Data Analytics

Predicting Medical Appointment cancelation

Final Project Presentation

Prof. Yang Yang
By: Divya Damahe

Agenda

- Problem description
- About the data
- Preprocessing the data
- Models
- Conclusion

Affect of missed appointments

Doctors lose their valuable time every time a patient decides to default.

Affects people who could have been given the appointment instead of that patient.

According to a survey, nearly 42 percent of patients skip their appointments.

Source- https://www.annfammed.org/content/2/6/541.full/

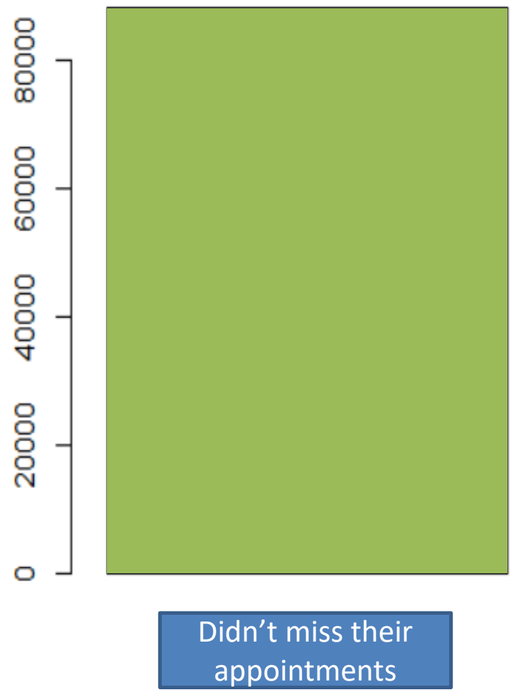
Let’s predict the appointments with no show

About the data

110,527 medical appointments its 14 associated variables

Reduced the data to ~8k rows

01 - PatientId
02 - AppointmentID
03 - Gender
04 – Schedule Date
05 – Appointment Date
06 - Age
07 - Neighbourhood
08 - Scholarship
09 - Hipertension
10 - Diabetes
11- Alcoholism
12- Handicap
13- SMS_received
14- No-show



Near 20% people missed their appointments

Data Preprocessing

- **Solved the issue of data skewness**

  No Yes
  4411 4464

- **Created a column with age from birth year**

- **Resolved the date column using as.date() function**

  “2016-04-27T07:51:14Z”

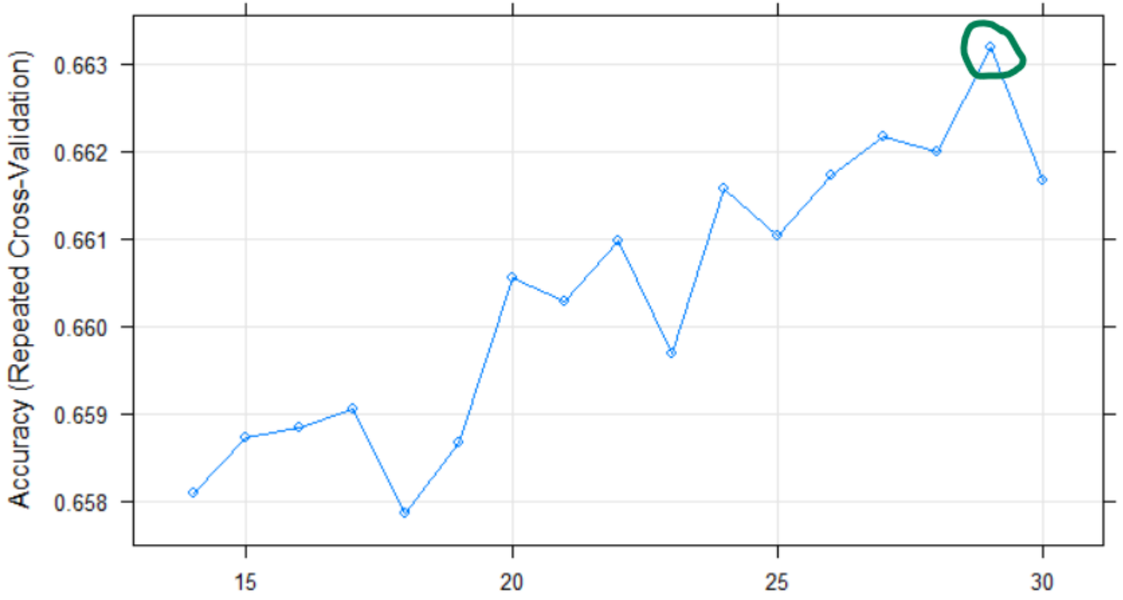
- **Removed irrelevant rows**

  Appointment id, patient id, Neighbourhood

KNN

```
k-Nearest Neighbors

6213 samples
   9 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (4 fold, repeated 5 times)
Summary of sample sizes: 4660, 4660, 4660, 4659, 4660, 4659, .
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.6445848  0.2889321
  7  0.6485123  0.2967028
  9  0.6497031  0.2989774

Accuracy was used to select the optimal model using the larges
The final value used for the model was k = 9.
```

```
modelbest_knn <- train(No.show~.,data = train,tuneGrid =
data.frame(k=14:30),method = "knn", trControl = trainControl(method =
"repeatedcv",number = 5,repeats = 3))

plot(modelbest_knn)
```




```
k-Nearest Neighbors

6213 samples
   9 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 4970, 4970, 4971, 4970, 4971, 4970, ...
Resampling results:

  Accuracy   Kappa
  0.6623745  0.3239874

Tuning parameter 'k' was held constant at a value of 29
```

Generalized linear model

```{r}
model1_lg<-glm(as.factor(No.show)~.,data = train, family = binomial(link="logit") )
summary(model1_lg)

```

```
Call:
glm(formula = as.factor(No.show) ~ ., family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0267  -1.0444   0.4927   1.1190   1.6334

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.330223   0.064712  -5.103 3.34e-07 ***
Age          -0.005509   0.001378  -3.999 6.37e-05 ***
GenderM       0.064161   0.056478   1.136   0.2559
Scholarship   0.207536   0.088348   2.349   0.0188 *
Hipertension -0.073114   0.085391  -0.856   0.3919
Diabetes      0.263066   0.115876   2.270   0.0232 *
Alcoholism    0.093994   0.160778   0.585   0.5588
Handcap      -0.090144   0.168726  -0.534   0.5932
SMS_received  0.497493   0.058291   8.535  < 2e-16 ***
c             0.027207   0.002043  13.319  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8612.8  on 6212  degrees of freedom
Residual deviance: 8173.8  on 6203  degrees of freedom
AIC: 8193.8

Number of Fisher Scoring iterations: 4
```

```{r}
model1_lg<-glm(as.factor(N
summary(model1_lg)

```

```
Call:
glm(formula = as.factor(N
    family = binomial(lin

Deviance Residuals:
    Min       1Q   Median
-3.0142  -1.0481   0.4943

Coefficients:
              Estimate St
(Intercept)  -0.260707
Age          -0.006407
c             0.026960
SMS_received  0.496232
Diabetes      0.224200
---
Signif. codes:  0 ‘***’ 0

(Dispersion parameter for

    Null deviance: 8612.8
Residual deviance: 8181.6
AIC: 8191.6

Number of Fisher Scoring iterations: 4
```

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  896 565
       Yes 427 774

               Accuracy : 0.6273
                 95% CI : (0.6087, 0.6458)
    No Information Rate : 0.503
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2551

 Mcnemar's Test P-Value : 1.363e-05

            Sensitivity : 0.5780
            Specificity : 0.6772
         Pos Pred Value : 0.6445
         Neg Pred Value : 0.6133
             Prevalence : 0.5030
         Detection Rate : 0.2908
   Detection Prevalence : 0.4512
      Balanced Accuracy : 0.6276

       'Positive' Class : Yes
```

Linear SVM

```
modelbest_svmLin <- train(No.show~.,data = train,method = "svmLinear",trControl=trainControl(method = "cv",number = 2),tuneGrid =
expand.grid(C = seq(1,2,0.1)))
```

```
Support Vector Machines with Linear Kernel

6213 samples
   9 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (2 fold)
Summary of sample sizes: 3106, 3107
Resampling results across tuning parameters:

  C    Accuracy   Kappa
  1.0  0.6043796  0.2095248
  1.1  0.6038966  0.2085668
  1.2  0.6043795  0.2095269
  1.3  0.6045405  0.2098451
  1.4  0.6040576  0.2088893
  1.5  0.6037357  0.2082556
  1.6  0.6045405  0.2098447
  1.7  0.6043796  0.2095280
  1.8  0.6045406  0.2098359
  1.9  0.6050234  0.2108051
  2.0  0.6045405  0.2098475

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was C = 1.9.
```

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  912 665
       Yes 411 674

               Accuracy : 0.5958
                 95% CI : (0.5769, 0.6145)
    No Information Rate : 0.503
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1925

 Mcnemar's Test P-Value : 1.23e-14

            Sensitivity : 0.5034
            Specificity : 0.6893
         Pos Pred Value : 0.6212
         Neg Pred Value : 0.5783
             Prevalence : 0.5030
         Detection Rate : 0.2532
   Detection Prevalence : 0.4076
      Balanced Accuracy : 0.5964

       'Positive' Class : Yes
```

Random Forest

```
model1_rf<-train(No.show~.,data = train,method = "rf",tuneGrid = expand.grid(mtry =seq(2,4,2)))
```

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No   615  171
       Yes  708 1168

               Accuracy : 0.6698
                 95% CI : (0.6516, 0.6877)
    No Information Rate : 0.503
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.338

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8723
            Specificity : 0.4649
         Pos Pred Value : 0.6226
         Neg Pred Value : 0.7824
             Prevalence : 0.5030
         Detection Rate : 0.4388
   Detection Prevalence : 0.7047
      Balanced Accuracy : 0.6686

       'Positive' Class : Yes
```

Decision Tree

```
#Decision tree
grid <- expand.grid(.M=c(2,3,4,5,6,7,8,9,10),

.C=c(0.01,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45))
optimal_model <- train(No.show~ .,
                       data=train,
                       method="J48",
                       trControl = trainControl(method = "cv",number =
3),tuneGrid = grid)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  1325  244
       Yes 1763 2881

               Accuracy : 0.677
                 95% CI : (0.6652, 0.6886)
    No Information Rate : 0.503
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.352

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4291
            Specificity : 0.9219
         Pos Pred Value : 0.8445
         Neg Pred Value : 0.6204
             Prevalence : 0.4970
         Detection Rate : 0.2133
   Detection Prevalence : 0.2525
      Balanced Accuracy : 0.6755

       'Positive' Class : No
```

Other models

- Naïve bayes
- ANN

```{r}
library(neuralnet)
ann<-neuralnet(formula = No.show~Scholarship+Diabetes,data = train,hidden = c(3,4),linear.output = F)
predict_ann<-predict(ann,test)
predict_ann
```

```
          [,1]      [,2]
1    0.5022147 0.4977937
11   0.5022147 0.4977937
13   0.5022147 0.4977937
16   0.5022147 0.4977937
17   0.5022147 0.4977937
18   0.5022147 0.4977937
20   0.4510587 0.5489320
23   0.5022147 0.4977937
24   0.5022147 0.4977937
26   0.5022147 0.4977937
33   0.4510587 0.5489320
34   0.5022147 0.4977937
35   0.5022147 0.4977937
40   0.5022147 0.4977937
```

Conclusion

Models	KNN	GLM	Linear SVM	Random forest	Decision tree
Accuracy	Reference Prediction No Yes No 776 393 Yes 547 946	Reference Prediction No Yes No 896 565 Yes 427 774	Reference Prediction No Yes No 897 615 Yes 426 724	Reference Prediction No Yes No 615 171 Yes 708 1168	Reference Prediction No Yes No 1325 244 Yes 1763 2881

Data Analytics Project Proposal

Thank you