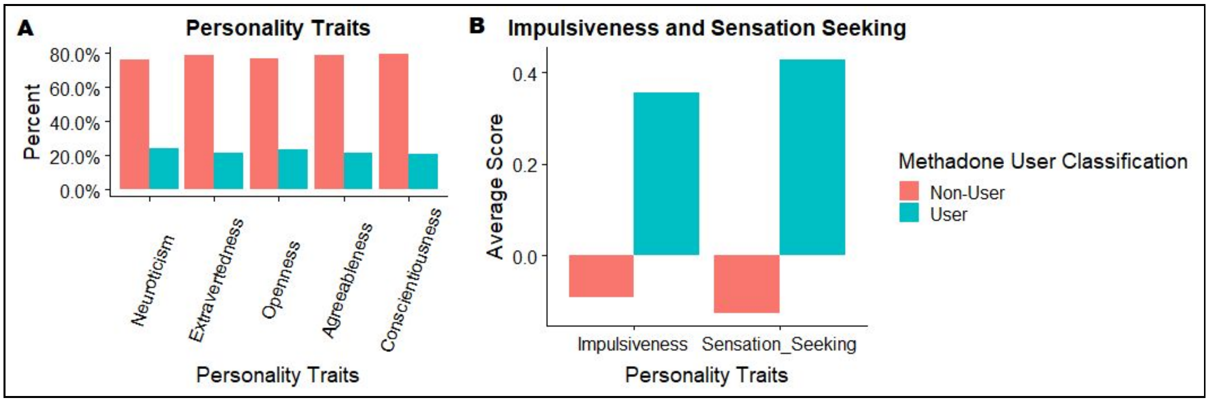


Figure 3. Distribution of personality trait and behavioral scores for methadone users and non-users.

Classification Task

In order to understand the data intuitively, I converted the 5 categorical variables in the data that were quantified using polychoric correlation and nonlinear CatPCA back into their respective actual categories then into dummy variables for the classification modelling. As I am not familiar with the methods applied for categorical conversion, this step is merely for the benefit of my understanding. This prompted me to apply all the classification methods selected below to both of the following data sets: (1) with the quantified categorical variables and (2) with the dummy categorical variables. I expected the models to perform similarly across the two sets. However, using the metrics set forth below, the “best” model for each set is different.

The classification techniques tested in this project are the *k* Nearest Neighbors, decision tree, naive bayes, logistic regression classifier, and support vector machine with polynomial and radial kernels.[2] For each classification, a 70:30 split ratio was applied to bifurcate training and testing sets. Additionally, each method’s accuracy, specificity, and sensitivity rates are taken as a measure of which classification is best fit for the data. The final metrics from the tables are from the models validated by a 10 fold cross validation approach.

High sensitivity and high specificity are the metrics we want to see in a great model. From this idea, the criterion used for selecting the “best” model is the one with the lowest difference between the sensitivity and specificity rates. This would imply that the classifier with the lowest difference between the two metric have sensitivity and specificity rate closer to each other. Accuracy would also be a secondary metric. However, since the target variable is not well-balanced, using accuracy as the main criterion may not be appropriate.

[2] Linear discriminant analysis was also applied, but collinearity issue was an issue for both of the data sets and therefore was not included in the list of classifiers tested.

Hence from the main criterion, the decision tree is the model that is most appropriate for the data with the dummy categorical variables because it has the lowest difference between its sensitivity rate and its specificity rate. Additionally, the decision tree classifier has the highest accuracy out of all the models applied to the data as shown in Table 1.

Table 1. Classification metrics for the data with dummy categorical variables.

Classifier	**Sensitivity**	**Specificity**	**Accuracy**	**Difference(Sensitivity, Specificity)**
*k*NN Classifier	90.85%	39.84%	79.29%	51.01%
Decision Tree	91.99%	47.66%	81.95%	44.33%
Naive Bayes	100%	0%	77.35%	100.00%
LR classifier	91.53%	42.97%	80.53%	48.56%
SVM (polynomial)	90.39%	44.53%	80%	45.86%
SVM (radial)	91.76%	39.84%	80%	51.92%

Regarding the classification results using the quantified categorical data provided, the naive bayes classifier performed the best in terms of the criterion outlined above. Its accuracy rate is also well above 70%, which is not as high as the SVM models, but the model has the best overall scores for all metrics combined compared to the other models examined.

Table 2. Classification model metrics for the data with quantified categorical data.

Classifier	**Sensitivity**	**Specificity**	**Accuracy**	**Difference(Sensitivity, Specificity)**
*k*NN Classifier	88.03%	43.88%	77.17%	44.15%
Decision Tree	81.69%	56.12%	75.40%	25.57%
Naive Bayes	75.12%	64.75%	72.57%	10.37%
LR classifier	84.51%	52.52%	76.64%	31.99%
SVM (polynomial)	88.50%	49.64%	78.94%	38.86%
SVM (radial)	89.91%	43.17%	78.41%	46.74%

The tree classifier and the naive bayes classifiers worked the best for the dummy categorical data and quantified categorical data, respectively. From Figure 3, the two classifiers are behaving similarly when it comes to the conscientiousness and neuroticism scores. The naive bayes did not perform well on the data with categorical dummy variables but performed the best with the data with quantified categorical variables. The naive bayes model classifies everyone in the data with the categorical dummy variables as non-methadone users.

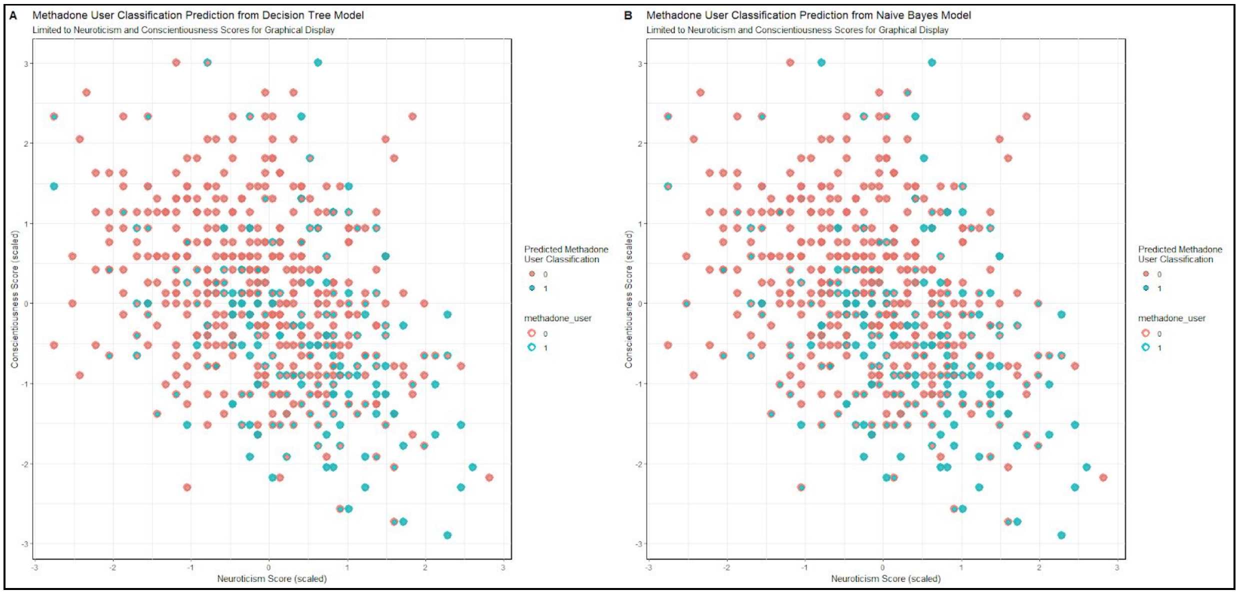


Figure 3. Decision tree classifier results on data with dummy categorical variables (left) and naive bayes classifier results on the data with quantified variables (right).

When I cross validated the naive bayes model for the data with categorical dummy variables, the model prompted warnings for the zero variances within the dummy variables. This was not the case in the data with quantified categorical variable due to all the transformations and input features ranking used for each quantification and standardization. This shows that a good classifier highly depends on if the data has been cleaned and prepped properly. The data provided by Fehrman, et al. was well prepared and transformed, which made it easy to test different classification models. However, it was necessary to transform the data into the traditional dummy categories to translate the information provided in a simplified manner. The data set with quantified categorical variables generally worked well with all the classification models, but understanding the models built for it were harder to explain. This was evident when I examined the tree model whose split node was the quantified country variable. It was not easily understandable what country being greater than a certain number is.

Overall, the classification methods applied to classify methadone users from non-users worked pretty well with the exception of the naive bayes for the data with dummy categorical variables. Each method on average has a sensitivity rate of 85% , specificity rate of 43%, and an accuracy of 80%. As the next steps in this study, I would delve into the principal component analysis of the input features and see which variables to combine for dimension reduction and maximum variance. These next steps would likely mitigate the variance and collinearity problems that emerged from the other classification methods applied, specifically LDA and naive bayes.

References

- Fehrman E, Mirkes EM, Muhammad AK, Egan V, Gorban AN. The Five Factor Model of personality and evaluation of drug consumption risk.
- Fehrman E, Egan V. Drug consumption, collected online March 2011 to March 2012, English-speaking countries. ICPSR36536-v1. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 2016-09-09. Deposited by Mirkes E. http://doi.org/10.3886/ICPSR36536.v1
- McCrae RR, Costa PT. A contemplated revision of the NEO Five-Factor Inventory. Personality and Individual Differences. 2004; 36(3):587–596. 29.
- Stanford MS, Mathias CW, Dougherty DM, Lake SL, Anderson NE, Patton JH. Fifty years of the Barratt Impulsiveness Scale: An update and review. Personality and Individual Differences. 2009; 47(5):385–395.
- Zuckerman M. Behavioral expressions and biosocial bases of sensation seeking. New York: Cambridge University Press; 1994.

R Code

```
#### STAT 208 PROJECT #####

library(data.table)
library(caret)
library(MASS)
library(class)
library(e1071)
library(rpart)
library(rpart.plot)
library(rattle)
library(stringr)
library(Hmisc)
library(gmodels)

# set location
proj.loc <- 'C:/UCR/STAT 208/Project'

# load data
setwd(proj.loc)
db_proj <- fread('drug_consumption.data')
nscore <- fread('NScoreMap.csv') # neuroticism
escore <- fread('EScoreMap.csv') # extravertedness
oscore <- fread('OScoreMap.csv') # openness to experience
ascore <- fread('AScoreMap.csv') # agreeableness
cscore <- fread('cscoremap.csv') # conscientiousness

db_names <- c('ID', 'age', 'gender', 'education', 'country', 'ethnicity'
         , 'neuroticism', 'extraversion', 'openness', 'agreeableness'
         , 'conscientiousness', 'impulsiveness', 'sensation'
         , 'alcohol', 'amphet', 'amyl', 'benzos', 'caff', 'cannabis'
         , 'choc', 'coke', 'crack', 'ecstasy', 'heroin', 'ketamine'
         , 'legalh', 'lsd', 'meth', 'mushroom', 'nicotine', 'semer'
         , 'vsa')

setnames(db_proj, names(db_proj), db_names)

# convert to binary classification on a yearly basis per doc desc: user/non-user
drug_lst <- db_names[14:32]

db_proj[, 14:32 := lapply(.SD, function(x) ifelse(x %in% c('CL0', 'CL1')
                              , 'Non-User', 'User'))
     , .SDcols = 14:32]

# only focusing on methadone, a type of opioid
db_methadone <- db_proj[, c(1:13, 28)]
db_methadone[, methadone_user := ifelse(meth == 'User', 1, 0)]
```

```
# Let's look at the personality classes for each of the user/non-user group ----
# create a copy of the non-transformed data
db_meth_graph <- copy(db_methadone)

# demographics ----
#age
db_meth_graph[age==-0.95197, age_grp := '18-24']
db_meth_graph[age==-0.07854, age_grp := '25-34']
db_meth_graph[age==0.49788, age_grp := '35-44']
db_meth_graph[age==1.09449, age_grp := '45-54']
db_meth_graph[age==1.82213, age_grp := '55-64']
db_meth_graph[age==2.59171, age_grp := '65+']

# gender
db_meth_graph[gender == 0.48246, gender_grp := 'Female']
db_meth_graph[gender == -0.48246, gender_grp := 'Male']

# education
db_meth_graph[education == -2.43591
        , edu_grp := 'Left school before 16 years']
db_meth_graph[education == -1.73790
        , edu_grp := ' Left school at 16 years']
db_meth_graph[education == -1.43719
        , edu_grp := 'Left school at 17 years']
db_meth_graph[education == -1.22751
        , edu_grp := 'Left school at 18 years']
db_meth_graph[education == -0.61113
        , edu_grp := 'Some college or university, no certificate or degree']
db_meth_graph[education == -0.05921
        , edu_grp := 'Professional certificate/ diploma']
db_meth_graph[education == 0.45468
        , edu_grp := 'University degree']
db_meth_graph[education == 1.16365
        , edu_grp := 'Masters degree']
db_meth_graph[education == 1.98437
        , edu_grp := 'Doctorate degree']

# country
db_meth_graph[country == -0.09765
        , country_grp := 'Australia']
db_meth_graph[country == 0.24923
        , country_grp := 'Canada']
db_meth_graph[country == -0.46841
        , country_grp := 'New Zealand']
db_meth_graph[country == -0.28519
        , country_grp := 'Other']
db_meth_graph[country == 0.21128
```

```
            , country_grp := 'Republic of Ireland']
db_meth_graph[country == 0.96082
            , country_grp := 'UK']
db_meth_graph[country == -0.57009
            , country_grp := 'USA']

# ethnicity
db_meth_graph[ethnicity == -0.50212, ethnic_grp := 'Asian']
db_meth_graph[ethnicity == -1.10702, ethnic_grp := 'Black']
db_meth_graph[ethnicity == 1.90725, ethnic_grp := 'Mixed-Black/Asian']
db_meth_graph[ethnicity == 0.12600, ethnic_grp := 'Mixed-White/Asian']
db_meth_graph[ethnicity == -0.22166, ethnic_grp := 'Mixed-White/Black']
db_meth_graph[ethnicity == 0.11440, ethnic_grp := 'Other']
db_meth_graph[ethnicity == -0.31685, ethnic_grp := 'White']

plot_age <-
ggplot(db_meth_graph, aes(age_grp, ..count../sum(..count..))) +
  geom_bar(aes(fill = meth), position = "dodge") +
  scale_y_continuous(labels = scales::percent) +
  labs( #title = "Age Group Distribution Across Methadone Users and Non-Users",
       y = "Percent", x = "Age Group"
      , fill = 'Methadone User Classification') +
  theme(panel.grid.major = element_line(),
       panel.grid.minor = element_line(),
       panel.background = element_rect(colour = "black", size=1)
       , legend.position = 'none')
# plot_age
# ggsave('Age_Demographics_Methadone.png')

plot_gender <-
ggplot(db_meth_graph, aes(gender_grp, ..count../sum(..count..))) +
  geom_bar(aes(fill = meth), position = "dodge") +
  scale_y_continuous(labels = scales::percent) +
  labs(#title = "Gender Distribution Across Methadone Users and Non-Users",
       y = "Percent", x = "Gender"
      , fill = 'Methadone User Classification') +
  theme(panel.grid.major = element_line(),
       panel.grid.minor = element_line(),
       panel.background = element_rect(colour = "black", size=1)
       , legend.position = 'none')

plot_edu <-
ggplot(db_meth_graph, aes(edu_grp, ..count../sum(..count..))) +
  geom_bar(aes(fill = meth), position = "dodge") +
  scale_y_continuous(labels = scales::percent) +
  labs(#title = "Education Level Distribution Across Methadone Users and Non-Users",
       y = "Percent", x = "Level of Education"
      , fill = 'Methadone User Classification')+
  coord_flip() +
  theme(panel.grid.major = element_line(),
        panel.grid.minor = element_line(),
        panel.background = element_rect(colour = "black", size=1)
        )

plot_country <-
ggplot(db_meth_graph, aes(country_grp, ..count../sum(..count..))) +
  geom_bar(aes(fill = meth), position = "dodge") +
  scale_y_continuous(labels = scales::percent) +
  labs(#title = "Country Distribution Across Methadone Users and Non-Users",
        y = "Percent", x = "Country"
       , fill = 'Methadone User Classification') +
  theme(panel.grid.major = element_line(),
        panel.grid.minor = element_line(),
        panel.background = element_rect(colour = "black", size=1)
        , legend.position = 'none'
        , axis.text.x = element_text(angle=90, vjust=0.5))

plot_eth <-
ggplot(db_meth_graph, aes(ethnic_grp, ..count../sum(..count..))) +
  geom_bar(aes(fill = meth), position = "dodge") +
  scale_y_continuous(labels = scales::percent) +
  labs(#title = "Ethnic Group Distribution Across Methadone Users and Non-Users",
       y = "Percent", x = "Ethnic Group"
       , fill = 'Methadone User Classification') +
  theme(panel.grid.major = element_line(),
        panel.grid.minor = element_line(),
        panel.background = element_rect(colour = "black", size=1)
        , legend.position = 'none'
        , axis.text.x = element_text(angle=90, vjust=0.5))

require(cowplot)
plot_grid(
  # row 1
  plot_grid(plot_age, plot_gender, nrow = 1, labels = c('A', 'B')) +
    theme(plot.background = element_rect(color = "black")),

  # row 2
  plot_grid(plot_country, plot_eth, nrow = 1, labels = c('C', 'D')) +
    theme(plot.background = element_rect(color = "black")),

  # row 2
  plot_grid(plot_edu, nrow = 1, labels = c('E')) +
    theme(plot.background = element_rect(color = "black")),

  nrow = 3)
```

```
# personality ----
setkey(nscore, Value)
setkey(db_meth_graph, neuroticism)
db_meth_graph <- nscore[, .(Nscore, Value)][db_meth_graph]
setnames(db_meth_graph, 'Value', 't_nscore')

setkey(escore, Value)
setkey(db_meth_graph, extraversion)
db_meth_graph <- escore[, .(Escore, Value)][db_meth_graph]
setnames(db_meth_graph, 'Value', 't_escore')

setkey(oscore, Value)
setkey(db_meth_graph, openness)
db_meth_graph <- oscore[, .(Oscore, Value)][db_meth_graph]
setnames(db_meth_graph, 'Value', 't_oscore')

setkey(ascore, Value)
setkey(db_meth_graph, agreeableness)
db_meth_graph <- ascore[, .(Ascore, Value)][db_meth_graph]
setnames(db_meth_graph, 'Value', 't_ascore')

setkey(cscore, Value)
setkey(db_meth_graph, conscientiousness)
db_meth_graph <- cscore[, .(Cscore, Value)][db_meth_graph]
setnames(db_meth_graph, 'Value', 't_cscore')

db_grp_sum <- db_meth_graph[, .(Nscore = sum(Nscore), Escore = sum(Escore)
                 , Oscore = sum(Oscore), Ascore = sum(Ascore)
                 , Cscore = sum(Cscore))
               , .(meth)]
db_grp_sum <- db_grp_sum[, `:=`(Neuroticism = Nscore/sum(Nscore)
                 , Extravertedness = Escore/sum(Escore)
                 , Openness = Oscore/sum(Oscore)
                 , Agreeableness = Ascore/sum(Ascore)
                 , Conscientiousness = Cscore/sum(Cscore))]
db_grp_sum[, c('Nscore', 'Escore', 'Oscore', 'Ascore', 'Cscore') := NULL]

db_grp_sum <- melt(db_grp_sum, id.vars = 'meth')

# distribution
plot_per1 <-
ggplot(db_grp_sum, aes(variable, value)) +
  geom_bar(aes(fill = meth), position = "dodge", stat="identity") +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Personality Share Traits for Methadone Users"
       , y = "Percent", x = "Personality Traits"
       , fill = 'Methadone User Classification') +
  theme(axis.text.x = element_text(angle=70, vjust=0.5))
```

```
db_grp_sum <- db_meth_graph[, .(Impulsiveness = mean(impulsiveness)
                                , Sensation_Seeking = mean(senssation))
                            , .(meth)]

db_grp_sum <- melt(db_grp_sum, id.vars = 'meth')

plot_per2 <-
ggplot(db_grp_sum, aes(variable, value)) +
  geom_bar(aes(fill = meth), position = "dodge", stat="identity") +
  labs(title = "Measure of Impulsiveness and Sensation Seeking for Methadone Users"
       , y = "Average Score", x = "Personality Traits"
       , fill = 'Methadone User Classification') +
  theme(
    panel.grid.major = element_line(),
    panel.grid.minor = element_line(),
    panel.background = element_rect(colour = "black", size=1)
  )

plot_grid(
  # row 1
  plot_grid(plot_per1, nrow = 1, labels = c('A')) +
    theme(plot.background = element_rect(color = "black")),

  # row 2
  plot_grid(plot_per2, nrow = 1, labels = c('B')) +
    theme(plot.background = element_rect(color = "black")),

  nrow = 2)

######### Building Preliminary Models Using Given Standardized Values #########
# Classification attempt 1 - knn --------------------------------------------
# select only relevant variables, dropping ID and character var meth
knn_train <- db_methadone[1:1320, 2:13] # approximately 70%
knn_test <- db_methadone[1321:1885, 2:13] # approximately 30%

# get labels
knn_train_labels <- db_methadone[1:1320, methadone_user]
knn_test_labels <- db_methadone[1321:1885, methadone_user]

# fit the knn model

knn_test_pred <- knn(train = knn_train, test = knn_test
                     , cl = knn_train_labels, k = 4)
# check performance - various ways of calculating confusion matrix
checktbl <- CrossTable(knn_test_labels, knn_test_pred, prop.chisq = F)

knn_cm <- confusionMatrix(knn_test_pred, as.factor(knn_test_labels))
```

```
# Display confusion matrix
confusion <- checktbl$t
colnames(confusion) <- c("Predicted = 0","Predicted = 1")
rownames(confusion) <- c("True = 0","True = 1")
print(as.table(confusion))

# FPR = FP/(FP+TN)
FPR = confusion[1,2]/sum(confusion[1,1:2])
print(FPR)
# FNR = FN/(FN+TP)
FNR = confusion[2,1]/sum(confusion[2,1:2])
print(FNR)
# OER
Mis_Er = (confusion[1, 2] + confusion[2, 1])/sum(confusion)
print(Mis_Er)
# accuracy
Acc = (confusion[1, 1] + confusion[2, 2])/sum(confusion)
print(Acc)

# k fold cross validation
require(caret)
set.seed(1234)
trControl <- trainControl(method  = "cv", number = 10)
knn_tst_fit <- train(as.factor(methadone_user) ~ .,
          method     = "knn",
          tuneGrid   = expand.grid(k = 1:10),
          trControl  = trControl,
          metric     = "Accuracy",
          data       = cbind(knn_train, methadone_user = knn_train_labels))

knn_test_pred2 <- predict(knn_tst_fit, knn_test)
confusionMatrix(knn_test_pred2, as.factor(knn_test_labels))

# Classification Attempt 2 - Decision Tree -----------------------------------
tree_train <- db_methadone[1:1320, c(2:13,15)]
tree_test <- db_methadone[1321:1885, c(2:13, 15)]
tree_train[, methadone_user := as.factor(methadone_user)]
tree_test[, methadone_user := as.factor(methadone_user)]

fittree <- rpart(as.factor(methadone_user) ~., tree_train, method="class")
rpart.plot(fittree)
summary(fittree)

rpartpred <- predict(fittree, tree_test, type="class")
confusionMatrix(rpartpred,tree_test$methadone_user)

prp(fittree, faclen = 0, cex = 0.8, extra = 1)
```

```
# total count
tot_count <- function(x, labs, digits, varlen)
{paste(labs, "\n\nn =", x$frame$n)}
## Decision Tree
prp(fittree, faclen = 0, cex = 0.8, node.fun=tot_count)

# prune the tree
bestcp <- fittree$cptable[which.min(fittree$cptable[,"xerror"]),"CP"]

#Pruning & classification matrix of Pruning
pruned <- prune(fittree, cp = bestcp)
prp(pruned, faclen = 0, cex = 0.8, extra = 1)

predictions <- predict(pruned, tree_test, type="class")
tree_cm <- confusionMatrix(predictions,tree_test$methadone_user)

trControl <- trainControl(method  = "cv", number = 10)
dt_tst_fit <- train(as.factor(methadone_user) ~ .,
           method     = "rpart",
           trControl  = trControl,
           metric     = "Accuracy",
           data       = tree_train)
dt_test_pred2 <- predict(dt_tst_fit, tree_test)
confusionMatrix(dt_test_pred2, tree_test$methadone_user)

# Classification Attemp 3 - Naive Bayes ------------------------------------
nb_train <- db_methadone[1:1320, c(2:13, 15)] # approximately 70%
nb_test <- db_methadone[1321:1885, c(2:13, 15)] # approximately 30%
nb_test[, methadone_user := as.factor(methadone_user)]

nb_fit <- naiveBayes(as.factor(methadone_user) ~., data = nb_train)
nb_pred <- predict(nb_fit, nb_test[,-13])
nb <- confusionMatrix(nb_pred, nb_test$methadone_user)

trControl <- trainControl(method  = "cv", number = 10)
nb_tst_fit <- train(as.factor(methadone_user) ~ .,
           method     = "nb",
           trControl  = trControl,
           metric     = "Accuracy",
           data       = nb_train)


nb_test_pred2 <- predict(nb_tst_fit, nb_test)
confusionMatrix(nb_test_pred2, nb_test$methadone_user)

# Classification Attemp 4 - Logistic Model ----------------------------------
log_train <- db_methadone[1:1320, c(2:13, 15)] # approximately 70%
```

```
log_test <- db_methadone[1321:1885, c(2:13, 15)] # approximately 30%
log_test[, methadone_user := as.factor(methadone_user)]

logitmodel <- glm(as.factor(methadone_user)~., data = log_train
          , family="binomial")
print(summary(logitmodel))

logit_pred <- predict(logitmodel, type = 'response', newdata = log_test)
logit_confusion <- table(log_test$methadone_user, logit_pred > 0.3)
colnames(logit_confusion) <- c("Predicted = 0","Predicted = 1")
rownames(logit_confusion) <- c("True = 0","True = 1")
print(as.table(logit_confusion))

# FPR = FP/(FP+TN)
FPR = logit_confusion[1,2]/sum(logit_confusion[1,1:2])
print(FPR)
# FNR = FN/(FN+TP)
FNR = logit_confusion[2,1]/sum(logit_confusion[2,1:2])
print(FNR)

# OER
logit_Mis_Er = (logit_confusion[1, 2] + logit_confusion[2, 1])/sum(logit_confusion)
print(Mis_Er)
# accuracy
logit_Acc = (logit_confusion[1, 1] + logit_confusion[2, 2])/sum(logit_confusion)
print(Acc)

trControl <- trainControl(method  = "cv", number = 10)
lr_tst_fit <- train(as.factor(methadone_user) ~ .,
           method     = "bayesglm",
           trControl  = trControl,
           metric     = "Accuracy",
           data       = log_train)

log_test_pred2 <- predict(lr_tst_fit, log_test)
confusionMatrix(log_test_pred2, log_test$methadone_user)

# Classification Attemp 5 - SVM --------------------------------------------
svm_train <- db_methadone[1:1320, c(2:13, 15)] # approximately 70%
svm_test <- db_methadone[1321:1885, c(2:13, 15)] # approximately 30%
svm_test[, methadone_user := as.factor(methadone_user)]

tc = tune.control(cross = 10)
svm_model <- tune.svm(as.factor(methadone_user)~., data=svm_train
         , kernel="polynomial", scale=F, gamma=1, coef0=1
         , cost=1, tunecontrol = tc)
bhat<-c(-svm_model$best.model$rho,t(svm_model$best.model$coefs)%*%svm_model$best.model$SV)
print(bhat)
```

```
svm_pred <- predict(svm_model$best.model, svm_test)

svm_poly <- confusionMatrix(svm_pred, svm_test$methadone_user)

svm_model <- tune.svm(as.factor(methadone_user)~., data=svm_train
           , kernel="radial", scale=F, gamma=1, coef0=1, cost=1
           , tunecontrol = tc)
bhat<-c(-svm_model$best.model$rho,t(svm_model$best.model$coefs)%*%svm_model$best.model$SV)
print(bhat)

svm_pred <- predict(svm_model$best.model, svm_test)

svm_radial <- confusionMatrix(svm_pred, svm_test$methadone_user)


############# End of Preliminary Models Using Standardized Values #############
# Dummy variable transformation for categorical variables --------------------
db_meth_dum <- db_meth_graph[, .(methadone_user
                      # categorical variables to be dummified
                      , age_grp, gender_grp, edu_grp, country_grp
                      , ethnic_grp
                      # scaled personality/behavior score variables
                      , Neuroticism = t_nscore
                      , Extravertedness = t_escore
                      , Openness = t_oscore
                      , Agreeableness = t_ascore
                      , Concientiousness = t_cscore
                      , Impulsiveness = impulsiveness
                      , Sensation = sensation)]
# create dummies for the categories, one column each category
db_meth_dum <-fastDummies::dummy_cols(db_meth_dum
                         , select_columns = c('age_grp'
                                       , 'gender_grp'
                                       , 'edu_grp'
                                       , 'country_grp'
                                       , 'ethnic_grp'))
# drop the categorical variables and keep respective dummy versions
db_meth_dum[, c('age_grp', 'gender_grp', 'edu_grp', 'country_grp'
          , 'ethnic_grp') := NULL]
######### Building Preliminary Models Using Given Standardized Values #########
partition <- createDataPartition(db_methadone$methadone_user, p =.7, list = F)
# Classification attempt 1 - knn -------------------------------------------
# select only relevant variables, dropping ID and character var meth
knn_train <- db_meth_dum[partition, 2:39] # approximately 70%
knn_test <- db_meth_dum[-partition, 2:39] # approximately 30%

# get labels
```

```
knn_train_labels <- db_meth_dum[partition, methadone_user]
knn_test_labels <- db_meth_dum[-partition, methadone_user]

# fit the knn model

knn_test_pred <- knn(train = knn_train, test = knn_test
            , cl = knn_train_labels, k = 4)
# check performance - various ways of calculating confusion matrix
knn_cm <- confusionMatrix(knn_test_pred, as.factor(knn_test_labels))

# k fold cross validation
set.seed(1234)
trControl <- trainControl(method  = "cv", number = 10)
knn_tst_fit <- train(as.factor(methadone_user) ~ .,
             method     = "knn",
             tuneGrid   = expand.grid(k = 1:10),
             trControl  = trControl,
             metric     = "Accuracy",
             data       = cbind(knn_train, methadone_user = knn_train_labels))

knn_test_pred2 <- predict(knn_tst_fit, knn_test)
confusionMatrix(knn_test_pred2, as.factor(knn_test_labels))

# Classification Attempt 2 - Decision Tree -----------------------------------
colnames(db_meth_dum) <- make.names(colnames(db_meth_dum))
tree_train <- db_meth_dum[partition]
tree_test <- db_meth_dum[-partition]
tree_train[, methadone_user := as.factor(methadone_user)]
tree_test[, methadone_user := as.factor(methadone_user)]

# clean names for the punctuations
setnames(tree_train, names(tree_train)
     , stringr::str_replace_all(names(tree_train), '[\ / - +]', ''))
setnames(tree_test, names(tree_test)
     , stringr::str_replace_all(names(tree_test), '[\ / - +]', ''))

fittree <- rpart(methadone_user~., tree_train, method="class")
rpart.plot(fittree)
summary(fittree)

rpartpred <- predict(fittree, tree_test, type="class")
confusionMatrix(rpartpred,tree_test$methadone_user)

prp(fittree, faclen = 0, cex = 0.8, extra = 1)

# total count
tot_count <- function(x, labs, digits, varlen)
{paste(labs, "\n\nn =", x$frame$n)}
```

```
## Decision Tree
prp(fittree, faclen = 0, cex = 0.8, node.fun=tot_count)

# prune the tree
bestcp <- fittree$cptable[which.min(fittree$cptable[,"xerror"]),"CP"]

#Pruning & classification matrix of Pruning
pruned <- prune(fittree, cp = bestcp)
prp(pruned, faclen = 0, cex = 0.8, extra = 1)

predictions <- predict(pruned, tree_test, type="class")
tree_cm <- confusionMatrix(predictions,tree_test$methadone_user)

trControl <- trainControl(method  = "cv", number = 10)
dt_tst_fit <- train(as.factor(methadone_user) ~ .,
                method     = "rpart",
                trControl  = trControl,
                metric     = "Accuracy",
                data       = tree_train)
dt_test_pred2 <- predict(dt_tst_fit, tree_test)
confusionMatrix(dt_test_pred2, tree_test$methadone_user)

# Classification Attemp 3 - Naive Bayes -----------------------------------
nb_train <- db_meth_dum[partition] # approximately 70%
nb_test <- db_meth_dum[-partition] # approximately 30%
nb_test[, methadone_user := as.factor(methadone_user)]
nb_train[, methadone_user := as.factor(methadone_user)]

nb_fit <- naiveBayes(as.factor(methadone_user) ~., data = nb_train)
nb_pred <- predict(nb_fit, nb_test[,-1])
nb <- confusionMatrix(nb_pred, nb_test$methadone_user)

trControl <- trainControl(method  = "cv", number = 10)
nb_tst_fit <- train(x = nb_train[, 2:39], y = nb_train$methadone_user,
                method     = "nb",
                trControl  = trControl,
                metric     = "Accuracy",
                data       = nb_train)

nb_test_pred2 <- predict(nb_tst_fit, nb_test)
confusionMatrix(nb_test_pred2, nb_test$methadone_user)

# Classification Attemp 4 - Logistic Model --------------------------------
log_train <- db_meth_dum[partition] # approximately 70%
log_test <- db_meth_dum[-partition] # approximately 30%
log_test[, methadone_user := as.factor(methadone_user)]

logitmodel <- glm(as.factor(methadone_user)~., data = log_train
              , family="binomial")
print(summary(logitmodel))

logit_pred <- predict(logitmodel, type = 'response', newdata = log_test)
logit_confusion <- table(log_test$methadone_user, logit_pred > 0.3)
colnames(logit_confusion) <- c("Predicted = 0","Predicted = 1")
rownames(logit_confusion) <- c("True = 0","True = 1")
print(as.table(logit_confusion))

# FPR = FP/(FP+TN)
FPR = logit_confusion[1,2]/sum(logit_confusion[1,1:2])
print(FPR)
# FNR = FN/(FN+TP)
FNR = logit_confusion[2,1]/sum(logit_confusion[2,1:2])
print(FNR)

# OER
logit_Mis_Er = (logit_confusion[1, 2] + logit_confusion[2, 1])/sum(logit_confusion)
print(Mis_Er)
# accuracy
logit_Acc = (logit_confusion[1, 1] + logit_confusion[2, 2])/sum(logit_confusion)
print(Acc)

trControl <- trainControl(method  = "cv", number = 10)
lr_tst_fit <- train(as.factor(methadone_user) ~ .,
               method     = "bayesglm",
               trControl  = trControl,
               metric     = "Accuracy",
               data       = log_train)

log_test_pred2 <- predict(lr_tst_fit, log_test)
confusionMatrix(log_test_pred2, log_test$methadone_user)

# Classification Attemp 5 - SVM -----------------------------------------------
svm_train <- db_meth_dum[partition] # approximately 70%
svm_test <- db_meth_dum[-partition] # approximately 30%
svm_test[, methadone_user := as.factor(methadone_user)]

svm_model <- svm(as.factor(methadone_user)~., data=svm_train
             , kernel="polynomial", scale=F, d=2, gamma=1, coef0=1, cost=1)
bhat<-c(-svm_model$rho,t(svm_model$coefs)%*%svm_model$SV)
print(bhat)

svm_pred <- predict(svm_model, svm_test)

svm_poly <- confusionMatrix(svm_pred, svm_test$methadone_user)

svm_model <- svm(as.factor(methadone_user)~., data=svm_train
            , kernel="radial", scale=F, d=2, gamma=1, coef0=1, cost=1)
bhat<-c(-svm_model$rho,t(svm_model$coefs)%*%svm_model$SV)
print(bhat)

svm_pred <- predict(svm_model, svm_test)

svm_radial <- confusionMatrix(svm_pred, svm_test$methadone_user)

# validation
svm_tst_fit <- train(as.factor(methadone_user) ~ .,
              method     = "svmPoly",
              trControl  = trControl,
              metric     = "Accuracy",
              data       = svm_train)
svm_test_pred2 <- predict(svm_tst_fit, svm_test)
confusionMatrix(svm_test_pred2, svm_test$methadone_user)

#radial
svm_tst_fit <- train(as.factor(methadone_user) ~ .,
              method     = "svmRadial",
              trControl  = trControl,
              metric     = "Accuracy",
              data       = svm_train)
svm_test_pred2 <- predict(svm_tst_fit, svm_test)
confusionMatrix(svm_test_pred2, svm_test$methadone_user)

########################## End of Classification ############################
```