Huom: Voit saada tästä harjoituskerrasta max. 6 pistettä.

Tehtävä 1. Naivi Bayes ja SPAM (3 pistettä)

a) (1 piste). Bayesin kaava:

$$P(\text{SPAM} \mid \text{SANA}_1 = \textit{Viagra}) = \frac{P(\text{SANA}_1 = \textit{Viagra} \mid \text{SPAM})P(\text{SPAM})}{P(\text{SANA}_1 = \textit{Viagra})}$$
$$= \frac{P(\text{SANA}_1 = \textit{Viagra} \mid \text{SPAM})P(\text{SPAM})}{P(\text{SANA}_1 = \textit{Viagra} \mid \text{SPAM})P(\text{SPAM}) + P(\text{SANA}_1 = \textit{Viagra} \mid \neg\text{SPAM})P(\neg\text{SPAM})}$$
$$= \frac{0.0021 \cdot 0.5}{0.0021 \cdot 0.5 + 0.0001 \cdot 0.5} \approx 0.95.$$

b) (2 pistettä). Filtterin toteutus (tällä kertaa C#:lla; kiitokset Juhani Markkulalle):
SpamFilter.cs

```
using System;
using System.Collections.Generic;

namespace JM.AI.Ex4
{
    internal class SpamFilter
    {
        private readonly IDictionary<string, decimal> _wordProbabilitiesInHam;
        private readonly IDictionary<string, decimal> _wordProbabilitiesInSpam;
        private readonly TextTokenizer _tokenizer = new TextTokenizer();

        public SpamFilter(IDictionary<string, decimal> spamProb,
                          IDictionary<string, decimal> hamProb)
        {
            _wordProbabilitiesInSpam = spamProb;
            _wordProbabilitiesInHam = hamProb;
            this.SpamPropability = 0.5m;
        }

        public decimal SpamPropability { get; set; }

        public decimal EstimateSpamicity(string text)
        {
            if (text == null) throw new ArgumentNullException(‘‘text’’);

            var words = _tokenizer.Tokenize(text);
            double pSpam = decimal.ToDouble(this.SpamPropability);
            double odds = Math.Log(pSpam/(1 - pSpam));

            foreach (string word in words)
```

```
            {
                bool hasSpamProb = _wordProbabilitiesInSpam.ContainsKey(word);
                bool hasHamProb = _wordProbabilitiesInHam.ContainsKey(word);

                if (!hasHamProb && !hasSpamProb) continue;
                // avoid zero
                decimal pWordSpam = hasSpamProb ? _wordProbabilitiesInSpam[word] : 0.0001m;
                // avoid zero
                decimal pWordHam = hasHamProb ? _wordProbabilitiesInHam[word] : 0.0001m;

                odds += Math.Log(decimal.ToDouble(pWordSpam/pWordHam));
            }

            return new decimal(Math.Exp(odds));
        }
    }
}
```

EstimateSpamicity()-funktio:

```
        private static void EstimateSpamicity(string path)
        {
            SpamFilter filter = new SpamFilter(PropabilitiesInSpam, PropabilitiesInHam);
            TextFileReader reader = new TextFileReader();

            try
            {
                string content = reader.Read(path);
                decimal odds = filter.EstimateSpamicity(content);
                decimal propability = decimal.Round((odds/(odds + 1)), 3);
                Console.WriteLine(``Odds: `` + odds);
                Console.WriteLine(``Probability of spam: `` + propability);
            }
            catch (ArgumentException e)
            {
                Console.WriteLine(``Given file could not be read.'');
            }
        }
```

Esimerkkiajo:

```
> .\AI-Ex4.exe .\ham.txt
Odds: 0
Probability of spam: 0
> .\AI-Ex4.exe .\ham2.txt
Odds: 0
Probability of spam: 0
> .\AI-Ex4.exe .\spam.txt
Odds: 10277051,7420194
Probability of spam: 1,000
> .\AI-Ex4.exe .\spam2.txt
Odds: 19871296166,299
Probability of spam: 1,000
```

(Koska todennäköisyydet ovat niin lähellä nollaa tai ykköstä ne pyörityivät näissä esimerkeissä nolliksi tai ykkösiksi.)

Tehtävä 2. Hahmontunnistus (3 pistettä)

a) (1 piste). Tulokset vaihtelivat kuvittain, mutta joissain tapauksissa, etenkin kun oli otettu samalla kameralla peräkköin monta kuvaa samasta kohteesta, SURF-hahmontunnistus tuntui toimivan melko hyvin.

b) (1 piste). Muut muutokset kuin sumentaminen näyttivät häiritsevät menetelmää melko vähän. Sumentaminen sen sijaan sotki tuloksia melko paljon.

c) (1 piste). Kuten luentokalvoilla on mainittu, digitaalisissa signaaleissa on usein hyvin paljon dataa (pikseleitä, tms) ja kohinaa. Symbolista aineistoa on sen sijaan vähemmän ja sen kohdalla kohinalla ei ole yhtä selkeää merkitystä.

   Symboliseen dataan sovelletaan tyypillisesti GOFAI-menetelmiä, kuten logiikkaa. Digitaalisten signaalien kohdalla tyypillisiä ongelmia ovat kohinanpoisto (johon voi soveltaa vaikkapa kuvan siloitusta (smoothing) tai aallokkeita (wavelets) sekä hahmontunnistus, johon voi soveltaa SURF:in kaltaisia menetelmiä.

Tehtävä 3. Yleiskuva (1 piste)

Ei esimerkkiratkaisua. Voit kerrata ensimmäisen luennon materiaaleja ja koealueen kuvasta, joka löytyy kursssin sivulta.