521160P Johdatus Tekoälyyn

Harjoitus #2

Regressio

Kevät 2019

Regressio on tilastollinen menetelmä, joka arvioi riippuvuutta tulomuuttujan X ja lähtömuuttujan Y välillä. Yksinkertaisin regressio-ongelma käsittelee yhtä selitettävää (tai riippuvaa) muuttujaa Y, joka riippuu ainoastaan selittävästä (tai riippumattomasta) muuttujasta X. Tehtävänä on löytää tälle tilastolliselle ongelmalle malli, joka sopii parhaiten olemassa olevaan dataan arvioiden X:n ja Y:n välistä riippuvuutta. Regressio-analyysissa peruskysymys on: Mitä matemaattista mallia tulisi käyttää ongelman ratkaisussa (suora, paraabeli, logaritmikäppyrä jne.)? Toinen peruskysymys on, että kuinka sovitamme sopivan mallin kuvaajaan?

Lineaarinen ja polynominen regressio

Yksinkertaisessa lineaarisessa regressiossa ennustetut Y:n arvot piirretään X:n funktiona, josta muodostuu malliksi suora. Matemaattisesti suoran yhtälö on kuvattu kaavassa 1:

$$y = kx + b \quad , \tag{1}$$

missä k on kulmakerroin ja b on y-akselin leikkauskohta

Joissain tapauksissa korkeamman asteen käyrä saattaa antaa paremman mallin datalle kuin pelkkä lineaarinen suora. Yksinkertaisin laajennus suoran yhtälöstä on toisen asteen polynominen käyrä, joka tunnetaan myös nimellä paraabeli. Lisäämällä malliin korkeamman asteen termejä kuten X^2 tai X^3, voidaan tätä rinnastaa uusien riippumattomien termien lisäämisenä perusmalliin. Matemaattisesti k:nnen asteen polynomista funktiota mallinnetaan kaavan 2 mukaisesti:

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k \tag{2}$$

Jos data sallii käyrän piirtämisen kaikkien datapisteiden läpi, parhaiten sopivan mallin löytämisessä ei tule olemaan ongelmaa. Valitettavasti oikeassa elämässä esimerkiksi saman ikäisillä ihmisillä ei ole tapana olla sama pituus tai paino. Tästä syystä jokaista yksittäistä datapistettä Y ei voida ennustaa täydellisesti X:n arvoista.

Ylivoimaisesti suoraviivaisin ja nopein menetelmä sovittaa malli kuvaajaan on piirtää käyrä silmämääräisesti. Vaikka kyseinen menetelmä antaa ymmärrettävän kuvan ongelmasta, saatu ratkaisu on tilastollisesti merkityksetön. Matemaattisen mallin sovittaminen datapisteisiin on kauan pohdittu ongelma regressio-analyysissa ja ongelman ratkaisemiseksi sopivat hyvin esimerkiksi varianssin minimointi menetelmä (engl. minimum-variance method or Gauss-Markov theorem) ja pienimmän neliösumman menetelmä (engl. least squares method).

Varianssin minimointi menetelmä ja pienimmän neliösumman menetelmä johtavat täsmälleen samaan lopputulokseen, kun sovitettavana mallina on lineaarinen suora. Varianssin minimointi menetelmä muistuttaa enemmän klassista tilastollista menetelmää ja sen avulla pystymme määrittämään sovitetun

suoran käyttämällä otoskeskiarvoja. Kulmakerroin $\hat{k}$ ja y-akselin leikkauskohta $\hat{b}$ voidaan laskea kaavojen 3 ja 4 avulla.

$$\hat{k} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} \tag{3}$$

$$\hat{b} = \bar{Y} - \hat{k}\bar{X} \quad , \tag{4}$$

missä $\bar{X}$ on otoskeskiarvo X:n arvoille ja $\bar{Y}$ on otoskeskiarvo Y:n arvoille.

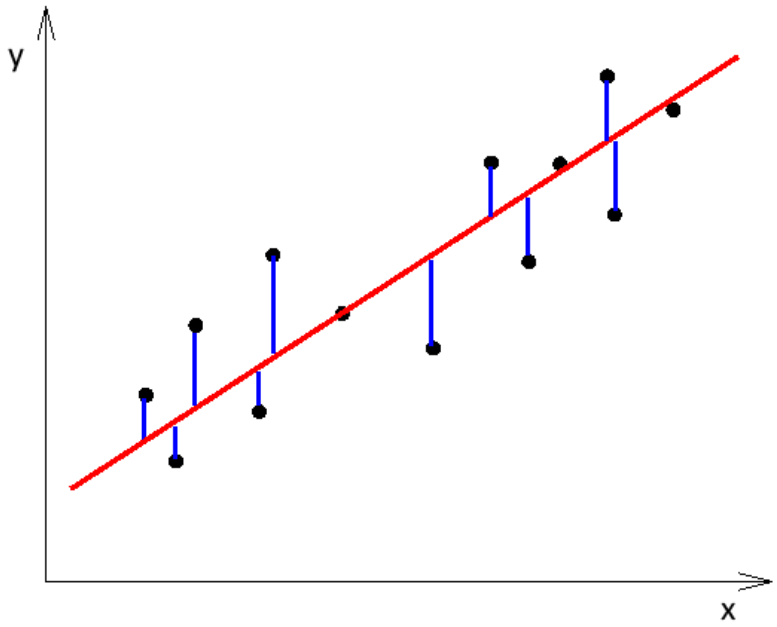
Polynomista mallia ei kuitenkaan voida sovittaa varianssin minimointi menetelmällä ja tällöin on käytettävä pienimmän neliösumman menetelmää. Pienimmän neliösumman menetelmä käyttää L2-normalisointia, joka yrittää löytää parhaiten sopivan käyrän minimoimalla pystysuuntaisten viiva segmenttien neliöiden summan. Toinen yleisesti käytetty normalisointitapa on L1-normalisointi, joka puolestaan yrittää minimoida pystysuuntaisten viiva segmenttien itseisarvojen summan. Kaksiulotteisessa tapauksessa L1-normi (manhattan-normi) ja L2 normi (euklidinen-normi) on esitetty kaavoissa 5 ja 6:

$$L1_{norm} = \sum_{i=1}^{n} |Y_i - f(X_i)| \tag{5}$$

$$L2_{norm} = \sum_{i=1}^{n} (Y_i - f(X_i))^2 \quad , \tag{6}$$

missä Y_i viittaa näytteiden y-koordinaatin arvoihin ja $f(X_i)$ viittaa sovitetun yhtälön avulla ennustettuihin y:n arvoihin

L2-normalisointi laskee neliöidyn virheen, kun taas L1-normalisointi laskee itseisarvoistetun virheen. Mitä pienempi havaittujen näytteiden etäisyyksien neliöiden summa (tai itseisarvojen summa) on, sitä lähempänä sovitettu käyrä on datapisteitä. Kuvassa 1 on esitetty pienimmän neliösumman menetelmän toimintaperiaate, kun mallina on lineaarinen suora. Kuvaajan mustat pisteet ovat datapisteitä, siniset viivat kuvaavat pystysuuntaisia segmenttejä eli virhevektoreita ja punainen käyrä kuvaa sovitettua suoraa.



Kuva 1. Pienimmän neliösumman menetelmällä sovitettu suora minimoi neliöityjen virhevektorien summan.

Matemaattisesti sovitetun käyrän kertoimet pienimmän neliösumman menetelmällä voidaan laskea matriisialgebralla kaavan 7 mukaisesti:

$$Y = XA \Leftrightarrow A = (X^T X)^{-1} X^T Y \quad , \tag{7}$$

missä X viittaa yhtälön x:n saamiin arvoihin matriisissa, Y viittaa yhtälön y:n saamiin arvoihin matriisissa ja A viittaa kertoimiin [a_0,a_1,a_2,…,a_k].

Sovitetun suoran suorituskyvyn mittaamiseen käytetään usein korrelaatiokerrointa r (tai korrelaatiokertoimen neliötä r^2) tilastollisena työkaluna, joka kertoo kuinka hyvin datapisteet ja malli riippuvat toisistaan. Kun korrelaatiokerroin on 1, niin datassa muuttujien välillä on täydellinen riippuvuus ja kun korrelaatiokerroin on -1, niin muuttujien välillä on täydellinen negatiivinen riippuvuus. Kun taas korrelaatiokerroin on 0, niin muuttujat eivät ole ollenkaan riippuvaisia toisistaan. Korrelaatiokerroin saadaan laskettua sovitetulle lineaariselle regressiomallille kaavalla 8 ja yleisesti ottaen mille tahansa sovitetulle mallille kaavalla 9.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{8}$$

$$r^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(Y_i - f(X_i))^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \quad , \tag{9}$$

missä SSE (engl. the sum of squared errors) on neliöityjen virhevektorien yhteenlaskettu summa ja SST (engl. the total sum of squares) kuvastaa selitettävän muuttujan vaihtelua sen keskiarvon ympärillä.

Esimerkiksi käytetään mallina toisen asteen yhtälöä, joka on muotoa $y = a_0 + a_1 x + a_2 x^2$ ja sovitetaan malli pisteille (2,5), (-2,5), (0,0) ja (0,2). Tällöin matriisit X,Y ja A saavat muodon:

$$X = \begin{bmatrix} 1 & 2 & 4 \\ 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \qquad Y = \begin{bmatrix} 5 \\ 5 \\ 0 \\ 2 \end{bmatrix}$$

Käyttämällä kaavaa 7 saadaan laskettua

$$A = (X^T X)^{-1} X^T Y = \begin{bmatrix} 0.5 & 0 & -0.125 \\ 0 & 0.125 & 0 \\ -0.125 & 0 & 0.0625 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & -2 & 0 & 0 \\ 4 & 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

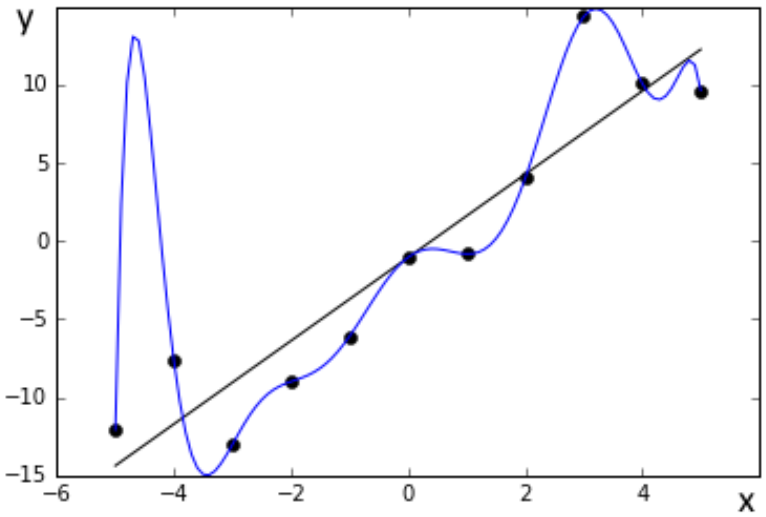
Eli kerroinmatriisiksi A saadaan ratkaistua $A = [1\ 0\ 1]^T$, joten toisen asteen yhtälöksi tulee $y = 1 + x^2$.

Lasketaan vielä sovitetun mallin korrelaatiokerroin käyttämällä kaavaa 9

$$r^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - f(X_i)\right)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} = 1 - \frac{(5-5)^2 + (5-5)^2 + (0-1)^2 + (2-1)^2}{(5-3)^2 + (5-3)^2 + (0-3)^2 + (2-3)^2} = 0.889$$

Ottamalla luvusta 0.889 neliöjuuri, korrelaatiokertoimeksi saadaan r = 0.943

Ylioppiminen on yleinen ongelma ohjatussa oppimisessa, johon myös regressioanalyysi kuuluu. Jos malliksi valitaan liian korkea-asteinen polynominen yhtälö, sovitettu käyrä alkaa mallintamaan datassa esiintyvää kohinaa eli värähtelyä ja tapahtuu ylioppiminen. Tämän kaltainen tilanne on esitetty kuvassa 2. Vaikka kyseinen malli kulkee kaikkien datapisteiden kautta ja korrelaatiokerroin on 1, ei se siltikään ole paras mahdollinen malli mallintamaan muuttujien välistä riippuvuutta. Selvästi parempi vaihtoehto kuvan 2 sovitettavaksi malliksi on lineaarinen suora. Sen sijaan, jos kompleksiselle datalle valitaan liian yksinkertainen malli, se ei onnistu jäljittelemään datan rakennetta riittävän tarkasti ja tapahtuu alioppiminen.



Kuva 2. Liian monimutkaisen mallin valinnasta aiheutuu ylioppiminen.

Logistinen regressio

Logistinen regressio on regressioanalyysin erikoistapaus, joka pyrkii ennustamaan, millä todennäköisyydellä määritelty tapahtuma tulee tapahtumaan. Se on epälineaarinen regressiomenetelmä, jota käytetään usein nimestään huolimatta näytteiden luokitteluun. Tällöin datassa näytteillä tulee olla myös tieto luokista, joihin ne kuuluvat. Mikäli luokiteltavia luokkia on vain kaksi, käytetään binääristä logistista regressiota, jonka logistinen funktio sigmoid on esitetty kaavassa 10.

$$P(Y = 1) = \frac{1}{1+e^{-t}} \quad , \tag{10}$$

missä $t = a_0 + a_1x + \cdots + a_kx^k$ *ja x on selittävä muuttuja*

Kahden luokan tapauksessa tapahtuman P(Y=1) vastatapahtuma saadaan yksinkertaisesti laskettua P(Y=0) = 1 – P(Y=1).

Mikäli luokkia on enemmän kuin kaksi, käytetään multinomiaalista logistista regressiota, jonka logistinen funktio softmax on esitetty kaavassa 11.

$$P(Y = j) = \frac{e^{t_j}}{\sum_{n=1}^{N} e^{t_n}} \quad , \tag{11}$$

missä $t_j = a_{0j} + a_{1j}x + \cdots + a_{kj}x^k$ *, x on selittävä muuttuja ja N on luokkien lukumäärä*

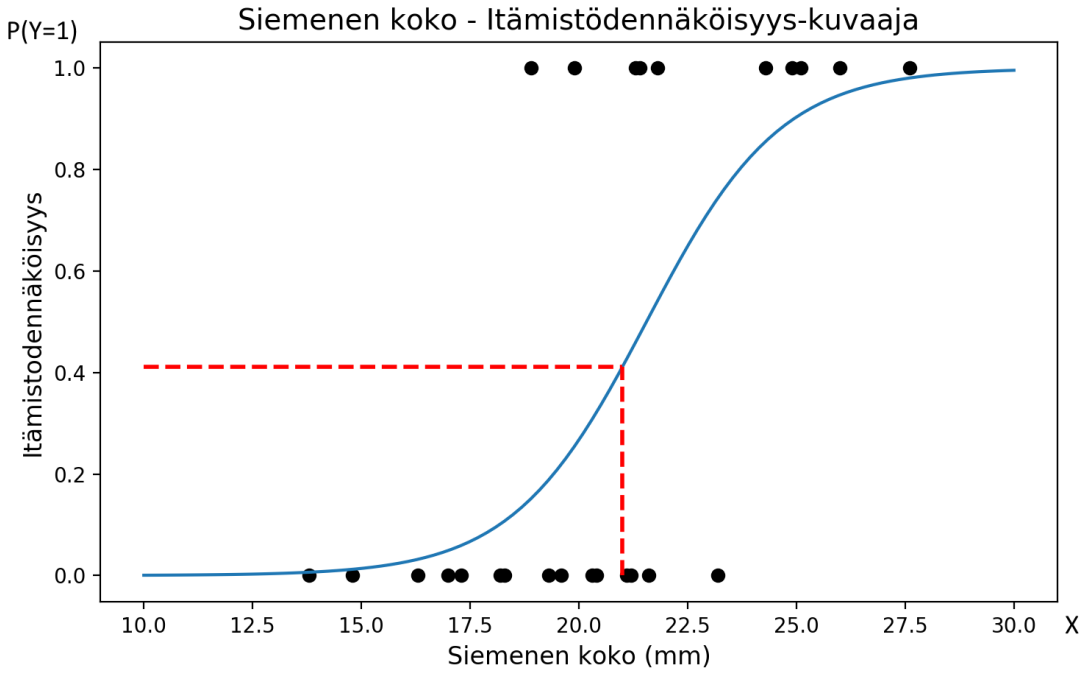
Tutkitaan esimerkkinä, miten erään kasvin siemenen koko vaikuttaa sen itämiseen. Kyseessä on kahden luokan tapaus, sillä siemen joko itää tai jää itämättä. Taulukossa 1 on esitetty 25 siemenen näytteestä koostuva data, jossa 10 siemenistä iti ja 15 ei itänyt. [1]

Taulukko 1. Datajoukko, joka sisältää siemenen koon sekä tiedon siemenen itämisestä.

Siemenen koko (mm)	13,8	14,8	16,3	17,0	17,3	18,2	18,3	18,9	19,3	19,6	19,9
Siemen iti (k/e)?	e	e	e	e	e	e	e	k	e	e	k

20,3	20,4	21,1	21,2	21,3	21,4	21,6	21,8	23,2	24,3	24,9	25,1	26,0	27,6
e	e	e	e	k	k	e	k	e	k	k	k	k	k

Optimoimalla iteratiivisesti löydetään logistiseksi funktioksi $P(Y=1)=\frac{1}{1+e^{-0.65x+14}}$. Nyt voidaan arvioida esimerkiksi, millä itämistodennäköisyydellä 21,0mm kokoinen siemen tulee itämään. Tätä tilannetta on tarkasteltu graafisesti kuvassa 3, johon on piirretty optimoitu logistinen funktio sekä taulukon 1 näytteet. Kuvaajassa itämistodennäköisyyden arvo 1.0 tarkoittaa, että siemen iti ja 0.0, että siemen ei itänyt. Kuvaajasta nähdään, että 21,0mm kokoinen siemen tulee itämään noin 40 % todennäköisyydellä.



Kuva 3. Siemenin koko – itämistodennäköisyys-kuvaaja.

Tehdessäsi harjoitusta omalla tietokoneella, asenna numpy, scikit-learn ja matplotlib python-kirjastot tietokoneellesi seuraavasti:

Mene komentoriville ja aja seuraava komento asentaaksesi tarvittavat kirjastot

```
pip install numpy scikit-learn matplotlib
```

Jos komento ei toimi, voit asentaa python-kirjastot myös seuraavien linkkien avulla:

Numpy: https://docs.scipy.org/doc/numpy-1.14.0/user/install.html

Scikit-learn: http://scikit-learn.org/stable/install.html

Matplotlib: https://matplotlib.org/users/installing.html

Tehtävä 1 (2.0p)

Tehtäväsi on luoda lineaarinen regressiomalli yhdelle seuraavista datajoukoista:

- **data1_ects_accumulation.txt**: Kuvitteellinen datajoukko 50 opiskelijan opintopistekertymistä. Data kuvaa kuinka paljon opintopisteitä on kertynyt joukolle opiskelijoita eri ajan hetkillä heidän opintojensa aikana. Vaaka-akselilla on opiskeluaika vuosina ja pystyakselilla opintopistekertymä.
- **data1_life_expectancy_finland.txt**: Eliniänodote Suomessa vuosina 1960-2015 vastasyntyneille. Vaaka-akselilla on vuosiluku ja pystyakselilla eliniänodote vuosina.
- **data1_population_growth_finland.txt**: Väkiluku Suomessa vuosina 1960-2016. Vaaka-akselilla on vuosiluku ja pystyakselilla on väkiluku.
- **data1_sea_level.txt**: Vedenpinnantason kasvaminen Venetsiassa vuosina 1909-2000 [2]. Vedenpinnantason nollatasoksi on määritetty mittaushistorian alkuajankohtana noin 7000 mm maapallon keskimääräistä vedenpinnantasoa pienempi arvo negatiivisten arvojen välttämiseksi. Vaaka-akselilla on vuosiluku ja pystyakselilla vedenpinnankorkeus nollatasosta millimetreissä.

Datajoukot on annettu tekstitiedostoina, missä x-akselin ja y-akselin pisteet ovat omilla riveillään eroteltuna toisistaan pilkulla. Muokkaa tiedostoa nimeltä **linearregression.py**.

Koneoppimisessa opetusjoukkoa käytetään mallin opettamiseen ja testijoukkoa käytetään opetetun mallin suorituskyvyn arvioimiseen. Tässä tapauksessa opettaminen tarkoittaa sopivien suoran yhtälön kertoimien selvittämistä opetusjoukon näytteiden perusteella. Aluksi alkuperäinen datajoukko jaetaan sattumanvaraisesti opetusjoukoksi ja testijoukoksi. Kyseinen jakaminen on helppo suorittaa sklearn-kirjaston **train_test_split()** funktiolla. Funktio ottaa parametreina x-akselin pisteet, y-akselin pisteet sekä testijoukon absoluuttisen koon. 10 prosenttia testijoukon kooksi on riittävä.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1)
```

Laskeaksesi lineaarisen regression mallin kertoimet ($\hat{k}$ ja $\hat{b}$), käytä numpy-kirjaston funktiota **np.polyfit()**. Funktio ottaa parametreina opetusjoukon X:n arvoille, opetusjoukon Y:n arvoille ja monennenko asteen polynominen funktio on kyseessä. Lineaarinen malli käyttää ensimmäisen asteen funktiota. Kun sovitetut kertoimet on löydetty mallille, voidaan muodostaa funktio numpy-kirjaston funktiolla **np.poly1d()**. Alla on esitetty esimerkki koodirivit, miten polynominen funktio sovitetaan. Siinä X_train viittaa opetusjoukon X:n arvoihin, Y_train viittaa opetusjoukon Y:n arvoihin ja numero 30 viittaa 30:n asteen polynomiseen funktioon.

```
coeffs = np.polyfit(X_train, Y_train, 30)

function = np.poly1d(coeffs)
```

Mallin piirtämistä varten on myös määrättävä x-akselille ja y-akselille vaihteluväli, jolta malli piirretään kuvaajaan. Mallin vaihteluväli x-akselin suunnassa vastaa opetusjoukon vaihteluväliä x-akselin suunnassa, jolloin *x_line* saa arvoja väliltä *[min(X_train), max(X_train)]*. Muuttujan *y_line* arvot saadaan laskettua aiemmin luodusta lineaarisesta funktiosta.

```
x_line = np.linspace(min(X_train), max(X_train))
```

```
y_line = function(x_line)
```

Tämän jälkeen piirretään opetusjoukon pisteet ja sovitettu suora samaan kuvaajaan funktioiden **plt.scatter()** ja **plt.plot()** avulla. Kuvaajaan voi myös lisätä otsikon sekä nimetä akselit komennoilla **plt.title(**”The title”**)**, **plt.xlabel**(”x-axis”**)** ja **plt.ylabel(**”y-axis”**)**.

Kun saat piirrettyä datapisteet ja suoran kuvaajaan valitsemallesi datajoukolle sekä nimettyä kuvaajan ja akselit, ota kommenttimerkki pois funktion performance() edestä ja komentoriville tulostuu suorituskyky sovitetulle suoralle. Kyseinen funktio laskee absoluuttisen virheen, keskiarvoistetun neliövirheen (MSE), explained variance score:n, R2 score:n (korrelaatiokertoimen neliö) sekä suoran yhtälön kertoimet.

Arvioi lisäksi toteuttamasi mallin perusteella printtaamalla vastaus komentoriville seuraavaan kysymykseen käyttämästäsi datajoukosta riippuen:

- ECTS: Kuinka monta opintopistettä oli kerätty 2 vuoden opiskelun jälkeen?
- Life expectancy: Mikä on suomalaisten eliniänodote vuonna 2030?
- Population growth: Mikä on Suomen väkiluku vuonna 2035?
- Sea level: Mikä on vedenpinnankorkeus nollatasosta vuonna 2025?

Huom. printattu arvo tulee laskea muuttujan *function* avulla.

Tehtävä 2 (1.5p)

Tehtäväsi on muodostaa polynominen regressiomalli yhdelle seuraavista datajoukoista:

- **data2_exchange_rate.txt**: Euron (EUR) ja Ruotsin kruunun (SEK) välinen valuuttakurssi 90 päivän aikajaksolta kesällä 2017. Vaaka-akselilla on aika päivinä aloitusajanhetkestä ja pystyakselilla on valuuttakurssi.
- **data2_weather_oulu.txt**: Sunnuntaina 18.3.2018 Oulun vihreäsaaren sääaseman mittaamat lämpötilat. Vaaka-akselilla on kellonaika (00:00-24:00) ja pystyakselilla kyseisellä ajanhetkellä mitattu lämpötila celsiusasteina.

Datajoukot on annettu tekstitiedostoina, missä x-akselin ja y-akselin pisteet ovat omilla riveillään eroteltuna toisistaan pilkulla. Muokkaa tiedostoa nimeltä **polynomialregression.py**.

Piirrä kuvaajaan polynominen regressiomalli niin, että ylioppimista tai alioppimista ei pääse tapahtumaan, jolloin kyseessä on ns. parhaiten sovitettu malli. Tämä tapahtuu valitsemalla polynomiselle mallille sopiva asteluku. Poistamalla kommenttimerkin funktion performance() edestä, voit arvioida polynomisen mallin suorituskykyä.

Seuraavan linkin esimerkeistä voi olla hyötyä tehtävien 1 ja 2 tekemiseen: https://docs.scipy.org/doc/numpy-1.14.0/reference/generated/numpy.polyfit.html

Tehtävä 3 (1.5p)

Tehtäväsi on muodostaa logistisen regression luokittelija yhdelle seuraavista datajoukoista:

- **data3_wine.csv**: Kolmesta italialaisesta viinilajikkeesta koostuva datajoukko (luokat numeroitu 1-3) [3]. Datajoukko sisältää yhteensä 178 näytettä. Piirteinä datajoukolle käytetään 13 viinin ominaisuutta, jotka on esitetty seuraavassa listassa:

wine_features = [alkoholipitoisuus, omenahapon määrä, tuhkan määrä, tuhkan alkaliteetti, magnesiumin määrä, fenolit, flavonoidit, ei-flavonoidiset fenolit, proantosyaniidit, viinin värin intensiteetti, viinin värisävy, laimennetun viinin OD280/OD315, proliinin määrä]

- **data3_seed.csv**: Kolmen vehnälajin (Kama, Rosa ja Canadian) siementen datajoukko [4]. Kutakin luokkaa kohden on 70 näytettä eli yhteensä näytteitä on 210 kappaletta (luokat numeroitu 1-3). Näytteet on alun perin laser-skannattu, jonka jälkeen kuvista on analysoitu näytteiden piirteet. Piirteinä käytetään seuraavia 7 siemenen muotoa kuvaavaa ominaisuutta:

seed_features = [siemenen pinta-ala, siemenen kehän ympärysmitta, ytimen pituus, ytimen leveys, tiiviys, epäsymmetrisyyskerroin, ytimen keskiuran leveys]

- **data3_leaf.csv**: Datajoukko koostuu 30 eri kasvilajin lehden näytteestä (luokat numeroitu 1-30) [5]. Datajoukossa on yhteensä 340 näytettä. Näytteet koostuvat seuraavista 14:stä lehden muotoa ja tekstuuria kuvaavista piirteistä:

leaf_features = [epäkeskisyys, kuvasuhde, venymä, kiinteys, stokastinen konveksisuus, isoperimetrinen vakio, maksimi tunkeumasyvyys, liuskamaisuus, keskimääräinen intensiteetti, keskimääräinen kontrasti, tasaisuus, 3. momentti, yhdenmukaisuus, entropia]

- **data3_glass.csv**: 7 erilaista lasityyppiä (valettu talon ikkunalasi, tavallinen talon ikkunalasi, valettu auton ikkunalasi, tavallinen auton ikkunalasi, säilytysastian lasi, pöytäkalustolasi ja taskulampun lasi) sisältävä datajoukko (luokat numeroitu 1-7) [6]. Näytteitä on yhteensä 214 kappaletta ja ne sisältävät seuraavat 9 lasin ominaisuuksia kuvaavaa piirrettä:

glass_features = [natriumin määrä, magnesiumin määrä, alumiinin määrä, piin määrä, kaliumin määrä, kalsiumin määrä, bariumin määrä, raudan määrä, lasin taitekerroin]

Datajoukot on annettu excel-tiedostoina, jossa näytteet ovat riveillä. Tiedoston ensimmäiseltä sarakkeelta löytyy näytteiden luokkatieto ja seuraavilta sarakkeilta näytteiden piirteet. Muokkaa tiedostoa nimeltä **logisticregression.py**.

Luodaan aluksi logistisen regression luokittelija käyttämällä sklearn-kirjaston valmista funktiota ja annetaan sille tieto datasta ja luokista opettamista varten.

```
lr=linear_model.LogisticRegressionCV(solver='liblinear',cv=5,multi_class='auto')
lr.fit(data, labels)
```

Nyt kun luokittelija on opetettu, voidaan ennustaa, millä todennäköisyydellä kuhunkin luokkaan satunnainen testinäyte kuuluu. Käytä testinäytteenä yhtä seuraavista piirrevektorilistoista datajoukosta riippuen:

```
wine_test_sample = [12.3, 1.5, 2, 15, 95, 2, 1.9, 0.4, 1.3, 3, 1, 2.5, 800]
seed_test_sample = [15, 15, 0.86, 5.1, 3.3, 4, 5]
leaf_test_sample = [0.55, 1.0, 0.55, 0.75, 0.78, 0.28, 0.08, 1.2, 0.1, 0.15, 0.02, 0.007, 0.0008, 2.5]
```

```
glass_test_sample = [1.516, 13, 3.5, 1.5, 72.9, 0.6, 8, 0, 0.1]
```

Testinäytteen syöttämisen jälkeen saamme luokittelutodennäköisyydet eri luokille luokittelijan perusteella komennolla:

```
predict_probs = lr.predict_proba([test_sample])[0]
```

Esitetään lopuksi luokittelutodennäköisyydet eri luokille pylväsdiagrammissa käyttämällä valmiiksi luotua drawBarDiagram(labels, predict_probs) funktiota, joka ottaa argumentteina datajoukon luokkatiedon sekä ennustetut luokittelutodennäköisyydet.

Lähteet

[1] Vanhoenacker D. Logistic Regression. URL:http://hem.bredband.net/didrik71/recostat/logreg_e.htm. Accessed 20.8.2018.

[2] Pirazzoli P. & Tomasin A. (2002) Recent evolution of surge-related events in the northern Adriatic area. Journal of Coastal Research 18(3): 537-554. (URL for data: http://www.psmsl.org/data/obtaining/stations/168.php Accessed 20.8.2018).

[3] Aeberhard S., Coomans D., & De Vel O. (1992) Comparison of classifiers in high dimensional settings. James Cook Univ., North Queensland, Australia. (URL for data: https://archive.ics.uci.edu/ml/datasets/wine. Accessed 20.8.2018).

[4] Charytanowicz M., Niewczas J., Kulczycki P., Kowalski P., Łukasik S. & Żak S. (2010) Complete gradient clustering algorithm for features analysis of x-ray images. In Information technologies in biomedicine (pp. 15-24). Springer, Berlin. (URL for data: https://archive.ics.uci.edu/ml/datasets/seeds. Accessed 20.8.2018).

[5] Silva P. (2013) Development of a System for Automatic Plant Species Recognition. Master's thesis. University of Porto, Portugal. (URL for data: https://archive.ics.uci.edu/ml/datasets/leaf. Accessed 20.8.2018).

[6] Kontkanen P., Myllymäki P., Silander T., Tirri H. & Grünwald P. (2000) On predictive distributions and Bayesian networks. Statistics and Computing, 10(1): 39-54. (URL for data: https://archive.ics.uci.edu/ml/datasets/glass+identification. Accessed 20.8.2018).

Harjoituksen 2 oppimistavoitteet

- erottaa ohjatun oppimisen suuntausten regression ja luokittelun keskeiset eroavaisuudet
- osaa muodostaa lineaarisen ja polynomisen regressiomallin datanäytteiden pohjalta
- ymmärtää regressioanalyysin pääkäsitteet sekä yleisimmät erikoistilanteet kuten yli- ja alioppiminen
- ymmärtää pääpiirteittäin logistisen regression toimintaperiaatteen

Palauta

Palauta muokkaamasi python tiedostot (.zip tai .rar tiedostoon pakattuna) Optiman palautuslaatikkoon Harjoitus 2 **2.4.2019 klo 23:59** mennessä. Tästä harjoituksesta on mahdollisuus tienata 5 pistettä (2.0p + 1.5p + 1.5p). Voit antaa palautetta tästä harjoituksesta liittämällä pakattuun tiedostoon erillisen tekstitiedoston palautetta varten. Harjoituksia tullaan kehittämään palautteen pohjalta tuleville vuosille vastaamaan paremmin oppimistavoitteita.