MIS 64036: Business Analytics

Lecture VII

Rouzbeh Razavi, PhD

Agenda

- Classification
- Logistic Regression Single Variable Models
- R Example
- Logistic Regression Multiple Variable Models
- Variable Importance
- Classification Error Types and Performance Metrics

Agenda

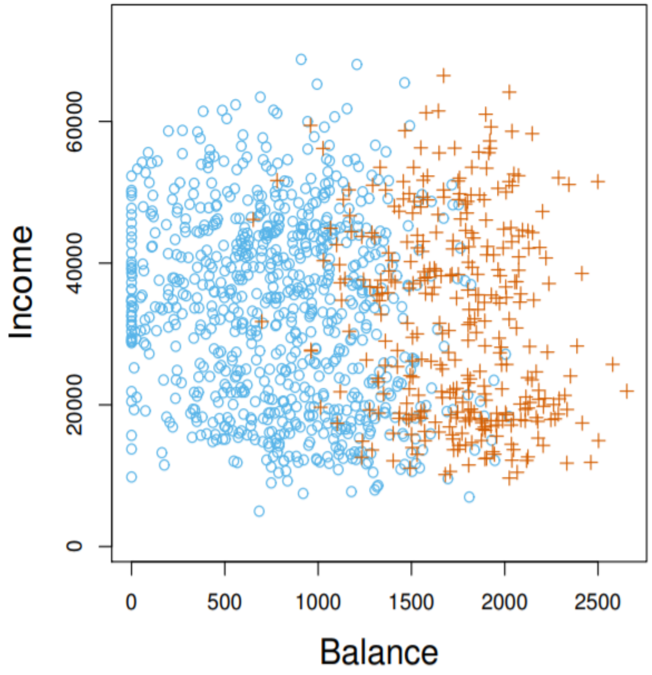
- **Classification**
- Logistic Regression Single Variable Models
- R Example
- Logistic Regression Multiple Variable Models
- Variable Importance
- Classification Error Types and Performance Metrics

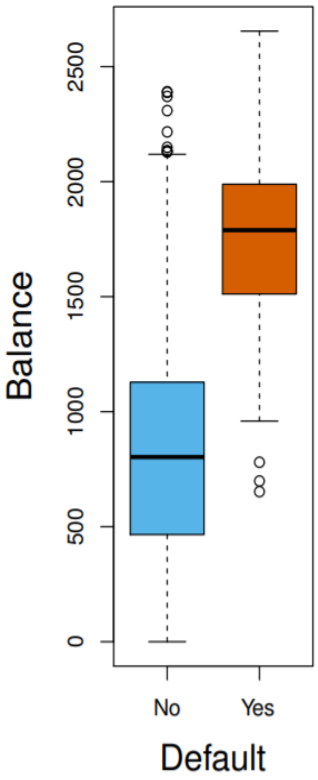
Classification

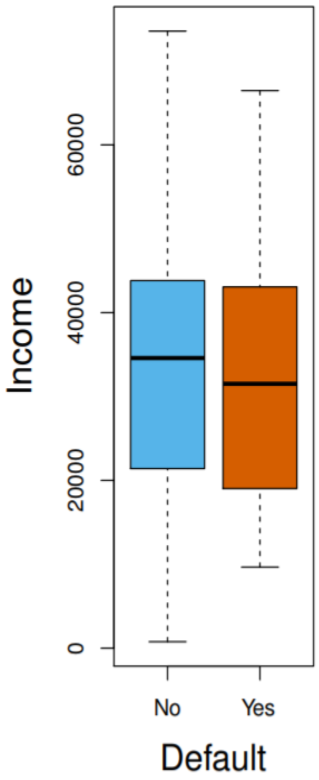
- Qualitative variables take values in an unordered set $\mathcal{C}$, such as:
  `eye color` $\in$ {`brown`, `blue`, `green`}
  `email` $\in$ {`spam`, `ham`}.
- Given a feature vector X and a qualitative response Y taking values in the set $\mathcal{C}$, the classification task is to build a function $C(X)$ that takes as input the feature vector X and predicts its value for Y; i.e. $C(X) \in \mathcal{C}$.
- Often we are more interested in estimating the *probabilities* that X belongs to each category in $\mathcal{C}$.

  For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not.

Example: Credit Card Default







Can we use Linear Regression?

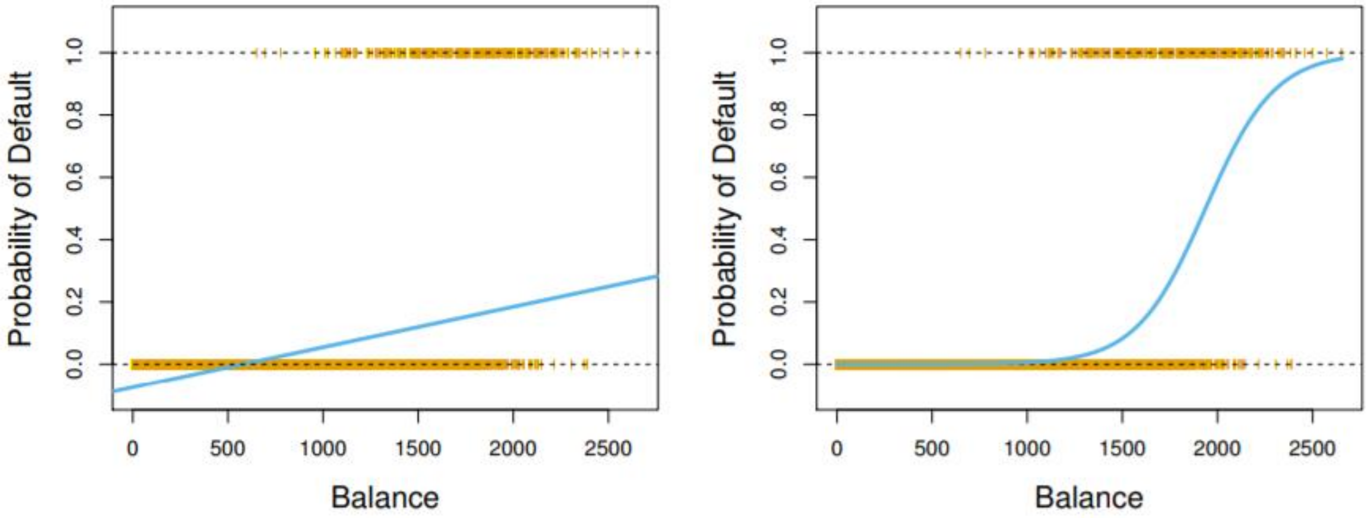
Suppose for the `Default` classification task that we code

$$Y = \begin{cases} 0 & \text{if } \texttt{No} \\ 1 & \text{if } \texttt{Yes}. \end{cases}$$

Can we simply perform a linear regression of Y on X and classify as `Yes` if $\hat{Y} > 0.5$?

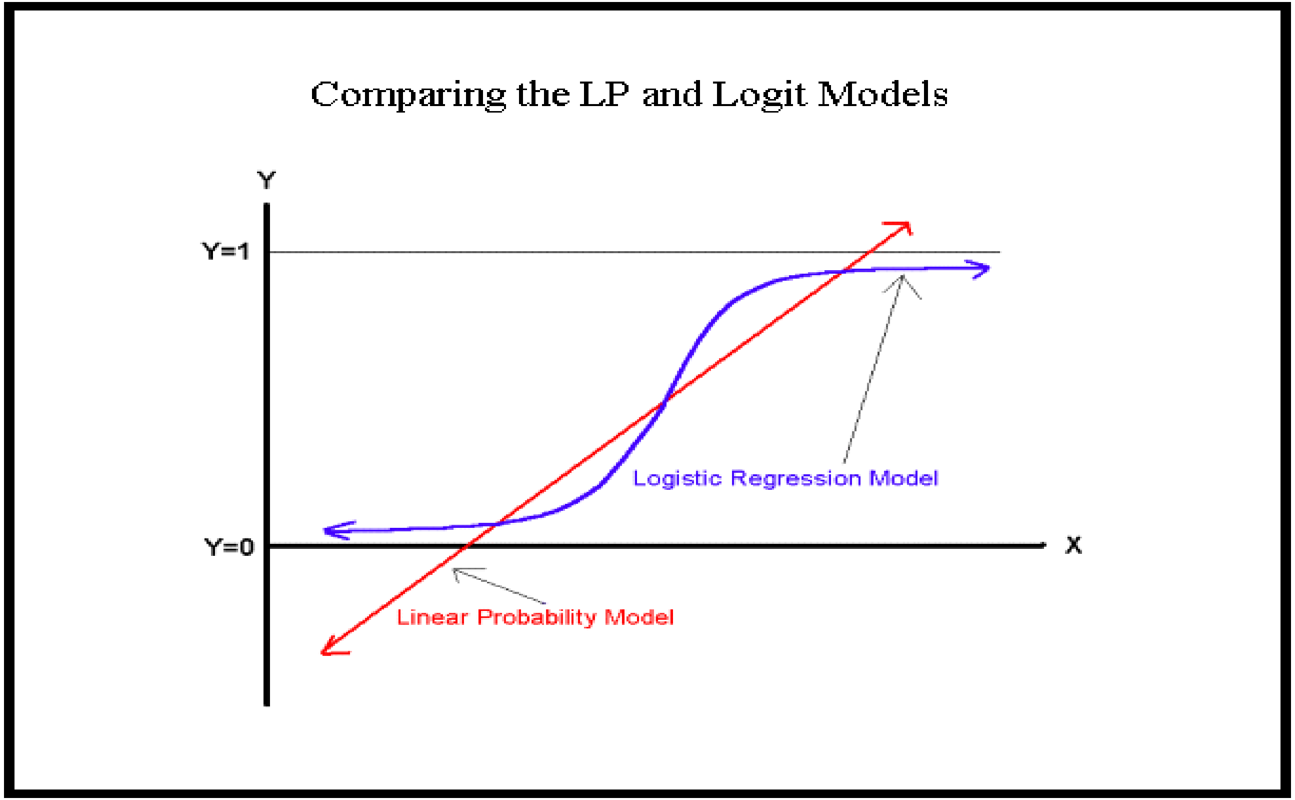
- *linear* regression might produce probabilities less than zero or bigger than one. *Logistic regression* is more appropriate.

Linear versus Logistic Regression



The orange marks indicate the response Y, either 0 or 1. Linear regression does not estimate $\Pr(Y = 1|X)$ well. Logistic regression seems well suited to the task.

Linear versus Logistic Regression



Agenda

- Classification
- **Logistic Regression Single Variable Models**
- R Example
- Logistic Regression Multiple Variable Models
- Variable Importance
- Classification Error Types and Performance Metrics

Logistic Regression

Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using `balance` to predict `default`. Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

($e \approx 2.71828$ is a mathematical constant [Euler's number.])
It is easy to see that no matter what values β_0, β_1 or X take, $p(X)$ will have values between 0 and 1.

A bit of rearrangement gives

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X.$$

This monotone transformation is called the *log odds* or *logit* transformation of $p(X)$.

From Probability to Odds

- If event A has probability P(A), then the odds in favor of A are P(A) to 1-P(A). It follows that the odds against A are 1-P(A) to P(A). The odds ratio will be P(A)/(1-P(A))

- If the probability of an earthquake in California is 0.25, then the odds in favor of an earthquake are 0.25 to 0.75 or 1 to 3.

- The coefficient, β_1, in the logistic regression can be interpreted as follows. For every unit increase in X, the logarithm of the odd ratios of the Y increases by β_1.

Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data.

Maximum Likelihood

- The formula says use P if Y=1 and 1-P if Y=0 and calculate the product of all terms.

- The likelihood, L, is obviously highest when, P, is closer to 1 for observations where the outcome (Y) is 1 and is closer to 0 for observations where the outcome is 0.

- Calculate the likelihood for the following vectors of P1 and P2 for the given vector of outcomes Y. You can see that the Likelihood is much higher for P2.

Y=c(1,1,0,0,1)

P1=c(0.6,0.75,0.35,0.6,0.9) P2=c(0.85,0.95,0.15,0.1,0.95)

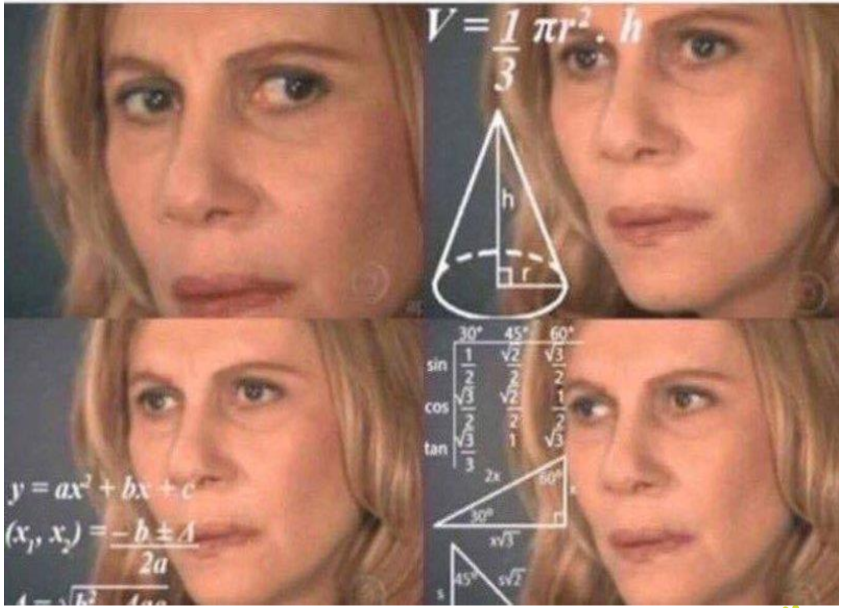
L=0.6*0.75*(1-0.35)*(1-0.6)*0.9=0.1053 L=0.85*0.95*(1-0.15)*0.95=0.6520562

Maximum Likelihood: Coefficients Calculation

- How can the regression coefficients are estimated using the maximum likelihood estimation method?

- Complex and beyond the scope!

We rely on “R” !



Maximum Likelihood Credit Card Default Example

Most statistical packages can fit linear logistic regression models by maximum likelihood. In R we use the glm function.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Making Predictions

What is our estimated probability of `default` for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

With a balance of \$2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

Making Predictions

Lets do it again, using student as the predictor.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Agenda

- Classification
- Logistic Regression Single Variable Models
- **R Example**
- Logistic Regression Multiple Variable Models
- Variable Importance
- Classification Error Types and Performance Metrics

R Example

```
library(ISLR)
summary(Default)
Model=glm(default~balance, family="binomial",data=Default)
Model
predict(Model, data.frame(balance =c(1000,2000)),  type = "response")
predict(Model, data.frame(balance =c(1000,2000)),  type = "response")
```

R Example

```
> library(ISLR)
> summary(Default)
 default     student       balance           income     
 No :9667   No :7056   Min.   :   0.0   Min.   :  772  
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340  
                       Median : 823.6   Median :34553  
                       Mean   : 835.4   Mean   :33517  
                       3rd Qu.:1166.3   3rd Qu.:43808  
                       Max.   :2654.3   Max.   :73554  
> Model=glm(default~balance, family="binomial",data=Default)
> Model

Call:  glm(formula = default ~ balance, family = "binomial", data = Default)

Coefficients:
(Intercept)      balance  
 -10.651331     0.005499  

Degrees of Freedom: 9999 Total (i.e. Null);  9998 Residual
Null Deviance:      2921 
Residual Deviance: 1596 	AIC: 1600
> predict(Model, data.frame(balance =c(1000,2000)),  type = "response")
          1           2 
0.005752145 0.585769370 
```

Agenda

- Classification
- Logistic Regression Single Variable Models
- R Example
- **Logistic Regression Multiple Variable Models**
- Variable Importance
- Classification Error Types and Performance Metrics

Logistic Regression with Several Variables

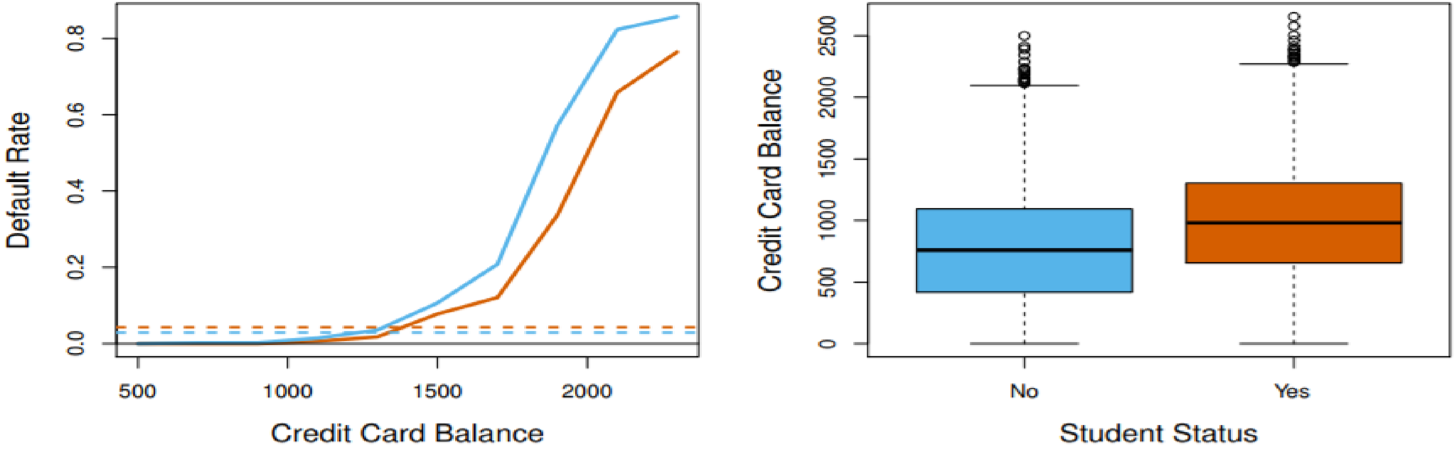
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

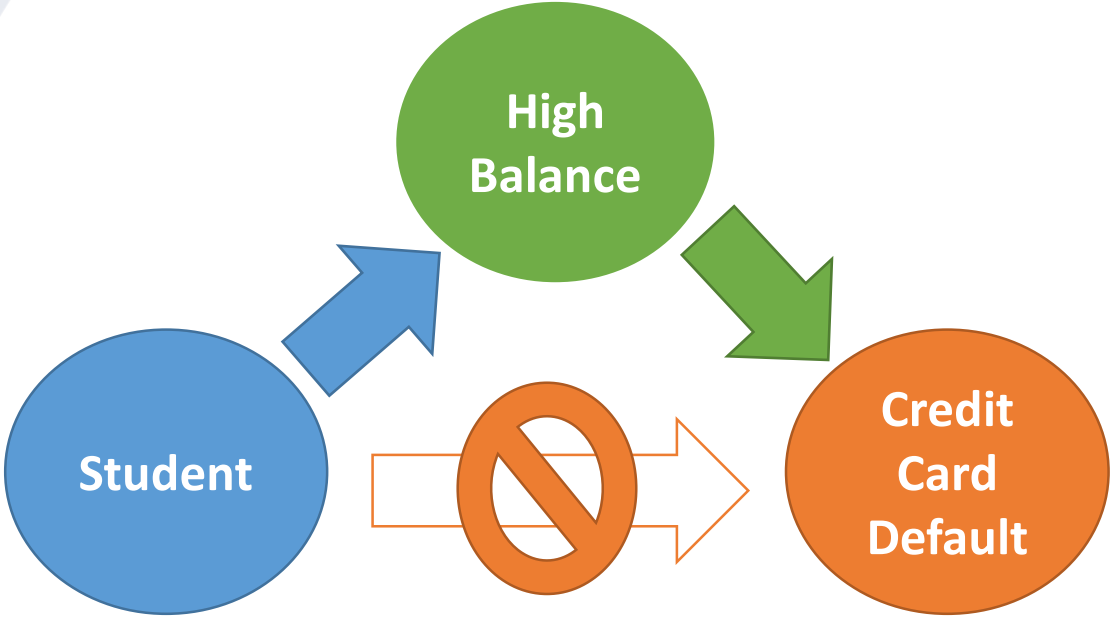
Why is coefficient for student negative, while it was positive before?

Confounding



- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- Multiple logistic regression can tease this out.

Confounding



Let's Make Some Money!

We want to predict if a given stock will go up or down in a given week, based on the data from the stock performance in previous 5 weeks (lag1 to lag5) and trade volumes.

```
library(ISLR)
summary(Weekly)
```

`Year` The year that the observation was recorded

`Lag1` Percentage return for previous week

`Lag2` Percentage return for 2 weeks previous

`Lag3` Percentage return for 3 weeks previous

`Lag4` Percentage return for 4 weeks previous

`Lag5` Percentage return for 5 weeks previous

`Volume` Volume of shares traded (average number of daily shares traded in billions)

`Today` Percentage return for this week

`Direction` A factor with levels `Down` and `Up` indicating whether the market had a positive or negative return on a given week

Example: Let's Make Some Money!

We want to predict if a given stock will go up or down in a given week, based on the data from the stock performance in previous 5 weeks (lag1 to lag5) and trade volumes.

```
library(ISLR)
levels(Weekly$Direction)
Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
family="binomial", data=Weekly)
Test_Data=data.frame(Lag1=0.7, Lag2= 3.1,Lag3=-2.5,Lag4= -0.1,
 Lag5= 0.816, Volume= 0.1537280)
predict(Model, newdata=Test_Data,type='response')
```

Example: Let's Make Some Money!

```
> library(ISLR)
> levels(Weekly$Direction)
[1] "Down" "Up"
> Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
+             family="binomial", data=Weekly)
> Test_Data=data.frame(Lag1=0.7, Lag2= 3.1,Lag3=-2.5,Lag4= -0.1,
+                        Lag5= 0.816, Volume= 0.1537280)
> predict(Model, newdata=Test_Data,type='response')
        1
0.6098714
```

The probability that the "Direction" of the market for the Test_Data is "**UP**" in the week. More specifically, the return value of the predict function is always the probobility that the outcome is **NOT** the first level of the outcome "**Down**" in our example (i.e. is the probability of "**UP**")

Agenda

- Classification
- Logistic Regression Single Variable Models
- R Example
- Logistic Regression Multiple Variable Models
- **Variable Importance**
- Classification Error Types and Performance Metrics

Variable Importance

When building a model, you may want to be able to evaluate the relative importance of different variables in your model.

For linear models you can use the **absolute** value of the **t-statistics** or **z-statistics** for each model parameter as a measure of variable importance (ignore the Intercept).

Variable Importance

```
> summary(Model)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = "binomial", data = Weekly)

Deviance Residuals:
    Min       1Q    Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
```

Lag2 is the most important variable

Lag5 is the least important variable

Agenda

- Classification
- Logistic Regression Single Variable Models
- R Example
- Logistic Regression Multiple Variable Models
- Variable Importance
- **Classification Error Types and Performance Metrics**

Model Performance

Confusion Matrix tells us how many of the predictions are correct. Example, Credit Card Default Example:

		True Default Status		
		No	Yes	Total
Predicted	No	9644	252	9896
Default Status	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$ errors — a 2.75% misclassification rate!

Examples: Two-class problems

- Good or bad?
- Present or absent?
- Diseased or not?
- Up or down?

truth	classifier	evaluation
+	+	true positive
-	+	false positive
-	-	true negative
+	-	false negative

Examples: Two-class problems

- Given a classifier and an instance:

Classifier / Predicted class	TRUE CLASS	
	p (positive)	n (negative)
Y	True Positives	False Positives
N	False Negatives	True Negatives
Total	P	N

P = True Positives + False Negatives

Types of Errors

TP	FP
FN	TN

$$TPR = \frac{TP}{P} = Recall,\ FPR = \frac{FP}{N}$$

$$Precision = \frac{TP}{TP + FP},\ Accuracy = \frac{TP + TN}{P + N}$$

$$Sensitivity = Recall,\ Specificity = 1 - FPR$$

Types of Errors

False positive rate: The fraction of negative examples that are classified as positive — 0.2% in example.

False negative rate: The fraction of positive examples that are classified as negative — 75.7% in example.

We produced this table by classifying to class Yes if

$$\widehat{\Pr}(\texttt{Default} = \texttt{Yes}|\texttt{Balance}, \texttt{Student}) \geq 0.5$$

We can change the two error rates by changing the threshold from 0.5 to some other value in $[0, 1]$:

$$\widehat{\Pr}(\texttt{Default} = \texttt{Yes}|\texttt{Balance}, \texttt{Student}) \geq \textit{threshold},$$

and vary *threshold*.

Example in R: Stock Example

```
library(ISLR)
Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
family="binomial", data=Weekly)
Predicted_Values<-predict(Model, newdata=Weekly,type='response')
head(Weekly$Direction)
head(Predicted_Values)
Predicted_Values=as.factor(Predicted_Values>0.5) #P>0.5 means UP
head(Predicted_Values)
levels(Predicted_Values) <- list( DOWN='FALSE', UP='TRUE') #change levels
table(Predicted=Predicted_Values, True=Weekly$Direction)
```

Example in R: Stock Example

```
> library(ISLR)
> Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
+           family="binomial", data=Weekly)
> Predicted_Values<-predict(Model, newdata=Weekly,type='response')
> head(Weekly$Direction)
[1] Down Down Up   Up   Up   Down
Levels: Down Up
> head(Predicted_Values)
        1         2         3         4         5         6 
0.6086249 0.6010314 0.5875699 0.4816416 0.6169013 0.5684190 
> Predicted_Values=as.factor(Predicted_Values>0.5) #P>0.5 means UP
> head(Predicted_Values)
    1     2     3     4     5     6 
 TRUE  TRUE  TRUE FALSE  TRUE  TRUE 
Levels: FALSE TRUE
> levels(Predicted_Values) <- list( DOWN='FALSE', UP='TRUE') #change levels
> table(Predicted=Predicted_Values, True=Weekly$Direction)
         True
Predicted Down  Up
     DOWN   54  48
     UP    430 557
```

Way too many False Positives! Business implications: You predicted the stock will be up which was not the case. This will cause financial losses!

Missed opportunities! False Negative means that you predicted the stock to go Down but it went Up!

You Make Money Here!

Example in R: Stock Example

- False Positives results in high financial losses. How can we avoid them?

- Well the company can become more conservative and decide to consider a stock as 'UP' only if the probability of 'UP' is greeter than 0.6 instead of 0.5.

- In this way, borderline predictions with probability less than 0.6 will be considered as 'Down'

- To do this, we need to change the comparison threshold from 0.5 to 0.6 (see the updated code in the next slide)

Example in R: Stock Example

```
library(ISLR)
Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
family="binomial", data=Weekly)
Predicted_Values<-predict(Model, newdata=Weekly,type='response')
head(Weekly$Direction)
head(Predicted_Values)
Predicted_Values=as.factor(Predicted_Values>0.6) #P>0.6 means UP
head(Predicted_Values)
levels(Predicted_Values) <- list( DOWN='FALSE', UP='TRUE') #change
levels
table(Predicted=Predicted_Values, True=Weekly$Direction)
```

Example in R: Stock Example

```
> library(ISLR)
> Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
+           family="binomial", data=Weekly)
> Predicted_Values<-predict(Model, newdata=Weekly,type='response')
> head(Weekly$Direction)
[1] Down Down Up   Up   Up   Down
Levels: Down Up
> head(Predicted_Values)
        1         2         3         4         5         6 
0.6086249 0.6010314 0.5875699 0.4816416 0.6169013 0.5684190 
> Predicted_Values=as.factor(Predicted_Values>0.6) #P>0.6 means UP
> head(Predicted_Values)
    1     2     3     4     5     6 
 TRUE  TRUE FALSE FALSE  TRUE FALSE 
Levels: FALSE TRUE
> levels(Predicted_Values) <- list( DOWN='FALSE', UP='TRUE') #change levels
> table(Predicted=Predicted_Values, True=Weekly$Direction)
         True
Predicted Down  Up
     DOWN  433 522
     UP     51  83
```

Significantly lower False Positives!

False Negatives increased significantly

Also less opportunities to invest and make gain.

Example in R: Stock Example

```
library(ISLR)
Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
family="binomial", data=Weekly)
Predicted_Values<-predict(Model, newdata=Weekly,type='response')
head(Weekly$Direction)
head(Predicted_Values)
Predicted_Values=as.factor(Predicted_Values>0.7) #P>0.7 means UP
head(Predicted_Values)
levels(Predicted_Values) <- list( DOWN='FALSE', UP='TRUE') #change
levels
table(Predicted=Predicted_Values, True=Weekly$Direction)
```

Least Squares Analysis: R Code

```
> library(ISLR)
> Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
+           family="binomial", data=Weekly)
> Predicted_Values<-predict(Model, newdata=Weekly,type='response')
> head(Weekly$Direction)
[1] Down Down Up   Up   Up   Down
Levels: Down Up
> head(Predicted_Values)
        1         2         3         4         5         6
0.6086249 0.6010314 0.5875699 0.4816416 0.6169013 0.5684190
> Predicted_Values=as.factor(Predicted_Values>0.7) #P>0.6 means UP
> head(Predicted_Values)
    1     2     3     4     5     6
FALSE FALSE FALSE FALSE FALSE FALSE
Levels: FALSE TRUE
> levels(Predicted_Values) <- list( DOWN='FALSE', UP='TRUE') #change levels
> table(Predicted=Predicted_Values, True=Weekly$Direction)
         True
Predicted Down  Up
     DOWN  483 599
     UP      1   6
>
```

Very conservative approach.
Only 7 stocks were predicted as "UP" but 6 of them were actually "UP"

Example in R: Stock Example

- The role of the threshold is to balance between false positive (Type I errors) and False Negative (Type II) errors.

- What threshold should I choose in my business application? Answer: Very much depends on the nature of the application and sensitivity of the application to different error types. While type II errors can be better tolerated for some cases, the complete opposite might be true for other cases.

Type I versus Type II Errors

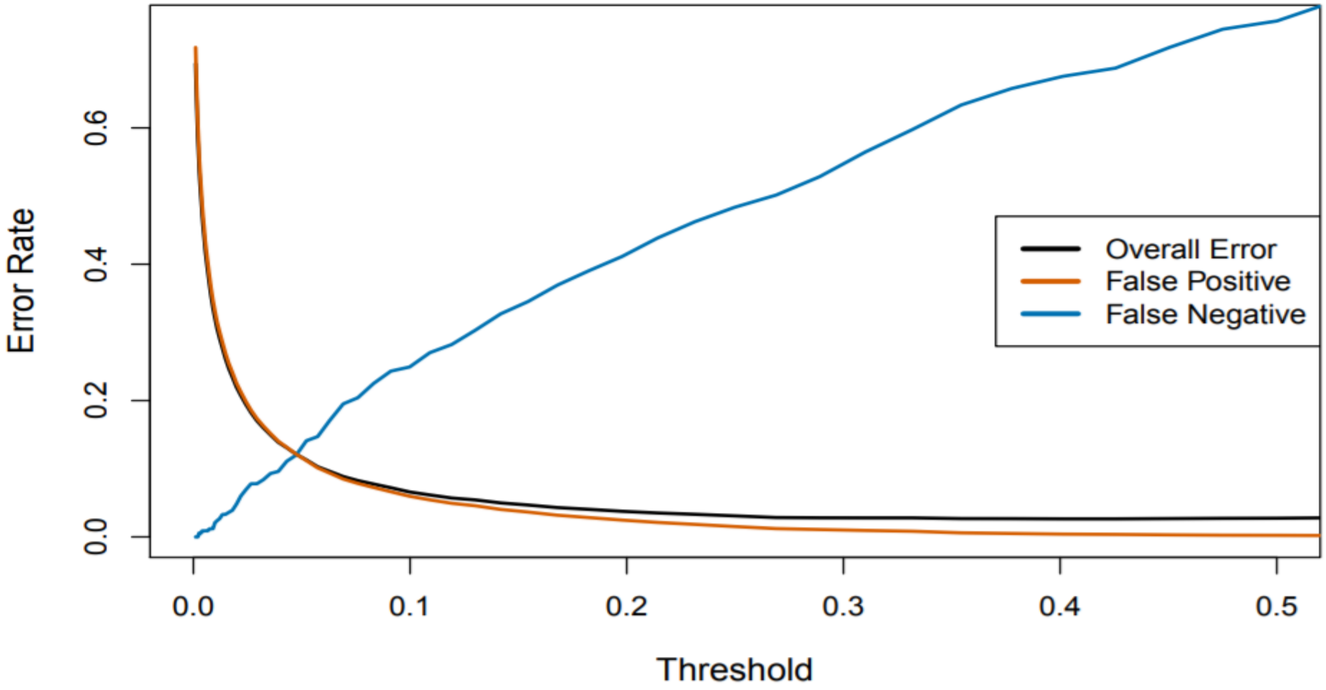
- Consider the following examples and discuss which error types are worse?

- Predicting whether a patent has a cancer? If predictions are positive, further tests will be done.

- Predicting whether an applicant may default on a loan that they have applied for?

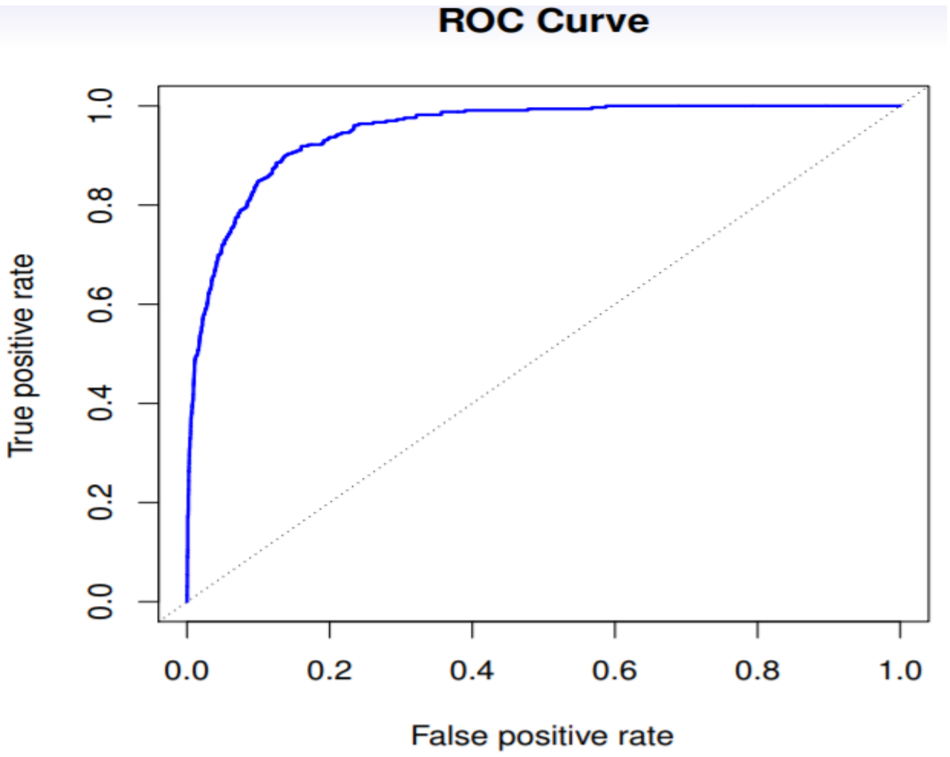
- Predicting whether a candidate is a good match for top position in a top company.

Varying the threshold



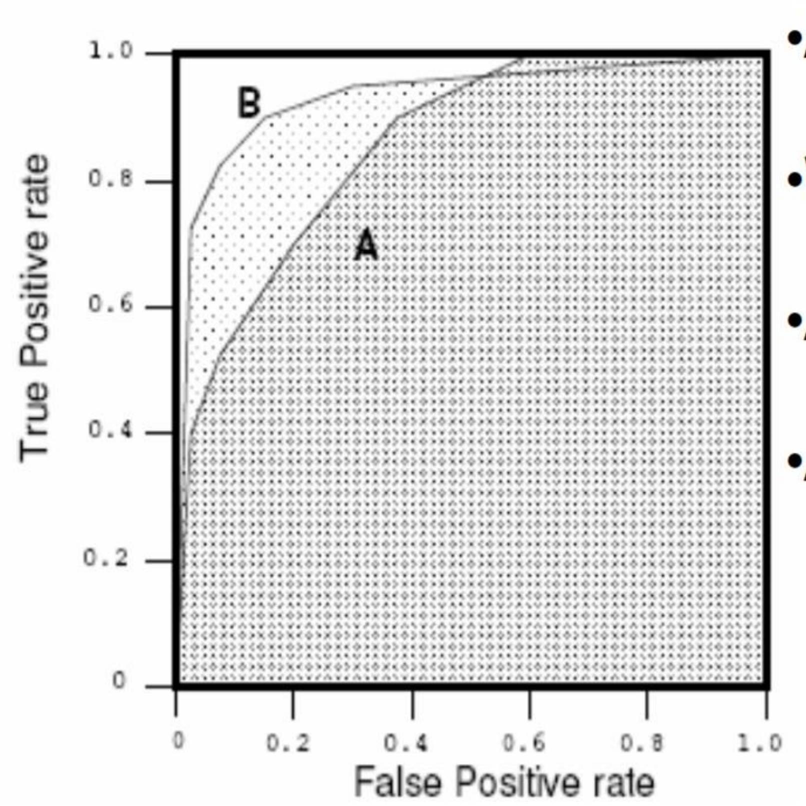
In order to reduce the false negative rate, we may want to reduce the threshold to 0.1 or less.

ROC Curve



The *ROC plot* displays both simultaneously.

ROC Curve



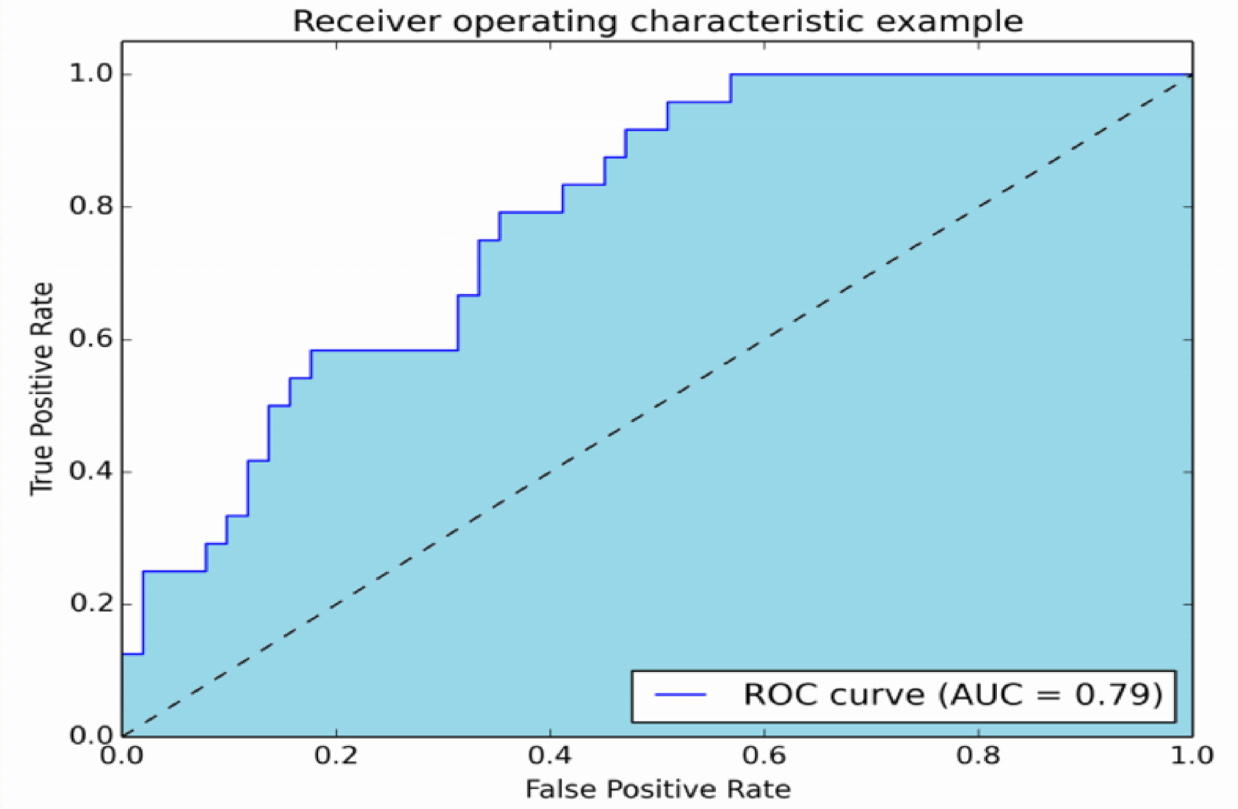
- AUC (Bradley, 1997)
- Wilcoxon test of ranks
- Area : Classifier B > A
- Average performance
  → B > A

Area Under Curve (AUC) of RoC Curve

- The Area Under Curve (AUC) of the RoC is a generic metric (i.e. independent of the selected threshold value) that can be used to compare classifiers.

- The value ranges from 0-1, but a random model should have an AUC of 0.5 so in practice the AUC is higher than 0.5.

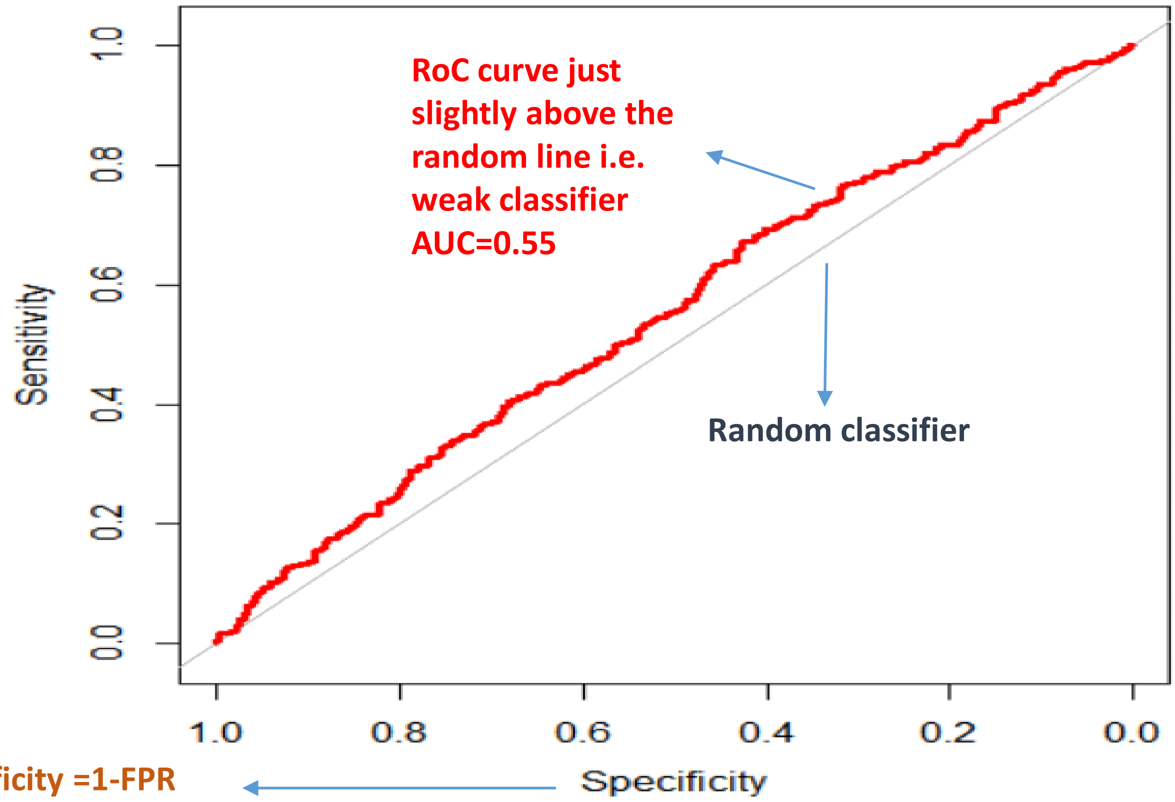
Example: Area Under Curve (AUC) of RoC Curve



Area Under Curve (AUC) of RoC Curve

library(pROC) #you need to install first that is **install.packages(‘pROC’)**
library(ISLR)
Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, family="binomial", data=Weekly)
Predicted_Values<-predict(Model, newdata=Weekly,type='response')
roc(Weekly$Direction, Predicted_Values)
plot(roc(Weekly$Direction, Predicted_Values), col='red', lwd=3)

Area Under Curve (AUC) of RoC Curve



Recall: Sensitivity is the same as TPR

Recall: Specificity =1-FPR

Area Under Curve (AUC) of RoC Curve

```
> library(pROC) #you need to install first that is install.package
> library(ISLR)
> Model=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
+              family="binomial", data=Weekly)
> Predicted_Values<-predict(Model, newdata=Weekly,type='response')
> roc(Weekly$Direction, Predicted_Values)

Call:
roc.default(response = Weekly$Direction, predictor = Predicted_Va

Data: Predicted_Values in 484 controls (Weekly$Direction Down) < (
Area under the curve: 0.5537
> plot(roc(Weekly$Direction, Predicted_Values), col='red', lwd=3)
```

What we covered

- Classification
- Logistic Regression Single Variable Models
- R Example
- Logistic Regression Multiple Variable Models
- Variable Importance
- Classification Error Types and Performance Metrics


ANY
QUESTIONS
?