0:00
[Music]
0:13
hi all uh in today's session we are
0:15
going to learn about scripting pdfs from
0:18
a given url
0:19
so most of the time of being a data
0:21
scientist is spent on sourcing the
0:24
required data to solve a problem
0:26
followed by
0:27
preparing the data which in a way is
0:29
kind of cleaning the data and
0:31
transforming the data to be consumed by
0:33
the model which is being built
0:35
so one such data sourcing which we are
0:38
going to learn today is scraping the
0:39
pdfs
0:41
so the objective of this particular
0:43
tutorial is uh
0:45
take a particular url which is which
0:47
kind of acts as a host to multiple pdf
0:50
files
0:51
and using python libraries
0:53
download
0:54
those particular pdfs and on top of that
0:57
uh from the downloaded pdfs we are going
0:59
to see how we can extract any table

1:02
information which is present in the pdfs
1:04
which could be used for further analysis
1:07
so
1:08
with this brief intro let's quickly jump
1:10
into the code so firstly i'm importing
1:13
the required libraries
1:16
so the two important libraries to make a
1:18
note of in this particular session are
1:20
the beautiful soup library which kind of
1:23
helps us to parse a given url and
1:25
download the required pdf content
1:28
and the other library of importance is
1:30
tabula so as the name suggests
1:33
tabla it kind of helps us to use i mean
1:36
it can be used to
1:38
read the tabular content which is
1:40
present in a pdf and it can be stored
1:42
into a structural format we will see
1:44
towards the later half of the uh
1:46
tutorial on how tabla is used
1:48
so once the uh
1:50
required libraries are imported
1:53
so next what i'm trying to specify is
1:56
the
1:57
location

1:58
where the script pdfs are to be saved
2:01
so for this purpose i am mounting my
2:03
google drive so essentially what i'm
2:05
trying to do here is
2:07
first thing is once we pass a url and
2:09
identify multiple pdfs there i am
2:12
specifying a location in my google drive
2:14
where the script pdfs are to be saved
2:17
so once my google drive is mounted
2:20
uh next is providing the input
2:23
essentially the url from which the uh
2:26
the pdfs are to be scraped and also the
2:29
location in my google drive where they
2:32
are to be saved after scraping so in
2:34
case of this particular location in my
2:36
drive is not existing
2:38
uh i'm just creating one using the os
2:40
library in python
2:43
let me quickly open this particular url
2:47
so as seen here this is the url for the
2:49
football fans out there this is uh
2:51
premier league url
2:52
where it where it public where it has
2:55
publications tab
2:56
where literally we have close to 50 pdfs

2:59
which have been presented every year for
3:01
example for the season 2021 2021-22
3:04
starting from the premier league
3:06
handbook and the rules
3:08
or the schedules of this particular
3:10
season everything is available as a pdf
3:13
so in case if i have to do it manually
3:15
i'll just go to the particular file
3:17
which is of interest to me and then
3:19
press download pdf so since you know
3:21
what we are going to do is from that
3:23
particular url using the beautiful soup
3:25
library and we are going to run a loop
3:28
and download all the pdfs so as seen
3:30
here we are specifying the url from
3:33
which the pdfs have to be picked and
3:35
also we are specifying the location
3:36
where this pdf files have to be saved
3:39
so
3:40
this this particular block of code is
3:42
the actual code which is going to be
3:44
used to download the pdfs so the first
3:46
step i am creating a soup object with a
3:49
particular url which is of interest to
3:50
me

3:53
and once a super object is created
3:55
what i am doing is for every link
3:58
within that particular object i am
4:00
picking up the uh the header anchor tags
4:03
which are ending with dot pdfs
4:06
so what happens so all the links which
4:08
are available in this particular url
4:10
ending with the dot pdf get picked
4:12
and among them uh i just for the
4:15
explanations purpose i just picked one
4:17
example here so i've seen here this is
4:19
one particular url starting with https
4:23
and it ends with a dot pdf
4:26
so for each of this particular links
4:28
which are coming in
4:30
we need to prepare the file name to
4:32
which the content has to be stored so
4:34
what we are doing uh
4:36
a simple split function we are using
4:38
with a forward slash essentially what
4:39
happens is each of the particular
4:44
forward slashes are identified within
4:45
this particular link and i'm picking the
4:48
last one so minus 1 position refers to
4:50
the last position in this particular

4:52
link
4:53
and so this get picked up this gets
4:56
picked up as the name of the particular
4:58
file
4:59
so what we're doing we are storing that
5:01
particular file
5:03
name and for the particular file name we
5:06
are passing the link and copying the
5:08
content so as seen here it is a for loop
5:10
which is written and for each pdf link
5:12
which is available in this particular
5:14
url
5:16
we are copying the content with the
5:17
exact file name which is getting picked
5:19
from the link
5:21
so let me quickly show how these files
5:23
look like
5:24
so as seen here
5:26
i kind of created or saved the location
5:30
uh from where these files are to be
5:32
saved they are in my collab notebooks
5:34
premiere link folder in my google drive
5:36
so my collab notebooks the premier
5:37
league folder all the files are getting
5:40
stored the pdfs so which are close to

5:41
about 50 files
5:43
which are present
5:45
and once these
5:47
are done now coming to the next part of
5:49
our exercise
5:51
let's say if you're interested in kind
5:53
of
5:53
[Music]
5:54
pulling out a particular table within
5:57
one of the pdfs and we have to store
5:59
them into a structured format structured
6:01
format i mean
6:02
can be a csv file or an excel file so uh
6:07
so
6:07
to start this particular exercise i have
6:10
pulled one particular uh file which is
6:13
which goes by the name this is pl
6:15
interactivecombine.pdf
6:17
so
6:18
this exactly is this
6:21
file
6:25
let me quickly do go to
6:27
jump on page number one so as seen here
6:30
this is the particular file which i have
6:31
pulled in so it kind of gives us

6:36
the overview of the previous season
6:38
so uh as seen here uh let me quickly
6:41
jump to page number 18.
6:45
let's assume that in this particular
6:48
page this particular table which
6:50
kind of tells us the final standings of
6:53
each of the clubs in that season i want
6:55
to i want to kind of save it into a data
6:57
frame
6:59
so that's essentially what tabla helps
7:01
me to do that so as seen here
7:04
what i'm doing is for this particular
7:05
pdf file i'm specifying the page number
7:08
from which the table has to be picked
7:12
so and what happened
7:15
it kind of pulled in uh the table or
7:17
information which is present in the
7:18
table but as seen here uh we can see
7:21
that it does not look
7:24
or it's not exactly the table which is
7:26
of interest to me it has pulled in other
7:28
information as well like the amount the
7:30
premier league invest per season in the
7:32
development of community facilities etc
7:34
which is exactly

7:36
getting picked from
7:38
this particular
7:40
block of text which is written here so
7:43
the reason may be as this is a
7:46
landscape layout
7:48
each of these is kind tablets kind of
7:51
picking up as a table and just getting
7:53
pulled in
7:55
however if
7:56
one big advantage of this particular
7:59
table is it kind of gives more control
8:02
to the user as well to specify
8:05
which area in a particular page should
8:08
be considered for pulling the table
8:11
so as i had stated
8:13
earlier this particular table is of
8:15
interest to me
8:16
so what we can do is we can
8:19
specify the area
8:21
from which the table has to be picked in
8:23
so i'm importing one more
8:25
function called the convert into from
8:27
tabular so what essentially convert into
8:29
does is
8:30
it picks the particular

8:33
file
8:34
or the pdf file from which we are going
8:36
to pick a table
8:37
and we are going to store it to a csv
8:40
file save it to a csv file that's
8:41
essentially what converted to does
8:44
and we are specifying the output format
8:46
and within that particular page
8:48
we are specifying the area
8:50
from which the table has to be picked
8:52
so uh these four parameters
8:54
can be picked from uh any of the uh
8:57
image uh softwares like a photoshop
8:59
where we specify the pointer and we get
9:01
the position of each of the pixel
9:03
positions here
9:04
so we specify that
9:06
and once we specify it and uh we are
9:09
storing it to a csv file
9:11
and once i read the csv file this is the
9:13
final output i'm just i'm getting from
9:15
that particular table so this is exactly
9:17
what the table is of interest to me
9:20
just getting stored here
9:22
uh as seen here uh we can see in

9:25
positions 10 and 11 we do have some text
9:27
which is not exactly which is present in
9:30
the table so let's see what exactly it
9:32
is
9:33
so
9:34
what happened is uh for the club
9:37
newcastle united
9:39
that is exactly post the newcastle
9:41
united we are getting some some extra
9:43
rows the reason being let me zoom in
9:46
here
9:55
so essentially what's happening is uh
9:58
for the newcastle united club
10:00
below their uh club badge they have
10:02
something this is which says they are
10:04
125 years old so this particular text is
10:07
getting picked up
10:09
and that's exactly what is getting
10:11
converted to one of the rows which goes
10:13
to the next phase of data cleaning
10:15
exercise where we kind of remove this
10:18
so uh in this way using uh tabula
10:22
and beautiful soup we can scrape the
10:25
pdfs and also
10:28
pick particular tables from each of the

10:30
pdfs which are interest to the user
10:33
thank you