Dysgu Peirianyddol

Cyflwyniad i algorithmau dysgu peirianyddol yn R a Python

Alun Owen

B.Sc. Traethawd Blwyddyn 3

Yr Ysgol Mathemateg Caerdydd

Diolchadau

Hoffwn ddiolch i Dr Geraint Palmer am yr holl gefnogaeth wrth oruchwylio'r prosiect ac yn ogystal fel tiwtor personol yn ystod y flwyddyn ddiwethaf.

Cynnwys

Rhestr Ddarluniau

Rhestr Dablau

Pennod 1

Cyflwyniad

Mi fyddwn yn ysgrifennu fy mhrosiect trydydd flwyddyn am algorithmau dysgu peirianyddol ag sut i'w defnyddio yn R ac Python. Fydd y prosiect yn cael ei ysgrifennu trwy cyfrwn y Gymraeg. Mi fyddwn yn creu traethawd a gwefan i gyfathrebu gwybodaeth am yr algorithmau. Dwi am ysgrifennu am glystyru k-cymedr, atchweliad logistaidd ag dosbarthiad naïf Bayes.

Bydd y tiwtorialau yn cael ei ddangos drwy'r ieithoedd rhaglennu Python ag R, gan taw nhw yw'r ddwy iaith fwyaf poblogaidd ar gyfer dysgu peirianyddol a gwyddor data yn ôl arolwg Kaggle nol yn 2018 [1]. Mae hefyd ganddynt trwyddedau am ddim, sy'n bwysig ar gyfer hygrychedd. Rheswm arall i ddysgu defnyddio'r ieithoedd rhaglennu yma yw bod nhw'n god agored sy'n wahanol i SQL er enghraifft. Gan ei fod nhw'n god agored poblogaidd, mae'n meddwl fod nhw'n cael ei diweddaru yn aml. Cafodd R ei greu gan ystadegwyr a gan fy mod yn creu tiwtorialau am ddysgu peirianyddol o'r ochr ystadegaeth, mae'n gymhelliad i ddefnyddio R. Ar gyfer dysgu Python, nol yn 2019 roedd Python yr iaith rhaglennu ail fwyaf i gael "pull requests" ar Github [10], a Github oedd y gwesteiwr fwyaf o god ffynhonnell yn y byd. Mae Python ag R yn cael ei ystyried i fod yn hawdd dysgu a deall yn gymhariaeth gydag ieithoedd rhaglennu eraill. Gwelwn yn y tiwtorialau i ddod nad yw gweithredu'r algorithmau yn yr ieithoedd hyn yn gymhleth.

1.1 Beth yw Dysgu Peirianyddol

Yn syml, dysgu peirianyddol yw algorithmau i optimeiddio rhyw feini prawf gan ddefnyddio data. Mae rhain yn cynnwys model wedi'i ddiffinio o rhai paramedrau mesuradwy, a'r darn dysgu fydd i optimeiddio gyda pharch tuag at y paramedrau hyn. Gall y model fod yn un disgrifiadol o'r data, neu un sy'n rhagfynegi rhyw agwedd o'r data. Mae dysgu peirianyddol yn defnyddio ystadegaeth i adeiladu'r modelau mathemategol, ac mae cyfrifiadureg yn edrych mwy i mewn i effeithiolrwydd y proses.[23]

Nid yw dysgu peirianyddol yn faes newydd, mae wedi bod o gwmpas ers y 50au pan wnaeth Arthur Samuel o IBM creu rhaglen ar y cyfrifiadur i chwarae'r gem draffts. Yr amser hwn cafodd y term ei bathu. Yna drwy ddatblygiadau technolegol diweddar, mae posibilrwyddau ddysgu peirianyddol bron yn ddiddiwedd. Ers cael ei sefydlu yn y 50au, cymerodd tan 1997 i ddatblygu rhaglen a all guro'r chwaraewr gwyddbwyll

orau yn y byd. Nid yn unig yw dysgu peirianyddol yn cael ei ddefnyddio i greu rhaglenni gemau, mae nawr yn cael ei ddefnyddio ym mhob math o raglenni megis llawer o apiau ar eich ffôn fel "Google Maps", "Uber" i "Netflix" [12].

Mae sylfaeni ddysgu peirianyddol yn cael eu defnyddio yn prosesu iaith naturiol a dadansoddi sentiment[7]. Mae'r proseses yma yn cael ei ddefnyddio i wneud ffwythiannau megis adnabod lleferydd, creu tecst i leferydd a chyfieithiad peirianyddol. Yn ogystal mae dysgu peirianyddol yn cael ei ddefnyddio i brosesu lluniau, mae'r defnydd yma'n cael ei weld yn aml gyda systemau adnabod wynebau.

Mae dysgu peirianyddol yn cael ei ddefnyddio yn aml i ddadansoddi data pryd bynnag gennym ddata mawr. Mae ein llywodraeth yn defnyddio dysgu peirianyddol yn aml, mae adran actiwari y llywodraeth yn defnyddio dysgu peirianyddol i ddatblygu mewnwelediadau i'w problemau. Hyd yn hyn maen nhw wedi defnyddio dysgu peirianyddol i rannu cynlluniau pensiwn i grwpiau gyda phriodweddau tebyg ac wedi rhagfynegi cyflog graddedigion yn y dyfodol. [16]

Mae deallusrwydd artiffisial wedi tyfu yn esbonyddol yn ddiweddar gyda dysgu peirianyddol. Mae deallusrwydd artiffisial yn y newyddion drwy'r adeg oherwydd datblygiadau parhaus. Yn ddiweddar rydym wedi gweld moduron heb yrrwr, diagnosau meddygol, a 'chat bots'.

Categoreiddiwn algorithmau dysgu peirianyddol i ddwy brif fath - dysgu o dan orchwyliaeth, a dysgu heb orchwyliaeth. Maent yn wahanol yn eu pwrpas a'u dulliau.

1.1.1 Dysgu dan Oruchwyliaeth

Diffiniwn α i fod y set o bob label, gall y labeli fod yn arwahanol neu yn ddi-dor, yna diffiniwn β fod y fector o ddimensiwn $D \in \mathbb{Z}_+$. Gadewch i'n data fod nifer o barau o bwyntiau $(\mathbf{x}_i, y_i)$. Yn y fan hyn mae $y_i \in \alpha$. Fydd $\mathbf{x}_i \in \beta$ yn fector gyda gwerthoedd ar gyfer priodoleddau gwahanol. Y nod ar gyfer dysgu dan oruchwyliaeth yw dysgu'r mapiad o $\mathbf{x}$ i y.[27]

Gwnawn hyn trwy hollti'r data i mewn i set hyfforddi a set profi, lle mae'n bwysig i dybio pob pwynt wedi'i samplu'n annibynnol a'i dosbarthu o ddosraniad dros $\alpha \times \beta$. Defnyddiwn y set hyfforddi i rhedeg yr algorithm dysgu peiranyddol a chanfod y mapiad. Defnyddiwn y set profi er mwyn gwirio'r mapiad hyn. Fel arfer defnyddiwn tua 70% o'r data fel y set hyfforddi, a'r 30% gweddill fel y set profi, er taw tra-baramedr (hyper-parameter) yr hon ac felly gellid ei ddewis.

Mae'r ffurfiau wahanol o ddysgu dan oruchwyliaeth yn cael ei rhannu i ddau faes yn ôl sut fath o ddata sydd gennym. Os yw'r label yn data arwahanol, mae gennym ddosbarthiad (classification). Os yw'r label yn data di-dor, mae gennym atchweliad (regression). Mae yna lwyth o wahanol fathau o ddosbarthiadau ag atchweliadau; dyma ambell o enghreifftiau ohonynt:

Dosbarthiad:

- Coed penderfyniadau
- Dosbarthiad naïf Bayes
- K cymydog agosaf

Atchweliad:

- Atchweliad logistaidd
- Atchweliad llinol
- Atchweliad Poisson

1.1.2 Dysgu heb Oruchwyliaeth

Diffiniwn γ i fod yn ddosraniad. Gadewch i'n data fod $X = (x_1, \ldots, x_n)$ sy'n dynodi n pwyntiau lle mae $x_i \in \gamma$ ag $i \in \{1, \ldots, n\}$. Tybiwn fod yr enghreifftiau x_i wedi'i samplu'n annibynnol a'i dosbarthu o ddosraniad unfath ar γ. Y nod o ddysgu heb oruchwyliaeth yw amcangyfrif dwysedd o'r dosraniad sy'n debygol o fod wedi creu X.[27]

Mae'r fatha boblogaidd o ddysgu heb oruchwyliaeth yn cymryd ffurf wannach o'r syniad hyn. Dyma ambell i enghraifft ar ffurfiau gwahanol o ddysgu heb oruchwyliaeth:

- Clystyru
- Lleihad dimensiwn
- Model Markov cudd

1.1.3 Dysgu atgyfnerthol

Mae dysgu atgyfnerthol yn dysgu drwy wrando ar adborth o'r system ei hyn. Fydd y peiriant yn dysgu i wneud dulliau sy'n arwain tuag at adborth da ac yna yn osgoi unrhyw dullai caiff adborth gwael. Mae hyn yn wahanol i ddysgu dan oruchwyliaeth gan fod does ddim gymaint o ddibyniaeth ar y ddata hyfforddi. Mae'n wahanol i ddysgu heb oruchwyliaeth gan ein bod angen dewis pryd i dderbyn adborth o'r system [7].

Enghraifft o ddysgu atgyfnerthol yw prosesau penderfynu Markov, mae'n cael ei ddiffinio fel proses stocastig dan reolaeth. Mae prosesau penderfynu Markov yn cael ei ddiffinio gan y plyg (S, A, T, p, r). Yn y plyg mae'r newidynnau yn cael ei ddiffinio fel: S yw'r gofod sy'n cynrychioli esblygiad y prosesau, A yw'r set o bob gweithred sy'n bosib, T yw'r set o amseroedd rhwng dewisiadau, p sy'n dynodi'r ffwythiant tebygolrwydd ar gyfer y trosglwyddo cyflwr ac r sy'n dynodi'r ffwythiant wobrwyo i'r trosglwyddo cyflwr. Mae prosesau penderfynu Markov yn efelychu system ac yna yn dewis gweithrediadau ar hap ac yn penderfynu ansawdd y dewis drwy'r ffwythiant wobrwyo. Felly fydd y system yn optimeiddio i ddewis yr opsiynau fydd yn allbwn y wobr orau ac felly yn dewis yr opsiwn o'r ansawdd orau.[28]

1.2 Adnoddau Cyfrwng Cymraeg ar gyfer Dysgu Peirianyddol

Dyma'r adnoddau Cymraeg sydd ar gael yn barod yn y faysydd Deallusrwydd Artiffisial, Cyfrifiadureg ag Mathemateg:

1.2.1 Gwerthuso adnoddau sy'n bodoli

Mae Tabl 1.2.1 wedi'i rhannu i adnoddau codio Cymraeg ag i adnoddau mathemategol Cymraeg. Fel gwelwn yn ebrwydd, mae'r nifer o adnoddau codio yn llawer fwy nag rheina o adnoddau mathemategol. Mae'r adnodd [11] yn cychwyn da i unrhyw un sydd gyda diddordeb o gychwyn codio yn Python, mae'n defnydd gwych ar gyfer myfyrwyr mathemateg israddedig gan ei fod yn benodol i fathemateg. Yn ogystal i'r cwrs yma mae gennym hefyd bach o gyflwyniad i Python ar y wefan Sgiliau Ymchwil Ailhynhyrchiadwy [17], er bod hyn yn canolbwyntio mwy ar sgiliau fwy eang megis rheolaeth fersiwn a datblygu meddalwedd ymchwil.

Adnodd	***Awdur***	***Fformat***	***Cynulleidfa Darged***	***Linc***
Cwrs 'Cyfrifiadureg ar gyfer Mathemateg' yn Python	Dr Vince Knight a Dr Geraint Palmer	Tiwtorial ar wefan	Dechreuwyr mewn codio Python	[11]
Cyrsiau allgyrsiol yn Scratch, HTML, CSS ag Python	Code Club	Tiwtorialau ar y wê	Plant rhwng 9 ac 13	[4]
Adnoddau ar lawer o testynau wahanol yn cyfrifiadureg	Technocamps	Mewn ffurdd pdf	Plant yn ysgol/coleg	[6]
Cwrs mewn sgiliau ymchwil cyfrifiadurol	Dr Geraint Palmer	Tiwtorialau ar y wê	Ymchwilwyr	[17]
Gwybodath ar gydrannau pwysicaf technolegau iaith	Uned technoleg iaith Prifysgol Bangor a Cymen Cyf.	Llawlyfr ar y wê	Myfyrwyr, datblygwyr neu academyddion heb gefndir yn y maes	[7]
ap Botio i hybu rhaglennu i plant	Tinopolis	ap ar cynnyrch apple	Plant rhwng 7 ac 11	[9]
Enghreifftiau a thiwtorialau o cynnyrch technolegau iaith	Uned technoleg iaith Prifysgol Bangor a Cymen Cyf.	Ystorfa ar github	Myfyrwyr, datblygwyr neu academyddion heb gefndir yn y maes	[22]
Tiwtorialau Python	Dr Geraint Palmer a Stephanie Jones	Fideos ar Youtube	Dechreuwyr yn rhaglennu gyda Python	[21]
Tiwtorialau SQL	pl/sql tiwtorial	Wefan gyda tiwtorialau	Dechreuwyr yn SQL	[20]
Nodiadau agored am algebra llinol	Dr Alun Morris	Nodiadau (pdf)	Myfyrwyr prifysgol	[26]
Amrhyw o adnoddau Mathemategol yn dilyn cwricwla CBAC	Dr Gareth Evans	Wefan gyda linciau i wahanol fathau o adnoddau	Myfyrwyr rhwng 11 a 18	[5]

Taflen 1.1: Tabl o adnoddau sydd allan yn barod.

Gwelwn yn ogystal fod yna adnoddau ar gyfer rhaglennu yn Python yn Gymraeg ar gael ar YouTube. Ar sianel Geraint Palmer [21] mae yna gasgliad o diwtorialau Python. Mae'r tiwtorialau yn cychwyn gyda'r sylfaen o raglennu yn Python ag yn gorffen gyda mynd dros yr algorithm genetig, sy'n mynd law yn llaw gyda fy mhrosiect. Mae gan y wefan codeclub amrywiaeth eang o adnoddau ar gyfer codio [4]. Mae yna gyrsiau ar HTML, CSS, Python a Scratch. Yn ogystal mae yna brosiectau pellach sy'n gweithredu Raspberry Pi.

Gwelwn gydag adnodd technocamps [6], fod yna amrywiaeth eang o adnoddau yn fan hyn fyd. Mae'r wefan wedi'i thargedu i oedran hŷn na'r wefan codeclub. Mae gan technocamps cyrsiau cychwynnol fel y canlynol: CS 101, Deallusrwydd Artiffisial, Greenfoot (Java), Python a Scratch.

Ar wefan technolegau iaith [7], mae yna gyflwyniad i'r darnau fwyaf sylfaenol i dechnolegau iaith. Mae yna lawlyfr yn cynnwys gwybodaeth am ddeallusrwydd artiffisial, dysgu dwfn a phrosesu iaith naturiol, mae'n rhoi cyflwyniad i fewn i'r ochr damcaniaethol ohono. Yn rhedeg yn gyfagos i'r wefan yma yw'r ystorfeydd ar GitHub [22]. Yn yr ystorfeydd mae yna diwtorial ar sut i greu robot sgwrsio syml drwy "turing test lessons", mae'n addas i rhai yng nghyfnod allweddol 2 a 3. Mae'n cynnwys tair gwers, un ar sut mae cyfrifiaduron yn meddwl, yr ail ar ydy cyfrifiaduron yn gallu meddwl drostynt eu hynain ag y dwythaf am creu robot sy'n sgwrsio drwy'r defnydd o Raspberry Pi.

Yn y siop ap Apple, fedrem lawrlwytho'r ap botio [9]. Mae'n ap am chwilota'r gofod am blanedau newydd drwy raglennu. Mae'r ap wedi'i ariannu gan Lywodraeth Cymru ag ar gael am ddim. Cafodd ei greu ar gyfer plant sydd â diddordeb cael i mewn i raglennu.

Ar yr ochr mathemategol, mae'r adnodd am ddim gan Alun Morris [26] yn trafod y sylfeini eu hangen i astudio algebra llinol yn mhrifysgol. Mae'r adnodd yma yn enghraifft wych o'r fath o adnoddau mae'r iaith Gymraeg angen fwy ohono. Mae prinder iawn ar adnoddau o'r ansawdd yma yn agored i'r cyhoedd. Mae yna tair prifysgol yn cynnig cyrsiau mathemateg drwy'r Coleg Cymraeg ond dydi dim un yn darparu eu nodiadau drwy borth Coleg Cymraeg. Serch hynny mae yna gyfoeth o adnoddau mathemateg Cymraeg ar gyfer addysg o dan 18 oed, gwelwn hyn gyda'r nifer mawr o adnoddau ar wefan Dr Gareth Evans o Ysgol Creuddyn [5].

Gwelwn fod yna llawer o adnoddau ar gyfer y maes mathemateg a chyfrifiadureg ar gyfer addysg ysgol. Mae yna ddigon o adnoddau i alluogi cenhedlaeth newydd i astudio'r sylfaen gofynnol drwy'r ysgol i astudio unrhyw un o'r ddau yn brifysgol. Unwaith fyddem yn cyrraedd lefel addysg prifysgol, mae'r prinder yn amlwg. Gan fod testunau dysgu peirianyddol yn cynnwys mathemateg o lefel uwch, mae rhaid cael sylfaen datblygedig o destunau mathemateg i'w ddeall. Gallwn ddweud yr un peth am yr angenrheidrwydd o sylfaen datblygedig o gyfrifiadureg. Y broblem sydd gennym yw does yna ddim yr adnoddau yw gwneud gyda mathemateg lefel uwch nag rhaglennu yn R yn y Gymraeg.

1.2.2 Yr angen am adnoddau Cyfrwng Cymraeg

I gychwyn, rwyf eisiau creu gwefan sy'n cynnwys tiwtorialau Cymraeg am algorithmau dysgu peirianyddol oherwydd y prinder ohonyn. Mae yna gymaint o adnoddau ar gyfer pob math o algorithm trwy gyfrwng Saesneg, ond yn Gymraeg, does yna ddim byd o'r fath! Fel gwelwn yn Nhabl 1.2.1, does yna ddim llawer o adnoddau Cymraeg sy'n cynnwys deunydd datblygedig ar gael i'r cyhoedd. O fy mhrofiad personol i, rwyf

yn gwybod fod prifysgolion yng Nghymru gydag adnoddau drwy'r cyfrwng Cymraeg ar gael i'w myfyrwyr nhw yn unig. Teimlaf fod hyn yn atal y parhad o addysg ar ôl i fyfyrwyr gorffen eu hastudiaethau ffurfiol oherwydd nad oes adnoddau am ddim ar gael.

Ers i'r iaith Gymraeg cael ei rhoi ar sylfaen gyfartal i Saesneg yn Gymru yn 1993 a chael gwasanaethau cyhoeddus yn y ddwy iaith, mae wedi bod effaith domino i'r sector preifat. Mae prifysgolion yn gyrff preifat ac felly nid yw'n angenrheidiol iddyn nhw ddilyn [2], ond yn 2011 cafodd y coleg Cymraeg ei sefydlu i ornest a hyn. Mae'r coleg yn hybu prifysgolion Cymru i newid i fod yn ddwyieithog drwy weithio yn bartneriaeth gyda nhw. Hyd at heddiw mae yna fwy o fodiwlau yn cael ei gyfieithu a fwy o fodiwlau yn cael ei chynnig drwy gyfrwng y Gymraeg. Ar gyfer y garfan fydd yn graddio yn 2020 dim ond naw modiwl oedd ar gael drwy'r cyfrwng Cymraeg a rhan fwyaf ohonyn nhw yn y flwyddyn gyntaf.

Rheswm arall i wneud yn Gymraeg yw cefnogi prosiect y llywodraeth a'r wlad i gael miliwn o siaradwyr Cymraeg erbyn 2050 [18]. I gymhorthi'r prosiect yma cafodd cynllun ei rhoi allan yn hybu'r defnydd a chynnyrch o adnoddau technolegol Cymraeg [19]. Yn ogystal i greu adnoddau mae'r cynllun yn targedu y datblygiad o ddeallusrwydd artiffisial drwy edrych ar brosesu iaith naturiol yn bennaf. Mae creu'r adnodd yma yn Gymraeg yn bodloni'r ddwy adran o'r cynllun, gobeithio bydd yn hybu fwy i gychwyn creu adnoddau addysg uwch yn y maes yma. Un ateb cyflym i'r prinder fysa os fysa'r prifysgolion yn gadael ei adnoddau allan am ddim ar borth coleg Cymraeg, fel bod rhai prifysgolion yn yr Unol Daleithiau America lle mae yna ystorfa o gyrsiau gydag adnoddau agored ar wefannau fel Coursera [3].

1.3 Strwythyr

Fydd y traethawd yma yn cynnwys tair pennod ar algorithm dysgu peirianyddol. Yn pob pennod fydd yna gefndir ar beth yw'r algorithm a pryd fyddem yn ei ddefnyddio, yn ogystal fydd yna gefndir ar sut mae'r algorithm yn gweithio a beth yw'r fathemateg tu ôl i'r algorithm. Yna fydd yna diwtorial o sut i'w defnyddio yn R ac yna tiwtorial arall yn Python.

Ar gyfer y wefan, mae dudalen cartref lle fydd yno linc i bob tiwtorialau yn y ddau Python ag R. Mae'r wefan yn cynnwys y tiwtorialau ag y wybodaeth angenrheidiol i allu gwneud yr algorithmau yn yr iaith rhaglennu o'ch dewis. Dyma yr wefan: `https://dysgupeirianyddol.github.io/`.

Pennod 2

Clystyru k-Cymedr

2.1 Cefndir

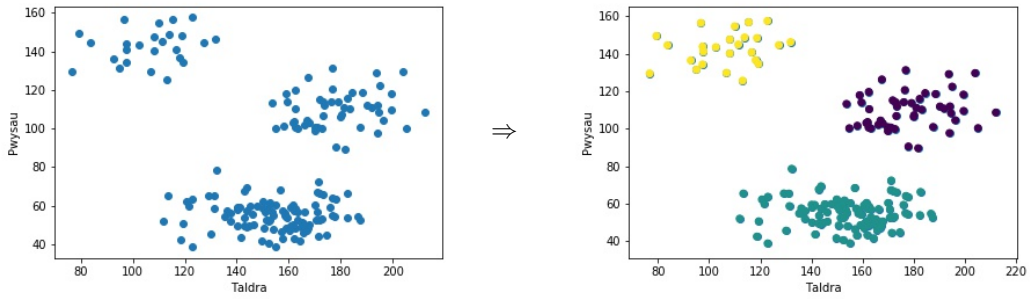
Mae clystyru k-cymedr yn ffordd o ddysgu heb oruchwyliaeth, mae'n cymryd data heb ei labelu ac yn eu sortio i mewn i k wahanol glwstwr yn yr obaith i ddarganfod rhyw strwythur doedden ddim yn gwybod yn gynharach.

I roi enghraifft gwelwch Ddarlun 2.1. Mae'r gwerthoedd ar echelin x yn cynrychioli taldra person a'r echelin y 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

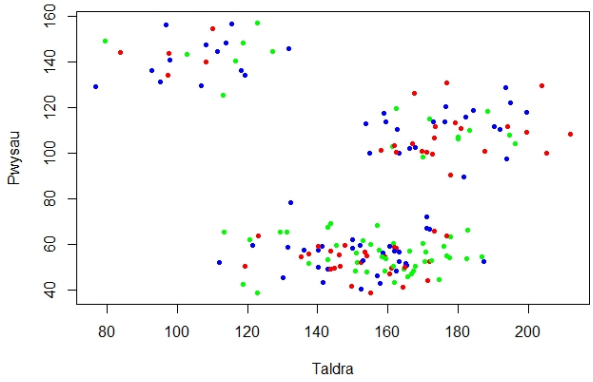
Mae clystyru k-cymedr yn syml, mae ond yn dilyn pedwar cam [29]. I wneud yn siŵr fod yn ei ffurf fwyaf sythweledol, fyddan yn defnyddio mesur pellter Ewclidaidd. Yn ogystal mae rhaid dewis k cyn cychwyn



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

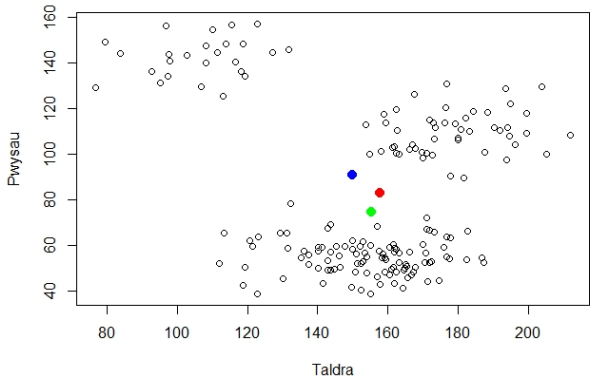
y proses. Mae'n bosib optimeiddio'r dewis o k, a gwnawn drafod hyn hwyrach ymlaen. Dyma bedwar cam yr algorithm a sut maent yn edrych pan fyddwn ni'n defnyddio'r algorithm ar y data y gwelwn yn Darlun 2.1:

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



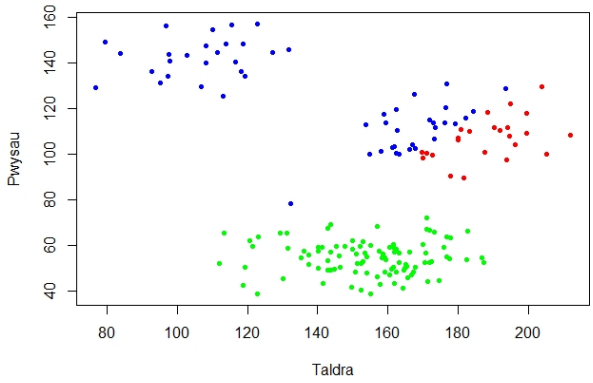
Darlun 2.2: Enghraifft o set cychwynol ar gyfer clystyru.

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



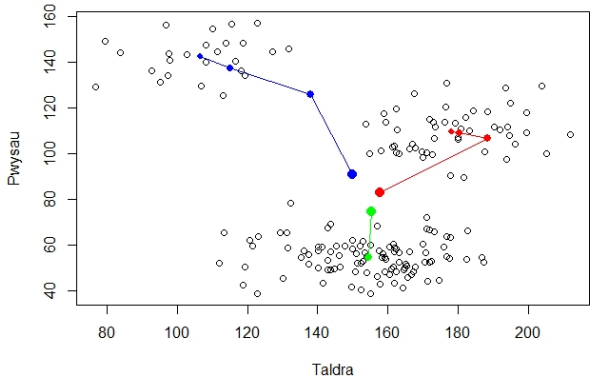
Darlun 2.3: Cyfrifiad o graidd cychwynnol phob clwstwr.

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



Darlun 2.4: Pob pwynt yn cael ei ail-aseinio i'r craidd cychwynnol agosaf.

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



Darlun 2.5: Cydgyfeiriant y broses.

2.2.2 Yr Algorithm

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 $\mathbf{c}_i$ bod yn bwynt sy'n graidd i clwstwr C_i. Ar gyfer y cam cyntaf angen aseinio pob $\mathbf{x}_j$ i ryw glwstwr C_i ar hap. Diffiniwn S_i fel y set o bwyntiau data sydd wedi'i aseinio i glwstwr C_i. Yna gan ein bod yn datgelu ein bod yn delio gyda phlân Ewclidaidd, mi fyddem yn darganfod craidd pob clwstwr gan y fformiwla ganlynol:

$$\mathbf{c}_i = \frac{1}{|S_i|} \sum_{\mathbf{x}_j \in S_i} \mathbf{x}_j \tag{2.1}$$

Yn y fformiwla uchod, gwelwn fod fectorau yn cael eu symio. Fyddem yn gwneud hyn gan symio dros elfennau, hynny yw $(x_1, x_2, \ldots, x_n) + (y_1, y_2, \ldots, y_n) = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$.

Nawr mae gan bob clwstwr craidd newydd, fedrwn aseinio pob pwynt data i'r craidd agosaf. Caiff hyn ei gwneud gan fynd drwy bob pwynt data a chyfrifo'r pellter Ewclidaidd [1] i bob canolbwynt. Yna fydd y pwynt priodol yn cael ei labelu gyda'r clwstwr sydd a'r pellter lleiaf o'i graidd i'r pwynt data. Hynny yw

$$\arg\min_{\mathbf{c}_i} ||\mathbf{c}_i, \mathbf{x}_j||^2 \tag{2.2}$$

Unwaith mae'r proses wedi'i chychwyn, angen ailadrodd y darn o ddarganfod y creiddiau newydd ac ail labelu'r pwyntiau data tan fod y creiddiau ddim yn symyd rhagor.

2.2.3 Sut i ddarganfod y k orau?

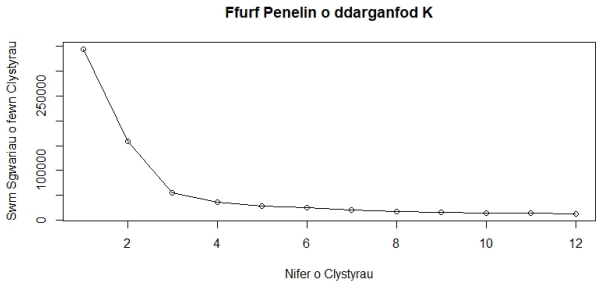
Mae yna wahanol ffurf i ddarganfod k, y ffordd fwyaf syml yw dadansoddi clystyrau drwy edrych ar y plot. Y broblem yw pan fydd y dimensiwn yn fwy nag tri, yna fydd rhaid edrych am ddulliau mwy mathemategol, edrychwn ar ddau wahanol ffordd.

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. Fydd swm sgwariau ar gyfer clwstwr C_i yn cael ei darganfod gan symio y pellter rhwng y craidd $\mathbf{c}_i$ a phob $\mathbf{x}_j$ yn ei tro ag yno ei sgwario fel welwn yn y fformiwla:

$$SS_i = \sum_{j=1}^{n} (\mathbf{x}_j - \mathbf{c}_i)^2$$

Unwaith mae gennym y graff, allwn ei ddadansoddi.



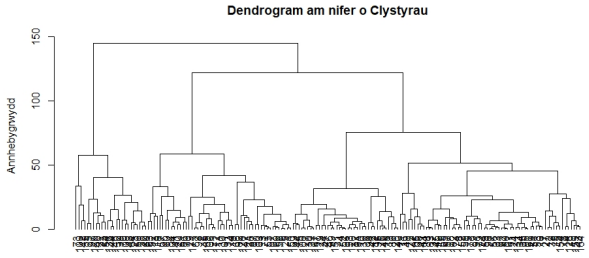
Darlun 2.6: Enghraifft o blot o k yn erbyn y cyfanswm swm o sgwariau

[1] $\mathrm{d}(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$

Yn y graff yn Narlun 2.6, gwelwn fod swm sgwariau yn fawr yn cychwyn gyda k=1 sydd yn gwneud synnwyr gan fod allanolion o unrhyw set yn cyfrannu i werth fawr o'r swm sgwariau gan ein fod methu cael ei ystyried pan does dim on un craidd i'w sortio. 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 i arddangos yr opsiwn orau o k. 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, gall hyn fod yn fesuriad o swm sgwariau. Mae Darlun 2.7 yn dangos dendrogram ar gyfer yr un data.



Darlun 2.7: Enghraifft o dendrogram

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 daldra a phwysau 175 wahanol berson. Mi allwch chi lawrlwytho y data yma o fan hyn: `https://dysgupeirianyddol.github.io/lawrlwythiadau/`

Yno fydd angen lawrlwytho a gosod y pecynnau `graphics`, `stats` ag `datasets` ar eich cyfrifiadur. 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 sesiwn ni.

Nawr mi wnawn lwytho'r data.

```
> taldrapwysau <- read.csv("taldra-pwysau.csv")
> View(taldrapwysau)
```

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:

	Taldra	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
11	158.53801	56.46608
12	143.74229	69.27893

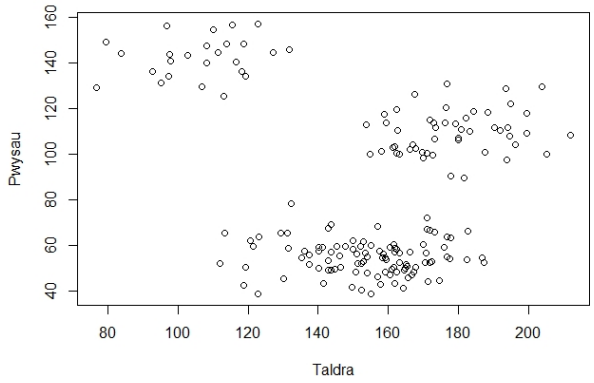
Gan fod y data hefo enwau ar gyfer y colofnau, gallwn atodi'r data i lwybr chwilio R. Bydd hyn yn gadael i ni gyfeirio at enwau colofnau'r data yn ein côd fydd yn gwneud yn lawer mwy symlach i ddeall.

```
> attach(taldrapwysau)
```

I wneud fwy o synnwyr o'r data, mi wnawn blotio'r data. Defnyddiwn `pch=21` i newid y pwyntiau ar y plot i alluogi lliwiau (ar gyfer y labeli hwyrach ymlaen).

```
> plot(Taldra, Pwysau, pch = 21)
```

Sy'n rhoi:



Gwelwn fod yna tri 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 opsiwn `nstart` i ddewis faint o setiau ar hap o ddata wedi'i labelu wnawn gymered. Welwn enghraifft o'r set ar hap hyn yn Darlun 1. Rydym yn neud hyn i wneud yn fwy debygol i ni ddarganfod yr optimwm eang, mae hyn oherwydd mae yna gymaint o optima lleol.

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

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

```
> taldrapwysau$Clwstwr3 <- kcymedr$cluster
```

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

```
> View(taldrapwysau)
```

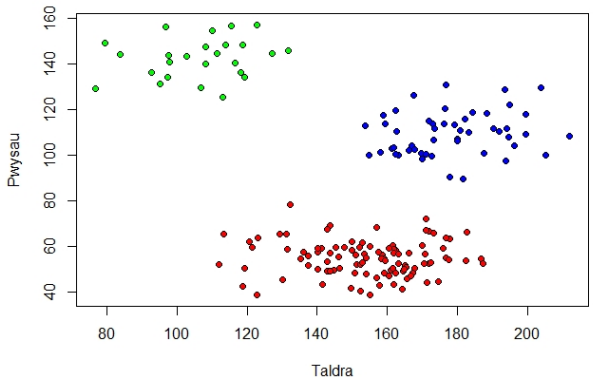
	Taldra	Pwysau	Clwstwr3
1	163.22687	100.09760	3
2	183.18087	110.18107	3
3	172.69407	99.79701	3
4	165.07549	51.66760	1
5	147.74605	59.79469	1
6	161.45039	103.04177	3
7	162.41267	58.50832	1
8	146.28025	50.36660	1
9	154.03614	47.93155	1
10	152.20904	59.79795	1
11	158.53801	56.46608	1
12	143.74229	69.27893	1

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(Taldra, 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 $k = 6$.

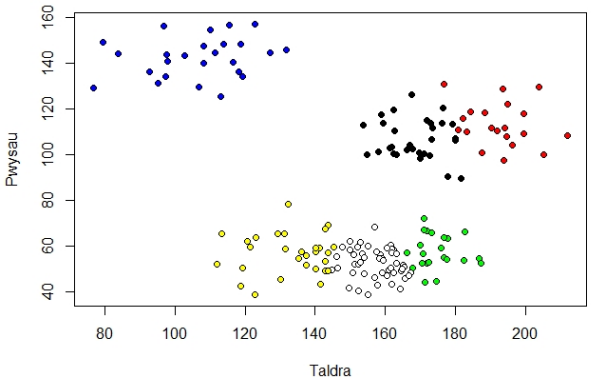
```
> kcymedr <- kmeans(taldrapwysau,6, nstart = 50)
> taldrapwysau$Clwstwr6 <- kcymedr$cluster
> View(taldrapwysau)
```

	Taldra	Pwysau	Clwstwr3	Clwstwr6
1	163.22687	100.09760	3	5
2	183.18087	110.18107	3	1
3	172.69407	99.79701	3	5
4	165.07549	51.66760	1	6
5	147.74605	59.79469	1	6
6	161.45039	103.04177	3	5
7	162.41267	58.50832	1	6
8	146.28025	50.36660	1	6
9	154.03614	47.93155	1	6
10	152.20904	59.79795	1	6
11	158.53801	56.46608	1	6
12	143.74229	69.27893	1	4

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(Taldra, Pwysau, pch = 21, bg=lliwiau[unclass(kcymedr$cluster)])
```

Sy'n rhoi:



2.4 Tiwtorial yn python

Yn y tiwtorial hwn mi wnawn edrych ar ddata o daldra a phwysau 175 wahanol berson. Mi allwch chi lawrlwytho y data yma o fan hyn: `https://dysgupeirianyddol.github.io/lawrlwythiadau/` 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('taldra-pwysau.csv')
```

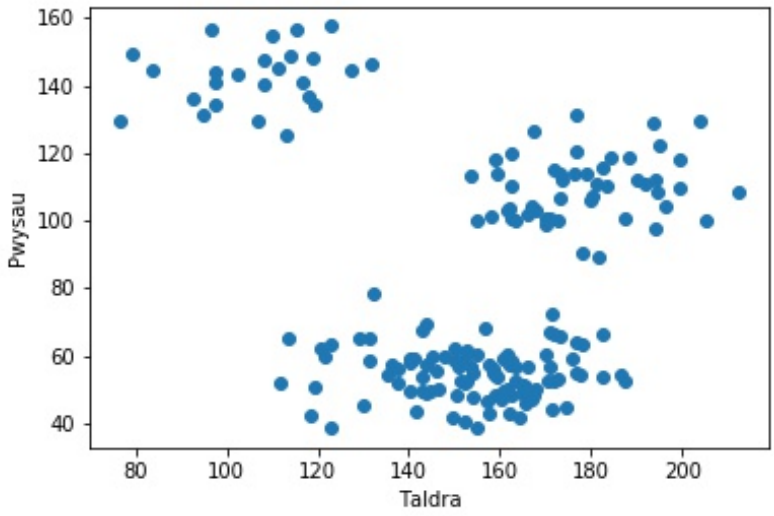
Mae'r string sydd mewnbwn y ffwythiant `pd.read_csv` yn cyfeirio at y lleoliad ar ein cyfrifiadur lle gallwn ganfod y ffeil csv priodol. Rhaid gwneud yn siŵr eich bod yn defnyddio'r lleoliad cywir i'r lleoliad o'ch ffeil chi. Unwaith fydd wedi cael ei llwytho, allwn ni gweld yn fras y data gennym ni.

```
>>> data.head()
```

	Taldra	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['Taldra'], data['Pwysau']);
>>> plt.xlabel('Taldra')
>>> plt.ylabel('Pwysau')
>>> plt.show()
```



Darlun 2.8: 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()
```

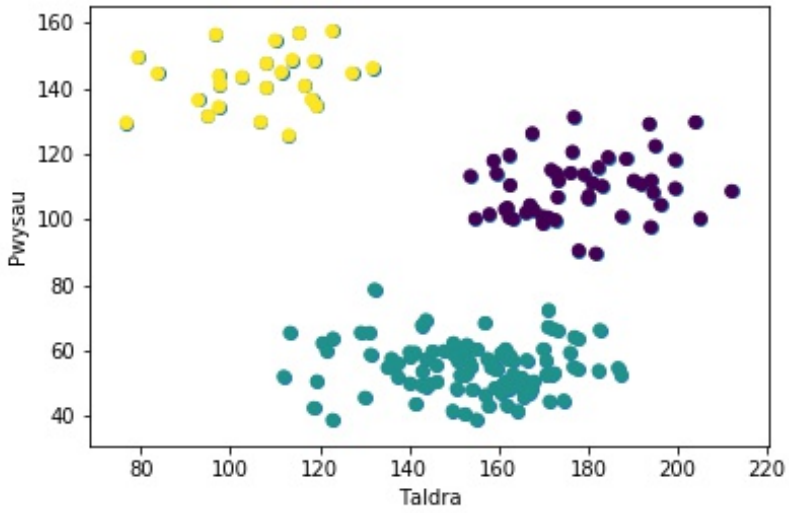
	Taldra	Pwysau	Cluster (k=3)
0	163.226866	100.097603	0
1	183.180871	110.181072	0
2	172.694074	99.797013	0
3	165.075492	51.667604	1
4	147.746048	59.794691	1

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['Taldra'], data['Pwysau'], c=data['Cluster (k=3)']);
>>> plt.xlabel('Taldra')
```

```
>>> plt.ylabel('Pwysau')
>>> plt.show()
```

Sy'n rhoi:



Darlun 2.9: 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:

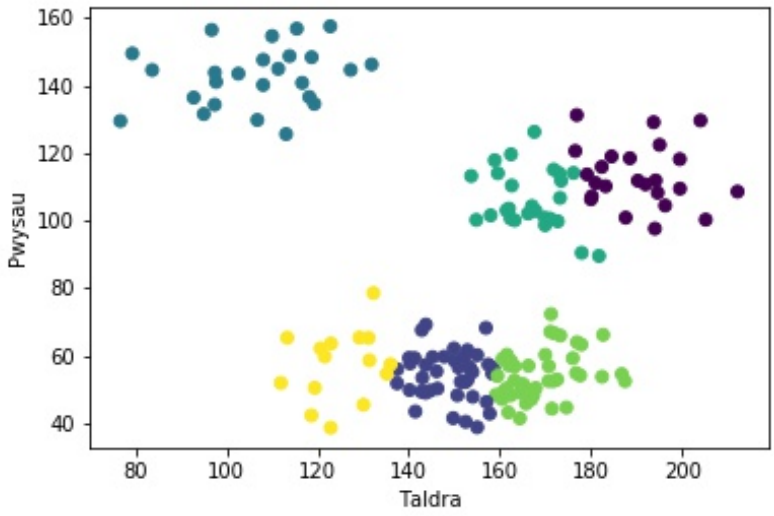
```
>>> data.head()
```

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

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

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

Sy'n rhoi:



Darlun 2.10: Sut ddylsa eich graff edrych gyda 6 clystwr.

Dyma sut y mae'r data yn edrych ar ôl phrosesu drwy glystyru 6-cymedr.

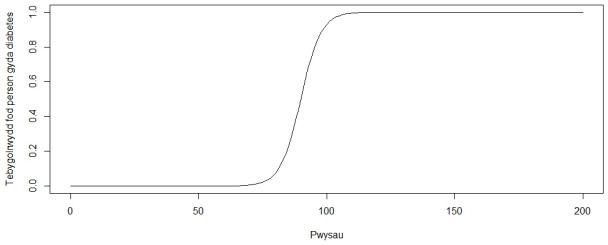
Pennod 3

Atchweliad Logistaidd

3.1 Cefndir

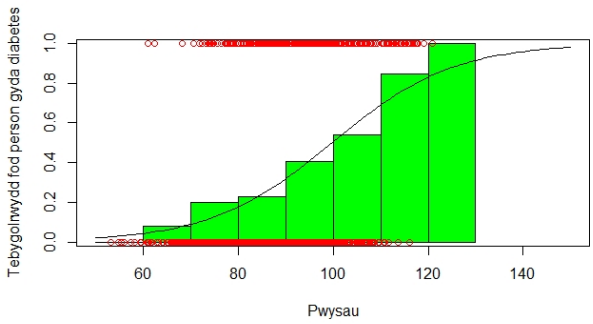
Defnyddiwn atchweliad logistaidd i fodelu'r tebygolrwydd o ddosbarthu gwrthrych i mewn i setiau deuaidd. Mae'n ddull o ddysgu dan oruchwyliaeth sy'n cael ei ddefnyddio yn aml yn academïau a diwydiannau. Gall y atchweliad cael ei ddefnyddio i weld os mae rhywun yn curo/colli, sâl/iachus neu basio/methu mewn rhyw sefyllfa benodol. Gall y syniad yma cael ei ymestyn, gall wahanol atchweliadau logistaidd cael ei rhoi yn baralel i geisio rhoi'r tebygolrwydd o liw llygaid rhyw berson er enghraifft. Mewn termau mwy cyffredinol, gall ymestyn atchweliadau logistaidd i weithio ar setiau o labeli di-deuaidd. O hyn ymlaen fyddem yn edrych ar atchweliadau gyda labeli deuaidd[24].

Mae'n hawdd delweddu sut fydd atchweliad logistaidd gydag un newidyn annibynnol. Gwelwn fod y model yn edrych fel y graff yn Ddarlun 3.1 pan hyn yw'r sefyllfa.



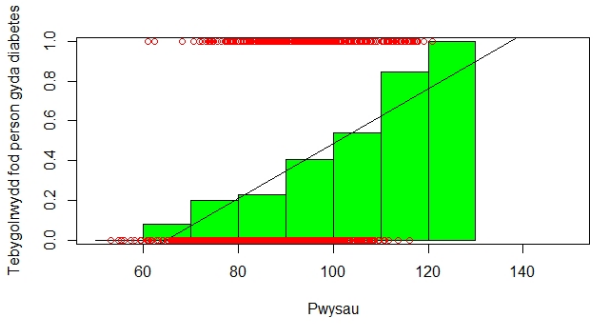
Darlun 3.1: Enghraiff o atchweliad logistaidd.

Gwelwn yn y graff nesaf fod y plot yn dangos ein data mewn ffordd rhesymol os wnawn gymharu i y cyfrannau o'r pwyntiau yn pob un o'r adrannau. Gwelwn y cyfrannau o phob adran yn wyrdd yn Ffigwr 3.1.



Darlun 3.2: Enghraiff o atchweliad logistaidd gyda siart bar i ddangos cyfrannau o'r labeli.

Os wnawn cymharu y model logistaidd i model llinol, gwelwn yn y plot yn Darlun 3.1 dydy'r model ddim yn cynnal cynhaliad ar gyfer mewnbwn llai na 60 a fwy na 120 gan fod y tebygolrwydd yn anniffiniedig (Hynny yw, dydi ddim yn bosib cael $P(\mathbf{x}) < 0$ a $P(\mathbf{x}) > 1$).



Darlun 3.3: Enghraiff o atchweliad llinol i ein data.

3.2 Sut mae atchweliad logistaidd yn gweithio?

Wnawn ddiffinio'r fector sy'n cynnwys gwybodaeth am berson j ($j \in 1, \ldots, n$) gyda $\mathbf{x}_j$ sydd hefo dimensiwn o m (hynny yw bod yna m priodoleddau). Yn ogystal, wnawn ddiffinio y_j fel label deuaidd i berson j, yr hyn rydyn ni eisiau rhagfynegi. Yna gydag ein data, gan ein bod yn perfformio dysgu dan oruchwyliaeth fyddwn yn hyfforddi'r algorithm yn defnyddio'r data hyfforddi ac yno yn gwirio'r algorithm drwy'r data profi. Felly wnawn hollti'r data fel:

Data hyfforddi: $\mathbf{x}_j$ a y_j ar gyfer $j \in \{1, \ldots, k\}$ lle mae $k < n$

Data profi: $\mathbf{x}_j$ a y_j ar gyfer $j \in \{k+1, \ldots, n\}$

Mae'r model logistaidd yn cymryd y gwrthdro o ffurf logit, mae hyn yn cael ei ddangos yn Hafaliad 3.1 lle mae $z \in (-\infty, \infty)$.

$$f(z) = \frac{1}{1 + e^{-z}} \tag{3.1}$$

lle:

$$z = \alpha + \beta_1 X_1 + \cdots + \beta_m X_m \tag{3.2}$$

Felly mae hafaliad 3.3 yn dangos y model cyfan.

$$P(\mathbf{x}) = P(y = 1|x_1 \ldots x_k) = \frac{1}{1 + e^{-(\alpha + \sum_{i=1}^{m} \beta_i x_i)}} \tag{3.3}$$

3.2.1 Yr Algorithm

Fydd α a β y paramedrau fyddem yn trio amcangyfrif o wybod $\mathbf{x}$ ac y y data hyfforddi. I amcangyfrif hyn wnawn ddefnyddio'r dull amcangyfrif tebygoliaeth fwyaf. Cymerwn $\hat{\mathbf{z}}$ i fod y fector o baramedrau fyddem yn amcangyfrif. Yna mae gennym y amcangyfrif tebygoliaeth ganlynol a fyddem yn trio cael y gwerth agosaf i 1. [**?**]

$$L(\hat{\mathbf{z}}) = \prod_{s \in y_i = 1} p(x_i) \prod_{s \in y_i = 0} (1 - p(x_i)) \tag{3.4}$$

Mae Hafaliad 3.4 yn trio uchafsymio y lluoswm o phob tebygolrwydd ag oedd yn edrych ar labeli y data hyfforddi. Gan fod rhai labeli am fod yn 0 a lleill gyda 1 ac rydym yn cymryd y lluoswm o'r rhifau gyda label o 0 ag 1, fydd y ddau luoswm ar wahân yn cydgyfeirio am ddau werth gwahanol. Felly rydym yn dewis newid y lluoswm gyda labeli 0 i gydgyfeirio tuag at 1 drwy luosi'r lluoswm $1 - p(x_i)$ i bob un gyda label 0. Mae'n neud synnwyr cydgyfeirio i 1 dros 0 gan fod i gydgyfeirio tuag 0 does dim ond angen un gwerth o 0 yn y lluoswm.

Sydd yn gallu cael ei symleiddio i:

$$L(\hat{\mathbf{z}}) = \prod_{i=1}^{k} p(x_i)^{y_i} (1 - p(x_i))^{1 - y_i}$$

Nawr fyddem yn cymryd y log o'r amcangyfrif tebygoliaeth.

$$\log L(\hat{\mathbf{z}}) = \sum_{i=1}^{n} y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i))$$

Sydd yn symleiddio i:

$$\log L(\hat{\mathbf{z}}) = \sum_{i=1}^{n} y_i \log\left(\frac{1}{1 + e^{-\hat{\mathbf{z}}\mathbf{x}}}\right) + (1 - y_i) \log\left(\frac{e^{-\hat{\mathbf{z}}\mathbf{x}}}{1 + e^{-\hat{\mathbf{z}}\mathbf{x}}}\right)$$

ac felly:

$$\log L(\hat{\mathbf{z}}) = \sum_{i=1}^{n} y_i \hat{\mathbf{z}} x_i - \log(1 + e^{\hat{\mathbf{z}} x_i})$$

Yna mae gennym y log o'r amcangyfrif tebygoliaeth. Rydym eisiau darganfod y gwerth o z lle mae'r log o'r amcangyfrif tebygoliaeth ar ei fwyaf.

$$\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} \log L(\mathbf{z})$$

Does yna ddim ffordd bendant o ddatrys yr hafaliad uchod, fydd angen defnyddio algorithmau fel swm lleiaf sgwariau wedi eu hail-bwyso drwy iteriadau [15] neu algorithm cof-cyfengedig Broyden–Fletcher–Goldfarb–Shanno (BFGS) [14] fel gwelwn yn y algorithmau yn R ac Python yn y drefn honno. Mae dull disgyniad fwyaf yn algorithm poblogaidd arall.

Mae'r dull disgyniad fwyaf yn algorithm optimeiddiaeth trefn cyntaf i darganfod isafbwynt lleol o ffwythiant gall ei ddifferu. Mae'r algorithm yn cymryd camau yn gyfraneddol i'r graddiant yn y pwynt yno. Fydd yr algorithm am phob cam yn edrych yn debyg i Hafaliad 3.5 gyda a yn rhyw bwynt a f yn y ffwythiant.

$$a_{n+1} = a_n - \nabla f \tag{3.5}$$

Mae algorithm BFGS yn cychwyn gydag amcangyfrif cychwynnol o'r gwerth optimaidd x_0, ac yno yn iteru drwy ddefnyddio elfennau o matrics gwrthdro Hessian sef yr ail ddeilliad o'r ffwythiant. Mae'r Hessian yn cynnwys gwybodaeth bwysig am y crymedd.

Ar gyfer y swm lleiaf o sgwariau wedi eu hail-bwyso drwy iteriadau, mae'r algorithm yn cydgyfeirio tuag at y pwysau optimaidd ar gyfer y cyfernodau yn Hafaliad 3.2. Mae angen ail bwyso oherwydd mae'r amrywiant yn newid gydag x. Mae lle mae'r amrywiant ar ei fwyaf yn y model am dynodi lle fydd gromlin ein model.

3.2.2 Profi'r model

Unwaith mae gennym amcangyfrif o'r paramedrau, mae angen darganfod pa mor dda yw ein model logistaidd. I wneud hwn byddwn yn rhoi ein data profi x_j i mewn i'r model, a cynharu'r allbwn gyda y_j. Fel allbwn cawn tebygolrywdd, rhif rhwng 0 ac 1, yna wnawn talgrynnu'r allbwn i cael label. Hynny yw os gawn ni allbwn llai na hanner, rhoddwn label 0, fel arall bydd yn derbyn label 1. Wedyn mae gennym ein rhagfynegiad am label pob person, yna gallwn ddarganfod cyfradd llwyddiant ein model gan:

$$1 - \sum_{j=k+1}^{n} \frac{(P(\mathbf{x}_j) - y_j)^2}{n - k} \tag{3.6}$$

Mae'r swm yma yn cymryd y canran o camgymeriadau rhwng y data hyfforddi a'r data profi ac yna yn ei tynnu i ffwrdd o 1, byddwn yn menegi hwn fel y cyfradd llwyddiant. I darganfod y canran o camgymeriadau

fyddwn yn tynnu y labeli y data profi oddi wrth labeli y data hyfforddi, ac yno yn sgwario yr fector canlyniadol. Fydd y gweithrediadau yma yn gweithio yn ôl elfen. Unwaith fydd y fector wedi'i sgwario fydd pob elfen sy'n dangos 1 yn dangos camgymeriad ac felly fydd 0 yn dangos rhagfynegiad cywir. Yna fydd y fector yn cael ei symio ac fydd y cyfanswm yn cael ei rhannu gan y nifer o elfennau.

3.3 Tiwtorial yn R

Yn yr enghraifft hon, fyddwn yn edrych ar ddata ar 1000 o bobol, fydd y data yn cynnwys gwybodaeth ar taldra, pwysau, maint gwasg, oed, rhyw ag oes gan y person clefyd y siwgr. Mae'n bosib lawrlwytho'r data oddi: `https://dysgupeirianyddol.github.io/lawrlwythiadau/`

Ar gyfer gwneud atchweliad logistaidd, mae angen y pecyn `stats` a wnawn ei lawrlwytho a'i gosod gan redeg y canlynol:

```
> install.packages("stats")
> library(stats)
```

Yna fydd rhaid lwytho'r data i mewn a'i arbed fel newidyn. Fydd rhaid neud yn siŵr fod y ffwythiant `read.csv` yn cael ei chyfeirio tuag at y lleoliad cywir o le mae eich data chi wedi'i gadw.

```
> data <- read.csv("data_logistic.csv")
```

Unwaith ei fod wedi llwytho, mae'n bosib gweld y data:

```
> View(data)
```

	Taldra	Pwysau	MaintGwasg	Oed	Rhyw	Clefyd_Siwgr
1	170.7197	87.70995	40.21594	57	Gwryw	0
2	159.2646	91.67977	39.95974	62	Gwryw	1
3	154.9078	98.35737	34.58633	29	Gwryw	0
4	168.5475	93.48007	41.47911	54	Benyw	1
5	175.8423	79.65120	34.53736	23	Benyw	0
6	169.7488	91.99920	39.35820	44	Benyw	1
7	143.8323	102.34901	42.13036	59	Benyw	1
8	157.2371	88.39940	41.19644	24	Benyw	1
9	197.6471	102.07920	36.38416	60	Gwryw	0
10	156.2165	91.67034	37.50940	27	Gwryw	0
11	166.3419	81.24311	33.45878	40	Benyw	1
12	173.4895	100.70393	38.05125	19	Gwryw	0

Nawr wnawn rannu'r data fel bod 70% o'r data yw'r data hyfforddi ac 30% o'r data yw'r data profi. Fyddwn yn rhannu'r data ar hap.

```
> rhifau <- c(1:1000)
> rhifauhyfforddi <- sample(x = rhifau, size = 700, replace = FALSE)
> rhifauprofi <- setdiff(rhifau, rhifauhyfforddi)
```

Mae'r côd uchod yn rhannu'r setiau gan ddefnyddio eu indecs (rhif y rhes) yn y data ac yno mae'r côd isod yn rhannu'r fectorau i mewn i setiau arwahan.

```
> hyfforddi <- data[rhifauhyfforddi,]
> profi <- data[rhifauprofi,]
```

Nawr rydym yn barod i greu'r model logistaidd. I greu'r model fyddem yn rhedeg y côd gan ddefnyddio y ffwythiant `glm`, sydd yn fyr am "Generalized Linear Models", sydd yn golygu gall y ffwythiant cael ei ddefnyddio am lawer fwy o atchweliadau na logistaidd yn unig. Oherwydd hyn mae angen gosod yr opsiwn `family` i `binomial`. I ddilyn strwythur o'r algorithm, byddwn yn creu'r model o'r data hyfforddi yn unig.

```
> atchweliad <- glm(Clefyd_Siwgr ~ Taldra + Pwysau + Oed + Rhyw + MaintGwasg,
+                   family = binomial,
+                   data = hyfforddi)
```

Unwaith mae'r model wedi'i greu, gallwn weld eu paramedrau sydd wedi cael ei amcangyfrif:

```
> atchweliad$coefficients
 (Intercept)       Taldra       Pwysau          Oed
22.858432583 -0.254347133  0.215272873  0.057360113
   RhywGwryw   MaintGwasg
-8.074052403 -0.003206262
```

Felly mae'r model sydd gennym, i dri lle degl, yn edrych fel:

$$P(\mathbf{x}) = \frac{1}{1 + e^{-22.858 + 0.254 x_{\text{Taldra}} - 0.215 x_{\text{Pwysau}} - 0.057 x_{\text{Oed}} + 8.074 x_{\text{Rhyw}} + 0.003 x_{\text{MaintGwasg}}}}$$

Gan fod ein model wedi'i chwblhau, gallwn weld sut mae'n perfformio yn penderfynu os oes gan bobl y set profi clefyd siwgr neu peidio. Geith hyn ei wneud yn defnyddio'r ffwythiant `predict` a dewis yr opsiwn `type` fel `response` i gael allbwn o debygolrwydd. Heb wneud hyn, fydd yr allbwn yn cyfrifo z o Hafaliad 3.2.

```
> canlyniad <- predict(object = atchweliad, newdata = profi, type = "response")
> canlyniad <- round(canlyniad, digits = 0)
> canlyniad <- unname(canlyniad)
```

Fyddem yn ogystal yn talgrynnu'r tebygolrwydd o bob person i cael dewis ar os gennym clefyd siwgr neu peidio. Wedyn fyddem yn tynnu i ffwrdd y rhifau o'r rhesi ar y fector o labeli. Nawr gennym y rhagfynegiad a'r canlyniadau gwreiddiol, gallwn gyfrifo'r canran o'r ddau set sy'n debyg. Gallwn gyfrifo yn y ffurf ganlynol gan fod ein setiau yn ddeuaidd:

```
> 1-(sum((test[,6]-unname(canlyniad))**2)/length(test[,6]))
0.8833333
```

Fel y gwelwn, mae ein model gyda cywirdeb o 88% ar gyfer y data sydd gennym. Gallwn ni defnyddio y model rydym wedi creu i benderfynu ar os gan berson newydd ar hap clefyd siwgr neu ddim. Gwelwn hyn gan gyflwyno dyn gydag Taldra o 160, pwysau 92, maint gwasg o 34 ag ugain oed yn y côd isod:

```
> unname(round(predict(object = atchweliad,
+                      newdata = data.frame(Taldra = 160,
+                                           Pwysau = 92,
+                                           MaintGwasg = 34,
+                                           Oed = 20,
+                                           Rhyw = "Gwryw"),
+                      type = "response"),
+              digits = 0))
+
0
```

Am y person yma gwelwn fod y model wedi rhagfynegu nad oes ganddo clefyd siwgr. Os wnawn ystyried person gyda'r un nodweddion ond yn fenyw:

```
> unname(round(predict(object = atchweliad,
+                      newdata = data.frame(Taldra = 160,
+                                           Pwysau = 92,
+                                           MaintGwasg = 34,
+                                           Oed = 20,
+                                           Rhyw = "Benyw"),
+                      type = "response"),
+              digits = 0))
+
1
```

Gwelwn fod y model yn rhagfynegi ei fod hi gyda clefyd siwgr.

3.4 Tiwtorial yn Python

Ar gyfer cynhyrchu atchweliad logistaidd yn python mae rhaid i ni ddefnyddio'r pecynnau `sklearn`, a `pandas` i trin y data. Wnawn lwytho'r pecynnau gan redeg y côd yma:

```
>>> from sklearn.linear_model import LogisticRegression
>>> import pandas as pd
```

Nawr mae angen llwytho'r data, wnawn ddefnyddio data sy'n cynnwys 1000 o gofnodion data ar fesuriadau pobl yn cynnwys taldra, pwysau, maint gwasg, oed, rhyw ag oes gan y person clefyd y siwgr. Cewch ei lawrlwytho o `https://dysgupeirianyddol.github.io/lawrlwythiadau/`.

```
>>> data = pd.read_csv('data_logistic.csv')
```

Gallwn gweld yr data gan rhedeg:

```
>>> data.head()
```

	Taldra	Pwysau	MaintGwasg	Oed	Rhyw	Clefyd_Siwgr
0	170.719652	87.709946	40.215944	57	1	0
1	159.264575	91.679774	39.959742	62	1	1
2	154.907775	98.357373	34.586330	29	1	0
3	168.547460	93.480071	41.479106	54	0	1
4	175.842260	79.651198	34.537361	23	0	0

Gan fod ein data gyda rhyw wedi cael ei diffinio gyda'r geiriau "Gwryw" a "Benyw", mae Python yn cael trafferth yn delio gyda nhw. Felly nawn trawsnewid nhw i newidyn deuaidd (set o 1 a 0).

```
>>> data['Rhyw'] = data['Rhyw'].apply(lambda x: int(x =='Gwryw'))
```

Nawr fydd rhaid i ni rannu'r data i ddata hyfforddi ag data profi.

```
>>> hyfforddi = data.sample(frac = 0.7)
>>> profi = data.drop(hyfforddi.index)
```

Mae'r wybodaeth rydym angen i greu model logistaidd angen fod yn fatrics yn Python, felly:

```
>>> X = hyfforddi[['Taldra', 'Pwysau', 'MaintGwasg', 'Oed', 'Rhyw']].as_matrix()
>>> y = hyfforddi['Clefyd_Siwgr'].as_matrix()
>>> X_profi = profi[['Taldra', 'Pwysau', 'MaintGwasg', 'Oed', 'Rhyw']].as_matrix()
>>> y_profi = profi['Clefyd_Siwgr'].as_matrix()
```

I redeg yr atchweliad logistaidd wnawn ddefnyddio'r ffwythiant yn `sklearn`. Wnawn wneud gan redeg y côd canlynol:

```
>>> clf = LogisticRegression(random_state=0).fit(X, y)
```

Gallwn edrych ar y rhyngdoriad gan

```
>>> clf.intercept_
array([ 1.70933553])
```

ac yna y paramedrau eraill:

```
>>> clf.coef_
array([[-0.12384414,  0.14710498,  0.12995053,  0.04682347, -4.80795833]])
```

Felly dyma yw ein model i dri lle degol:

$$P(\mathbf{x}) = \frac{1}{1 + e^{-1.709 + 0.124 x_{\text{Taldra}} + 0.147 x_{\text{Pwysau}} - 0.047 x_{\text{Oed}} + 4.808 x_{\text{Rhyw}} - 0.130 x_{\text{MaintGwasg}}}}$$

Nawr gallwn ni cyfrifo'r gyfradd llwyddiant o ein model ar y data profi. Gallwn ei chyfrifo yn y ffordd ganlynol oherwydd ein bod yn delio gyda data deuaidd.

```
>>> 1-(sum((clf.predict(X_profi)-y_profi)**2)/len(y_profi))
0.9166666666666663
```

Felly mae ein model logistaidd yn python yn rhoi cyfradd llwyddiant o tua 92%. Gallwn nawr ei ddefnyddio ar gyfer rhyw berson tu allan i ein data. Os oes gennym wryw gyda thaldra o 171, pwysau o 130 a maint gwasg ag oed o 40; gallwn ragfynegi os oes gan y person clefyd siwgr ta ddim.

```
>>> clf.predict([[171, 130, 40, 40, 1]])
array([1], dtype=int64)
```

Gwelwn fod y model logistaidd yn rhagfynegu bod gan y person hwn clefyd siwgr. Nawr nawn drio gyda person tebyg ond gyda phwysau o 90 yn lle.

```
>>> clf.predict([[171, 90, 40, 40,1]])
array([0], dtype=int64)
```

Gwelwn nad oes gan y person yma clefyd siwgr.

Pennod 4

Dosbarthiad Naïf Bayes

4.1 Cefndir

Mae dosbarthiad naïf Bayes fel arfer yn cael ei ystyried i fod y dull hawsaf o ddosbarthu sydd ogystal yn weddol gywir. Gall dosbarthiad naïf Bayes ei ddefnyddio ar ddata di-dor ag arwahanol. I gael creu dosbarthiad ar set ddata di-dor mae rhaid i ni dybio'r fath o ddata sydd gennym, fel arfer y dybiaeth hon yw bod yn dilyn dosraniad normal. O hyn ymlaen fyddwn yn tybio fod y data sydd gennym yn arwahanol.

Ystyriwch fod gennych set ddata yn cynnwys gwybodaeth am y tywydd am y penwythnosau blaenorol ac p'un ai fod rhyw berson wedi chwarae golff y penwythnos yna. Yn defnyddio'r wybodaeth yma gallwn cyfrifo tebygolrwydd o fynd i chwarae golff rhyw benwythnos i ddod gan ddefnyddio gwybodaeth am y tywydd y benwythnos yna. Enghraifft arall fwy defnyddiol yw categoreiddio dogfenni ar gyfer pwnc, yn syml fydd hyn yn cael ei wneud drwy edrych ar amledd geiriau.[8]

4.2 Yr Algorithm

Mae'r dull yma wedi cael ei adeiladu ar sylfaen o reol Bayes sy'n ymwneud gyda thebygolrwydd amodol ag tebygolrwydd ymylol [25]. Mae Hafaliad 4.2 yn dangos rheol Bayes, lle mae P(A) ydy rhagdebygolrwydd o A, P(A|B) yw'r ôl debygolrwydd o A, ag P(B|A) yw'r tebygolrwydd amodol o B rhoddir A. P(B) yw tebygolrwydd ymylol o B.

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{4.1}$$

Mae'r algorithm yn defnyddio rheol Bayes ag yn ei estyn gan dybio fod pob pâr o briodoleddau yn dilyn annibyniaeth amodol [13]. Er mwyn gweld hyn wnawn ddiffinio ein newidynnau.

Gadwn i y cynrychioli'r labeli o ddosbarthiadau, hefyd wnawn ddiffinio x_i i fod y priodoledd dibynnol lle mae $i \in \{1, \ldots, n\}$ yn cynrychioli n priodoleddau gwahanol.

Felly mae rheol Bayes yn edrych fel:

$$P(y|x_1,\ldots,x_n) = \frac{P(y)P(x_1,\ldots,x_n|y)}{P(x_1,\ldots,x_n)} \tag{4.2}$$

Yn defnyddio rhagdybiaeth naïf ar annibyniaeth amodol:

$$P(x_i|y,x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n) = P(x_i|y) \tag{4.3}$$

Gallwn drawsnewid rheol Bayes i reol lawer iawn symlach i'w cyfrifo.

$$P(y|x_1,\ldots,x_n) = \frac{P(y)\Pi_{i=1}^{n}P(x_i|y)}{P(x_1,\ldots,x_n)} \tag{4.4}$$

Fyddwn yn defnyddio y fformiwla yma i cyfrifo yr ôl tebygolrwydd ar gyfer pob gwerth o y sydd gennym ar gael. Felly oherwydd dim ond y fydd yn newid gennym, gallwn sylwi fod yr enwadur am fod yn hafal i bob cyfrifiad, felly i gymharu yr ôl tebygolrwydd i bob gwerth gallwn ei anwybyddu. Felly i cymharu tebygolrwydd am wahanol labeli fyddwn yn cymharu drwy edrych ar y rhifiadur yn unig.

Mae sut rydym yn cyfrifo $P(x_i|y)$ yn dibynnu yn hollol ar y priodoledd, gan ein bod am edrych ar set ddata sy'n gategoreiddiol yn y tiwtorialau fyddwn yn edrych ar ffurf dosbarthiad naïf Bayes gategoreiddiol yn benodol. Felly i ddata wedi'i chategoreiddio, ar gyfer categori k ag dosbarth h:

$$P(x_i = k|y = h) = \frac{N_{k,h}}{N_h} \tag{4.5}$$

Lle mae N yn dynodi'r nifer o samplau. Gan fod y rhif yma o fewn lluoswm ac mae'n bosib cael tebygolrwydd o 0, fyddem yn defnyddio ffurf llyfnhau Laplace i addasu ein niferoedd i fod yn fwy tebygol i ddosbarthiad unffurf. Mi wnawn wneud hyn gan gyflwyno newidyn newydd θ a dynodwyd n_i y nifer o categoreiddiau gwahanol o fewn priodoledd i. Yna:

$$P(x_i = k|y = h;\theta) = \frac{N_{k,h}+\theta}{N_h+\theta n_i} \tag{4.6}$$

Y fwyaf mae'r gwerth o θ, y fwyaf agos i'r dosraniad unffurf fydd y niferoedd yn mynd. Mae'r tebygolrwydd ymylol $P(y)$ yn ddigon hawdd i'w cyfrifo:

$$P(y) = \frac{N_h}{N} \tag{4.7}$$

Felly i gloi, gallwn ysgrifennu fformiwla dosbarthiad naïf Bayes categoreiddiol fel

$$P(y|x_1,\ldots,x_n;\theta) = \frac{N_h}{NP(x_1,\ldots,x_n)}\prod_{i=1}^{n}\frac{N_{k,h}+\theta}{N_h+\theta n_i} \tag{4.8}$$

ac felly:

$$P(y|x_1,\ldots,x_n;\theta) \propto \frac{N_h}{N}\prod_{i=1}^{n}\frac{N_{k,h}+\theta}{N_h+\theta n_i} \tag{4.9}$$

4.3 Tiwtorial yn R

Ar gyfer y tiwtorial yma fydd rhaid i chi lawrlwytho'r set ddata o `https://dysgupeirianyddol.github.io/lawrlwythiadau/`. Mae'r set ddata rydym yn edrych ar yn cynnwys gwybodaeth am 200 person a wnaeth pleidleisio mewn etholiad. Mae'r data yn cynnwys y blaid wleidyddol a wnaeth phob un pleidleisio am ag y priodoleddau canlynol: gwlad, rhyw, os yw'n hapus gyda'r blaid yn bwêr, math o dŷ ag os yw'n berchen ar fwy nag un tŷ.

Unwaith fod y set ddata wedi lawrlwytho, wnawn lwytho'r data i mewn a'i arbed fel newidyn. Dewisom etholiad fel newidyn.

```
> etholiad <- read.csv("etholiad.csv")
```

Nawr gallwn weld y data gan redeg y canlynol:

```
> View(etholiad)
```

	Gwlad	Rhyw	Hapus.gyda.r.plaid.yn.pwer	Math.o.dy	Perchen.ar.fwy.nag.un.ty	Plaid.Wleidyddol
1	Lloegr	Benyw	Na	Datgysylltiedig	Na	Democrataidd Rhyddfrydol
2	Cymru	Benyw	Na	Ty Rhes	Ydw	Plaid Geidwadol
3	Lloegr	Benyw	Na	Byngalo	Na	Plaid Geidwadol
4	Alban	Benyw	Na	Ty Rhes	Na	Plaid Lafur
5	Alban	Gwryw	Na	Ty Par	Na	Plaid Geidwadol
6	Gogledd Iwerddon	Benyw	Na	Ty Par	Na	Plaid Geidwadol
7	Lloegr	Benyw	Na	Datgysylltiedig	Na	Plaid Lafur
8	Alban	Gwryw	Na	Byngalo	Na	Plaid Geidwadol
9	Lloegr	Benyw	Na	Fflat	Na	Plaid Geidwadol
10	Cymru	Benyw	Na	Ty Rhes	Na	Plaid Lafur
11	Gogledd Iwerddon	Benyw	Na	Datgysylltiedig	Na	Plaid Lafur
12	Alban	Benyw	Na	Ty Rhes	Na	Plaid Lafur

Darlun 4.1: Data ar etholiad.

Mae'r ffwythiant canlynol yn R yn cyfrifo'r tebygolrwydd amodol pryd rydym yn gwybod y blaid, gall y ffwythiant cael ei ddefnyddio ar gyfer unrhyw blaid fel gwelwn yn cael ei ddefnyddio pan fyddem yn rhagfynegi hwyrach ymlaen.

```
> tebygolrwydd_amodol <- function(plaid, colofn, gwerth, data)
+ {
+     rhifiadur <- sum(data[which(data[,"Plaid.Wleidyddol"]==plaid),names(data)[colofn]]==gwerth)+1
```

```
+    enwadur <- length(data[which(data[,"Plaid.Wleidyddol"]==plaid),names(data)[colofn]])
+               +length(levels(etholiad[,colofn])))
+    rhifiadur/enwadur
+ }
```

Gwelwn fod y ffwythiant uchod yn defnyddio'r dull cyfrif ffug. Yn y ffwythiant nesaf rydym yn cyfrifo y tebygolrwydd o bob plaid yn unigol.

```
> tebygolrwydd <- function(plaid, data)
+ {
+   sum(etholiad[,"Plaid.Wleidyddol"]==plaid)/length(etholiad[,"Plaid.Wleidyddol"])
+ }
```

Nawr mae gennym y ddwy gydran o'r fformiwla naïf Bayes. Mae'r ffwythiant isod yn defnyddio'r ddau ffwythiant blaenorol i gyfrifo'r tebygolrwydd o ddosbarthu y plyg i bob plaid. Wnawn wneud hyn gan creu fector gyda'r tebygolrwydd i bob plaid yn drefn briodol i hyn wnaethom gyda'r fector pleidiau ac yna cyfrifo'r uchafswm.

```
> rhagfynegi <- function(data, plyg)
+ {
+   pleidiau <- c("Plaid Geidwadol", "Plaid Lafur", "Democrataidd Rhyddfrydol", "Arall")
+   fector_tebygolrwydd <- c()
+   for (i in pleidiau)
+   {
+     tebygolrwydd_plaid <- 1
+     for (j in 1:(length(names(data))-1))
+     {
+       tebygolrwydd_plaid <- tebygolrwydd_plaid *
+                             tebygolrwydd_amodol(i, j, plyg[j], data)
+     }
+     tebygolrwydd_plaid <- tebygolrwydd_plaid * tebygolrwydd(i, data)
+     fector_tebygolrwydd <- c(fector_tebygolrwydd, tebygolrwydd_plaid)
+   }
+   return(pleidiau[which.max(fector_tebygolrwydd)])
+ }
```

Nawr wnawn greu plyg i gael dosbarthu ei blaid. Defnyddiwn y nodweddion yma:

```
> plyg <- c("Lloegr", "Benyw", "Na", "Ty Rhes", "Na")
```

```
> rhagfynegi(data = etholiad,plyg = plyg)
[1] "Plaid Lafur"
```

Defnyddiwn y ffwythiant a chawsom allbwn fod y plyg yn cael ei dosbarthu i bleidleisio am Lafur. Wnawn creu plyg arall i profi'r algorithm:

```
> plyg <- c("Lloegr", "Gwryw", "Yndw", "Ty Rhes", "Yndw")
```

```
> rhagfynegi(data = etholiad,plyg = plyg)
[1] "Plaid Ceidwadol"
```

Gan ddefnyddio'r ffwythiant cafodd y plyg yma ei dosbarthu i bleidleisio am Blaid Geidwadol.

4.4 Tiwtorial yn Python

Yn y tiwtorial hwn fyddwn yn defnyddio set ddata o ryw etholiad sydd yn cynnwys y priodoleddau o 200 berson, y priodoleddau yw gwlad breswyl, rhyw, os yw'n hapus gyda'r blaid yn bwêr, math o dŷ sydd gennym ac os oes gennym fwy nag un tŷ. Mae'r set ddata ar gael ar y wefan `https://dysgupeirianyddol.github.io/lawrlwythiadau/`. I gychwyn dosbarthiad naïf Bayes yn Python mae rhaid llwytho'r pecyn `pandas`. Llwythwn y pecyn fel:

```
>>> import pandas as pd
```

Yna wnawn lwytho'r data. Yna fedrem weld y data gan ddefnyddio `data.head()`.

```
>>> data = pd.read_csv('etholiad.csv')
>>> data.head()
```

Gwelwn fod pob priodoledd yn ddata wedi'i chategoreiddio sydd yn gadael ni ddefnyddio

	Gwlad	Rhyw	Hapus.gyda.r.plaid.yn.pwer	Math.o.dy	Perchen.ar.fwy.nag.un.ty	Plaid.Wleidyddol
0	Lloegr	Benyw	Na	Datgysylltiedig	Na	Democrataidd Rhyddfrydol
1	Cymru	Benyw	Na	Ty Rhes	Yndw	Plaid Geidwadol
2	Lloegr	Benyw	Na	Byngalo	Na	Plaid Geidwadol
3	Alban	Benyw	Na	Ty Rhes	Na	Plaid Lafur
4	Alban	Gwryw	Na	Ty Par	Na	Plaid Geidwadol

Darlun 4.2: Data etholiad yn Python

Byddwn yn creu ddau ffwythiant, un i'r ddau debygolrwydd sy'n cael ei chynnwys yn y rhifiadur o'r fformiwla naïf Bayes. Mae'r ffwythiant cyntaf yn edrych ar y tebygolrwydd amodol, fydd y ffwythiant yn cymryd

mewnbwn o blaid, colofn, gwerth y plyg ag y data defnyddiwn. Mae'r ffwythiant hefyd yn defnyddio cyfrififfug i wneud yn siŵr fydd yna ddim tebygolrwydd o faint 0 yn cael ei lluosi yn y lluoswm nes ymlaen.

```
>>> def tebygolrwydd_amodol(plaid, colofn, gwerth, data):
...     rhifiadur = len(data[(data["Plaid.Wleidyddol"]==plaid)&(data[colofn]==gwerth)]) + 1
...     enwadur = len(data[data["Plaid.Wleidyddol"]==plaid])+len(data[colofn].unique())
...     return rhifiadur/enwadur
```

Ag isod mae gennym y ffwythiant ar gyfer cyfrifo'r tebygolrwydd o ryw blaid.

```
>>> def tebygolrwydd(plaid, data):
...     return len(data[data["Plaid.Wleidyddol"]==plaid]) / len(data)
```

Yna i greu rhif pendant ar gyfer y rhifiadur o'r tebygolrwydd o bob plaid ac i ddewis yr uchafswm.

```
>>> def rhagfynegi(data, plyg):
...     colofnau = data.columns[:-1]
...     pleidiau = data['Plaid.Wleidyddol'].unique()
...     tebygolrwyddau = {p: 1 for p in pleidiau}
...     for plaid in pleidiau:
...         for i, colofn in enumerate(colofnau):
...             gwerth = plyg[i]
...             tebygolrwyddau[plaid] *= tebygolrwydd_amodol(plaid, colofn, gwerth, data)
...         tebygolrwyddau[plaid] *= tebygolrwydd(plaid, data)
...     return max(tebygolrwyddau.keys(), key=lambda x: tebygolrwyddau[x])
```

Dyma'r plyg fyddem yn defnyddio i drio rhagfynegi ei blaid drwy ddosbarthu.

```
>>> plyg = ["Lloegr", "Benyw", "Na", "Ty Rhes", "Na"]
```

Gawn ni allbwn o blaid lafur o redeg y ffwythiant `rhagfynegi` ar ein plyg.

```
>>> rhagfynegi(data, plyg)
'Plaid Lafur'
```

Nawr wnawn drio plyg wahanol i'r algorithm drio rhagfynegi ei blaid.

```
>>> plyg = ["Lloegr", "Gwryw", "Yndw", "Ty Rhes", "Yndw"]
```

```
>>> rhagfynegi(data, plyg)
'Plaid Geidwadol'
```

Gwelwn drwy ddefnyddio'r algorithm ar blyg gwahanol cafodd ei ddosbarthu i Blaid Geidwadol.

Pennod 5

Termau

Cymraeg	***Saesneg***
Adnabod lleferydd	Speech recognition
Algorithm genetig	Genetic algorithm
Anniffiniegid	Undefined
Arwahanol	Discrete
Atchweliad	Regression
Atchweliad logistaidd	Logistic regression
Clwstwr	Cluster
Clystyru	Clustering
Clystyru hierarchaidd	Hierarchical clustering
Clystyru k-cymedr	K-means clustering
Coed penderfynu	Decision trees
Craidd	Centroid
Cyfradd llwyddiant	Success rate
Deallusrwydd artiffisial	Artificial intelligence
Deuaidd	Binary
Di-dor	Continuous
Dosbarthiad	Classification
Dosbarthiad naïf Bayes	Naive Bayes classification
Dosraniad	Distribution
Dull penelin	Elbow method
Dysgu atgyfnerthol	Reinforcement learning
Dysgu dan oruchwyliaeth	Supervised learning
Dysgu dan oruchwyliaeth rannol	Semi-supervised learning
Dysgu heb oruchwyliaeth	Unsupervised learning
Dysgu peirianyddol	Machine learning
K cymydog agosaf	K nearest neighbours
Lleihad dimensiwn	Dimension reduction
Model Markov cudd	Hidden Markov model
Ôl debygolrwydd	Posterior probability
Priodoledd	Attribute
Prosesau penderfynu Markov	Markov decision processes
Prosesu iaith naturiol	Natural language processing
Rhagdebygolrwydd	Prior probability
Rheolaeth fersiwn	Version control
Set profi	Test set
Set hyfforddi	Training set
Swm lleiaf sgwariau wedi eu hail-bwyso	Iterative weighted least squares
Tra-baramedr	Hyperparameter
Trosglwyddo cyflwyr	State transition

Taflen 5.1: Tabl o termau sydd wedi'i chynnwys yn y traethawd yma.

Llyfryddiaeth

[1] 2018 arolwg kaggle ml and ds. https://www.kaggle.com/sudhirnl7/data-science-survey-2018. Accessed: 2020-03-24.

[2] Addysg cymraeg yn prydain. https://www.mercator-research.eu/fileadmin/mercator/documents/regional_dossiers/welsh_in_the_uk_2nd.pdf. Accessed: 2020-03-27.

[3] Adnoddau addysg uwch ar coursera. https://www.coursera.org/. Accessed: 2020-03-27.

[4] Adnoddau code club. https://projects.raspberrypi.org/cy-GB/codeclub. Accessed: 2020-03-14.

[5] Adnoddau mathemateg ar gyfer ysgolion. https://www.mathemateg.com/. Accessed: 2020-03-14.

[6] Adnoddau technocamps. https://www.technocamps.com/cy/resources. Accessed: 2020-03-14.

[7] Adnoddau technolegau iaith. http://techiaith.cymru/yr-adnoddau/llawlyfr-technolegau-iaith/. Accessed: 2020-03-14.

[8] Amledd geiriau i categoreiddio geiriau. https://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn-note07-2up.pdf. Accessed: 2020-04-15.

[9] App botio. https://apps.apple.com/us/app/botio/id1296278646?ls=1. Accessed: 2020-03-14.

[10] Canran o pull requests ar github yn wahanol ieithoedd rhaglennu. https://madnight.github.io/githut/#/pull_requests/2019/4. Accessed: 2020-03-26.

[11] Cyfrifiadureg ar gyfer mathemateg. https://vknight.org/cfm/cy/. Accessed: 2020-03-14.

[12] Cymwysiadau wahanol dysgu peirianyddol. https://www.edureka.co/blog/machine-learning-applications/. Accessed: 2020-03-30.

[13] Gwybodaeth ar dosbarthiad naif bayes. https://www.edureka.co/blog/machine-learning-applications/. Accessed: 2020-04-08.

[14] Gwybodaeth ar dull atchweliad logistaidd yn python. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Accessed: 2020-04-15.

[15] Gwybodaeth ar dull glm yn r. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/glm. Accessed: 2020-04-15.

[16] Gwyddor data yn gad. https://www.gov.uk/government/news/special-feature-data-science-at-gad. Accessed: 2020-04-15.

[17] Sgiliau ymchwil cyfrifiadurol. https://sgiliauymchwilcyfrifiadurol.github.io/. Accessed: 2020-03-14.

[18] Targed o miliwn o siaradwyr cymraeg erbyn 2050. https://gov.wales/sites/default/files/publications/2018-12/cymraeg-2050-welsh-language-strategy.pdf. Accessed: 2020-03-24.

[19] Technological action plan. https://gov.wales/sites/default/files/publications/2018-12/welsh-language-technology-and-digital-media-action-plan.pdf. Accessed: 2020-03-24.

[20] Tiwtorial yn sql. https://plsql-tutorial.com/cy/index.htm. Accessed: 2020-03-26.

[21] Videos ar sut i rhaglennu yn python. https://www.youtube.com/playlist?list=PLSkPgScy-DkFdCzwJW9X_B9IfTouojem7. Accessed: 2020-03-14.

[22] Ystorfeydd technolegau iaith. https://github.com/porthtechnolegauiaith. Accessed: 2020-03-14.

[23] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2014.

[24] K.Dietz; M.Gail; K.Krickeberg; B.Singer. *Logistic regression: a self-learning text*. New York : Springer, 1994.

[25] Paweł Cichosz. *Data Mining Algorithms: Explained using R*. Chichester, UK: John Wiley and Sons, Ltd, 2015.

[26] Alun Morris. Algebra llinol. https://llyfrgell.porth.ac.uk/View.aspx?id=1716~4p~QbzBunJu. Accessed: 2020-03-14.

[27] Alexander Zien Olivier Chapelle, Bernhard Schölkopf. *Introduction to Semi-Supervised Learning*. MIT Press, 2006.

[28] Garcia F Sigaud O. *Reinforcement Learning*. Scopus, 2013.

[29] 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.