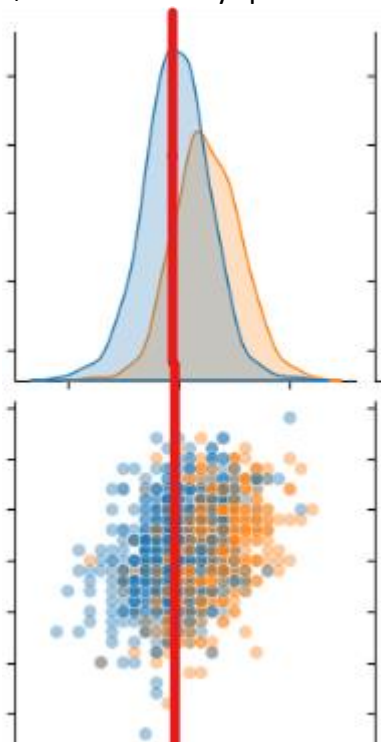


2a.1)	Hvorfor ønsker vi å dele dataene inn i trening-, validering- og test-sett?
Svar	<p>Å dele dataene inn i trening-, validering- og test- sett er en metode man ofte ser i maskinlæring.</p> <p>Treningssettet brukes for å gjenkjenne mønstrene i dataene og er det settet som får den største prosentandelen av dataene.</p> <p>Valideringssettet brukes for å evaluere og finjustere modellen, for å da velge den beste.</p> <p>Testsettet brukes for å evaluere den valgte modell tilpasningen. Etter treningssettet er gjennomført, tester testsettet modellens fremgangsmåte med dens 20% av dataene, for å kunne antyde hvordan modellen vil holde opp med senere data</p> <p>Ved å dele opp dataen i forskjellige sett kan vi forsikre oss om at modellen vil fungere like bra på ny, usett data som med treningsdata, som er målet til enhver maskinlærings modell.</p>
2a.2)	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?
Svar	<p>Datasettet har i starten 2036 rader/kamper som da er 100% av dataen</p> <p>Da dataen skal deles inn i de tre datasettene gis først 80% til trening- og valideringssettet og 20% til testsettet. Deretter deles de den 80% som trening- og valideringssettet hvor treningssettet får 75% og valideringssettet får 25%</p> <p>Treningssettet har da $100\% * 80\% * 75\% = 60\%$ av dataen, som man ser i tabellen tilsvarer 1221 av de 2036 kampene.</p> <p>Valideringssettet har $100\% * 80\% * 25\% = 20\%$ av dataen, som tilsvarer 407.2 kamper og vi kan se i tabellen at valideringssettet har 407 kampen.</p> <p>Testsettet har også sine 20% av den totale dataen, det tilsvarer og 407.2 av 2036, men testsettet har en ekstra kamp (408) for at ikke den resterende kampen skal bli utelatt.</p>
2a.3)	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?
Svar	<p>Når man skal trene opp en modell er det viktig at de tre settene inneholder data med likt grunnlag og representerer de samme problemstillingene man vil møte i praksis.</p> <p>Det ville derfor vært dumt å velge ulike divisjoner for ulike sett ettersom det er tre helt ulike nivåer og spillkvaliteter i de divisjonene. Ved å trene på Ligue1 og 2 og teste på de Premier League vil ikke modellen være generell nok til å brukes for</p>

	all ny data. Treningsmodellen vil lære seg mønstre som ikke gjelder spesifikt for Ligue 1 og Ligue 2 og ikke andre liga og divisjoner.
--	--

2a.4)	Kommenter kort på hva du ser i plottene og utskriften (maks 5 setninger).
Svar	<p>I plottene ser vi distribusjonen av forskjellene i skudd på mål, corner og forseelser, farget i utfallet av kampene (hjemmeseiere eller ikke). I Plottene kan man se et betydelig flertall i hjemmeseiere på den positive siden av skudd på mål diff, og ikke hjemmeseiere i den negative siden av skudd på mål. Forseelser og corner har derimot ikke en like tydelig trend, hvor hjemmeseiere og ikke er spredd ut likt over kryssplottene.</p> <p>I de empiriske tetthetsplottene ser vi at det er totalt flere kamper som har resultert i borteseier. Og at laget som vinner og skudd på mål differansen korrelerer.</p>

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	<p>Skudd på mål er definert den variabelen som direkte leder til flest hjemmeseiere ettersom skudd på mål leder til flere mål, og mål er en viktig faktor i å vinne en fotballkamp. Denne sammenhengen ser vi også utspiller seg i plottene, hjemmeseiere har i de fleste tilfeller positiv skudd på mål diff (høyre siden av den røde streken betyr positiv mål differanse i favør hjemmelaget).</p> 

2b.1)	I en kamp der <code>skudd_paa_maal_diff</code> er 2, <code>corner_diff</code> er -2 og <code>forseelse_diff</code> er 6, hva er ifølge modellen sannsynligheten for at hjemmelaget vinner? Vis utregninger og/eller kode, og oppgi svaret med tre desimaler.
--------------	--

Svar	<p>Sannsynligheten for at hjemmelaget vinner er 0.610 eller 61%</p> <p>Input:</p> <pre>input_data = pd.DataFrame({ 'skudd_paa_maal_diff': [2], 'corner_diff': [-2], 'forseelse_diff': [6] })</pre> <p>sannsynlighet_hjemmeseier = resultat.predict(input_data) print(f"Sannsynligheten for at hjemmelaget vinner er: {sannsynlighet_hjemmeseier[0]:.3f}")</p> <p>Output:</p> <p>"Sannsynligheten for at hjemmelaget vinner er 0.610"</p>
-------------	---

2b.2)	Hvordan kan du tolke verdien av $e^{\beta_{\text{skudd-paa-maal-diff}}}$?
Svar	<p>$e^{\beta_{\text{skudd-paa-maal-diff}}}$ kan tolkes som hvor mye sannsynligheten øker for at hjemmelaget vinner for hvert ekstra skudd på mål. Dersom $\beta_{\text{skudd-paa-maal-diff}} = 0.5$ vil $e^{0.5}$ som er omtrent lik 1.64 bety at hvor hvert ekstra skudd på mål hjemmelaget har, øker sjansen for at hjemmelaget vinner med en faktor på 1.64.</p> <p>Dette regnestykket går da ut på at corner og forseelser er uendret</p>

2b.3)	Hva angir feilraten til modellen? Hvilket datasett er feilraten regnet ut fra? Er du fornøyd med verdien til feilraten?
Svar	<p>Feilraten angir andelen feilaktige predikasjoner modellen gjør.</p> <p>Vanligvis er feilraten regnet ut fra enten validerings- eller testsettet. I dette tilfellet er det regnet ut fra valideringssettet.</p> <p>Ettersom fotballkamper er vanskelige å predikere og hvor tilfeldigheter ofte spiller en stor rolle er 0.285 (ca 28%) en ganske god feilrate.</p>

2b.4)	Diskuter kort hvordan koeffisientene (β – ene) og feilraten endrer seg når <code>forseelse_diff</code> tas ut av modellen (maks 3 setninger).
Svar	<p>Når man fjerner <code>forseelse_diff</code> fra modellen vil de gjenværende koeffisientene for <code>skudd_paa_maal_diff</code> og <code>corner_diff</code> justeres under tilpasningen for å kompensere for endringen i informasjonen. Dette kan føre til enten oppgang eller nedgang på modellens evne til å predikere hjemmeseier, alt etter hvor mye <code>forseelse_diff</code> påvirker resultatet. Som diskutert tidligere er ikke <code>forseelse_diff</code> den variabelen som bidro mest til modellen, men det kan være at den hadde noe effekt som nå ikke vil bidra til modellen.</p>

2b.5)	Med den nye modellen: I en kamp der <code>skudd_paa_maal_diff = 2</code> , <code>corner_diff = -2</code> og <code>forseelse_diff = 6</code> , hva er sannsynligheten for at hjemmelaget vinner ifølge den nye modellen? Oppgi svaret med tre desimaler.
Svar	<p>Med den nye modellen er sannsynligheten for at hjemmelaget vinner 0.591 eller 59.1%</p> <p>Input:</p> <pre>new_input_data = pd.DataFrame({ 'skudd_paa_maal_diff': [2], 'corner_diff': [-2] })</pre> <p><code>new_sannsynlighet_hjemmeseier = new_resultat.predict(new_input_data)</code></p> <p><code>print(f"Sannsynligheten for at hjemmelaget vinner er: {new_sannsynlighet_hjemmeseier[0]:.3f}")</code></p> <p>Output:</p> <p>Sannsynligheten for at hjemmelaget vinner er: 0.591</p>

2b.6)	Hvis du skal finne en så god som mulig klassifikasjonsmodell med logistisk regresjon, vil du velge modellen med eller uten <code>forseelse_diff</code> som kovariat? Begrunn kort svaret (maks 3 setninger).
Svar	Om den beste modellen er med eller uten <code>forseelse_diff</code> avhenger av variabelens prediktive kraft og effekten den har på modellen. Tidligere har vi argumentert for at <code>forseelse_diff</code> ikke har stor effekt på en fotballkamp, dette kan vi og se i kryssplott grafen. I tillegg har den nye modellen uten <code>forseelse_diff</code> en feilrate på 0.283 og modellen med hadde 0.285, som betyr at modellen uten <code>forseelse_diff</code> er litt bedre og vi vil derfor velge den modellen.

2c.1)	Påstand: kNN kan bare brukes når vi har maksimalt to forklaringsvariabler. Fleip eller fakta?
Svar	Fleip. KNN (k-Nearest Neighbors) kan brukes med et hvilket som helst antall forklaringsvariabler. For å klassifisere et nytt datapunkt x, finner algoritmen de k nærmeste datapunktene definert av forklaringsvariablene, uavhengig av hvor mange variabler det er.

2c.2)	Hvilken verdi av k vil du velge?
Svar	Den beste verdien a k vil være en plass i grafen med lav feilrate på valideringssettet. Vi ser i tabellen vår at feilraten synker med økende verdi av k. frem til ca 175 hvor den øker litt igjen. Vi vil derfor velge en verdi av k mellom 100 – 150. Ved å se på alle utskriftene av valideringsfeilrate velger vi k = 115. Den har lavest feilrate og nabooverdiene er generelt lave og.

2d.1)	Gjør logistisk regresjon eller k -nærmeste-nabo-klassifikasjon det best på fotballkampdataene?
--------------	---

Svar	<p>Ettersom vi valgte logistisk regresjon med resultatobjektet fra modellen uten <code>forseelse_diff</code> gjorde logistisk regresjon og k-nærmeste-nabo-klassifikasjon gjør det helt likt på fotballkampdataene:</p> <pre>Feilrate logistisk regresjon: 0.31862745098039214 Feilrate kNN: 0.31862745098039214</pre> <p>Dersom vi hadde valgt modellen med <code>forseelse_diff</code> ville k-nærmeste-nabo-klassifikasjon gjøre det best:</p> <pre>Feilrate logistisk regresjon: 0.32843137254901966 Feilrate kNN: 0.31862745098039214</pre>
-------------	--

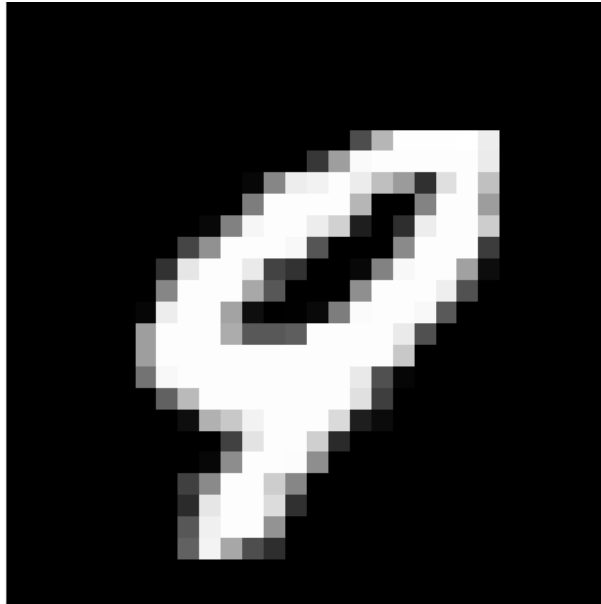
2d.2)	Drøft klassegrensene (plottet under) for de to beste modellene (én logistisk regresjon og én kNN). Hva forteller klassegrensene deg om problemet? Skriv maksimalt 3 setninger.
Svar	<p>Klassegrensene i plottet viser hvordan logistisk regresjon lager en lineær klassegrense, som antyder at det er en antatt lineær sammenheng mellom corner- og skudd på mål-differanse og sannsynlighet for hjemmeseier. KNN-grensen tilpasser seg mer til de individuelle datapunktene, som reflekterer KNNs egenskap til å vektlegge lokal likhet. Dette tyder på at logistisk regresjon prøver å finne en global trend, og kNN prøver å skape mer komplekse grenser som fokuserer på den underliggende fordelingen av dataene.</p>

3a.1)	Hvilke 3 siffer har vi i datasettet? Hvor mange bilder har vi totalt i datasettet?
Svar	Datasettet har 10 bilder totalt. Tre av bildene viser tallet 9, seks av bildene viser tallet 3 og det siste bildet viser 8.

3a.2)	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])
--------------	---

Svar

Bilde 500 i datasettet ligner på tallet 9.



For å få ut dette bilde tok jeg koden som skrev ut de 10 første bildene og endret det for å vise bilde 500 i rekken:

```
# her kan du lime inn og redigere kode for å plotte bildet
fig.add_subplot(1, 11, 11)
plt.imshow(features[499], cmap='gray')
plt.xticks([])
plt.yticks([])
plt.tight_layout()
```

3b.1) Tegn sentroidene av de 3 klyngene fra K -gjennomsnitt modellen. Tilpass koden over for å plotte. 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.

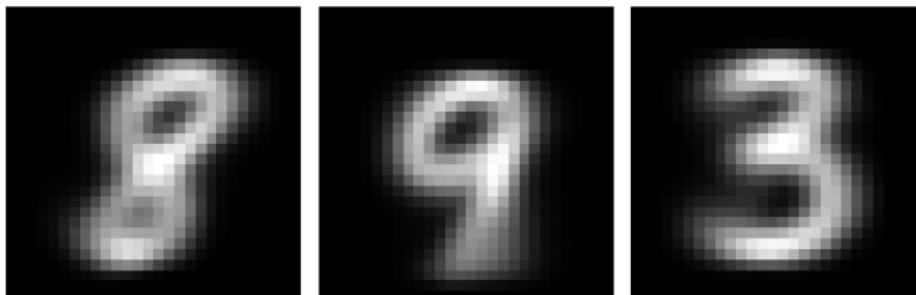
Svar

```
: # her kan du skrive koden for å plote
sentroider_resized = sentroider.reshape(-1, 28, 28)

fig, axes = plt.subplots(1, antall_klynger, figsize=(8, 3))

for i, ax in enumerate(axes.flat):
    ax.imshow(sentroider_resized[i], cmap='gray')
    ax.axis('off')

plt.tight_layout()
plt.show()
```



3b.2) Synes du at grupperingen i klynger er relevant og nyttig? Forklar. Maks 3 setninger.

Svar I tilfelle med håndskrevne tall mener vi at gruppering i klynger er relevant og nyttig. Sentroidene er lett tolkbare ettersom pikslene som lager tallet står godt ut av bakgrunnen og da kan lett tyde hvilket tall som er skrevet. For eksempel i de 10 første tallene hvor tallet 3 er skrevet 6 ganger på 6 litt forskjellige måter, ser man fortsatt lett hvilket tall det er.

3b.3) 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.

Svar For å best velge K i k-means kan man bruke albue-metoden, hvor man velger den k verdien der Within-Cluster-Sum (WSS) begynner å minske i plottet til WSS mot k . Denne nedstigningen vil se ut som en arm, og verdien som wss begynner å minske i er albuen til den armen. Man kan også bruke silhuett metoden hvor man sammenligner andre klynger for å måle hvor lik et objekt er sin egen klynge, den optimale k vil ha en høy silhuettverdi som betyr at det er god klyngetilpasning.

3b.4) Kjør analysen igjen med $K = 2$ og $K = 4$. Synes du de nye grupperingene er relevante?

Svar De nye grupperingene med $K = 2$ og $K = 4$ er relevante ettersom man fortsatt kan se hvilket tall det er. Vi vil argumentere for at $K = 3$ er litt bedre enn 2 og 4 ettersom enkelte bilder er lettere å se med $K = 3$.

3c.1) Vurder dendrogrammet nedenfor. Synes du at den hierarkiske grupperingsalgoritmen har laget gode/meningfulle grupper av bildene? (Maks 3 setninger).

Svar	Dendrogrammet viser at den hierarkiske grupperingsalgoritmen har laget gode grupper av bildene, med flere riktige grupperinger og fått opp til 7 rette tall i en og samme gruppe. Derimot er den ikke perfekt og har avvik, med noen overlapp og blandinger av forskjellige siffer i samme grupper. Det er forventet at noen slike avvik gitt kompleksiteten i å tolke håndskrift, og vil derfor vurdere at grupperingsalgoritmen har gjort en bra jobb.
-------------	--

3c.2)	I koden under har vi brukt gjennomsnittskobling (<code>method = 'average'</code>). Hvordan fungerer gjennomsnittskobling? (Maks 3 setninger).
Svar	Gjennomsnittskobling fungerer ved å beregne gjennomsnittsavstanden mellom alle par av objekter i to forskjellige klynger og deretter kombinere de to klyngene som har den lavest gjennomsnittlige avstanden. Slik blir klynger som er mer lik hverandre slått sammen først og er årsaken til at dendrogrammet illustrerer disse sammenhengene i hierarkisk form. Gjennomsnittskobling gir mer balanserte klynger enn enkeltkobling.

3c.3)	Velg en annen metode enn <code>'average'</code> 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).
Svar	Vi valgte å sammenligne average metoden med single metoden, og det vi ser er et dendrogram som ser bedre ut. Single metoden har større klynger av samme tall med færre feilplasserte tall. Single metoden fungerer ved å slå sammen det paret av klynger som har de to nærmeste medlemmene, så blir disse klyngene kombinert til en.

3d.1)	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?
Svar	For å predikere hvilket siffer håndskrevne tall er ville vi brukt en overvåket maskinlæringsmodell som er spesialisert på mønsterkjenning og bildeklassifisering. Derfor faller valget på en Convolutional Neural Networking (CNN) metode, som er god på bildegjenkjenning og ofte brukt i optisk tegngjenkjenning systemer. CNN-er fanger opp de romlige hierarkiene i bilder ved å lære seg de lokale mønstrene, som former og kanter, og deretter bygger opp til mer komplekse strukturer som i dette tilfelle siffer.