Medical Data Analysis with Random Forest

Introduction:

The goal of this project is to show that it is possible to create a model that uses random forest to predict if a patient is likely to be readmitted to the hospital. Readmission is a major problem facing hospitals, according to (V. A., 2019) "Each year, Medicare analyzes the readmission rate for every hospital in the United States and then imposes financial penalties on those hospitals determined to have excessively high readmission rates. And every year, most U.S. hospitals get penalized. This year is no exception – 83% of all hospitals face penalties."

Part One: Research Question

Research Question: Which factors, if any, contribute most to a patient being readmitted to the hospital.

Step One: Install the nessecary packages.

In [1]:
```
install.packages("tidyverse")
```

```
package 'tidyverse' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\ContactTracer\AppData\Local\Temp\RtmpohUTxi\downloaded_packages
```

In [5]:
```
install.packages("randomForest")
```

```
package 'randomForest' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\ContactTracer\AppData\Local\Temp\RtmpohUTxi\downloaded_packages
```

In [14]:
```
library(readxl)
library(class)
library(tidyverse)
library(caret)
library(pROC)
library(ROCR)
library(repr)
library(randomForest)
```

```
Warning message:
"package 'randomForest' was built under R version 3.6.3"randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

The following object is masked from 'package:dplyr':

    combine
```

```
The following object is masked from 'package:ggplot2':

    margin
```

Step Two: Import and clean the data.

In [7]:
```
medical_clean <- read.csv("C:/Users/ContactTracer/Desktop/D207 Data files/medical_clean
```

In [9]:
```
medical_data <- medical_clean%>% select(15,16,17,19,20,21,22)
```

Data selection:

7 variables are selected from the initial dataframe. These variables were selected on the basis of them having the most obvious effect on patient readmittance. The seven variables selected are # of children, patient's age, income, gender, vitamin D levels, and previous doctor visits.

In [10]:
```
medical_data$Gender <- as.numeric(medical_data$Gender)
```

In [113…
```
write.csv(medical_data,"C:\\Users\\ContactTracer\\Desktop\\medical_data.csv", row.names
```

Part II: Method Justification: "Random forest is a machine learning algorithm that uses a collection of decision trees providing more flexibility, accuracy, and ease of access in the output." (Adjusted R-Squared, 2015) The problem to be solved in this project is a binary classification problem. Random forest is great at handeling situations like this and will be a great choice for this particular problem. The expected outcome of this analysis is that the random forest method will be over 50% successful in predicting if a patient will need to be readmitted to the hospital.

Pro's and con's of random forest (New Tech Dojo, 2018):

- Random forests are extremely flexible and have very high accuracy.
- They also do not require preparation of the input data. You do not have to scale the data.
- It also maintains accuracy even when a large proportion of the data are missing.
- The main disadvantage of Random forests is their complexity. They are much harder and time-consuming to construct than decision trees.
- They also require more computational resources and are also less intuitive. When you have a large collection of decision trees it is hard to have an intuitive grasp of the relationship existing in the input data.

In [15]:
```
output.forest <- randomForest(ReAdmis ~.,
           data = medical_data)
```

In [16]:
```
print(output.forest)
```

```
Call:
 randomForest(formula = ReAdmis ~ ., data = medical_data)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 39.94%
Confusion matrix:
      No Yes class.error
No  5644 687   0.1085137
Yes 3307 362   0.9013355
```

In [99]:
```
output.forest
```

```
Call:
 randomForest(formula = ReAdmis ~ ., data = medical_data)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 39.94%
Confusion matrix:
      No Yes class.error
No  5644 687   0.1085137
Yes 3307 362   0.9013355
```

In [18]:
```
importance(output.forest,type = 2)
```

A matrix: 6 × 1 of type dbl

	MeanDecreaseGini
Children	339.8205
Age	881.9105
Income	1264.1335
Gender	154.5128
VitD_levels	1258.0803
Doc_visits	327.2502

In [20]:
```
medical_final<-medical_data%>% select(2,3,5,6)
```

In [36]:
```
med_samp <- sample(2, nrow(medical_final), replace=TRUE, prob=c(0.67, 0.33))
```

In [21]:
```
head(medical_final)
```

A data.frame: 6 × 4

	Age	Income	ReAdmis	VitD_levels
	<int>	<dbl>	<fct>	<dbl>
1	53	86575.93	No	19.14147
2	51	46805.99	No	18.94035
3	53	14370.14	No	18.05751
4	78	39741.49	No	16.57686

	Age	Income	ReAdmis	VitD_levels
	<int>	<dbl>	<fct>	<dbl>
5	22	1209.56	No	17.43907
6	76	81999.88	No	19.61265

In [23]:
```
med_samp <- sample(2, nrow(medical_final), replace=TRUE, prob=c(0.67, 0.33))
```

In [24]:
```
med_training <- medical_final[med_samp==1, 1:3]
```

In [25]:
```
med_test <- medical_final[med_samp==2, 1:3]
```

In [26]:
```
med_training_group <- medical_final[med_samp==1, 4]
```

In [27]:
```
med_testing_group <- medical_final[med_samp==2, 4]
```

In [31]:
```
med_prediction<- randomForest(ReAdmis ~.,
           data = med_training)
```

In [49]:
```
med_prediction
```

```
Call:
 randomForest(formula = ReAdmis ~ ., data = med_training)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 40.48%
Confusion matrix:
      No Yes class.error
No  3589 602   0.1436411
Yes 2083 359   0.8529894
```

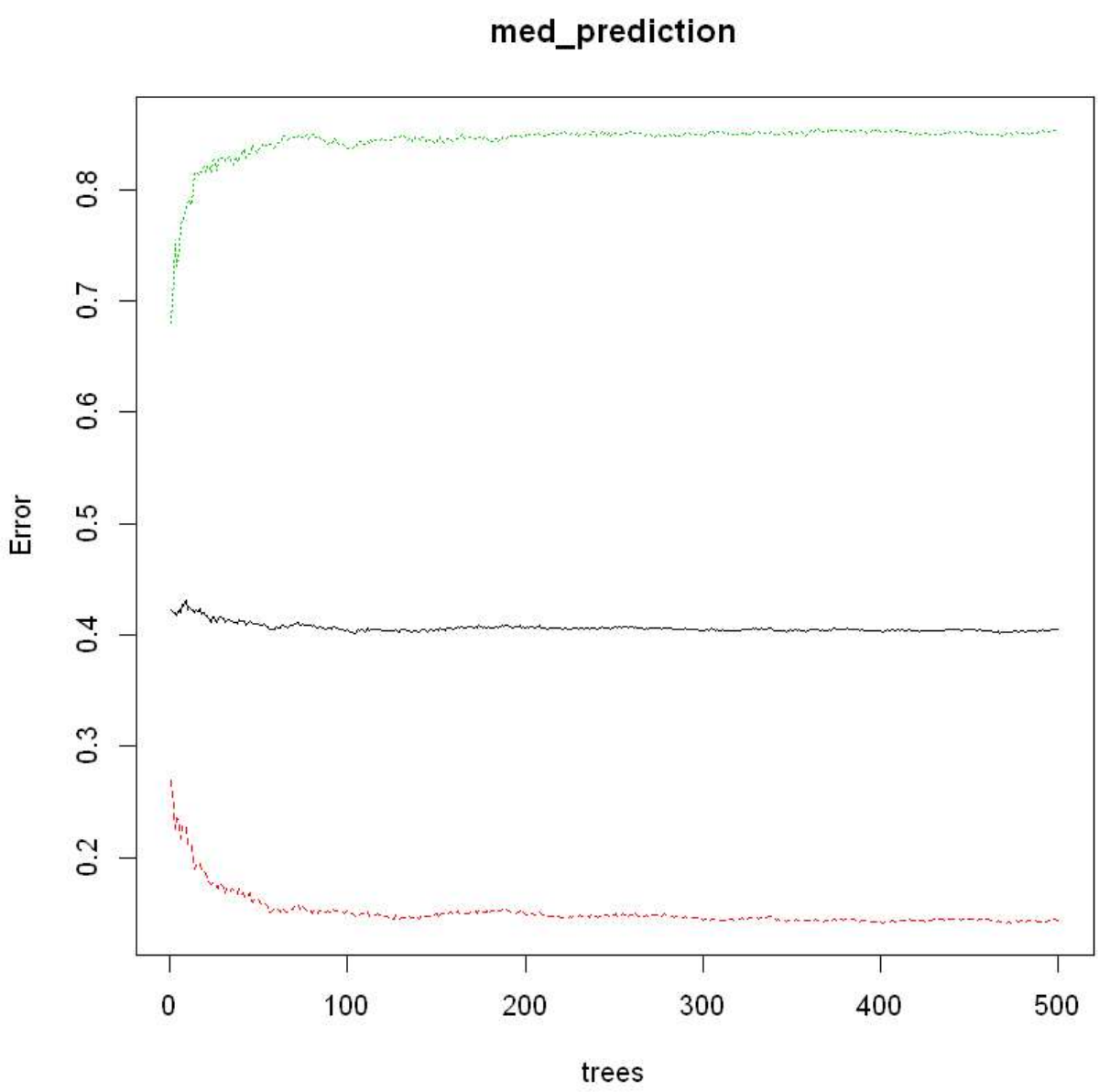
In [34]:
```
med_testing<- randomForest(ReAdmis ~.,
           data = med_test)
```

In [50]:
```
med_testing
```

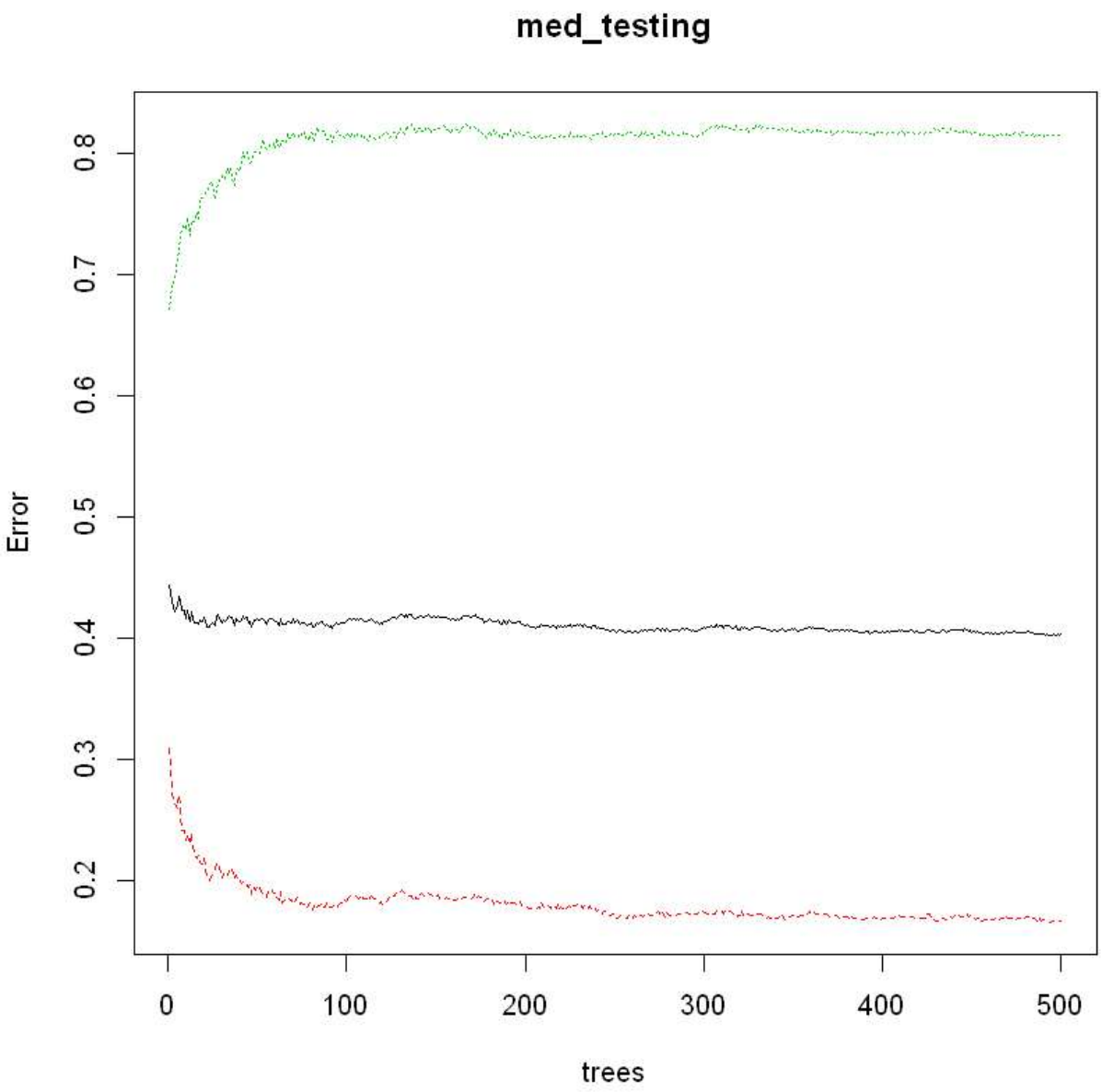
```
Call:
 randomForest(formula = ReAdmis ~ ., data = med_test)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 40.33%
Confusion matrix:
      No Yes class.error
No  1783 357   0.1668224
Yes 1001 226   0.8158109
```

In [54]:
```
plot(med_prediction)
```



In [55]:
```
plot(med_testing)
```



From the graphs above 100 trees is the ideal value for minimizing errors.

Next create new models featuring ntree = 100.

In [56]:
```
med_prediction100<- randomForest(ReAdmis ~.,
           data = med_training, ntree=100)
```

In [57]:
```
med_prediction100
```

```
Call:
 randomForest(formula = ReAdmis ~ ., data = med_training, ntree = 100)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 1

        OOB estimate of  error rate: 40.52%
Confusion matrix:
      No Yes class.error
```

```
No  3529 662   0.1579575
Yes 2026 416   0.8296478
```

The OOB estimate of error rate is just the mean square error, so we see that this model has a MSE of 40%

In [62]:
```
prediction_accuracy=100-40.52
```

In [63]:
```
prediction_accuracy
```

59.48

The accuracy of the predicted model is 60%

In [58]:
```
med_testing100<- randomForest(ReAdmis ~.,
            data = med_test, ntree=100)
```

In [112…
```
med_testing100
```

```
Call:
 randomForest(formula = ReAdmis ~ ., data = med_test, ntree = 100)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 1

        OOB estimate of  error rate: 41.02%
Confusion matrix:
      No Yes class.error
No  1759 381   0.1780374
Yes 1000 227   0.8149959
```

The OOB estimate of error rate is just the mean square error, so we see that this model has a MSE of 41%

In [64]:
```
test_accuracy=100-41.02
```

In [65]:
```
test_accuracy
```

58.98

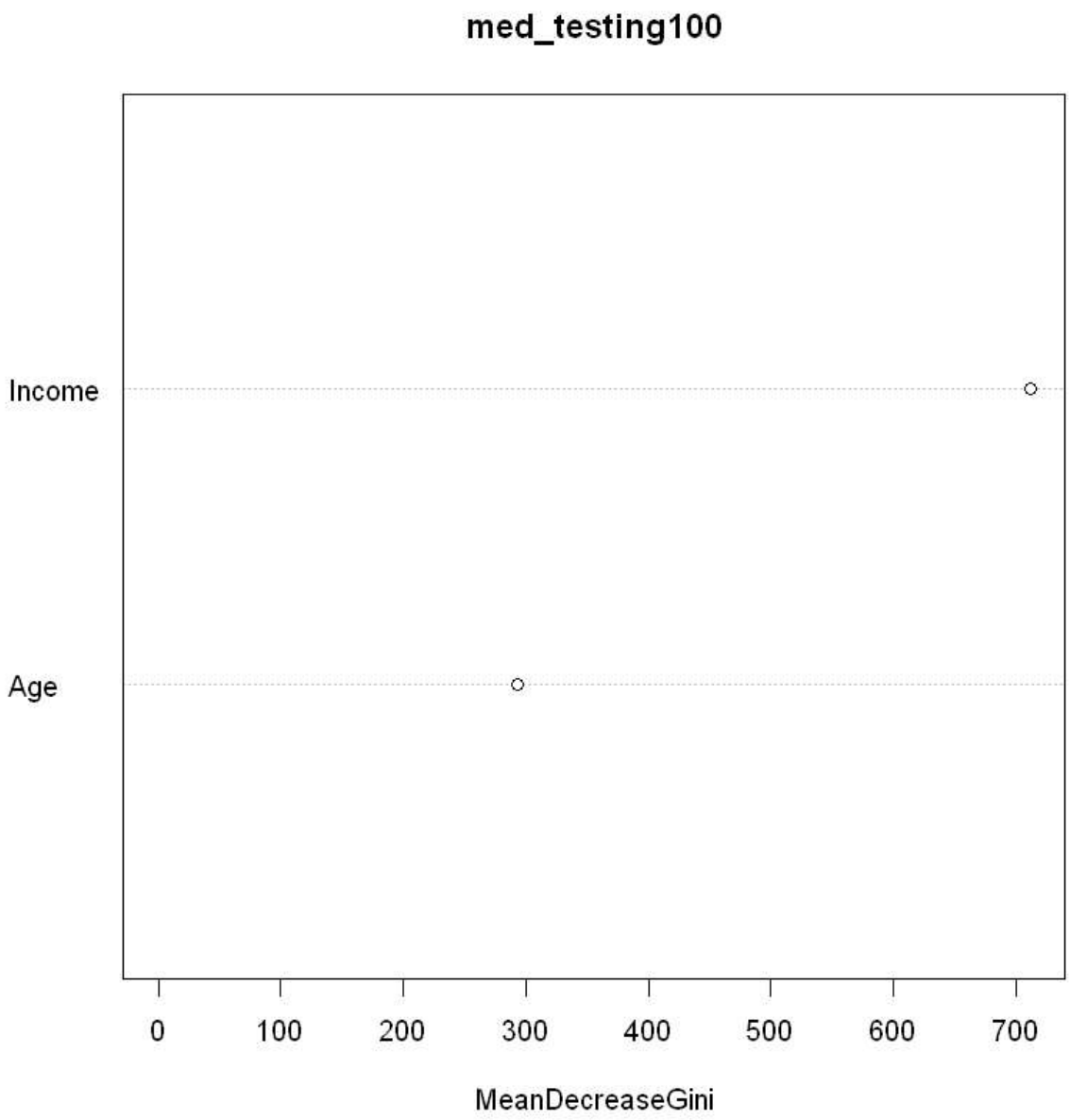
The accuracy of the test model is 60%

In [108…
```
importance(med_testing100)
```

A matrix: 2 × 1 of type dbl

	MeanDecreaseGini
Age	292.7835
Income	711.4841

In [109…
```
varImpPlot(med_testing100)
```

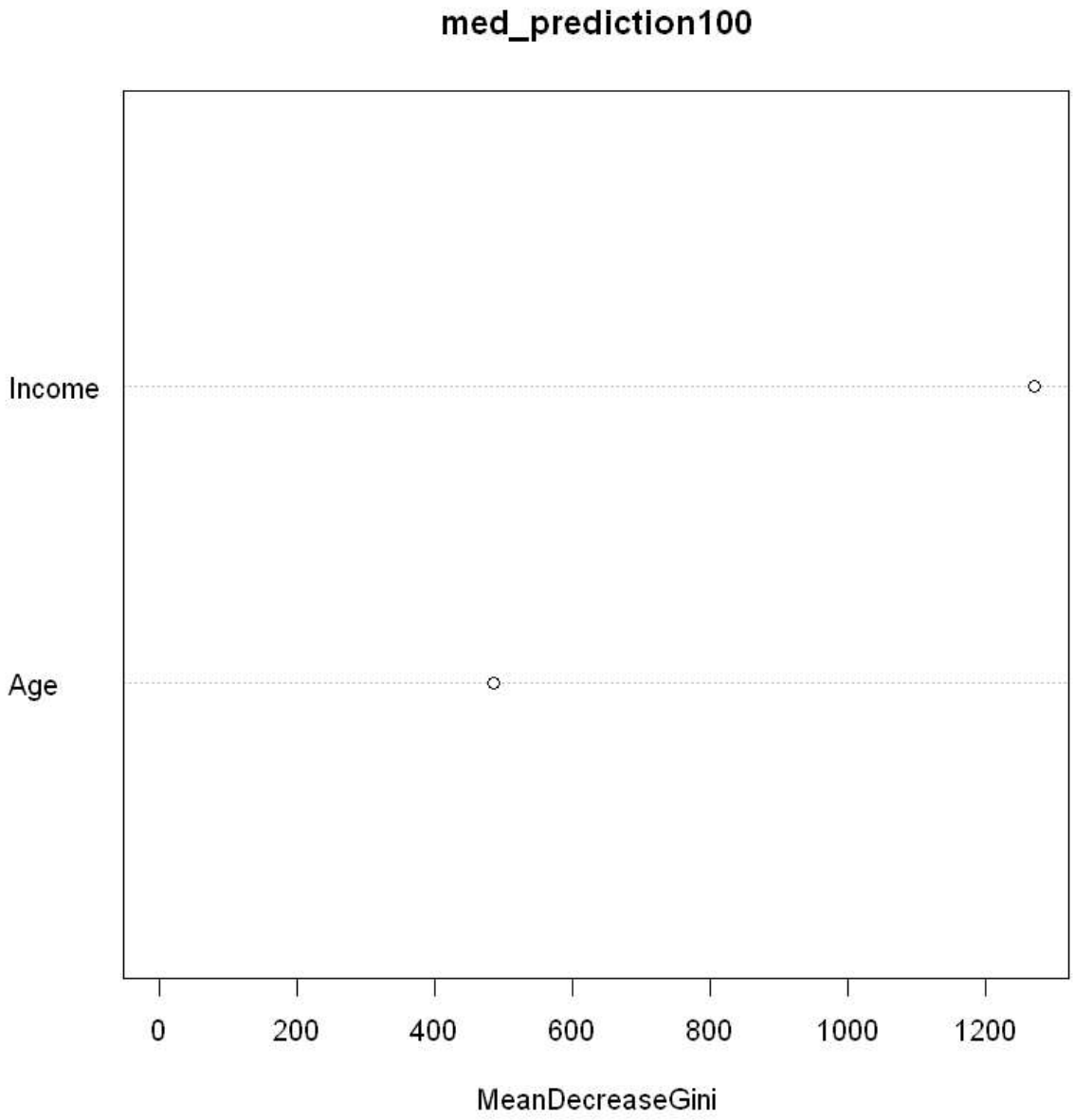


In [110…
```
importance(med_prediction100)
```

A matrix: 2 × 1 of type dbl

	MeanDecreaseGini
Age	485.7555
Income	1269.9648

In [111…
```
varImpPlot(med_prediction100)
```



Conclusion:

The two biggest factors in terms of hospital readmission are shown to be income and age. People with higher income tend to have better insurance so it would make sense that they would be more likely to be admitted to a hospital. Older people tend to have more medical issues and so it makes sense that they would be readmitted more often.

References:

Adjusted R-squared. (2015). Corporatefinanceinstitute.Com. https://corporatefinanceinstitute.com/resources/knowledge/other/adj r-squared/

N. (2018, October 18). Learn Random Forest using Excel. New Tech Dojo. https://www.newtechdojo.com/learn-random-forest-using-excel/#:%7E:text=Pros%20and%20Cons%20of%20Random%20Forest%

V. A. (2019, November 17). The 2020 Medicare Readmission Penalty Program. The Hospital Medical Director. https://hospitalmedicaldirector.com/the-2020-medicare-readmission-penalty-program/