Day 5: Classification

ME314: Introduction to Data Science and Machine Learning

Jack Blumenau and Kenneth Benoit

15 July 2024

Roadmap

What have we done so far?

1. Working with data
2. Supervised learning – linear regression

Where are we going?

1. Supervised learning – classification
2. Non-linear and tree-based methods
3. Tools for selecting between models
4. Unsupervised learning
5. Text analysis (next week)

Motivation

- Last lecture, we considered ways of predicting quantitative responses using linear regression
- Today, we focus on predicting qualitative responses using a variety of methods
- Many interesting applications require us to classify observations into different groups
    - Will an individual buy, or not buy, a product?
    - Will a business file for bankcruptcy?
    - Will a candidate win an election?
- Our goal is to build models that can make accurate classifications on such tasks

Day 5 Outline

Classification

The Linear Probability Model

Logistic Regression

Multinomial Classification

Characterizing performance of classifiers

Running Example

Who will win this match?

Prediction of sports results is a key application of data science methods. Often, we are interested in qualitative and discrete outcomes – such as whether a given team will win a match – rather than quantitative outcomes. For example, we might be interested in predicting whether the home team of a Premier League football match will win.

- Unit of analysis: 380 Premier League football matches from the 2021-22 season.
- Outcome (Y): `home_win`, equal to 1 if the home team won the match, and 0 otherwise
- Predictors (X): The current position of the home and away teams in the league; the number of red cards received by each team; etc

Running Example

```
glimpse(results)
```

```
## Rows: 380
## Columns: 16
## $ HomeTeam            <chr> "Brentford", "Man United", "Burnley", "Chelsea", ~
## $ AwayTeam            <chr> "Arsenal", "Leeds", "Brighton", "Crystal Palace",~
## $ Date                   <date> 2021-08-13, 2021-08-14, 2021-08-14, 2021-08-
14, ~
## $ outcome             <fct> Home win, Home win, Away win, Home win, Home win,~
## $ home_win            <lgl> TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, FALSE,~
## $ away_win            <lgl> FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, T~
## $ draw                <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ home_goals          <dbl> 2, 5, 1, 3, 3, 1, 3, 0, 2, 1, 2, 2, 0, 2, 5, 2, 1~
## $ away_goals          <dbl> 0, 1, 2, 0, 1, 0, 2, 3, 4, 0, 0, 0, 0, 2, 0, 0, 1~
## $ home_reds           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ away_reds           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ league_position_home <dbl> 15.5, 15.5, 5.5, 15.5, 15.5, 15.5, 15.5, 5.5, 5.5~
## $ league_position_away <dbl> 5.5, 5.5, 15.5, 5.5, 5.5, 5.5, 5.5, 15.5, 15.5, 5~
## $ league_position_diff <dbl> 10.0, 10.0, -10.0, 10.0, 10.0, 10.0, 10.0, -
10.0,~
## $ last_match_away     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 8, 7, 9, 8, 8, 8, 9~
## $ last_match_home     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 8, 8, 8, 8, 7, 8, 9~
```

Running Example

```r
table(results$outcome, results$home_win)
```

```
##
##            FALSE TRUE
##   Away win   129    0
##   Draw        88    0
##   Home win     0  163
```

Classification

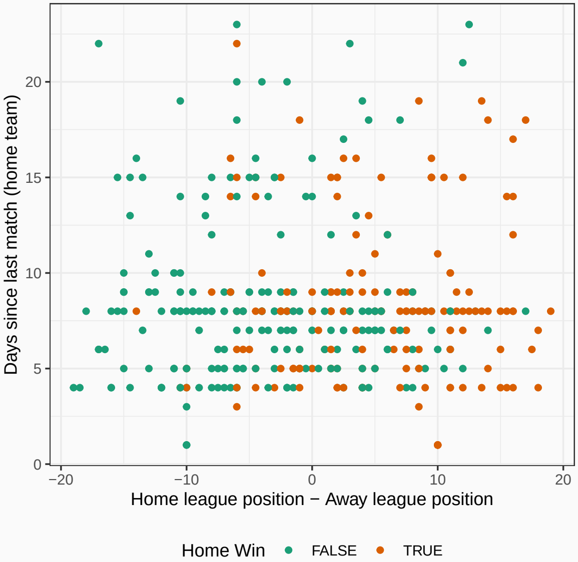
Classification

- Qualitative variables take values in an unordered set $\mathcal{C}$, such as: *eye color* $\in$ *{brown,blue,green}; email* $\in$ *{spam, ham}*; *football results* $\in$ *{away win,draw,home win}*.
- 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 $\mathcal{F}(\mathcal{X})$ that takes as input the feature vector X and predicts its value for Y; i.e. $\mathcal{F}(\mathcal{X}) \in \mathcal{C}$.

Classification

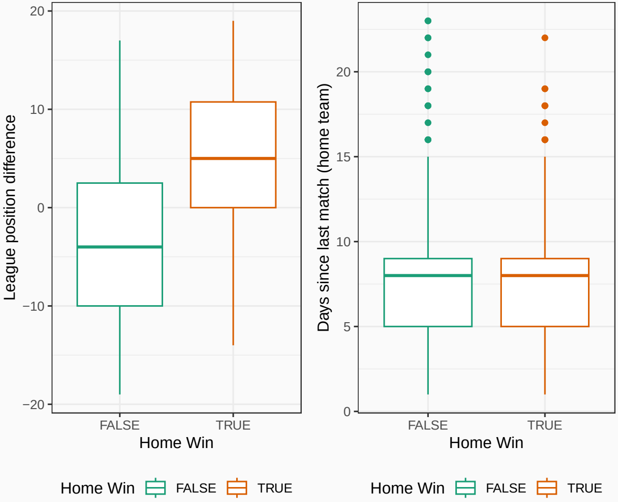
- Often we are more interested in estimating the probabilities that X belongs to each category in $\mathcal{C}$.
- For example, it is sometimes more valuable to have an estimate of the *probability* that an insurance claim is fraudulent, than a *classification* fraudulent or not.
- A successful gambling strategy, for instance, requires placing bets on outcomes to which you believe the bookmakers have assigned incorrect probabilities. Knowing the most likely outcome is not enough!

Example



Days since last match (home team)
0
5
10
15
20
−20
−10
0
10
20
Home league position − Away league position
Home Win
FALSE
TRUE

Example



League position difference
20
10
0
−10
−20
FALSE
TRUE
Home Win
Home Win FALSE TRUE
Days since last match (home team)
20
15
10
5
0
FALSE
TRUE
Home Win
Home Win FALSE TRUE

Example

- These plots suggest that we have some information that could be used to predict match outcomes
- Which methods are suitable for this task, given the nature of the outcome variable?

The Linear Probability Model

Can we just use Linear Regression?

Suppose for the home-win classification task we code

$$Y = \begin{cases} 0 & \text{if } Draw \text{ or } AwayWin \\ 1 & \text{if } HomeWin. \end{cases}$$

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

- In this case of a binary outcome, linear regression can do a reasonable job as a classifier!
- Since in the population $E(Y|X = x) = Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.
- However, linear regression applied to limited dependent variables has some undesirable properties as a classifier.

Linear Probability Model

The linear regression for binary outcome variables is known as the linear probability model:

Linear Probability Model

$$
\begin{aligned}
E[Y|X_1, X_2, ..., X_k] &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k \\
Pr(Y = 1|X_1, X_2, ...) &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k
\end{aligned}
$$

Advantages:

- We can use a well-known model for a new class of phenomena
- Easy to interpret the marginal effects of X variables

Disadvantages:

- The linear model assumes a continuous dependent variable, if the dependent variable is binary we run into problems.

Linear Probability Model – Advantages

Let's estimate a standard linear regression model using OLS for the football data:

```r
mod <- lm(home_win ~ league_position_diff, data = results)
```

```
##
## ================================
##                       Model 1
## --------------------------------
## (Intercept)             0.43 ***
##                        (0.02)
## league_position_diff    0.03 ***
##                        (0.00)
## --------------------------------
## R^2                     0.25
## Adj. R^2                0.25
## Num. obs.             380
## ================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Linear Probability Model – Advantages

- In the LPM, $\hat{\beta}_1$ estimates the change in *the probability that* $Y = 1$ associated with a unit increase in X
- An increase of 1 league position is associated with a .03 increase in the probability of a home win
- For equally placed teams (difference in league positions = 0), the probability of a home win is .43 (remember draws!)

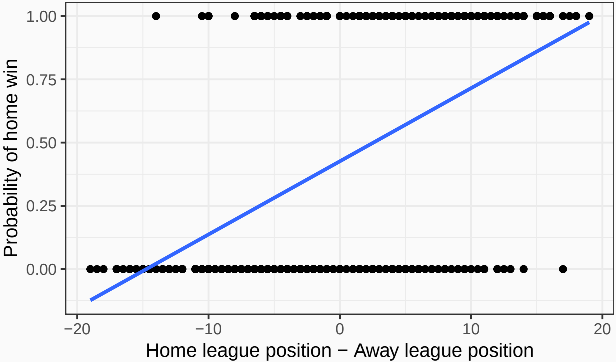
Linear Probability Model – Disadvantages

Problems with Linear Probability Model

Predictions, $\widehat{Y}$, are interpreted as probability for $Y = 1$

- $P(Y = 1) = \hat{Y} = \beta_0 + \beta_1 X$
- Can be above 1 if X is large enough
- Can be below 1 if X is small enough

Linear Probability Model – Disadvantages



Problem: linear regression can predict probabilities < 0 and > 1.

Linear Probability Model – Disadvantages

Now suppose we have a response variable with three possible values. A patient presents at a hospital, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if } stroke; \\ 2 & \text{if } drug\ overdose; \\ 3 & \text{if } epileptic\ seizure. \end{cases}$$

- This coding suggests an ordering, and in fact implies that the difference between $stroke$ and $drug\ overdose$ is the same as between $drug\ overdose$ and $epileptic\ seizure$.
- Linear regression is not appropriate here!

Linear Probability Model – Disadvantages

Problems:

1. Linear regression can predict probabilities < 0 and > 1.
2. Linear probability models don't work *at all* when we have more than two (unordered) categories
3. OLS requires homoskedastic residuals, with $E(u_i|X_i) = 0$. In the LPM the errors will have non-constant variance (thus messing up our standard errors)

Implication:

- We want a model that will provide predictions restricted to the 0-1 interval!
- Logistic regression is well suited to this task

Logistic Regression

Functions

A function maps values from X onto exactly one value of Y. We would write a function of X as $f(X)$

We can think of a function as a rule which tells us how to transform X, the argument of the function, to another specific value.

For example:

- $f(X) = X^2$ (quadratic function)
- $f(X) = log(X)$ (logarithmic function)
- $f(X) = \beta_0 + \beta_1 X$ (linear function)

For example, if we were to give the value $X = 2$ to the following function:

$$\begin{aligned} f(X) &= 2 + 3 \cdot X \\ f(X) &= 2 + 3 \cdot 2 \\ f(X) &= 8 \end{aligned}$$

Link functions

With a binary dependent variable:

- We want to model the probability of an outcome
- Probabilities can be a maximum of 1 and a minimum of 0
- $\longrightarrow$ we need a function that only returns values between 0 and 1.

Link functions

Rather than a model like this:

$$P(Y_i = 1) = \alpha + \beta_1 X_{1i}$$

We can instead have a model like this:

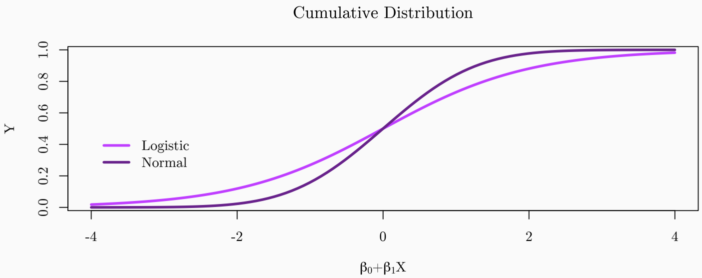
$$P(Y_i = 1) = f(\alpha + \beta_1 X_{1i})$$

Where $F(\cdot)$ is a function which never returns values below 0 or above 1

There are two functions that we might use:

Logit and probit

- The **logit** model, which is based on the cumulative logistic distribution (Δ)
- The **probit** model, which is based on the cumulative normal distribution (Φ)



Implications:

- We now have models which provide predictions that can be interpreted as probabilities
- Both will give very similar results but we focus on the logit model (it is a little more convenient)

Logistic Regression Model

The logit model is also known as the logistic regression model, and has the following features:

- Y is a binary response variable, with values 0 and 1
- $X_1, \dots, X_k$ are k explanatory variables of any type
- For each observation i, the following equation holds for $P(Y_i = 1) = \pi_i$:

$$\log(\texttt{Odds}_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \alpha + \beta_1 X_{1i} + \cdots + \beta_k X_{ki}$$

where α and $\beta_1, \dots, \beta_k$ are the unknown parameters of the model, to be estimated from data

Logistic regression models the *log-odds* that Y belongs to a given category.

Model for the probabilities

- Although the model is written first for the log-odds, it also implies a model for the probabilities, π_i:

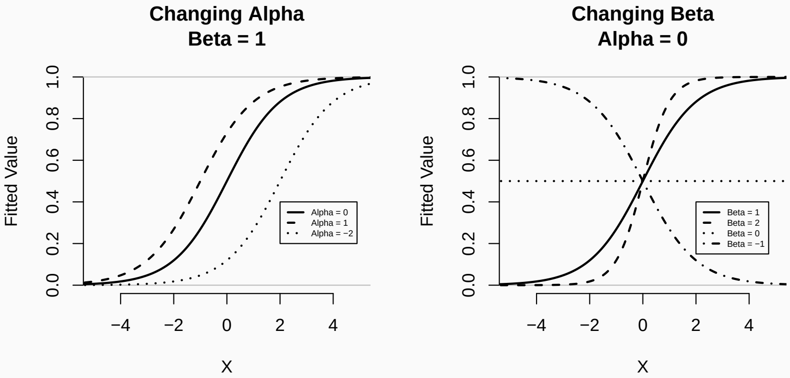
$$\pi_i \quad = \quad \frac{\exp(\alpha + \beta_1 X_{1i} + \cdots + \beta_k X_{ki})}{1 + \exp(\alpha + \beta_1 X_{1i} + \cdots + \beta_k X_{ki})}$$

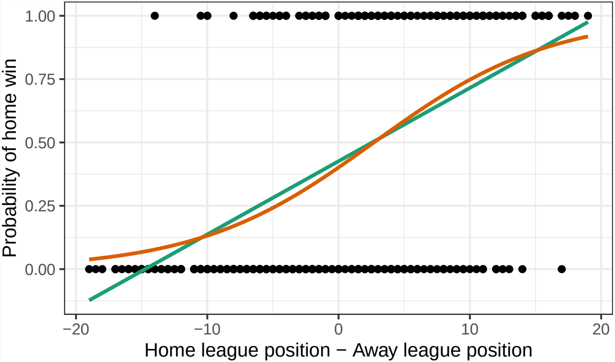
- This is always between 0 and 1
- The plots on the next slide give examples of

$$\pi = \frac{\exp(\alpha + \beta X)}{1 + \exp(\alpha + \beta X)}$$

for a simple logistic model with one continuous X

Probabilities from a logistic model





Logistic regression ensures that our estimate for $p(X)$ lies between 0 and 1.

Maximum Likelihood

- As with linear regression, the coefficients – α and β – are unknown and need to be estimated from training data
- We use maximum likelihood estimation (MLE) to estimate the parameters.
- Intuition: What are the values for α and β that generate predicted probabilities, $\widehat{Y}_i$ for each training observation that are as close as possible to the realised outcomes, Y_i?

Maximum Likelihood

- π_i is the probability that observation i has $Y = 1$:

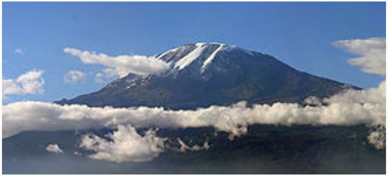
$$\pi_i \quad = \quad \frac{\exp(\alpha + \beta_1 X_{1i} + \cdots + \beta_k X_{ki})}{1 + \exp(\alpha + \beta_1 X_{1i} + \cdots + \beta_k X_{ki})}$$

- The likelihood function for logit regression is:

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} \hat{\pi}_i \prod_{i:y_i=0} (1 - \hat{\pi}_i).$$

- This likelihood gives the probability of the *observed* zeros and ones in the data, given values for $\beta_0, \beta_1, ...\beta_k$.
- That is, we want to pick values of $\beta_0, \beta_1, ...\beta_k$ to *maximize the likelihood* of the observed data.

Maximum Likelihood - An analogy



- How do you find the latitude and longitude of a mountain peak if you can't see very far?

    1. Start somewhere.
    2. Look around for the best way to go up.
    3. Go a small distance in that direction.
    4. Look around for the best way to go up.
    5. Go a small distance in that direction.
    6. ...

- This is what we do when we estimate the binary logistic regression model.

Implementation

```r
logistic_model <- glm(home_win ~ league_position_diff,
                data = results,
                family = binomial)
```

Implementation

```r
summary(logistic_model)
```

```
##
## Call:
## glm(formula = home_win ~ league_position_diff, family = binomial,
##     data = results)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.39944    0.12214  -3.270  0.00107 **
## league_position_diff  0.14856    0.01697   8.754  < 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: 519.09  on 379  degrees of freedom
## Residual deviance: 412.18  on 378  degrees of freedom
## AIC: 416.18
##
## Number of Fisher Scoring iterations: 4
```

Interpretation

Some aspects of interpretation are straightforward:

- The sign of the coefficients indicate the direction of the associations
  - $\beta_{\text{league\_position\_diff}} > 0 \rightarrow$ bigger difference in league position between home and away teams *increases* probability of a home win
- The significance of the coefficients are still determined by $\frac{\hat{\beta}}{SE(\hat{\beta})}$
  - We can reject the null that the relationship between league position and the home team winning is zero

Interpretation

- It is possible to interpret the coefficients directly…
    - $\rightarrow$ an increase of one league position is associated with an increase of $\beta_{\text{league\_position\_diff}} = 0.15$ in the log-odds of the home team winning
    - $\rightarrow$ the log-odds of the home team winning are $\alpha = -.4$ when the league position difference is zero
- …but no-one thinks in terms of log-odds!
- You do not need to be able to interpret the coefficients' magnitude for the assessment

Calculating predicted probabilities

- Just as we were interested in fitted values for linear regression, we are often interested in fitted probabilities for logistic regression
- The logistic regression gives us an equation for calculating the fitted log-odds that $Y = 1$ for a given set of X values:

$$\log(\widehat{\frac{\pi_i}{1-\pi_i}}) = \alpha + \hat{\beta}_1 * X_1 + \hat{\beta}_2 * X_2$$

- To recover the fitted probability that $Y = 1$, we use

$$\hat{\pi}_i = \frac{\exp(\hat{\alpha} + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i})}{1 + \exp(\hat{\alpha} + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i})}$$

for selected values of the explanatory variables $X_1, \ldots, X_k$

Calculating predicted probabilities

- What is our estimated probability of *a home win* when the home team is 10 places above the away team in the league?

$$\hat{p}(X) = \frac{\exp(\hat{\alpha} + \hat{\beta}_1 X_{1i})}{1 + \exp(\hat{\alpha} + \hat{\beta}_1 X_{1i})} = \frac{exp(-0.4 + 0.15 \times 10)}{1 + exp(-0.4 + 0.15 \times 10)} = 0.75$$

- How about when the home team is 5 places *below* the away team in the league?

$$\hat{p}(X) = \frac{\exp(\hat{\alpha} + \hat{\beta}_1 X_{1i})}{1 + \exp(\hat{\alpha} + \hat{\beta}_1 X_{1i})} = \frac{exp(-0.4 + 0.15 \times -5)}{1 + exp(-0.4 + 0.15 \times -5)} = 0.24$$

Calculating predicted probabilities

In R, we can calculate the predicted probabilities using the following:

```r
predict(logistic_model, newdata = data.frame(league_position_diff = 10),
        type = "response")
```

```
##         1
## 0.7476565
```

```r
predict(logistic_model, newdata = data.frame(league_position_diff = -5),
        type = "response")
```

```
##         1
## 0.2419103
```

where `type = "response"` tells R to calculate predicted probabilities (rather than fitted log-odds)

Calculating predicted probabilities

We can also calculate predicted probabilities for every match in our data:

```r
results$p_home_win <- predict(logistic_model, type = "response")
```

```r
summary(results$p_home_win)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03834 0.21572 0.40145 0.42895 0.65486 0.91858
```

Calculating predicted probabilities

```
results[which.max(results$p_home_win),
        c("HomeTeam", "AwayTeam",
          "league_position_home", "league_position_away",
          "p_home_win",
          "home_goals", "away_goals")]
```

```
## # A tibble: 1 x 7
##   HomeTeam AwayTeam league_position_home league_position_away p_home_win
##   <chr>    <chr>                   <dbl>                <dbl>      <dbl>
## 1 Chelsea  Norwich                    20                    1      0.919
##   home_goals away_goals
##        <dbl>      <dbl>
## 1          7          0
```


29
21
24
22

Calculating predicted probabilities

```
results[which.min(results$p_home_win),
        c("HomeTeam", "AwayTeam",
          "league_position_home", "league_position_away",
          "p_home_win",
          "home_goals", "away_goals")]
```

```
## # A tibble: 1 x 7
##   HomeTeam AwayTeam league_position_home league_position_away p_home_win
##   <chr>    <chr>                   <dbl>                <dbl>      <dbl>
## 1 Norwich  Man City                    1                   20     0.0383
##   home_goals away_goals
##        <dbl>      <dbl>
## 1          0          4
```


No room for racism

Multiple logistic regression

It is straightforward to extend the logistic model to include multiple predictors:

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

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

Implementation

```r
logistic_model_2 <- glm(home_win ~ league_position_diff + home_reds + away_reds,
                        data = results,
                        family = binomial)
```

```
##
## ===================================
##                        Model 1
## -----------------------------------
## (Intercept)              -0.38 **
##                          (0.13)
## league_position_diff      0.15 ***
##                          (0.02)
## home_reds                -2.60 *
##                          (1.08)
## away_reds                 0.92
##                          (0.51)
## -----------------------------------
## AIC                     405.57
## BIC                     421.33
## Log Likelihood         -198.79
## Deviance                397.57
## Num. obs.               380
## ===================================
```

Interpretation

1. Differences in league position increase the probability that the home team wins
2. If the home team receives a red card, they are significantly less likely to win (`home_reds` < 0)
3. If the away team receives a red card, the home team is somewhat more likely to win (`away_reds` > 0, but $p > 0.05$)

```r
predict(logistic_model_2,
        newdata = data.frame(league_position_diff = 0,
                             home_reds = c(0,1),
                             away_reds = 0),
        type = "response")
```

```
##          1          2
## 0.40527887 0.04809615
```

Implication: For equally matched teams, the home team receiving a red card reduces their probability of winning from .4 to .05.


REFEREE
FIFA

Example

South African Heart Disease

Public health policy often requires predicting which types of people are at risk of disease, and which individual-level characteristics are important risk factors for diseases. In this example, we use logistic regression to predict the occurance of coronary heart disease from a set of demographic factors and health measures. This data is drawn from a study in South Africa in the 1980s which aimed to evaluate the relative strengths and directions of different risk factors.

- Unit of analysis: 303 individuals
- Outcome (Y): **AHD**, equal to 1 if the individual has coronary heart disease (as measured from an aniographic test), and 0 otherwise
- Predictors (X): 13 variables measuring demographics and heart and lung function measurements

Example: South African Heart Disease

```
## Rows: 303
## Columns: 14
## $ Age       <dbl> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, 56, 44, 52, ~
## $ ChestPain <chr> "typical", "asymptomatic", "asymptomatic", "nonanginal", "no~
## $ RestBP    <dbl> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140, 140, 140, ~
## $ Chol      <dbl> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203, 192, 294, ~
## $ Fbs       <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, ~
## $ RestECG   <dbl> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, ~
## $ MaxHR     <dbl> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155, 148, 153, ~
## $ ExAng     <dbl> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ~
## $ Oldpeak   <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1, 0.4, 1.3, ~
## $ Slope     <dbl> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, 1, 3, 1, 1, 1, ~
## $ Ca        <dbl> 0, 3, 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ~
## $ Thal      <chr> "fixed", "normal", "reversable", "normal", "normal", "normal~
## $ AHD       <dbl> 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, ~
## $ Female    <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, ~
```

```r
heart_logit <- glm(AHD ~ . , data = SAheart, family = binomial)
```

```r
summary(heart_logit)
```

```
##
## Call:
## glm(formula = AHD ~ ., family = binomial, data = SAheart)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.536439   2.711852  -0.935 0.349625
## Age                  -0.012296   0.024664  -0.499 0.618120
## ChestPainnonanginal -1.804627   0.492607  -3.663 0.000249 ***
## ChestPainnontypical -0.935649   0.556725  -1.681 0.092835 .
## ChestPaintypical    -2.006802   0.652608  -3.075 0.002105 **
## RestBP               0.023981   0.011110   2.159 0.030889 *
## Chol                 0.004930   0.003944   1.250 0.211306
## Fbs                 -0.610758   0.599184  -1.019 0.308052
## RestECG              0.255433   0.189565   1.347 0.177829
## MaxHR               -0.021281   0.010821  -1.967 0.049224 *
## ExAng                0.739431   0.434687   1.701 0.088931 .
## Oldpeak              0.353095   0.230102   1.535 0.124903
## Slope                0.670508   0.371616   1.804 0.071184 .
## Ca                   1.269290   0.271304   4.678 2.89e-06 ***
## Thalnormal          -0.011430   0.795090  -0.014 0.988530
## Thalreversable       1.429947   0.783279   1.826 0.067912 .
## FemaleTRUE          -1.431422   0.513185  -2.789 0.005282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 409.95  on 296  degrees of freedom
## Residual deviance: 194.83  on 280  degrees of freedom
##   (6 observations deleted due to missingness)
## AIC: 228.83
##
## Number of Fisher Scoring iterations: 6
```

Predicted probabilities

How does the probability of having heart disease vary as a function of age and maximum heart rate?

We can generate predicted probabilities via:

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

where we set all variables to their sample means or modes, and then vary the values of `Age` and `MaxHR`

Predicted probabilities

```
vals_age <- data.frame(Age = 29:77,
           ChestPain = "asymptomatic",
           RestBP = mean(SAheart$RestBP),
           Chol = mean(SAheart$Chol),
           Fbs = mean(SAheart$Fbs),
           RestECG = mean(SAheart$RestECG),
           MaxHR = mean(SAheart$MaxHR),
           ExAng = mean(SAheart$ExAng),
           Oldpeak = mean(SAheart$Oldpeak),
           Slope = mean(SAheart$Slope),
           Ca = mean(SAheart$Ca, na.rm = TRUE),
           Thal = "normal",
           Female = FALSE)
```

Predicted probabilities

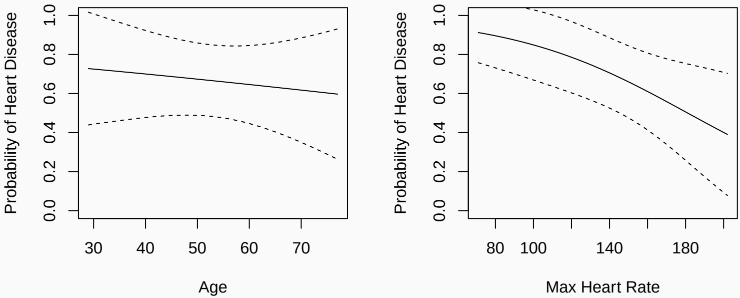
```r
vals_maxhr <- data.frame(Age = mean(SAheart$Age),
           ChestPain = "asymptomatic",
           RestBP = mean(SAheart$RestBP),
           Chol = mean(SAheart$Chol),
           Fbs = mean(SAheart$Fbs),
           RestECG = mean(SAheart$RestECG),
           MaxHR = 71:202,
           ExAng = mean(SAheart$ExAng),
           Oldpeak = mean(SAheart$Oldpeak),
           Slope = mean(SAheart$Slope),
           Ca = mean(SAheart$Ca, na.rm = TRUE),
           Thal = "normal",
           Female = FALSE)
```

Predicted probabilities

```
age_probs <- predict(heart_logit,
                     newdata = vals_age,
                     type = "response",
                     se.fit = TRUE)
```

```
maxhr_probs <- predict(heart_logit,
                     newdata = vals_maxhr,
                     type = "response",
                     se.fit = TRUE)
```

Predicted probabilities



Break

Multinomial Classification

Multinomial Logistic Regression

- So far we have discussed logistic regression with two classes.
- It is easily generalized to more than two classes.
- Here there is a non-linear function for the probability of each class.
- Multiclass logistic regression is also referred to as multinomial regression.

The log-odds for each non-reference category $j = 1, \dots, C-1$ against the *reference category* 0 depends on the values of the explanatory variables through:

$$\log\left(\frac{\pi_i^{(j)}}{\pi_i^{(0)}}\right) = \alpha^{(j)} + \beta_1^{(j)} X_{1i} + \cdots + \beta_k^{(j)} X_{ki}$$

for each $j = 1, \dots, C-1$ where $\alpha^{(j)}$ and $\beta_1^{(j)}, \dots, \beta_k^{(j)}$ are unknown population parameters

Multinomial Logistic Regression

- Multinomial logit regression can be estimated using the `glmnet` package in R
- Because there are many more parameters to estimate versus a binary logit model, multinomial models typically take much longer to estimate (particularly if N or P are large)
- As before, inference can be performed directly on the estimated coefficients
- As before, the coefficients are hard to interpret and so calculating predicted probabilities is normally preferable

Naive Bayes Classifier

- Logistic regression involves modelling $P(Y = k|X)$ using the logistic distribution.
- An alternative approach to estimating the conditional distribution of Y given X is to use Bayes' rule
- Bayes's rule tells us that:

$$P(Y = k|X_i = x) \propto P(Y)P(X_i = x|Y = k)$$

where:

- $P(Y)$ is the prior probability of the outcome (i.e. the probability of a given class before we see any data)
- $P(X_i = x|Y = k)$ is the likelihood or conditional probability of X_i given the class Y

Our goal is therefore to estimate these probabilities in order to calculate the conditional probability that we care about: $P(Y = k|X)$

Naive Bayes Classifier

- $P(Y)$
  - the probability that a randomly chosen observation is in class k
  - can be estimated from the sample proportions of k
- $P(X_i = x|Y = k)$
  - the probability of a randomly chosen observation in class k having $X_i = x$
  - higher when it is likely that an observation in k has $X_i = x$
  - lower when it is unlikely that an observation in k has $X_i = x$
  - Because X_i is a *vector* of covariates, we need to work out this probability from a multivariate probability distribution
  - Or we can cheat and use the Naive Bayes classifier

Naive Bayes Classifier

- The key simplification step here is to assume that features are independent
  - While this assumption is pretty heroic and generally not true, it significantly simplifies the estimation.
- The probability of an observation, Y_i, being assigned to a class, k:

$$P(Y_i = k|X_i) \propto P(k) \prod_{j=1}^{J} P(x_j|k)$$

- We then assign the observation to kth class for which it has the highest posterior probability:

$$\hat{Y}_i = \underset{k \in \{1,\ldots,k\}}{\operatorname{argmax}} P(k) \prod_{j=1}^{J} P(x_j|k)$$

Naive Bayes Classifier

- Despite the strong assumptions it makes, NB classifiers often outperform far more sophisticated alternatives.
- Naive Bayes is especially appropriate when the dimension p of the feature space is high
- We will come back to Naive Bayes in our text classification lecture next week

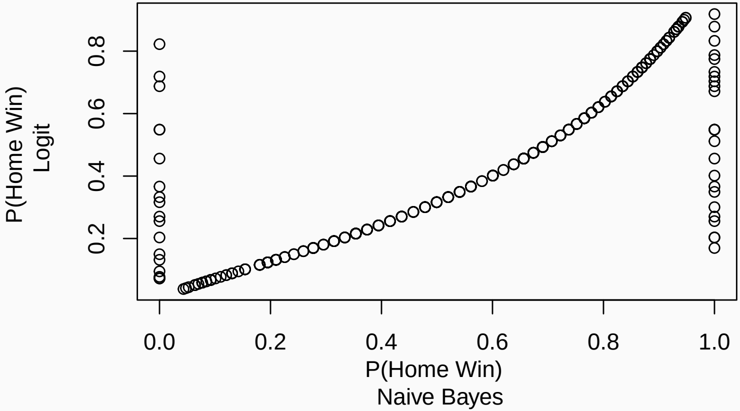
Naïve Bayes example

```r
library(e1071)

nb_model <- naiveBayes(home_win ~ league_position_diff + home_reds + away_reds,
                       data = results)

results$p_home_win_nb <- predict(nb_model, newdata = results, type = "raw")[,2]
```

Naïve Bayes example



Naïve Bayes Multiclass example

```
library(e1071)

nb_model_multiclass <- naiveBayes(outcome ~  home_reds + away_reds +
                                      HomeTeam + AwayTeam,
                       data = results)

results$pred_outcome_nb <- predict(nb_model_multiclass, newdata = results)

table(results$pred_outcome_nb, results$outcome)
```

```
##
##             Away win Draw Home win
##   Away win        81   21       10
##   Draw             2    5        3
##   Home win        46   62      150
```

Other classification approaches

1. **Tree-based methods**
    - Partition the covariate space into discrete regions
    - Classify observations into the modal outcome class in each region
    - More on these tomorrow!
2. **Support Vector Machines**
    - Estimate a set of (non-linear) boundaries through the covariate space that separate between classes
    - Classify observations according to which side of the boundaries they fall
3. **Deep learning/Neural networks**
    - Derive new features which are non-linear functions of existing covariates
    - Use these transformations as inputs to a (generalised) linear model for Y
    - Classify new observations by applying the transformations and predicting from the fitted model

Other classification approaches

- These methods, in different ways, allow for complex non-linearities in the relationship between predictors and outcome and also allow for interactions between predictors.
- Success tends to be somewhat task specific, but there is also often little variation is success (at least for simple problems).

Characterizing performance of classifiers

How good is our football classifier?

We are often most interested in whether we get each classification decision "right", rather than how close we came to being right.

```
results$home_win_pred <- results$p_home_win > .5

table(Prediction = results$home_win_pred,
      Result = results$home_win)
```

```
##            Result
## Prediction FALSE TRUE
##      FALSE   168   64
##      TRUE     49   99
```

```
(168 + 99)/380
```

```
## [1] 0.7026316
```

Is this good?

The naïve guess

Best prediction without a model

Suppose you had to come up with a prediction of whether any home team would win without using a statistical model. What would you predict?

- One reasonable guess would be just to use the mean outcome in your data
- We can get $\hat{\pi}$, unconditional on predictors, by taking the mean of Y
- If $\hat{\pi} > 0.5$:
    - $Pr(Y = 1) > Pr(Y = 0)$
    - 1's in our binary DV are **more** common than 0's
- If $\hat{\pi} < 0.5$:
    - $Pr(Y = 1) < Pr(Y = 0)$
    - 1's in our binary DV are **less** common than 0's

The naïve guess

The naïve guess is the most common outcome of the dependent variable

In our data, `home_win` is the dependent variable:

```
mean(results$home_win)
```

```
## [1] 0.4289474
```

Thus, $P(Y = 1) < P(Y = 0)$.

$\longrightarrow$ the naïve guess is therefore 0, that the home team will not win.

The naïve guess

```r
results$home_win_naive <- FALSE

mean(results$home_win_naive == results$home_win)
```

```
## [1] 0.5710526
```

- Even making the simplest possible guess, we get an accuracy of 57%
- Thankfully our logit regression does better than that!
- The general point here is that classification accuracy can be misleading...

COVID confusion

How accurate are PCR tests? (Ai et al., Radiology, 2020)

- True COVID status = measured by an x-ray + doctor
- Predicted COVID status = people swabbing themselves with a PCR test

		True COVID Status		
		Does not have COVID	Has COVID	Total
Predicted	Negative Test	105	308	413
COVID Status	Positive test	21	580	601
	Total	126	888	1014

- Error rate = $\frac{21+308}{1014} = 32.4\%$
- Accuracy = $\frac{105+580}{1014} = 67.5\%$

But, note that the error-rates are different for the healthy and the sick!

- Proportion of *healthy* classified as having COVID = $\frac{21}{126} = 16.7\%$
- Proportion of *sick* classified as *not* having COVID = $\frac{308}{888} = 34.7\%$

- False positive rate: The fraction of negative examples that are classified as positive – 16.7% in this example.
- False negative rate: The fraction of positive examples that are classified as negative – 34.7% in this example.

Sensitivity and specificity

- The performance of a classifier is often characterized in terms of sensitivity and specificity.
- Here, the sensitivity is the percentage of sick people that are correctly identified: $\frac{580}{888} = 65.3\%$
- The specificity is the percentage of healthy people that are correctly identified: $\frac{105}{126} = 83.3\%$

Which is best?

- Our prioritization of false-negative/false-positive rates will often depend on the application
- For judicial decisions, maybe we'd prefer false negatives than false positives
  - Would you rather put an innocent person in jail or let a guilty one go free?
- For COVID tests, we'd probably be more happy to accept false positives that false negatives
  - Would you rather isolate for no reason, or be coughed on by a sick person?

Application

```r
confusion_tab <- table(Prediction = results$home_win_pred,
                       Result = results$home_win)

confusion_tab
```

```
##           Result
## Prediction FALSE TRUE
##      FALSE   168   64
##      TRUE     49   99
```

- Accuracy = $\frac{99+168}{380} = 70.3\%$
- Sensitivity = $\frac{99}{163} = 60.7\%$
- Specificity = $\frac{168}{217} = 77.4\%$

caret and confusionMatrix()

```r
library(caret)
confusionMatrix(confusion_tab, positive = "TRUE")
```

```
## Confusion Matrix and Statistics
##
##           Result
## Prediction FALSE TRUE
##      FALSE   168   64
##      TRUE     49   99
##
##                Accuracy : 0.7026
##                  95% CI : (0.6539, 0.7482)
##     No Information Rate : 0.5711
##     P-Value [Acc > NIR] : 8.627e-08
##
##                   Kappa : 0.386
##
##  Mcnemar's Test P-Value : 0.1878
##
##             Sensitivity : 0.6074
##             Specificity : 0.7742
##          Pos Pred Value : 0.6689
```

- We produced the confusion matrix above by classifying to `home_win = TRUE` if

$$\widehat{Pr}(HomeWin|X) \geq 0.5$$

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

$$\widehat{Pr}(HomeWin|X) \geq threshold,$$

Varying the $threshold$

For example, if we classify according to $\widehat{Pr}(HomeWin|X) \geq .2$,

```
results$home_win_pred_tmp <- results$p_home_win > .2
confusionMatrix(table(results$home_win_pred_tmp, results$home_win),
                positive = "TRUE")
```

```
## Confusion Matrix and Statistics
##
##
##         FALSE TRUE
##   FALSE    81    5
##   TRUE    136  158
##
##                Accuracy : 0.6289
##                  95% CI : (0.5782, 0.6777)
##     No Information Rate : 0.5711
##     P-Value [Acc > NIR] : 0.01253
##
##                   Kappa : 0.3115
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.9693
##             Specificity : 0.3733
##          Pos Pred Value : 0.5374
##          Neg Pred Value : 0.9419
##              Prevalence : 0.4289
##          Detection Rate : 0.4158
##    Detection Prevalence : 0.7737
##       Balanced Accuracy : 0.6713
##
##        'Positive' Class : TRUE
##
```

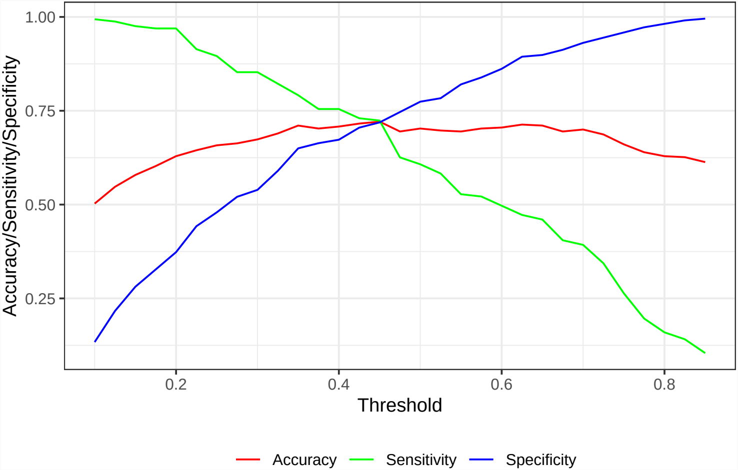
Varying the $threshold$

For example, if we classify according to $\widehat{Pr}(HomeWin|X) \geq .8$,

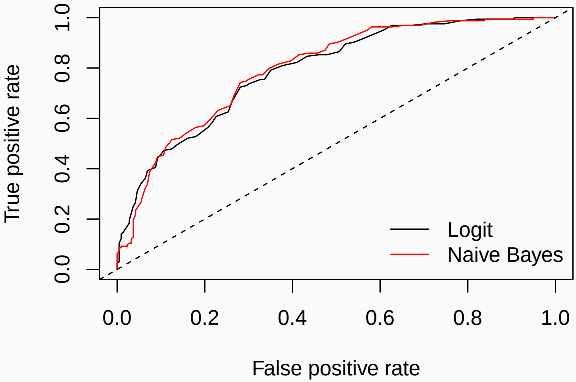
```r
results$home_win_pred_tmp <- results$p_home_win > .8
confusionMatrix(table(results$home_win_pred_tmp, results$home_win),
                positive = "TRUE")
```

```
## Confusion Matrix and Statistics
##
##
##         FALSE TRUE
##   FALSE   213  137
##   TRUE      4   26
##
##                Accuracy : 0.6289
##                  95% CI : (0.5782, 0.6777)
##     No Information Rate : 0.5711
##     P-Value [Acc > NIR] : 0.01253
##
##                   Kappa : 0.157
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.15951
##             Specificity : 0.98157
##          Pos Pred Value : 0.86667
##          Neg Pred Value : 0.60857
##              Prevalence : 0.42895
##          Detection Rate : 0.06842
##    Detection Prevalence : 0.07895
##       Balanced Accuracy : 0.57054
##
##        'Positive' Class : TRUE
##
```

Varying the $threshold$



ROC curve



- The ROC plot displays both the true positive rate and the false positive rate simultaneously (for different thresholds).
- Sometimes we use the AUC or area under the curve to summarize the overall performance and to compare models.
- Higher AUC is good.

Characterizing performance of classifiers

		Predicted	class	
		- or Null	+ or Non-null	Total
True	- or Null	True Neg. (TN)	False Pos.(FP)	N
class	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

- "+'' is "disease" or alternative (non-null) hypothesis (e.g. "home win");
- "-'' is "non-disease" or the null hypothesis (e.g. "away win or draw").

Performance measures for classifiers

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1- Specificity
True Pos. rate	TP/P	1 - Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1-false discovery proportion
Neg. Pred. value	TN/N*	

- The denominators for the false positive and true positive rates are the actual population counts in each class.
- The denominators for the positive predictive value and the negative predictive value are the total predicted counts for each class.

Summary

- Classification methods differ from regression methods because we are interested in qualitative outcomes, rather than continuous ones
- Logistic regression is very popular for classification, particularly when the number of classes is low (i.e. $k = 2$)
- Naive Bayes is useful when p is very large and is cheap to implement.
- Confusion matrices help us to assess the performance of our classifiers, but we need to think carefully about which metrics are most informative for each task
- (Football matches are hard to forecast)

Midterm

- **Released**: this afternoon.
- **Focus**: mostly linear regression, a small amount on logistic regression.
- **Deadline**: Wednesday 27th July, 15.00.