Obesity Models

Aleksander Rodriguez

2024-04-26

Introduction

Within the data modeling and building these model to predict BMI levels according to variables like ones physical activity and how much water they consume. This file will use the caret, ggplot2, and metrics library to build, evaluate and communicate the models created. The linear regression, support vector machine with the radial function, and the random forest models are used to predict BMI. Within our finding some models are better than others and show tools on how to make a model better.

First, we must call the libraries that will be at use, read the csv file and create. the BMI column

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(ggplot2)
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.3.3
```

```
##
## Attaching package: 'Metrics'
```

```
## The following objects are masked from 'package:caret':
##
##     precision, recall
```

```
obesity <- read.csv("C:/Users/Alek4/OneDrive/Documents/CsvFiles/ObesityDataSet.csv")

obesity$BMI <- obesity$Weight/(obesity$Height ^ 2)
```

Model 1: Linear Regression Model

The following model will create a linear regression model to predict BMI numbers with variables found to have a strong correlation with BMI. These variables consist of Physical_Activity, age, High_Cal, Veggies_consumed, Monitor_Cals, Water_Consumed, and family_history_with_overweight (the other models will use the same variables). We must first divide the data and place 80% of the data in a train data frame, and 20% in a test data frame. We will use these data frames to first train the model with the train data frame and then test it with the test data frame. Once built, we will see how well the model can predict in comparison to the test data frame. We will test its correctness with the rooted mean squared error formula (RMSE) and the R-Squared formula. For this model the RMSE value stays between 6 and 7 which is a reasonable value but should be better. The R-squared value stays between 33 and 40, which is not great. If one had a BMI of 25 they could be predicted to have a BMI of 32 which would make them obese. This model is not the best at predicting BMI but there are other models and refining that could make those numbers better.

```
# divide as training and testing: 20% test 80% train and get the training data size
sample_size <-floor(.8* nrow(obesity))
sample_size
```

```
## [1] 1688
```

```
# get the train data index
train_ind <- sample(seq_len(nrow(obesity)), size = sample_size)

# generate training and test data set
train <- obesity[train_ind,]
test <- obesity[-train_ind,]

# creating a linear regression model to predict BMI with
# Physical_Activity, age, High_Cal, Veggies_consumed, Monitor_Cals,
# Water_Consumed, and family_history_with_overweight
BMI_Linear <- lm(BMI ~ Physical_Activity + Age + High_Cal +
                   Veggies_consumed + Monitor_Cals + Water_Consumed +
                   family_history_with_overweight,
                 data = train)
summary(BMI_Linear)
```

```
##
## Call:
## lm(formula = BMI ~ Physical_Activity + Age + High_Cal + Veggies_consumed +
##     Monitor_Cals + Water_Consumed + family_history_with_overweight,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2061  -4.2877   0.6322   4.3079  18.1239
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       6.23409    1.16615   5.346 1.02e-07 ***
## Physical_Activity                -1.33567    0.19029  -7.019 3.23e-12 ***
## Age                               0.17376    0.02576   6.745 2.09e-11 ***
## High_Calyes                       3.25608    0.52562   6.195 7.33e-10 ***
```

```
## Veggies_consumed                   3.57978    0.29094  12.304  < 2e-16 ***
## Monitor_Calsyes                   -2.19614    0.82460  -2.663  0.00781 **
## Water_Consumed                     1.31095    0.26412   4.964 7.62e-07 ***
## family_history_with_overweightyes  8.04065    0.43751  18.378  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.403 on 1680 degrees of freedom
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3571
## F-statistic: 134.9 on 7 and 1680 DF,  p-value: < 2.2e-16
```

```
# predict on test data set
PredictedBMI <- predict(BMI_Linear, newdata = test)
head(PredictedBMI)
```

```
##        2        6        8       16       19       20
## 26.39281 24.31077 15.83133 26.59129 34.79402 28.05276
```

```
# calculate the rmse
sqrt(mean((test$BMI - PredictedBMI)^2))
```

```
## [1] 6.28103
```

```
# R squared
cor(test$BMI, PredictedBMI)^2
```

```
## [1] 0.3981468
```

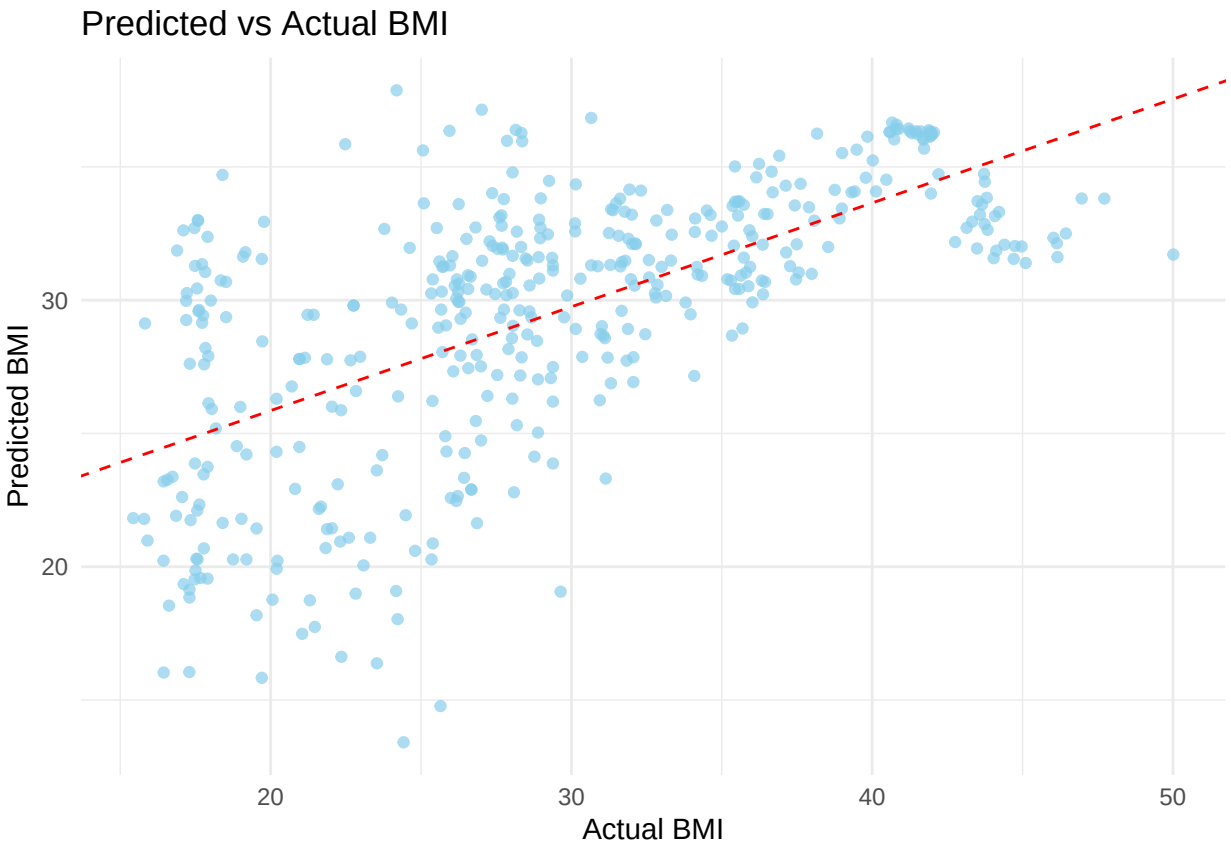
Model 1: Visual

The following is a graph comparing the predicted values compared to the actual values. It gives a visual on the models predictability. Through the graph one can see that the model is does not have a good capability of predicting BMI which the RMSE and the R-squared value also concluded. The predictability can be a lot better. We can conclude that the linear regression models is not a good choice to predict BMI based on the variables we have chosen.

```
# make a data frame with the test$BMI values and the predictedBMI so
# we can graph it
plot_data <- data.frame(test_BMI = test$BMI, predicted_BMI = PredictedBMI)

# make the the predictive liner regression line so we can use it in the plot
linear_model <- lm(predicted_BMI ~ test_BMI, data = plot_data)

# Create scatter plot with predicted and actual BMI
ggplot(plot_data, aes(x = test_BMI, y = predicted_BMI)) +
  geom_point(color = "skyblue", alpha = 0.7) +  # Predicted BMI in blue
  geom_abline(intercept = coef(linear_model)[1], slope = coef(linear_model)[2], color = "red",
              linetype = "dashed") +  # Add a line of equality
  labs(x = "Actual BMI", y = "Predicted BMI", title = "Predicted vs Actual BMI") +
  theme_minimal()
```



Model 2: Support Vector Machine With Radial Function

The following model is a support vector machine with a radial basis function kernel to predict a persons BMI. It will use that same train and test data fames as model one. The difference between this model is the usage of refinement tools. The usage of the trainControl function allows us to use cross validation. The "number" part in the trainControl function determines the number of how many times the data is divided in folds. One of the folds is used for training and the rest is used for testing. The "repeats" part of the function is used to state how many times the process of the trainControl is repeated. We then use this trainControl function and add it to our model allowing us to have a more refined model. Once the model is made and compared our model to the test data frame one can now test if the model is able to predict BMI. The RMSE of this model stayed between the values of 5.4 and 5.9 which is better than our first model. The R-squared value is between 50 and 54, which is also better than our first model. This model is good but there are possibilities to improve. A different type of model or more refining could help the model to improve.

```
fitControl <- trainControl(
  method = "repeatedcv",
  number = 5,
  repeats = 2
)

BMI_radial <- train(BMI ~ Physical_Activity + Age + High_Cal +
                      Veggies_consumed + Monitor_Cals + Water_Consumed +
                      family_history_with_overweight,
                    data = train, method = "svmRadial", trControl = fitControl)
```

```r
# predict on test data set
Predicted_radial_BMI <- predict(BMI_radial, newdata = test)
head(Predicted_radial_BMI)
```

```
## [1] 19.68598 26.76965 23.27491 19.70377 36.32345 26.93829
```

```r
# calculate the rmse
sqrt(mean((test$BMI - Predicted_radial_BMI)^2))
```

```
## [1] 5.137005
```

```r
# R squared
cor(test$BMI, Predicted_radial_BMI)^2
```

```
## [1] 0.6048124
```

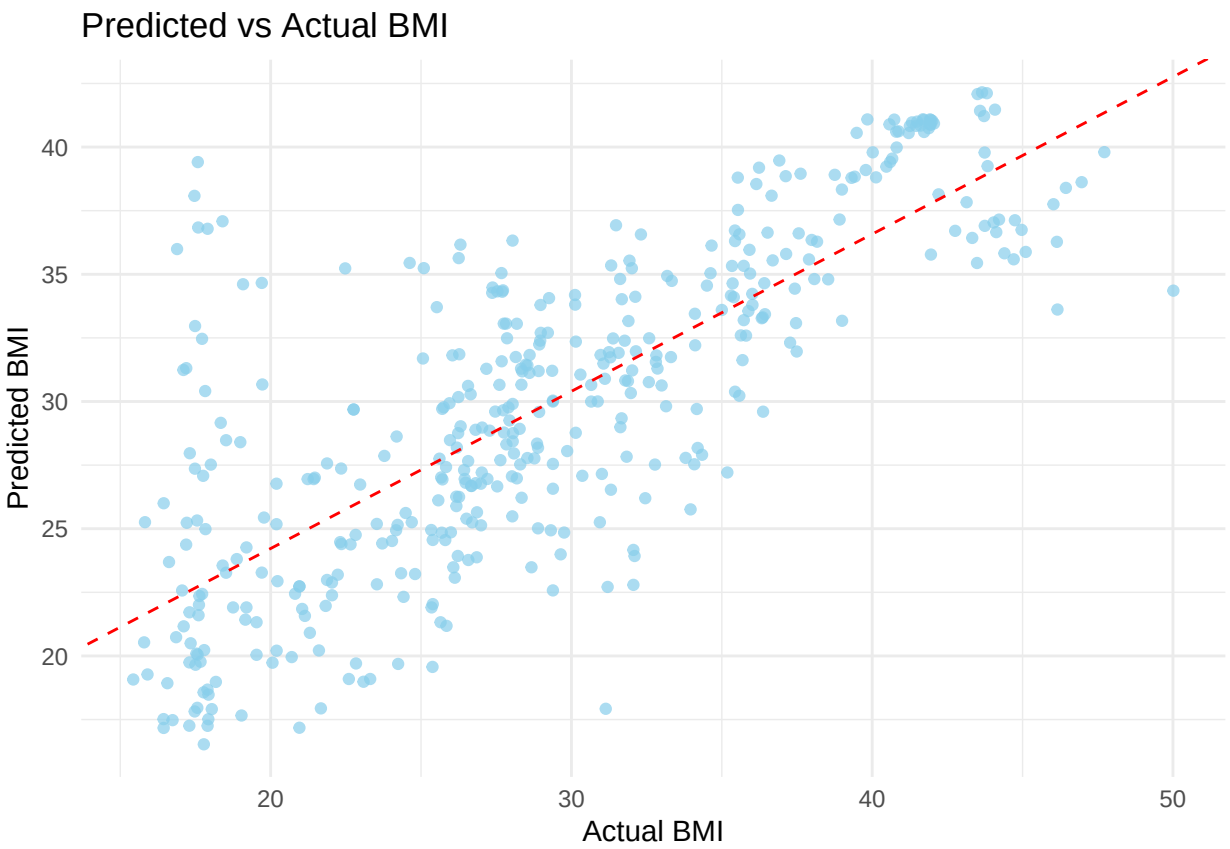
Model 2: Visual

The following is a graph comparing the predicted values compared to the actual values. It gives a visual on the models predictability capability. Through the graph one can see that it has a decent capability at predicting BMI which the RMSE and the R-squared value also concluded. This model is good but the predictability can be better. Through looking at this graph, RMSE values, and R-Squared values we can conclude that this model is decent at predicting BMI but cannot fully be relied on.

```r
# make a data frame with the test$BMI values and the predicted_BMI so
# we can graph it
plot_data <- data.frame(test_BMI = test$BMI, predicted_BMI = Predicted_radial_BMI)

# make the the predictive liner regression line so we can use it in the plot
linear_model <- lm(predicted_BMI ~ test_BMI, data = plot_data)

# Create scatter plot with predicted and actual BMI
ggplot(plot_data, aes(x = test_BMI, y = predicted_BMI)) +
  geom_point(color = "skyblue", alpha = 0.7) +  # Predicted BMI in blue
  geom_abline(intercept = coef(linear_model)[1], slope = coef(linear_model)[2],
              color = "red",linetype = "dashed") +  # Add a line of equality
  labs(x = "Actual BMI", y = "Predicted BMI", title = "Predicted vs Actual BMI") +
  theme_minimal()
```



Model 3: Random Forest Model

The following model is a random forest model that uses the same control function as model 2 which well have have 5-folds and repeat itself twice to refine the model. It will also use the same test and train data frames as model one. The random forest model will be created with the "ranger" method. Once it is created it will be tested by trying to predict the test data. After, the results are checked by using the RMSE formula and the R-squared formula. For this model, the RMSE value stay between 3.8 and 4.2. The R-squared value stays between 73 and 77. Both of these are immensely better compared to the past two models. One can state that this model is the best at predicting BMI levels with the variables that we found to have a good correlation with BMI levels. There are always possible ways in creating a better model with more data, variables, and different models. Overall, the random forest model was the best out of the three.

```
ranger_BMI <- train(BMI ~ Physical_Activity + Age + High_Cal +
                      Veggies_consumed + Monitor_Cals + Water_Consumed +
                      family_history_with_overweight,
                    data = train, method = "ranger", trControl = fitControl)

# predict on test data set
Predicted_ranger_BMI <- predict(ranger_BMI, newdata = test)
head(Predicted_ranger_BMI)
```

```
## [1] 23.76738 28.34123 23.56382 21.84417 30.09327 23.38412
```

```r
# calculate the rmse
sqrt(mean((test$BMI - Predicted_ranger_BMI)^2))
```

```
## [1] 3.881212
```

```r
# R squared
cor(test$BMI, Predicted_ranger_BMI)^2
```

```
## [1] 0.7729365
```

```r
#plot(model)
```

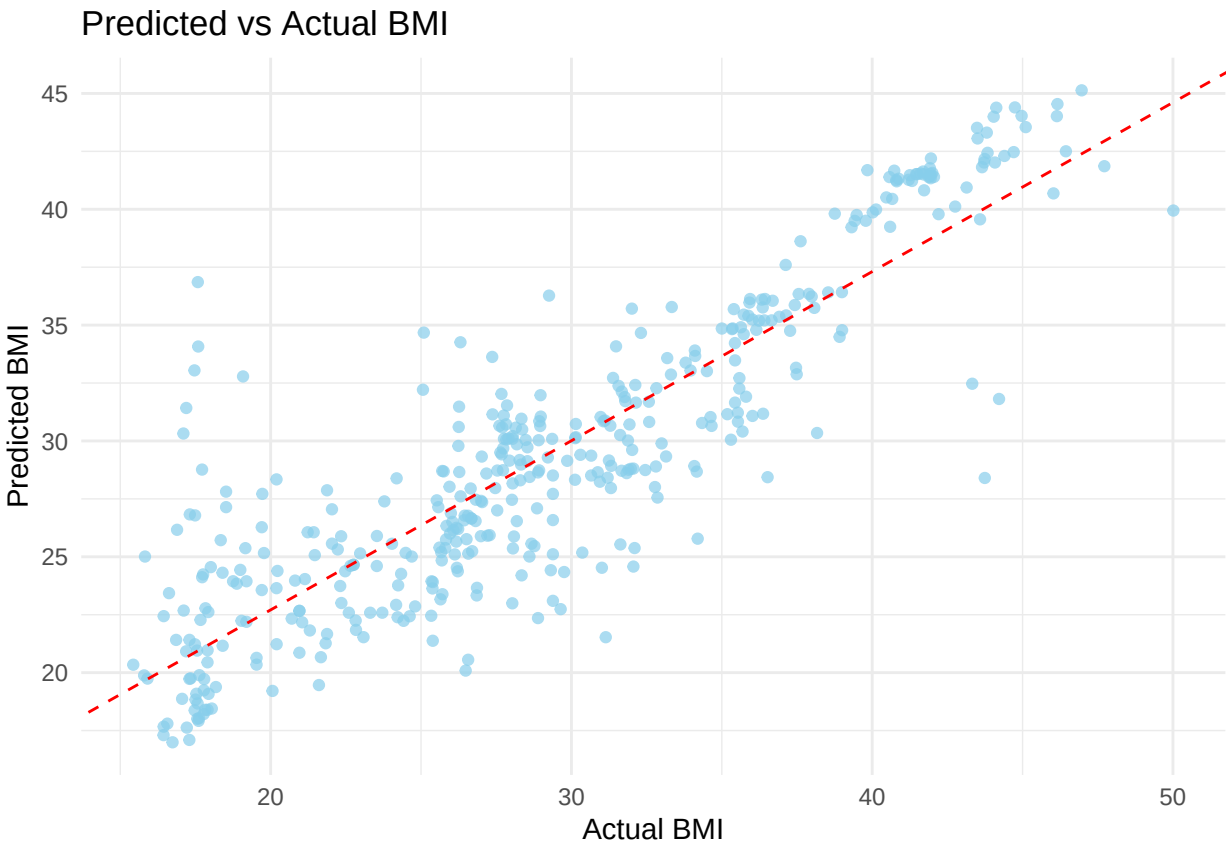
Model 3: Visual

The following is a graph comparing the predicted values compared to the actual values. It illustrates the predictability the model has. This model is very good at predicting BMI. This is the best model out of the three. It is not perfect but can predict BMI at a good rate which the RMSE value, R-Squared value, and this visual can prove.

```r
# make a data frame with the test$BMI values and the predicted_BMI so
# we can graph it
plot_data <- data.frame(test_BMI = test$BMI, predicted_BMI = Predicted_ranger_BMI)

# make the the predictive liner regression line so we can use it in the plot
linear_model <- lm(predicted_BMI ~ test_BMI, data = plot_data)

# Create scatter plot with predicted and actual BMI
ggplot(plot_data, aes(x = test_BMI, y = predicted_BMI)) +
  geom_point(color = "skyblue", alpha = 0.7) +  # Predicted weights in blue
  geom_abline(intercept = coef(linear_model)[1], slope = coef(linear_model)[2], color = "red",
              linetype = "dashed") +  # Add a line of equality
  labs(x = "Actual BMI", y = "Predicted BMI", title = "Predicted vs Actual BMI") +
  theme_minimal()
```


Predicted vs Actual BMI
Predicted BMI
Actual BMI
45
40
35
30
25
20
20
30
40
50