جزوه امتحانی — داده‌های حجیم (Big Data Analytics)

استاد: دکتر فرساد زمانی بروجنی | **امتحان:** جزوه‌باز — ۹۰ دقیقه | **منبع:** لکچرهای ترم + نمونه سوالات + MMDS

روش پاسخ در امتحان: تعریف یک‌خطی → فرمول → مثال عددی کوتاه. اول **چیت‌شیت** و **فهرست صفحات** را ببین، بعد برو بخش مربوطه. اگر سوال شبیه نمونه استاد بود → بخش ۱۰.

چیت‌شیت سریع

مفهوم	تعریف یک‌خطی (همان را بنویس)
Shingle	دنبالهٔ k حرف/کلمهٔ پشت‌سرهم در سند
Jaccard	اندازه اشتراک دو مجموعه ÷ اندازه اجتماع آن‌ها
MinHash	درهم‌سازی طوری که احتمال برابر شدن hash دو مجموعه = Jaccard باشد
LSH	روش سریع برای پیدا کردن فقط جفت‌هایی که احتمالاً مشابه‌اند (candidate)
Collaborative Filtering	توصیه کالا/فیلم بر اساس کاربران یا آیتم‌های شبیه هم
Closed itemset	مجموعه پرتکراری که هیچ ابرمجموعه‌ای با همان support ندارد
Maximal itemset	مجموعه پرتکراری که هیچ ابرمجموعه پرتکراری ندارد
MapReduce	پردازش موازی داده حجیم با دو تابع Map و Reduce و تحمل خرابی

فرمول‌های حیاتی (سبک Word)

$$TF_{ij} = f_{ij} / \max_k(f_{kj})$$

$$IDF_i = \log_2(N / n_i)$$

$$TF\text{-}IDF = TF_{ij} \times IDF_i$$

که در آن:

f_{ij} = تعداد تکرار کلمه i در سند j

$\max_k(f_{kj})$ = بیشترین تکرار هر کلمه در همان سند j

N = تعداد کل اسناد

n_i = تعداد اسنادی که کلمه i در آن‌ها آمده

$$J(A, B) = |A \cap B| / |A \cup B|$$

$$d_J(A, B) = 1 - J(A, B)$$

$$Pr[\, h_\pi(C_1) = h_\pi(C_2) \,] = J(C_1, C_2)$$

$$P(candidate) = 1 - (1 - s^r)^b$$

$$\text{آستانه تقریبی} \approx (1/b)^{(1/r)}$$

که در آن:

s = شباهت واقعی دو سند

r = تعداد سطر در هر band

b = تعداد bandها

$$L_2 \text{ (اقلیدسی)} = \sqrt{\left(\Sigma (x_i - y_i)^2 \right)}$$

$$L_1 \text{ (منهتن)} = \Sigma |x_i - y_i|$$

$$L_r = \left(\Sigma |x_i - y_i|^r \right)^{(1/r)}$$

$$L_\infty = \max |x_i - y_i|$$

$$d_{\text{اسمی}} = (p - m) / p$$

$$d_{\text{دودویی نامتقارن}} = (r + s) / (q + r + s)$$

$$\text{support}(A \Rightarrow B) = \text{support}(A \cup B)$$

$$\text{confidence}(A \Rightarrow B) = \text{support}(A \cup B) / \text{support}(A)$$

$$M \subseteq C \subseteq F$$

$$\text{Map: } (i\ ,\ m_{ij} \times v_j)$$

$$\text{Reduce: } u_i = \Sigma_j\ (m_{ij} \times v_j)$$

که در آن:

برای ضرب ماتریس M در بردار V و به‌دست آوردن U

موضوع	نکته طلایی
RDBMS ↔ Cluster	عمودی/گران/ACID ↔ افقی/commodity/replication و انتقال کد به داده
K-Means ↔ K-Medoids	میانگین (حساس به پرت) ↔ نقطه واقعی داده (مقاوم‌تر)
Closed ↔ Maximal	Closed support را نگه می‌دارد؛ Maximal فقط مرز را
Combiner	فقط اگر Reduce انجمنی و جابجایی‌پذیر باشد

فهرست صفحات (راهنمای سریع)

در حال محاسبه شماره صفحات...

§۱ — مقدمه TF-IDF ، Big Data و مفاهیم پایه

Big Data (سه V)

- **Volume:** حجم خیلی زیاد
- **Velocity:** تولید با سرعت بالا
- **Variety:** شکل‌های مختلف داده

داده‌کاوی: استخراج الگو/دانش مفید از داده.

دو تکنیک مهم: ۱) خلاصه‌سازی (PageRank، خوشه‌بندی) ۲) استخراج ویژگی (frequent itemsets، مشابهت، توصیه‌گر)

PageRank: صفحه مهم‌تر است اگر از صفحات مهم لینک بگیرد.

اصل Bonferroni: در داده عظیم، رخدادهای خیلی نادر هم زیاد دیده می‌شوند → خطر False Positive.

Index ، Hash ، حافظه ثانویه

- **Hash:** کلید را به شماره bucket بین 0 تا B−1 می‌برد. مثال ساده: h(x) = x mod B
- **Index:** ساختمان‌داده برای بازیابی سریع رکورد با داشتن مقدار یک صفت

- **Secondary Storage:** دیسک خیلی کندتر از RAM است (~10^5 برابر)؛ الگوریتم‌های Big Data باید I/O دیسک را کم کنند

TF-IDF

کلمه مفید = در تعداد کمی سند، زیاد تکرار شده. Stop wordها (مثل «و»، «the») مفید نیستند.

$$TF_{ij} = f_{ij} / \max_k(f_{kj})$$
$$IDF_i = \log_2(N / n_i)$$
$$TF\text{-}IDF = TF_{ij} \times IDF_i$$

مثال ۱ — قدم‌به‌قدم (مثل سوال استاد)

فرض کن ۱۰۲۴ سند داریم. کلمه w در ۱۲۸ سند آمده. در سند j این کلمه ۱۰ بار آمده و پرتکرارترین کلمه همان سند ۱۰۰ بار آمده.

```
۱ گام) TF = 10 / 100 = 0.1
۲ گام) IDF = log2(1024 / 128) = log2(8) = 3
۳ گام) TF-IDF = 0.1 × 3 = 0.3
```

نکته: $\log_2(8)=3$ چون $8=2^3$.

پاسخ نهایی: ۰٫۳

مثال ۲ — برای فهم بهتر

۳ سند داریم. کلمه «داده» در هر ۳ سند آمده → $N=3$ ، $n=3$ → $IDF = \log_2(3/3)=0$. پس این کلمه هیچ اهمیتی ندارد (مثل stop word).

کلمه «هادوپ» فقط در ۱ سند آمده و در آن سند ۸ بار آمده؛ بیشترین تکرار کلمات آن سند ۲۰ است.

```
TF = 8/20 = 0.4
IDF = log2(3/1) = log2(3) ≈ 1.58
TF-IDF ≈ 0.4 × 1.58 = 0.63
```

پاسخ نهایی: ≈ ۰٫۶۳

Collaborative Filtering

سیستم توصیه‌گر: از رفتار کاربران مشابه (یا آیتم‌های مشابه) برای پیشنهاد استفاده می‌کند.

- **User-based:** کاربر شبیه تو چه چیزهایی را پسندیده؟
- **Item-based:** آیتم‌های شبیه آنچه پسندیدی کدام‌اند؟

در درس به Recommendation Engine (Netflix، Amazon) و Behavioral Data اشاره شده.

§۲ — Cluster Computing در برابر RDBMS و Hadoop

محدودیت RDBMS

مقیاس‌پذیری گران، سرعت، تحمل خرابی، پردازش پیشرفته، پرس‌وجو روی داده خیلی حجیم.

ویژگی‌های Cluster (حفظ کن)

1. High Availability
2. Fault Tolerance
3. Data Replication
4. Load Balancing

5. Automated Failover
6. Backup & Recovery
7. Monitoring & Alerting
8. Scalability افقی

موضوع	RDBMS	Cluster / Hadoop
سخت‌افزار	سرور گران	commodity ارزان
مقیاس	عمودی	افقی
داده	ساخت‌یافته	ساخت‌یافته + بدون ساختار
تحمل خرابی	سخت‌افزاری	نرم‌افزاری (replication)
ایده	داده → کد	**کد → داده**

مثال پاسخ‌نویسی سوال «تفاوت Cluster و RDBMS»

۱) بنویس RDBMS برای داده ساخت‌یافته و تراکنش ACID خوب است ولی در پتابایت گران و کند می‌شود.
۲) بنویس Cluster با سرورهای ارزان، replication و failover، مقیاس افقی دارد.
۳) جمله طلایی: در Hadoop/Cluster کد را پیش داده می‌بریم، نه اینکه همه داده را روی یک سرور جمع کنیم.

Hadoop

- دو ویژگی: Scalable + Fault Tolerant
- HDFS + MapReduce
- NameNode / DataNode / JobTracker / TaskTracker
- معمولاً هر chunk سه کپی (replication=3)

SQL در برابر NoSQL: SQL اسکیما ثابت و ACID؛ NoSQL اسکیما انعطاف‌پذیر و مقیاس افقی.

§۳ — MapReduce

تعریف: سبک پردازش موازی داده حجیم. کاربر فقط Map و Reduce می‌نویسد؛ سیستم بقیه کار (موازی‌سازی، خرابی، هماهنگی) را انجام می‌دهد.

سه مرحله به زبان ساده

1. **Map:** هر تکه داده را بخوان و جفت‌های (کلید، مقدار) بساز
2. **Shuffle:** همه مقدارهایی که کلید یکسان دارند را کنار هم بگذار
3. **Reduce:** برای هر کلید، مقدارها را با هم ترکیب کن (جمع، ماکزیمم، ...)

Map: چند عنصر ورودی → (key, value)
Shuffle: (k,v1), (k,v2), ... → (k, [v1,v2,...])
Reduce: (k, [v1,v2,...]) → (k, نتیجه)

مثال خیلی ساده — شمارش کلمه (Word Count)

متن‌ها: «big data» و «big data systems»

Map از متن اول: (big,1) ، (data,1)

Map از متن دوم: (big,1) ، (data,1) ، (systems,1)

Shuffle: big→[1,1] ، data→[1,1] ، systems→[1]

```
Reduce:
big → 1+1 = 2
data → 1+1 = 2
systems → 1
```

پاسخ نهایی: big=2 ، data=2 ، systems=1

Combiner: اگر عمل Reduce جابجایی‌پذیر و انجمنی باشد (مثل جمع)، می‌توان قبل از ارسال روی شبکه، روی همان نود Map یک Reduce محلی زد تا ترافیک کم شود.

خرابی	چه می‌شود؟
Master	معمولاً کل Job از نو
Map Worker	Mapها دوباره روی نود سالم
Reduce Worker	Reduce دوباره اجرا می‌شود

Workflow: به‌جای فقط Map→Reduce، یک گراف DAG از توابع (Clustera، Hyracks).

Pregel: سیستم محاسبات گرافی بازگشتی با SuperStep و Checkpoint برای تحمل خرابی.

Recursive MR: مثل PageRank: $v^{(t+1)} = M \times v^{(t)}$ تا همگرایی.

§۴ — جبر رابطه‌ای با MapReduce

عملیات	ایده MapReduce
Selection	فقط تاپل‌هایی که شرط دارند را emit کن
Projection	صفات را کم کن؛ Reduce تکراری‌ها را حذف کند
Union	همه تاپل‌ها را بفرست؛ تکراری یکی شود
Intersection	تاپلی که از هر دو رابطه آمده خروجی شود
Difference R−S	تاپلی که فقط در R است
Join	کلید = صفت مشترک؛ ترکیب سمت چپ و راست
Group+Agg	کلید=گروه؛ Reduce=SUM/COUNT/...

مثال Join ساده — قدم‌به‌قدم

R(A,B): (علی, تهران) ، (مینا, اصفهان)

S(B,C): (تهران, ایران) ، (اصفهان, ایران)

می‌خواهیم Join روی B (شهر):

```
Map:
  از R: کلید=تهران → (R, علی)
  از R: کلید=اصفهان → (R, مینا)
  از S: کلید=تهران → (S, ایران)
  از S: کلید=اصفهان → (S, ایران)

Reduce برای تهران: (علی, تهران, ایران)
Reduce برای اصفهان: (مینا, اصفهان, ایران)
```

پاسخ نهایی: (علی,تهران,ایران) و (مینا,اصفهان,ایران)

Multi-way Join (خلاصه): Join همزمان چند رابطه (مثل R⋈S⋈T). هزینه ارتباطی مهم است؛ پارامترهای Reducer Size و Replication Rate کیفیت موازی‌سازی را تعیین می‌کنند.

§۵ — ضرب ماتریس با MapReduce

ماتریس × بردار (سوال امتحانی)

هدف: هر سطر ماتریس را در بردار ضرب کنی و یک عدد برای آن سطر بگیری.

$$u_i = m_{i1}\cdot v_1 + m_{i2}\cdot v_2 + \dots + m_{in}\cdot v_n$$

که در آن:

کلید خروجی Map = شماره سطر i

مقدار = حاصل‌ضرب یک عنصر ماتریس در عنصر متناظر بردار

مثال کامل — طوری که انگار اولین بار می‌بینی

ماتریس ۳×۳ و بردار ۳تایی:

```
M =  1  0  5        V =  2
     7  3  0             4
     1  3  5             6
```

گام Map: برای هر خانه ماتریس یک جفت بساز. کلید = شماره سطر.

```
سطر1: (2=2×1 ,1) ، (0=4×0 ,1) ، (30=6×5 ,1)
سطر2: (14=2×7 ,2) ،(12=4×3 ,2) ،(0=6×0 ,2)
سطر3: (2=2×1 ,3) ، (12=4×3 ,3) ،(30=6×5 ,3)
```

گام Shuffle: همه مقادیر هر سطر کنار هم:

```
کلید 1 → [30 ,0 ,2]
کلید 2 → [0 ,12 ,14]
کلید 3 → [30 ,12 ,2]
```

گام Reduce: جمع کن:

```
u1 = 2+0+30 = 32
u2 = 14+12+0 = 26
u3 = 2+12+30 = 44
```

پاسخ نهایی: U = [32, 26, 44]

ماتریس × ماتریس (دو Job)

Job1: روی j جوین کن و ضرب‌های جزئی بساز. Job2: برای هر (i,k) جمع بزن.

$$p_{ik} = \Sigma_j \left(m_{ij} \times n_{jk} \right)$$

مثال نتیجه

```
اگر N = |2 0|
        |4 8|
        |6 2|

آنگاه P = M×N = |32 10|
                |26 24|
                |44 34|
```

چک سطر اول ستون اول: 32 = 30+0+2 = 6×5 + 4×0 + 2×1

§۶ — Finding Similar Items (Shingle، MinHash، LSH)

Document → Shingles → Sets → MinHash → Signatures → LSH → Candidates → Exact check

Shingle

پنجرهٔ kتایی روی سند را سر می‌دهیم و هر پنجره یک shingle است.

مثال Shingle برای مبتدی

سند: abcab و k=2 (یعنی هر ۲ حرف پشت‌سرهم)

```
موقعیت‌ها:
ab | bc | ca | ab

مجموعه (بدون تکرار): {ab, bc, ca}
```

اگر دو سند مجموعه shingle خیلی شبیه داشته باشند، near-duplicate هستند.

Jaccard

$$J(A,B) = |A \cap B| / |A \cup B|$$

مثال Jaccard

A = {سیب، پرتقال، موز} و B = {موز، انگور}

```
اشتراک = {موز}          → اندازه 1
اجتماع  = {سیب، پرتقال، موز، انگور} → اندازه 4
J = 1/4 = 0.25
فاصله = 1 - 0.25 = 0.75
```

پاسخ نهایی: J = 0.25

MinHash

ایده: ترتیب تصادفی به عناصر بده؛ برای هر مجموعه، **اولین** عنصری که در آن ترتیب می‌بینی را به‌عنوان hash بردار.

Pr[هش دو مجموعه برابر شود] = Jaccard همان دو مجموعه

مثال MinHash با یک permutation

جهان عناصر: {1,2,3,4,5}

A = {1,3,4} و B = {2,3,4,5}

اول Jaccard واقعی:

```
اشتراک={3,4} → 2
اجتماع={1,2,3,4,5} → 5
J = 2/5 = 0.4
```

حالا یک ترتیب تصادفی فرضی: 3,1,5,2,4

```
در این ترتیب: 3 A اولین عنصر
در این ترتیب: 3 B اولین عنصر
پس h(A)=h(B) — برابر شدند
```

اگر خیلی permutation بگیری، نسبت دفعاتی که برابر می‌شوند ≈ 0.4 می‌شود.

LSH

امضا را به چند نوار (band) تقسیم کن. اگر دو سند حتی در **یک** نوار کاملاً یکسان بودند، آن‌ها را candidate حساب کن.

$$P = 1 - (1 - s^r)^b$$

که در آن:

s=شباهت، r=تعداد سطر هر band، b=تعداد band

مثال LSH با عدد

r=5 ، b=20 ، شباهت s=0.8

```
s^r = 0.8^5 ≈ 0.328
1 - s^r ≈ 0.672
(0.672)^20 ≈ 0.00035
P = 1 - 0.00035 ≈ 0.99965
```

یعنی حدود ۹۹٫۹۷٪ احتمال دارد جفت واقعاً مشابه را پیدا کنیم.

پاسخ نهایی: P ≈ 0.99965

PCY (بهبود یافتن جفت‌های پرتکرار)

در Pass1 علاوه بر شمارش اقلام، جفت‌ها را به bucketها hash کن. در Pass2 فقط جفت‌هایی را بشمار که هر دو قلم frequent باشند **و** bucketشان هم frequent باشد.

§۷ — فاصله‌ها و تشابه تاپل‌ها

چهار شرط Metric: نامنفی، صفر فقط برای خود شیء، تقارن، نامساوی مثلثی.

اقلیدسی: $\sqrt{(x1-y1)^2 + (x2-y2)^2 + \dots}$

منهتن: $|x1-y1| + |x2-y2| + \dots$

L_∞: بزرگ‌ترین $|xi-yi|$

مثال فاصله‌ها روی یک جفت نقطه

نقطه X=(1,2) و Y=(4,6)

```
اقلیدسی: √(2^(6-2)+2^(4-1))=√(16+9)=25=5
منهتن:   |6-2|+|4-1|=3+4=7
L∞:      max(3,4)=4
```

صفت اسمی: $d = (p - m) / p$

که در آن:

p = تعداد کل صفات

m = تعداد صفاتی که مقدارشان یکی است

مثال صفات اسمی

شخص۱: (مرد، تهران، لیسانس)

شخص۲: (زن، تهران، لیسانس)

```
p=3 صفات
m=2 تطابق (شهر و مدرک)
d = (3−2)/3 = 1/3 ≈ 0.33
```

پاسخ نهایی: ۱/۳

دودویی: q=هر دو ۱، r=(۱٬۰)، s=(۰٬۱)، t=هر دو ۰.

متقارن: d = (r+s)/(q+r+s+t)

نامتقارن: d = (r+s)/(q+r+s) ← صفر-صفر مهم نیست

Jaccard: J = q/(q+r+s)

Cosine: برای متن خوب است چون صفرهای مشترک را زیاد جریمه نمی‌کند.

Hamming: تعداد خانه‌های متفاوت دو رشته هم‌طول.

Edit distance: حداقل درج/حذف برای تبدیل یک رشته به دیگری = |X|+|Y|−2×|LCS|

§۸ — Maximal ، Closed ، Apriori ، Frequent Itemsets

support(X) = (تعداد تراکنش‌های شامل X) / (کل تراکنش‌ها)

Frequent یعنی: support(X) ≥ minsup

support(A⇒B) = support(A∪B)

confidence(A⇒B) = support(A∪B) / support(A)

مثال قانون انجمنی

۱۰۰ سبد خرید. نان در ۴۰ سبد، نان+شیر در ۳۰ سبد.

```
قانون: نان ⇒ شیر
support = 30/100 = 0.3
confidence = 30/40 = 0.75 = 75%
```

یعنی ۷۵٪ کسانی که نان گرفته‌اند شیر هم گرفته‌اند.

نوع	تعریف ساده
Closed	اگر چیزی به آن اضافه کنی، support کم می‌شود (یا frequent نمی‌ماند با همان support)
Maximal	اگر چیزی به آن اضافه کنی، دیگر frequent نیست

M ⊆ C ⊆ F

مثال Closed و Maximal

تراکنش‌ها: {A,B,C} ، {A,B} ، {A,B,C} ، {A,C} و minsup count=2

```
supportها:
A=4, B=3, C=3
AB=3, AC=3, BC=2
ABC=2
```

Closedها: A (چون هیچ ابرمجموعه‌ای support=4 ندارد)، AB، AC، ABC

B closed نیست چون AB همان support=3 را دارد.

Maximal فقط: ABC (چون ابرمجموعه frequent ندارد).

اصل Apriori: اگر مجموعه‌ای frequent باشد، همه زیرمجموعه‌هایش هم frequentاند. پس کاندیدهای بزرگ را فقط از frequentهای کوچک‌تر می‌سازیم.

CHARM: الگوریتم استخراج Closed با اشتراک tidsetها.

§۹ — خوشه‌بندی

هدف: اعضای یک خوشه شبیه هم؛ خوشه‌های مختلف دور از هم. یادگیری بدون سرپرست است.

- Partitioning: K-Means، K-Medoids، PAM، CLARA، CLARANS
- Hierarchical: AGNES (پایین به بالا)، DIANA (بالا به پایین)
- Density: DBSCAN، OPTICS، DENCLUE
- Grid: STING، CLIQUE

مثال K-Means — یک تکرار کامل

نقاط: A(1,1)، B(1,2)، C(4,4)، D(5,4) و k=2

مراکز اولیه: μ1=A(1,1) و μ2=C(4,4)

گام تخصیص: هر نقطه به نزدیک‌ترین مرکز

```
A→μ1 ، B→μ1 (نزدیک‌تر به A)
C→μ2 ، D→μ2 (نزدیک‌تر به C)
پس C1={A,B} و C2={C,D}
```

گام به‌روزرسانی مرکز: میانگین بگیر

```
μ1 = ((1+1)/2, (1+2)/2) = (1, 1.5)
μ2 = ((4+5)/2, (4+4)/2) = (4.5, 4)
```

پاسخ نهایی: مراکز جدید (1 , 1.5) و (4.5 , 4)

K-Medoids/PAM: نماینده خوشه یک نقطه واقعی است → مقاوم‌تر به پرت. CLARA روی نمونه؛ CLARANS جستجوی تصادفی.

Linkage: Single=کمینه فاصله، Complete=بیشینه، Average=میانگین، Centroid=فاصله مراکز.

DBSCAN: اگر در شعاع Eps حداقل MinPts نقطه باشد → Core

نیست Core ولی خودش Core کنار = Border

هیچ‌کدام = Noise

BFR: برای داده بزرگ اقلیدسی؛ از هر خوشه فقط N و SUM و SUMSQ را نگه می‌دارد.

$$centroid_i = SUM_i / N$$

$$variance_i = SUMSQ_i/N - (SUM_i/N)^2$$

CURE: چند نقطه نماینده برای خوشه‌های غیرکروی.

§۱۰ — حل کامل نمونه سوالات استاد

امتحان نمونه: ۹۰ دقیقه — ۱۰ نمره توصیفی

سوال ۱ — تعاریف

الف) Min-hashing: ساخت امضای کوتاه از مجموعه طوری که احتمال برابر شدن hash = Jaccard باشد.

$$Pr[h(C1)=h(C2)] = J(C1,C2)$$

ب) Collaborative Filtering: توصیه بر اساس کاربران/آیتم‌های مشابه.

ج) Jaccard:

$$J(A,B) = |A \cap B| / |A \cup B|$$

د) Shingle: دنباله k توکن متوالی در سند.

سوال ۲ — Cluster در برابر RDBMS

RDBMS: مقیاس عمودی، ACID، داده ساخت‌یافته، در حجم خیلی بالا گران/کند.
Cluster: مقیاس افقی، commodity، replication/failover، انتقال کد به داده.

سوال ۳ — Matrix×Vector با MapReduce

$$Map: (i, m_{ij} \times v_j)$$

$$Reduce: u_i = \text{مجموع مقادیر کلید } i$$

مثال عددی در §۵.

سوال ۴ — TF-IDF

حل سوال استاد

```
TF=10/100=0.1
IDF=log2(1024/128)=log2(8)=3
TF-IDF=0.1×3=0.3
```

پاسخ نهایی: ۰٫۳

سوال ۵ — Apriori با MST=0.3

TID	اقلام	TID	اقلام
1	a,b	5	a,b,c
2	b,c	6	d,f
3	a,b,c	7	c,d,e,f
4	d,e,f	8	a,b,c,d,e

حل Apriori قدم‌به‌قدم

تعداد تراکنش = 8 ، minsup=0.3 → حداقل شمارش باید ≥ 3 باشد (چون 0.25=2/8 کمتر از 0.3 است).

Pass1: شمارش تکی

```
a=4, b=5, c=5, d=4, e=3, f=3
3≤ همه → L1={a,b,c,d,e,f}
```

Pass2: فقط جفت‌هایی از L1 را بشمار

```
ab=4 ✓  ac=3 ✓  bc=4 ✓
de=3 ✓  df=3 ✓
cd=2 ✗  ce=2 ✗  ef=2 ✗
L2={ab,ac,bc,de,df}
```

Pass3: از L2 فقط abc قابل ساخت است (چون ab و ac و bc همه هستند). def ساخته نمی‌شود چون ef نیست.

```
abc آمده 3،5،8 در تراکنش‌های → count=3 ✓
L3={abc}
```

Pass4 کاندیدی ندارد → توقف.

پاسخ نهایی: L1 همه اقلام؛ L2={ab,ac,bc,de,df}؛ L3={abc}

چک‌لیست ۳۰ثانیه‌ای: تعریف؟ فرمول؟ مثال عددی؟ تفاوت با مفهوم نزدیک؟

موضوعات فهرست‌شده در overview درس مثل Data Streams / Online Advertising اگر در لکچرهای همین ترم اسلاید کامل نداشتند، فقط در فهرست کلی آمده‌اند؛ تمرکز امتحان نمونه روی Cluster/RDBMS، TF-IDF، MapReduce، Similar Items و Apriori است.