Advanced Analytics with R

Generalized Linear Model

Variety of Regression Analysis

Type of regression	Typical use
Simple linear	Predicting a quantitative dependent variable from a quantitative independent variable.
Polynomial	Predicting a quantitative dependent variable from a quantitative independent variable, where the relationship is modelled as an *n*th order polynomial.
Multiple linear	Predicting a quantitative dependent variable from two or more independent variables.
Multilevel	Predicting a dependent variable from data that have a hierarchical structure.
Logistic	Predicting a categorical dependent variable from one or more independent variables.
Poisson	Predicting a dependent variable representing counts from one or more independent variables.
Cox proportional hazards	Predicting time to an event (death, failure, goal) from one or more independent variables
Time series	Modeling time-series data with correlated errors.
Nonlinear	Predicting a quantitative dependent variable from one or more independent variables, where the form of the model is nonlinear.
Nonparametric	Predicting a quantitative dependent variable from one or more independent variables, where thr form of model is derived from the data and not specified a priori.
Robust	Predicting a quantitative dependent variable from one or more independent variables using an approach that is resistent to the effect of influential observations.

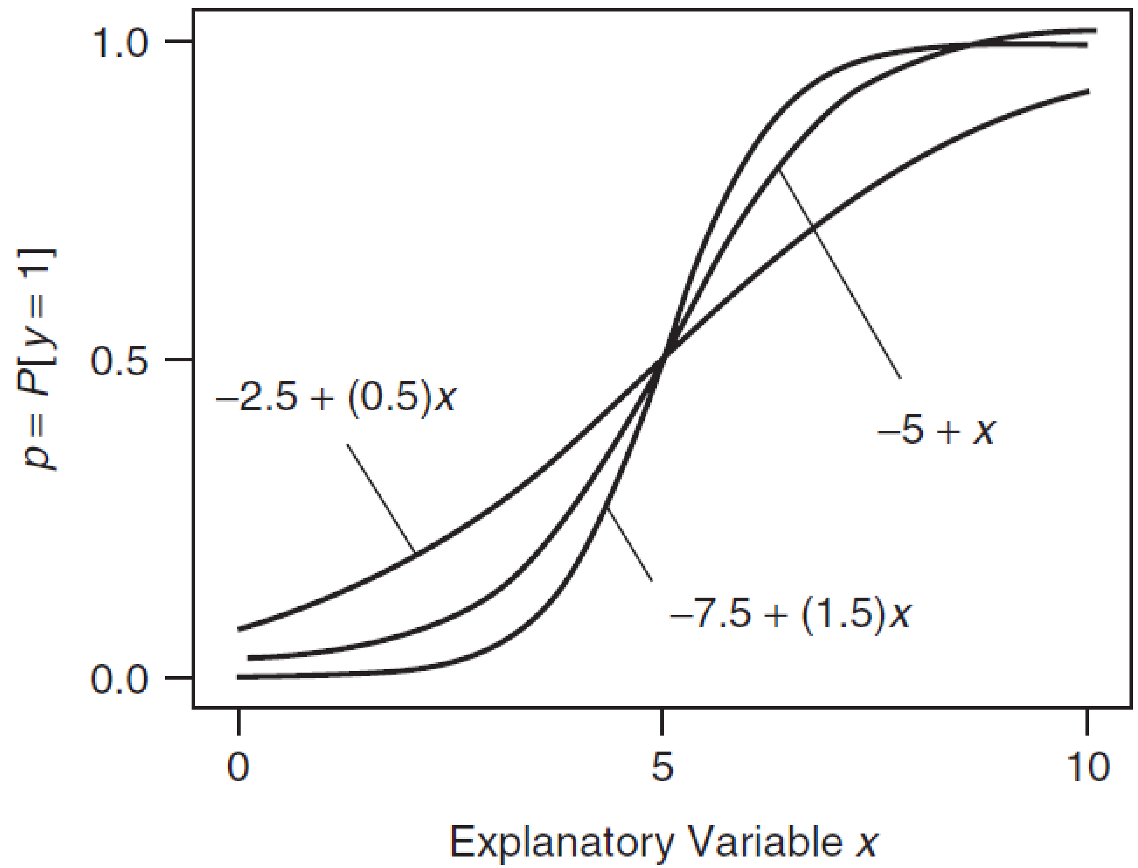
Logistic Regression

Predicting Binary Outcome

- Profit or loss
- Pass or rejected
- Win or lose
- Buy or not buy
- Default or not
- …

Idea Behind Logistics

$$p = f(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k) = \frac{\exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}{1 + \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}$$



$$p = f(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k) = \frac{\exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}{1 + \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)}$$

$$\longrightarrow \quad \log \frac{p}{1-p} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

$\frac{p}{1-p}$ relates the probability of success, *p*, with the probability of failure, *1-p*, and is referred to as the ***odd of success***.

$log\left(\frac{p}{1-p}\right)$ is called the ***logit of p***.

Lasagna Triers

Case 1

Prediction Error with Naïve Method

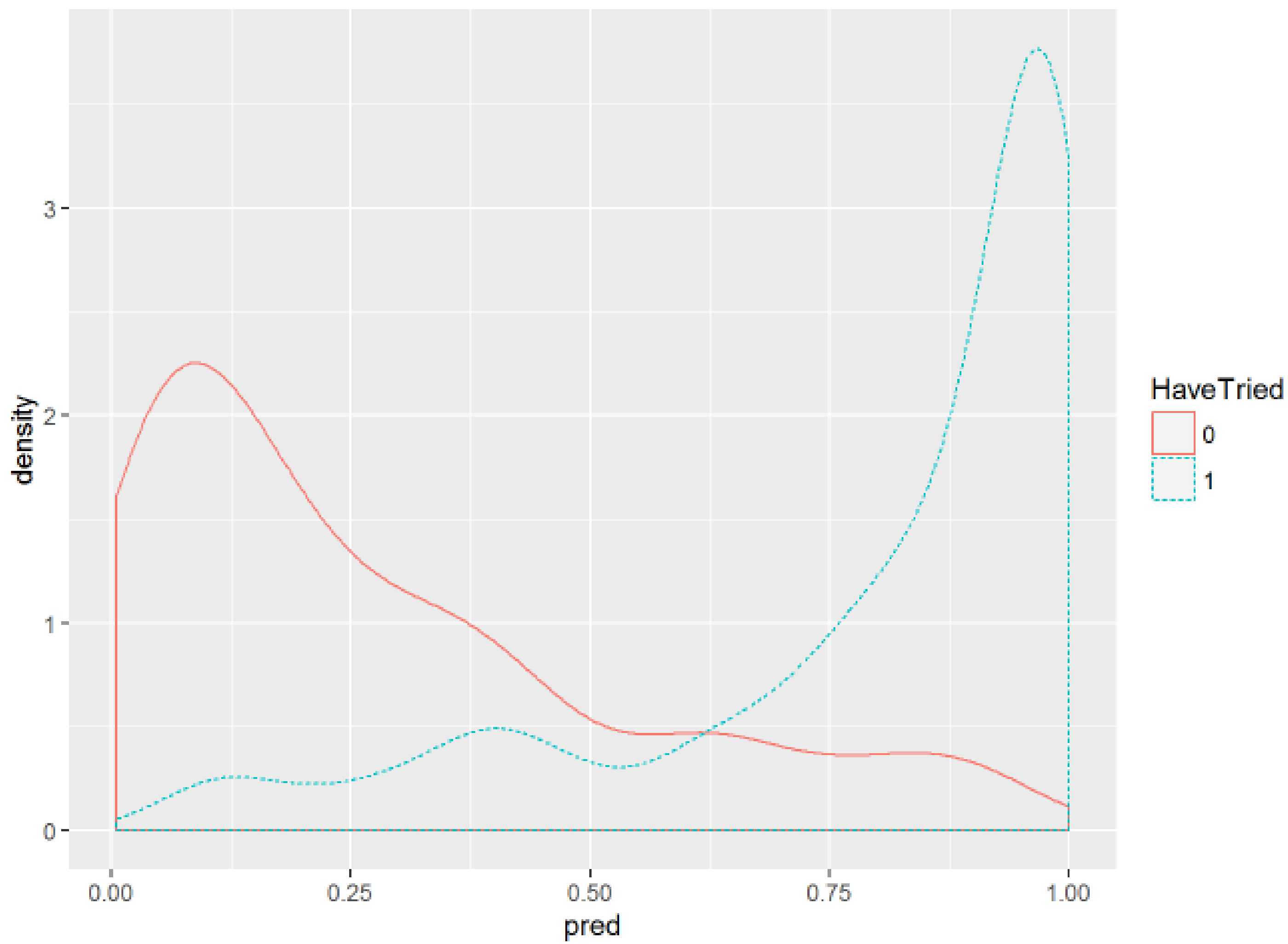
Variable	Classification Method	Error
Age	Age < 41 as Triers	0.335
Weight	All people as triers	0.422
Income	All people as triers	0.422
CarValue	All people as triers	0.422
CCDebt	All people as triers	0.422
MallTrips	All people as triers	0.422
PayType	“Salaried” as Triers	0.341
Gender	“Male” as Triers	0.461
LiveAlone	“Yes” as Triers	0.495
DwellType	Not “Condo” as Triers	0.451
Nbhd	Not “East” as Triers	0.315

Model

```
fit.full <- glm(HaveTried ~ ., data=triers.train, family=binomial())
```

```
fit.simplified <- glm(HaveTried ~ Age  + PayType  + LiveAlone  + MallTrips + Nbhd, data=triers.train,
family=binomial())
```

Prediction Output



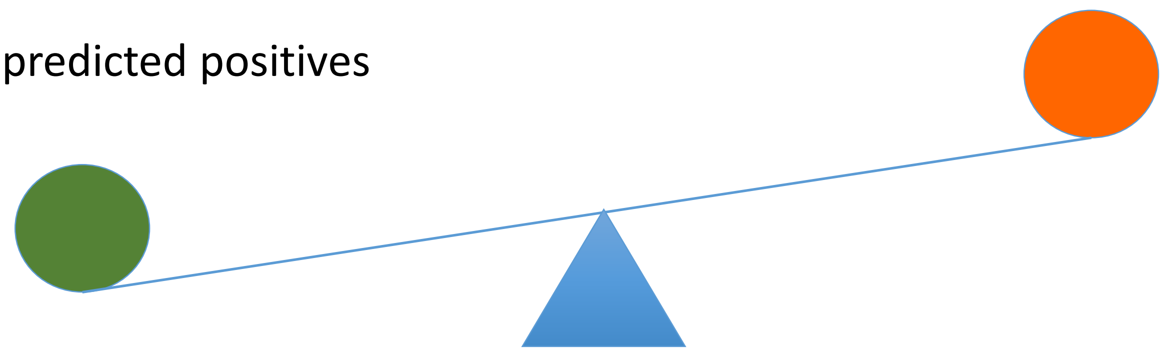
Identifying classifier

Recall

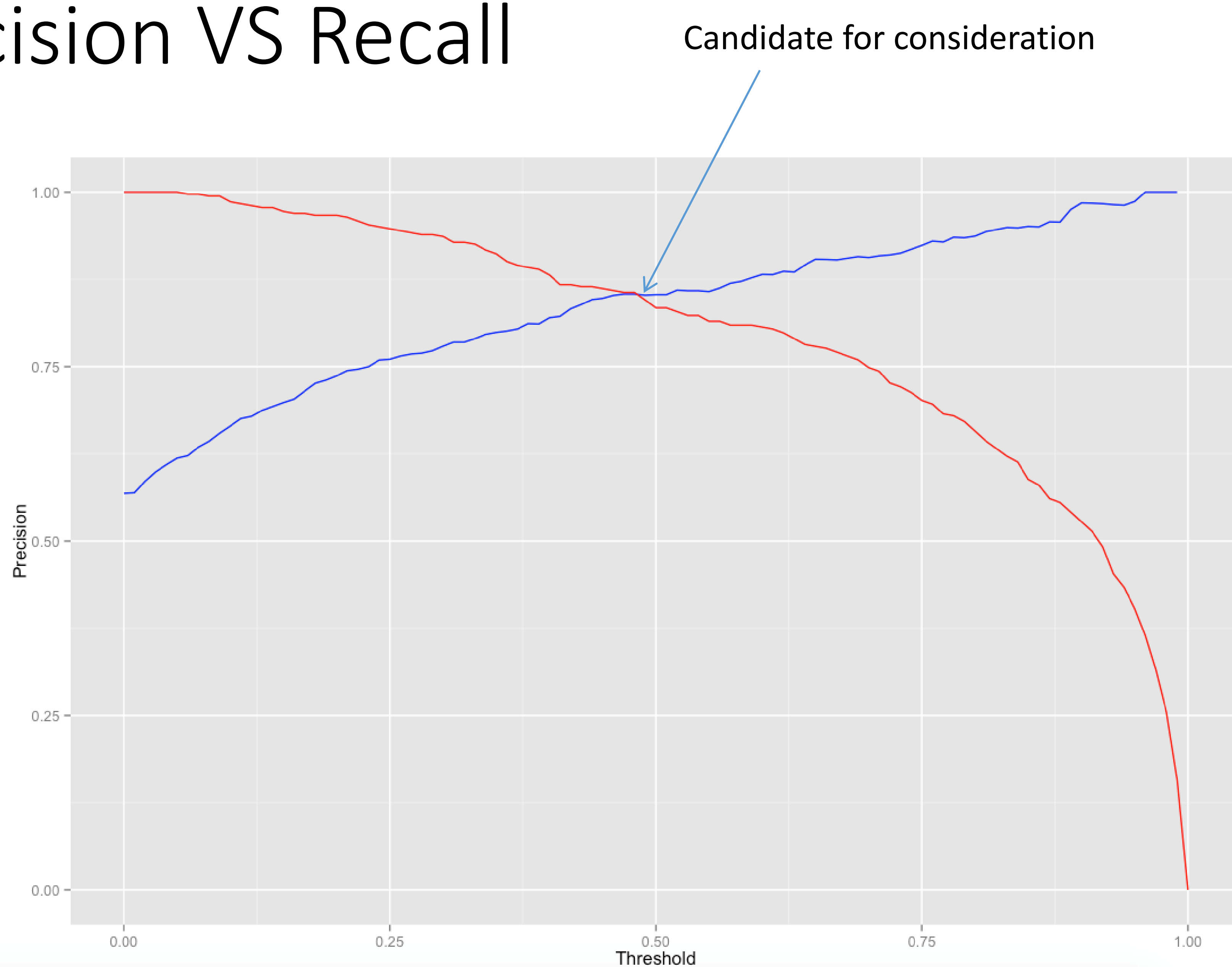
The fraction of the true positives identified by the predictor

Precision

The fraction of the predicted positives are true positives



Precision VS Recall



Validate on test data set

```
> triers.test$pred <- predict(fit.simplified,newdata = triers.test,type="response")
> ctab.test <- table(pred=triers.test$pred > 0.5, triers=triers.test$HaveTried)
> ctab.test
       triers
pred     No Yes
  FALSE  75  21
  TRUE   11 112
```

Interpretation of Coefficients

$$\log \frac{p}{1-p} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

1 unit increase of x_1 $\Longrightarrow$ β_1 unit increase of $log\left(\frac{p}{1-p}\right)$

The odds of success are increased by the multiplicative factor e^{β_1}

```
coefficients(fit.simplified)
  (Intercept)             Age PayTypeSalaried      GenderMale    LiveAloneYes       MallTrips       NbhdSouth        NbhdWest 
  -2.58143033     -0.05726402      1.52158506      0.20488389      1.11791224      0.68603195      0.85136814      2.19828384 
```

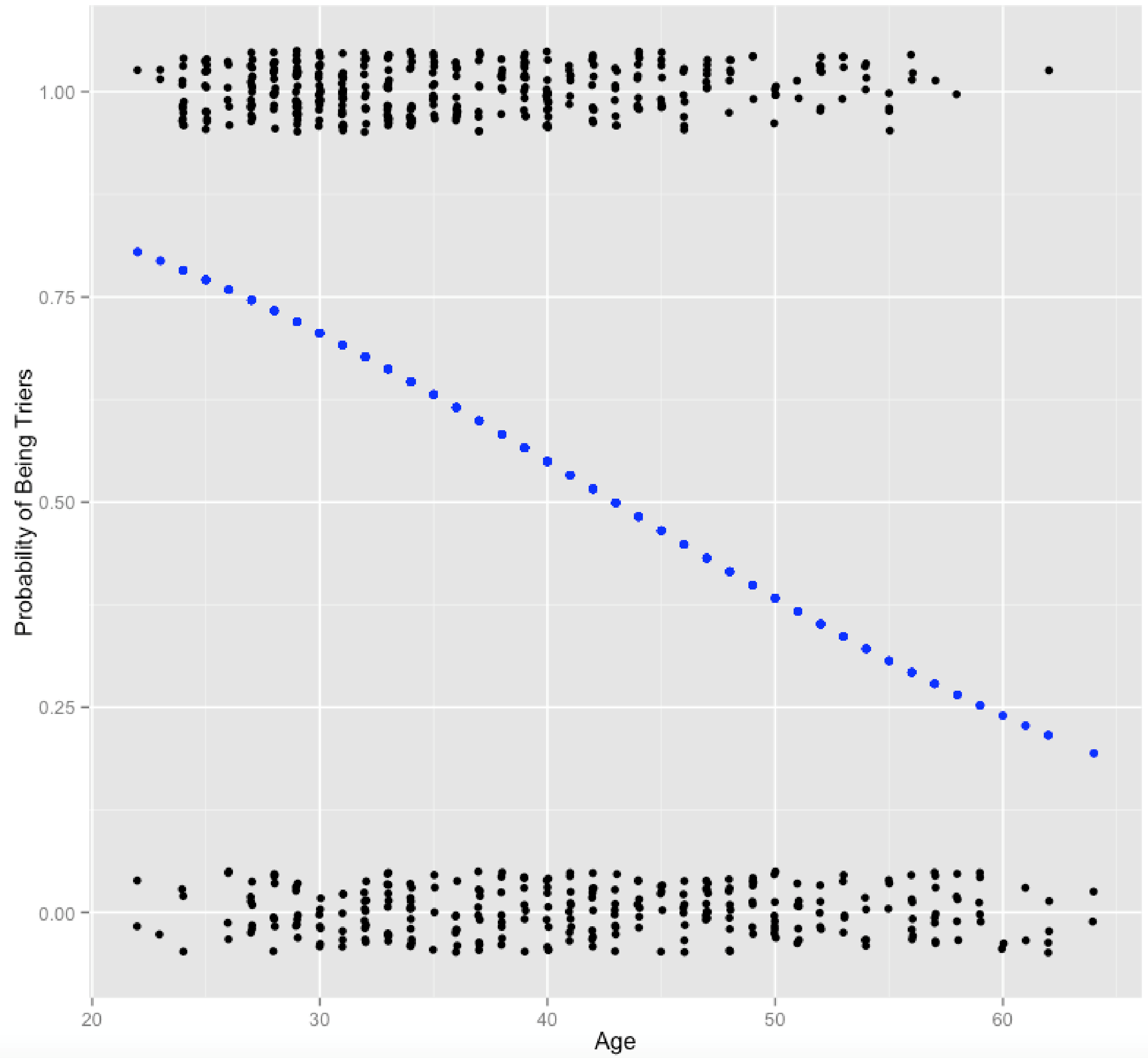
```
exp(coef(fit.simplified))
  (Intercept)             Age PayTypeSalaried      GenderMale    LiveAloneYes       MallTrips       NbhdSouth        NbhdWest 
    0.0756657       0.9443447       4.5794782       1.2273825       3.0584622       1.9858201       2.3428500       9.0095384 
```

- Every one year older, the odds of customers being lasagna trier will decrease by a factor of exp(-0.05726402) = 94.4%.

Evaluating impact of one variable

```
> testdata<-data.frame(Age=rep(30,3),Weight=mean(triers$Weight),PayType=rep("Hourly",3),CarValue=mean(triers$CarValue),CCDebt
=mean(triers$CCDebt),Gender="Male",LiveAlone="No",DwellType="Home",MallTrips=4,Nbhd=c("East","South","West"))
> testdata
  Age   Weight PayType CarValue   CCDebt Gender LiveAlone DwellType MallTrips  Nbhd
1  30 192.6612  Hourly 5908.481 1431.203   Male        No      Home         4  East
2  30 192.6612  Hourly 5908.481 1431.203   Male        No      Home         4 South
3  30 192.6612  Hourly 5908.481 1431.203   Male        No      Home         4  West
> testdata$prob<-predict(fit.simplified,newdata = testdata,type="response")
> testdata$prob
[1] 0.2058149 0.3777825 0.7001358
```

Visualize the impact of one variable



Systematic Way of Evaluating Performance

- https://hopstat.wordpress.com/2014/12/19/a-small-introduction-to-the-rocr-package/

Confusion Matrix

- A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Definition

- **Accuracy:** Overall, how often is the classifier correct?
- **Misclassification Rate:** Overall, how often is it wrong?
- **True Positive Rate (**also known as **Recall** or **Sensitivity):** When it's actually yes, how often does it predict yes?
- **False Positive Rate:** When it's actually no, how often does it predict yes?
- **True Negative Rate (**also known as **Specificity):** When it's actually no, how often does it predict no?
- **Positive Predicted Rate (**also know as **Precision):** When it predicts yes, how often is it correct?
- **Prevalence:** How often does the yes condition actually occur in our sample?

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Accuracy = (50+100)/165

Misclassification Rate = (5 + 10)/165

True Positive Rate = 100 /(100+5)

False Positive Rate = 10/60

True Negative Rate = 50/60

Positive Predicted Rate = 100/110

Prevalence = (100+5)/165

ROC curve

- A ***Receiver Operating Characteristics*** curve or ROC curve, is a graphical plot that illustrates the performance of a ***binary classifier*** system as its discrimination threshold is varied.
- The curve is created by plotting the ***true positive rate*** (TPR) against the ***false positive rate*** (FPR) at various threshold settings.

From wikipedia

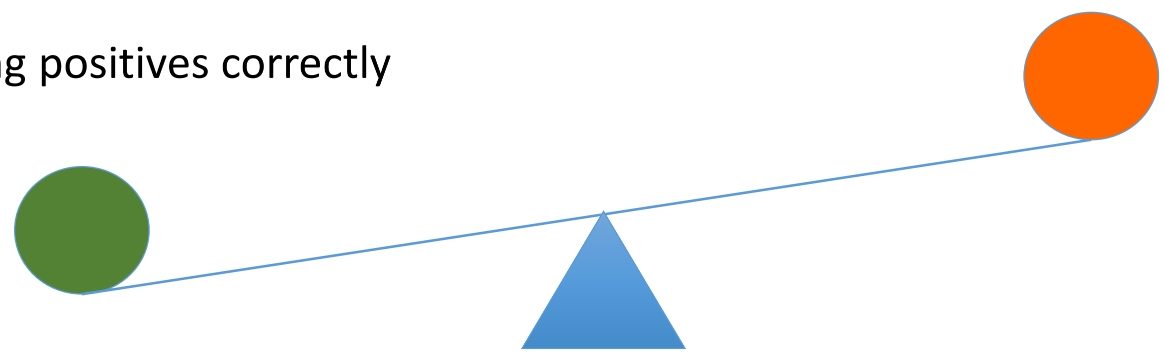
Identifying classifier

FPR

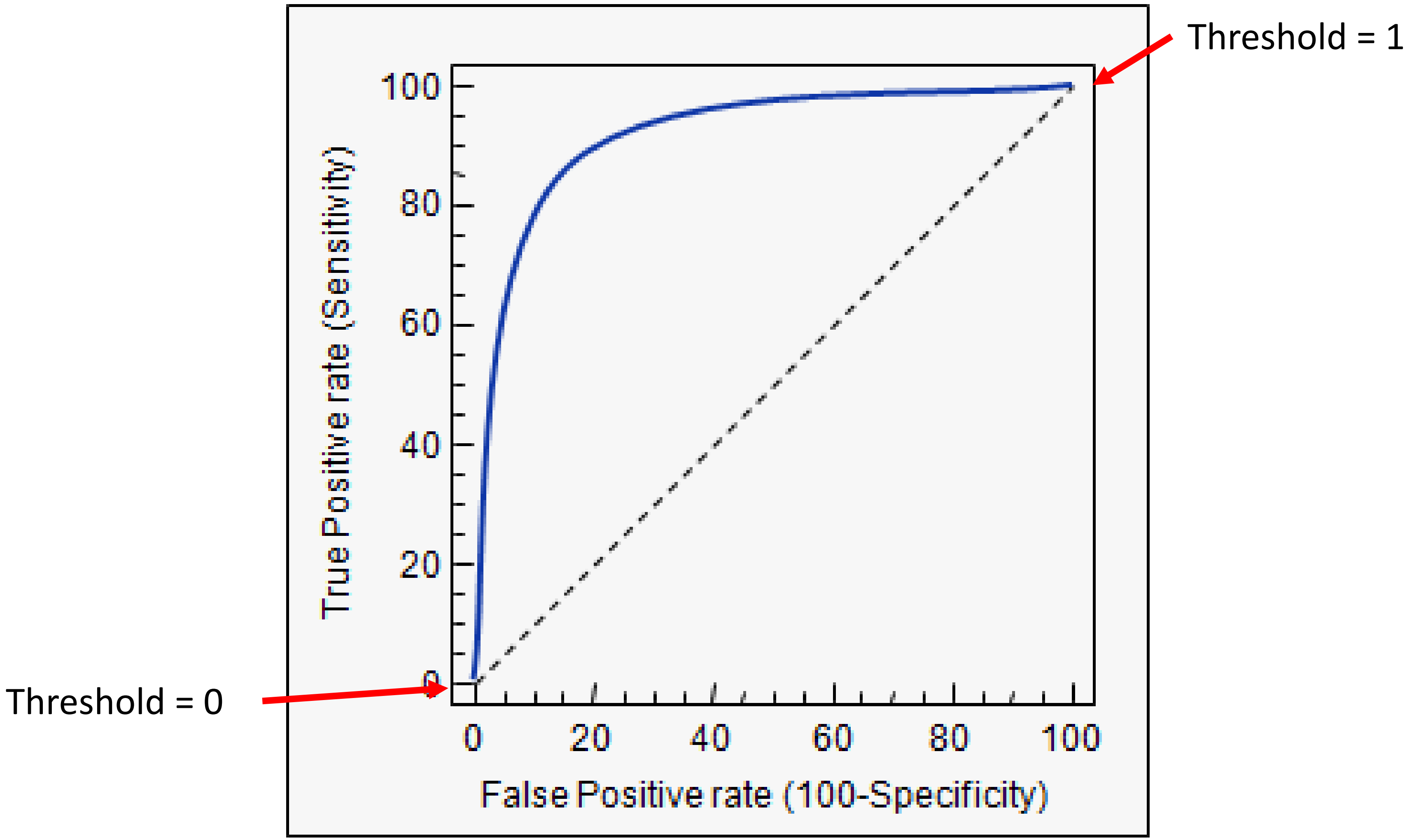
Cost of predicting positives (false alarm) wrongly

TPR

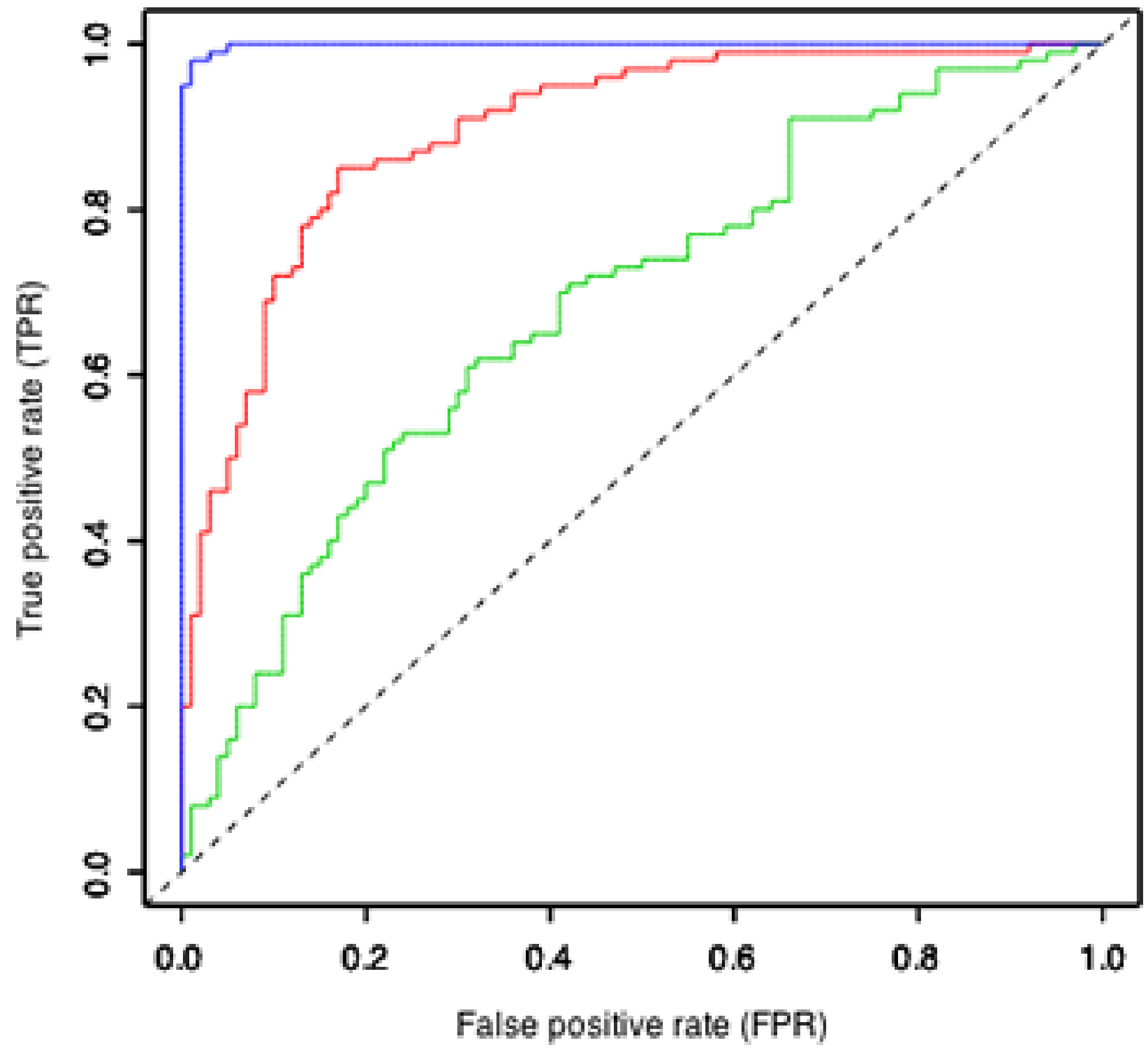
Benefit of predicting positives correctly



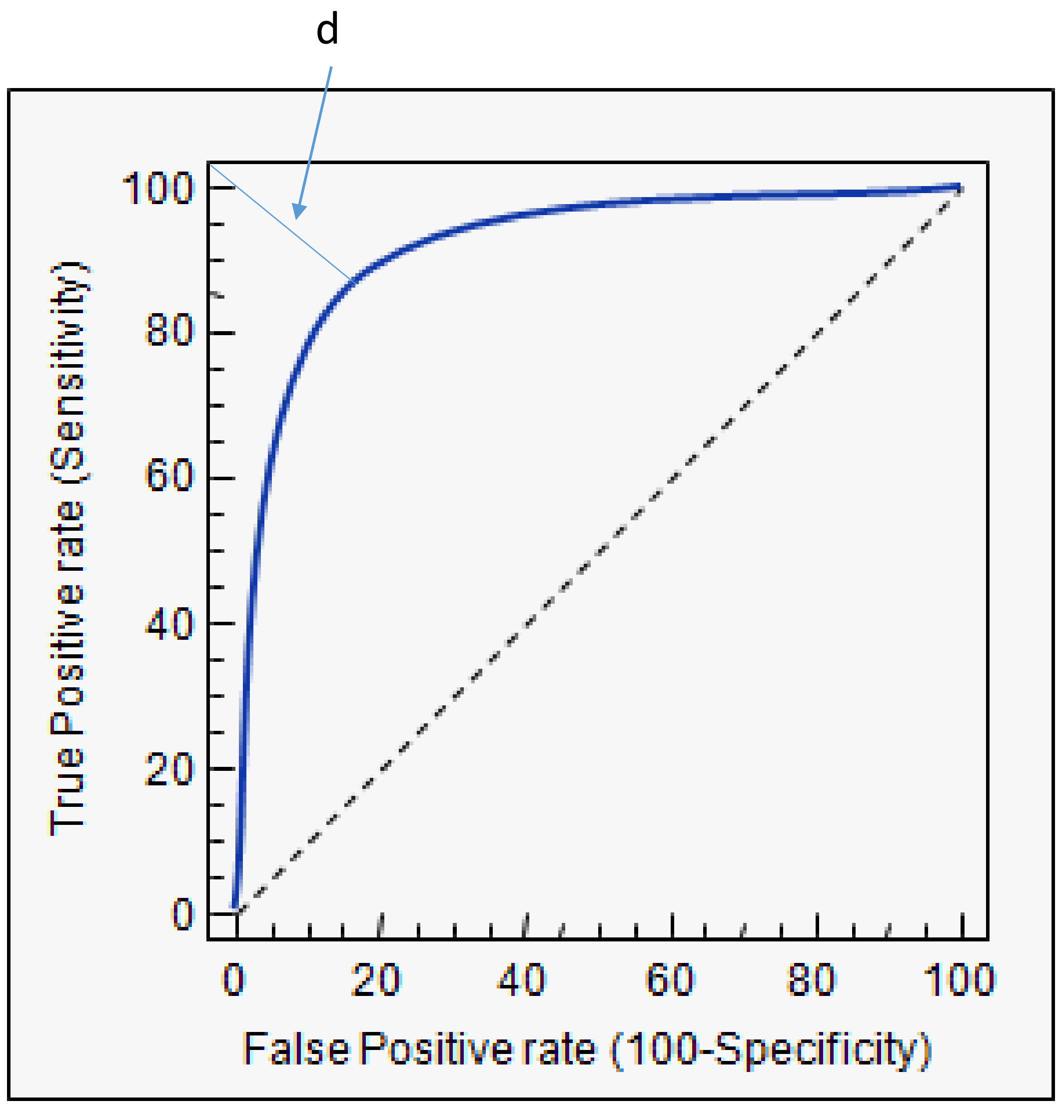
ROC Curve


Threshold = 1
Threshold = 0
100
80
60
40
20
0
0
20
40
60
80
100
True Positive rate (Sensitivity)
False Positive rate (100-Specificity)

Which model is better?

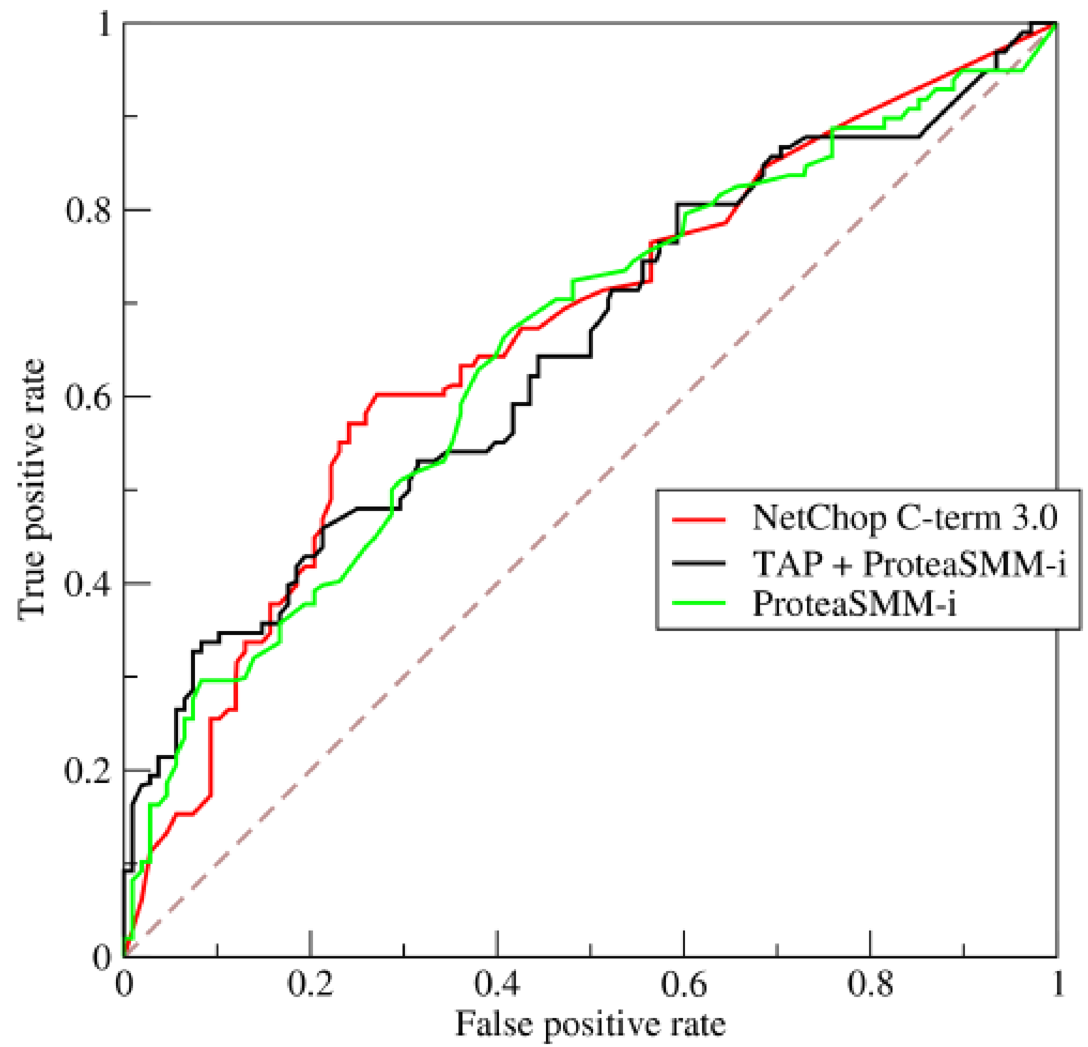


Which threshold is the best?

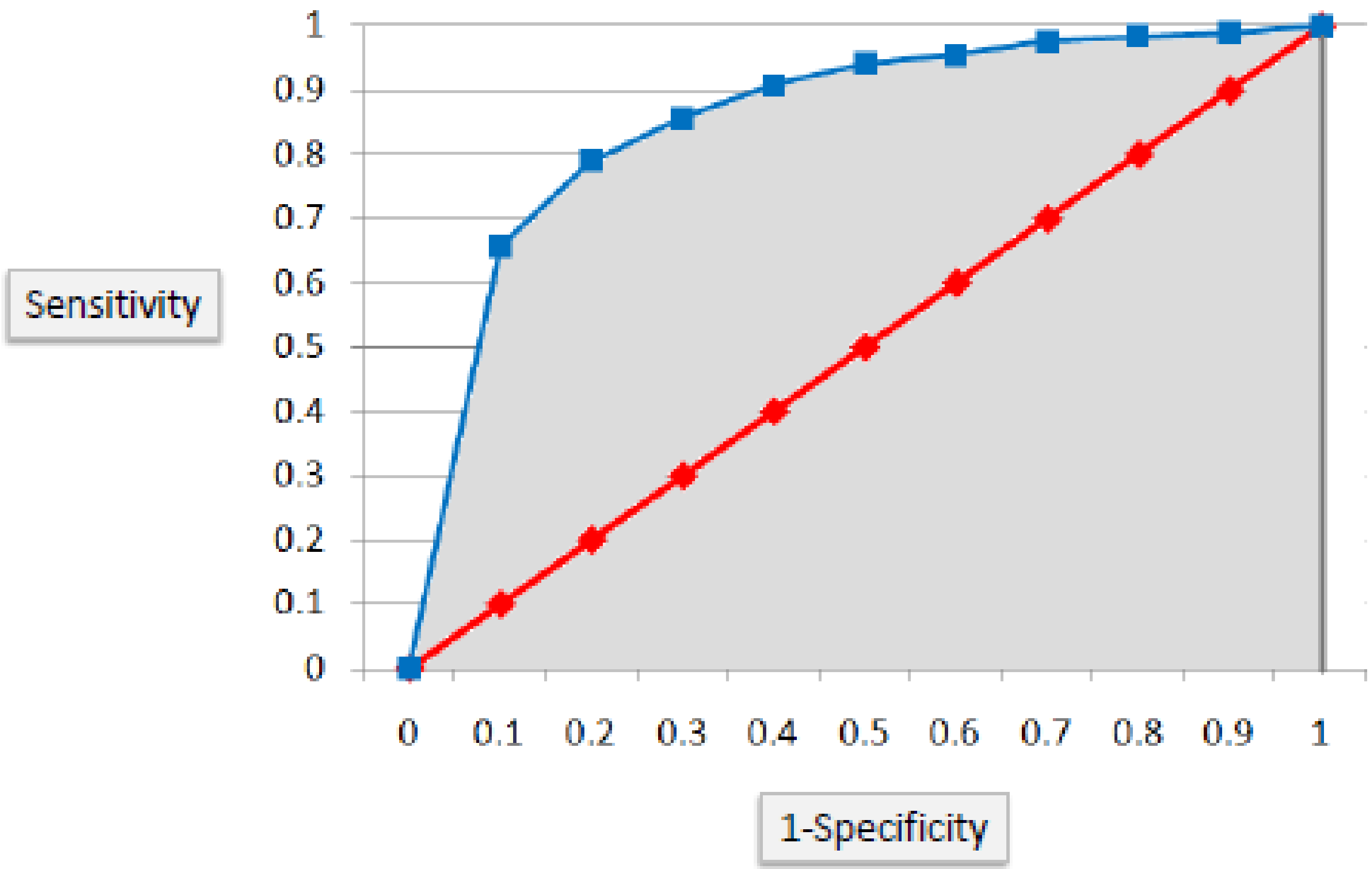


Choose the threshold that gives shortest distance to (0,1)

Which model is better?



Area Under an ROC Curve (AUC)



The bigger the AUC, the better the model