NTNU
Kunnskap for en bedre verden

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4305 Big Data-Arkitektur

Faglig kontakt under eksamen: Kjetil Nørvåg/Heri Ramampiaro

Tlf.: 73596755/73591459

Eksamensdato: 16. mai 2017

Eksamenstid (fra-til): 09.00-13.00

Hjelpemiddelkode/Tillatte hjelpemidler: D: Ingen trykte eller håndskrevne hjelpemiddel tillatt.
Bestemt, enkel kalkulator tillatt.

Annen informasjon:

Målform/språk: Bokmål

Antall sider (uten forside): 4

Antall sider vedlegg: 0

Kontrollert av:

_____

Dato Sign

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig X **2-sidig** ☐

sort/hvit X **farger** ☐

Merk! Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

Oppgave 1 – Hadoop – 15 % (alle deler teller likt)

a) Forklar replikering i HDFS.

b) Hva er hensikten med NameNode, og hvilken informasjon har denne lagret?

c) Hva er input og output på hhv. Map og Reduce-funksjonene?

Oppgave 2 – Spark – 20 % (alle deler teller likt)

Anta at man har en fil *countries.tsv* (tsv = tab-separert) som inneholder informasjon om land og karakteristiske termer for landene (i dette tilfellet kan dere anta en tab "\t" per linje, mellom land og term-listen):

```
France     wine,art,trains,wine,trains
Canada     glaciers,lakes,hockey,maple,grizzly
Norway     fjords,fjords,glaciers,trolls
Japan      trains,sushi,origami,sushi,fuji
Argentina  wine,glaciers,football,glaciers
```

I denne oppgaven skal dere for hver av deloppgavene under vise hvordan de kan løses vha. Spark-transformasjoner/aksjoner (Scala, Python eller Java).
Hint: Splitte en tekststreng x basert på "\t" (tab): val y = x.split("\t")

a) Lag en RDD med navn ”data” basert på filen, hvor hver post/objekt er en tekststreng (String) inneholdende en linje fra filen.

b) Lag en ny RDD ”data2” der hvert objekt i ”data” er en ”array/list of strings”, der første tekst-streng er land og tekststreng nummer to inneholder termene. Eksempelresultat (Python):

```
[['France', 'wine,art,trains,wine,trains'],
 ['Canada', 'glaciers,lakes,hockey,maple,grizzly'],
 ['Norway', 'fjords,fjords,glaciers,trolls'],
 ['Japan', 'trains,sushi,origami,sushi,fuji'],
 ['Argentina', 'wine,glaciers,football,glaciers']]
```

c) Lag en ny par-RDD (pair RDD) ”data3” der nøkkel (key) er land, og verdi (value) er en “array/list of strings” av karakteristiske termer, eksempelresultat:

```
[('France', ['wine', 'art', 'trains', 'wine', 'trains']),
 ('Canada', ['glaciers', 'lakes', 'hockey', 'maple', 'grizzly']),
 ('Norway', ['fjords', 'fjords', 'glaciers', 'trolls']),
 ('Japan', ['trains', 'sushi', 'origami', 'sushi', 'fuji']),
 ('Argentina', ['wine', 'glaciers', 'football', 'glaciers'])]
```

d) Finn antall karakteristiske termer i datasettet (ikke inkludert navn på landene).

e) Finn antall *distinkte* karakteristiske termer i datasettet.

f) Basert på ”data3”, lag en RDD som inneholder antall distinkte termer for hvert land. Eksempelresultat:

```
[('Argentina', 3), ('Norway', 3), ('France', 3), ('Canada', 5),
('Japan', 4)]
```

Oppgave 3 – NoSQL – 5 %

Forklar CAP-teoremet.

Oppgave 4 – MinHashing – 10 % (alle deler teller likt)

lx	Element	S1	S2	S3	S4	lx2
0	p	0	0	0	1	3
1	a	0	1	1	1	2
2	g	1	1	0	0	1
3	i	1	1	1	1	0
4	k	0	1	0	0	4

a) Forklar ”shingling” og hensikten med ”shingling”.

b) Figuren ovenfor viser en forekomstmatrise for elementene p/a/g/i/k i settene S1/S2/S3/S4. Hva er MinHash-signaturene gitt permutasjonene lx og lx2?

Oppgave 5 – Datastrømmer (streaming data) – 30 % (alle deler teller likt)

Anta at du er ansatt i et firma som analyserer interessetrender for det nye kameraet Sony A9 som Amazon nettopp har startet å selge. Firmaet ditt har bestemt seg for å gjøre dette basert på sosiale media data, inkl. Twitter og Facebook.

a) Gå ut fra at du starter med å bestemme deg for finne antall meldinger som nevner ”Sony A9” ved å bruke såkalte stående spørring (”Standing Query”). Forklar hva som er forskjellen(e) mellom ”standing query” og ”ad-hoc query”. Hvordan kan ”standing query” løse oppgaven?

b) Siden vi kan se sosiale media data som datastrøm har de også de andre karakteristikker og/eller utfordringer enn statiske data. Drøft hva disse er.

c) Som alternativ til ”standing query” velger du ”bit counting” for løse deler av oppgaven. Forklar hvordan kan oppgaven oversettes til ”bit counting”.

d) Analysen din skal se på trendene de siste 2 ukene og du ser for deg antall ”klikk” og ”kjøp” av produkter i Amazon direkte som en del av analysen. Se for deg at du skal finne ut hvor stor andel av ”klikk” og ”kjøp” er gjort på ”Sony A9”. Til dette formålet velger du nå å bruke glidende-vindu-prinsippet (“sliding window”). Anta at dette vinduet har en størrelse på 500 ”klikk” og ”kjøp” tilsammen. Forklar hvordan du går fram for å beregne denne andelen.

e) Når vi først er inne på ”klikk” drøft hvordan ”bloom filter” kan være nyttig til å finne antall klikk. Gjør de antakelsene du finner nødvendig.

f) Bruk “bloom filter”-prinsippet til å fylle ut tabellen nedenfor

Strømelement	**Hash-funksjon - h_1**	**Hash-funksjon - h_2**	**Filtrere Innhold** 00000000000
56 = 11 1000			
428 = 1 1010 1100			
875 = 11 0110 1011			

Hint: bruk *h1(x)=y1 mod 11* og *h2(x)=y2 mod 11* som hash-funksjoner, der verdiene av *y1 og y2* er hentet henholdsvis fra oddetalls-bits fra *x* og partalls-bits fra *x*.

Oppgave 6 – Anbefalingssystem og sosiale grafer (recommender systems and social graphs) – 20 % (alle deler teller likt)

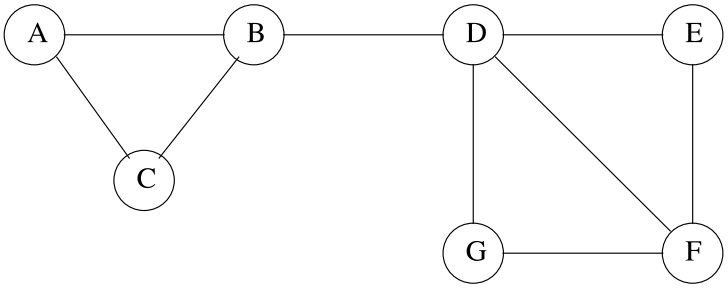
Firmaet du er nyansatt i er spesialist på anbefalingssystemer. Din oppgave er å utvikle gode anbefalingsalgoritmer og metoder.

a) Med fokus på fordeler og ulemper sammenlikn ”collaborative filtering” mot ”content-based recommendation”.

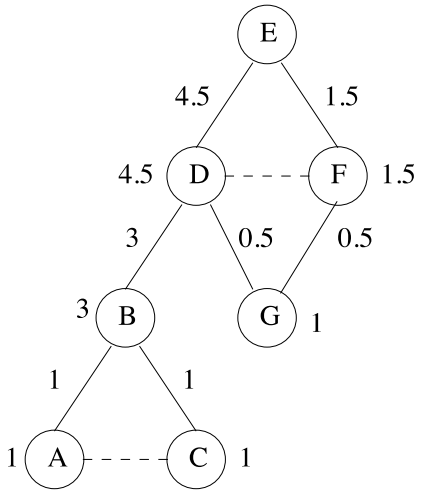
b) Anta at du vil bruke bruker-relasjoner, som feks. “friends” og “followers” fra sosiale media, til å bygge opp brukerprofiler. Slike relasjoner kan tegnes i en graf. Bruk dette som utgangspunkt til å svare på følgende spørsmål:

   i. Forklar forskjellene på ”*overlapping*” og ”*non-overlapping communities”* i grafer. Hvordan kan vi enkelt finne ut om to grafer overlapper.
   ii. Hva er hovedhensiktene med ”community detection”? Forklar.

c) Girvan-Newmans metode for beregning av ”betweenness” er en vanlig metode innen graf-mining-teori. Vis hvordan du går fra Figur 1 til Figur 2 nedenfor ved å bruke Girvan-Newmans metode til å beregne ”*betweenness”* i grafen i Figur 1.



Figur 1: Start-graf



Figur 2: Steg 3 i første iterasjon.

d) Hvordan kan resultatfiguren nedenfor brukes til detektere ”communities”?