

<b>2a.1)</b>	Hvorfor ønsker vi å dele dataene inn i trening-, validering- og test-sett?
<b>Svar</b>	Inndeling av dataene gir oss en mulighet til å validere modellens ytelse. Der treningssettet brukes til å trene modellen. Valideringssettet bidrar til å finjustere modellparameterne/hyperparameterne og testsettet brukes til å evaluere hvor godt generalisert modellen er og dermed hvor godt den vil fungere på ny og ukjent data. Denne oppdelingen hjelper også med å forhindre at modellen blir overtilpasset akkurat den dataen som brukes. Dersom modellen blir for tilpasset den spesifikke dataen, kan den fungere dårligere på ny og ukjent data. Denne oppdelingen hjelper også med å evaluere modellens generelle ytelse for ukjent data. Valideringssettet blir også ofte brukt til å finjustere parametre for å forbedre modellen. Samlet sett er inndeling i trenings-, validerings- og testsett en god bestep praksis for å tilpasse en modell til å fungere godt på ny og ukjent data.

<b>2a.2)</b>	Hvor stor andel av dataene er nå i hver av de tre settene? Ser de tre datasettene ut til å ha lik fordeling for de tre forklaringsvariablene og responsen?
<b>Svar</b>	Datasettet blir først delt inn i en 80/20 fordeling for "trenval" (trening og validering) og "test", deretter deles den 80% inn i 75% og 25% ("trenval" blir til "tren" og "val"). Dette vil medføre at "df_tren" blir $0.8 \cdot 0.75 = 0.6$ , "df_val" blir $0.8 \cdot 0.25 = 0.2$ og "df_test" blir 0.2.

<b>2a.3)</b>	La oss si at vi hadde valgt League 1 og 2 som treningssett, Championship som valideringssett, og Premier League som testsett. Hvorfor hadde dette vært dumt?
<b>Svar</b>	Når trenings, validering og testsett velges er det viktig at alle tre settene er mest mulig lik det modellen kommer til å møte i den virkelige verden. Siden forskjellige fotball ligaer kan ha forskjellige dynamikker, spillestiler og kvalitet er det derfor uhensiktsmessig å velge forskjellige ligaer til de forskjellige settene, siden dataen kan være forskjellig. I tillegg brukes ofte valideringssettet til å finjustere modellen, og ved å bruke en annen liga i valideringssettet enn i testsettet kan modellen bli dårligere enn den ellers ville vært. Oppsummert er det viktig at trenings-, validerings- og testsettene representerer mest mulig like forhold som modellen kommer til å møte senere slik at modellen fungerer best mulig for ny og ukjent data.

<b>2a.4)</b>	Kommenter kort på hva du ser i plottene og utskriften (maks 5 setninger).
<b>Svar</b>	Kryssplottene som har 'skudd_paa_maal_diff' som forklaringsvariabel har en tydelig separasjon mellom seier og ikke seier til hjemmelaget, dette indikerer at 'skudd_paa_maal_diff' er brukbar for å predikere utfallet av en fotballkamp. Man kan også observere at empiriske tetthetsplottet som angår 'skudd_paa_maal_diff' er forskjøvet, som da kan bety at denne forklaringsvariabelen har en betydning for resultatet. Man kan ikke se den samme oppførselen like tydelig ved de andre empiriske tetthetsplottene. Kryssplottene som inneholder variablene 'corner_diff' og 'forseelse_diff' har ingen tydelig separasjon mellom seier og ikke til hjemmelaget, dette tyder på at disse variablene fungerer dårlig på å predikere utfallet av en kamp.

--	--

**2a.5)** Hvilke(n) av de tre variablene tror du vil være god(e) til å bruke til å predikere om det blir hjemmeseier? Begrunn svaret kort (maks 3 setninger).

**Svar** Som forklart over vil 'skudd\_paa\_maal\_diff' være en god variabel for å predikere om det blir hjemmeseier. Ettersom det er tydelige skiller på oransje og blå datapunkter langs aksene som angår 'skudd\_paa\_maal\_diff', man kan også se at det er en høy korrelasjonskoeffesient mellom vinn og 'skudd\_paa\_maal\_diff' på 0,42. Variablene 'corner\_diff' og 'forseelse\_diff' derimot har ingen tydelig seperasjon mellom seier og ikke, siden de har korrelasjonskoeffesienter som er nærme null, vil de dermed fungere dårlig til å predikere utfallet av en kamp.

**2b.1** I en kamp der skudd\_paa\_maal\_diff er 2, corner\_diff er -2 og forseelse\_diff er 6, hva er ifølge modellen sannsynligheten for at hjemmelaget vinner? Vis utregninger og/eller kode, og oppgi svaret med tre desimaler.

**Svar**

Koeffisientverdier:

Intercept	-0.591661	$\beta_0$
skudd_paa_maal_diff	0.382565	$\beta_1$
corner_diff	-0.100377	$\beta_2$
forseelse_diff	0.012009	$\beta_3$

linær regresjon

$$y = -0,592 + 0,383x_1 - 0,100x_2 + 0,012x_3$$

$$p_i = \frac{e^{\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki}}}{1 + e^{\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki}}}$$

$$x_1 = 2$$

$$x_2 = -2$$

$$x_3 = 6$$

$$p_i = \frac{e^{(-0,592 + 0,383 \cdot 2 - 0,100 \cdot (-2) + 0,012 \cdot 6)}}{1 + e^{(-0,592 + 0,383 \cdot 2 - 0,100 \cdot (-2) + 0,012 \cdot 6)}}$$

$$\underline{\underline{p_i \approx 0.610}}$$

Utrekning

```

1 # Disse er de estimerte koeffisientene fra din modellen
2 beta_0 = resultat.params['Intercept'] # koeffisient for intercept fra resultat.params
3 beta_1 = resultat.params['skudd_paa_maal_diff'] # koeffisient for skudd_paa_maal_diff fra resultat.params
4 beta_2 = resultat.params['corner_diff'] # koeffisient for corner_diff fra resultat.params
5 beta_3 = resultat.params['forseelse_diff'] # koeffisient for forseelse_diff fra resultat.params
6
7 # Definer variabelverdiene for en kamp
8 skudd_paa_maal_diff = 2
9 corner_diff = -2
10 forseelse_diff = 6
11
12 # Beregn logg-odds for hjemmeseier
13 logg_odds = beta_0 + (beta_1 * skudd_paa_maal_diff) + (beta_2 * corner_diff) + (beta_3 * forseelse_diff)
14
15 # Konverter logg-odds til sannsynlighet
16 sannsynlighet_hjemmeseier = (np.exp(logg_odds)) / (1 + np.exp(logg_odds))
17
18 print(f"Sannsynligheten for hjemmeseier er:")
19 print("{:.3}".format(sannsynlighet_hjemmeseier))

```

Executed at 2023.11.17 12:36:56 in 97ms

Sannsynligheten for hjemmeseier er:  
0.61

Kode (tallet er 0.610 med 3 desimaler, men den siste nullen blir fjernet.)

## 2b.2) Hvordan kan du tolke verdien av $e^{\beta_{\text{skudd-paa-maal-diff}}}$ ?

Svar

```

1 # her kan du skrive kode for å regne ut exp(beta)
2 beta_skudd_paa_maal_diff = np.exp(0.382565)
3 print("exp(beta_skudd_paa_maal_diff): ", beta_skudd_paa_maal_diff)
4

```

Executed at 2023.11.16 11:48:48 in less than 1ms

('exp(beta\_skudd\_paa\_maal\_diff): ', 1.466040163871275)

$e^{\beta_{\text{skudd-paa-maal-diff}}}$  sier noe om hvilken påvirkning skudd\_paa\_maal\_diff har på oddsen for hjemmeseier.  $e^{\beta_{\text{skudd-paa-maal-diff}}}$  er i dette tilfellet 1,467. Dette tallet ganges med skudd\_paa\_maal\_diff i en kamp for å finne påvirkningen for hjemmeseier. Her er  $e^{\beta_{\text{skudd-paa-maal-diff}}}$  større enn 1, det vil si at jo større skudd\_paa\_maal\_diff, jo høyere odds for hjemmeseier. Dersom  $e^{\beta_{\text{skudd-paa-maal-diff}}}$  hadde vært mindre enn 1 så hadde skudd\_paa\_maal\_diff hatt en negativ effekt på oddsen for hjemmeseier. Og dersom  $e^{\beta_{\text{skudd-paa-maal-diff}}}$  hadde vært lik 0, hadde ikke skudd\_paa\_maal\_diff hatt noe å si for oddsen for hjemmeseier.

## 2b.3) Hva angir feilraten til modellen? Hvilket datasett er feilraten regnet ut fra? Er du fornøyd med verdien til feilraten?

Svar

Feilraten blir angitt av andel predikerte verdier som ikke stemmer overens med det faktiske datasettet. Datasettet som brukes er valideringssettet. Feilraten er i dette tilfellet 0.285 eller 28.5%, dette betyr at litt over en fjerdedel av prediksjonene var feil. En større kontekst trengs nok for å fastslå om dette er en feilrate man kan være misfornøyd ut med, men subjektivt ser ikke dette ut som en veldig nøyaktig.

## 2b.4) Diskuter kort hvordan koeffisientene ( $\beta$ – ene) og feilraten endrer seg når forseelse\_diff tas ut av modellen (maks 3 setninger).

Svar

Fjernes 'forseelse\_diff' variabelen fra modellen endres  $\beta$ -verdiene minimalt. Samme konklusjon trekkes ved feilraten, feilraten med 'forseelse\_diff' i modellen er 0.285 eller 28.5%, uten denne variabelen blir feilraten 0.283 eller 28.3%, som igjen er en veldig liten forskjell.

**2b.5** Med den nye modellen: I en kamp der `skudd_paa_maal_diff = 2`, `corner_diff = -2` og `forseelse_diff = 6`, hva er sannsynligheten for at hjemmelaget vinner ifølge den nye modellen? Oppgi svaret med tre desimaler.

**Svar** Svaret er 0.592. Altså litt mindre sannsynlighet enn med modellen som inkluderte `forseelse_diff`. I den nye modellen har ikke `forseelse_diff` noen invirkning på oddsen for hjemmeseier.

```
1 # Disse er de estimerte koeffisientene fra din modellen
2 beta_0 = resultat.params['Intercept'] # koeffisient for intercept fra resultat.params
3 beta_1 = resultat.params['skudd_paa_maal_diff'] # koeffisient for skudd_paa_maal_diff fra resultat.params
4 beta_2 = resultat.params['corner_diff'] # koeffisient for corner_diff fra resultat.params
5 #beta_3 = resultat.params['forseelse_diff'] # koeffisient for forseelse_diff fra resultat.params
6
7 # Definer variabelverdiene for en kamp
8 skudd_paa_maal_diff = 2
9 corner_diff = -2
10 forseelse_diff = 6
11
12 # Beregn logg-odds for hjemmeseier
13 logg_odds = beta_0 + (beta_1 * skudd_paa_maal_diff) + (beta_2 * corner_diff) #+ (beta_3 * forseelse_diff)
14
15 # Konverter logg-odds til sannsynlighet
16 sannsynlighet_hjemmeseier = (np.exp(logg_odds)) / (1 + np.exp(logg_odds))
17
18 print(f"Sannsynligheten for hjemmeseier er:")
19 print("{:.3}".format(sannsynlighet_hjemmeseier))
20
21 Sannsynligheten for hjemmeseier er:
22 0.592
```

(I oppgaven står det at `'forseelse_diff' = 6`, men denne infoen er da overflødig siden vi ikke har variabelen i modellen).

**2b.6)** Hvis du skal finne en så god som mulig klassifikasjonsmodell med logistisk regresjon, vil du velge modellen med eller uten `forseelse_diff` som kovariat? Begrunn kort svaret (maks 3 setninger).

**Svar** Den beste modellen er modellen uten `forseelse_diff`. Feilraten for modellen uten `forseelse_diff` er 28,3%, mens modellen med `forseelse_diff` er på 28,5%. Det er en liten forskjell i feilrate, men den beste modellen blir derfor modellen uten `forseelse_diff`, siden den modellen har mindre feil.

**2c.1)** Påstand: kNN kan bare brukes når vi har maksimalt to forklaringsvariabler. Fleip eller fakta?

**Svar** Fleip. KNN kan brukes når vi har flere forklaringsvariabler.

**2c.2)** Hvilken verdi av  $k$  vil du velge?

**Svar** I vårt tilfelle er verdien av  $k$  som er mest fornuftig å bruke  $k = 139$ , dette er siden ved denne verdien gir den laveste feilraten. Feilraten i vårt tilfelle er da 0,278 (rundet til 3-desimaler).

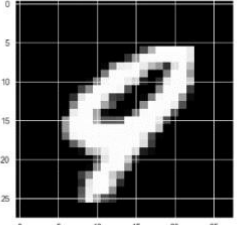
```
Minste feilrate: 0.27764127764127766
K verdi for minste feilrate: 139.0
```

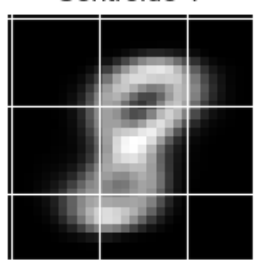
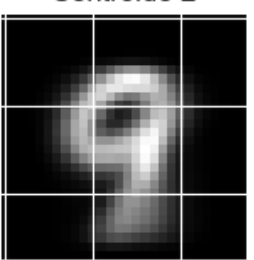
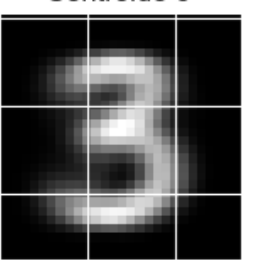
(Vi har kjørt kode som går gjennom lista av feilrater og leter etter den minste verdien, og deretter printet den korresponderende  $k$ -verdien)

<b>2d.1)</b>	Gjør logistisk regresjon eller $k$ -nærmeste-nabo-klassifikasjon det best på fotballkampdataene?
<b>Svar</b>	k-nærmeste-nabo får en feilrate på 0,316 på test-settet, mens logistisk regresjon får en feilrate på 0,328 på test-settet. Dette tilsier at k-nærmeste-nabo er den beste modellen siden den har minst feilrate.

<b>2d.2)</b>	Drøft klassegrensene (plottet under) for de to beste modellene (én logistisk regresjon og én kNN). Hva forteller klassegrensene deg om problemet? Skriv maksimalt 3 setninger.
<b>Svar</b>	Plottet viser liten forskjell i klassegrense mellom kNN og logistisk i områdene der det er mye data, dette gjenspeiler den lille forskjellen i feilrate mellom de to. I området der det er lite data (området med mindre enn -10 i corner-difference) er det større forskjell mellom klassegrensene for de to modellene, siden det er lite data her vil denne forskjellen heller ikke ha så stor innvirkning på feilraten. Det er også mange datapunkter på feil side av klassegrensene, noe som igjen gjenspeiler feilraten og at modellen ikke er perfekt med kun variablene skudd_paa_maal_diff og corner_diff.

<b>3a.1)</b>	Hvilke 3 siffer har vi i datasettet? Hvor mange bilder har vi totalt i datasettet?
<b>Svar</b>	Tallene i datasettet er: 9, 3, og 8. Det er 6000 bilder totalt i datasettet.

<b>3a.2)</b>	Hvilket siffer ligner det 500. bildet i datasettet vårt på? Lag et bilde som viser dette sifferet. (Husk at Python begynner nummereringen med 0, og derfor refereres det 500. bildet til [499])
<b>Svar</b>	 <p>Tallet i posisjon 500 er 9.</p>

<b>3b.1)</b>	Tegn sentroidene av de 3 klyngene fra $K$ -gjennomsnitt modellen. Tilpass koden over for å plote. Her kan du ta skjermbilde av sentroidene og lime inn i svararket. Hint: Sentroidene har samme format som dataene (de er 384-dimensjonale), og hvis de er representative vil de se ut som tall.
<b>Svar</b>	<div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>Sentroide 1</p>  </div> <div style="text-align: center;"> <p>Sentroide 2</p>  </div> <div style="text-align: center;"> <p>Sentroide 3</p>  </div> </div>

<b>3b.2)</b>	Synes du at grupperingen i klynger er relevant og nyttig? Forklar. Maks 3 setninger.
<b>Svar</b>	I dette datasettet som inneholder tall vil det være naturlig å ønske gruppere tallene. Dette gir en strukturert visualisering av likheter innenfor datasettet. Slik kan det effektivt brukes til å identifisere mønstre og forenkle analyse av variasjon mellom tallene.
<b>3b.3)</b>	Vi har valgt $K = 3$ for dette eksempelet fordi vi vil finne klynger som representerer de 3 sifrene. Men generelt er $K$ vilkårlig. Kom opp med et forslag for hvordan man (generelt, ikke nødvendigvis her) best kan velge $K$ . Beskriv i egne ord med maks 3 setninger.
<b>Svar</b>	For å velge beste $K$ må man ta inn i betraktning dataen som blir brukt, har datasettet en tydelig inndeling som i vårt tilfelle vil det være naturlig å ha en $K$ som korresponderer med antall grupperinger (Det er jo tre tall i datasettet derfor er $K=3$ naturlig). Ved data som ikke kan kategoriseres av seg selv eller det ikke er øyeblikkelig selvsagt, kan man bruke "The Elbow Method" som gjøres ved å plote inn resultatene av WSS av forskjellige mulige $K$ -verdier, når dette plottes skal det se ut som en "albue" ved en passende $K$ verdi. "The Silhouette Method" tar inn all dataen og returnerer en enkelt verdi, og brukes for hver instans av potensielle $K$ -verdier, den $K$ -verdien med høyest verdi gjennom metoden er da aktuell å bruke som $K$ -verdi.
<b>3b.4)</b>	Kjør analysen igjen med $K = 2$ og $K = 4$ . Synes du de nye grupperingene er relevante?
<b>Svar</b>	Ved $K=2$ er de visualiserte sentroidene mye mindre tydelige, man kan se at bildene av 8-tallene har blitt fordelt mellom de to sentroidene. Dermed blir de visualiserte sentroidene utydelige. Bildene ser ut som veldig 8 formete 3 og 9 tall. Ved $K=4$ blir den ytterligere sentroiden en kombinasjon mellom 8, og 9. Den visualiserte sentroiden blir da en veldig 8 formet 9-tall. Siden toleransen for klyngene blir lavere, vil de resterende sentroidene være skarpere. Dette kan brukes til å se overlappet på tvers av klynger, men ikke særlig brukelig ellers. Ved slik utydelighet faller strukturen sammen, og dermed vil grupperingene ikke være like hjelpsomme, til formålene diskutert i oppgave 3b.2
<b>3c.1)</b>	Vurder dendrogrammet nedenfor. Synes du at den hierarkiske grupperingsalgoritmen har laget gode/meningfulle grupper av bildene? (Maks 3 setninger).
<b>Svar</b>	Dendrogrammet som blir vist klarer til en viss grad å lage gode grupperinger. Man kan se at den klarer å gruppere niere sammen med andre niere, åttre sammen med åttre og treere sammen med treere. Det er en del meningsfulle grupper, allikevel er det en høy grad med avstikkere, dermed kan det sies at den generelle kvaliteten på grupperingsalgoritmen ikke er så bra.
<b>3c.2)</b>	I koden under har vi brukt gjennomsnittskobling ( <code>method = 'average'</code> ). Hvordan fungerer gjennomsnittskobling? (Maks 3 setninger).

<b>Svar</b>	I en hypotetisk situasjon der man har data som kan klynges i 2 klynger, la oss kalle dem klynge en og klynge to. Gjennomsnittskobling er da å beregne alle parvise avstander mellom observasjonene i klynge en og klynge to, og dermed registrere gjennomsnittet av disse avstandene.
-------------	---

<b>3c.3)</b>	Velg en annen metode enn 'average' til å koble klyngene sammen (vi har lært om dette i undervisningen, her heter de <code>single</code> , <code>complete</code> og <code>centriod</code> ) og lag et nytt dendrogram ved å tilpasse koden nedenfor. Ser det bedre/verre ut? (Maks 3 setninger).
--------------	---

<b>Svar</b>	Et nytt dendrogram ble laget med metoden 'single'. Man kan se at det er større avstander fra løvnodene til nye sammenkoblinger, som betyr at 'single' metoden ikke diskriminerer like tydelig som 'average'. Denne metoden vil være mindre hjelpsom siden man ikke får betydelige inndelinger siden de nye grenene er så nærme hverandre, metoden kan da ansees som mindre nøyaktig.
-------------	--

<b>3d.1)</b>	Hvis vi skulle brukt en metode for å predikere/klassifisere hvilket siffer et håndskrevet tall er, og ikke bare samle dem i klynge, hva ville du brukt?
--------------	---

<b>Svar</b>	K-nærmeste-nabo (KNN) er en effektiv og simpel algoritme som kan brukes til formålet å predikere og klassifisere og ikke bare same klynger.
-------------	---