Internship Report in R

Internship Report

Importing Packages

Data Preprocessing Steps

```
###### Read the data in ######
data <- read.csv(file='insurance.csv')

###### Print the first rows ######
print(head(data, 5))
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
```

```
###### Print the columns' names ######
print(colnames(data))
```

```
## [1] "age"      "sex"      "bmi"      "children" "smoker"   "region"   "charges"
```

```
###### Print number of rows ######
print(nrow(data))
```

```
## [1] 1338
```

```
###### Converting to Numeric Variables ######
sex <- ifelse(data["sex"] == "female", 0, 1)
smoker  <- ifelse(data["smoker"] == "yes", 1, 0)
region <- as.numeric(data$region)

##### Replacing columns in the Data ######
data["sex"] <-  sex
data["smoker"] <-  smoker
data["region"] <- region
```

Linear Models - using the `purrr` package to get individual models

```r
###### Linear Regression ######
vars = c('age', 'sex', 'bmi', 'children', 'smoker', 'region')
#Using the purrr package to run all the models corresponding to the predictors
models <- vars %>% paste ('charges ~', .) %>% map(as.formula) %>% map(lm, data = data)
```

Summaries of the Models

Age

```r
# age summary
summary(models[[1]])
```

```
##
## Call:
## .f(formula = .x[[i]], data = ..1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8059  -6671  -5939   5440  47829
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3165.9      937.1   3.378 0.000751 ***
## age            257.7       22.5  11.453  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11560 on 1336 degrees of freedom
## Multiple R-squared:  0.08941,    Adjusted R-squared:  0.08872
## F-statistic: 131.2 on 1 and 1336 DF,  p-value: < 2.2e-16
```

Sex

```r
# sex summary
summary(models[[2]])
```

```
##
## Call:
## .f(formula = .x[[i]], data = ..1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12835  -8435  -3980   3476  51201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12569.6      470.1  26.740   <2e-16 ***
## sex           1387.2      661.3   2.098   0.0361 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 12090 on 1336 degrees of freedom
## Multiple R-squared:  0.003282,   Adjusted R-squared:  0.002536
## F-statistic:   4.4 on 1 and 1336 DF,  p-value: 0.03613
```

BMI

```
# bmi summary
summary(models[[3]])
```

```
##
## Call:
## .f(formula = .x[[i]], data = ..1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -20956  -8118  -3757   4722  49442
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1192.94    1664.80   0.717    0.474
## bmi           393.87      53.25   7.397 2.46e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11870 on 1336 degrees of freedom
## Multiple R-squared:  0.03934,    Adjusted R-squared:  0.03862
## F-statistic: 54.71 on 1 and 1336 DF,  p-value: 2.459e-13
```

Children

```
# children summary
summary(models[[4]])
```

```
##
## Call:
## .f(formula = .x[[i]], data = ..1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11585  -8759  -4071   3468  51248
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12522.5      446.5  28.049   <2e-16 ***
## children       683.1      274.2   2.491   0.0129 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12090 on 1336 degrees of freedom
## Multiple R-squared:  0.004624,   Adjusted R-squared:  0.003879
## F-statistic: 6.206 on 1 and 1336 DF,  p-value: 0.01285
```

Smoker

```r
# smoker summary
summary(models[[5]])
```

```
##
## Call:
## .f(formula = .x[[i]], data = ..1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -19221  -5042   -919   3705  31720
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8434.3      229.0   36.83   <2e-16 ***
## smoker       23616.0      506.1   46.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7470 on 1336 degrees of freedom
## Multiple R-squared:  0.6198, Adjusted R-squared:  0.6195
## F-statistic:  2178 on 1 and 1336 DF,  p-value: < 2.2e-16
```

Region

```r
# region summary
summary(models[[6]])
```

```
##
## Call:
## .f(formula = .x[[i]], data = ..1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12116  -8517  -3930   3347  50533
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13441.60     823.85  16.316   <2e-16 ***
## region        -68.04     299.86  -0.227    0.821
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12110 on 1336 degrees of freedom
## Multiple R-squared:  3.854e-05,  Adjusted R-squared:  -0.0007099
## F-statistic: 0.05149 on 1 and 1336 DF,  p-value: 0.8205
```

Linear Model with All Predictors

```r
###### Model with all the predictors  ######
allpreds <- lm(charges ~ ., data = data)
```

Summary of the Model

```r
###### Summary ######
summary(allpreds)
```

```
##
## Call:
## lm(formula = charges ~ ., data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11343  -2807  -1017   1408  29752
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11461.81     983.00 -11.660  < 2e-16 ***
## age            257.29      11.89  21.647  < 2e-16 ***
## sex           -131.11     332.81  -0.394 0.693681
## bmi            332.57      27.72  11.997  < 2e-16 ***
## children       479.37     137.64   3.483 0.000513 ***
## smoker       23820.43     411.84  57.839  < 2e-16 ***
## region        -353.64     151.93  -2.328 0.020077 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6060 on 1331 degrees of freedom
## Multiple R-squared:  0.7507, Adjusted R-squared:  0.7496
## F-statistic: 668.1 on 6 and 1331 DF,  p-value: < 2.2e-16
```

Linear Model with the Most Relevant Predictors

```r
most_rel <- lm(charges ~ age + bmi + children + smoker, data = data)
```

Summary of the Model

```r
summary(most_rel)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + children + smoker, data = data)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11897.9  -2920.8   -986.6   1392.2  29509.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12102.77     941.98 -12.848  < 2e-16 ***
## age            257.85      11.90  21.675  < 2e-16 ***
## bmi            321.85      27.38  11.756  < 2e-16 ***
## children       473.50     137.79   3.436 0.000608 ***
## smoker       23811.40     411.22  57.904  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6068 on 1333 degrees of freedom
## Multiple R-squared:  0.7497, Adjusted R-squared:  0.7489
## F-statistic: 998.1 on 4 and 1333 DF,  p-value: < 2.2e-16
```

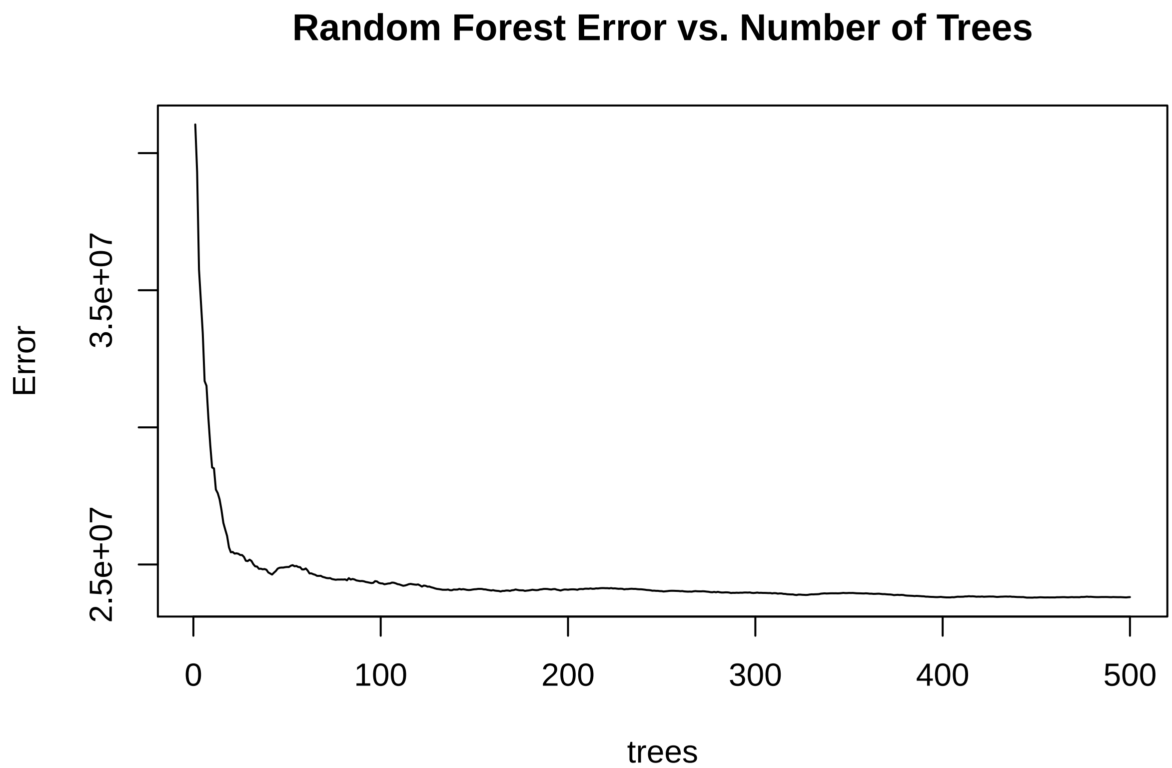
Random Forest Model

```
###### Random Forest Model ######
set.seed(100)

#setting a train and test set
train <- sample(nrow(data), 0.8*nrow(data), replace = FALSE)
trainset <- data[train,]
testset <- data[-train,]

random.forest1 <- randomForest(charges ~ ., data = trainset, ntree = 500, mtry = 6,
                               importance = TRUE)

random.forest1
```

```
##
## Call:
##  randomForest(formula = charges ~ ., data = trainset, ntree = 500,      mtry = 6, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 23810804
##                     % Var explained: 83.23
```

Generating the plot

```
plot(main = "Random Forest Error vs. Number of Trees", random.forest1)
```



Generating a Confusion Matrix

In order to get a better model, I decided to use the `ifelse()` function in R and get a cutoff of the data i.e. using the Mean and Median in this case **10,000 USD** to predict charges. **Less than or equal** to **10,000** is 0, and **more than or equal** is a 1.

Summary of the testset$charges variable

```
summary(testset$charges)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1136    4748    8277   13459   16357   63770
```

Confusion Matrix - using the `Caret` Package

```
###### Testing the model ######
prediction <- predict(random.forest1, newdata = testset)

prediction <- ifelse(prediction <= 10000, 0, 1)
testing <- ifelse(testset$charges <= 10000, 0, 1)

confusionMatrix(factor(prediction, levels = min(testing):max(testing)),
      factor(testing, levels = min(testing):max(testing)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 127   6
##          1  24 111
##
##                Accuracy : 0.8881
##                  95% CI : (0.8441, 0.9232)
##     No Information Rate : 0.5634
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7763
##
##  Mcnemar's Test P-Value : 0.001911
##
##             Sensitivity : 0.8411
##             Specificity : 0.9487
##          Pos Pred Value : 0.9549
##          Neg Pred Value : 0.8222
##              Prevalence : 0.5634
##          Detection Rate : 0.4739
##    Detection Prevalence : 0.4963
##       Balanced Accuracy : 0.8949
##
##        'Positive' Class : 0
##
```

Tuning the Random Forest Model

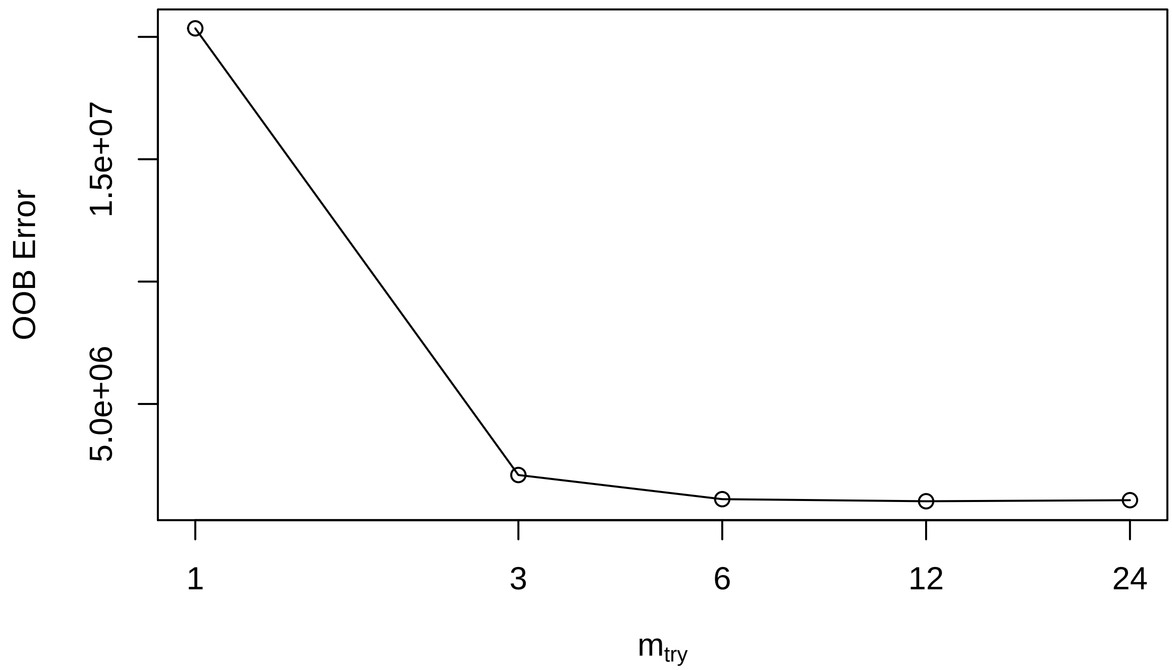
The tuneRF() function comes from the `randomForest` package.

According to the documentation, this function starts from the given parameter of `mtry` - 3 in this example - and searches for the **optimal value of mtry**.

With respect to Out-of-Bag error estimate

```
set.seed(100)
tuning.model <- tuneRF(
  x = testset,
  y = testset$charges,
  ntreeTry = 600,
  mtryStart = 3,
  stepFactor = 0.5,
  improve = 0.03,
  trace = FALSE
)
```

```
## 0.4708423 0.03
## 0.076087 0.03
## -0.04261731 0.03
## -18.84259 0.03
```



Benefits of Random Forest

-Easy to interpret the models
-Could be used for regression or classification
-Could be used in large datasets

Pitfalls of Random Forest

-Are prone to overfitting
-Accuraccy tends to be lower than other Machine Learning techniques
-High Variance

Citation: Towards AI

For Comparison with Python Models (Links)

GitHub Pages for the Internship | GitHub Repository
Heroku App - using Dash and Plotly