The code and the top 10 recommendations for every user are available in the attached files `HW1.html` and `top10_recommendations.txt`.

**Algorithm.** *Map-Reduce.*
From the given input file, formatted as `ID<tab>f1,f2,...fn`, the algorithm generates network

$$\begin{aligned}&(\mathtt{ID},\ [\mathtt{f}_1,\mathtt{f}_2,\ldots\mathtt{f}_n])\\ &\Rightarrow ((\mathtt{ID}_1,\mathtt{ID}_2),\mathtt{i}),\end{aligned}$$

where $\mathtt{i} = 0$ if `ID1` and `ID2` are friends and 1 if they have a mutual friend and group such lines by key. Now given (($\mathtt{ID}_1$, $\mathtt{ID}_2$), [$i_1$, $i_2$, ..., $i_n$]) the algorithm removes all pairs ($\mathtt{ID}_1$, $\mathtt{ID}_2$), for which the condition $0 \in [i_1, i_2, ..., i_n]$ holds, since that means they are already friends. When all such pairs are removed, the algorithm counts the number of mutual friends, which is the sum of all elements in list [$i_1$, $i_2$, ..., $i_n$]. Then, it reorganises the structure to ($\mathtt{ID}_1$, ($\mathtt{ID}_2$, `count`)), group by key and outputs the file formatted as `ID<tab>recomID1,recomID2,...recomID10`, sorted in the decreasing order of IDs of recommended friends.

**Recommendations.**

- `924:  439, 2409, 6995, 11860, 15416, 43748, 45881`
- `8941:  8943, 8944, 8940`
- `8942:  8939, 8940, 8943, 8944`
- `9019:  9022, 317, 9023`
- `9020:  9021, 9016, 9017, 9022, 317, 9023`
- `9021:  9020, 9016, 9017, 9022, 317, 9023`
- `9022:  9019, 9020, 9021, 317, 9016, 9017, 9023`
- `9990:  13134, 13478, 13877, 34299, 34485, 34642, 37941`
- `9992:  9987, 9989, 35667, 9991`
- `9993:  9991, 13134, 13478, 13877, 34299, 34485, 34642, 37941`

*Confidence.*
Confidence is defined as

$$\text{conf}(A \to B) = \Pr(B|A) = \frac{\Pr(A \cap B)}{\Pr(A)} = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

A drawback of using confidence is that it ignores $\Pr(B)$, therefore it might misrespect the importance of an association. In other words, it only accounts for how popular the $A$ is but not $B$. So the main problem here is that a confidence is high even though $A$ may have no influence on $B$ – if $B$ is very popular in general, there will be a higher chance that a transaction containing $A$ will also contain $B$ so the confidence rate will get higher. In order to take into account the popularity of both items $A$ and $B$, such that $A$ somehow affect $B$, we define *lift* and *conviction*. The presence of $\Pr(B)$ is visible from the equations themselves, however, the more precise correlation between $A$ and $B$ is described in the following paragraphs.

*Lift.*
Lift is defined as

$$\text{lift}(A \to B) = \frac{\text{conf}(A \to B)}{\text{S}(B)},$$

where $\text{S}(B) = \frac{\text{Support}(B)}{N}$ and $N$ = total number of transactions (baskets), and it describes the "power" of correlation between $A$ and $B$ (how $A$ affects $B$) as follows:

- $A$ has no influence on $B$: $\text{lift}(A \to B) = 1$.
- $A$ has positive influence on $B$ ($B$ is more likely to occur): $\text{lift}(A \to B) > 1$
- $A$ has negative influence on $B$ ($B$ is less likely to occur): $\text{lift}(A \to B) < 1$

*Conviction.*
The conviction is defined as

$$\text{conv}(A \to B) = \frac{1 - \text{S}(B)}{1 - \text{conf}(A \to B)}$$

and it describes how $A$ and $B$ are correlated as follows:

- $A$ has no influence on $B$: $\text{conv}(A \to B) = 1$
- $A$ has positive influence on $B$ ($B$ is more likely to occur):

$$\begin{aligned} \text{lift}(A \to B) &> 1 \\ \text{conf}(A \to B) &> \text{S}(B) \\ 1 - \text{S}(B) &> 1 - \text{conf}(A \to B) \\ \Rightarrow \text{conv}(A \to B) &> 1 \end{aligned}$$

- $A$ has negative influence on $B$ ($B$ is less likely to occur):

$$\begin{aligned} \text{lift}(A \to B) &< 1 \\ \text{conf}(A \to B) &< \text{S}(B) \\ 1 - \text{S}(B) &< 1 - \text{conf}(A \to B) \\ \Rightarrow \text{conv}(A \to B) &< 1 \end{aligned}$$

*Confidence.*
Clearly, the confidence in not symmetrical. Usually, $\text{conf}(A \to B)$ is different from $\text{conf}(B \to A)$, since $\Pr(A) \neq \Pr(B)$ in general.
A counterexample to see that is the one provided on slides [1].

$$\text{conf}(\{beer\} \to \{diapers\}) = 1.0$$
$$\text{conf}(\{diapers\} \to \{beer\}) = 0.67$$

$$\Rightarrow \text{conf}(\{beer\} \to \{diapers\}) \neq \text{conf}(\{diapers\} \to \{beer\})$$

| tid | vsebina košarice |
|---|---|
| 1 | kruh, mleko |
| 2 | kruh, plenice, union |
| 3 | mleko, plenice, union |
| 4 | kruh, mleko, plenice, union |
| 5 | kruh, mleko, plenice |
| 6 | mleko, plenice |
| 7 | plenice |
| 8 | mleko, union, plenice |
| 9 | plenice, union |
| 10 | mleko, plenice, union |

Figure 1: Baskets' content.

*Lift.*
Intuitively, the lift is symmetrical, since it measures the correlation between $A$ and $B$.
Formally, the proof goes as follows

$$\begin{aligned}
\text{lift}(A \to B) &= \frac{\text{conf}(A \to B)}{\text{S}(B)} \\
&= \frac{\text{Support}(A \cup B)}{\text{Support}(A) \cdot \frac{\text{Support}(B)}{N}} \\
&= \frac{\text{Support}(B \cup A)}{\text{Support}(B) \cdot \frac{\text{Support}(A)}{N}} \\
&= \frac{\text{conf}(B \to A)}{\text{S}(A)} \\
&= \text{lift}(B \to A)
\end{aligned}$$

*Conviction.*
Intuitively, the conviction is not symmetrical. To provide the counterexample, we calculate the conviction from the same example (on picture 1) as before (from slides [1]).

$$\begin{aligned}
\mathrm{conv}(\{beer\} \to \{diapers\}) &= \frac{1 - \mathrm{S}(\{diapers\})}{1 - \mathrm{conf}(\{beer\} \to \{diapers\})} \\
&= \frac{1 - 0.9}{1 - 1} := +\infty
\end{aligned}$$

$$\begin{aligned}
\mathrm{conv}(\{diapers\} \to \{beer\}) &= \frac{1 - \mathrm{S}(\{beer\})}{1 - \mathrm{conf}(\{diapers\} \to \{beer\})} \\
&= \frac{1 - 0.6}{1 - 0.67} = 1.2
\end{aligned}$$

$$\Rightarrow \mathrm{conv}(\{beer\} \to \{diapers\}) \neq \mathrm{conv}(\{diapers\} \to \{beer\})$$

Since the perfect implications are defined as rules that hold 100% or equivalently, that the associated conditional probability is 1, the contition

$$\begin{aligned}\Pr(B|A) &= 1\\ \frac{\Pr(A\cap B)}{\Pr(A)} &= 1\\ \Pr(A\cap B) &= \Pr(A)\end{aligned}$$

holds, meaning that $B$ occurs every time $A$ occurs.

*Confidence.*
First note that $\mathrm{conf}(A \to B) \in [0, 1]$. We have the condition $\Pr(A\cap B) = \Pr(A)$ and, clearly, confidence is 1, meaining that $B$ is contained in all baskets containing $A$, hence confidence is desirable.

*Lift.*
First note that the maximal value of $\mathrm{conv}(A \to B)$ is $+\infty$. Suppose lift is desirable. We have the condition $\Pr(A \cap B) = \Pr(A)$, so $\mathrm{conf}(A \to B) = 1$ and $\mathrm{lift}(A \to B) = \frac{N}{\mathrm{Support}(B)}$. Now let us look at $\mathrm{lift}(C \to D)$, where $A \to B$ and $C \to D$ are differend conditions. It follows that $\mathrm{lift}(C \to D) = \frac{N}{\mathrm{Support}(D)}$ and $\mathrm{lift}(A \to B) = \mathrm{lift}(C \to D)$ iff. $\mathrm{Support}(B) = \mathrm{Support}(D)$, which generally is not the case. We confirm that by providing a counterexample.
We have the following baskets (s. t. $\Pr(A \cap B) = \Pr(A)$ and $\Pr(C \cap D) = \Pr(C)$):
{A, B}, {A, B}, {C, D}, {E, F}.
Since $\mathrm{Support}(B) = 2 \neq \mathrm{Support}(D) = 1$, then $\mathrm{lift}(A \to B) \neq \mathrm{lift}(C \to D)$. By providing this example we have come to contradiction, therefore lift is not desirable.

*Conviction.*
First note that the maximal value of $\mathrm{conv}(A \to B)$ is $+\infty$. Again, we have the condition $\Pr(A \cap B) = \Pr(A)$, so $\mathrm{conf}(A \to B) = 1$, so $\mathrm{conv}(A \to B) = \frac{1-\mathrm{S}(B)}{0} := +\infty$, hence conviction is desirable.

The code is available in the attached file `HW1.html`.

Top 5 rules with confidence scores for itemsets of size 2 in decreasing order.

- {'DAI93865'} $\rightarrow$ {'FRO40251'} : 1.0
- {'GRO85051'} $\rightarrow$ {'FRO40251'} : 0.999176276771005
- {'GRO38636'} $\rightarrow$ {'FRO40251'} : 0.9906542056074766
- {'ELE12951'} $\rightarrow$ {'FRO40251'} : 0.9905660377358491
- {'DAI88079'} $\rightarrow$ {'FRO40251'} : 0.9867256637168141

The code is available in the attached file `HW1.html`.

Top 5 rules with confidence scores for itemsets of size 3 in decreasing order.

- {'DAI23334', 'ELE92920'} $\rightarrow$ {'DAI62779'} : 1.0
- {'DAI31081', 'GRO85051'} $\rightarrow$ {'FRO40251'} : 1.0
- {'DAI55911', 'GRO85051'} $\rightarrow$ {'FRO40251'} : 1.0
- {'DAI62779', 'DAI88079'} $\rightarrow$ {'FRO40251'} : 1.0
- {'DAI75645', 'GRO85051'} $\rightarrow$ {'FRO40251'} : 1.0

We want to calculate the probability that 1 is *not* contained in the $k$-selected rows.

$$\mathrm{P} = \Pr(1 \text{ not contained in the } k\text{-selected rows}) = \frac{\binom{n-k}{m}}{\binom{n}{m}}$$

where $\binom{n}{m}$ is the number of columns with $m$ 1's (and $n-m$ 0's) and $\binom{n-k}{m}$ counts only the columns with no 1 among $k$-selected rows.

$$\begin{aligned}
\mathrm{P} &= \frac{\frac{(n-k)!}{m!\ (n-k-m)!}}{\frac{n!}{m!\ (n-m)!}} \\
&= \frac{(n-k)!}{m!\ (n-k-m)!} \cdot \frac{m!\ (n-m)!}{n!} \\
&= \frac{(n-k)!}{(n-k-m)!} \cdot \frac{(n-m)!}{n!} \\
&= \frac{(n-k)(n-k-1)(n-k-2)\cdots 1}{(n-k-m)(n-k-m-1)(n-k-m-2)\cdots 1} \cdot \frac{(n-m)(n-m-1)(n-m-2)\cdots 1}{n(n-1)(n-2)\cdots 1} \\
&= \frac{(n-k)(n-k-1)\cdots(n-k-m+1)}{n(n-1)\cdots(n-m+1)} \\
&= \underbrace{\left(\frac{n-k}{n}\right) \cdot \left(\frac{n-k-1}{n-1}\right) \cdot\ \cdots\ \cdot \left(\frac{n-k-m+1}{n-m+1}\right)}_{m} \leq \left(\frac{n-k}{n}\right)^m,
\end{aligned}$$

since each of the $m$ factors is at most $\frac{n-k}{m}$.
The probability of getting "don't know" with given conditions really is at most $\left(\frac{n-k}{n}\right)^m$.

Now we want the probability of ”don’t know” to be at most $e^{-10}$ ($\left(\frac{n-k}{n}\right)^m \leq e^{-10}$). We want to find the apptoximation to the smallest value of $k$ that will ensure the condition (assuming $n$ and $m$ are both large and $k << n$).
Following the hint (2):

$$\left(\frac{n-k}{n}\right)^m \leq e^{-10}$$

$$\left(\left(1-\frac{k}{n}\right)^{m\cdot\frac{n}{k}}\right)^{\frac{k}{n}} \leq e^{-10}$$

$$\left(\left(1-\frac{k}{n}\right)^{\frac{n}{k}}\right)^{m\cdot\frac{k}{n}} \leq e^{-10}$$

Since $n$ is much larger than $k$, $\left(1-\frac{k}{n}\right)^{\frac{n}{k}} \approx \frac{1}{e}$. Thus it follows

$$\left(\frac{1}{e}\right)^{m\frac{k}{n}} \leq e^{-10}$$

$$e^{-m\cdot\frac{k}{n}} \leq e^{-10}$$

$$\Rightarrow -m\cdot\frac{k}{n} \leq -10$$

$$\Rightarrow m\cdot\frac{k}{n} \geq 10$$

$$\Rightarrow k \geq 10\cdot\frac{n}{m}$$

The approximation for smallest value of $k$ ensuring the upper probability bound is $k = 10\cdot\frac{n}{m}$.

Given the matrix

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

where

$$S_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ and } S_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

we claim that the probability (over cyclic permutations only) that minhash values of $S1$ and $S2$ agree is not the same as their Jaccard similarity.
Jacard similarity is defined as $J(S1, S2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$ where we are only interested in the appearance of 1s. So $|S_1 \cap S_2| = 1$, $|S_1 \cup S_2| = 2$ and $J(S_1, S_2) = \frac{1}{2}$.
The cycle can start in any row, but only if it starts in the first, second and last row, the minhash values from $S_1$ and $S_2$ agree, while in the third row the values differ.
Therefore, described probability is $\frac{3}{4}$ which is not the same as Jacard similarity.

Let us consider LSH family $\mathcal{H}$ of hash functions that is $(\lambda, c\lambda, p_1, p_2)$-sensitive for the distance measure $d(\cdot,\cdot)$. Let $\mathcal{G} = \mathcal{H}^k = \{g = (h1, \ldots, h_k) | h_1 \in \mathcal{H}, \forall\ 1 \leq i \leq k\}$, where $k = \log_{1/p_2}(n)$.
(Note: $\mathcal{G}$ is then $(\lambda, c\lambda, p_1^k, p_2^k)$- sensitive).

Let $W_j = \{x \in \mathcal{A} | g_j(x) = g_j(z)\}$, where $1 \leq j \leq L$ be the set of data points $x$ mapping to the same value as the query point $z$ by the hash function $g_j$. Define $T = \{x \in \mathcal{A} | d(x,z) > c\lambda\}$.
Following the hint and considering the linearity of the expected value:

$$\Pr\left[\sum_{j=1}^{L} |T \cap W_j| \geq 3L\right] \leq \frac{1}{3L} \cdot \mathrm{E}\left[\sum_{j=1}^{L} |T \cap W_j|\right] = \frac{1}{3L} \cdot \sum_{j=1}^{L} \mathrm{E}\left[|T \cap W_j|\right]$$

To show that

$$\frac{1}{3L} \cdot \sum_{j=1}^{L} \mathrm{E}\left[|T \cap W_j|\right] \leq \frac{1}{3}$$

it is enough to show that $E\left[|T \cap W_j|\right] \leq 1$ for $1 \leq j \leq L$.
For each $x_i \in \mathcal{A}$ we define an indicator $\mathbb{I}_i$ ($\mathbb{I} = \mathbb{I}_1 + \ldots + \mathbb{I}_n$) such that $\mathbb{I}_i = 1$ if $x_i \in T \cap W_j$ and $\mathbb{I}_i = 0$ otherwise.

$$\mathrm{E}\left[|T \cap W_j|\right] = \mathrm{E}[\mathbb{I}] = \mathrm{E}\left[\sum_{i=1}^{n} \mathbb{I}_i\right] = \sum_{i=1}^{n} \mathrm{E}[\mathbb{I}_i] \leq np_2^k,$$

and following the equality $k = \log_{1/p_2} n$, we get

$$\left(\frac{1}{p_2}\right)^k = n \ \Rightarrow\ np_2^k = 1$$

and

$$\mathrm{E}\left[|T \cap W_j|\right] \leq 1$$

$$\Rightarrow \Pr\left[\sum_{j=1}^{L} |T \cap W_j| \geq 3L\right] \leq \frac{1}{3}$$

Let $x* \in \mathcal{A}$ be a point such that $d(x^*, z) \leq \lambda$. We want to prove that condition

$$\Pr[\forall\ 1 \leq j \leq L,\ g_j(x^*) \neq g_j(z)] < \frac{1}{e} \tag{1}$$

holds.
Since $\mathcal{G} = \mathcal{H}^k$, $\mathcal{G}$ is $(\lambda, c\lambda, p_1^k, p_2^k)$- sensitive for the distance $d$, hence the condition

$$\Pr[g_j(x^*) = g_j(z)] \geq p_1^k$$

holds for $d(x^*, z) \leq \lambda$.
Then,

$$\Pr[g_j(x^*) \neq g_j(z)] \leq 1 - p_1^k < e^{-p_1^k} = \left(\frac{1}{e}\right)^{p_1^k}$$

and

$$\Pr[\forall\ 1 \leq j \leq L,\ g_j(x^*) \neq g_j(z)] = \prod_{j=1}^{L} \Pr[g_j(x^*) \neq g_j(z)] \ \leq \ (1 - p_1^k)^L \ < \ \left(\frac{1}{e}\right)^{L \cdot p_1^k},$$

where the inequality $1 - p_1^k < e^{-p_1^k}$ holds for $p_1^k \neq 0$, which is obviously true.
To satisfy the condition [1], it is sufficient to show that $L \geq \frac{1}{p_1^k}$, however, the equality $L = \frac{1}{p_1^k}$ can be proved and the proof goes as follows.

$$L = \frac{1}{p_1^k}$$

$$\log L = \log \frac{1}{p_1^k}$$

$$\log L = -k \log p_1$$

$$\log\left(n^{\frac{\log(1/p_1)}{\log(1/p_2)}}\right) = -\log_{1/p_2}(n) \log p_1$$

$$\frac{\log(1/p_1)}{\log(1/p_2)} \log n = -\frac{\log n}{\log(1/p_2)} \log p_1$$

$$\frac{\log(1/p_1)}{\log(1/p_2)} \log n = \frac{\log(1/p_1)}{\log(1/p_2)} \log n$$

which is obviously true.
We can now conclude that

$$\Pr[\forall\ 1 \leq j \leq L,\ g_j(x^*) \neq g_j(z)] < \frac{1}{e}$$

holds.

We want to conclude that with probability greater than some fixed constant the reported point is an actual $(c, \lambda)$-ANN.

Let $\mathcal{S} = \{x \in \mathcal{A} | d(x, z) \leq c\lambda\}$, where $z$ is the query point. Let $x^*$ be the reported point by the procedure. Reported point $x^*$ is a $(c, \lambda)$-ANN if $x^* \in \mathcal{S}$.
Remember that $W_j = \{c \in \mathcal{A} | g_j(x) = g_j(z)\}$ $(1 \leq j \leq L)$ and $T = \{x \in \mathcal{A} | d(x, z) > c\lambda\}$.

To prove the desired, we will observe different cases.

- If $g_j(x) \neq g_j(z)$ $\forall$ $1 \leq j \leq L$ $\forall x \in \mathcal{S}$, then the reported point $x^* \notin \mathcal{S}$, thus $x^*$ is not a $(c, \lambda)$-ANN.
- If there exists at least one $j$ and $x \in \mathcal{S}$ s. t. $g_j(x) = g_j(z)$ and
  - $\sum_{j=1}^{L} |T \cap W_j| < 3L$ then the procedure always reports $x^* \in \mathcal{S}$, thus $x^*$ is a $(c, \lambda)$-ANN.
  - $\sum_{j=1}^{L} |T \cap W_j| \geq 3L$ then the procedure may or may not report $x^* \in \mathcal{S}$.

Now we observe the reverse event, bounding probability with inequalities from qeustions 4(a) and 4(b):

$$\Pr[d(x^*, z) \leq c\lambda] \geq 1 - \left( \underbrace{\Pr\left[\forall\ 1 \leq j \leq L \mid g_j(x) \neq g_j(z)\right]}_{<\frac{1}{e}} + \underbrace{\Pr\left[\sum_{j=1}^{L} |T \cap W_j| \geq 3L\right]}_{\leq\frac{1}{3}} \right)$$

$$> 1 - \left(\frac{1}{e} + \frac{1}{3}\right) = \frac{2}{3} - \frac{1}{e}$$

So $\Pr[d(x^*, z) \leq c\lambda]$ is always greater than the constant $\frac{2}{3} - \frac{1}{e}$.

The code is available in the attached file `lsh.py`.

**Answers:**

- Average searches are:
```
--------------------------------------------------------------------------
Average search time for LSH is 0.1394000768661499 seconds
Average search time for linear search is 0.9721627235412598 seconds
--------------------------------------------------------------------------
```

- Error for different $L$ values (with fixed $k$ value) is presented in figure 2. With increasing of $L$, the error decreases, meaning that we are more likely to hash two similar point to the same bucket if we have more hash tables.
  Error for different $k$ values (with fixed $L$ value) is presented in figure 3. With increasing of $k$ also the error increases, meaning that we are more likely to hash two similar points to different buckets if we have more buckets.

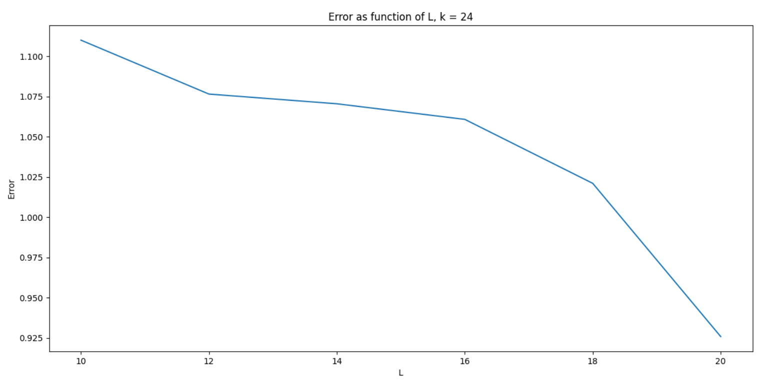


Figure 2: Error for L = 10, 12, 14, 16, 18, 20 and k = 24.

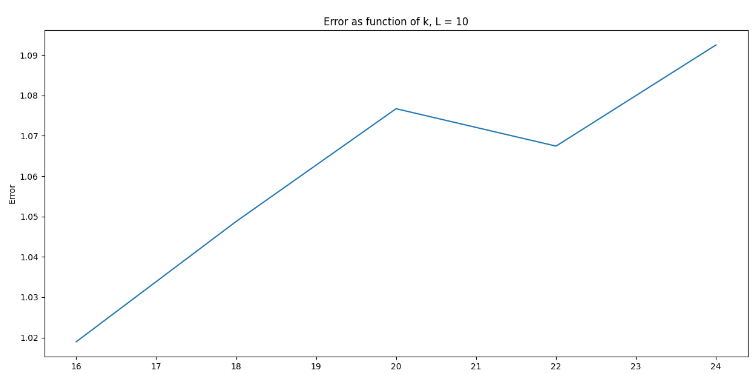


Figure 3: Error for k = 16, 18, 20, 22, 14 and L = 10.

- Figures 5 and 6 show top 10 near neighbors (using LSH and linear search with $L = 10$ and $k = 24$) of query image on figure 4. We can see that some images were found with both LSH and linear search. In general, however, we do not see much similarities between query image and its top 10 neighbors.

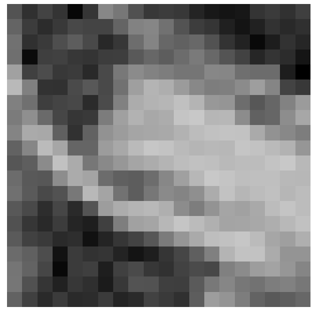

Figure 4: Query image from row 100.

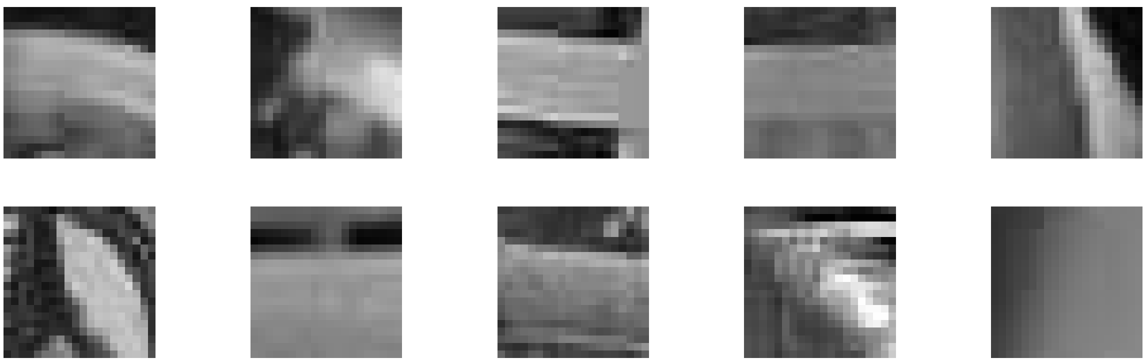

Figure 5: Top 10 near neighbors found using LSH ($L = 10$ and $k = 24$).

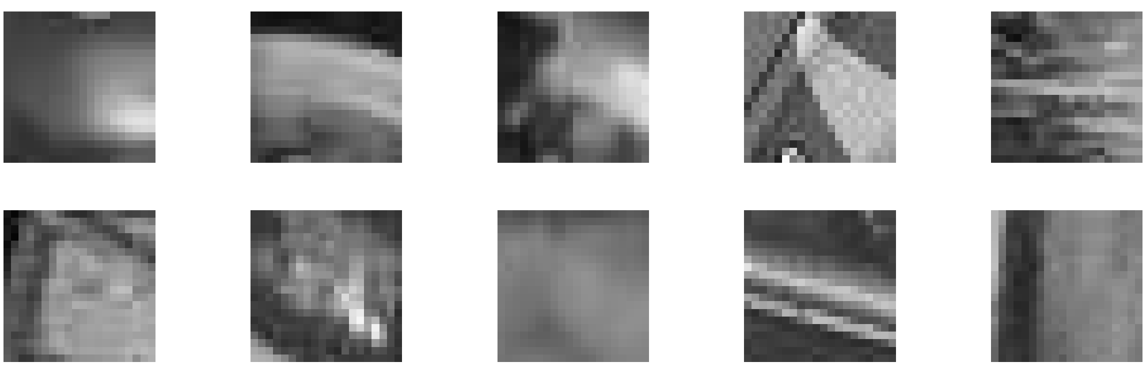

Figure 6: Top 10 near neighbors found using linear search ($L = 10$ and $k = 24$).

# References


[1] *Association rules*, slides, examples, spletna učilnica FRI, ucilnica.fri.uni-lj.si/pluginfile.php/97086/mod_resource/content/6/01_MMDS-Povezovalna_pravila.pdf.

[2] A. Merceron, K. Yacef, *Revisiting interestingness of strong symmetric association rules in educational data*, University of Applied Sciences TFH Berlin, 2007, available at citeseerx.ist.psu.edu/viewdoc/download;jsessionid=7C78F046DED842B5E6D9EC58C151F8E3?doi=10.1.1.142.6648&rep=rep1&type=pdf.

[3] J. Leskovec, A. Rajaraman, J. D. Ullman, *Miminf od Massive Datasets*, available at infolab.stanford.edu/~ullman/mmds/bookL.pdf.

# Information sheet
# CS246: Mining Massive Data Sets

**Assignment Submission** Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via Gradescope (http://www.gradescope.com). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Put all the code for a single question into a single file and upload it.

**Late Homework Policy** Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT*. Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

**Honor Code** We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

**Your name:** Sara Bizjak

**Email:** sarabizjak97@gmail.com **SUID:**

Discussion Group: Petra Podlogar, Mark Vale

I acknowledge and accept the Honor Code.

*(Signed)*