Stats 151A Final Project

Suchir Joshi, Yanze (Spencer) Qu, Zhendong (Eli) Xie

12/17/2020

Introduction

In this project, our goal is to to accurately predict if a person has heart disease using a regression model with demographic information such as age and medical tests results such as serum cholesterol level. We hope our model can be used as a pre-testing tool for doctors to decide if further medical testing is necessary.

Throughout our analysis, we face two major challenges. The first one is to control the false negative rate (fnr) because telling a patient no further test is needed when he/she has heart disease can lead to fatal consequences and is therefore much more costly than false positives. Secondly, we need to limit to size of our model to ensure our model is realistic and interpretable for users such as doctors/patients. If our model is too large or complex, doctors would not be be able verify the model with their domain expertise and therefore the model would not trusted. However, in general, both controlling fnr and reducing model size can make it harder for the model to achieve high accuracy. Therefore, it is critical that we design our model structure and select model features intelligently to deal with these two challenges while moving toward our goal to make accurate predictions.

Data Description

The dataset we use was from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V attributed to four doctors: Andras Janosi, M.D (Hungarian Institute of Cardiology. Budapest co), William Steinbrunn, M.D (University Hospital, Zurich, Switzerland), Matthias Pfisterer, M.D (University Hospital, Basel, Switzerlan) and Robert Detrano, M.D. (V.A. Medical Center, Long Beach and Cleveland Clinic Foundation). We chose this dataset because it has both demographics information (age and sex) and medical test results (chest pain type, serum cholesterol level etc.).

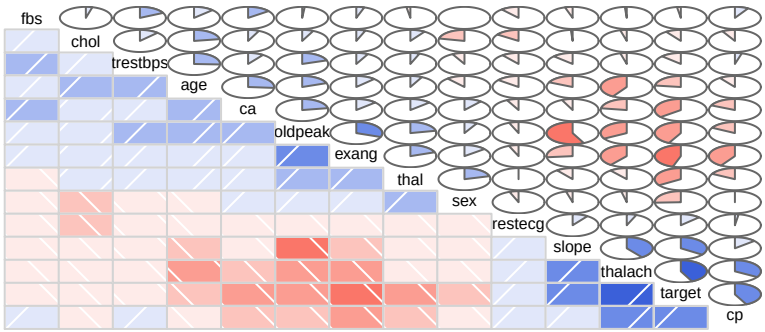
Here is summary of the 14 attributes in the dataset as follows: 1. age:age in years 2. sex: (1 for male, 0 for female) 3. cp: chest pain type (4 values) 4. trestbps: resting blood pressure (in mm Hg on admission to the hospital) 5. chol: serum cholesterol (in mg/dl) 6. fbs: fasting blood sugar $>$ 120 mg/dl (1 = true; 0 = false) 7. restecg: resting electrocardiographic results (values 0,1,2) 8. thalach: maximum heart rate achieved 9. exang: exercise induced angina(1 = yes; 0 = no) 10. oldpeak: ST depression induced by exercise relative to rest 11. slope: the slope of the peak exercise ST segment(values 0,1,2) 12. ca: number of major vessels (0-3) colored by flourosopy 13. thal: 0 = normal; 1 = fixed defect; 2 = reversable defect 14: Target (response): 0=no heart disease, 1=has heart disease

In this dataset, our response variable is target. Prior to performing our data analysis, we randomly selected 90% of the data as our training set and 10% of the data as our validation set. The rest of 10% data will be our test set, which will be used in the end to assess our model's predictive prowess.

EDA

First we will perform some basic EDAs using our training set. We only included a corrolgram in our report because it provided the most information for our project while most of other plots are not very informative.

Heart Disease Data Corrgram



From the plot, we see the features with the highest correlations with target are oldpeak, ca, exang, slope and thalach. And we also observe there are some highly colinear variables such as oldpeak and slope, However, since our research question focuses on prediction, we will not explicitly deal with this problem in this report.

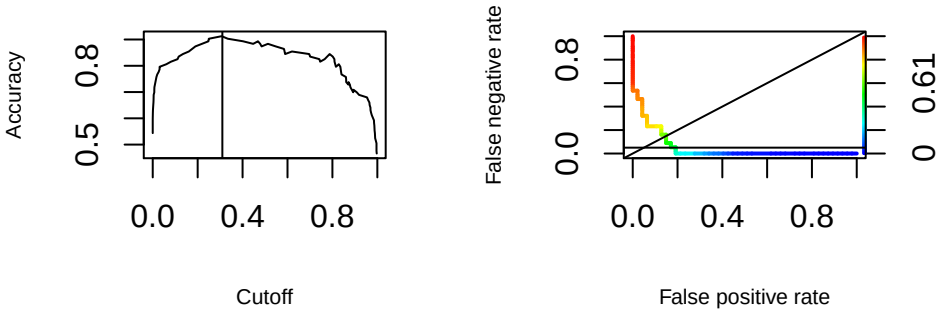
Methodology

For this report, we decided to use a logistic regression model and to use forward BIC method to select the most relevant explanatory variables and the interaction terms between two explanatory variables with training set for our final regression model. Then, to find the optimal decision boundary, we used our final regression model to predict the validation set and drew a fpr vs fnr curve using the predictions. The final decision boundary was chosen so that false positive rate (fpr) is minimized while keeping fnr under 5%, which we chose to be our target fnr.

We chose a logistic regression model because our response variable is binary and we chose BIC because it finds the model that maximizes the likelihood of the data (which is closely associated with higher accuracy) while penalizing large models, Therefore it is ideal to deal with the second challenge to limit the model size. We chose forward method because it would be too computationally expensive to perform exhaustive search. Even though forward method doesn't guarantee that we would find the best model, it still finds a model with high performance in general and hence we believe it serves as a valid alternative to exhaustive search. The use of fpr vs fnr plot to find decision boundary is to deal with first challenge of controlling fnr. The reasons for our choice to use 5% as fnr cutoff and to use only interactions between two explanatory variables will be discussed in the discussion section.

Using BIC, we decided that the most helpful features for our final model are exang, oldpeak, cp, ca, thal, sex, thalach, restecg, ca:thal, oldpeak:ca, cp:ca, exang:sex, exang:oldpeak, exang:cp, ca:restecg, oldpeak:sex, cp:thalach (17 in total). Many of the features selected by our model also appear to have a high correlation with target in the corrolgram during EDA.

Then to decide on the optimal decision boundary, we used our regression model to predict the validation set and drew a tpr vs tnr curve using the predictions:



By observing the tpr vs tnr curve, we found that the best fpr we can achieve while controlling fnr to be under 5% is around 19% when the cutoff is 0.31. The cutoff also happens to be the decision boundary that maximizes the overall accuracy.

Hence we decided that our final model would be a logistical regression model with exang, oldpeak, cp, ca, thal, sex, thalach, restecg, ca:thal, oldpeak:ca, cp:ca, exang:sex, exang:oldpeak, exang:cp, ca:restecg, oldpeak:sex and cp:thalach as our features and 0.31 as the decision boundary. We included a summary of the model in the additional work section.

Finally we will assess the performance of our model using the test set and the our test set accuracy is 84.31%, fpr is 28.85% and fnr is 2%.

Discussion

Overall our model seems to be performing reasonable well with an overal accuracy around 85% and false negative rate (fnr) under 2% on the test sets. The false positive rate (fpr) is relatively high at around 30%, but it is to be expected because our final threshold was chosen to be biased for fnr. However, one can argue that 2% fnr is still way too high for our purpose and even more so for 5%, which we chose to be our target fnr for the model. We admit that our choice is not very well justified but without further guidance from medical professionals, choosing a relatively low control rate seems to be the best available option.

During feature engineering, we only consider interaction between 2 explanatory variables because we realized using interaction between 3 or more variables would lead the model to overfit the training data. The model achieves a very high test accuracy (~97%) but the test fnr would stay at 6% even with a decision boundary as small as 0.01, which we believe indicates the model overfitted the training data and wasn't able to distinguish a portion of positive cases in test set.

Since our primary focus is on prediction, we will discuss the diagnostic plots and model interpretations in the addition work section.

Going forward, there are several things we can do to improve our current model. First, we can consult medical professionals to gather feedback on our choice of specific parameters (e.g. target fnr at 5%) and the field-specific criteria/statistical tests used to evaluate models in medical research. In addition, it would be helpful to assess if there's any potential bias in our data set (e.g. are people of color not well represented) by gathering more demographic information and adjust our model accordingly. Last but not the least, we can use ANOVA test to detect if colinear terms are present in our model to improve the stability of our model for better interpretability.

Conclusion

In conclusion, we believe that our model design tackles the challenges to control fnr and model size well while maintaining a relatively high overall accuracy. We recommend that this model should be used by medical professionals in conjunction with their medical expertise as an indication if more comprehensive tests are needed for people with potentially heart diseases and doctors should refer to full model interpretation part in the additional work section for guidance on how to interpret and use the model.

Reference

[1] Source of data: https://www.kaggle.com/johnsmith88/heart-disease-dataset

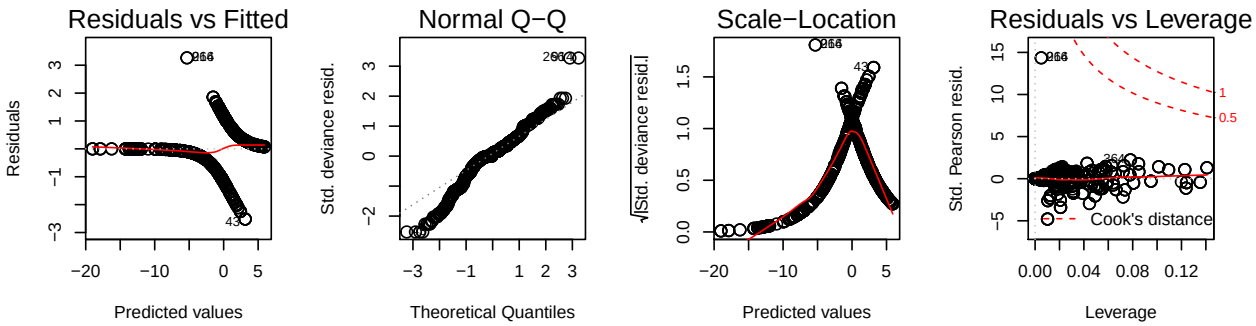
Addtional work

Further details on our final model

Summary of our final model

```
##
## Call:
## glm(formula = final.formula, family = "binomial", data = train_dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5173  -0.2918   0.0870   0.4736   3.2643
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.753850   1.426400   3.333 0.000860 ***
## exang         -2.486205   0.580429  -4.283 1.84e-05 ***
## oldpeak        1.163450   0.449218   2.590 0.009599 **
## cp            -2.228274   0.858053  -2.597 0.009407 **
## ca            -5.239450   0.737072  -7.108 1.17e-12 ***
## thal          -1.557915   0.236379  -6.591 4.38e-11 ***
## sex           -1.804000   0.431441  -4.181 2.90e-05 ***
## thalach        0.008801   0.007673   1.147 0.251354
## restecg       -0.089687   0.278527  -0.322 0.747449
## ca:thal        1.340260   0.258068   5.193 2.06e-07 ***
## oldpeak:ca    -0.794980   0.185670  -4.282 1.85e-05 ***
## cp:ca          0.587989   0.155433   3.783 0.000155 ***
## exang:sex      2.543009   0.675166   3.766 0.000166 ***
## exang:oldpeak -1.905454   0.386047  -4.936 7.98e-07 ***
## exang:cp       1.471858   0.338693   4.346 1.39e-05 ***
## ca:restecg     1.488021   0.305354   4.873 1.10e-06 ***
## oldpeak:sex   -1.430109   0.427668  -3.344 0.000826 ***
## cp:thalach     0.016556   0.005536   2.991 0.002783 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1136.27  on 819  degrees of freedom
## Residual deviance:  505.93  on 802  degrees of freedom
## AIC: 541.93
##
## Number of Fisher Scoring iterations: 7
```

Diagnosis plots



In residuals vs fitted, we observe the average roughly line up with y=0, though we do observe some large negative residuals which is probably a result of our choice to biased the model towards reducing false negatives.The QQ plot follows a straight line for the most part but the trend disappeared towards the left end where the negative residuals are, probably for the same reason. In scale vs location, we observe some large residuals values as expected after seeing the two previous plots. Interstingly, in all three plots, we observe several points with a very large postive residuals and in the residuals vs leverage, these points are shown to have very large leverages. We believe these points represent patients who don't have heart disease but are very different from other healthy patients as measured by the features in our model. These data points are probably the cause of the overfitting of three-interaction-term model. Therefore, it would be interesting to explore these influential points in our future analysis.

Full model interpretation

The mathematical function of our logistic regression model is:

$$\pi_i = \frac{1}{1+E}.$$

$$\begin{aligned} where: E = \exp(-(&\alpha_{intercept} + \beta_1 x_{exang} + \beta_2 x_{oldpeak} + \beta_3 x_{cp} + \beta_4 x_{ca} + \beta_5 x_{thal} \\ &+ \beta_6 x_{sex} + \beta_7 x_{thalach} + \beta_8 x_{restecg} + \beta_9 x_{ca:thal} + \beta_{10} x_{oldpeak:ca} \\ &+ \beta_{11} x_{cp:ca} + \beta_{12} x_{exang:sex} + \beta_{13} x_{exang:oldpeak} + \beta_{14} x_{exang:cp} \\ &+ \beta_{15} x_{ca:restecg + \beta_{16} x_{oldpeak:sex} + \beta_{17} x_{cp:thalach}})) \end{aligned}$$

The left side of our equation, π_i, represent the output prediction of probability that people have heart disease, and the right side we have the expression using our model coefficients and the corresponding explanatory variables. The final logistic result will be determined after comparing the output prediction probability to the cutoff we choose. Here is some examples of how to interpret each coefficient: Same equation after transmition:

$$\frac{\pi_i}{1-\pi_i} = \frac{1}{E}$$

As for β_1 with corresponding variable exang, from the model we know that the coefficient is around -2.486, which means that with one unit increase of the variable exang while other variables staying the same, the odds $\frac{\pi_i}{1-\pi_i}$ decreases by a factor exp(-2.486), which is 0.083. As for $\beta_1, \beta_{12}, \beta_{13}, \beta_{14}$ with corresponding variables exang, exang∗sex, exang∗oldpeak and exang∗cp, from the model we know that the coefficients are around -2.486, 2.543, -1.905 and 1.471. In order to interpret these coefficients, for a patient with exang = 0, sex = 1, oldpeak = 0 and cp = 1, with other variables staying the same, with one unit increase of the variable exang, the odds $\frac{\pi_i}{1-\pi_i}$ changes by exp(-2.486 + 2.543 + 1.471) = exp(1.528), which is increasing by a factor approximately 4.609.

Figure 1: Model interpretation

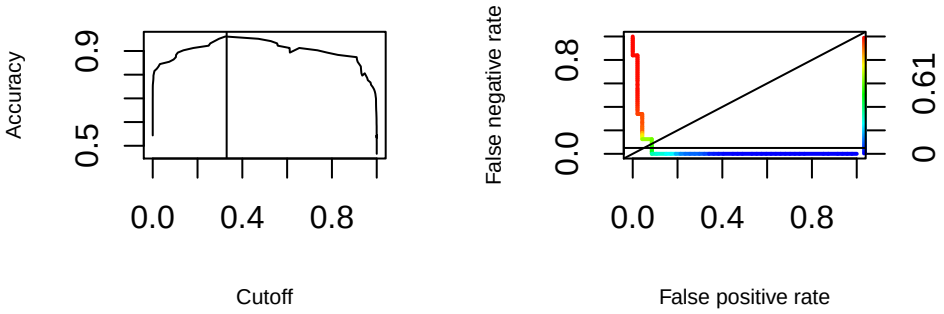
It should be noted that we observed many higly coolinear terms in the corrolgram which we didn't explicitly deal with in this report. Hence individual coefficient values should not be used separately and the result from this report should not be used for the purpose of causal inference.

Other methods attempted

Besides BIC, we attempted five other methods to perform variable selection including forward AIC, R-squared, Mallow's CP, LASSO regularization and a custom method we designed. Besides regression models, we also fitted a 10-layer decision tree. For all the models mentioned above, we used the same method to determine the optimal decision boundary as our final model: namely we use a fpr vs fnr plot and find the boundary that would minimize fpr while keeping the fnr under 5%. However, it should be noted that since LASSO and our custom method rely on cross validation, we only split the dataset intro training (90%) and test data (10% and no validation set). Therefore, the fpr vs fnr plot was constructed using the fitted values on the training set instead of the predictions of the validation set. However, we keep the same random seed to ensure our comparison between different models are valid and fair.

Mallow's CP

We attempted Mallow's CP because Mallow's CP minimizes the model's residual deviance (which is equvalient with maximizing the likelihood of the data in most cases and is closely associated with high accuracy) and therefore seems a valid criteria to select the most helpful features.

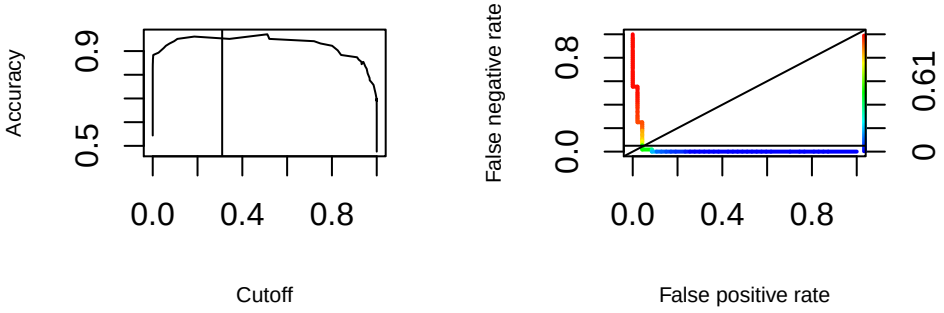


For Mallow's CP, the decision boundary that maximizes te overall accuracy (0.33) also keeps the fnr below 5%.

When tested against the test set, the model achieves an amazing overall accuracy of 92.16%, fpr 13.46% and fnr 2%. Although the acccuracy and fpr were signficantly improved without increasing the fnr compared to BIC, the Mallow's CP selected 46 features and given such a huge model has no interpretability in the medical practice, we decided not to use it even with its superior performance.

Adjusted R-squared

We attempted adjusted R-squared because adjusted R-squared directly minimizes the model's residual deviance (which is equvalient with maximizing the likelihood of the data in most cases and is closely associated with high accuracy) while penalizing the larger models and therefore seems a valid criteria for feature selection.

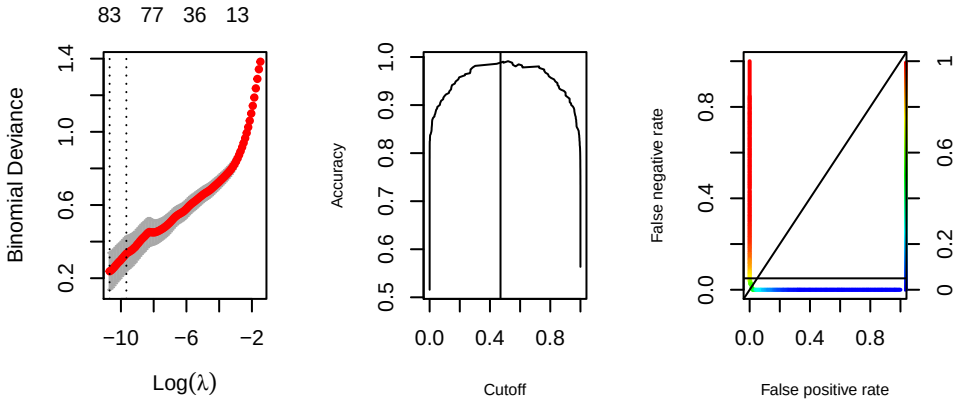


For adjusted R-squared, the decision boundary that maximizes te overall accuracy (0.51) also keeps the fnr below 5%.

When tested against the test set, the model also achieved an amazing overall accuracy of 93.14% and fpr 5.78%, but the fnr is very high at 8%. which is higher than our control rate. Therefore, even though adjusted R-squared model has a reasonable size (only 18 features in total) and higher accuracy, we decided to not use it because of the high fnr.

LASSO

We also attempted to fit a LASSO regularized logistic model. LASSO can be a valid method because it does feature selection implicitly by minimizing the residual deviance (which is closely associated with high accuracy) while penalizing large coefficients and forcing many small coefficients to 0. To find the best beta (regularization parameter), we used 10-fold cross-validation. Cross validation is a valid method to find beta because it finds the beta that minimizes the overall residuan deviance.

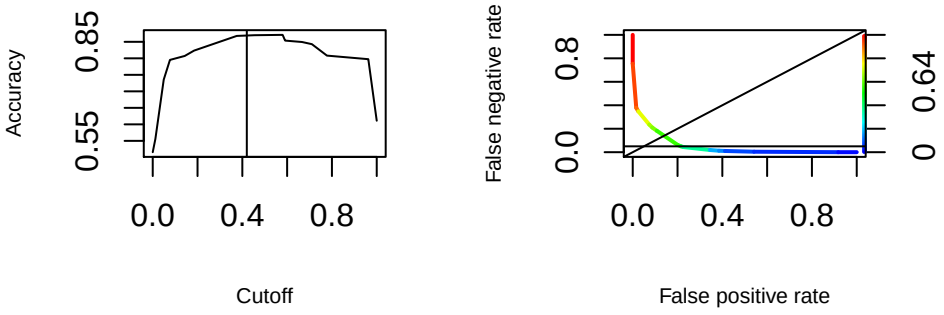


For LASSO, the decision boundary that maximizes te overall accuracy (0.47) also keeps the fnr below 5%.

When tested against the test set, the model achieved the best peformance acrosss all the models we attempted with an overall accuracy of 96.1%, fpr of 5.77% and the fnr of 2%. In addition to outperforming our final model in all metrics, the decisioon boundary is very close to 0.5, which means the model was able distinguish between positive cases and negative cases without relying on the bias towards fnr we introduced by lowering the decision boudary in other models. The diagnostic plots also reflected that as we are no long seeing the skewnessed in residuals in our final model. However, the LASSO has a fatal flaw: it uses 84 out 93 features available and LASSO coefficients don't have an intuitive interpretation. Moreover, having such a large model made us worry that the model is overfitting for this specific dataset. It is possible that the dataset itself is biased (e.g. most of white males/females) even though we couldn't confirm this without further information. As such, we arrived at the conclusion that we should keep BIC as our final method.

Decision Tree

In addition to the logitistic regression, we also fitted 10-layer decision tree. Decision tree provides an intuitive interpretation, which is a huge advantage for our purpose, and we limit our tree to 10 layers to avoid overfitting.



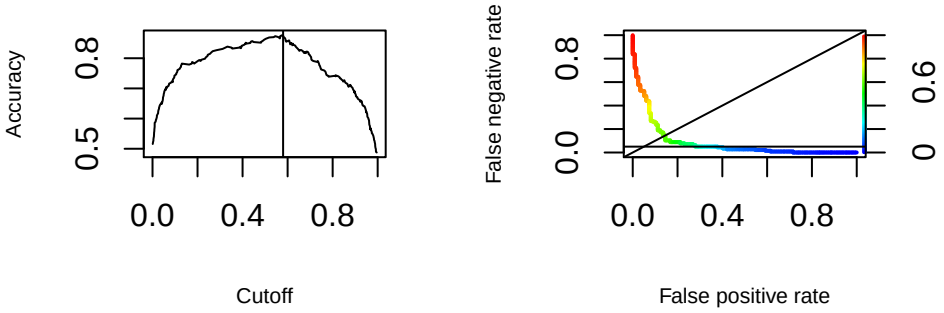
	Num features	Threshold	Test accuracy	Test FPR	Test FNR
AIC	92	0.01	0.971	0	0.06
BIC	17	0.31	0.843	0.288	0.02
Mallow's CP	48	0.33	0.922	0.135	0.02
Adjusted R-squared	18	0.31	0.902	0.115	0.08
Lasso	84	0.47	0.961	0.058	0.02
Decision Tree	maxdepth=10	0.37	0.863	0.212	0.06
Custom	13	0.21	0.833	0.308	0.02

For the 10-layer decision tree, the decision boundary that maximizes te overall accuracy (0.37) also keeps the fnr below 5%.

When tested against the test set, the model achieved a slightly better overall accuracy of 86.27% and fpr 21.15%, but the fnr is very high at 6%. which is higher than our control rate. Therefore, even though the decision tree has the best interpretability and higher overall accuracy, we choose our final model over it because of the high fnr.

Custom Method

In addition to the traditional methods, we also attempted build a custom method for our project. We manually build a cross-validation function that uses accuracy or fpr + fnr as the validation metric. We then selected the most helpful individual feature based on the cross-validation accuracy. To determine which interaction term to include, we design a custom algorithm to calculate the error metric: the error metric we use for interactions is fpr + fnr if fnr is below our desired threshold (5%) and fpr + 100fnr if fnr is above the threshold. Essentially, we penalizes fnr heavily if it is above the threshold. For example, if after adding a certain feature, the current fpr decrease by 0.2 but the fnr increases by 0.01 to become 0.61, the feature would not be added because the over gain (0.2 - 0.01*100) is negative. The choice of 100 is an arbitrary decision that seems to give us a reasonable performance.



For our custom method, the decision boundary that maximizes the overall accuracy (0.58) failed to keep the fnr below 5% so we eventually used 0.21 decrease the fnr below 5%.

When tested against the test set, the model achieved a slightly worse overall accuracy of 83.33% and fpr 30.77%, but the fnr is the same at 2%. However, our custom method used less features (12 in total) and hence the resulting model gives better interpretability. Still, due to its complex design, it is probably less convincing to doctors compared to our original model and BIC seems to have the same effect while being a well-established method. Therefore, we decide to keep BIC forward as our final method.

In addition to the models mentioned above, we also used AIC but we chose not to include the details of the model building process due to page limit. We summarized our findings in the table above.

```
---
title: "\\vspace{-1.5cm}  Stats 151A Final Project"
author: "Suchir Joshi, Yanze (Spencer) Qu, Zhendong (Eli) Xie"
date: "12/17/2020"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
library(lattice)
library(leaps)
library(magrittr)
library(caret)
library(tidyverse)
library(corrgram)
library(ROCR)
library(pracma)
library(party)
# library(car)
library(glmnet)
library(tinytex)
```

Introduction
In this project, our goal is to to accurately predict if a person has heart disease using a regression model with demographic information such as age and medical tests results such as serum cholesterol level. We hope our model can be used as a pre-testing tool for doctors to decide if further medical testing is necessary.

Throughout our analysis, we face two major challenges. The first one is to control the false negative rate (fnr) because telling a patient no further test is needed when he/she has heart disease can lead to fatal consequences and is therefore much more costly than false positives. Secondly, we need to limit to size of our model to ensure our model is realistic and interpretable for users such as doctors/ patients, If our model is too large or complex, doctors would not be be able verify the model with their domain expertise and therefore the model would not trusted. However, in general, both controlling fnr and reducing model size can make it harder for the model to achieve high accuracy. Therefore, it is critical that we design our model structure and select model features intelligently to deal with these two challenging while moving toward our goal to make accurate predictions.

Data Description
The dataset we use was from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V attributed to four doctors: Andras Janosi, M.D (Hungarian Institute of Cardiology. Budapest co), William Steinbrunn, M.D (University Hospital, Zurich, Switzerland), Matthias Pfisterer, M.D (University Hospital, Basel, Switzerlan) and Robert Detrano, M.D. (V.A. Medical Center, Long Beach and Cleveland Clinic Foundation). We chose this dataset because it has both demographics information (age and sex) and medical test results (chest pain type, serum cholesterol level etc.).

Here is summary of the 14 attributes in the dataset as follows:
1. age:age in years
2. sex: (1 for male, 0 for female)
```

```
## Split data into train_val and test
split_dummy.1 <- sample(c(rep(0, 9/10 * nrow(dat)),
                          rep(1, 1/10 * nrow(dat))))


train.val.dat <- dat[split_dummy.1 == 0, ]
test_dat <- dat[split_dummy.1 == 1, ]


## Split train_val into train and validatioin
split_dummy.2 <- sample(c(rep(0, 8/9 * nrow(train.val.dat)),
                          rep(1, 1/9 * nrow(train.val.dat))))


train_dat <- train.val.dat[split_dummy.2 == 0, ]
val_dat <- train.val.dat[split_dummy.2 == 1, ]
```


EDA
First we will perform some basic EDAs using our training set. We only included a corrolgram in our report because it provided the most information while most of other plots are not very informative.
```{r, echo=FALSE, warning=FALSE, include=FALSE}
# pairs(train_dat)
```


```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 5, fig.height = 3}
corrgram(train_dat, order=TRUE, lower.panel=panel.shade,
  upper.panel=panel.pie, text.panel=panel.txt,
  main="Heart Disease Data Corrgram")
```

From the plot, we see the features with the highest correlations with target are oldpeak, ca, exang, slope and thalach. And we also observe there are some highly colinear variables such as oldpeak and slope, However, since our research question focuses on prediction, we will not explicitly deal with this problem in this report.

Methodology
For this report, we decided to use a logistic regression model and to use forward BIC method to select the most relevant explanatory variables and the interaction terms between two explanatory variables with training set for our final regression model. Then, to find the optimal decision boundary, we used our final regression model to predict the validation set and drew a fpr vs fnr curve by using the predictions. The final decision boundary was chosen so that false positive rate (fpr) is minimized while keeping fnr under 5%, which we chose to be our target fnr.

We chose a logistic regression model because our response variable is binary and we chose BIC because it finds the model that maximizes the likelihood of the data (which is closely associated with higher accuracy) while penalizing large models, Therefore it is ideal with the second challenge to limit the model size. We chose forward method because it would be too computationally expensive to perform exhaustive search. Even though forward method doesn't guarantee that we would find the best model, it still finds model with high performance in general and hence we

```
                    scope = list(upper = mod.full,
                                 lower = mod.minimal), k = k_num, trace=0)

### Choosing the final model
final.formula = bic.forward$formula
final.model <- glm(final.formula, data = train_dat, family = "binomial")
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
final.formula
```

Using BIC, we decided that the most helpful features for our final model are exang, oldpeak, cp, ca, thal, sex, thalach, restecg, ca:thal, oldpeak:ca, cp:ca, exang:sex, exang:oldpeak, exang:cp, ca:restecg, oldpeak:sex, cp:thalach (17 in total). Many of the features selected by our model also appear to have a high correlation with target in the corrolgram during EDA.

Then to decide on the optimal decision boundary, we used our regression model to predict the validation set and drew a tpr vs tnr curve using the predictions:

```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Error criteria helper functions
## Calculate false positive rate
cal.fp <- function(pred, observation) {
    num.fp = sum((pred == 1) & (observation == 0))
    return(num.fp / sum(observation == 0))
}

## Calculate false negative rate
cal.fn <- function(pred, observation) {
    num.fn = sum((pred == 0) & (observation == 1))
    return (num.fn / sum(observation == 1))
}

## Calculate accuracy
cal.accuracy <- function(pred, observation) {
    num.acc = sum(pred == observation)
    return (num.acc / length(observation))
}
```

```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 5, fig.height = 2.5}
### Choose the threshold (ROC Curve)
par(mfrow =c(1, 2))

val_pred = predict(final.model, val_dat, type='response')
pred.confusion = ROCR::prediction(val_pred, val_dat$target)
perf.acc = ROCR::performance(pred.confusion, "acc")
plot(perf.acc, cex.lab = 0.7)
```

```
abline(a = 0, b = 1)
abline(a = 0.05, b = 0)
```

By observing the tpr vs tnr curve, we found that the best fpr we can achieve while controlling fnr to be under 5% is around 19% when the cutoff is 0.31. The cutoff also happens to be the decision boundary that maximizes the overall accuracy.

```{r, echo=FALSE, warning=FALSE, include=FALSE}
final.threshold = 0.31
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
final.pred.val = predict(final.model, val_dat, type='response') > final.threshold
cal.accuracy(final.pred.val, val_dat$target)
cal.fp(final.pred.val, val_dat$target)
cal.fn(final.pred.val, val_dat$target)
```

Hence we decided that our final model would be a logistical regression model with exang, oldpeak, cp, ca, thal, sex, thalach, restecg, ca:thal, oldpeak:ca, cp:ca, exang:sex, exang:oldpeak, exang:cp, ca:restecg, oldpeak:sex and cp:thalach as our features and 0.31 as the decision boundary. We included a summary of the model in the additional work section.

```{r, echo=TRUE, warning=FALSE, include=FALSE}
### Test the final model against the test set
final.model.plus.val = glm(final.formula, data = train.val.dat, family =
"binomial")
final.pred.test = predict(final.model, test_dat, type='response') > final.threshold
cal.accuracy(final.pred.test, test_dat$target)
cal.fp(final.pred.test, test_dat$target)
cal.fn(final.pred.test, test_dat$target)
```

Finally we will assess the performance of our model using the test set and the our test set accuracy is 84.31%, fpr is 28.85% and fnr is 2%.

Discussion

Overall our model seems to be performing reasonable well with an overal accuracy around 85% and false negative rate (fnr) under 2% on the test sets. The false positive rate (fpr) is relatively high at around 30%, but it is to be expected because our final threshold was chosen to be biased for fnr. However, one can argue that 2% fnr is still way too high for our purpose and even more so for 5%, which we chose to be our target fnr for the model. We admit that our choice is not very well justified but without further guidance from medical professionals, choosing a relatively low control rate seems to be the best available option.

During feature engineering, we only consider interaction between 2 explanatory variables because we realized using interaction between 3 or more variables would lead the model to overfit the training data. The model achieves a very high test

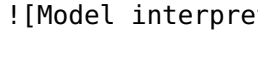
```{r, echo=FALSE, warning=FALSE, include=TRUE}
summary(final.model)
```

Diagnosis plots

```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 8, fig.height = 2.5}
par(mfrow =c(1, 4))

plot(final.model, cex=1.5, cex.title=0.5, cex.label=0.7)
```

In residuals vs fitted, we observe the average roughly line up with y=0, though we do observe some large negative residuals which is probably a result of our choice to biased the model towards reducing false negatives.The QQ plot follows a straight line for the most part but the trend disappeared towards the left end where the negative residuals are, probably for the same reason. In scale vs location, we observe some large residuals values as expected after seeing the two previous plots. Interstingly, in all three plots, we observe several points with a very large postive residuals and in the residuals vs leverage, these points are shown to have very large leverages. We believe these points represent patients who don't have heart disease but are very different from other healthy patients as measured by the features in our model. These data points are probably the cause of the overfitted of three-interaction-term model. Therefore, it would be interesting to explore these influential points in our future analysis.

Full model interpretation


It should be noted that we observed many higly coolinear terms in the corrolgram which we didn't explicitly deal with in this report. Hence individual coefficient values should not be used separately and the result from this report should not be used for the purpose of causal inference.

Other methods attempted
Besides BIC, we attempted five other methods to perform variable selection including forward AIC, R-squared, Mallow's CP, LASSO regularization and a custom method we designed. Besides regression models, we also fitted a 10-layer decision tree. For all the models mentioned above, we used the same method to determine the optimal decision boundary as our final model: namely we use a fpr vs fnr plot and find the boundary that would minimize fpr while keeping the fnr under 5%. However, it should be noted that since LASSO and our custom method rely on cross validation, we only split the dataset intro training (90%) and test data (10% and no validation set). Therefore, the fpr vs fnr plot was constructed using the fitted values on the training set instead of the predictions of the validation set. However, we keep the same random to ensure our comparison between different models are valid and fair.

```{r, echo=FALSE, warning=FALSE, include=FALSE}
## Variable selection using information criterias (AIC/BIC/adjusted r^2)


formula.full =formula("target ~ (age + sex + cp + trestbps + chol + fbs + restecg + thalach + exang + oldpeak + slope + ca + thal)^2")
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
### adjusted r^2
```

```
  cols = names(which(features.logical))
  cols = cols[2:length(cols)]

  return(paste0('target ~ ', paste(cols, collapse=" + " )))
}

adjr2.formula = formula(regsubset_to_formula.str(adjr2.forward))
aic.formula = aic.forward$formula
cp.formula = formula(regsubset_to_formula.str(cp.forward))
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Build separate models
aic.model <- glm(aic.formula, data = train_dat, family = "binomial")
adjr2.model <- glm(adjr2.formula, data = train_dat, family = "binomial")
cp.model <- glm(cp.formula, data = train_dat, family = "binomial")
```

Mallow's CP
We attempted Mallow's CP because Mallow's CP minimizes the model's residual deviance (or maximize the liklihood of the data) and therefore seems a valid criteria to select the most helpful features.

```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 5, fig.height = 2.5}
#### CP
par(mfrow =c(1, 2))

cp.val_pred = predict(cp.model, val_dat, type='response')
pred.confusion = ROCR::prediction(cp.val_pred, val_dat$target)
perf.acc = ROCR::performance(pred.confusion, "acc")
plot(perf.acc, cex.lab=0.7)
abline(v = 0.33)

perf.roc = ROCR::performance(pred.confusion,"fnr","fpr")
plot(perf.roc, colorize = T, lwd = 2, cex.lab=0.7)
abline(a = 0, b = 1)
abline(a = 0.05, b = 0)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
length(cp.model$coefficients)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
cp.threshold = 0.33

cp.pred.val = predict(cp.model, val_dat, type='response') > cp.threshold
cal.accuracy(cp.pred.val, val_dat$target)
cal.fp(cp.pred.val, val_dat$target)
cal.fn(cp.pred.val, val_dat$target)
```

For Mallow's CP, the decision boundary that maximizes te overall accuracy (0.33)

```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Test the final model against the test set
cp.model.plus.val = glm(cp.formula, data = train.val.dat, family = "binomial")
cp.pred.test = predict(cp.model.plus.val, test_dat, type='response') > cp.threshold
cal.accuracy(cp.pred.test, test_dat$target)
cal.fp(cp.pred.test, test_dat$target)
cal.fn(cp.pred.test, test_dat$target)
```

When tested against the test set, the model achieves an amazing overall accuracy of 92.16%, fpr 13.46% and fnr 2%. Although the acccuracy and fpr were signficantly improved without increasing the fnr, the Mallow's CP selected 46 features and given such a huge model has no interpretability in the medical practice, we decided not to use it even with its superior performance.

Adjusted R-squared
We attempted adjusted R-squared because adjusted R-squared directly minimizes the model's residual deviance (or maximize the liklihood of the data) while penalizing the larger models and therefore seems a valid criteria to select the most helpful features.

```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 5, fig.height = 2.5}
#### Adjr2
par(mfrow =c(1, 2))

adjr2.val_pred = predict(adjr2.model, val_dat, type='response')
pred.confusion = ROCR::prediction(adjr2.val_pred, val_dat$target)
perf.acc = ROCR::performance(pred.confusion, "acc")
plot(perf.acc, cex.lab=0.7)
abline(v = 0.31)


perf.roc = ROCR::performance(pred.confusion,"fnr","fpr")
plot(perf.roc, colorize = T, lwd = 2, cex.lab=0.7)
abline(a = 0, b = 1)
abline(a = 0.05, b = 0)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
adjr2.threshold = 0.31

adjr2.pred.val = predict(adjr2.model, val_dat, type='response') > adjr2.threshold
cal.accuracy(adjr2.pred.val, val_dat$target)
cal.fp(adjr2.pred.val, val_dat$target)
cal.fn(adjr2.pred.val, val_dat$target)
```

For adjusted R-squared, the decision boundary that maximizes te overall accuracy (0.51) also keeps the fnr below 5%.

```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Test the final model against the test set
adjr2.model.plus.val = glm(adjr2.formula, data = train.val.dat, family =
"binomial")
```

LASSO
We also attempted to fit a LASSO regularized logistic model. LASSO can be a valid method because it does feature selection implicitly by minimizing the residual deviance while penalizing large coefficients and forcing many small coefficients to 0. To find the best beta (regularization parameter), we used 10-fold cross-validation. Cross validation is a valid method to find beta because it finds the beta that minimizes the overall residuan deviance.

```{r, echo=FALSE, warning=FALSE, include=FALSE}
## Split data into train and test
set.seed(123)
dat <- read.csv("heart.csv", sep = ",")

split_dummy <- sample(c(rep(0, 9/10 * nrow(dat)),
                        rep(1, 1/10 * nrow(dat))))

train_dat <- dat[split_dummy == 0, ]
test_dat <- dat[split_dummy == 1, ]
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
print(sum(train_dat$target == 0) /  nrow(train_dat))
print(sum(test_dat$target == 0) / nrow(test_dat))
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
features = model.matrix(target ~ (age + sex + cp + trestbps + chol + fbs + restecg
+ thalach + exang + oldpeak + slope + ca + thal)^2, train_dat)

response = train_dat$target

#perform k-fold cross-validation to find optimal lambda value
lasso.model <- cv.glmnet(features, response, family="binomial", alpha = 1)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
coef(lasso.model)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
print(sum(coef(lasso.model) != 0))
```

```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 5, fig.height = 2.5}
### Choose the threshold (ROC Curve)
```

```
perf.acc = ROCR::performance(pred.confusion, "acc")
plot(perf.acc, cex.lab=0.7)
abline(v = 0.47)


perf.roc = ROCR::performance(pred.confusion,"fnr","fpr")
plot(perf.roc, colorize = T, lwd = 2, cex.lab=0.7)
abline(a = 0, b = 1)
abline(a = 0.05, b = 0)
```



```{r, echo=FALSE, warning=FALSE, include=FALSE}
lasso.threshold = 0.47


lasso.pred.train = predict(lasso.model, features, type='response') >
lasso.threshold
cal.accuracy(lasso.pred.train, train_dat$target)
cal.fp(lasso.pred.train, train_dat$target)
cal.fn(lasso.pred.train, train_dat$target)
```


For LASSO, the decision boundary that maximizes te overall accuracy (0.47) also keeps the fnr below 5%.


```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Test the final model against the test set
features.test = model.matrix(target ~ (age + sex + cp + trestbps + chol + fbs +
restecg + thalach + exang + oldpeak + slope + ca + thal)^2, test_dat)

lass.pred.test = predict(lasso.model, features.test, type='response') >
lasso.threshold
cal.accuracy(lass.pred.test, test_dat$target)
cal.fp(lass.pred.test, test_dat$target)
cal.fn(lass.pred.test, test_dat$target)
```


When tested against the test set, the model achieved the best peformance acrosss all the models we attempted with an overall accuracy of 96.1%, fpr of 5.77% and the fnr of 2%. In addition to outperforming our final model in all metrics, the decisioon boundary is very close to 0.5, which means the model was able distinguish between positive cases and negative cases without relying on the bias towards fnr we introduced by lowering the decision boudary in other models. The diagnostic plots also reflected that as we are no long seeing the skewnessed in residuals in our final model. However, the LASSO has a fatal flaw: it uses 84 out 93 features available and LASSO coefficients don't have an intuitive interpretation. Moreover, having such a large model made us worry that the model is overfitting for this specific dataset. It is possible that the dataset itself is biased (e.g. most of white males/females) even though we couldn't confirm this without further information. As such, we arrived at the conclusion that we should keep BIC as our final method.

Decision Tree

```{r, echo=FALSE, warning=FALSE, include=FALSE}
## Decision Tree
tree <- ctree(target~. ,data=train_dat, control=ctree_control(maxdepth=10))
```


```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 5, fig.height = 2.5}
par(mfrow =c(1, 2))

# plot(tree)

pred.confusion.tree = ROCR::prediction(predict(tree, train_dat), train_dat$target)
perf.acc = ROCR::performance(pred.confusion.tree, "acc")
plot(perf.acc, cex.lab=0.7)
abline(v = 0.42)


perf.roc = ROCR::performance(pred.confusion.tree,"fnr","fpr")
plot(perf.roc, colorize = T, lwd = 2, cex.lab=0.7)
abline(a = 0, b = 1)
abline(a = 0.05, b = 0)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
tree.threshold = 0.37

tree.pred.train.val = predict(tree, train_dat)
cal.accuracy(tree.pred.train.val > tree.threshold, train_dat$target)
cal.fp(tree.pred.train.val > tree.threshold, train_dat$target)
cal.fn(tree.pred.train.val > tree.threshold, train_dat$target)
```

For the 10-layer decision tree, the decision boundary that maximizes te overall accuracy (0.37) also keeps the fnr below 5%.

```{r, echo=FALSE, warning=FALSE, include=FALSE}
tree.pred.test = predict(tree, test_dat)
cal.accuracy(tree.pred.test > tree.threshold, test_dat$target)
cal.fp(tree.pred.test > tree.threshold, test_dat$target)
cal.fn(tree.pred.test > tree.threshold, test_dat$target)
```

When tested against the test set, the model achieved a slightly better overall accuracy of 86.27% and fpr 21.15%, but the fnr is very high at 6%. which is higher than our control rate. Therefore, even though the decision tree has the best interpretability and higher overall accuracy, we choose our final model over it because of the high fnr.

Custom Method
In addition to the traditional methods, we also attempted build a custom method for our project. We manually build a cross-validation function that uses accuracy or fpr + fnr as the validation metric. We then selected the most helpful individual feature based on the cross-validation accuracy. To determine which interaction term to include, we design a custom algorithm to calculate the error metric: the error metric we use for interactions is fpr + fnr if fnr is below our desired threshold

```
train_dat <- dat[split_dummy == 0, ]
test_dat <- dat[split_dummy == 1, ]
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
## Single Variable selection CV and accuracy
### Cross-validation

## Calculate the mean CV error
cv.func <- function(dat, k, response_name, formula, error.func, is.logistic=TRUE,
thres=0.5) {

    ## Create index for each fold
    folds <- createFolds(dat$target, k, list = TRUE, returnTrain = FALSE)
    cv.errors = c()

    for (i in 1:k) {
        ## Initialize current train/test in thei fold
        test_index = folds[[i]]
        cur_test = dat[test_index, ]
        cur_train = dat[-test_index, ]

        ## Fit the model using training data
        if (is.logistic) {
            cur_model = glm(formula, family = "binomial", data=cur_train)
        } else {
            cur_model = glm(formula, data=cur_train)
        }


        ## Calculate CV error using the error.funct
        if (is.logistic) {
            pred = predict(cur_model, cur_test, type='response') > thres
        } else {
            pred = predict(cur_model, cur_test)
        }
        cur.cv_error = error.func(pred, cur_test$target)
        cv.errors = c(cv.errors, cur.cv_error)
    }

    return(mean(cv.errors))
}


## Calculate false positive rate
cal.fp <- function(pred, observation) {
    num.fp = sum((pred == 1) & (observation == 0))
    return(num.fp / sum(observation == 0))
}


## Calculate false negative rate
cal.fn <- function(pred, observation) {
    num.fn = sum((pred == 0) & (observation == 1))
```

```
}


## Calculate accuracy
cal.accuracy <- function(pred, observation) {
    num.acc = sum(pred == observation)
    return (num.acc / length(observation))
}
```


```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Function that does auto feature selection
cv.feature_select <- function(data, response.var.str, cv.function, accuracy.func,
fp.func, fn.func, fold, baseline.formula.str, fn.threshold, features,
use_accuracy=FALSE, use_fp=TRUE, use_fn=TRUE) {

  ## Basline accuracy, fp and fn
  baseline.formula = formula(baseline.formula.str)
  best.accuracy = cv.function(data, fold, response.var.str, baseline.formula,
accuracy.func, TRUE, 0.5)
  best.fp = cv.function(data, fold, response.var.str, baseline.formula, fp.func,
TRUE, 0.5)
  best.fn = cv.function(data, fold, response.var.str, baseline.formula, fn.func,
TRUE, 0.5)
  best.formula.str = baseline.formula.str

  for (feature in features) {
    cur.formula.str = paste0(best.formula.str, ' + ', feature) # Create a new
formula with an interactions term
    cur.formula = formula(cur.formula.str)

    ## Calculate the accuracy, fp and fn of the current model
    cur.accuracy = cv.function(data, fold, response.var.str, cur.formula,
accuracy.func, TRUE, 0.5)
    cur.fp = cv.func(data, fold, response.var.str, cur.formula, fp.func, TRUE, 0.5)
    cur.fn = cv.function(data, fold, response.var.str, cur.formula, fn.func, TRUE,
0.5)

    ## Penalize false negative rate
    if (cur.fn > fn.threshold){
      cur.fn.diff = (best.fn - cur.fn) * 100 ## Penalize the fn if fn is above the
desired threshold
    } else {
      cur.fn.diff = best.fn - cur.fn
    }

    ## Check if the feature should be included
    cur.net.benefit = 0 ## Overall metric

    ## Add the difference in accuracy if accuracy is used as a metric
    if (use_accuracy){
      cur.net.benefit = cur.net.benefit + cur.accuracy - best.accuracy
    }

    ## Substract the difference in fp if fp is used as a metric
    if (use_fp){
      cur.net.benefit = cur.net.benefit + best.fp - cur.fp
    }
```

```
      best.accuracy = cur.accuracy
      best.fp = cur.fp
      best.fn = cur.fn
    }
  }

  return(best.formula.str)
}
```


```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Choosing base model

# min.formula.str = 'target ~ sex + cp + trestbps + restecg + thalach + exang +
oldpeak + slope + ca + thal'
min.formula.str = 'target ~ sex'

all_base_features = c('thal', 'cp', 'trestbps', 'restecg', 'thalach', 'exang',
'oldpeak', 'slope', 'ca')


cv.base.formula.str = cv.feature_select(train_dat, 'target', cv.func, cal.accuracy,
cal.fp, cal.fn, 10, min.formula.str, 0.05, all_base_features, TRUE, FALSE, FALSE)
cv.base.formula.str
```


```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Test basline model

baseline.formula = formula(cv.base.formula.str)
cv.func(train_dat, 10, 'target', baseline.formula, cal.accuracy, TRUE, 0.5)
cv.func(train_dat, 10, 'target', baseline.formula, cal.fp, TRUE, 0.5)
cv.func(train_dat, 10, 'target', baseline.formula, cal.fn, TRUE, 0.5)
```



```{r, echo=FALSE, warning=FALSE, include=FALSE}
### Choose helpful interaction terms

baseline.formula.str = 'target ~ sex + thal + cp + trestbps  + thalach + exang +
oldpeak + slope + ca'
all_inter_features = c('sex:cp', 'sex:trestbps', 'sex:thalach', 'sex:exang',
'sex:oldpeak', 'sex:slope', 'sex:ca', 'sex:thal', 'cp:trestbps', 'cp:thalach',
'cp:exang', 'cp:oldpeak', 'cp:slope', 'cp:ca', 'cp:thal', 'trestbps:thalach',
'trestbps:exang', 'trestbps:oldpeak', 'trestbps:slope', 'trestbps:ca',
'trestbps:thal', 'thalach:exang', 'thalach:oldpeak', 'thalach:slope', 'thalach:ca',
'thalach:thal', 'exang:oldpeak', 'exang:slope', 'exang:ca', 'exang:thal',
'oldpeak:slope', 'oldpeak:ca', 'oldpeak:thal', 'slope:ca', 'slope:thal', 'ca:thal')


final.formula.str = cv.feature_select(train_dat, 'target', cv.func, cal.accuracy,
cal.fp, cal.fn, 10, baseline.formula.str, 0.05, all_inter_features)
```


```{r, echo=FALSE, warning=FALSE, include=FALSE}
final.formula.str
```

```
cv.func(train_dat, 10, 'target', final.formula, cal.accuracy, TRUE, 0.5)
cv.func(train_dat, 10, 'target', final.formula, cal.fp, TRUE, 0.5)
cv.func(train_dat, 10, 'target', final.formula, cal.fn, TRUE, 0.5)
```

```{r, echo=FALSE, warning=FALSE, include=TRUE, fig.width = 5, fig.height = 2.5}
### Choose the threshold (ROC Curve)
par(mfrow =c(1, 2))

pred.confusion = ROCR::prediction(final.model$fitted.values, train_dat$target)
perf.acc = ROCR::performance(pred.confusion, "acc")
plot(perf.acc, cex.lab=0.7)
abline(v = 0.58)


perf.roc = ROCR::performance(pred.confusion,"fnr","fpr")
plot(perf.roc, colorize = T, lwd = 2, cex.lab=0.7)
abline(a = 0, b = 1)
abline(a = 0.05, b = 0)
```

```{r, echo=FALSE, warning=FALSE, include=FALSE}
final.threshold = 0.21
cv.func(train_dat, 10, 'target', final.formula, cal.accuracy, TRUE,
final.threshold)
cv.func(train_dat, 10, 'target', final.formula, cal.fp, TRUE, final.threshold)
cv.func(train_dat, 10, 'target', final.formula, cal.fn, TRUE, final.threshold)
```

For our custom method, the decision boundary that maximizes the overall accuracy (0.58) failed to keep the fnr below 5% so we eventually used 0.21 decrease the fnr below 5%.

```{r, echo=FALSE, warning=FALSE, include=FALSE}
final.pred.test = predict(final.model, test_dat, type='response') > final.threshold
cal.accuracy(final.pred.test, test_dat$target)
cal.fp(final.pred.test, test_dat$target)
cal.fn(final.pred.test, test_dat$target)
```

When tested against the test set, the model achieved a slightly worse overall accuracy of 83.33% and fpr 30.77%, but the fnr is the same at 2%. However, our custom method used less features (12 in total) and hence the resulting model gives better interpretability. Still, due to its complex design, it is probably less convincing to doctors compared to our original model and BIC seems to have the same effect while being a well-established method. Therefore, we decide to keep BIC forward as our final method.

In addition to the models mentioned above, we also used AIC but we chose not to include the details of the model building process due to page limit. We summarized our findings in the table above.

```
Custom              & 13           & 0.21      & 0.833         & 0.308    & 0.02
\end{tabular}
\end{table}
```