# Capstone Project

Prudential Life Insurance

Author: Francia Moscoso

Reference: https://github.com/moscosof/Prudential_CapstoneProject

# Prudential Life Insurance

**Introduction**

- The internet changed the insurance industry. Cheapest rates can be found online for the right coverage.
- Application process is antiquated. Customers provide extensive information to identify risk classification and eligibility, a process that takes an average of 30 days.

# Prudential Life Insurance

**The Challenge**

Predict the rating system in the existing Prudential Life Insurance assessment given a training data set from Kaggle competition.

If we could find out the significant attributes that determine the assessment, Prudential could streamline the process to make it quicker and less labor intensive.

# Prudential Life Insurance

## Training data set (https://www.kaggle.com/c/prudential-life-insurance-assessment/data)

train.csv:

- **Variable**
- **Description**
- Id — A unique identifier associated with an application.
- Product_Info_1-7 — A set of normalized variables relating to the product applied for
- Ins_Age — Normalized age of applicant
- Ht — Normalized height of applicant
- Wt — Normalized weight of applicant
- BMI — Normalized BMI of applicant
- Employment_Info_1-6 — A set of normalized variables relating to the employment history of the applicant.
- InsuredInfo_1-6 — A set of normalized variables providing information about the applicant.
- Insurance_History_1-9 — A set of normalized variables relating to the insurance history of the applicant.
- Family_Hist_1-5 — A set of normalized variables relating to the family history of the applicant.
- Medical_History_1-41 — A set of normalized variables relating to the medical history of the applicant.
- Medical_Keyword_1-48 — A set of dummy variables relating to the presence of/absence of a medical keyword being associated with the application.
- Response — This is the target variable, an ordinal variable relating to the final decision associated with an application

# Prudential Life Insurance

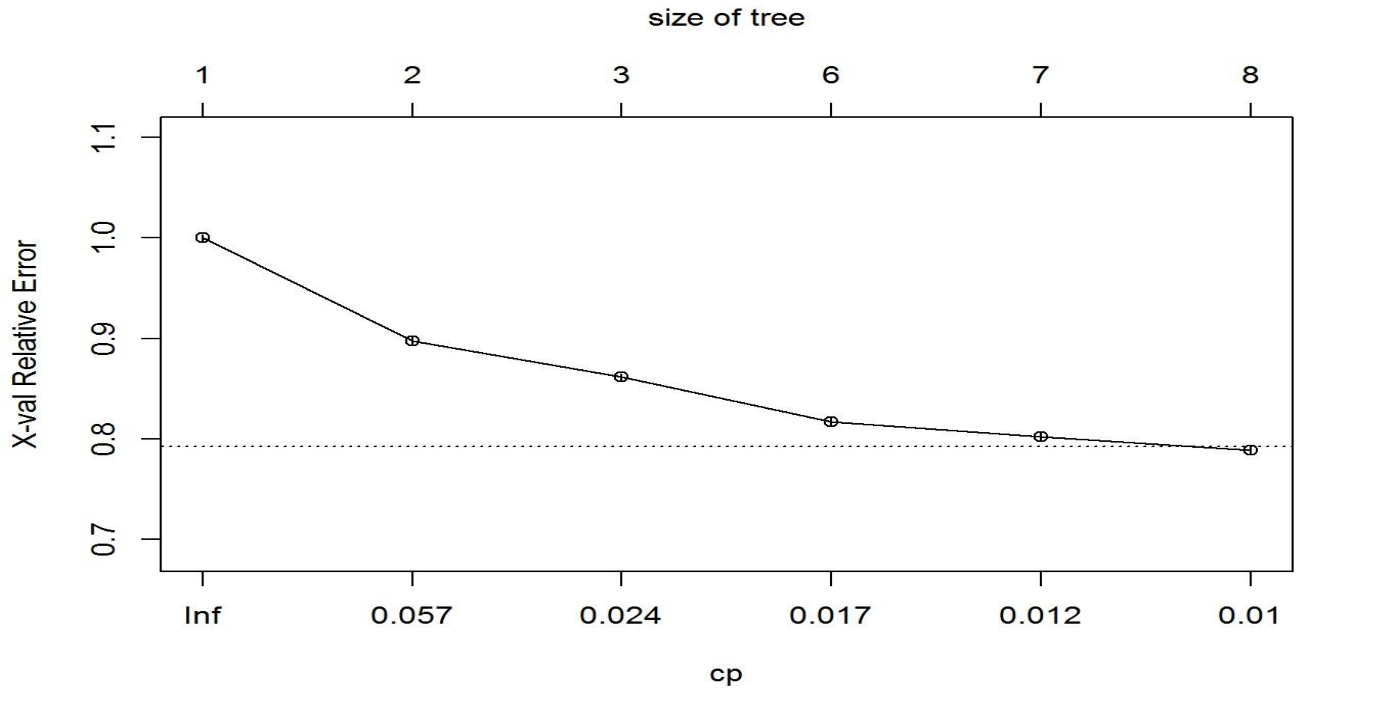
What models to apply ?

- **Classification Tree**
- **Random Forest**
- I decided to add a binomial output variable called 'Approved' that is set to 1 when Response = 8. Otherwise Approved = **0.** I could predict 'Approved' using **Logistic Regression** to have an idea of which independent variables are significant.

# Prudential Life Insurance

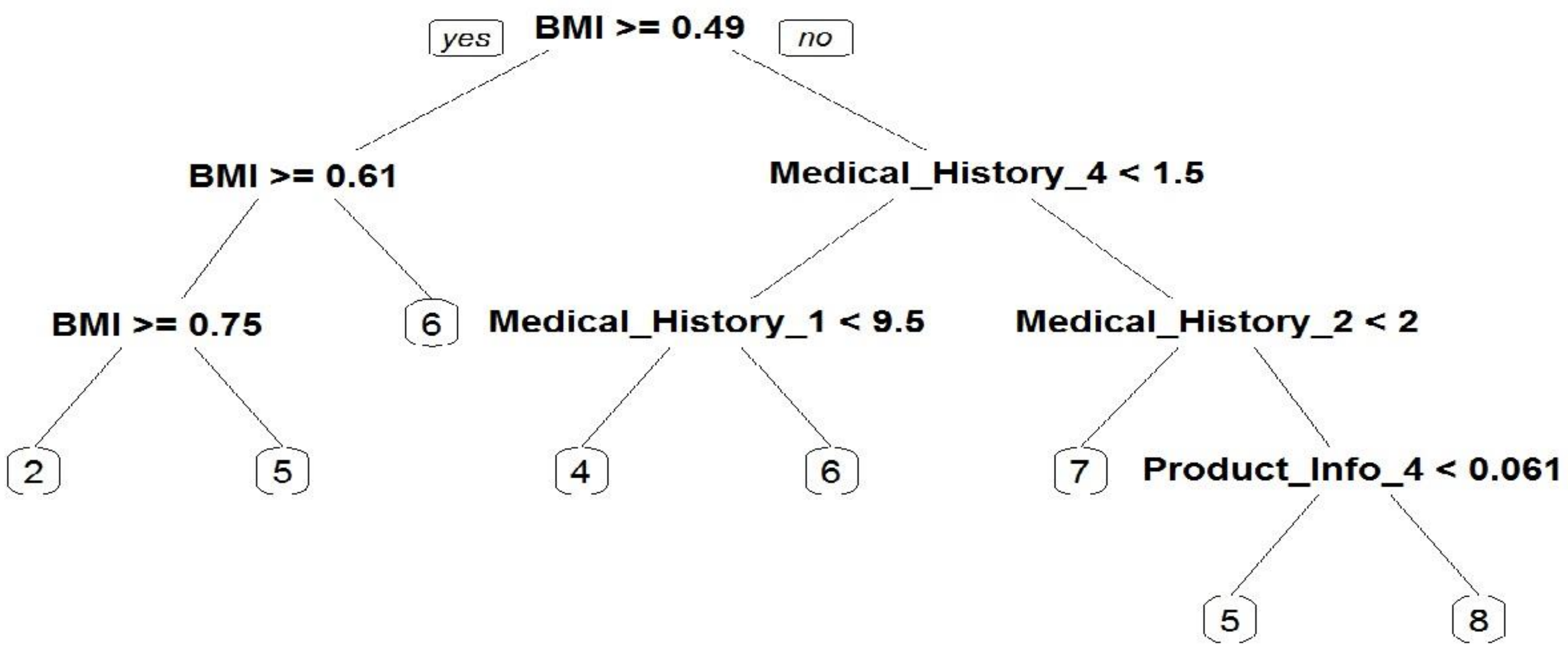
- **Classification Tree via “rpart”**

fit <- rpart(Response ~ ., method="class", data=train)



# Prune the tree

pfit<- prune(fit, cp=fit$cptable[which.min(fit$cptable[,"xerror"]),"CP"])



# Prudential Life Insurance

- **Classification Tree**

Making predictions:

pfit.predict = predict(pfit, newdata = testLog, type = "class")

table(pfit.predict)pfit.predict

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 0 | 432 | 0 | 513 | 1509 | 5898 | 968 | 5525 |

# Prudential Life Insurance

**Classification Tree: Accuracy**

table(testLog$Response, as.numeric(pfit.predict))

| | 2 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|
| 1 | 106 | 89 | 244 | 668 | 135 | 310 |
| 2 | 212 | 56 | 273 | 664 | 133 | 300 |
| 3 | 5 | 99 | 56 | 88 | 2 | 3 |
| 4 | 0 | 234 | 1 | 92 | 1 | 29 |
| 5 | 92 | 1 | 613 | 476 | 74 | 102 |
| 6 | 14 | 16 | 274 | 1885 | 149 | 470 |
| 7 | 1 | 3 | 41 | 1054 | 368 | 540 |
| 8 | 2 | 15 | 7 | 971 | 106 | 3771 |

Overall accuracy = (212 + 234 + 613 + 1885 + 368 + 3771) / 14,845 = 47.77%

# Prudential Life Insurance

## Random Forest

```
fit <- randomForest(Response ~ .,  nodesize = 25, ntree = 200, data=trainLog)
```

Number of trees: 200
No. of variables tried at each split: 42
Mean of squared residuals: 3.498086
% Var explained: 42.04

importance(fit) # *importance of each predictor*

| | | | |
|---|---|---|---|
| Bmi | 27737.38079 | Ins_Age | 6016.96877 |
| Wt | 12167.35153 | Medical_History_4 | 5375.22888 |
| Medical_History_23 | 10228.93635 | Medical_Keyword_15 | 4000.91795 |
| Medical_Keyword_3 | 9406.69365 | Family_Hist_3 | 2736.29612 |
| Product_Info_2 | 8957.23896 | Family_Hist_5 | 2577.72095 |
| Product_Info_4 | 6610.62719 | Medical_History_1 | 2513.09601 |

...........................

# Prudential Life Insurance

**Random Forest**

**Making Predictions:**

predictForest <- predict(fit, newdata = testLog )

predictForestRound <- round(predictForest,0)table(testLog$Response)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1552 | 1638 | 253 | 357 | 1358 | 2808 | 2007 | 4872 |

# Prudential Life Insurance

**Random Forest**

Accuracy

```
table(testLog$Response, predictForestRound)
```

predictForestRound

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 161 | 301 | 353 | 317 | 246 | 153 | 17 |
| 2 | 0 | 208 | 317 | 335 | 374 | 251 | 144 | 9 |
| 3 | 0 | 25 | 173 | 29 | 16 | 6 | 4 | 0 |
| 4 | 0 | 6 | 208 | 77 | 11 | 19 | 32 | 4 |
| 5 | 0 | 24 | 146 | 497 | 491 | 115 | 82 | 3 |
| 6 | 0 | 11 | 104 | 355 | 915 | 946 | 447 | 30 |
| 7 | 0 | 0 | 3 | 102 | 463 | 811 | 588 | 40 |
| 8 | 0 | 0 | 2 | 22 | 201 | 786 | 901 | 960 |

Overall Accuracy = (4+208+173+77+ 491+ 946+588+960) / 14845 = 21.40 %

# Prudential Life Insurance

## Logistic Regression

I decided to add a binomial output variable called 'Approved' that is set to 1 when Response = 8. Otherwise Approved = **0.** I could predict 'Approved' using **Logistic Regression** to have an idea of which independent variables are significant.

# Prudential Life Insurance

## Logistic Regression

trainFit = glm(Approved ~ Product_Info_1 + Product_Info_2 + Product_Info_4 + Product_Info_5 + InsuredInfo_2 + InsuredInfo_4 + Medical_Keyword_3 + Medical_Keyword_12 + Medical_Keyword_15 + Medical_Keyword_22 + Ht + Wt + Ins_Age, data = trainLog, family=binomial)

**Which Threshold to pick?**

After calculating the Sensitivity and Specificity with Threshold 0.7, 0.5, and 0.2, I would pick a Threshold of 0.2 because I would like to have a high True Positive Rate while having a low False Positive Error Rate.

# Prudential Life Insurance

**Logistic Regression**

<u>Making predictions</u>

predictTest = predict(trainFit, type="response",  newdata = testLog)

ConfusionMatrix <- table(testLog$Approved,predictTest > **0.2**)

ConfusionMatrix

| | FALSE | TRUE |
|---|---|---|
| 0 | 7563 | 2410 |
| 1 | 554 | 4318 |

# Prudential Life Insurance

**Logistic Regression**

Evaluating the model

Accuracy = 0.80%

False Positive Error Rate = 0.2416525

True Positive Rate = 0.886289

# Prudential Life Insurance

Classification Tree

- Accuracy = 47.77%
- Tree that was easy to read in 7 splits for the variables: BMI, Medical_History_4, Medical_History_1, Medical_History_2 and Product_Info_4.

Random Forest

- Accuracy = 21.40 %.
- 200 tress with 42 splits, mean squared residuals of 3.49
- The top 8 variables with more importance for the model were: BMI, Wt, Medical_History_23, Medical_Keyword_3, Product_Info_2, Product_Info_4, Ins_age, Medical_History_4.

- The model with the highest accuracy was obtain using Logistic Regression after creating a binary outcome variable (Approved) to be predicted instead of a nominal (Response), giving us an overall accuracy of 80%. There were around 40 significant coefficients that probably could have been reduced to improve the AIC.