# PROJECT REPORT

## Painkiller Misuse in the US

Group 35: Michael Crist

ISYE7406

04/16/2023

# Contents

## Abstract

In this report I build, evaluate, and analyze a variety of classification models to obtain the best performing model for predicting prescription painkiller misuse in US citizens. In addition to model evaluation, I determine and quantify the factors that have the greatest impact on the odds of painkiller misuse.

The dataset used in this report was obtained from the National Survey on Drug Use and Health 2015-2017. The models evaluated in the report include Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naïve Bayes, K-Nearest Neighbors (KNN), Random Forest, and Boosting. KNN was determined to be the best performing model, predicting painkiller misuse with 99.13% accuracy. The five factors with the greatest impact on odds of painkiller misuse were determined to be age, marital status, hallucinogen use, amphetamine use, and heroin use.

## Introduction / Problem Statement

I centered my project on prescription painkiller misuse in the United States. Opioid abuse/misuse results in $35 billion in healthcare costs each year in the US ("The High Price…", 2021). In the US alone, opioid-involved overdoses have nearly quadrupled from 21,089 in 2010 to 80,411 in 2021 ("Drug Overdose Death Rates", 2023). This is a problem that has plagued the US healthcare system for decades, and has negatively impacted millions of lives. This report aims to accomplish two goals:

1. Build a variety of classification models to predict whether or not a person will misuse painkillers, and determine the best performing model.
2. Determine the factors that most impact the odds of painkiller misuse, and quantify those factors' impact.

The motivation for this project is to provide the necessary analytics to implement healthcare / political policy aimed at reducing prescription painkiller misuse.

## Data Source

The dataset I used for this project can be found at the following link:
https://www.kaggle.com/datasets/thedevastator/predicting-pain-reliever-misuse-abuse

The dataset was obtained from the National Survey on Drug Use and Health 2015-2017, and published on Kaggle. This dataset contains 170317 rows and 21 columns. 18 of the columns are predictor variables, and 3 are potential response variables ("misused prescription medication", "abused prescription medication", and "misused/abused minimum prescription medication"). For the intent of my project, I will use "misused prescription medication" as the response variable. This is a binary variable, labeled "PRLMISEVR", with a 0 indicating no painkiller misuse and a 1 indicating painkiller misuse.

There are a variety of predictors in the dataset, both categorical and numeric. Examples of some of the predictor variables are age, marital status, education level, employment status, etc. The response variable, PLMISEVR, is a binary variable.

## Exploratory Data Analysis

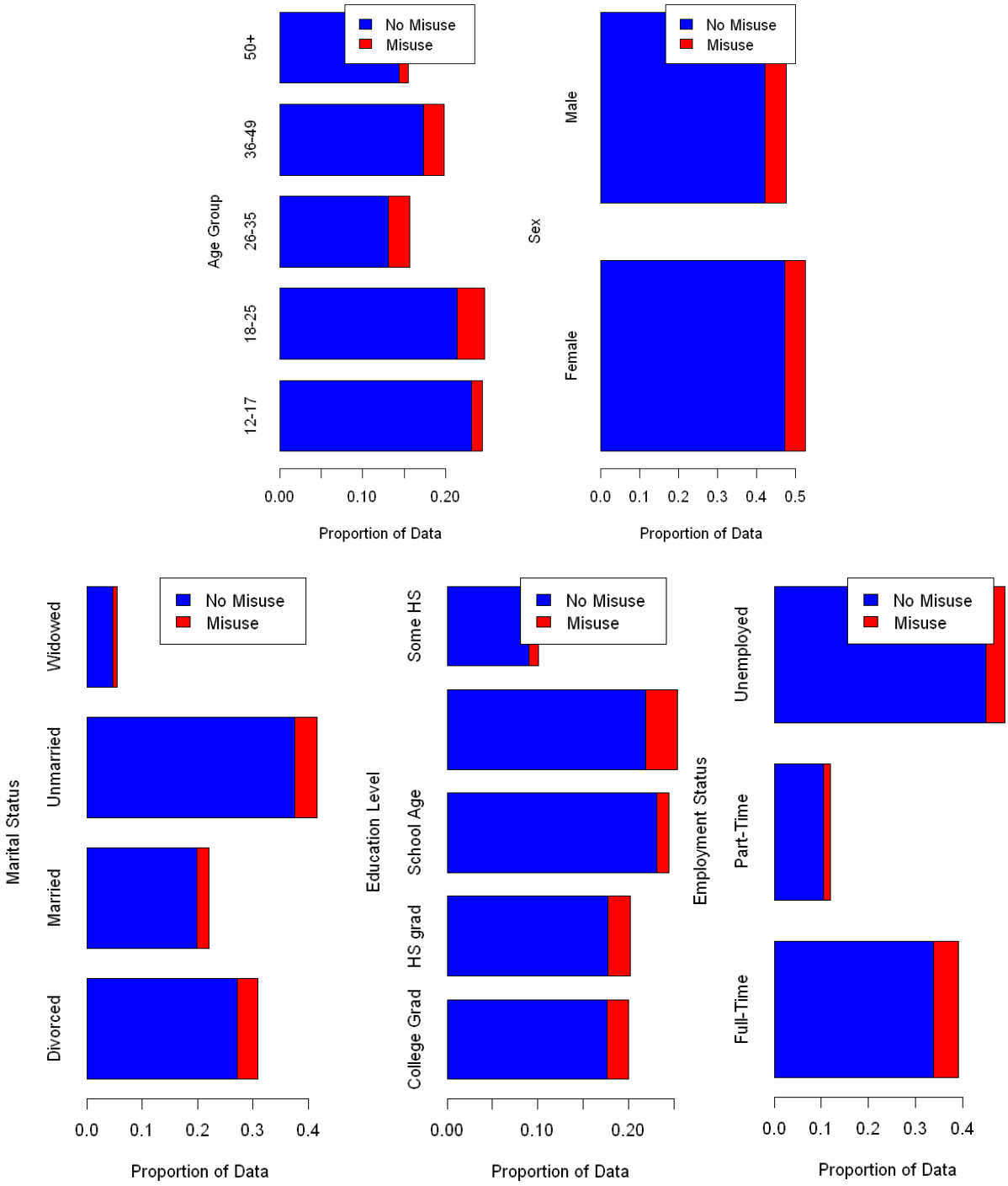
The first step in any modeling problem is to perform exploratory data analysis. The dataset has total dimensions 170317 x 21. 18 of the columns are predictor variables, and 3 are potential response variables. For the purposes of this project, I selected just one variable as the response, PRLMISEVR, and removed the other 2 possible response variables from the dataset. After doing so, the total dataset dimensions were 170317 x 19 (18 predicting variables and 1 response variable).

### Variable Definition

| Variable | Type | Label |
|---|---|---|
| Target variable: binary variable indicating pain reliever misuse | Categorical | PRLMISEVR |
| Year of data | Categorical | YEAR |
| Age range | Categorical | AGECAT |
| Sex (male or female) | Categorical | SEX |
| Marital status | Categorical | MARRIED |
| Education status | Categorical | EDUCAT |
| Employment status | Categorical | EMPLOY18 |
| Size of city / metropolitan region | Categorical | CTYMETRO |
| Health problems (Likert scale: 0-7) | Numeric | HEALTH |
| Mental health: depression, emotional distress (Likert scale: 0-10) | Numeric | MENTHLTH |
| Ever used heroin (binary) | Categorical | HEROINEVR |
| Heroin use in past year (Likert scale: 0-5) | Numeric | HEROINUSE |
| Tranquilizer use in past year (Likert scale: 0-5) | Numeric | TRQLZRS |
| Sedative use in past year (Likert scale: 0-5) | Numeric | SEDATVS |
| Cocaine and crack cocaine use in past year (Likert scale: 0-5) | Numeric | COCAINE |
| Amphetamine use in past year (Likert scale: 0-5) | Numeric | AMPHETMN |
| Hallucinogen use in past year (Likert scale: 0-5) | Numeric | HALUCNG |
| Treatment for drugs or alcohol in past year (Likert scale: 0-10) | Numeric | TRTMENT |
| Mental health treatment (Likert scale: 0-10) | Numeric | MHTRTMT |

### Categorical Variables

Many of the variables in this dataset are categorical. To explore the categorical variables, the first step I took was to plot a bar chart of each category, overlayed with the proportion of the response. This can be found below:

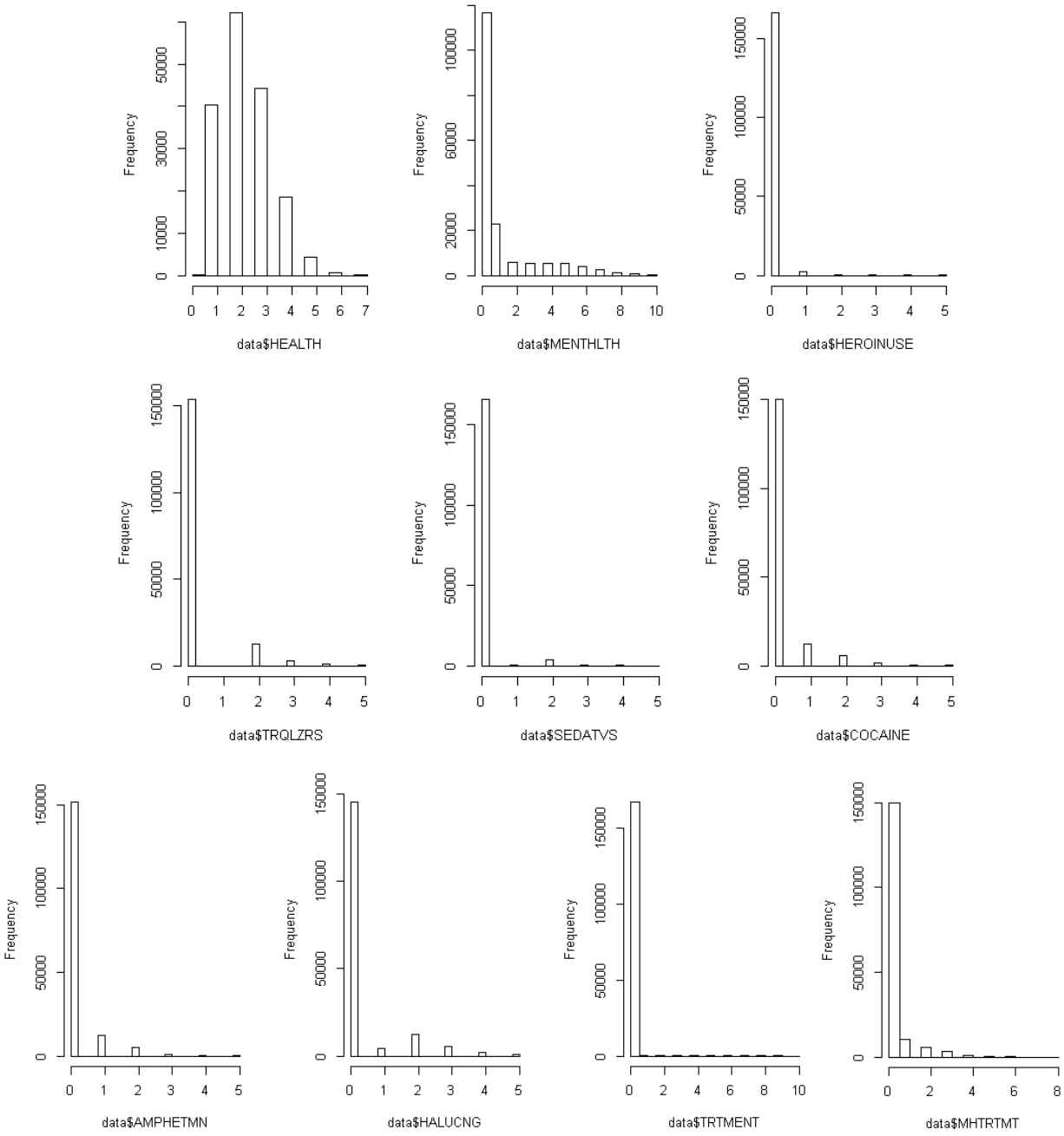



*Barplots of Categorical Variables*

Looking at the plots, there is clear correlation between the response and the value for some of the categorical variables. For example, the rate of misuse is higher for age group 18-25 than for age group 12-17. It is also apparent that the data is not evenly distributed between each of the values for each category. For example, for the "Marital Status" variable, there are many more people who are unmarried than there are widowed.

## Numeric Variables

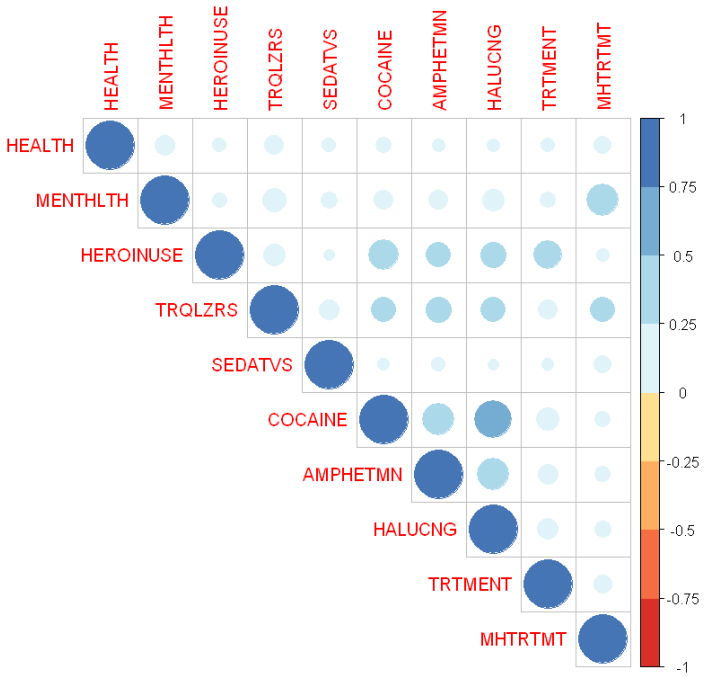
Next, I looked at the quantitative predicting variables. All the quantitative predicting variables have values that correspond to the Likert scale. The Likert scale is a ranking, usually in the range [0,10] or [0,5], that is used to express how "strongly" a person agrees or disagrees with a prompt (Mcleod). In this dataset, the Likert values for the predicting variables typically correspond to "how much". For example, the variable COCAINE has a range [0, 5]. A value of 0 means the person has not used cocaine in the past year, and a value of 5 means the person has used cocaine frequently. For the quantitative variables, I first plotted histograms of the data:




*Histograms for quantitative variables*

As is clear from the histograms above, most of the quantitative variables are biased toward the low end of the range (most of the datapoints are on the low end of the range). This is reasonably expected, because many of the variables correspond to illict drug use, and most people do not use illicit drugs. Some variables, like HEALTH, have a closer to normal distribution, although it is still heavily right-tailed.

Next, I checked the correlation between the quantitative variables:




*Correlation Matrix for quantitative variables*

From the correlation matrix above, we see that there are no negative correlations between any of the quantitative variables. All quantitative variables are positively correlated with each other, but the correlation between most variables is weak. However, there are some variables with moderate to strong positive correlation, such as COCAINE and HALUCNG. This implies that cocaine use is correlated with hallucinogen use, and vice versa.

## Methodology

For this project, I built and evaluated the following classification models for predicting painkiller misuse:

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- Naïve Bayes
- K-Nearest Neighbors (KNN)
- Random Forest
- Boosting

In addition to model building, I also performed variable analysis to determine and quantify the factors that most impact the odds of painkiller misuse. Both model building and variable analysis are discussed in the following sections.

## Splitting the Data

Prior to building the models, I split the data into a 75% training set and 25% testing set. I used the sample() function in R to randomly select 75% of the data as training and 25% as testing. The training set was used for model building, and testing set was used for model evaluation.

## Models

The following models were built, trained, and evaluated:

### Logistic Regression

In logistic regression, the response is modeled as a Bernoulli distribution, and we link the model parameters to the predictors using the log-likelihood (logit) function below. In this equation, 'p' is the probability of success (ie, a response value of '1').

$$\ln\left(\frac{\mathrm{p}}{1-\mathrm{p}}\right) = \beta_0 + \beta_1 x$$

We then optimize the coefficients by maximizing the likelihood estimator:

$$\max_{\beta_0,\beta_1,\cdots,\beta_p} \mathcal{L}(\beta_0,\beta_1,\cdots,\beta_p) = \prod_{i=1}^{n} \mathrm{p}\left(\boldsymbol{X}_{i,1},\boldsymbol{X}_{i,2},\cdots,\boldsymbol{X}_{i,p}\right)^{Y_i} \left(1-\mathrm{p}\left(\boldsymbol{X}_{i,1},\boldsymbol{X}_{i,2},\cdots,\boldsymbol{X}_{i,p}\right)\right)^{1-Y_i}$$

The default output of the logistic regression model is the log-likelihood. In my R code, I set the logistic regression output type as "response" to output the probability of painkiller misuse on the testing set, rather than the log-likelihood. After that, to get the final predictions on the testing set, I considered any probability >= 0.5 as a 1 (painkiller misuse), and any probability < 0.5 as a 0 (no painkiller misuse)

### Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naïve Bayes

In discriminant analysis, a classifier is associated with a k-dimensional vector where each of the k components represents the strength of evidence that a datapoint belongs to that class. For this project, there are two classes: 0 representing no painkiller misuse, or 1 representing painkiller misuse.

$$\hat{k} = \underset{k=1,2,\cdots,K}{\operatorname{argmax}}\, d_k(\boldsymbol{x_{new}})$$

LDA, QDA, and Naïve Bayes are all based upon the Bayes classifier, which is presented below:

$$\underset{k}{\operatorname{argmax}}\left(\log \boldsymbol{\pi_k} + \log \boldsymbol{f_k}(\boldsymbol{x})\right)$$

In the Bayes classifier, $\pi_k$ is the prior distribution and $f_k$ is the density function of the k-th class. The density function $f_k$ follows the normal distribution $N(\mu_k, \Sigma_k)$. The difference between LDA, QDA, and Naïve Bayes is in the assumptions about the density function $f_k$.

***LDA**: $\Sigma_k$ is estimated by within-sample covariance*

$$\hat{\pi}_k = \frac{n_k}{n}; \qquad \hat{\mu}_k = \frac{1}{n_k}\sum_{y_i=k} x_i$$

$$\hat{\Sigma} = \frac{1}{\sum_{k=1}^{K}(n_k - 1)}\sum_{k=1}^{K}\sum_{y_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

***QDA**: $\Sigma_k$ is estimated by the sample covariance of the k-th class*

$$\hat{\pi}_k = \frac{n_k}{n}; \qquad \hat{\mu}_k = \frac{1}{n_k}\sum_{y_i=k} x_i$$

$$\hat{\Sigma}_k = \frac{1}{n_k - 1}\sum_{y_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

***Naïve Bayes**: each component of X is independent*

$$\hat{\pi}_k = \frac{n_k}{n}; \qquad \hat{\mu}_{kj} = \frac{1}{n_k}\sum_{y_i=k} x_{ij}$$

$$\hat{\sigma}_{kj}^2 = \frac{1}{n_k - 1}\sum_{y_i=k}\left(x_{ij} - \bar{x}_{\cdot j}\right)^2$$

### K-Nearest Neighbors (KNN)

The KNN classification algorithm is the simplest algorithm explored in this project. The algorithm assigns a datapoint to a class based upon the classification of its k "nearest neighbors" in the p-dimensional space. As such, we can only use the numeric predictor variables in KNN. The algorithm is tuned by evaluating several values for k and obtaining the cross-validation error for each k value to determine the best performing algorithm.

### Random Forest

Random Forest is an "ensemble method", meaning that the model prediction is obtained from an average of many predictions. In Random Forest, we utilize the bagging algorithm to draw a bootstrap sample from the training data, and fit a base tree model to each bootstrap sample. Additionally, we randomly select only a portion of the predictors to use in each bootstrap sample. The random forest model is optimized by tuning the number of trees to use in the model, the number of predictors to use at each branching, and the minimum size to use for tree nodes. These parameters are tuned by evaluating the cross validation error for several options, and selecting the values that achieve the smallest cross validation error.

### Boosting

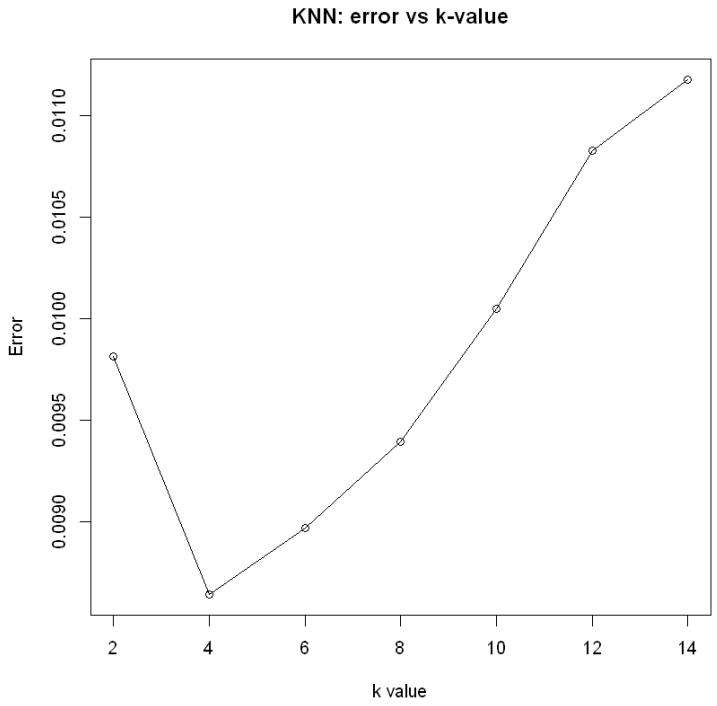
The boosting algorithm is similar to the random forest algorithm, in that it is also an ensemble method. It uses many "base learners" (ie, trees) to obtain an overall model with improved predictive performance. The final classifier is based on the weighted average of all the base learners. The boosting model is optimized by tuning the number of trees, the shrinkage parameter, and the interaction depth. The number of trees is the number of base learners used in the model. Shrinkage is a parameter in the range [0, 1] that slows the rate at which the model "learns". Interaction depth is a parameter that models the interaction between predictors.

## Model Tuning

Initially, I built all models using the default parameters in R. Once that was complete, I further tuned the KNN, Random Forest, and Boosting models.

### KNN Model Tuning

The KNN model was tuned for the optimal k-value by training the model on several k-values in the range [2, 14] and evaluating each k-value using cross validation error.




*KNN model tuning*

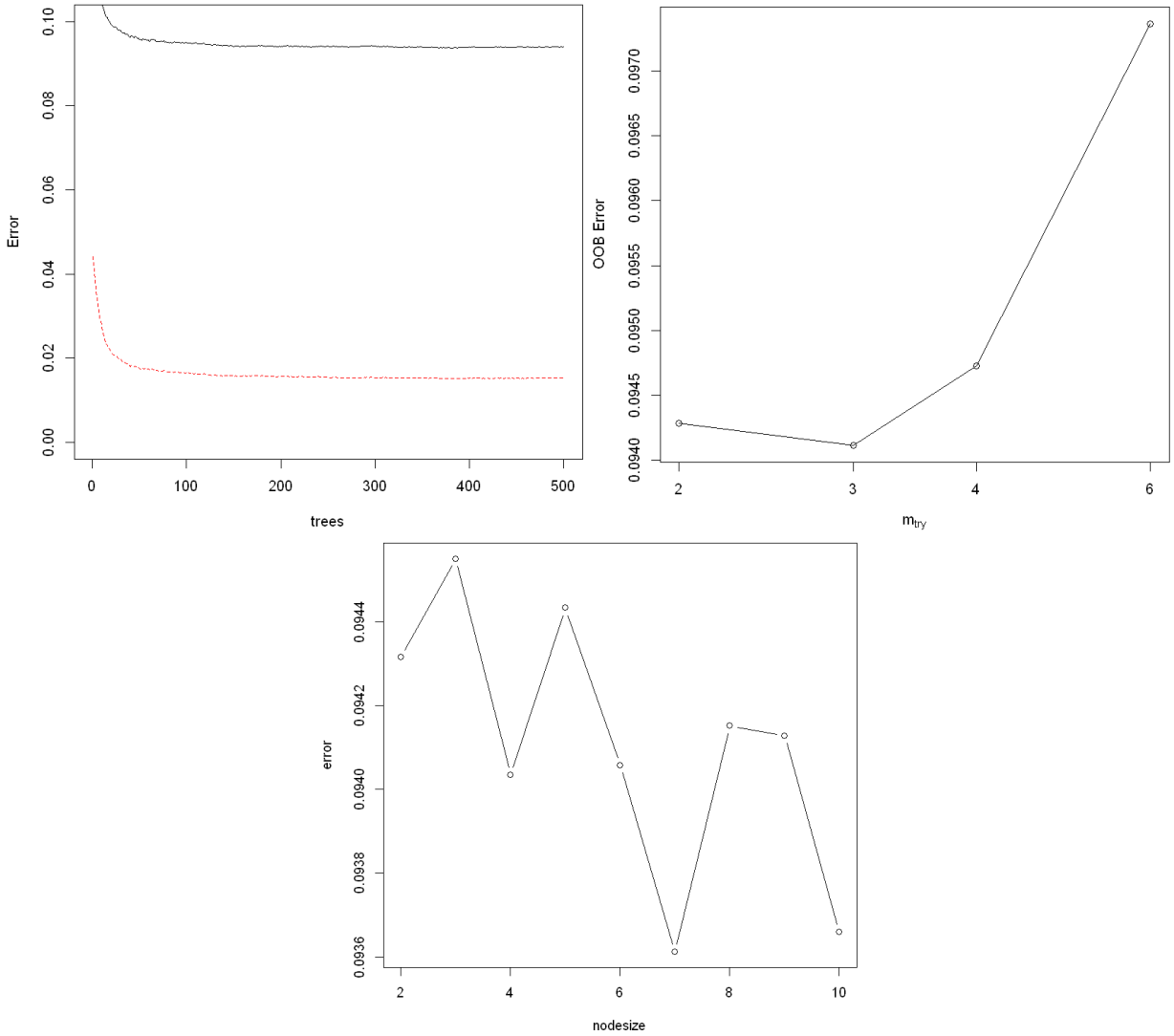
As can be seen in the plot of k-value versus cross-validation error, the optimal k-value is k=4. This is the value that minimizes the cross-validation error of the model.

### Random Forest Model Tuning

The random forest model was tuned by optimizing the hyperparameters below:

- 'ntree': number of trees to use in the forest
- 'mtry': number of predictors that are selected as candidates at each branching
- 'nodesize': minimum size of the tree nodes

I tuned each hyperparameter by evaluating a range of hyperparameter values vs the out-of-bag (OOB) error. The results are shown below:




*Random Forest model tuning*

As can be seen in the hyperparameter versus error plots above, the optimal Random Forest hyperparameters are:

- 'ntrees' = 300
- 'mtry' = 3
- 'nodesize' = 7

The error for ntrees stabilizes at ~300, meaning that we can use any value for ntrees >= 300. To balance error and computation time, we use 300 trees, which is the minimum number of trees required to minimize error.

## Boosting Model Tuning

The boosting model was tuned by optimizing the hyperparameters below:

- 'n.trees': this is the number of trees, ie "base learners"
- 'shrinkage': a parameter in the range [0, 1] that slows the rate at which the model "learns"

- ‘interaction.depth’: models the interaction between the predictors

I tuned the hyperparameters by first creating a range of possible values for each hyperparameter. I tested shrinkage in the range [0.05, 0.20], interaction.depth in the range [1, 5], and did not put a limit on the number of trees. I then built a matrix containing all combinations of hyperparameters, and trained models using all combinations of hyperparameters. For each model, I obtained the loss using cross-validation, and selected the hyperparameters that minimized the cross-validation loss.

| shrinkage | interaction.depth | optimal_trees | min_loss |
|---|---|---|---|
| 0.05 | 3 | 853 | 0.5899293 |
| 0.10 | 3 | 456 | 0.5902036 |
| 0.10 | 5 | 249 | 0.5905598 |
| 0.20 | 3 | 248 | 0.5913094 |
| 0.20 | 5 | 92 | 0.5919366 |
| 0.05 | 1 | 870 | 0.6046635 |
| 0.10 | 1 | 546 | 0.6046678 |
| 0.20 | 1 | 320 | 0.6046870 |

*Boosting model hyperparameter tuning*

As can be seen in the table above, the boosting model is optimized for the following hyperparameters:

- ‘n.trees’ = 543
- ‘shrinkage’ = 0.05
- ‘interaction.depth’ = 5

## Factor Importance

To evaluate and quantify the factors that most impact the odds of painkiller misuse, I performed the following steps:

1. Build generic logistic regression model using all factors
2. Obtain the 90% confidence interval for all model coefficients
3. Sort by absolute value of the coefficients (in decreasing order)
4. Apply the exponential function to all coefficients
5. Determine the top 5 most influential factors, and calculate their impact to the odds of painkiller misuse

The logistic regression model provides the regression equation for the log-odds (logit) of the response (painkiller misuse). To determine each factor’s impact on the odds of painkiller misuse, we must obtain the coefficient for each factor. We then calculate the exponential of each coefficient to determine the increase/decrease in odds of painkiller misuse for a unit increase in the value of that factor.

# Analysis and Results

## Model Performance

To evaluate the performance of each model, I generated predicted response values for each model using the testing dataset. Recall from the previous section that the testing set is composed of 25% of the full dataset. The models were evaluated using the following metrics:

- Error: prediction error rate, evaluated on testing set
- Accuracy: prediction accuracy, evaluated on testing set
- Sensitivity: true positive rate (TP / TP+FN)
- Specificity: true negative rate (TN / TN+FP)
- Precision: TP / TP+FP
- P-value: McNemar's p-value

| | Error | Accuracy | Sensitivity | Specificity | Precision | pvalue |
|---|---|---|---|---|---|---|
| **KNN** | 0.0087 | 0.9913 | 0.9497 | 0.9962 | 0.9676 | 1.682351e-05 |
| **Boosting** | 0.0928 | 0.9072 | 0.2631 | 0.9839 | 0.6604 | 2.200000e-16 |
| **Logistic Regression** | 0.0939 | 0.9061 | 0.2442 | 0.9849 | 0.6587 | 0.000000e+00 |
| **Random Forest** | 0.0940 | 0.9061 | 0.2227 | 0.9873 | 0.6587 | 2.200000e-16 |
| **LDA** | 0.0992 | 0.9008 | 0.3472 | 0.9667 | 0.5539 | 6.822641e-149 |
| **QDA** | 0.1345 | 0.8655 | 0.4631 | 0.9134 | 0.3891 | 5.363345e-30 |
| **Naive Bayes** | 0.1369 | 0.8631 | 0.4830 | 0.9084 | 0.3857 | 1.331960e-50 |

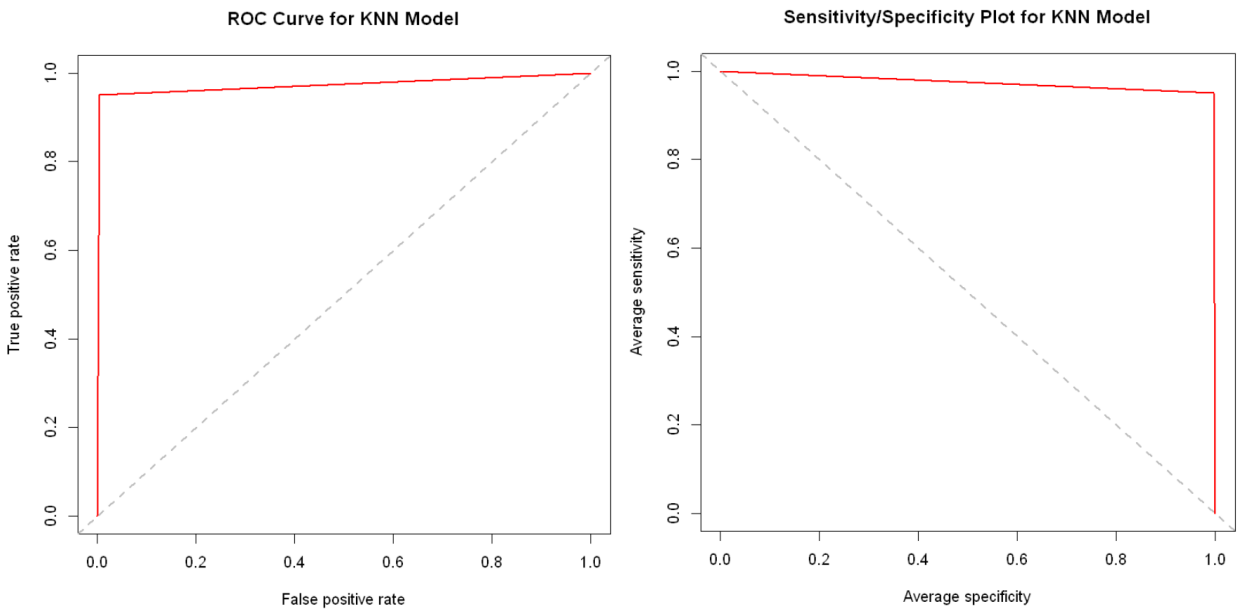
*Results summary for all models*

The model results can be found above. All models were significant at a significance level of 0.01, because they all had p-values of < 0.01. KNN was clearly the best performing model because it performed the best on all metrics, with an accuracy of 99.13%. The confusion matrix for the KNN is shown below.

| | Reference | |
|---|---|---|
| Prediction | 0 | 1 |
| 0 | 37905 | 228 |
| 1 | 144 | 4302 |

*Confusion Matrix for KNN model*

The KNN model correctly predicted 4302 out of 4530 cases of painkiller misuse in the testing set, and correctly predicted 37905 out of 38049 cases of no painkiller misuse. I also obtained the ROC curve and sensitivity/specificity plot for the KNN model:




Better model performance is indicated by how close the ROC curve is to the top left corner of the plot, and how close the sensitivity/specificity curve is to the top right corner of the plot. The ROC Curve and Sensitivity/Specificity plots for the KNN model are nearly perfect.

## Factor Importance

Factor importance was determined using the steps discussed in the "Methodology" section of this report. In the table below, you may find the 90% confidence interval of impact on odds of painkiller misuse, for the five most influential factors.

| Attribute | Min | Max |
|---|---|---|
| Age 50+ | -89% | -92% |
| Unmarried | -35% | -48% |
| Halucinogen Use | 58% | 64% |
| Amphetamin Use | 47% | 56% |
| Ever used Heroin | 45% | 102% |

*90% confidence interval of factor impact on odds of painkiller misuse*

The five factors that most impact odds of painkiller misuse are summarized below:

1. Age
    a. We expect an 89-92% decrease in odds of painkiller misuse if a person's age is >= 50 (versus baseline of age 12-17), holding all other factors constant.
2. Marital Status
    a. We expect a 35-48% decrease in odds of painkiller misuse if a person is unmarried (versus baseline of divorced), holding all other factors constant.
3. Hallucinogen Use

    a. We expect a 58-64% increase in odds of painkiller misuse for each unit increase in hallucinogen use on the Likert scale, holding all other factors constant.
4. Amphetamine Use
    a. We expect a 47-56% increase in odds of painkiller misuse for each unit increase in amphetamine use on the Likert scale, holding all other factors constant.
5. Ever Used Heroin
    a. We expect a 45-102% increase in odds of painkiller misuse if a person has ever used heroin, holding all other factors constant.

# Conclusions

## Models and Factors

| | Error | Accuracy | Sensitivity | Specificity | Precision | pvalue |
|---|---|---|---|---|---|---|
| **KNN** | 0.0087 | 0.9913 | 0.9497 | 0.9962 | 0.9676 | 1.682351e-05 |
| **Boosting** | 0.0928 | 0.9072 | 0.2631 | 0.9839 | 0.6604 | 2.200000e-16 |
| **Logistic Regression** | 0.0939 | 0.9061 | 0.2442 | 0.9849 | 0.6587 | 0.000000e+00 |
| **Random Forest** | 0.0940 | 0.9061 | 0.2227 | 0.9873 | 0.6587 | 2.200000e-16 |
| **LDA** | 0.0992 | 0.9008 | 0.3472 | 0.9667 | 0.5539 | 6.822641e-149 |
| **QDA** | 0.1345 | 0.8655 | 0.4631 | 0.9134 | 0.3891 | 5.363345e-30 |
| **Naive Bayes** | 0.1369 | 0.8631 | 0.4830 | 0.9084 | 0.3857 | 1.331960e-50 |

*Results summary for all models*

The best performing model was the KNN model, which obtained an accuracy of 99.13%. The optimal k-value was determined by evaluating the cross-validation error for many values of k. This model should be used to predict painkiller misuse in individuals so that healthcare professionals can take the necessary steps to avoid dangerous outcomes.

The KNN model was clearly the best performing model because it performed best on all metrics. However, selecting the second and third best models becomes more interesting, because accuracy / total error is not necessarily the most important metric to use for evaluation. The objective of the model is to predict whether an individual will misuse painkillers, which is very dangerous for the individual and can result in overdose or death. For this reason, I feel that sensitivity is the most important metric, because this is the "true positive rate", meaning the percentage of cases of painkiller misuse that were correctly classified. It is more dangerous for the model to miss a case of painkiller misuse (type II error) than it is for the model to incorrectly predict painkiller misuse (type I error). As such, I would rank the best performing models in order of sensitivity:

1. **KNN (sensitivity 94.97%)**
2. Naïve Baes (sensitivity 48.30%)
3. QDA (sensitivity 46.31%)

In addition to evaluating model performance, I also evaluated factor importance to determine the factors that had the greatest impact on the odds of painkiller misuse. The five factors that most impact odds of painkiller misuse are summarized below:

1. Age
    a. We expect an 89-92% decrease in odds of painkiller misuse if a person's age is >= 50 (versus baseline of age 12-17), holding all other factors constant.
2. Marital Status
    a. We expect a 35-48% decrease in odds of painkiller misuse if a person is unmarried (versus baseline of divorced), holding all other factors constant.
3. Hallucinogen Use
    a. We expect a 58-64% increase in odds of painkiller misuse for each unit increase in hallucinogen use on the Likert scale, holding all other factors constant.
4. Amphetamine Use
    a. We expect a 47-56% increase in odds of painkiller misuse for each unit increase in amphetamine use on the Likert scale, holding all other factors constant.
5. Ever Used Heroin
    a. We expect a 45-102% increase in odds of painkiller misuse if a person has ever used heroin, holding all other factors constant.

These factors can be used as critical predictors for healthcare professionals to evaluate the risk of prescribing pain medication to patients.

## Proposed Policy

Based upon the results of this project, I feel that the potential policy actions below would reduce painkiller misuse in the US:

1. Educate students on the dangers of painkiller misuse, beginning in elementary school. Continue education throughout high school.
    a. Painkiller misuse is more likely in younger age categories, so education must start early.
2. Incentive programs to reduce use of all illicit drugs.
    a. Use of illicit drugs is a clear risk factor for painkiller misuse. Offer incentive programs to encourage drug users to reduce or eliminate illicit drug use.
3. Offer marriage counseling to reduce divorce rate in the US.
    a. Divorce is a clear risk factor for painkiller misuse. Cultivating healthy marital relationships would reduce risk of painkiller misuse.
4. Regular screening of patients.
    a. Healthcare providers should request patients to complete the survey used to obtain the dataset for this project. Survey results can be entered into the KNN model to predict whether the patient will misuse painkillers. If the model predicts painkiller misuse, healthcare providers may decide against painkiller prescription, or further educate the patient on responsible painkiller use.

## Lessons Learned

The key ISYE 7406 course concepts utilized throughout this project were:

- Understanding the theory behind several classification models including Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naïve Bayes, K-Nearest Neighbors (KNN), Random Forest, and Boosting.
- Methods for training, optimizing, and evaluating model performance.
- Evaluating factors to determine their contribution to the probability/odds of a classification outcome.

This course provided education on a wide variety of modeling concepts in both regression and classification. My project focused on classification to solve a real-world problem. I utilized most of the classification models covered in this course to compare and contrast the performance of each model on a real-world dataset. In particular, I felt this course provided excellent education on methods of model evaluation and optimization.

*Pros of the Course*

What I enjoyed about this course was the wide variety of models that were covered, and the deep level of detail provided to explain the theory / concepts behind each model. I felt that the R code examples provided in lectures were the most useful tool to understand the concepts and implementation / evaluation of the various models covered. I also enjoyed the structure of the course, in terms of having five homework assignments, a course project, and one final exam. I feel that the best way to learn the material is by implementing it in the homework and project, rather than being tested in multiple exams throughout the semester. Another aspect of the course that I found beneficial was the quick responses and quality of responses provided by TAs in the Piazza forum.

*Cons of the Course*

While this course covered many models in detail, I would like to see more focus on variable selection and evaluation of variable contribution. I felt the course provided excellent detail on the statistical theory that composed the various models, but was a bit lacking in providing methods for selecting variables to include in the models. Another constructive criticism I have for this course is to provide clearer guidance on homework requirements and grade expectations. On a couple of the homework assignments I received a lower-than-expected grade, but did not fully understand why I was graded poorer than I expected.

## Citations


"Drug Overdose Death Rates." *National Institutes of Health*, U.S. Department of Health and Human Services, 9 Feb. 2023, https://nida.nih.gov/research-topics/trends-statistics/overdosedeathrates#:~:text=Opioid%2Dinvolved%20overdose%20deaths%20rose,with%2080%2C411%20reported%20overdose%20deaths.

"Opioid Data Analysis and Resources." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 1 June 2022, https://www.cdc.gov/opioids/data/analysis-resources.html.

"The High Price of the Opioid Crisis, 2021." *The Pew Charitable Trusts*, The Pew Charitable Trusts, 27 Aug. 2021, https://www.pewtrusts.org/en/research-and-analysis/data-visualizations/2021/the-high-price-of-the-opioid-crisis-2021.

# Appendix

All necessary plots, figures, and output are included within the body of the report. This appendix serves solely to highlight the important R code and functions used.

*R Code – dividing data into training and testing set*

```
# train/test split
set.seed(123)

fractrain = 0.75
fractest = 1-fractrain
flag <- sort(sample(dim(data)[1], dim(data)[1]*fractest, replace = FALSE))
train <- data[-flag,]
test <- data[flag,]
```

*R Code – functions used for model building*

- *Logistic Regression model building using glm() function*

```
# build logistic regression model
glm1 <- glm(PRLMISEVR ~ ., data = train, family="binomial")
```

- *LDA model building using lda() function*

```
# build LDA model
lda1 <- lda(PRLMISEVR ~ ., data = train)
```

- *QDA model building using qda() function*

```
# build QDA model
qda1 <- qda(PRLMISEVR ~
            HEROINUSE+TRQLZRS+COCAINE+SEDATVS+AMPHETMN+HALUCNG+TRTMENT+MHTRTMT+
            YEAR+AGECAT+SEX+MARRIED+EMPLOY18+HEALTH+MENTHLTH+HEROINEVR,
            data = train, cv=5)
```

- *Naïve Bayes model building using naiveBayes() function*

```
# build Naive Bayes model
bayes1 <- naiveBayes(PRLMISEVR ~ ., data = train)
```

- *KNN model building and optimization*

```
# create knn models for several k-values
k.error = c()
k.vals = c(2,4,6,8,10,12,14)

for (i in k.vals) {
    knn.model <- knn(train = train2, test = test2,
                     cl = train2$PRLMISEVR, k=i)
    k.error <- append(k.error, mean(knn.model != test2$PRLMISEVR))
}

# plot error vs k-value
plot(k.vals, k.error,
    main="KNN: error vs k-value",
    xlab="k value",
    ylab="Error")
lines(k.vals[order(k.vals)], k.error[order(k.vals)],
      xlim=range(k.vals), ylim=range(k.error), pch=16)

# build knn model with optimal k-value
knn1 <- knn(train = train2, test = test2, cl = train2$PRLMISEVR, k=4)
```

- *Random Forest model building and optimization*

```
# build random forest model with default parameters
rf1 <- randomForest(PRLMISEVR ~ ., data = train, importance=TRUE)

# plot error rate vs number of trees
plot(rf1, ylim=c(0,0.10), main="rf1 Error versus ntrees")

# tune the mtry parameter
try <- tuneRF(x=train[,-10], y=train[,10],
              stepFactor = 1.5,
              plot = TRUE,
              ntree = 100,
              trace = TRUE,
              improve = 1e-5)

# get best mtry
best.mtry <- try[try[, 2] == min(try[, 2]), 1]

# tune nodesize parameter
nodes <- tune(randomForest, PRLMISEVR ~ ., data = train,
              ranges = list(nodesize = 2:10),
              tunecontrol = tune.control(sampling = "fix"))

# get best nodesize
best.nodesize <- nodes$best.parameters[1,]

# create new random forest model with tuned ntrees, mtry, nodesize
rf2 <- randomForest(PRLMISEVR ~ ., data = train, importance=TRUE, ntree=300,
                    mtry=best.mtry, nodesize=best.nodesize)
```

- *Boosting model building and optimization*

```
# build default gbm model
gbm1 <- gbm(PRLMISEVR ~ ., data=train, distribution = 'bernoulli',
            n.trees=2000,
            cv.folds = 3)

# create parameter grid to find best shrinkage and interaction depth
hyper_grid <- expand.grid(
  shrinkage = c(.05, .1, .2),
  interaction.depth = c(1, 3, 5),
  optimal_trees = 0,
  min_loss = 0
)

# grid search
for(i in 1:nrow(hyper_grid)) {
    set.seed(123)

    # train model
    gbm.tune <- gbm(PRLMISEVR ~ ., data=train,
                    distribution = "bernoulli",
                    n.trees = 2000,
                    shrinkage = hyper_grid$shrinkage[i],
                    interaction.depth = hyper_grid$interaction.depth[i],
                    train.fraction = .75,
                    cv.folds=2,
                    n.cores = NULL, # will use all cores by default
                    verbose = FALSE
                   )

    # add min training error and trees to grid
    hyper_grid$optimal_trees[i] <- which.min(gbm.tune$cv.error)
    hyper_grid$min_loss[i] <- min(gbm.tune$cv.error)
}

# arrange in order of best performance
hyper_grid <- hyper_grid %>% arrange(min_loss) %>% head(10)

# build optimized gbm model
gbm2 <- gbm(PRLMISEVR ~ ., data=train, distribution = 'bernoulli',
            n.trees=hyper_grid$optimal_trees[1],
            shrinkage=hyper_grid$shrinkage[1],
            interaction.depth=hyper_grid$interaction.depth[1],
            cv.folds = 3)
```

*R Code – Factor Importance*

```
# build logistic regression model to evaluate feature importance
glm1 <- glm(PRLMISEVR ~ ., data = train, family="binomial")

# get confidence interval of model coefficients
coeffs <- confint(glm1, level=0.9)

# order by most important coefficients
coeffs.sorted <- coeffs[order(abs(coeffs[,1]), decreasing=TRUE),]

# calculate change in odds of drug misuse by taking exponent of coefficients
agecat50.odds <- round(exp(coeffs.sorted[1,]), 2)
unmarried.odds <- round(exp(coeffs.sorted[5,]), 2)
halucng.odds <- round(exp(coeffs.sorted[6,]), 2)
amphetmn.odds <- round(exp(coeffs.sorted[7,]), 2)
heroinevr.odds <- round(exp(coeffs.sorted[8,]), 2)
```

*R Code – obtain performance metrics (example using Logistic Regression model)*

```
# get predictions
pred.glm1 <- predict(glm1, test, type="response")
pred.glm1 <- replace(pred.glm1, pred.glm1>=0.5, 1)
pred.glm1 <- replace(pred.glm1, pred.glm1<0.5, 0)

# output confusion matrix and error rate for glm1
pred.glm1 <- as.factor(pred.glm1)
confusionMatrix(pred.glm1, test$PRLMISEVR)
glm1.error = mean(pred.glm1 != test$PRLMISEVR)

# other metrics
glm.accuracy=confusionMatrix(pred.glm1, test$PRLMISEVR)$overall[[1]]
glm.sensitivity=confusionMatrix(pred.glm1, test$PRLMISEVR)$byClass[[2]]
glm.specificity=confusionMatrix(pred.glm1, test$PRLMISEVR)$byClass[[1]]
glm.precision=confusionMatrix(pred.glm1, test$PRLMISEVR)$byClass[[4]]
glm.pvalue=confusionMatrix(pred.glm1, test$PRLMISEVR)$overall[[7]]
```