1.1 TF-IDF

Definer formelen til vektingsmålet TF-IDF ("Term Frequency – Inverse Document Frequency") og forklar notasjonen du bruker. Diskuter kort hva som er hensikten med å anvende TD-IDF.

Vekten for en term t_i i et dokument d_j er gitt ved:

$$\text{tf-idf}(t_i, d_j) = \text{tf}(t_i, d_j) \times \text{idf}(t_i)$$

$$\log\left(\frac{N}{df(t_i)}\right)$$

term frequency: antall ganger termen $\mathbf{t_i}$ forekommer i dokument $\mathbf{d_j}$

Inverse document frequency, der **N** er det totale antall dokumenter i samlingen. **Høy verdi betyr at det forekommer i få dokumenter.**

document frequency: det totale antall dokumenter termen forekommer i.

1.1 TF-IDF

Definer formelen til vektingsmålet TF-IDF ("Term Frequency – Inverse Document Frequency") og forklar notasjonen du bruker. Diskuter kort hva som er hensikten med å anvende TF-IDF.

- Rå frekvens er en dårlig indikator for relevans - vekting
- En høy verdi for tf-idf -> høy frekvens i dokumentet, men lav frekvens i samlingen som helhet
- Vanlige ord får lav vekt

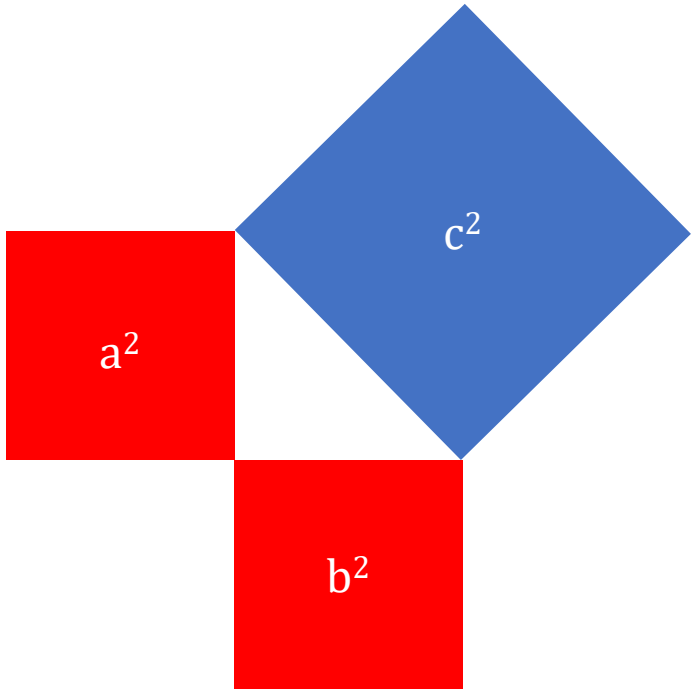
1.2 Lengdenormalisering

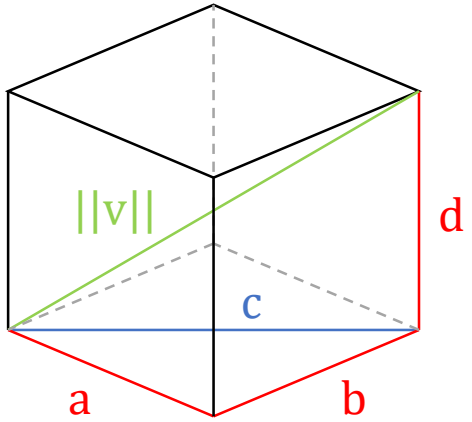
Når vi jobber med vektorrom-representasjoner av dokumenter benytter vi oss ofte av lengde-normalisering. Forklar hva dette innebærer og hvilken praktisk nytte det kan ha.

- Sørge for at alle vektorer har en euclidisk norm lik 1
- Oppnås ved å dele hvert element (dimensjon) på lengden (normen):

$$x \frac{1}{\|x\|}$$

- Motvirker problemet med at ordfrekvenser og lengden på dokumenter påvirker euklidisk avstand
- Med normaliserte vektorer kan cosine similarity regnes ut med prikkproduktet av vektorene, noe som gjør det svært effektivt.
- Rekkefølgen på hvordan vektorene står i forhold til hverandre blir det samme i cosine similarity og Euclidean distance

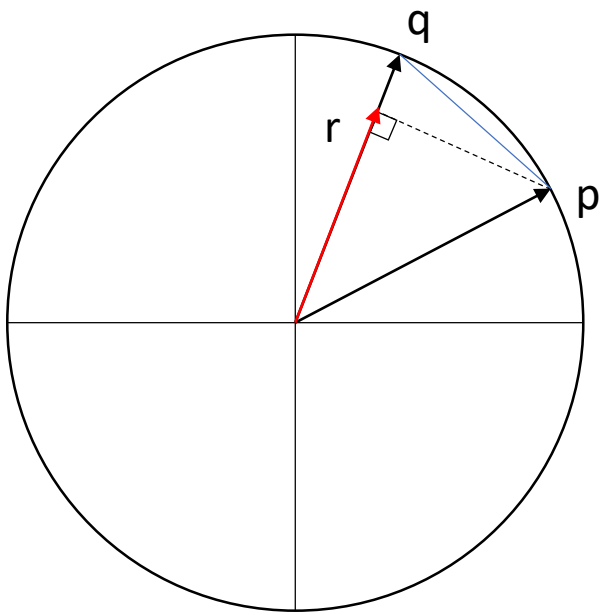


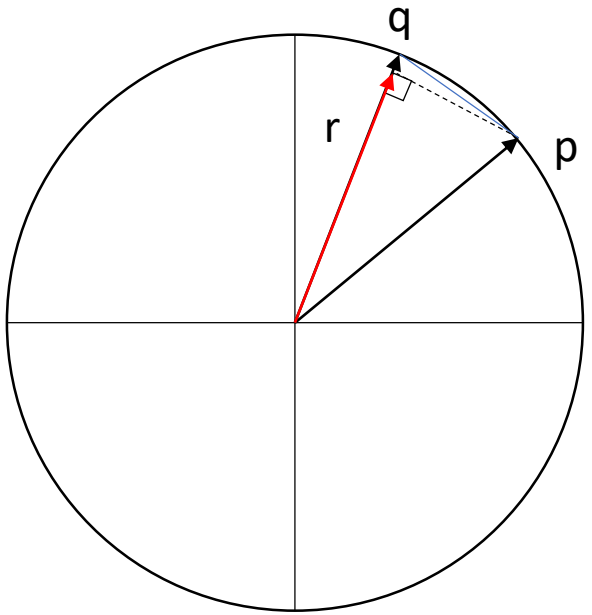


$$c^2 = a^2 + b^2$$

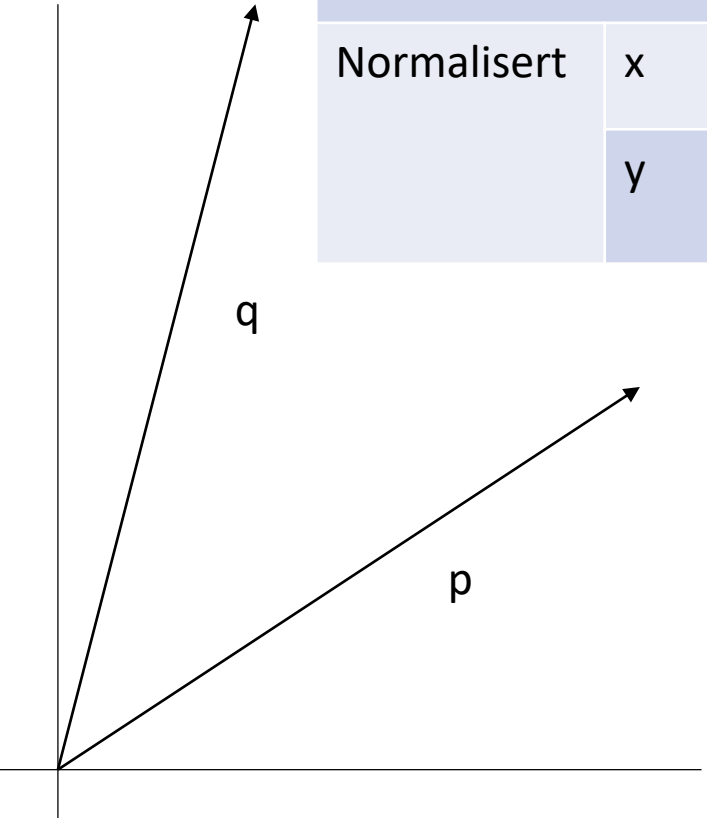
$$c = \sqrt{a^2 + b^2}$$

$$\|v\| = \sqrt{c^2 + d^2} = \sqrt{\sqrt{a^2 + b^2}^2 + d^2} = \sqrt{a^2 + b^2 + d^2}$$


q
r
p


q
r
p

		p	q	Euclidean distance	Cosine similarity
x		3	1	$\sqrt{(1-3)^2+(4-2)^2} = 2{,}83$	$\frac{3\times 1+2\times 4}{3{,}61\times 4{,}12} = 0{,}74$
y		2	4		
Euklidisk norm		$\sqrt{3^2+2^2} = 3.61$	$\sqrt{1^2+4^2} = 4.12$		
Normalisert	x	3/3.61 = 0.83	1/4.12 = 0.24	$\sqrt{(0.24-0.83)^2+(0.97-0.55)^2} = 0{,}72$	$0{,}83\times 0{,}24+0{,}55\times 0{,}97 = 0{,}74$
	y	2/3.61 = 0.55	4/4.12 = 0.97		



2.1 Evaluering

Flere av evalueringsmålene vi har sett på i kurset har vært definert på basis av fire mer grunnleggende kategorier av hvordan prediksjonene til en klassifikator kan være riktige eller gale, sammenliknet med gullstandarden:

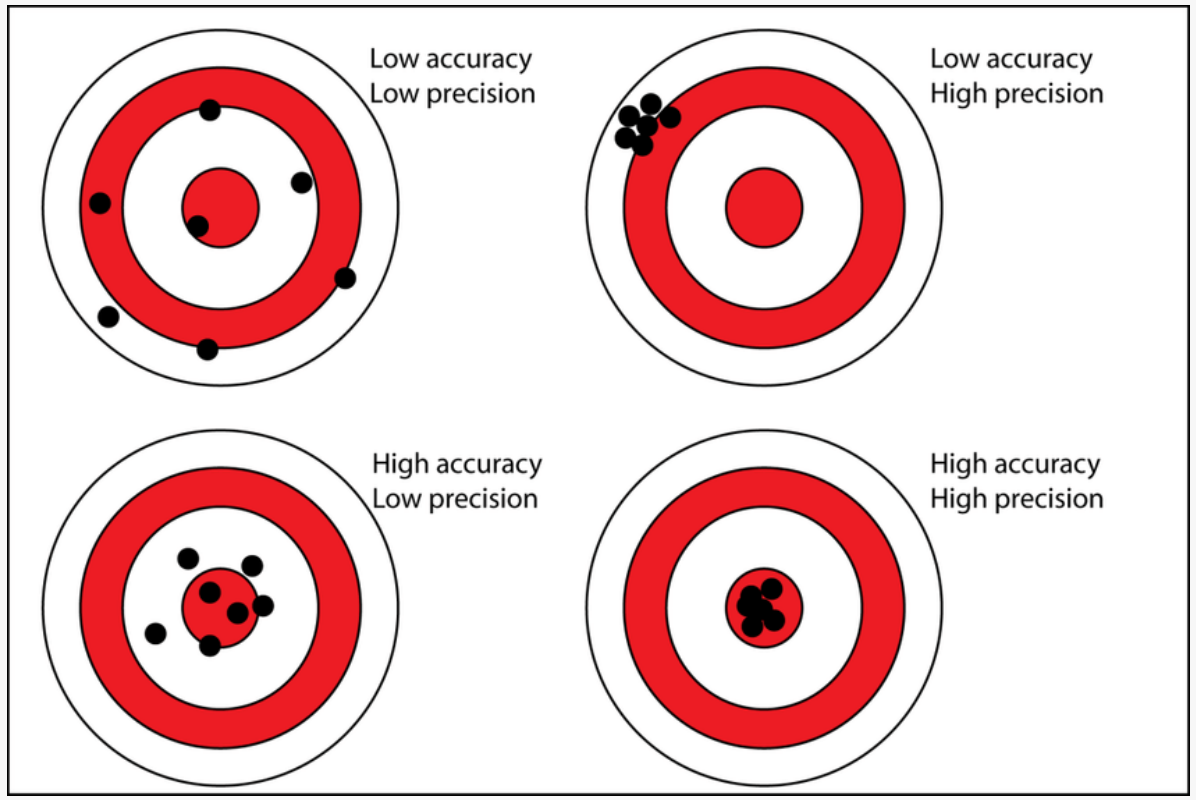
		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Vis hvordan de tre målene Accuracy, Recall og Precision kan defineres på bakgrunn av dette.

Accuracy $\frac{TP + TN}{N}$

Precision $\frac{TP}{TP + FP}$

Recall $\frac{TP}{TP + FN}$

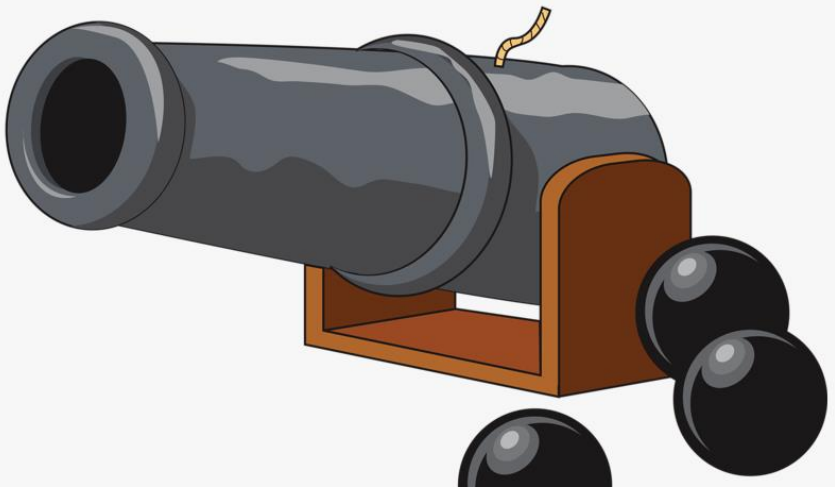


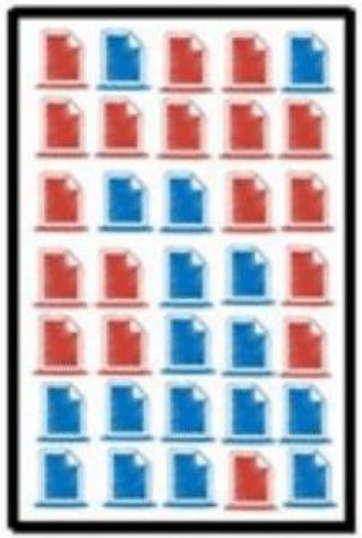
Riktige delt på alle: **Accuracy** $\frac{TP + TN}{N}$

Lite falske positive: **Precision** $\frac{TP}{TP + FP}$

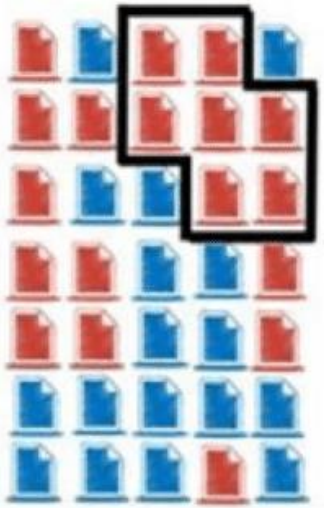
Lite falske negative: **Recall** $\frac{TP}{TP + FN}$

High recall

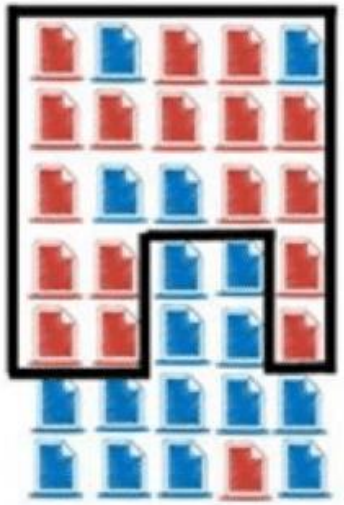




1) Perfect Recall;
Low precision



2) Low Recall;
Perfect Precision



3) Arguably Good
Recall and Precision

Accuracy $\frac{TP + TN}{N}$

Precision $\frac{TP}{TP + FP}$

Recall $\frac{TP}{TP + FN}$

2.2 Accuracy

Tenk deg at vi jobber med binær klassifikasjon og at vi har mange flere eksempler i den negative klassen enn den positive (la oss anta et forhold på 9:10). Diskuter hvorvidt Accuracy er et egnet eller uegnet evalueringsmål for dette problemet.

Accuracy er ikke egnet hvis det er mange negative – vi kan få nokså høy accuracy ved å klassifisere alt som negativt. Årsaken til dette er at accuracy gir uttelling for bade true negatives og true positives:

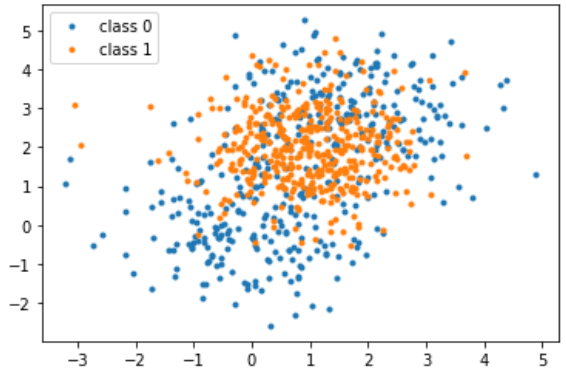
$$\frac{TP + TN}{N}$$

2.3 kNN

Beskriv kort klassifikasjonsmetoden kNN. Diskuter kort dens styrker og svakheter

- k Nearest Neighbors er en veiledet metode (supervised learning): lærer fra eksempler som allerede har blitt annotert med riktig klasse.
- kNN klassifiserer etter majoriteten blant de k nærmeste naboene (typisk etter avstand i vektorrommodell)
- Vanlig å vekte etter avstand, slik at nærmere naboer får mer å si

2.3 kNN



Styrker

- Enkel å forstå
- håndterer ikke-lineært separerbare klasser
- Kompleksiteten ved klassifikasjon (testing) er uavhengig av antall klasser

Svakheter

- Vanskelig å avgjøre hva som er optimal k
- **minnebruk** - metoden memorerer alle treningseksemplene. (memory-based learning eller instance-based learning) – ingen egentlig læring
- Relativt høy **tidskompleksitet** for å finne nærmeste nabo – lineært etter antall treningseksempler og dimensjoner (jfr. memory-based)