## Tehtävä 1. Naivi-Bayes ja SPAM (2 pistettä).

Naivi-Bayes-mallia käytetään roskapostin suodattamiseen. Mallin todennäköisyysparametrien oppimista varten on kerätty valmiiksi luokiteltuja sähköpostiviestejä. Seuraavassa on joidenkin sanojen esiintymismäärät kummassakin luokassa (spam ja ham):

| **sana** | **spam** | **ham** |
|---:|---:|---:|
| Bayesin | 0 | 5 |
| kaavaa | 2 | 8 |
| me | 9 | 10 |
| rakastamme | 11 | 1 |
| *yht.* | 9 000 | 12 000 |

Arvioi ehdolliset todennäköisyydet $\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{spam})$ ja $\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{ham})$, missä $s \in \{\text{kaavaa, Bayesin, rakastamme, me}\}$.

Laske näitä hyödyntäen seuraavat todennäköisyydet:

i. (1 piste) $\mathrm{P}(\mathrm{spam} \mid \mathrm{me})$

ii. (1 piste) $\mathrm{P}(\mathrm{spam} \mid \mathrm{me}, \mathrm{rakastamme}, \mathrm{Bayesin}, \mathrm{kaavaa})$

Käytä prioritodennäköisyytenä arvoa $\mathrm{P}(\mathrm{spam}) = 0.5$.

(*Vihje:* Pienten todennäköisyyksien kohdalla on syytä käyttää jotakin alarajaa, esim. 0.0001. Muista että jos saat laskettua osamäärän $Odds = P(\mathrm{spam} \mid \mathrm{evidenssi})/P(\mathrm{ham} \mid \mathrm{evidenssi})$, saat todennäköisyyden kaavalla $P(\mathrm{spam} \mid \mathrm{evidenssi}) = Odds/(1 + Odds)$.)

## Tehtävä 2. Numeroiden luokittelu neuroverkolla (2 pistettä).

Toteuta valmiiseen Java-ohjelmarunkoon, tai kokonaan alusta, perseptronialgoritmi numeroiden tunnistamiseen. Tiedosto `mnist-x.data` sisältää yhteensä 6000 kuvaa, yksi kuva jokaisella tiedoston rivillä, ja jokainen kuva on 28 x 28 pikseliä (eli jokaisella rivillä siis 28 x 28 = 784 arvoa). Jokainen pikseli on joko musta (-1) tai valkoinen (1). Tiedosto `mnist-y.data` sisältää näitä kuvia vastaavat luokat (0-9).

a) Suorita Java-pohja kerran ja varmista, että projektin runkoon ilmestyy tiedosto `test100.bmp`, jossa on sata ensimmäistä numeroa suuruusjärjestyksessä. Tällä tavalla verifioidaan että datan lukeminen on onnistunut.

b) (1 piste) Muokkaa perseptronialgoritmia (`Perseptroni`-luokan `train()`-metodi) siten, että se oppii erottamaan numeroa '3' (`targetChar`) esittävät kuvat numeroa '5' (`oppositeChar`) esittävistä kuvista. (Aseta luokkamuuttujan arvoksi 1, kun kuva esittää kolmosta, ja -1 jos se esittää vitosta.) Minkä luokitteluvirheen saat aikaiseksi?

c) (1 piste) Kokeile eri numeroparien erottelua (muitakin kuin 3 vs 5). Mitkä numerot on helpoin erottaa toisistaan, mitkä vaikein?

## Tehtävä 3. Numeroiden luokittelu lähimmän naapurin luokittimella (1 piste).

Korvaa edellisen tehtävän ohjelmassa perseptroniluokittelija lähimmän naapurin luokittelijalla (tai koodaa kokonaan alusta).

Luokittelija etsii siis kullekin testiaineiston esimerkille, $X$, sitä vastaavan lähimmän opetusaineiston esimerkin, $X^{\text{train}}$, ja palauttaa sitä vastaavan luokan $Y^{\text{train}}$. Huomaa, että toisin kuin perseptroniluokitinta, lähimmän naapurin luokitinta ei varsinaisesti tarvitse opettaa vaan kaikki työ tehdään luokitteluvaiheessa.

Testaa taas luokittelijaasi samoilla opetus- ja testijoukoilla kuin edellisessä tehtävässä. Voit nyt ottaa mukaan kaikki luokat (numerot 0-9). Huomaa, että luokittelu 10:en luokkaan on vaikeampaa kuin luokittelu kahteen luokkaan, joten luokittelutarkkuus luultavasti alenee.