

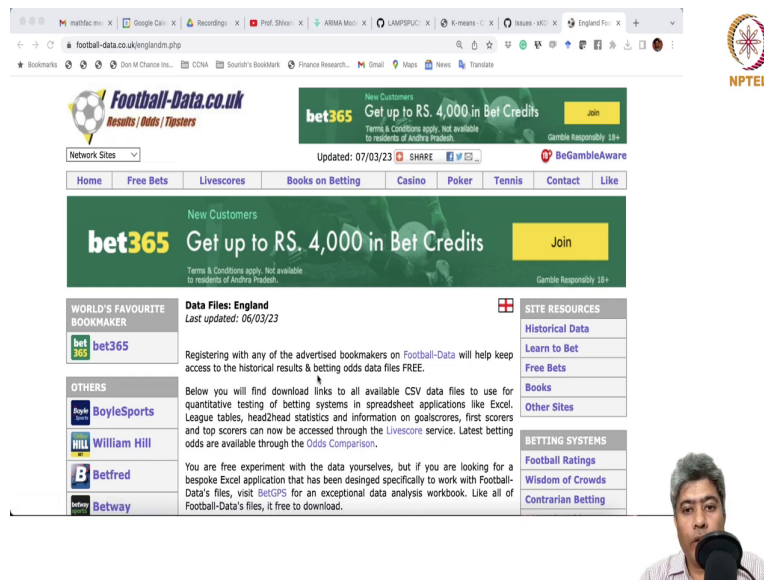
**Predictive Analytics - Regression and Classification**  
**Prof. Sourish Das**  
**Department of Mathematics**  
**Mathematical Institute, Chennai**

**Lecture - 53**

**Hands on with R: Implement Tree Regression and Random Forest with EPL football Data**

Hello all. In this video we are going to Implement the Regression tree and Random Forest using R. We will use real life English Premier League dataset and we will see; we will also check how good these models are in out of the sample.

(Refer Slide Time: 00:43)



The screenshot shows the Football-Data.co.uk website. The main content area features a large green banner for bet365 with the text "Get up to RS. 4,000 in Bet Credits" and a "Join" button. Below this, there is a section titled "Data Files: England" with a sub-heading "Last updated: 06/03/23". This section lists several bookmakers: bet365, BoyleSports, William Hill, Betfred, and Betway. A paragraph of text explains that users can find download links to CSV data files for quantitative testing of betting systems in spreadsheet applications like Excel. To the right of the main content, there is a "SITE RESOURCES" sidebar with links for "Historical Data", "Learn to Bet", "Free Bets", "Books", and "Other Sites". Below that, a "BETTING SYSTEMS" sidebar lists "Football Ratings", "Wisdom of Crowds", and "Contrarian Betting". The website's navigation menu includes "Home", "Free Bets", "Livescores", "Books on Betting", "Casino", "Poker", "Tennis", "Contact", and "Like". The NPTEL logo is visible in the top right corner of the browser window.

So, I am going to take call these, I am going to you know directly call these datasets.

(Refer Slide Time: 00:49)

The screenshot shows the website [football-data.co.uk/englandm.php](http://football-data.co.uk/englandm.php). The page features several sections:

- Top Navigation:** Includes logos for William Hill, Betfred, and Betway.
- Main Content:** A large green banner for **bet365** advertising a match between **AEK Larnaca** and **West Ham** on Friday, 03:45. Below this, there are sections for "Season 2022/2023", "Season 2021/2022", and "Season 2020/2021", each listing various football leagues (Premier League, Championship, League 1, League 2, Conference) with links to match stats and odds.
- Right Sidebar:** Contains several utility sections:
  - BETTING SYSTEMS:** Football Ratings, Wisdom of Crowds, Contrarian Betting, Pinnacle Odds Drop (NEW).
  - BETTING ARTICLES:** Football-Data, Pinnacle Sportsbook.
  - BET CALCULATORS:** Fair Odds, P-value, Yields, Bank growth, EV-odds, Staking Animation.
  - ODDS & RESULTS: MAIN LEAGUES:** Latest Matches, England (with a flag icon), Scotland (with a flag icon).
- Bottom Right:** A small video inset shows a man speaking into a microphone.

(Refer Slide Time: 00:54)

The screenshot shows a web browser displaying the website [football-data.co.uk/englandm.php](http://football-data.co.uk/englandm.php). The page features a sidebar with a Betway logo and a main content area with a list of matches and their odds. A small video inset of a man speaking is visible in the bottom right corner.

**bet365**

UEFA Europa Confe...  
**AEK Larnaca**  
v **West Ham**  
Fri 03:45

1	X	2
5.25	5.25	5.25

UEFA Europa Confe...  
**Anderlecht**  
v **Villarreal**  
Fri 03:45

Notes.txt  
(text file key to the data files and data source acknowledgements)

**Season 2022/2023**

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)
- League 1 (FT & HT results; match stats; match, total goals & AH odds)
- League 2 (FT & HT results; match stats; match, total goals & AH odds)
- Conference (FT & HT results; match, total goals & AH odds)

**Season 2021/2022**

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)
- League 1 (FT & HT results; match stats; match, total goals & AH odds)
- League 2 (FT & HT results; match stats; match, total goals & AH odds)
- Conference (FT & HT results; match, total goals & AH odds)

**Season 2020/2021**

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)
- League 1 (FT & HT results; match stats; match, total goals & AH odds)
- League 2 (FT & HT results; match stats; match, total goals & AH odds)
- Conference (FT & HT results; match, total goals & AH odds)

**Season 2019/2020**

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)

**Wisdom of Crowds**

- Contrarian Betting
- Pinnacle Odds Drop **NEW**

**BETTING ARTICLES**

- Football-Data
- Pinnacle Sportsbook

**BET CALCULATORS**

- Fair Odds
- P-value
- Yields
- Bank growth
- EV-odds
- Staking Animation

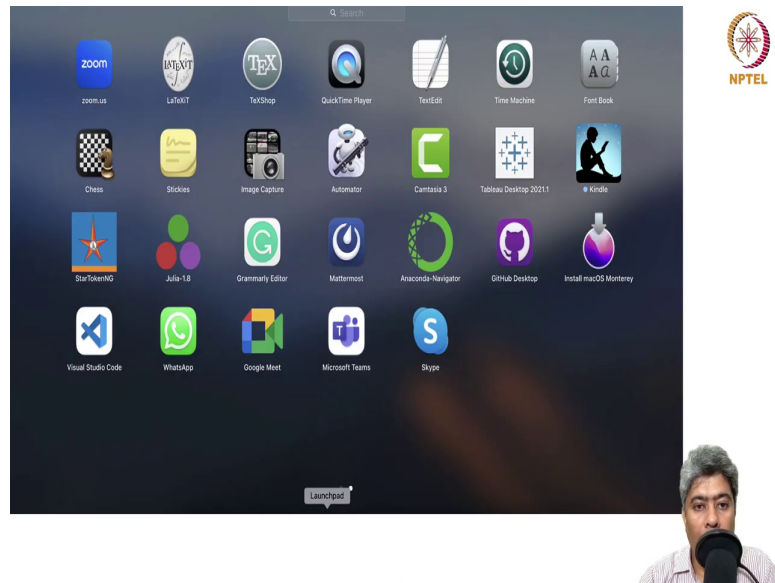
**ODDS & RESULTS: MAIN LEAGUES**

Latest Matches

- England
- Scotland
- Germany
- Italy

So, there is 22, 23. I am not sure if all the data available for the last league. We can check it out. Let me just see. So, what we I will do? Let me first open R.

(Refer Slide Time: 01:12)

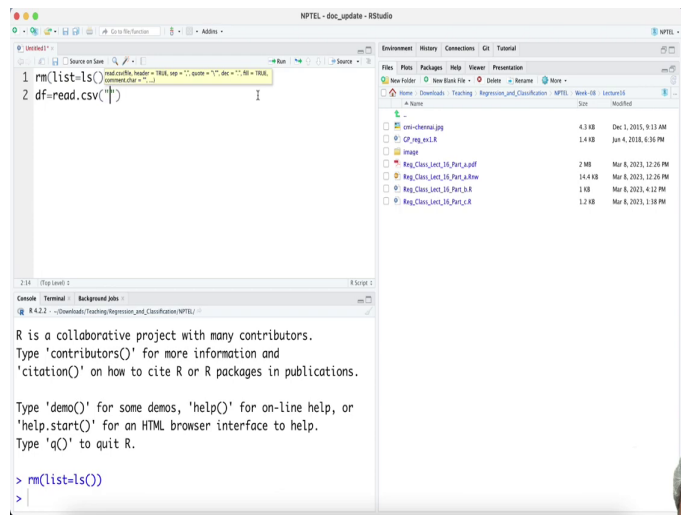


Open R.

(Refer Slide Time: 01:17)



(Refer Slide Time: 01:20)



The screenshot shows the RStudio interface. The script editor contains the following R code:

```
1 rm(list=ls())
2 df=read.csv("...")
```



The console output displays the following text:

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> rm(list=ls())
>
```

On the right side of the RStudio window, there is a file explorer showing a list of files and folders, including 'img', 'Reg\_Class\_LM1\_16\_Ppt\_1.pdf', 'Reg\_Class\_LM1\_16\_Ppt\_2.pdf', 'Reg\_Class\_LM1\_16\_Ppt\_3.pdf', and 'Reg\_Class\_LM1\_16\_Ppt\_4.pdf'.



So, first what I will do df say equal to, let me first write rm, list equal to ls. So, it always a good practice to have that start with a clean environment and then data frame read dot csv.

(Refer Slide Time: 01:54)

The screenshot displays the website [football-data.co.uk/englandm.php](http://football-data.co.uk/englandm.php). The page features a navigation bar with a Betway logo and a main content area with several sections:

- bet365**: A large green banner for the England Premier League match between Crystal Palace and Man City, scheduled for Sunday, 03:30. The odds are listed as 10.00 for home, 10.00 for draw, and 10.00 for away.
- UEFA Europa League**: A banner for the match between Bayer Leverkusen and Ferencváros, scheduled for Friday, 03:45.
- Season 2022/2023**: A list of matches for the current season, including Premier League, Championship, League 1, League 2, and Conference.
- Season 2021/2022**: A list of matches for the previous season, including Premier League, Championship, League 1, League 2, and Conference.
- Season 2020/2021**: A list of matches for the season before last, including Premier League, Championship, League 1, League 2, and Conference.
- Season 2019/2020**: A list of matches for the season before that, including Premier League and Championship.

On the right side of the page, there are several utility sections:

- Contrarian Betting**: A section with a link for "Pinnacle Odds Drop NEW".
- BETTING ARTICLES**: A list of articles including "Football-Data" and "Pinnacle Sportsbook".
- BET CALCULATORS**: A section with links for "Fair Odds", "P-value", "Yields", "Bank growth", "EV-odds", and "Staking Animation".
- ODDS & RESULTS: MAIN LEAGUES**: A section with a "Latest Matches" link and a list of leagues: England, Scotland, Germany, Italy, and France.

In the bottom right corner, there is a small video inset showing a man speaking into a microphone.

First what I will do.

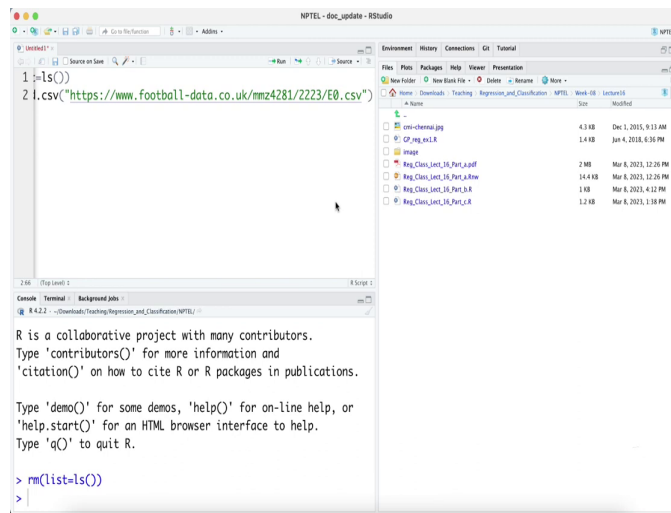
(Refer Slide Time: 01:56)

The screenshot displays the website [football-data.co.uk/englandm.php](https://www.football-data.co.uk/englandm.php). The page features a list of football matches categorized by season (2022/2023, 2021/2, 2020/2, 2019/2020). A context menu is open over a link, with 'Open Link in Incognito Window' highlighted. The right sidebar includes sections for 'Contrarian Betting', 'BETTING ARTICLES', 'BET CALCULATORS', and 'ODDS & RESULTS: MAIN LEAGUES'. A small video inset in the bottom right corner shows a man speaking into a microphone.

I will just take the copy the link address and I will put it here and head equal to true.



(Refer Slide Time: 02:02)



The screenshot displays the RStudio environment. The script editor contains the following R code:

```
1 ls()  
2 l.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv")
```

The file explorer on the right shows a directory structure with the following files and folders:

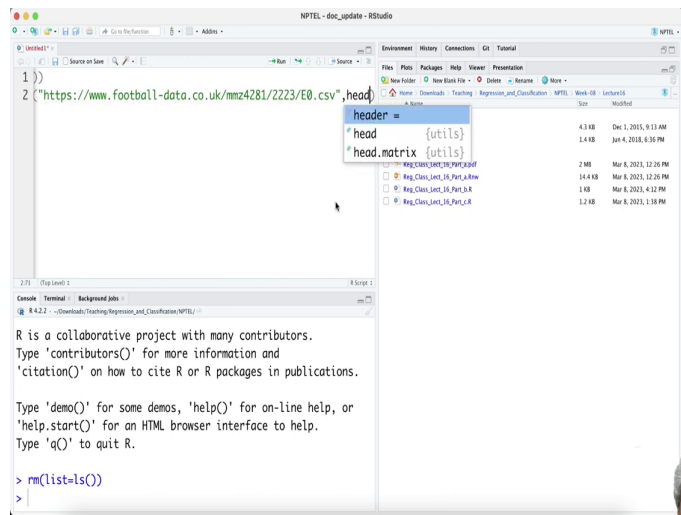
Name	Size	Modified
..		
00_000-channels.jpg	4.3 KB	Dec 1, 2015, 9:13 AM
01_CP_img_en1.R	1.4 KB	Jan 4, 2018, 4:36 PM
image		
Rmg_Class_Lmt_16_Pmt_1.pdf	2 MB	Mar 8, 2021, 12:26 PM
Rmg_Class_Lmt_16_Pmt_2.pdf	14.4 KB	Mar 8, 2021, 12:26 PM
Rmg_Class_Lmt_16_Pmt_3.R	1.8 KB	Mar 8, 2021, 4:12 PM
Rmg_Class_Lmt_16_Pmt_4.R	1.2 KB	Mar 8, 2021, 1:58 PM

The console window at the bottom shows the following text:

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> rm(list=ls())  
>
```



(Refer Slide Time: 02:05)



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 })  
2 ("https://www.football-data.co.uk/mmz4281/2223/E0.csv", header =
```

A tooltip is visible over the code, listing the following functions:



- header =
- head {utils}
- head.matrix {utils}

The Environment pane on the right shows the following objects:

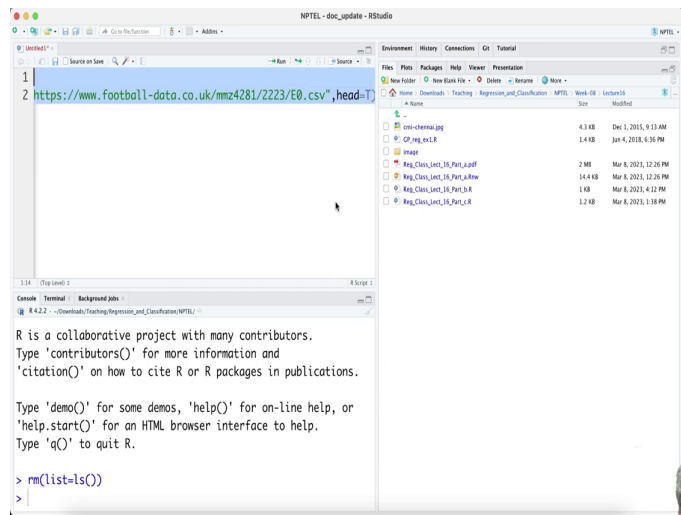
Object	Size	Modified
header	4.3 KB	Dec 1, 2015, 9:13 AM
head	1.4 KB	Jan 4, 2018, 4:36 PM
head.matrix	2 MB	Mar 8, 2021, 12:26 PM
Reg_Class_LIME_16_Part_2_10m	14.4 KB	Mar 8, 2021, 12:26 PM
Reg_Class_LIME_16_Part_2_8	1 KB	Mar 8, 2021, 4:12 PM
Reg_Class_LIME_16_Part_2_8	1.2 KB	Mar 8, 2021, 1:58 PM

The Console window shows the following text:

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> rm(list=ls())  
>
```



(Refer Slide Time: 02:08)





The screenshot shows the RStudio interface. The script editor contains two lines of code:

```
1  
2 https://www.football-data.co.uk/mmz4281/2223/E0.csv, head=T
```

The file explorer on the right shows a directory structure with files like `00_00000000.jpg`, `CP_img_enL.R`, `image`, `Rmg_Class_Lmt_16_Pmt_1.pdf`, `Rmg_Class_Lmt_16_Pmt_2.pdf`, `Rmg_Class_Lmt_16_Pmt_3.R`, `Rmg_Class_Lmt_16_Pmt_4.R`, and `Rmg_Class_Lmt_16_Pmt_5.R`.

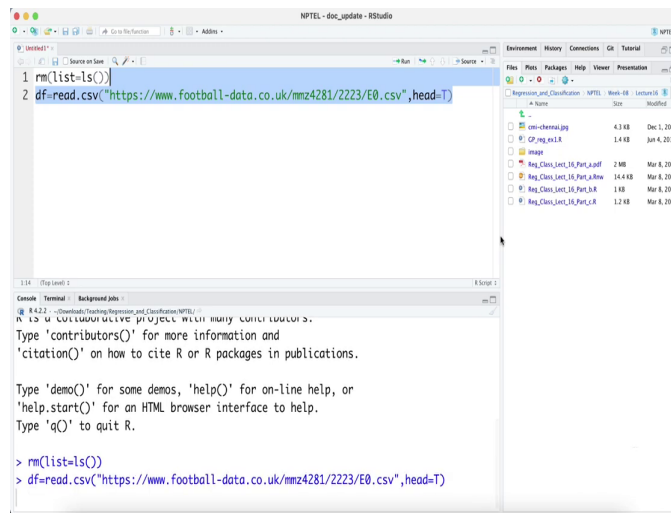
The console window shows the following text:

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> rm(list=ls())  
>
```



So, yeah let me run that.

(Refer Slide Time: 02:16)



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 rm(list=ls())  
2 df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
```

The console shows the following output:

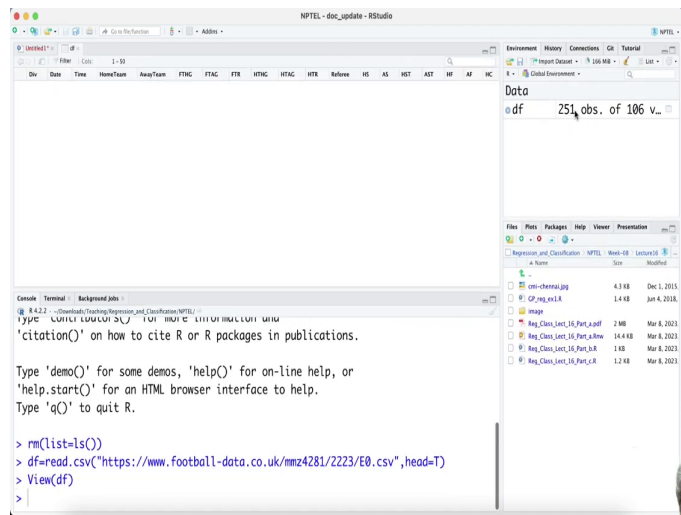
```
R 4.2.2 - Check for updates: Training, Regression, and Classification: NPTEL  
R 4.2.2: All R packages are available for installation. Please refer to the R help page.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> rm(list=ls())  
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
```

The environment pane on the right shows the following files:

Name	Size	Modified
..		
df	4.3 KB	Dec 1, 2015
df_csv	1.4 KB	Jun 4, 2018
image		
Reg_Class_1st_18_Part_1.pdf	2 KB	Mar 8, 2023
Reg_Class_1st_18_Part_1.ppt	14.4 KB	Mar 8, 2023
Reg_Class_1st_18_Part_2.R	1 KB	Mar 8, 2023
Reg_Class_1st_18_Part_2.R	1.2 KB	Mar 8, 2023





(Refer Slide Time: 02:20)



The screenshot displays the RStudio environment. The top toolbar includes options for Environment, History, Connections, and Tutorial. The top-right pane shows the 'Data' environment with a data frame 'df' containing 251 observations and 106 variables. The bottom-left pane is the console, showing the following R code and its output:

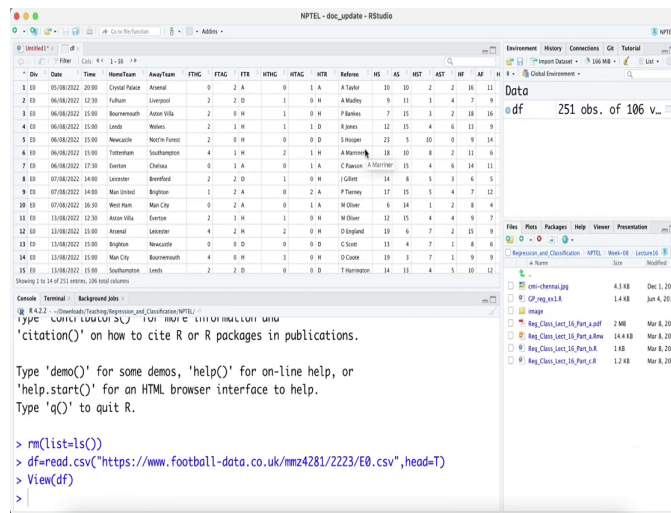
```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
>
```

The console also displays help text for the 'read.csv()' function, including instructions on how to cite R or R packages in publications and how to use 'demo()', 'help()', 'help.start()', and 'q()'.



So, here is. So, there are 251 observation.

(Refer Slide Time: 02:21)



The screenshot shows the RStudio interface with a data frame named 'df' containing 251 observations and 106 variables. The console shows the following R code and output:

```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
>
```

The console also displays help text for the 'citation()' function:

```
'citation()' on how to cite R or R packages in publications.

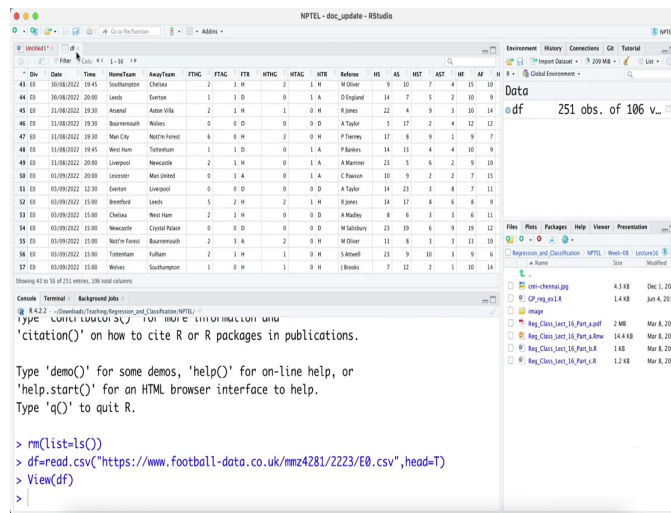
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
>
```

Div	Date	Time	HomeTeam	AwayTeam	FTAG	FTAG	FTE	HTAG	HTAG	HTE	Referee	HS	AS	HST	AST	HF	AF
1	05/08/2022	20:00	Crystal Palace	Arsenal	0	2	A	0	1	A	A Taylor	10	10	2	2	16	11
2	06/08/2022	12:30	Fulham	Liverpool	2	2	D	1	0	H	A Madley	9	11	3	4	7	9
3	06/08/2022	15:00	Bournemouth	Aston Villa	2	0	H	1	0	H	P Barnes	7	15	3	2	18	16
4	06/08/2022	15:00	Leeds	Hull City	2	1	H	1	1	D	R Jones	12	15	4	6	13	9
5	06/08/2022	15:00	Newcastle	Nottm Forest	2	0	H	0	0	D	S Hooper	23	5	10	0	9	14
6	06/08/2022	15:00	Tottenham	Southampton	4	1	H	2	1	H	A Mannix	18	10	8	2	11	6
7	06/08/2022	17:30	Everton	Chelsea	0	1	A	0	1	A	C Pawson	15	4	6	14	11	
8	07/08/2022	14:00	Leicester	Brentford	2	2	D	1	0	H	J Gillet	14	8	5	3	6	5
9	07/08/2022	14:00	Mill Wall	Brighton	1	0	A	0	2	A	P Rowe	17	15	5	4	7	12
10	07/08/2022	16:30	West Ham	Man City	0	2	A	0	1	A	M Oliver	6	14	1	2	8	4
11	11/08/2022	12:30	Aston Villa	Everton	2	1	H	1	0	H	M Oliver	12	15	4	4	9	7
12	11/08/2022	15:00	Arsenal	Leicester	4	2	H	2	0	H	D England	19	6	7	2	15	9
13	11/08/2022	15:00	Brighton	Newcastle	0	0	D	0	0	D	C Scott	11	4	7	1	8	6
14	11/08/2022	15:00	Man City	Bournemouth	4	0	H	3	0	H	D Cook	19	3	7	1	9	9
15	11/08/2022	15:00	Southampton	Leeds	2	2	D	0	0	D	T Harrison	14	11	4	5	10	12



(Refer Slide Time: 02:23)



The screenshot displays the RStudio interface. The main window shows a data frame with columns for Date, Time, HomeTeam, AwayTeam, FTAG, FTAG, FTE, HTAG, HTE, Referee, HS, AS, HST, AST, HF, and AF. The console window contains the following R code and output:

```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mm24281/2223/E0.csv",head=T)
> View(df)
>
```

The console also shows the following text:

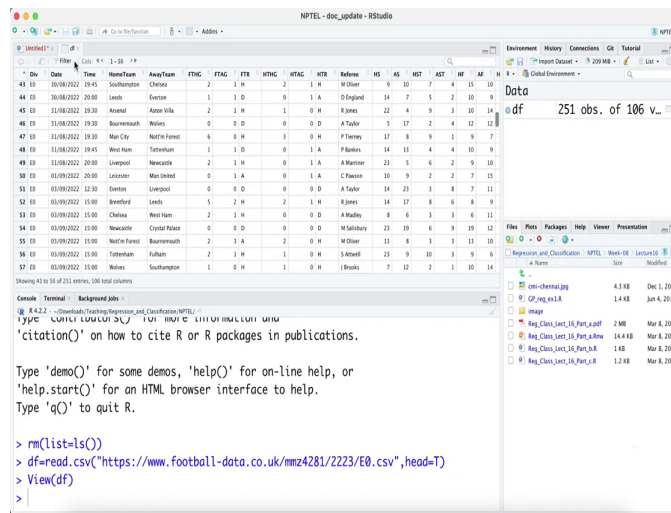
```
© R 4.2.2 ... Download Teaching Regression and Classification NPTEL
> ?citic 'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

The right-hand pane shows a file explorer with the following files:

- cm-chemical.jpg (4.3 KB, Dec 1, 2015)
- CP\_04\_01.R (1.4 KB, Jun 4, 2018)
- image
- Reg\_Class\_Lect\_16\_Part\_1.pdf (2 MB, Mar 8, 2023)
- Reg\_Class\_Lect\_16\_Part\_2.pptx (14.4 KB, Mar 8, 2023)
- Reg\_Class\_Lect\_16\_Part\_3.R (1 KB, Mar 8, 2023)
- Reg\_Class\_Lect\_16\_Part\_4.R (1.2 KB, Mar 8, 2023)



(Refer Slide Time: 02:23)



The screenshot shows the RStudio interface with a data frame named 'df' containing 251 observations and 106 variables. The main window displays a table of football match statistics. The console shows the following R code and output:

```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
>
```

The console also displays the following text:

```
© R 4.2.2 ... downloaded by Teaching Regression and Classification (NPTEL)
"citation()" on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

The environment pane on the right shows the following files:

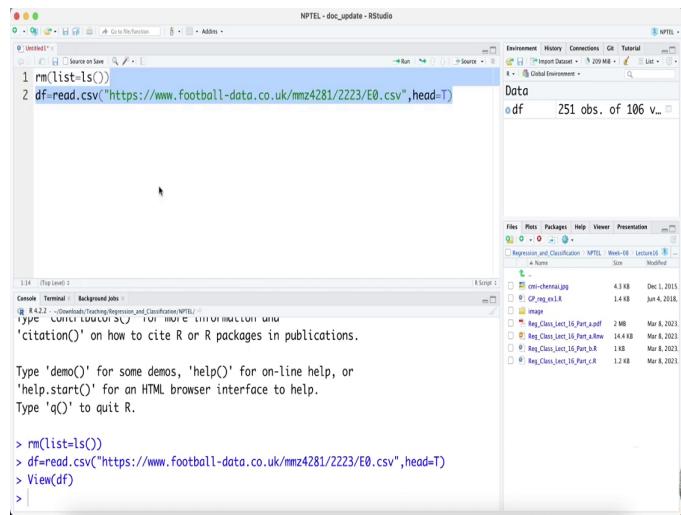
- cm-chemical.jpg (4.3 KB, Dec 1, 2015)
- CP\_04g\_kv1.R (1.4 KB, Jun 4, 2018)
- image
- Reg\_Class\_Lect\_16\_Part\_1.pdf (2 MB, Mar 8, 2023)
- Reg\_Class\_Lect\_16\_Part\_1.html (14.4 KB, Mar 8, 2023)
- Reg\_Class\_Lect\_16\_Part\_3.R (1 KB, Mar 8, 2023)
- Reg\_Class\_Lect\_16\_Part\_4.R (1.2 KB, Mar 8, 2023)



That means English Premier League is all still running.



(Refer Slide Time: 02:25)



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 rm(list=ls())
2 df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
```



The console shows the following output:

```
1:14 (Top level) >
R 4.2.2 ... Downloading Teaching Regression and Classification NPTEL
>df<= read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
>
```

The Data viewer on the right shows a data frame with 251 observations and 106 variables. The Files pane on the right shows a list of files, including 'reg\_class\_list\_16\_Part\_1.pdf', 'reg\_class\_list\_16\_Part\_2.pdf', 'reg\_class\_list\_16\_Part\_3.pdf', 'reg\_class\_list\_16\_Part\_4.pdf', and 'reg\_class\_list\_16\_Part\_5.pdf'.



So, they are not all observations are available, ok. So, not all results are available. So, what I will do?

(Refer Slide Time: 02:36)

The screenshot shows the website football-data.co.uk. The main content area displays betting odds for various leagues and seasons. On the left, there are advertisements for bet365, including matches like Roma v Real Socie... and Sporting v Arsenal. The main content lists seasons from 2022/2023 down to 2019/2020, with links for Premier League, Championship, League 1, League 2, and Conference. The right sidebar contains sections for Contrarian Betting, Betting Articles, Bet Calculators, and Odds & Results for Main Leagues.

bet365

UEFA Europa League

Roma v Real Socie...

Fri 03:45

1	X	2
2.10	2.10	2.10

UEFA Europa League

Sporting v Arsenal

Fri 03:45

1	X	2
2.10	2.10	2.10

Season 2022/2023

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)
- League 1 (FT & HT results; match stats; match, total goals & AH odds)
- League 2 (FT & HT results; match stats; match, total goals & AH odds)
- Conference (FT & HT results; match, total goals & AH odds)

Season 2021/2022

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)
- League 1 (FT & HT results; match stats; match, total goals & AH odds)
- League 2 (FT & HT results; match stats; match, total goals & AH odds)
- Conference (FT & HT results; match, total goals & AH odds)

Season 2020/2021

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)
- League 1 (FT & HT results; match stats; match, total goals & AH odds)
- League 2 (FT & HT results; match stats; match, total goals & AH odds)
- Conference (FT & HT results; match, total goals & AH odds)

Season 2019/2020

- Premier League (FT & HT results; match stats; match, total goals & AH odds)
- Championship (FT & HT results; match stats; match, total goals & AH odds)
- League 1 (FT & HT results; match stats; match, total goals & AH odds)
- League 2 (FT & HT results; match stats; match, total goals & AH odds)
- Conference (FT & HT results; match, total goals & AH odds)

Contrarian Betting

Pinnacle Odds Drop

NEW

BETTING ARTICLES

Football-Data

Pinnacle Sportsbook

BET CALCULATORS

Fair Odds

P-value

Yields

Bank growth

EV-odds

Staking Animation

ODDS & RESULTS: MAIN LEAGUES

Latest Matches

England

Scotland

Germany

Italy

Spain

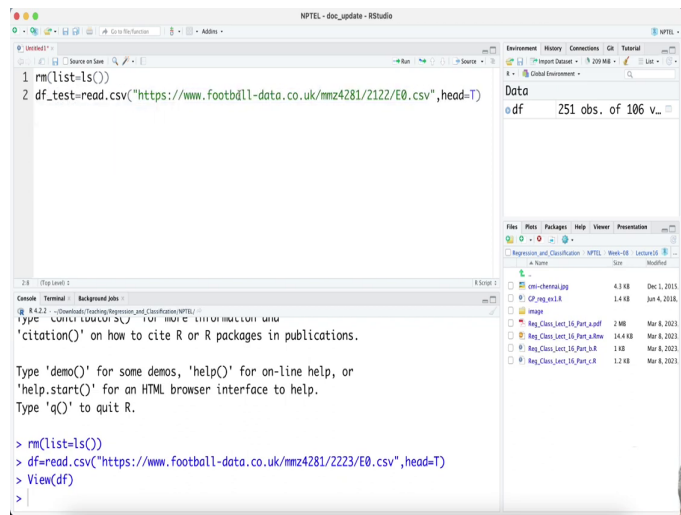
Let us instead of that, let us take 21, 22 data. So, we will take 20, 21 seasons data and train the model and we will use 21, 22 data to test the model.

(Refer Slide Time: 02:55)

The screenshot shows a web browser window with the URL <https://www.football-data.co.uk/eng43912122852.com>. The page content includes a Betway advertisement for a UEFA Europa League match between Union Berlin and Union Sain... on Friday, 03:45, with odds of 1.62 for home, draw, and away. Below it is another advertisement for Sevilla vs Fenerbahce on Friday, 06:00. The main content area lists football leagues and seasons with links to data files, such as 'Premier League (FT & HT results; match stats; match, total goals & AH odds)'. A context menu is open over the 'Copy Link Address' option for the 'Premier League' link. The right sidebar contains sections for 'Contrarian Betting', 'BETTING ARTICLES', 'BET CALCULATORS', and 'ODDS & RESULTS: MAIN LEAGUES'. The NPTEL logo is visible in the top right corner. A small video inset in the bottom right shows a man speaking into a microphone.

So, let me link address. I will just do that.

(Refer Slide Time: 03:06)



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
```

The console shows the output of the code execution:



```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
>
```

The right-hand pane shows the 'Data' tab with the following information:

Variable	Value
df	251 obs. of 106 v...

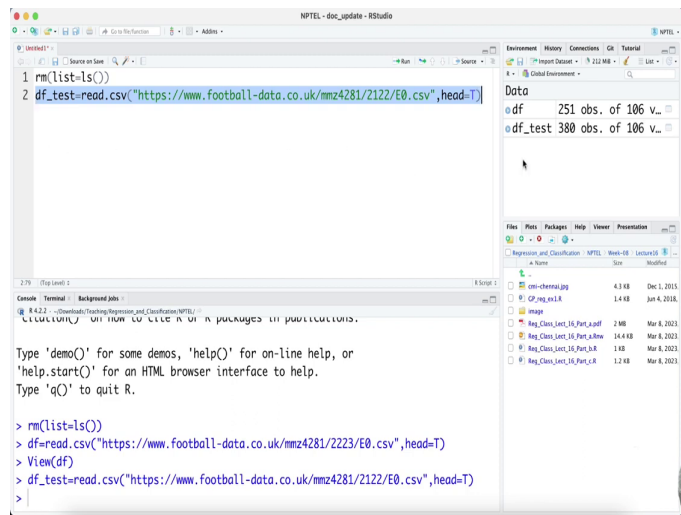
The file explorer on the right shows the following files:

Name	Size	Modified
cm-chemical.jpg	4.1 KB	Dec 1, 2015
CP_Reg_1.R	1.4 KB	Jun 4, 2018
image		
Reg_Class_1st_16_Part_1.pdf	2 MB	Mar 8, 2023
Reg_Class_1st_16_Part_1.html	14.4 KB	Mar 8, 2023
Reg_Class_1st_16_Part_3.R	1 KB	Mar 8, 2023
Reg_Class_1st_16_Part_3.R	1.2 KB	Mar 8, 2023



So, this is our test data set.

(Refer Slide Time: 03:14)



The screenshot shows an RStudio window titled "NPTEL - dev\_update - RStudio". The script editor contains the following code:

```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
```

The console shows the execution of these commands:



```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
>
```

The environment pane on the right shows the following data objects:

Object	Value
df	251 obs. of 106 v...
df_test	380 obs. of 106 v...

The file browser pane shows a list of files in the current directory:

Name	Size	Modified
...	...	...
con-chemical.jpg	4.1 KB	Dec 1, 2015
CP_Reg_1.R	1.4 KB	Jun 4, 2018
image	...	...
Reg_Class_1st_16_Part_1.pdf	2.9 KB	Mar 8, 2023
Reg_Class_1st_16_Part_1.html	14.4 KB	Mar 8, 2023
Reg_Class_1st_16_Part_1.R	1.8 KB	Mar 8, 2023
Reg_Class_1st_16_Part_1.R	1.2 KB	Mar 8, 2023



So, you can see 21, 22 season, we will use it as a test data set, ok. So, you can see that 380 observations are all there.

(Refer Slide Time: 03:19)

The screenshot displays the RStudio interface. The console window shows the following R commands and their output:

```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
>
```

A confirmation dialog box is open in the center, asking: "Are you sure you want to remove all objects from the environment? This operation cannot be undone." The "Yes" button is highlighted.

The Environment pane on the right shows the following objects:

Object	Class	Attributes
df	data.frame	251 obs. of 106 v...
df_test	data.frame	380 obs. of 106 v...

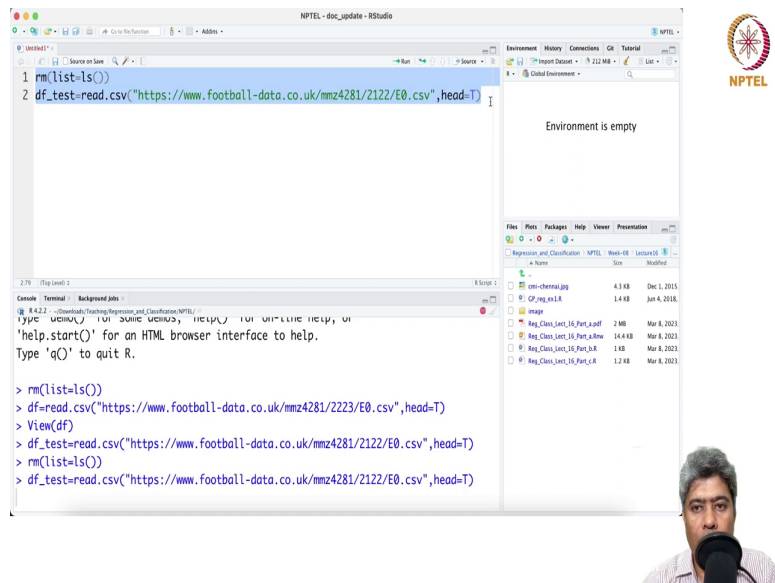
The Files pane on the right shows a list of files and folders, including:

Name	Size	Modified
...	...	...
img	4.1 KB	Dec 1, 2015
CP_img_e1.R	1.4 KB	Jun 4, 2018
Reg_Class_1st_16_Part_1.pdf	2 MB	Mar 8, 2023
Reg_Class_1st_16_Part_1.htm	14.4 KB	Mar 8, 2023
Reg_Class_1st_16_Part_1.R	1 KB	Mar 8, 2023
Reg_Class_1st_16_Part_1.R	1.2 KB	Mar 8, 2023

A small video inset in the bottom right corner shows a man with grey hair speaking into a black microphone.

So, you can just clean this overall.

(Refer Slide Time: 03:23)



The image shows a screenshot of the RStudio interface. The main editor window contains the following R code:

```
1 rm(list=ls())  
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
```

The console window shows the execution of the code:

```
> rm(list=ls())  
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)  
> View(df)  
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)  
> rm(list=ls())  
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
```

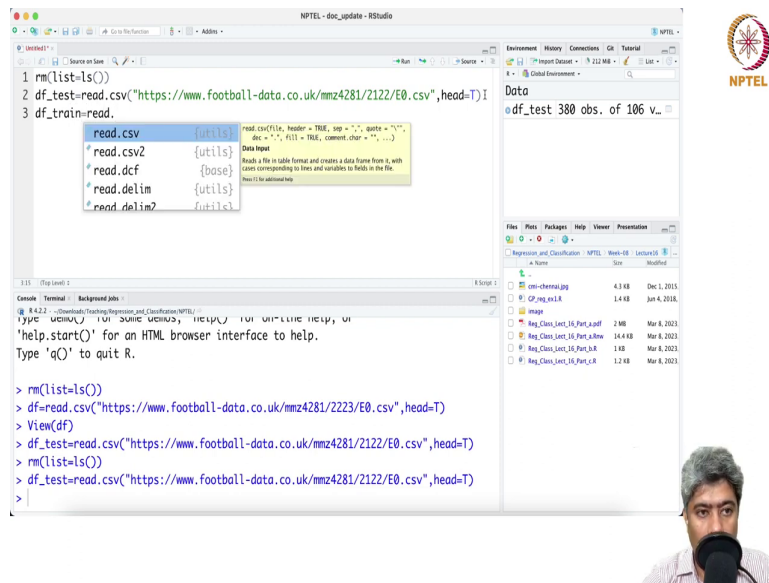
The environment pane on the right shows a table of loaded objects:

Name	Size	Modified
df	4.3 KB	Dec 1, 2015
df_test	1.4 KB	Jun 4, 2018
Reg_Class_1st_16_Part_1.pdf	2 KB	Mar 8, 2023
Reg_Class_1st_16_Part_2.pdf	1.4 KB	Mar 8, 2023
Reg_Class_1st_16_Part_3.pdf	1 KB	Mar 8, 2023
Reg_Class_1st_16_Part_4.pdf	1.2 KB	Mar 8, 2023

Below the RStudio window, there is a small video inset showing a man speaking into a microphone.

And, yeah.

(Refer Slide Time: 03:28)



The screenshot displays the RStudio interface. The script editor contains the following R code:

```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
```

A tooltip for the `read.csv` function is visible, showing its signature: `read.csv(file, header = TRUE, sep = ",", quote = "\"", as.is = FALSE, fill = TRUE, comment.char = "#", ...)`. The console shows the execution of the code, with the output for `df_test` being 380 observations and 106 variables.

In the bottom right corner, there is a video inset of a man speaking into a microphone.

Now, and then df train sorry, train equal to read dot csv.

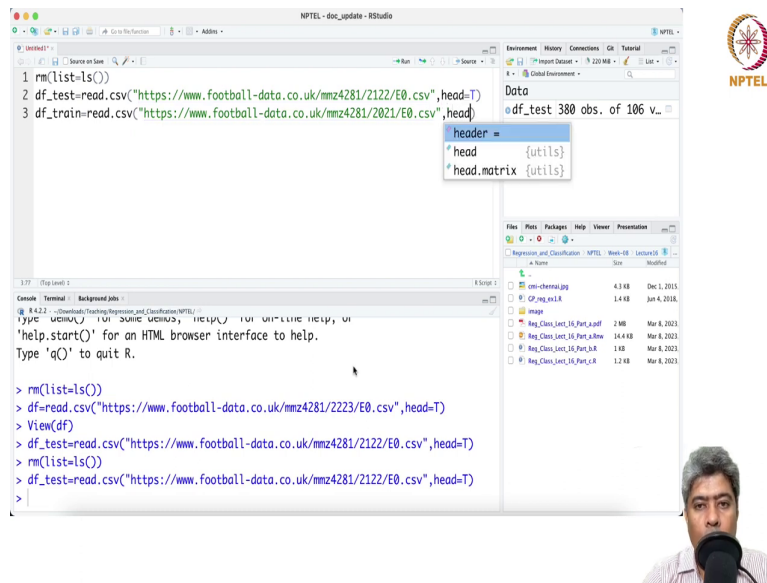


(Refer Slide Time: 03:42)

The screenshot shows a web browser window with the URL <https://www.football-data.co.uk/englandm.php>. The page content includes a sidebar with advertisements for Betway and bet365, and a main area listing football data for seasons 2022/2023, 2021/2022, 2020/21, and 2019/20. A context menu is open over a link, showing options like 'Open Link in New Window', 'Copy Link Address', and 'Copy Link to Highlight'. A small video inset in the bottom right corner shows a man speaking into a microphone.

And then we take 20, 21 data, copy the link address.

(Refer Slide Time: 03:49)



The screenshot shows the RStudio interface with the following code in the script editor:

```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",head=T)
```

The console output shows the execution of the code and the output of the head() function:

```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
> rm(list=ls())
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
>
```

The output of the head() function is shown in a small inset window:

```
header =
head {utils}
head.matrix {utils}
```

The NPTEL logo is visible in the top right corner of the screenshot.

And head equal to true, ok. So, so the 2021 series data we will use as a train and 21, 22 series we will use as a test and we will see what is happening in that. So, the first what I will do. So, as we have explored this data a little bit before in a previous hands on. So, we know that we have built the Poisson regression model.

Let us first fit the Poisson regression model and then we will compare the performance of the Poisson regression model with the other models with the regression tree and the random forest.

(Refer Slide Time: 04:36)

The screenshot shows an RStudio window titled 'NPTEL - dev\_update - RStudio'. The editor pane contains the following R code:

```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
4
5
6 ## Fir Poisson Regression model
7 #fit1 = glm(FTHG~HST+AST+HC+AC)
8 fit1 = glm(FTHG~HST+AST+HC+AC)
```

The console pane shows the following output:

```
R 4.2.2 - Downloads/Teaching/Regression_and_Classification/NPTEL/
RStudio: 2.10.1 (64-bit) 1.61 GB RAM 11196 MB SWAP 11152 MB LU HELP.
Type 'q()' to quit R.

> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
> rm(list=ls())
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
> df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
>
```

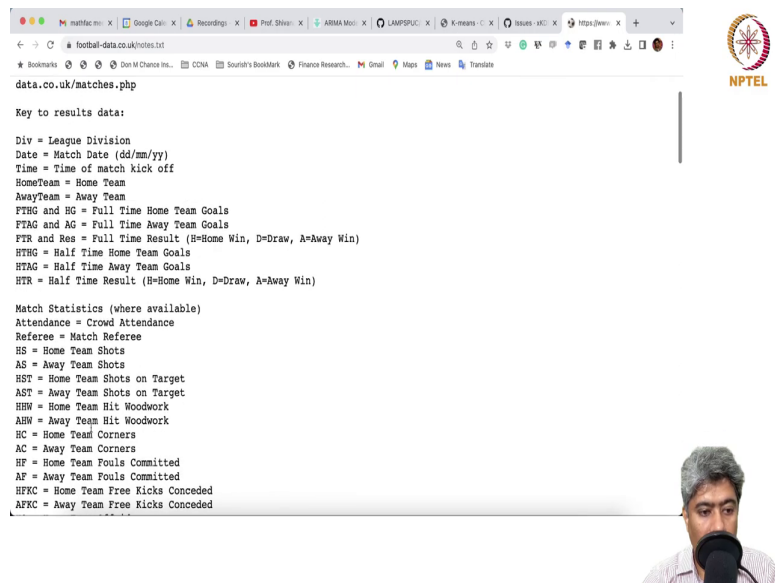
The right-hand pane shows the 'Data' tab with the following information:

df_te_	df_tr_
380 obs. of 10...	380 obs. of 10...

The bottom right corner of the slide features a small inset image of a man with a microphone, likely the instructor.

So, first fit Poisson Regression model, ok. So, fit1 equal to glm FTHG is the target variable full time how many goals scored by the home team and then HST how many shots are on target by the home team? AST how many shots are on target by the away team? HC, AC if we actually go and check the notes.

(Refer Slide Time: 05:30)





data.co.uk/matches.php

Key to results data:

- Div = League Division
- Date = Match Date (dd/mm/yy)
- Time = Time of match kick off
- HomeTeam = Home Team
- AwayTeam = Away Team
- FTHG and HG = Full Time Home Team Goals
- FTAG and AG = Full Time Away Team Goals
- FTR and Res = Full Time Result (H=Home Win, D=Draw, A=Away Win)
- HTHG = Half Time Home Team Goals
- HTAG = Half Time Away Team Goals
- HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win)

Match Statistics (where available)

- Attendance = Crowd Attendance
- Referee = Match Referee
- HS = Home Team Shots
- AS = Away Team Shots
- HST = Home Team Shots on Target
- AST = Away Team Shots on Target
- HHW = Home Team Hit Woodwork
- AHW = Away Team Hit Woodwork
- HC = Home Team Corners
- AC = Away Team Corners
- HF = Home Team Fouls Committed
- AF = Away Team Fouls Committed
- HFKC = Home Team Free Kicks Conceded
- AFKC = Away Team Free Kicks Conceded



HC is home team corners, away team corners. And then actually we can put many more things. For example, shots target yeah.

(Refer Slide Time: 05:55)

The screenshot displays the RStudio interface with the following R code in the editor:

```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
4
5
6 ## Fir Poisson Regression model
7
8 fit1 = glm(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A,data)
```

A tooltip for the `data` object is shown, listing its attributes: `data` (utils), `data.class` (base), `data.entry` (utils), and `data.frame` (base).

The console shows the following output:

```
*Type 'q()' to quit R.*
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
> rm(list=ls())
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
> df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
>
```

The Environment pane on the right shows the loaded objects: `df_te_` (380 obs. of 10...) and `df_tr_` (380 obs. of 10...). A small video inset in the bottom right corner shows a man speaking into a microphone.

And then HR plus AR. HR stands for home team red card, AR stands for away team's red card, HO and off side was not there, ok. And then bet we can also take the bet 365 if you go a little bit below. So, B365H plus B365 draw plus B365 away teams thing. So, this is a betting odds by bet 365 or house, betting house and we are going to use the train data set, ok.

(Refer Slide Time: 06:50)

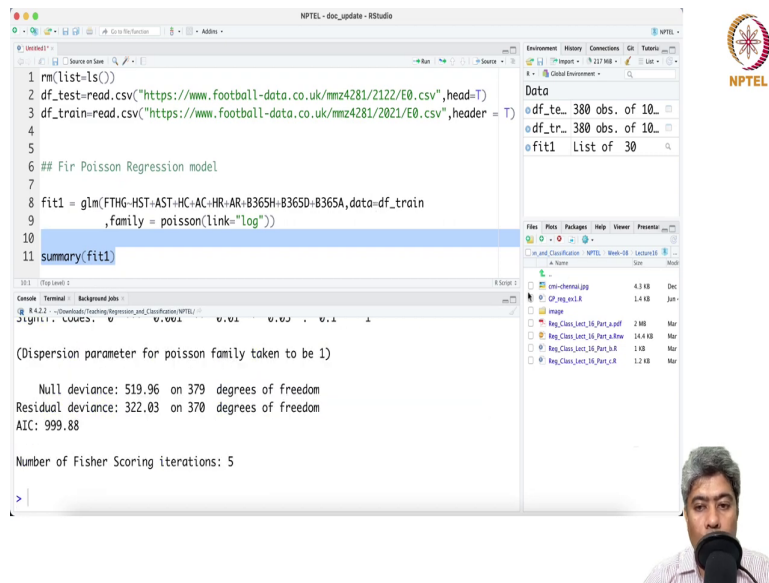
```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
4
5
6 ## Fir Poisson Regression model
7
8 fit1 = glm(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A,data=df_train
9           ,family = poisson(link="log"))
10
11 summary(fit1)
```

```
> rm(list=ls())
> df=read.csv("https://www.football-data.co.uk/mmz4281/2223/E0.csv",head=T)
> View(df)
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
> rm(list=ls())
> df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
> df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
> fit1 = glm(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A,data=df_train
+           ,family = poisson(link="log"))
>
```



Data equal to df train and family, family equal to Poisson link equal to log, ok. So, if we run this and summery equal to fit1.

(Refer Slide Time: 07:13)



The screenshot shows an RStudio interface with the following content:

```
1 rm(list=ls())
2 df_test=read.csv("https://www.football-data.co.uk/mmz4281/2122/E0.csv",head=T)
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
4
5
6 ## Fir Poisson Regression model
7
8 fit1 = glm(FTHG~HST+AST+HC+AC+HR+AR-B365H-B365D+B365A,data=df_train
9           ,family = poisson(link="log"))
10
11 summary(fit1)
```

Console output:

```
(Dispersion parameter for poisson family taken to be 1)

Null deviance: 519.96 on 379 degrees of freedom
Residual deviance: 322.03 on 370 degrees of freedom
AIC: 999.88

Number of Fisher Scoring iterations: 5
```


Environment: Global Environment

Data

- df\_te\_ 380 obs. of 10...
- df\_tr\_ 380 obs. of 10...
- fit1 List of 30

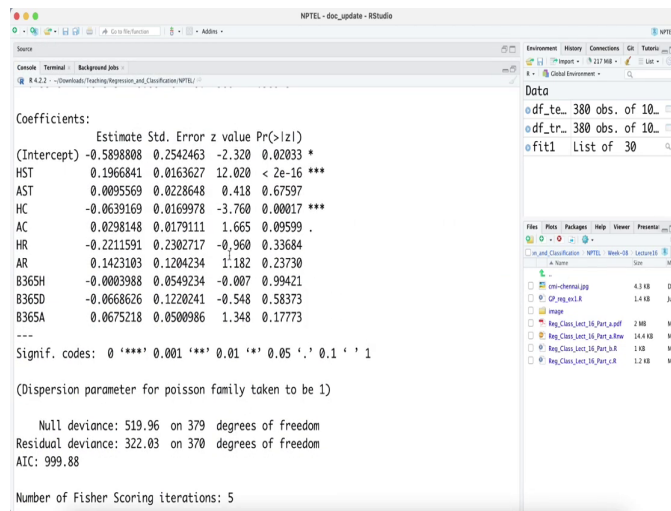
File Edit Packages Help Viewer Presenter

Name	Size	Modif
01_01-Chennai.jpg	4.1 KB	Dec
CP_MPG_v1.8	1.4 KB	Jan
image		
Reg_Class_1st_18_Part_1a.pdf	2.9 KB	Mar
Reg_Class_1st_18_Part_1a.Rnw	18.4 KB	Mar
Reg_Class_1st_18_Part_1a.R	1.8 KB	Mar
Reg_Class_1st_18_Part_1a	1.2 KB	Mar



If you run this.

(Refer Slide Time: 07:15)



```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.5898808  0.2542463  -2.320  0.02033 *
HST          0.1966841  0.0163627  12.020 < 2e-16 ***
AST          0.0095569  0.0228648   0.418  0.67597
HC          -0.0639169  0.0169978  -3.760  0.00017 ***
AC           0.0298148  0.0179111   1.665  0.09599 .
HR          -0.2211591  0.2302717  -0.960  0.33684
AR           0.1423103  0.1204234   1.182  0.23730
B365H       -0.0003988  0.0549234  -0.007  0.99421
B365D       -0.0668626  0.1220241  -0.548  0.58373
B365A        0.0675218  0.0500986   1.348  0.17773
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 519.96 on 379 degrees of freedom
Residual deviance: 322.03 on 370 degrees of freedom
AIC: 999.88

Number of Fisher Scoring iterations: 5
```



So, what we are seeing that home team shot made by the home team and home team how many corners they have made has a significant effect and rest we are not seeing many effect. One possibility could be one possibility could be there are multi collinearity issue. We can, but what we can do?



(Refer Slide Time: 07:44)

NPTEL - dev\_update - RStudio

```
Source
Console Terminal Background Jobs
R 4.2.2 .../Downloads/TrackingRegression_and_Classifiers/npfl/

Deviance Residuals:
  Min       1Q   Median       3Q      Max
-2.24572 -1.06248 -0.00016  0.47426  2.45990

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.58704    0.11369  -5.163 2.42e-07 ***
HST           0.20519    0.01558  13.170 < 2e-16 ***
HC           -0.06777    0.01657  -4.091 4.30e-05 ***
B365A        0.03930    0.01258   3.123 0.00179 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 519.96 on 379 degrees of freedom
Residual deviance: 326.99 on 376 degrees of freedom
AIC: 992.83

Number of Fisher Scoring iterations: 5
> |
```

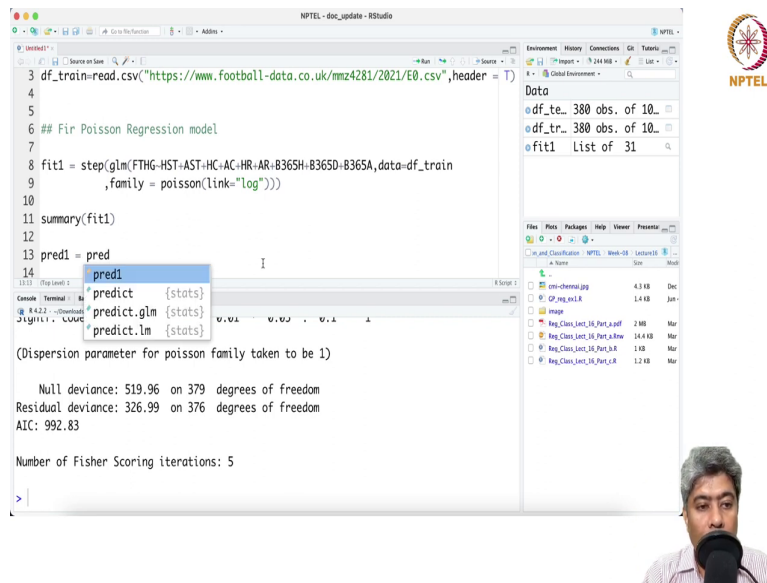
Data  
df\_te\_ 380 obs. of 10...  
df\_tr\_ 380 obs. of 10...  
ofit1 List of 31

File Plot Packages Help Viewer Presenter

File Name	Size	Modif
01_anti-chromal.jpg	4.3 KB	Dec
01_CP_MEG_01.R	1.4 KB	Jan
image		
Reg_Class_Jact_18_Part_1a.pdf	2.9 MB	Mar
Reg_Class_Jact_18_Part_1a.Rnw	18.4 KB	Mar
Reg_Class_Jact_18_Part_1a.R	1.8 KB	Mar
Reg_Class_Jact_18_Part_1a	1.2 KB	Mar

Before that we can just do a stepwise selection method to get the best model. So, if we just get a sort of a based on AIC. So, yeah, now, what is going here? That along with home team number of shots on target by home team and number corner by the home team we are also getting B365A the; that means, what is the away teams bet by B365 betting ratio odds that also playing a role, alright. So, we can we have some model. Let us take this is the model we have.

(Refer Slide Time: 08:32)



The screenshot shows an RStudio interface with the following content:

```
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)
4
5
6 ## Fir Poisson Regression model
7
8 fit1 = step(glm(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A,data=df_train
9             ,family = poisson(link="log")))
10
11 summary(fit1)
12
13 pred1 = pred
14
```

The console output shows the following summary statistics:

```
(Dispersion parameter for poisson family taken to be 1)

Null deviance: 519.96 on 379 degrees of freedom
Residual deviance: 326.99 on 376 degrees of freedom
AIC: 992.83

Number of Fisher Scoring iterations: 5
```

A dropdown menu is open for the variable 'pred1', showing the following options:

- predict {stats}
- predict.glm {stats}
- predict.lm {stats}

The NPTEL logo is visible in the top right corner of the RStudio window.

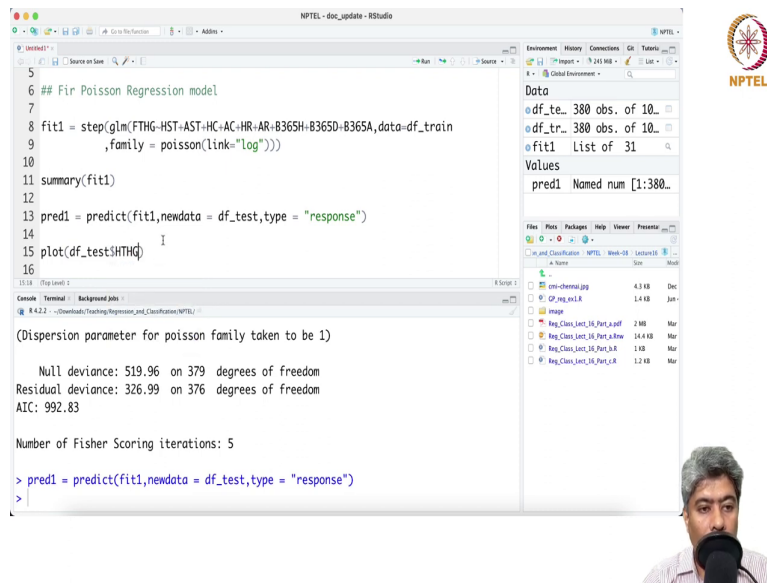
Now, what we can do? We can make the prediction for prediction for the test data set the next seasons data set pred predict predict df fit1 fit1 comma new data equal to df test, df test and type equal to type equal to response, ok.

(Refer Slide Time: 09:06)

```
NPTEL - dev_update - RStudio  
3 df_train=read.csv("https://www.football-data.co.uk/mmz4281/2021/E0.csv",header = T)  
4  
5  
6 ## Fir Poisson Regression model  
7  
8 fit1 = step(glm(FTHG+HST+AST+HC+AC+HR+AR-B365H-B365D-B365A,data=df_train  
9             ,family = poisson(link="log"))  
10  
11 summary(fit1)  
12 |  
13 pred1 = predict(fit1,newdata = df_test,type = "response")  
14  
Console Terminal | Background jobs |  
R 4.2.2 - Downloads/Teaching/Regression_and_Classification/NPTEL/ |  
(Dispersion parameter for poisson family taken to be 1)  
  
Null deviance: 519.96 on 379 degrees of freedom  
Residual deviance: 326.99 on 376 degrees of freedom  
AIC: 992.83  
  
Number of Fisher Scoring iterations: 5  
  
> pred1 = predict(fit1,newdata = df_test,type = "response")  
>
```



(Refer Slide Time: 09:24)



The screenshot shows the RStudio interface with the following content:

```
5  
6 # Fir Poisson Regression model  
7  
8 fit1 = step(glm(FTHG~HST+AST+HC+AC+HR+AR-B365H-B365D-B365A,data=df_train  
9             ,family = poisson(link="log"))  
10  
11 summary(fit1)  
12  
13 pred1 = predict(fit1,newdata = df_test,type = "response")  
14  
15 plot(df_test$FTHG) |  
16
```

Console output:



```
15:14 (Top level) |  
R 4.2.2 - Downloads/Teaching/Regression_and_Classification/NPTEL/ |  
(Dispersion parameter for poisson family taken to be 1)  
  
Null deviance: 519.96 on 379 degrees of freedom  
Residual deviance: 326.99 on 376 degrees of freedom  
AIC: 992.83  
  
Number of Fisher Scoring iterations: 5  
  
> pred1 = predict(fit1,newdata = df_test,type = "response")  
>
```

Environment pane:

Object	Class	Attributes
df_te	data.frame	380 obs. of 10...
df_tr	data.frame	380 obs. of 10...
fit1	glm	List of 31
pred1	Named num	[1:380_

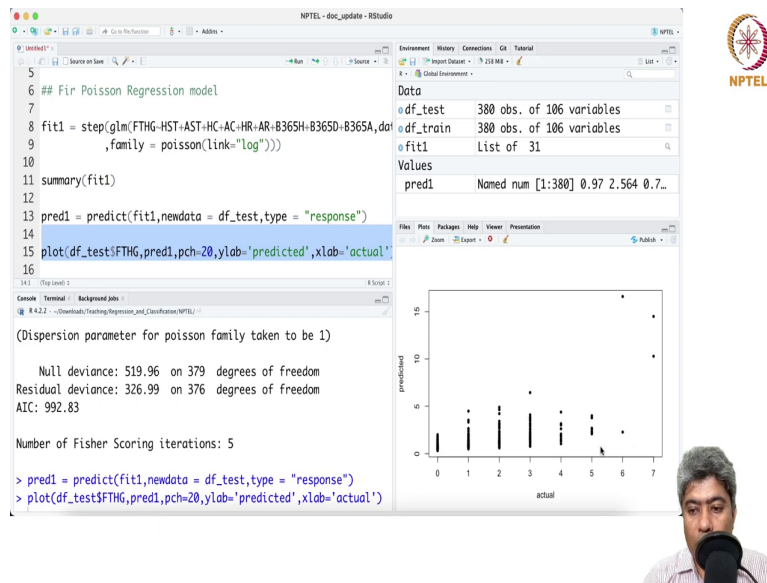
Files pane:

File Name	Size	Modif
fit1-chemical.jpg	4.3 KB	Dec
GP_MG_v1.8	1.4 KB	Jan
image		
Reg_Class_1st_18_Part_1a.pdf	2.9 KB	Mar
Reg_Class_1st_18_Part_1a.Rnw	15.4 KB	Mar
Reg_Class_1st_18_Part_1a.R	1.8 KB	Mar
Reg_Class_1st_18_Part_1a	1.2 KB	Mar



Now, what I can do I can plot the from the df test what is the HTHG or sorry, FTHG full time goals scored by home team and predicted values through predicted values.

(Refer Slide Time: 09:38)



The screenshot displays the RStudio interface. The script editor on the left contains the following R code:

```
5  
6 # Fir Poisson Regression model  
7  
8 fit1 = step(glm(FTHG~HST+AST+HC+AC+HR+AR-B365H-B365D-B365A, data=df_train, family = poisson(link="log"))  
9  
10 summary(fit1)  
11  
12  
13 pred1 = predict(fit1, newdata = df_test, type = "response")  
14  
15 plot(df_test$FTHG, pred1, pch=20, ylab='predicted', xlab='actual')  
16
```

The console on the bottom left shows the output of the model fit:

```
(Dispersion parameter for poisson family taken to be 1)  
  
Null deviance: 519.96 on 379 degrees of freedom  
Residual deviance: 326.99 on 376 degrees of freedom  
AIC: 992.83  
  
Number of Fisher Scoring iterations: 5  
  
> pred1 = predict(fit1, newdata = df_test, type = "response")  
> plot(df_test$FTHG, pred1, pch=20, ylab='predicted', xlab='actual')
```

The environment pane on the right shows the data and fit objects:

```
Data  
df_test 380 obs. of 106 variables  
df_train 380 obs. of 106 variables  
fit1 List of 31  
Values  
pred1 Named num [1:380] 0.97 2.564 0.7...
```

The plot on the bottom right shows a scatter plot of predicted vs actual values. The x-axis is labeled 'actual' and ranges from 0 to 7. The y-axis is labeled 'predicted' and ranges from 0 to 15. The data points are represented by black circles (pch=20). Most points are clustered around an actual value of 3, with some points at actual values of 1, 2, 4, 5, and 6. There are a few points at predicted values of 10 and 15, which are significantly higher than the actual values of 7.

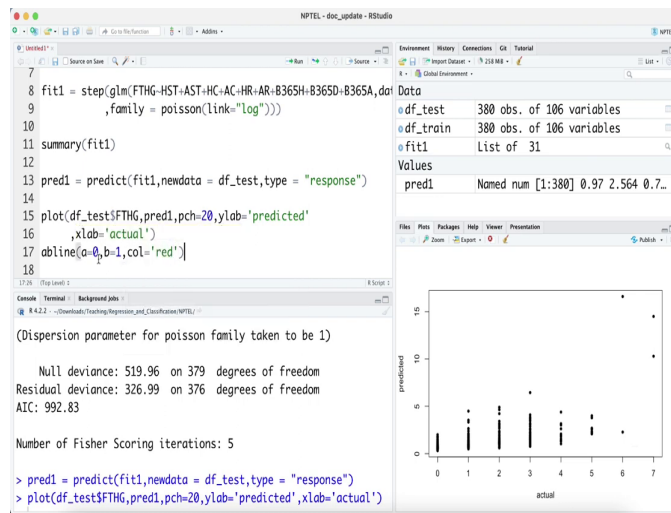
The NPTEL logo is visible in the top right corner of the slide.

So, pch equal to 20, pred equal to ylab equal to predicted predicted and x lab x lab equal to actual.

So, what we are seeing that actual was 7 you know somewhat it has some power, but and also there are cases where they predicted 15 goal, 10 goals it is little bit over estimating and actual was it was 7. So, the model has bit of a over estimating feeling like you know here there are cases where it has predicted 5 goals, but most of the goals were on the 3 goals.

So, it is bit of a predicting over estimating the bias, there is some looks like. So, one possible way is also we can do a abline a equal to 0 and b equal to 1 we put like this then color equal to red that also.

(Refer Slide Time: 11:03)



The screenshot displays the RStudio interface with the following components:

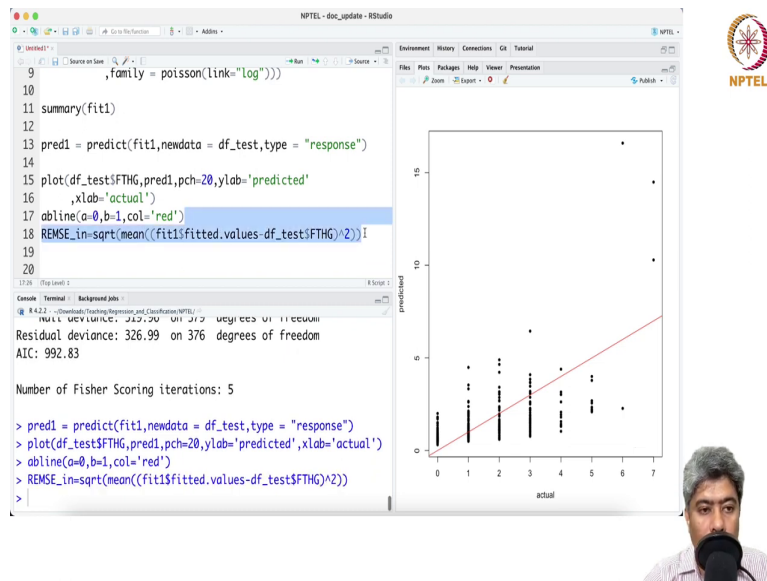
- Source Editor:** Contains R code for fitting a Poisson regression model and generating predictions.

```
7  
8 fit1 = step(glm(FTHG+HST+AST+HC+AC+HR+AR+B365H+B365D+B365A, data, family = poisson(link="log")))  
9  
10  
11 summary(fit1)  
12  
13 pred1 = predict(fit1, newdata = df_test, type = "response")  
14  
15 plot(df_test$FTHG, pred1, pch=20, ylab='predicted',  
16       ,xlab='actual')  
17 abline(a=0, b=1, col='red')  
18
```
- Environment:** Shows the objects created: `df_test` (380 obs. of 106 variables), `df_train` (380 obs. of 106 variables), `fit1` (List of 31), and `pred1` (Named num [1:380] 0.97 2.564 0.7...).
- Console:** Displays the output of the `summary(fit1)` command:

```
(Dispersion parameter for poisson family taken to be 1)  
  
Null deviance: 519.96 on 379 degrees of freedom  
Residual deviance: 326.99 on 376 degrees of freedom  
AIC: 992.83  
  
Number of Fisher Scoring iterations: 5  
  
> pred1 = predict(fit1, newdata = df_test, type = "response")  
> plot(df_test$FTHG, pred1, pch=20, ylab='predicted', xlab='actual')
```
- Plot:** A scatter plot with 'actual' on the x-axis (ranging from 0 to 7) and 'predicted' on the y-axis (ranging from 0 to 15). The data points are represented by black circles (pch=20). A red horizontal line is drawn at y=1, and a red vertical line is drawn at x=1, intersecting at (1,1). The plot shows a positive correlation between actual and predicted values, with some outliers at higher predicted values.



(Refer Slide Time: 11:18)



So, yeah. So, looks like some over estimation some under estimation here there are some under estimations, there are these places there are over estimations are happening, ok. So, this is what we are seeing. Now, what is what we are seeing here is the let us calculate the RMSC because one of the problem you know Poisson regression and the Poisson regression is a statistical model whereas, regression tree random forest are machine learning model for regression.

Now, what happens in the machine learning models there in these machine learning models there is no there are no likelihood you cannot write down the likelihood. Since you do not have a likelihood you cannot calculate AIC, BIC for these decision tree models, random forest models. So, you cannot use AIC, BIC type criteria to compare the random forest or decision tree against Poisson regression.

So, what we can do? We can use root mean square error which can be used for both Poisson regression as well as for the regression tree and random forest. So, we are going to calculate root mean square error for Poisson regression. So, RMSE in equal to. So, first from the fit1 we have to take the fitted values out minus from the df test dollar FTHG square it, take mean and then take the square root this is in sample root mean square error this is for RMSE1, ok.

(Refer Slide Time: 14:14)

The screenshot shows an RStudio session with the following code in the script editor:

```

11 summary(fit1)
12
13 pred1 = predict(fit1, newdata = df_test, type = "response")
14
15 plot(df_test$FTHG, pred1, pch=20, ylab='predicted'
16       , xlab='actual')
17 abline(a=0, b=1, col='red')
18 RMSE1_in=sqrt(mean((fit1$fitted.values-df_test$FTHG)^2))
19 RMSE1_out=sqrt(mean((pred1-df_test$FTHG)^2))
20 c(RMSE1_in, RMSE1_out)
21
22 ## Decision Tree Regression

```

The console output shows the results of the calculations:

```

> RMSE1_in
[1] 1.63096
> RMSE1_in=sqrt(mean((fit1$fitted.values-df_test$FTHG)^2))
> RMSE1_out=sqrt(mean((pred1-df_test$FTHG)^2))
> RMSE1_out
[1] 1.26232
> c(RMSE1_in, RMSE1_out)
RMSE1_in RMSE1_out
1.63096  1.26232

```

The plot on the right shows 'predicted' values on the y-axis (ranging from 0 to 15) and 'actual' values on the x-axis (ranging from 0 to 7). A red regression line is plotted, showing a positive correlation between actual and predicted values. The data points are represented by small black circles.

And then sorry, I think I have RMSE1 and then RMSE1 for out sample what I have to do I have to basically instead of this I have to take the predicted value I already predicted it I have to just go this and which is 1.26.

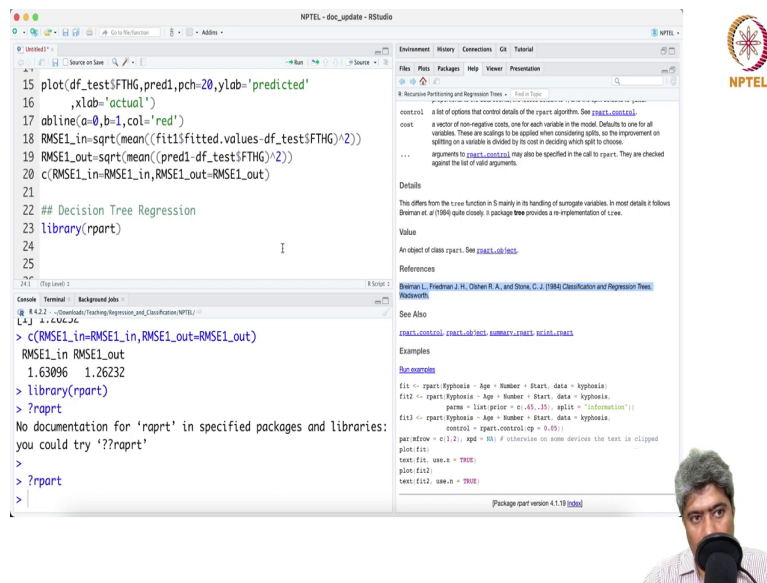
So, in sample there is lot of issues, but in out sample it looks like it is better that is sometime that is bit weird. Because typically in samples are better than the out of the sample, but we



will see how it is doing in the yeah, we will see how it is doing in the regression t let us check how it how it is doing in the regression t and random forest decision tree and random forest.

So, next we will compare the decision tree regression. We will fit Decision Tree Regression.

(Refer Slide Time: 15:28)





The screenshot shows an RStudio interface with the following code in the editor:

```
15 plot(df_test$FTHG, pred1, pch=20, ylab='predicted'
16       , xlab='actual')
17 abline(a=0, b=1, col='red')
18 RMSE1_in = sqrt(mean((fit1$Fitted.values - df_test$FTHG)^2))
19 RMSE1_out = sqrt(mean((pred1 - df_test$FTHG)^2))
20 c(RMSE1_in, RMSE1_out, RMSE1_out - RMSE1_in)
21
22 ## Decision Tree Regression
23 library(rpart)
24
25
```

The console output shows:

```
> c(RMSE1_in=RMSE1_in, RMSE1_out=RMSE1_out)
RMSE1_in RMSE1_out
1.63096  1.26232
> library(rpart)
> ?rpart
No documentation for 'rpart' in specified packages and libraries:
you could try '??rpart'
>
> ?rpart
```

The right-hand pane shows the help page for the `rpart` package, titled "Recursive Partitioning and Regression Trees". It includes details about the `rpart` function, its arguments, and examples of how to use it.

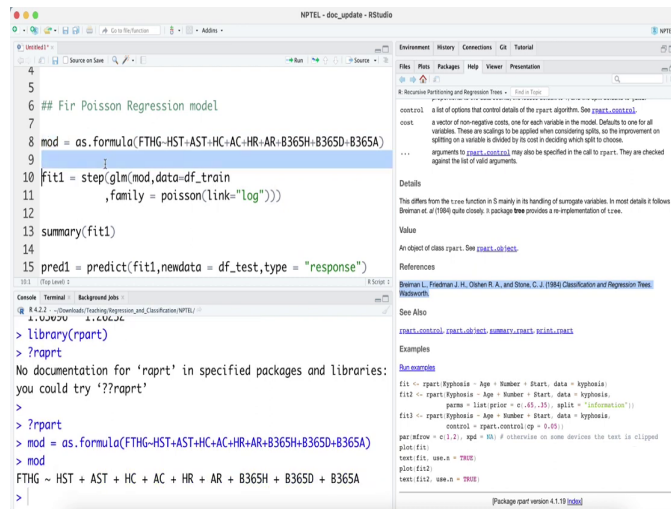


So, in the decision tree I am going to use the library called rpart ok, and if you come question mark rpart. So, it is called recursive partitioning and regression tree, ok. So, you just it is basically a regression tree it fits the essentially tree function in S mainly used to support it follows the Breiman's 1984 quite closely it says.

It most details most detail it follows Breiman et. al and in the main model main model proposed by Leo Breiman in 1984. So, rpart mostly follow that major thing this back paper

classification and regression tree. So, this is being implemented in rpart. So, I am going to use the classical decision tree regression proposed by Leo Breiman implemented in R, ok.

(Refer Slide Time: 16:46)



```
4
5
6 # Fit Poisson Regression model
7
8 mod = as.formula(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A)
9
10 fit1 = step(glm(mod,data=df_train
11             ,family = poisson(link="log")))
12
13 summary(fit1)
14
15 pred1 = predict(fit1,newdata = df_test,type = "response")
```

```
@ 4.4.2 ... Downloads/Teaching/Regression_and_Classification/NPTEL/
A. 4.4.4.2.1.1 A. 4.4.4.4.2.1.1
```

```
> library(rpart)
> ?rpart
No documentation for 'rpart' in specified packages and libraries:
you could try '??rpart'
>
> ?rpart
> mod = as.formula(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A)
> mod
FTHG ~ HST + AST + HC + AC + HR + AR + B365H + B365D + B365A
>
```

Environment: History Connections Git Tutorial

Files Plots Packages Help Viewer Presentation

4.4.4.2.1.1

fit1: A stepAes object of class "stepAes".

Details

This differs from the lme4 function in 5 mainly in its handling of surrogate variables. In most details it follows Breiman et al (1984) quite closely. A package [tree](#) provides a re-implementation of lme4.

Value

An object of class rpart. See [rpart.object](#).

References

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984) Classification and Regression Trees. Wadsworth.

See Also

[rpart.control](#), [rpart.object.summary](#), [rpart.print](#), [rpart](#)

Examples

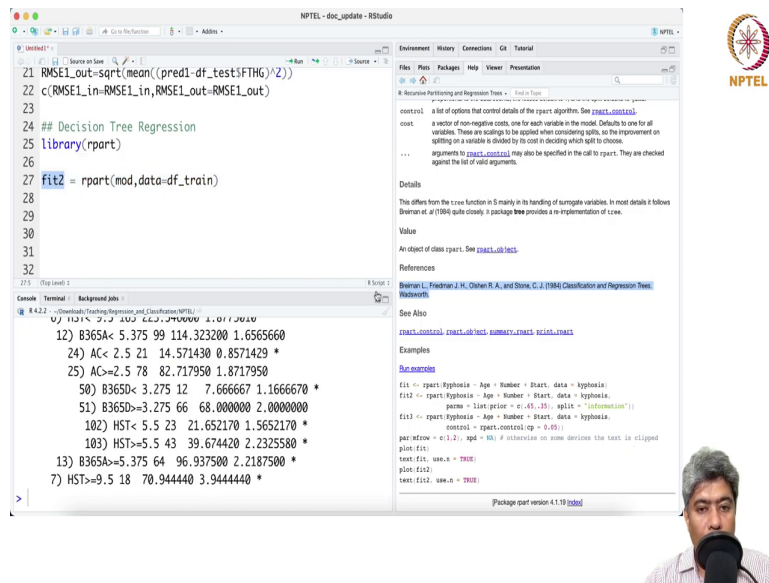
```
R console
fit1 <- rpart(kyphosis ~ Age + Number + Start, data = kyphosis)
fit2 <- rpart(kyphosis ~ Age + Number + Start, data = kyphosis,
  param = list(prior = c(.65, .35), split = "information"))
fit3 <- rpart(kyphosis ~ Age + Number + Start, data = kyphosis,
  control = rpart.control(cp = 0.5))
par(mfrow = c(1,2), mfd = 2) # observe on some devices the text is clipped
plot(fit)
text(fit, use.n = TRUE)
plot(fit2)
text(fit2, use.n = TRUE)
```

Package: rpart version 4.1.19 [source](#)



So, I am going to fit my second model the decision tree rpart. And then I am what I am going to do actually you know what? We can we are going to have the same model mod equal to at the rate formula sorry, s dot formula and I can have it and if I just run it here I think that will be, yeah.


(Refer Slide Time: 17:33)



```
21 RMSE1_out=sqrt(mean((pred1_df_test$FTHG)^2))
22 c(RMSE1_in-RMSE1_in,RMSE1_out-RMSE1_out)
23
24 # Decision Tree Regression
25 library(rpart)
26
27 fit2 = rpart(mod,data=df_train)
28
29
30
31
32
```

```
12) B365A< 5.375 99 114.323200 1.6565660
24) AC< 2.5 21 14.571430 0.8571429 *
25) AC>=2.5 78 82.717950 1.8717950
50) B365D< 3.275 12 7.666667 1.1666670 *
51) B365D>=3.275 66 68.000000 2.0000000
102) HST< 5.5 23 21.652170 1.5652170 *
103) HST>=5.5 43 39.674420 2.2325580 *
13) B365A>=5.375 64 96.937500 2.2187500 *
7) HST>=9.5 18 70.944440 3.9444440 *
```

NPTEL



I think it is working fine. So, I will just call this model here, ok. If you run this model you can see this is the model I am going to fit and data equal to train ok, df train data equal to df train, ok. Now, if you run this. So, if you now run it.

(Refer Slide Time: 18:07)

```
NPTEL - dee_update - RStudio

> fit2 = rpart(mod, data=df_train)
> fit2
n= 380

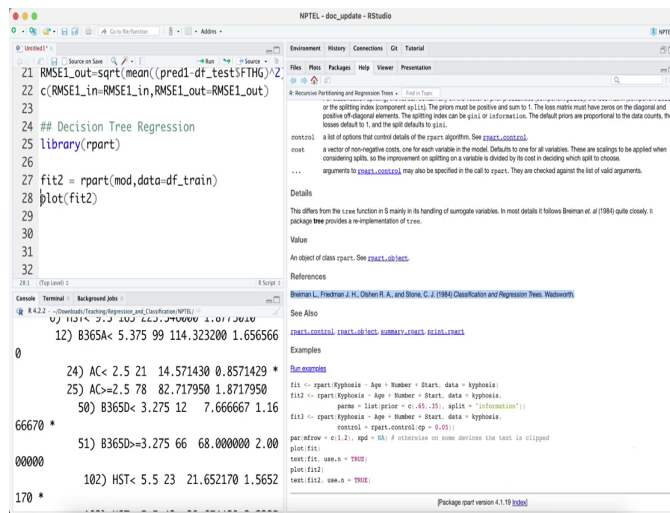
node), split, n, deviance, yval
* denotes terminal node

1) root 380 660.747400 1.3526320
2) HST< 4.5 199 112.683400 0.6884422
4) HST< 2.5 86 29.209300 0.4418605 *
5) HST>=2.5 113 74.265490 0.8761062 *
3) HST>=4.5 181 363.756900 2.0828730
6) HST< 9.5 163 223.546000 1.8773010
12) B365A< 5.375 99 114.323200 1.6565660
24) A< 2.5 21 14.571430 0.8571429 *
25) A<=2.5 78 82.717950 1.8717950
50) B365D< 3.275 12 7.666667 1.1666670 *
51) B365D>=3.275 66 68.000000 2.0000000
102) HST< 5.5 23 21.652170 1.5652170 *
103) HST>=5.5 43 39.674420 2.2325580 *
13) B365A>=5.375 64 96.937500 2.2187500 *
7) HST>=9.5 18 70.944440 3.9444440 *
```



You will see that at the root it is saying what should be what it should does and then if HST equal to is less than 4.5 it should do something HST is less than 2.5, it should do something in this way. The decision tree is being made you can if you do plot fit2. So, generally it plot the decision tree nicely.

(Refer Slide Time: 18:41)



```
21 RMSE1_out=sqrt(mean((pred1_df_tests$FTHG)^2
22 c(RMSE1_in-RMSE1_in,RMSE1_out-RMSE1_out)
23
24 # Decision Tree Regression
25 library(rpart)
26
27 fit2 = rpart(mod,data_df_train)
28 plot(fit2)
29
30
31
32
```

Console

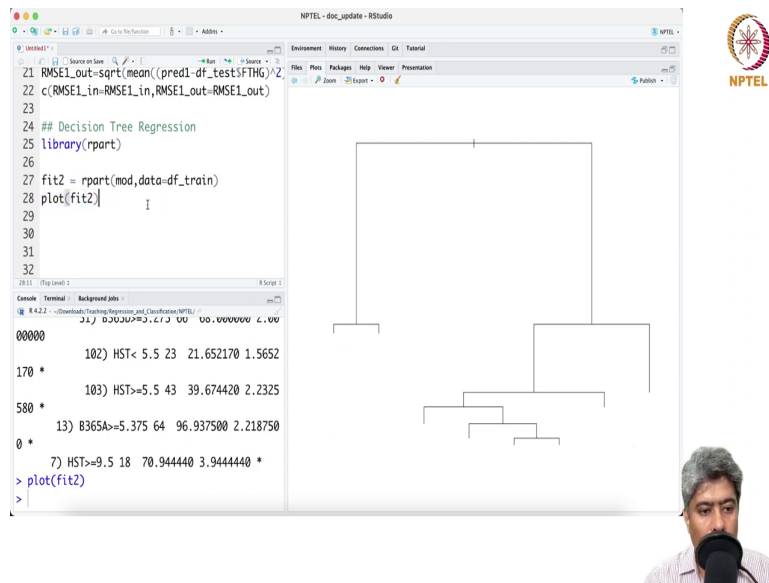
```
12) B365A< 5.375 99 114.323200 1.656566
0
24) A<= 2.5 21 14.571430 0.8571429 *
25) A<= 2.5 78 82.717950 1.8717950
50) B365D< 3.275 12 7.666667 1.16
66670 *
80000
102) HST< 5.5 23 21.652170 1.5652
170 *
```

Package rpart version 4.1-19 [Index](#)



Let me try, let me try.

(Refer Slide Time: 18:43)



The screenshot shows an RStudio window titled "NPTEL-dec\_update - RStudio". The script editor contains the following R code:


```
21 RMSE1_out=sqrt(mean((pred1_df_tests$FTHG)^2
22 c(RMSE1_in-RMSE1_in,RMSE1_out-RMSE1_out)
23
24 # Decision Tree Regression
25 library(rpart)
26
27 fit2 = rpart(mod,data=df_train)
28 plot(fit2)
29
30
31
32
```

The console output shows the following results:

```
00000
170 * 102) HST<=5.5 23 21.652170 1.5652
580 * 103) HST>=5.5 43 39.674420 2.2325
0 * 13) B365A>=5.375 64 96.937500 2.218750
7) HST>=9.5 18 70.944440 3.944440 *
```

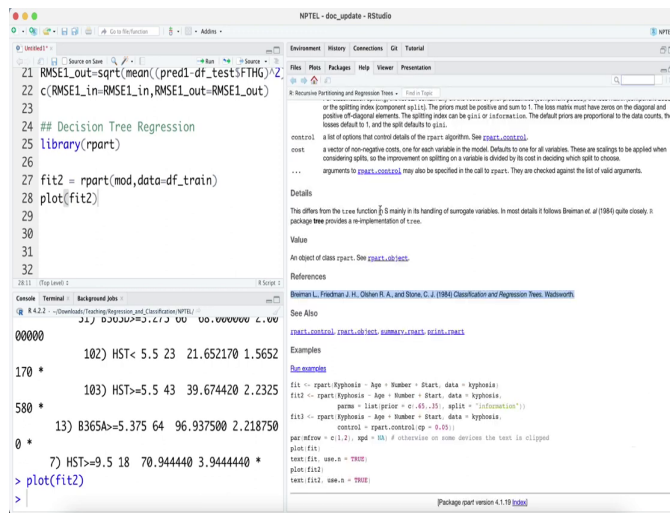
The plot area shows a decision tree structure with a root node and several internal nodes, but no numerical values or labels are visible on the plot.

The NPTEL logo is visible in the top right corner of the RStudio window.



So, yeah, it is it does a decision tree, but for some reason it is not showing the plots.

(Refer Slide Time: 18:54)



```
21 RMSE1_out=sqrt(mean((pred1_df_tests$FTHG)^2
22 c(RMSE1_in-RMSE1_in,RMSE1_out-RMSE1_out)
23
24 # Decision Tree Regression
25 library(rpart)
26
27 fit2 = rpart(mod,data=df_train)
28 plot(fit2)
29
30
31
32
```

00000  
182) HST< 5.5 23 21.652170 1.5652  
170 \*  
103) HST>=5.5 43 39.674420 2.2325  
580 \*  
13) B365A>=5.375 64 96.937500 2.218750  
0 \*  
7) HST>=9.5 18 70.944440 3.944440 \*

```
> plot(fit2)
>
```

Package rpart version 4.1-19 [Index](#)



So, ok, it is and I have to say text use n dot true, alright.

(Refer Slide Time: 19:08)

```
NPTEL - dee_update - RStudio
21 RMSE1_out=sqrt(mean((pred1_df_test$FTHG)^2
22 c(RMSE1_in-RMSE1_in,RMSE1_out-RMSE1_out)
23
24 # Decision Tree Regression
25 library(rpart)
26
27 fit2 = rpart(mod,data_df_train)
28 plot(fit2)
29 text
30 text(x,...)
31 *text.default {graphics}
32 *textConnection {base}
33 *textConnectionValue {base}
34 *textshaping
00000
170 * 102) HST< 5.5 23 21.652170 1.5652
580 * 103) HST>=5.5 43 39.674420 2.2325
0 * 13) B365A>=5.375 64 96.937500 2.218750
7) HST>=9.5 18 70.944440 3.9444440 *
> plot(fit2)
>
```

NPTEL

Examples  
fit1 <- rpart(kyphosis ~ Age + Number + Start, data = kyphosis)  
fit2 <- rpart(kyphosis ~ Age + Number + Start, data = kyphosis,  
          parms = list(prune = c(.45, .35), split = "information"))  
fit3 <- rpart(kyphosis ~ Age + Number + Start, data = kyphosis,  
          control = rpart.control(cp = 0.05))  
par(mfrow = c(1,2), mfcol = 2) # otherwise on some devices the text is clipped  
plot(fit1)  
text(fit1, use.n = TRUE)  
plot(fit2)  
text(fit2, use.n = TRUE)

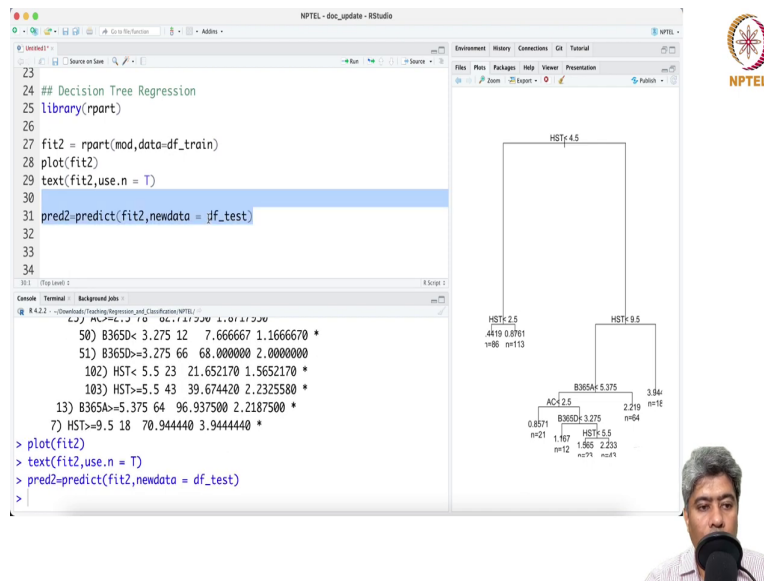
Package rpart version 4.1-19 [Index](#)



So, I will say text.



(Refer Slide Time: 19:11)



The screenshot displays the RStudio interface. The script editor on the left contains the following R code:

```
23
24 # Decision Tree Regression
25 library(rpart)
26
27 fit2 = rpart(mod, data=df_train)
28 plot(fit2)
29 text(fit2, use.n = T)
30
31 pred2=predict(fit2, newdata = df_test)
32
33
34
```

The console on the bottom left shows the output of the commands:

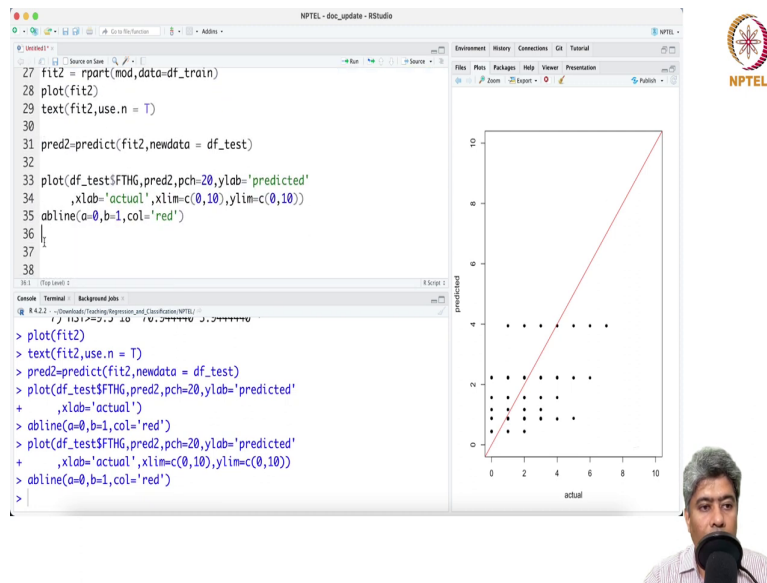
```
R 4.2.2 -- Downloaded Teaching, Regression, and Classification (NPTEL)
C:\J> Rscript.exe 19_11_04.R
50) B365D< 3.275 12 7.666667 1.1666670 *
51) B365D>=3.275 66 68.000000 2.0000000
102) HST< 5.5 23 21.652170 1.5652170 *
103) HST>=5.5 43 39.674420 2.2325580 *
13) B365A>=5.375 64 96.937500 2.2187500 *
7) HST>=9.5 18 70.944440 3.9444440 *
> plot(fit2)
> text(fit2, use.n = T)
> pred2=predict(fit2, newdata = df_test)
>
```

The plot on the right shows a decision tree structure. The root node is HST<4.5. The left branch is HST<2.5 (n=113) and the right branch is HST<=9.5 (n=104). The HST<2.5 branch further splits into AC<2.5 (n=21) and B365D<=3.275 (n=92). The AC<2.5 branch splits into 0.8571 (n=12) and 1.167 (n=9). The B365D<=3.275 branch splits into 0.8571 (n=12) and 1.167 (n=9). The HST<=9.5 branch splits into B365A<=5.375 (n=64) and 3.944 (n=18). The B365A<=5.375 branch splits into 0.8571 (n=12) and 1.167 (n=9). The HST<=9.5 branch also splits into HST<=5.5 (n=23) and HST<=9.5 (n=43).

Text fit2 use dot n equal to true let me try, yeah, alright. So, it is saying that if HST is less than 4.5 it should go in this way otherwise it should go in this way. If it is less than 4 to 2.5 then with 0.4419 it take a decision it is I am not sure how this guys are doing these things, ok.

Let us do the prediction, let us do the prediction first then we will understand how it is doing. So, pred2 and pred2 equal to predict predict fit2 new data equal to df test, ok. If I just do that and now what I am going to do, I am going to just simply plot this guy with rate 2 and let us see how it is doing, ok.

(Refer Slide Time: 20:34)

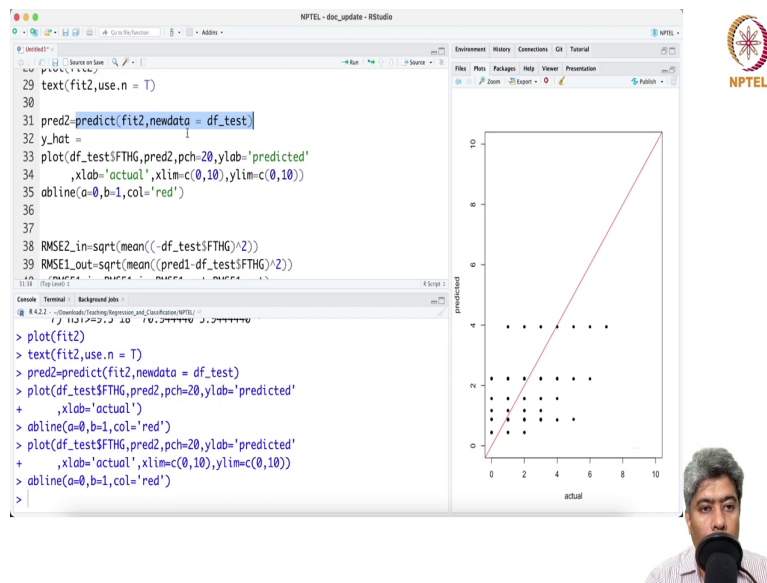


Interesting. So, the predicted values are somewhat constant you know for different 2 it is it is not much varying, but it is actually constant. And that is what we were saying like it should be constant, it takes within a region it takes a some bunch of constant values. But one interesting thing is it looks like it is it is not over estimating maybe it is little bit under estimating.

You can see that it is going somewhere between 0 the averages are going between half to 4 it is not making much differences. So, abline if we draw abline a equal to 0 and b equal to 1 with color equal to red, ok. So, yeah, it is not it is bit of a my feeling is it is bit of under estimating. Because you can see that x is taking range xlim is taking range between 0 to say 10 and ylim is taking if I just say the same scaling effect if we let us see if it what happens if it is gets how see, yeah.

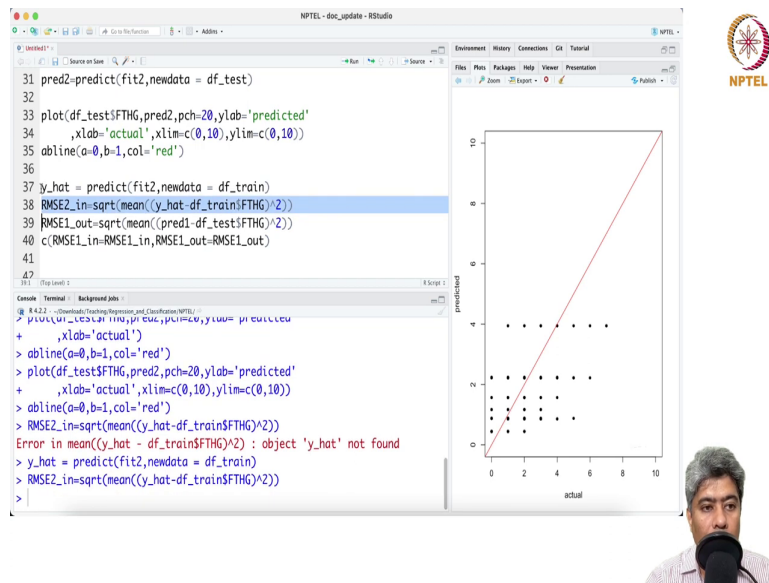
So, it is not even going up to the 6, 7, 8 whereas, in actual values are going up to this and it is trying to a little bit, it is trying to it is as a under estimation. So, that happens that is fine. Let us do some calculate the RMSE for in sample and out of the sample, ok. So, RMSE for the second model and instead of that what we have to do, ok.

(Refer Slide Time: 22:45)



So, what I will do is red I will do a y hat here, y hat and fit to instead of test I will do train, ok. Let me just here and then y hat will be here and instead of test I have to take train FTHG, I have to first run the y hat of course.



(Refer Slide Time: 23:13)



The screenshot shows an RStudio window with the following R code in the editor:

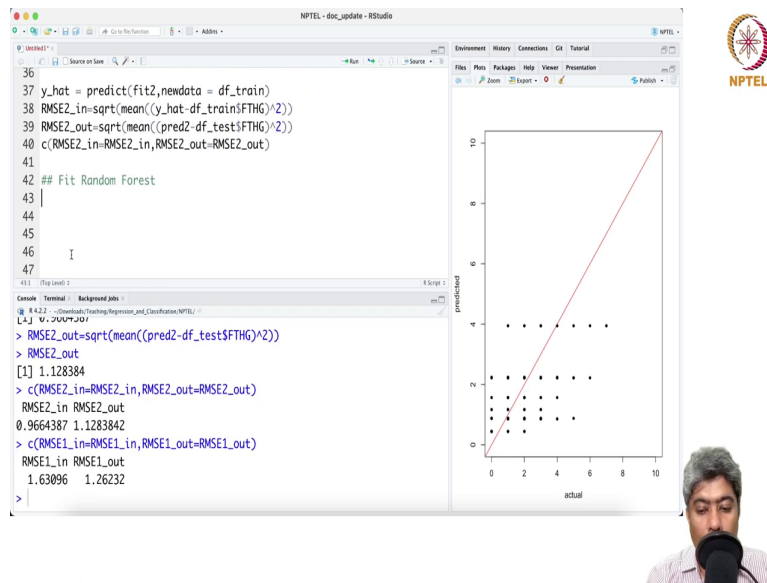
```
31 pred2=predict(fit2,newdata = df_test)
32
33 plot(df_test$FTHG,pred2,pch=20,ylab='predicted'
34       ,xlab='actual',xlim=c(0,10),ylim=c(0,10))
35 abline(a=0,b=1,col='red')
36
37 y_hat = predict(fit2,newdata = df_train)
38 RMSE2_in=sqrt(mean((y_hat-df_train$FTHG)^2))
39 RMSE1_out=sqrt(mean((pred1-df_test$FTHG)^2))
40 c(RMSE1_in-RMSE1_in,RMSE1_out-RMSE1_out)
41
42
43
```

The console shows the execution of the code, including an error message: "Error in mean((y\_hat - df\_train\$FTHG)^2) : object 'y\_hat' not found". The plot on the right shows a scatter plot of predicted values (y-axis) versus actual values (x-axis). The data points are black dots, and a red diagonal line represents the identity function (y=x). The axes range from 0 to 10.



And then this and then RMSE2 out pred2 and this now this out is 1.12. So, RMSE2 in and RMSE2 out is 2 out, ok. So, 0.96 and 1.12 whereas, we are.

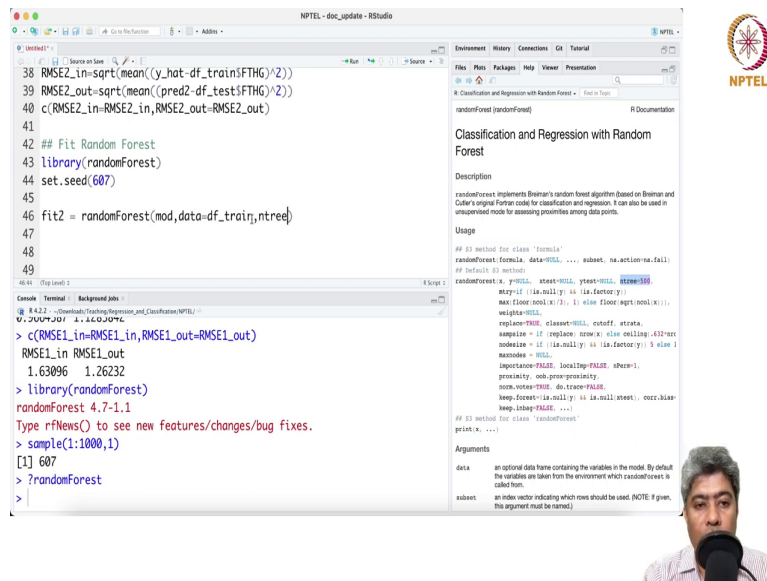
(Refer Slide Time: 23:52)



It was quite improvement if you see in sample Poisson was very high, but in sample is doing fit in out of the sample Poisson is doing still slightly better 1.26 whereas, oh no, decision tree is doing better because it is going to 1.128 which is smaller than 1.262. So, decision tree is doing better than the Poisson regression, ok.

So, now finally, finally, we will do, but we can see a little bit over estimation as well because in sample RMSE is 0.966 whereas, out of the sample is 1.12, it is slightly over estimation is happening. But in Poisson regression in sample is very high and out of the sample is bit low. So, that is also bit of a weird I do not see why it is happening, but let us see what is there, ok. So, let me fit the random forest, random forest.

(Refer Slide Time: 25:17)





The screenshot shows an RStudio window with the following R code in the editor:

```
38 RMSE2_in=sqrt(mean((y_hat-df_train$FTHG)^2))
39 RMSE2_out=sqrt(mean((pred2-df_test$FTHG)^2))
40 c(RMSE2_in,RMSE2_out,RMSE2_out)
41
42 ## Fit Random Forest
43 library(randomForest)
44 set.seed(607)
45
46 fit2 = randomForest(mod,data=df_train,ntree)
47
48
49
```

The console output shows:

```
> c(RMSE1_in,RMSE1_out,RMSE1_out-RMSE1_out)
RMSE1_in RMSE1_out
1.63096 1.26232
> library(randomForest)
randomForest 4.7-1.1
Type rNews() to see new features/changes/bug fixes.
> sample(1:1000,1)
[1] 607
> ?randomForest
>
```

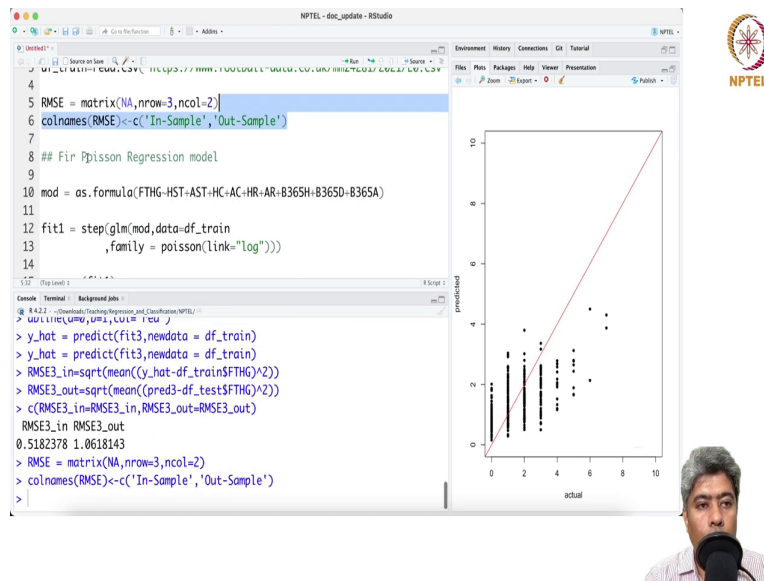
The right pane shows the documentation for the randomForest package, including a description and usage instructions.



So, what I will do first I will call the library that is of library called randomForest, ok. And now since it will do sort of random sampling of the things that we will do sample 1 is to 1000 comma 1. So, 607 this is the set I am going to use.

And then what I am going to do, I am going to fit the third model instead copy this instead of that randomForest mod hm. Of course, one more thing I have to check few things randomForest. So, it is saying y NULL x test y test and then entry is number of trees that you have to give. So, maybe I will give 1000 and see fit3.

(Refer Slide Time: 26:32)





The screenshot shows an RStudio window with the following R code in the editor:

```
4  
5 RMSE = matrix(NA,nrow=3,ncol=2)  
6 colnames(RMSE)<-c('In-Sample','Out-Sample')  
7  
8 ## Fir Ppisson Regression model  
9  
10 mod = as.formula(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A)  
11  
12 fit1 = step(glm(mod,data=df_train  
13             ,family = poisson(link="log")))  
14
```

The console shows the execution of the following commands:

```
> y_hat = predict(fit3,newdata = df_train)  
> y_hat = predict(fit3,newdata = df_train)  
> RMSE3_in=sqrt(mean((y_hat-df_train$FTHG)^2))  
> RMSE3_out=sqrt(mean((pred3-df_train$FTHG)^2))  
> c(RMSE3_in=RMSE3_in,RMSE3_out=RMSE3_out)  
RMSE3_in RMSE3_out  
0.5182378 1.0618143  
> RMSE = matrix(NA,nrow=3,ncol=2)  
> colnames(RMSE)<-c('In-Sample','Out-Sample')  
>
```

The plot on the right shows predicted values on the y-axis and actual values on the x-axis, both ranging from 0 to 10. A red diagonal line represents the identity function (y=x). Data points are plotted as vertical bars with dots at the top, showing a general positive correlation but with some underestimation at higher values.



This is my third model and there are some other things are also there. So, what I will do, we will let us just keep it in there and ok, it does very good job if you fast it calculate things very fast. So, what I am going to do? I am going to copy this and first thing I will do is predict the my predict 3, ok. And let us see how predict does, ok. So, predict 3, ok.

So, looks like also some kind of under estimation because in the eighth prediction is in the out of the sample prediction it is not so high, ok. So, at the higher level it is some kind of under estimation is happening on the model site. Then what I am going to do I am going to calculate the RMSE for the third model y hat equal to fit3; obviously, this and then whatever the th this is my third RMSE, this is the third RMSE with pred 3rd prediction this is 3 this is 3, 3 and 3, ok, nice.

So, clearly the random forest is winning hands down, ok. So, let me write it down. So, let me do one thing, let me just define a matrix called RMSE, ok. RMSE equal to matrix, ok. And NA nrow equal to 3 ncol equal to 2 col names equal to RMSE equal to In sample In sample and Out sample, ok.

(Refer Slide Time: 29:28)

The screenshot shows an RStudio session with the following code and output:

```

54 plot(df_test$FTHG, preds, pch=c(0,10), ylab='predicted'
55       ,xlab='actual', xlim=c(0,10), ylim=c(0,10))
56 abline(a=0, b=1, col='red')
57
58 y_hat = predict(fit3, newdata = df_train)
59 RMSE3_in=sqrt(mean((y_hat-df_train$FTHG)^2))
60 RMSE3_out=sqrt(mean((pred3-df_test$FTHG)^2))
61 RM=c(RMSE3_in=RMSE3_in, RMSE3_out=RMSE3_out)
62
63
64
65

```

The console output shows the calculation of RMSE for three models:

```

RMSE1_in RMSE1_out
1.63096  1.26232
> RMSE[1,]=c(RMSE1_in=RMSE1_in, RMSE1_out=RMSE1_out)
> RMSE
      In-Sample Out-Sample
Poisson Regression 1.63096  1.26232
Decision Tree      NA      NA
Random Forest      NA      NA
> RMSE[2,]=c(RMSE2_in=RMSE2_in, RMSE2_out=RMSE2_out)
>

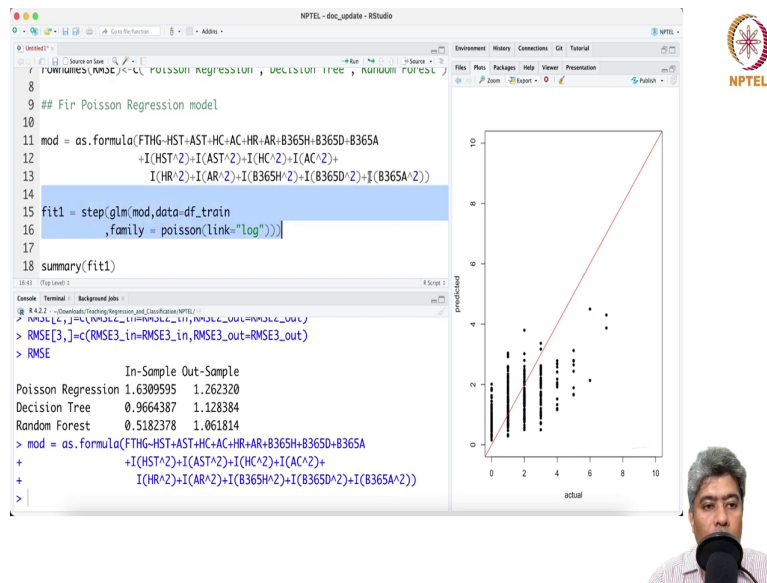
```

The plot shows predicted values on the y-axis and actual values on the x-axis, both ranging from 0 to 10. A red diagonal line represents the identity function (y=x). Data points are plotted as small circles, with vertical error bars indicating uncertainty. The points are clustered around the diagonal line, showing a strong positive correlation between predicted and actual values.

So, this is In sample this is Out sample and row names equal to RMSE. If is first model was Poisson regression comma second model is decision tree and the third model is what is that third model is random forest, ok. And now what I am going to do I am going to put that in the 1, ok. So, yeah, and then going to put that 2 comma equal to this and finally, RMSE3 comma equal to.



(Refer Slide Time: 30:59)



Now, if I just run you can see in sample it is dropping constantly out of the sample also it dropped and Random Forest has 1 hands down, though we are seeing quite a bit of over fitting in both Decision Tree and Random Forest because Out sample RMSE is higher than the In sample. But in out of the sample the root mean square error is significantly lower, ok.

When we plotting this, we are seeing a bit of a underestimation is going on in a conservative site, but random forest is site sort of winning hands down, alright. So, that is how typically you do compare across the statistical models and the machine learning models.

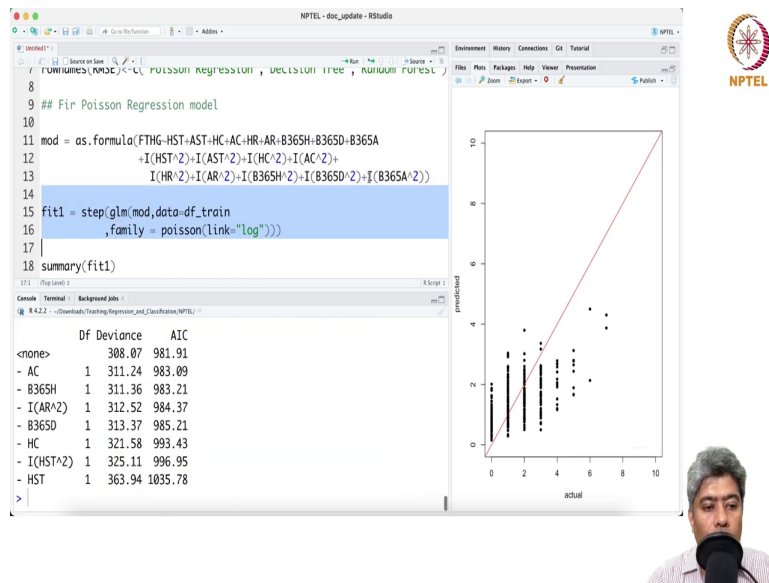
Remember that comparing Poisson Regression with Decision Tree and Random Forest is a bit of a unfair, because if you look into the model Poisson Regression model what I am doing is I am just fitting a linear hyperplane and I have not add any engineering in feature engineering.

If I add feature engineering perhaps possible that Poisson Regression model will start improving ok, whereas, Decision Tree and Random Forest the way the algorithms are being developed it will automatically try to fit the model in a way it will the algorithm will capture the non-monotonic non-linear behavior between the x and y in the higher dimension. So, this is a very strong very flexible model Decision Tree and Random Forest, Poisson Regression in that sense is a much more rigid conservative model.

So, in that sense it is not perhaps a very fair fight if you want to really apple to apple to apple thing comparison probably, we have to add few more lines like you know in feature engineering with HST square plus I and AST square plus I HC square plus I AC square plus I HR square plus I AR square plus I B365H square plus B365D square.

So, I am doing some feature engineering here just to check I do not know whether it will help the Poisson regression model or not, but you can try always try some feature engineering.

(Refer Slide Time: 34:26)





The screenshot shows an RStudio window with the following code in the editor:

```
8  
9 ## Fir Poisson Regression model  
10  
11 mod = as.formula(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A  
12 +I(HST^2)+I(AST^2)+I(HC^2)+I(AC^2)+  
13 I(HR^2)+I(AR^2)+I(B365H^2)+I(B365D^2)+I(B365A^2))  
14  
15 fit1 = step(glm(mod,data=df_train  
16 ,family = poisson(link="log")))  
17  
18 summary(fit1)
```

The console output shows the following summary for the fit:

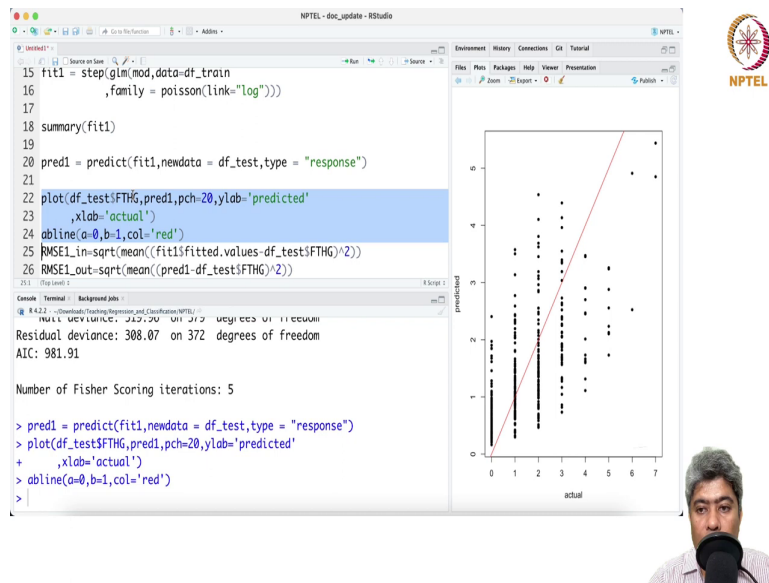
	Df	Deviance	AIC
<none>		308.07	981.91
- AC	1	311.24	983.09
- B365H	1	311.36	983.21
- I(AR^2)	1	312.52	984.37
- B365D	1	313.37	985.21
- HC	1	321.58	993.43
- I(HST^2)	1	325.11	996.95
- HST	1	363.94	1035.78

The plot on the right shows predicted values on the y-axis and actual values on the x-axis, both ranging from 0 to 10. A diagonal line represents the identity function (y=x). Data points are plotted as small black dots, and vertical error bars are shown for each point. The points are generally clustered around the diagonal line, indicating a good fit.



And if you run this and let us see summary. And you can see that some of the you know more features are now higher order features are now actually quite significant. Now, if you run this and let see how.

(Refer Slide Time: 34:50)



The image shows an RStudio window titled "NPTEL - dev\_update - RStudio". The script editor contains the following R code:

```
15 fit1 = step(glm(mod,data=df_train
16             ,family = poisson(link="log")))
17
18 summary(fit1)
19
20 pred1 = predict(fit1,newdata = df_test,type = "response")
21
22 plot(df_test$FTHG,pred1,pch=20,ylab='predicted'
23      ,xlab='actual')
24 abline(a=0,b=1,col='red')
25 RMSE1_in=sqrt(mean((fit1$fitted.values-df_test$FTHG)^2))
26 RMSE1_out=sqrt(mean((pred1-df_test$FTHG)^2))
```



The console output shows:

```
Residual deviance: 308.07 on 372 degrees of freedom
AIC: 981.91

Number of Fisher Scoring iterations: 5

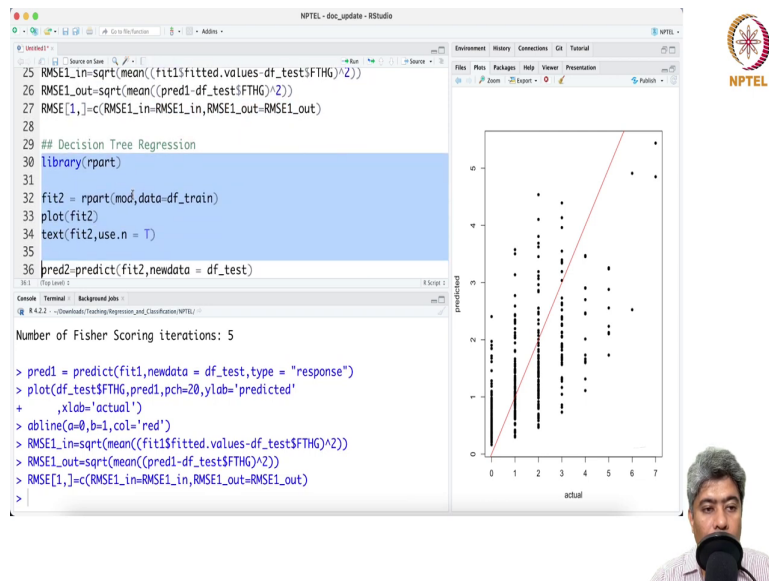
> pred1 = predict(fit1,newdata = df_test,type = "response")
> plot(df_test$FTHG,pred1,pch=20,ylab='predicted'
+      ,xlab='actual')
> abline(a=0,b=1,col='red')
>
```

The plot on the right shows a scatter plot of predicted values (y-axis) versus actual values (x-axis). The data points are represented by black dots (pch=20). A red diagonal line (abline(a=0,b=1)) represents the identity line where predicted values equal actual values. The plot shows a strong positive correlation between actual and predicted values, indicating a good fit of the model.



So, now it is looks like doing better.

(Refer Slide Time: 34:58)





The screenshot displays the RStudio interface with the following R code in the editor:

```
25 RMSE1_in=sqrt(mean((fit1$fitted.values-df_test$FTHG)^2))
26 RMSE1_out=sqrt(mean((pred1-df_test$FTHG)^2))
27 RMSE[1,]=c(RMSE1_in,RMSE1_out-RMSE1_out)
28
29 ## Decision Tree Regression
30 library(rpart)
31
32 fit2 = rpart(mod,data=df_train)
33 plot(fit2)
34 text(fit2,use.n = T)
35
36 pred2=predict(fit2,newdata = df_test)
```

The console output shows:

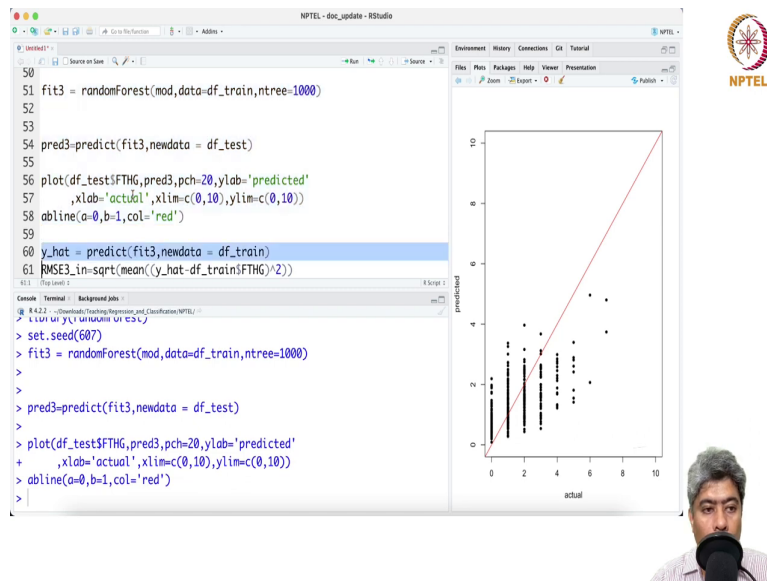
```
Number of Fisher Scoring iterations: 5
> pred1 = predict(fit1,newdata = df_test,type = "response")
> plot(df_test$FTHG,pred1,pch=20,ylab='predicted'
+ ,xlab='actual')
> abline(a=0,b=1,col='red')
> RMSE1_in=sqrt(mean((fit1$fitted.values-df_test$FTHG)^2))
> RMSE1_out=sqrt(mean((pred1-df_test$FTHG)^2))
> RMSE[1,]=c(RMSE1_in,RMSE1_out-RMSE1_out)
>
```

The plot on the right shows a scatter plot of predicted values (y-axis) versus actual values (x-axis). The data points are represented by small black circles. A red diagonal line (y=x) is drawn across the plot, indicating a perfect prediction. The x-axis ranges from 0 to 7, and the y-axis ranges from 0 to 6. The data points are scattered around the red line, showing some deviation from the ideal prediction.



If I run the same model and run me let us run these models, ok.

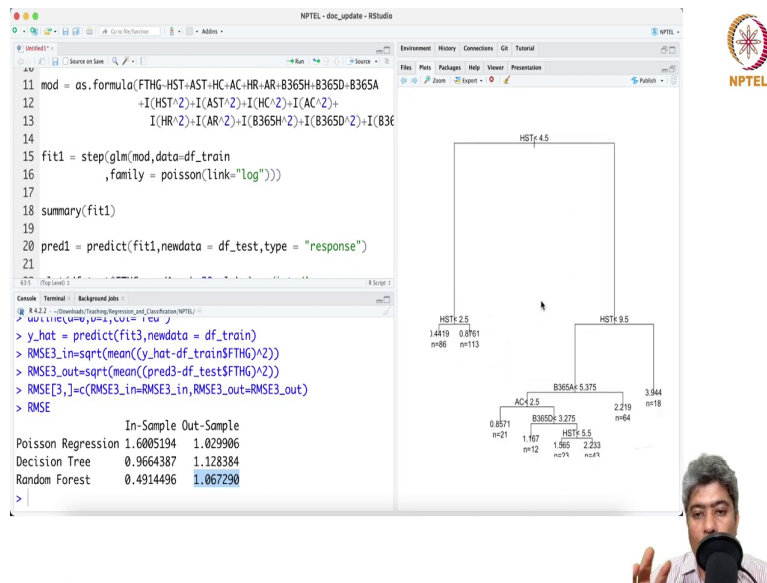
(Refer Slide Time: 35:18)



And now if I compare now, you see Poisson after adding the features Poisson Regression is doing even better than the higher order features, Poisson Regression is doing better than the Decision Tree. In fact, it is doing better than even Random Forest.

So, a simple Poisson Regression can do even better than machine learning model if you add the features engineered features correctly. So, sometimes simple and the advantage of Poisson Regression is you if the model is completely explainable, you can explain the model why it is happening. Well, Decision Tree also you can explain unfortunately this plot is not very coming up very nicely.

(Refer Slide Time: 36:15)



The screenshot shows an RStudio environment with a script editor on the left and a plot window on the right. The script editor contains the following code:

```
11 mod = as.formula(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A
12               +I(HST^2)+I(AST^2)+I(HC^2)+I(AC^2)+
13               I(HR^2)+I(AR^2)+I(B365H^2)+I(B365D^2)+I(B365A^2))
14
15 fit1 = step(glm(mod,data=df_train
16             ,family = poisson(link="log")))
17
18 summary(fit1)
19
20 pred1 = predict(fit1,newdata = df_test,type = "response")
21
```

The console output shows the following results:

```
> y_hat = predict(fit3,newdata = df_train)
> RMSE3_in=sqrt(mean((y_hat-df_train$FTHG)^2))
> RMSE3_out=sqrt(mean((pred3-df_test$FTHG)^2))
> RMSE3[,]=c(RMSE3_in=RMSE3_in,RMSE3_out=RMSE3_out)
> RMSE
```

	In-Sample	Out-Sample
Poisson Regression	1.6005194	1.029906
Decision Tree	0.9664387	1.128384
Random Forest	0.4914496	1.067290

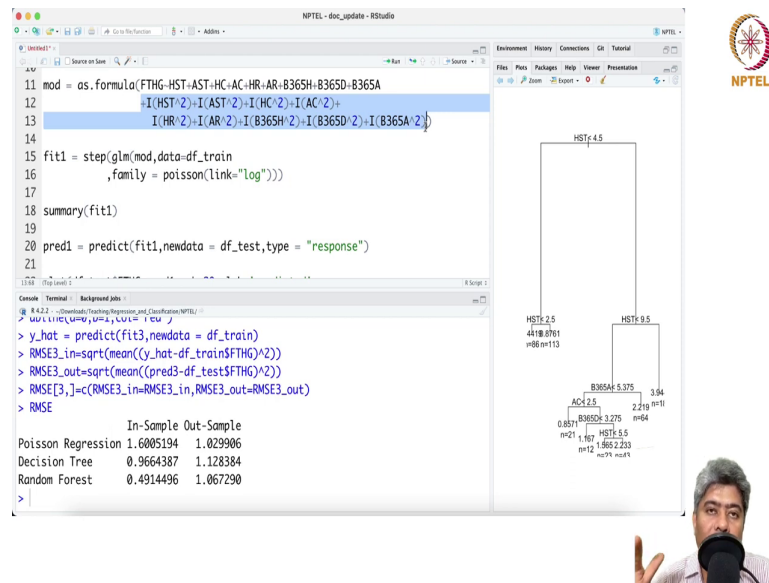
The plot window displays a decision tree structure. The root node is labeled "HST<4.5". The left branch leads to a node labeled "HST<2.5" with a predicted value of 1.4419 and n=86. The right branch leads to a node labeled "HST<9.5" with a predicted value of 0.8761 and n=113. The "HST<9.5" node further splits into two branches: "B365A<5.375" with a predicted value of 2.219 and n=64, and "B365A>5.375" with a predicted value of 3.944 and n=18. The "B365A<5.375" node splits into two branches: "AC<2.5" with a predicted value of 0.8571 and n=21, and "AC>2.5" with a predicted value of 3.275 and n=12. The "AC>2.5" node splits into two branches: "HST<5.5" with a predicted value of 1.666 and n=71, and "HST>5.5" with a predicted value of 2.231 and n=47.

The NPTEL logo is visible in the top right corner of the RStudio window.

But overall, you can nicely explain what is happening here and, yeah. So, I think if you just let me zoom that, ok. So, what it is doing, it is basically saying that if HST is less than 4.5 and then you come by this route if HST is less than 2.5 then you will score a goal with rate 0.4419.

If it is greater than 2.5 then it you will score a goal with 0.8761, if it is greater than 4.5 then you come this side, if HST is less than 9.5 then you come this way, if the bit 365 is less than greater than 5365 5.37 then you will score a goal with rate 2.219, if it is greater than that then you will score a goal with rate 3.944. So, that is what it is saying effectively whereas. So, Decision Tree is also very good in terms of modelling, but you can see at the end.

(Refer Slide Time: 37:35)



The screenshot shows an RStudio window with the following R code in the script editor:

```
11 mod = as.formula(FTHG~HST+AST+HC+AC+HR+AR+B365H+B365D+B365A
12 +I(HST^2)+I(AST^2)+I(HC^2)+I(AC^2)+
13 I(HR^2)+I(AR^2)+I(B365H^2)+I(B365D^2)+I(B365A^2))
14
15 fit1 = step(glm(mod,data=df_train
16 ,family = poisson(link="log")))
17
18 summary(fit1)
19
20 pred1 = predict(fit1,newdata = df_test,type = "response")
21
```

The console output shows the results of the model comparison:

```
> y_hat = predict(fit3,newdata = df_train)
> RMSE3_in=sqrt(mean((y_hat-df_train$FTHG)^2))
> RMSE3_out=sqrt(mean((pred3-df_test$FTHG)^2))
> RMSE3[,]=c(RMSE3_in,RMSE3_out=RMSE3_out)
> RMSE
```

	In-Sample	Out-Sample
Poisson Regression	1.6005194	1.029906
Decision Tree	0.9664387	1.128384
Random Forest	0.4914496	1.067290

On the right side of the RStudio window, a decision tree plot is visible. The root node is labeled 'HST < 4.5'. The left branch leads to a leaf node 'HST < 2.5' with a predicted value of 4418.0761 and n=86. The right branch leads to a node 'HST < 8.5' with a predicted value of 3.94 and n=11. This node further splits into 'AC < 2.5' (n=21, predicted 0.85) and 'AC > 2.5' (n=4, predicted 2.219). The 'AC > 2.5' node splits into 'B365A < 5.375' (n=2, predicted 0.85) and 'B365A > 5.375' (n=2, predicted 3.275). The 'B365A > 5.375' node splits into 'HST < 5.5' (n=1, predicted 1.167) and 'HST > 5.5' (n=1, predicted 1.562).

When we add these engineered feature, when we add this engineered feature, ok. We found that Poisson Regression is doing even better than Decision Tree and random forest. But if you do not add this engineered feature then; obviously, it is just fitting a simple hyper linear line I mean linear hyper plane, basically straight line in two dimension and obviously, it is not doing very poorly compared to decision tree and random forest.

So, in my opinion you have to try all kinds of model with feature engineering and then you have to apply stepwise selection and dimension model dimension reduction technique and come up with a more parsimonious model and eventually you will see.

And effectively it is very difficult to say in my experience which model will be the best, there is no uniformly best model, in my experience you should try to fit all the models and see



which model gives you the best fit. So, with this I will stop here and see you in the next week, next lecture.

Thank you very much see you.