Dysgu Peirianyddol

Cyflwyniad i algorithmau dysgu peirianyddol yn R a Python

Alun Owen

B.Sc. Traethawd Blwyddyn 3

Yr Ysgol Mathemateg Caerdydd

Diolchadau

Cynnwys

Rhestr Ddarluniau

Pennod 1

Cyflwyniad

1.1 Beth yw Dysgu Peirianyddol

1.1.1 Dysgu dan Oruchwyliaeth

1.1.2 Dysgu heb Oruchwyliaeth

1.1.3 Darllen Pellach

1.2 Pam

1.2.1 Be sydd yna yn barod?

1.2.2 Pam Python ag R?

1.2.3 Pam Cymraeg?

1.3 Strwythyr

Pennod 2

Clystyru k-cymedr

2.1 Cefndir

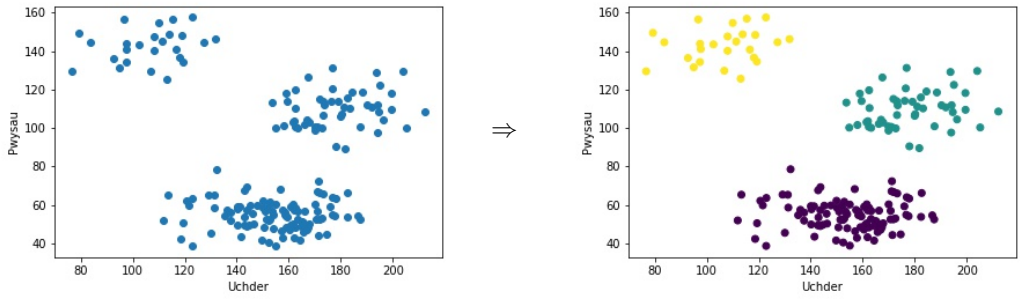
Mae clystyru k-cymedr yn ffordd o ddysgu heb oruchwyliaeth, mae'n cymryd data heb ei labelu ac yn eu sortio i mewn i k wahanol glwstwr yn yr obaith i ddarganfod rhyw strwythur doedden ddim yn gwybod yn gynharach.

I roi enghraifft gwelwch Ddarlun 2.1. Mae'r gwerthoedd ar echelin x yn cynrychioli uchder rhyw berson a'r llall yn cynrychioli pwysau'r person. Fel gwelwn yn y llun ar y chwith gallwn weld tri grŵp naturiol wedi'i ffurfio. Rydym nawr eisiau eu grwpio yn ffurf Fathemategol. Mae clystyru k-cymedr yn medru dosrannu'r tri grŵp fel gwelwn ar ochr dde'r darlun.

2.2 Sut mae Clystyru K-cymedr yn gweithio?

2.2.1 Y Dull

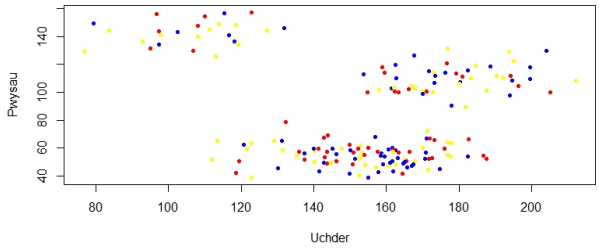
Mae clystyru k-cymedr yn syml, mae ond yn dilyn pedwar cam [1]. I wneud yn siŵr fod yn ei ffurf fwyaf cyntefig, fyddan yn defnyddio mesur pellter Ewclidaidd. Yn ogystal mae rhaid dewis k cyn cychwyn y



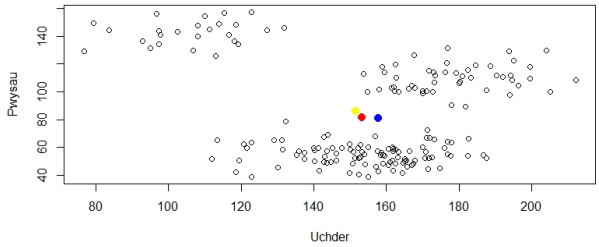
Darlun 2.1: Cyn ac ar ôl clystyru k-cymedr.

proses. Mae'n bosib optimeiddio'r dewis o k, a gwnawn drafod hyn hwyrach ymlaen. Dyma bedwar cam yr algorithm a sut maent yn edrych pan fyddwn ni'n defnyddio'r algorithm ar y data y gwelwn yn 2.1:

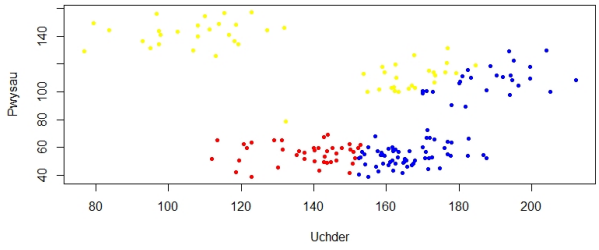
1. Aseinio pob elfen i un o'r k clystyrau ar hap.



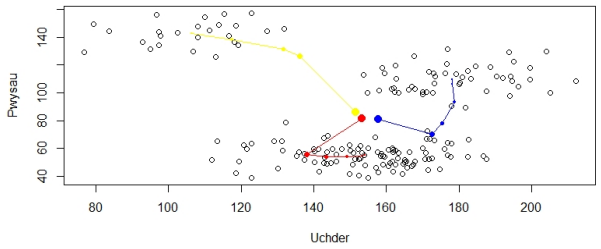
2. Cyfrifo canolbwynt (hynny yw cymedr) pob clwstwr.



3. Ail-aseinio pob elfen unwaith eto i'r clwstwr gyda chymedr agosaf.



4. Ailadrodd camau dau a tri tan fod y canolbwyntiau ddim yn symud rhagor.



2.2.2 Darn Mathemategol

Diffiniwn bob clwstwr rydym yn ceisio darganfod fel C_i lle bydd $i \in \{1, 2, \ldots, k\}$, mae gennym hefyd n pwyntiau data $x_1, x_2, \ldots, x_n$. Gadewch i c_i bod yn bwynt sy'n ganolbwynt y clwstwr C_i. Ar gyfer y cam cyntaf angen aseinio pob x_j i ryw glwstwr C_i ar hap. Yna gan ein bod yn datgelu ein bod yn delio gyda phlân Ewclidaidd, mi fyddem yn darganfod cymedr pob clwstwr gan y fformiwla ganlynol:

$$C_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j \tag{2.1}$$

lle diffiniwn S_i fel y set o bwyntiau data sydd wedi'i aseinio i glwstwr C_i.

Nawr mae gan bob clwstwr cymedr newydd, fedrwn aseinio pob pwynt data i'r canolbwynt agosaf. Caiff hyn ei gwneud gan fynd drwy bob pwynt data a chyfrifo'r pellter Ewclidaidd i bob canolbwynt. Yna fydd y pwynt priodol yn cael ei labelu gyda'r clwstwr sydd a'r pellter lleiaf o'i chanolbwynt i'r pwynt data. Hynny yw

$$\arg\min_{c_i} dist(c_i, x_j)^2 \tag{2.2}$$

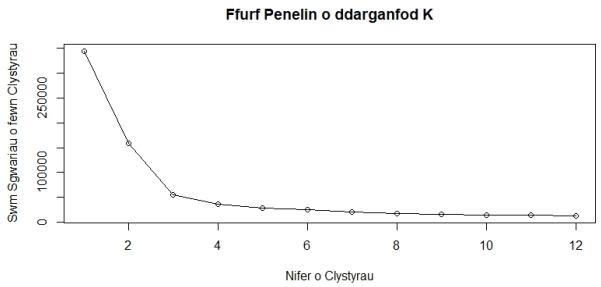
Unwaith mae'r proses wedi'i chychwyn, angen ailadrodd y darn o ddarganfod y creiddiau newydd ac yna ail labelu'r pwyntiau data.

2.2.3 Sut i ddarganfod y k orau?

Mae yna wahanol ffurf i ddarganfod k, edrychwn ar ddau wahanol ffordd o wneud hyn.

Dull Penelin

Mae'r dull penelin yn cymharu'r cyfanswm o swm sgwariau o fewn y clystyrau. Unwaith gennym y cyfanswm o swm sgwariau o fewn clystyrau i bob k rydym eisiau cymharu, fyddem yn creu plot o bob k yn erbyn y cyfanswm o swm sgwariau o fewn y clystyrau ar gyfer y k hynny. Unwaith mae gennym y graff, allwn ei ddadansoddi.

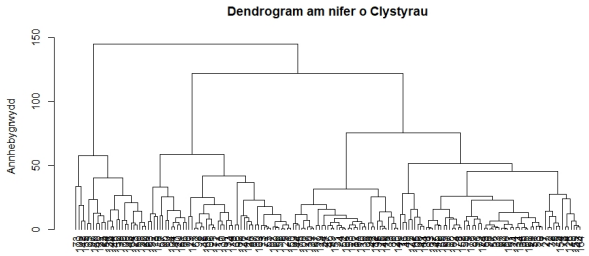


Darlun 2.2: Enghraifft o blot o k yn erbyn y cyfanswm swm o sgwariau

Yn y graff yn Narlun 2.2, gwelwn fod swm sgwariau yn fawr yn cychwyn gyda k=1 sydd yn gwneud synnwyr. O'r pwynt yma wedyn fydd yna newid mawr yn y swm sgwariau. Unwaith mae'r newid mawr hwn yn dod i ben fydd gennym ongl yn cael ei greu lle bydd newid k dim ond yn creu newid bach. Y pwynt yma fydd yr optimwm ar gyfer nifer k o glystyrau. Fel gwelwn yn glir yn ein henghraifft ni, mae'n glir fod K=3 yw dewis orau ar K.

Dendrogram

Mae dendrogram yn ffordd wahanol iawn i canfod y nifer orau k o glystyrau. Mae'n defnyddio darn o glystyru hierarchaidd i greu diagram canghennog. Mae'r echelin llorweddol yn dangos pob gwrthrych yn ein set o ddata. Mae'r echelin fertigol yn dangos mesur o annhebygrwydd. Mae Darlun 2.3 yn dangos dendogram ar gyfer yr un data.



Darlun 2.3: Enghraifft o dendogram

I ddadansoddi'r dendrogram mi fyddwn edrych yn bennaf ar yr echelin fertigol. Edrychwn allan am yr annhebygrwydd fwyaf rhwng cyflwyniad o gangen arall yn y goeden. Welwn ni hyn yn ein henghraifft ni ar ôl i'r drydydd clwstwr cael ei gyflwyno yn dendrogram. Mae hyn yn datganu'r un peth a'r dull penelin.

2.3 Tiwtorial yn R

Mi fyddwn yn edrych ar ddata o uchder a phwysau 175 wahanol berson. Mi allwch chi lawrlwytho y data yma o fan hyn.

Yno fydd angen lawrlwytho a gosod y pecynnau `graphics`, `stats` ag `datasets` ar eich fersiwn chi o RStudio. Ffordd hawdd i wirio hyn fydd i ddefnyddio'r côd canlynol:

```
install.packages("graphics")
install.packages("stats")
install.packages("datasets")
library(graphics)
library(stats)
library(datasets)
```

Mae'r darn gyntaf o'r côd uchod yn gosod/diweddaru'r pecynnau angenrheidiol. Mae'r ail ddarn yn llwytho'r pecynnau i ein fersiwn ni o RStudio.

Nawr mi wnawn lwytho'r data.

```
heightvsweight <- read.csv("C:/Users/User/Desktop/Dysgu_Peirianyddol/heightvsweight.csv")
View(heightvsweight)
```

Mae'r string sydd mewnbwn y ffwythiant `read.csv` yn cyfeirio at y lleoliad ar ein cyfrifiadur lle gallwn ganfod y ffeil csv priodol. Rhaid gwneud yn siŵr eich bod yn defnyddio'r lleoliad cywir i'r lleoliad o'ch ffeil chi. Ar ôl rhedeg y côd ddylai eich data edrych yn debyg i'r canlynol:

	Uchder	Pwysau
1	163.22687	100.09760
2	183.18087	110.18107
3	172.69407	99.79701
4	165.07549	51.66760
5	147.74605	59.79469
6	161.45039	103.04177
7	162.41267	58.50832
8	146.28025	50.36660
9	154.03614	47.93155
10	152.20904	59.79795

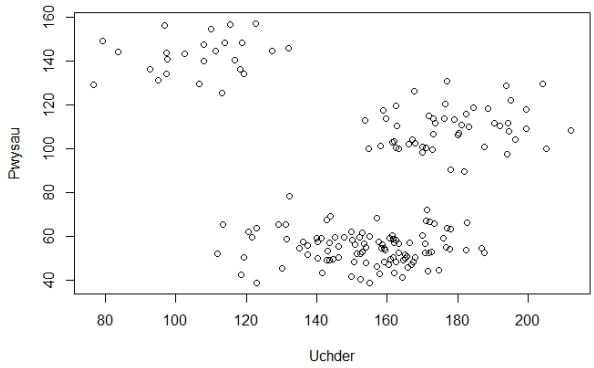
Gan fod y data hefo enwau ar gyfer y colofnau, gallwn atodi'r data i lwybr chwilio R. Bydd hyn yn gadael i ni gyfeirio at enwau colofnau'r data yn ein côd fydd yn gwneud yn lawer mwy symlach i ddeall.

```
attach(heightvsweight)
```

I wneud fwy o synnwyr o'r data, mi wnawn blotio'r data.

```
plot(Uchder, Pwysau, pch = 21)
```

Sy'n rhoi:



Gwelwn fod yna 3 clwstwr clir.

Rŵan rydym yn gallu tybio fod y data yn gallu cael i rannu i dri chlwstwr gwahanol, mi wnawn ddefnyddio'r algorithm dysgu peirianyddol i'w ddehongli. Rhedwn y canlynol i redeg clystyru k-cymedr yn R. Rydym yn defnyddio'r ymresymiad `nstart` i ddewis faint o setiau ar hap o greiddiau wnawn gymered.

```
kcymedr <- kmeans(heightvsweight,3, nstart = 50)
```

Allwn nawr adio colofn newydd i'r data sef y clystyrau newydd mae'r algorithm wedi'i darganfod.

```
heightvsweight$Clwstwr3 <- kcymedr$cluster
```

Gallwn weld y newid hwn gan ddefnyddio'r un côd a ddefnyddion yn gynharach.

```
View(heightvsweight)
```

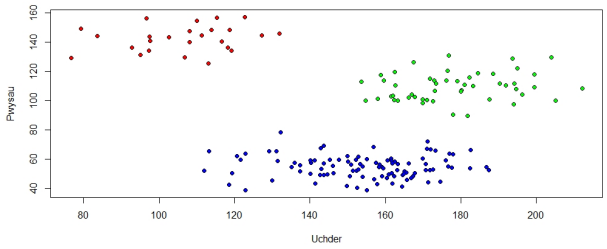
	Uchder	Pwysau	Clwstwr3
1	163.22687	100.09760	1
2	183.18087	110.18107	1
3	172.69407	99.79701	1
4	165.07549	51.66760	2
5	147.74605	59.79469	2
6	161.45039	103.04177	1
7	162.41267	58.50832	2
8	146.28025	50.36660	2
9	154.03614	47.93155	2
10	152.20904	59.79795	2

Mae'n bosib fydd yr algorithm wedi labeli'r clystyrau gwahanol gyda rhifau gwahanol i'r hyn a welwch fan hyn, ddylai'r clystyrau ei hun fod yn hafal. Mae hyn oherwydd y setiau ar hap cychwynnol mae'r algorithm yn ei gymered i gychwyn.

Rhedwn y côd canlynol liwio'r clystyrau newydd ar graff.

```
plot(Uchder, Pwysau, pch = 21, bg=c("red","green","blue")[unclass(kcymedr$cluster)])
```

Sy'n rhoi:



I gymharu, nawr mi nawn rhedeg yr algorithm ar gyfer 6 clwstwr i weld y clystyrau pan fydd $k = 6$.

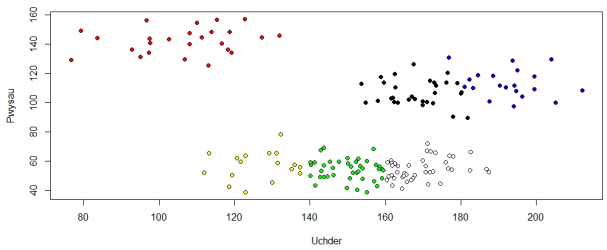
```
kcymedr <- kmeans(heightvsweight,6, nstart = 50)
heightvsweight$Clwstwr6 <- kcymedr$cluster
View(heightvsweight)
```

	Uchder	Pwysau	Clwstwr3	Clwstwr6
1	163.22687	100.09760	1	1
2	183.18087	110.18107	1	6
3	172.69407	99.79701	1	1
4	165.07549	51.66760	2	4
5	147.74605	59.79469	2	2
6	161.45039	103.04177	1	1
7	162.41267	58.50832	2	4
8	146.28025	50.36660	2	2
9	154.03614	47.93155	2	2
10	152.20904	59.79795	2	2

Gwelwn fod y labeli newydd wedi cael ei ychwanegu i'n tabl. Yna gan blotio graff arall, fedrem weld y 6 clwstwr yn gliriach.

```
lliwiau <- c("red","green","blue", "yellow", "black", "white")
plot(Uchder, Pwysau, pch = 21, bg=lliwiau[unclass(kcymedr$cluster)])
```

Sy'n rhoi:



2.4 Tiwtorial yn python

Yn y tiwtorial hwn mi wnawn edrych ar yr un data a welom yn y tiwtorial diwethaf. I gychwyn bydd rhaid llwytho'r pecynnau `pandas`, `matplotlib.pyplot` ag `sklearn.cluster` drwy redeg y côd canlynol:

```
import pandas as pd
import matplotlib.pyplot as plt
import sklearn.cluster
```

Y rŵan mi wnawn lwytho'r data i mewn i'n gwaith gan redeg y côd:

```
data = pd.read_csv('heightvsweight.csv')
```

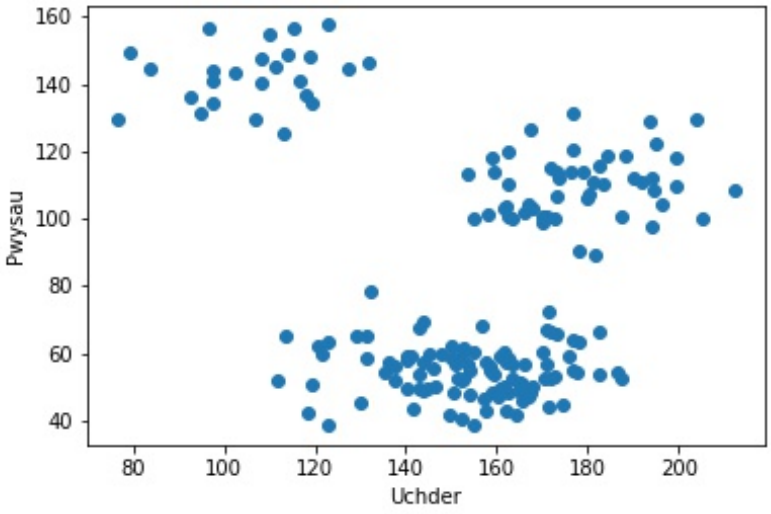
Mae'r string sydd mewnbwn y ffwythiant `pd.read_csv` yn cyfeirio at y lleoliad ar ein cyfrifiadur lle gallwn ganfod y ffeil csv priodol. Rhaid gwneud yn siŵr eich bod yn defnyddio'r lleoliad cywir i'r lleoliad o'ch ffeil chi. Unwaith fydd wedi cael ei llwytho, allwn ni gweld yn fras y data gennym ni.

```
data.head()
```

	Uchder	Pwysau
0	163.226866	100.097603
1	183.180871	110.181072
2	172.694074	99.797013
3	165.075492	51.667604
4	147.746048	59.794691

I weld y data mewn ffordd fwy gweledol, wnawn blotio graff gwasgariad o'r data

```
plt.scatter(data['Uchder'], data['Pwysau']);
plt.xlabel('Uchder')
plt.ylabel('Pwysau')
plt.show()
```



Darlun 2.4: Enghraifft o ddata da i cael ei clystyru.

Fel gwelwn, mae'r data yn edrych fel ei fod mewn tri chlwstwr. Felly wnawn ddefnyddio'r ffurf algorithm dysgu peirianyddol i'w labelu.

```
kmeans = sklearn.cluster.KMeans(n_clusters=3).fit(data)
data['Cluster (k=3)'] = kmeans.predict(data)
```

Gallwn weld y newid hwn gan ddefnyddio'r un côd a ddefnyddion yn gynharach.

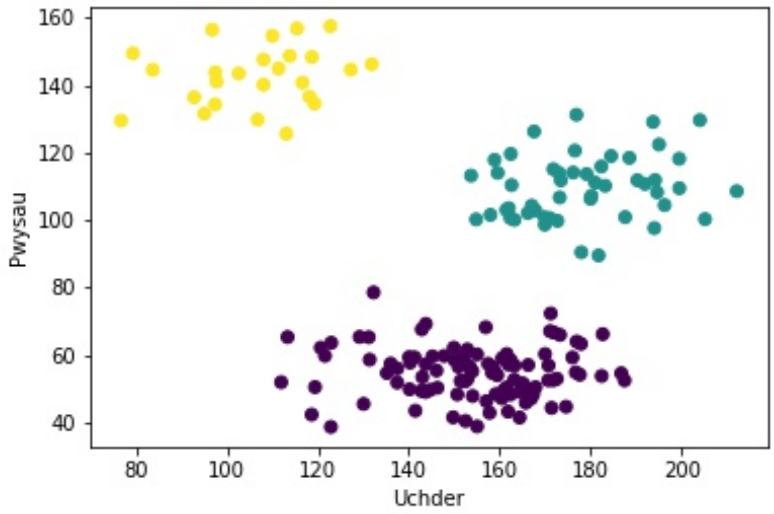
```
data.head()
```

	Uchder	Pwysau	Cluster (k=3)
0	163.226866	100.097603	1
1	183.180871	110.181072	1
2	172.694074	99.797013	1
3	165.075492	51.667604	0
4	147.746048	59.794691	0

Fel y gwelwyd, mae'r data wedi'i rhoi i mewn i dri chlwstwr ac wedi'i labelu gyda rhif y clwstwr. Gan fod pob pwynt yn y data nawr gyda label, allwn ni creu'r plot eto ond gyda bob clwstwr yn lliw gwahanol.

```
plt.scatter(data['Uchder'], data['Pwysau'], c=data['Cluster (k=3)']);
plt.xlabel('Uchder')
plt.ylabel('Pwysau')
plt.show()
```

Sy'n rhoi:



Darlun 2.5: Sut ddylsa eich graff edrych gyda 3 clystwr.

Fel y gwelwn, gweithiodd yr algorithm yn wych. Wnawn nawr trio clystyru k-cymedr gyda k yn hafal i 6.

```
kmeans = sklearn.cluster.KMeans(n_clusters=6).fit(data)
data['Cluster (k=6)'] = kmeans.predict(data)
```

Sy'n rhoi:

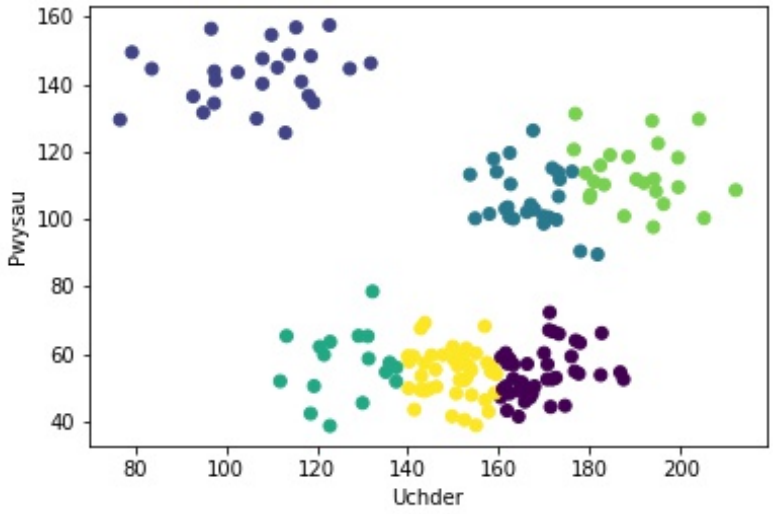
```
data.head()
```

	Uchder	Pwysau	Cluster (k=3)	Cluster (k=6)
0	163.226866	100.097603	1	4
1	183.180871	110.181072	1	2
2	172.694074	99.797013	1	4
3	165.075492	51.667604	0	0
4	147.746048	59.794691	0	3

Gallwn hefyd gweld canlyniad rhoi'r data i mewn i 6 clwstwr gwahanol:

```
plt.scatter(data['Uchder'], data['Pwysau'], c=data['Cluster (k=6)']);
plt.xlabel('Uchder')
plt.ylabel('Pwysau')
plt.show()
```

Sy'n rhoi:



Darlun 2.6: Sut ddylsa eich graff edrych gyda 6 clystwr.

Dyma sut dylaf eich data edrych fel ar ôl a phrosesu drwy glystyru 6-cymedr.

Pennod 3

Termau

Llyfryddiaeth

[1] David M. J. Tax; Ferdinand van der Heijden; Robert Duin; Dick de Ridder. Classification, parameter estimation and state estimation: An engineering approach using matlab. 2012.