PROFESSIONAL Postgres

Machine learning for better query planning

Oleg Ivanov

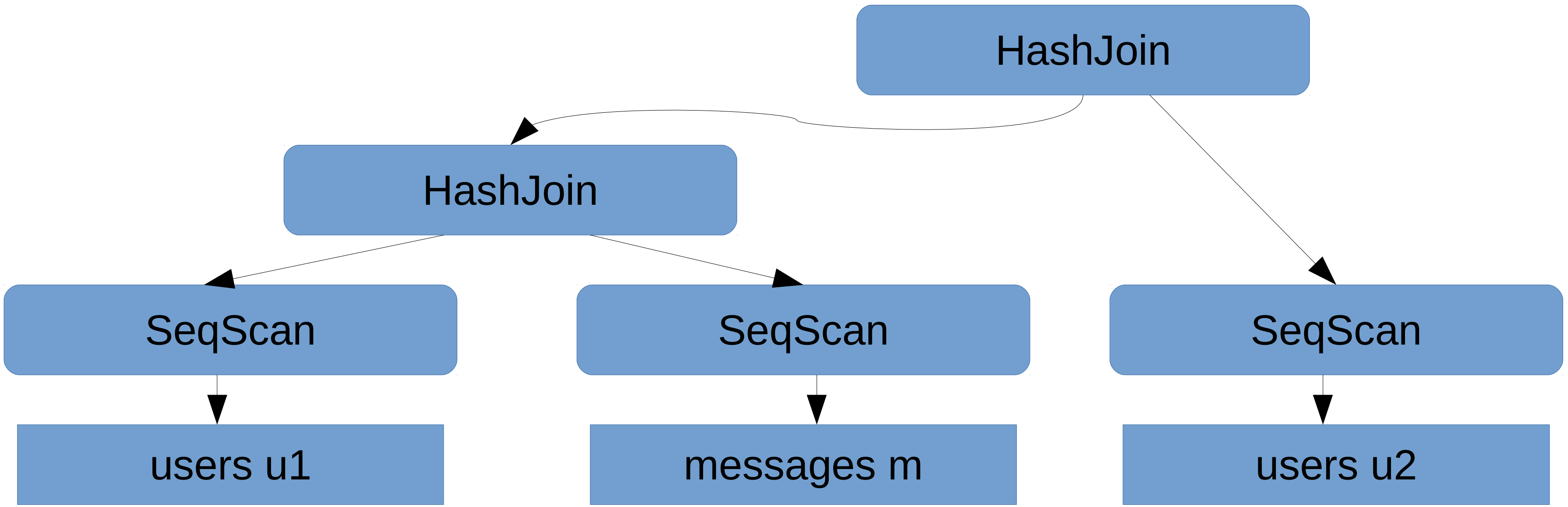
5th of February, 2016

Outline

1. Query planning
2. Machine learning
3. Machine learning for better query planning

Query execution plan

```
SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
```

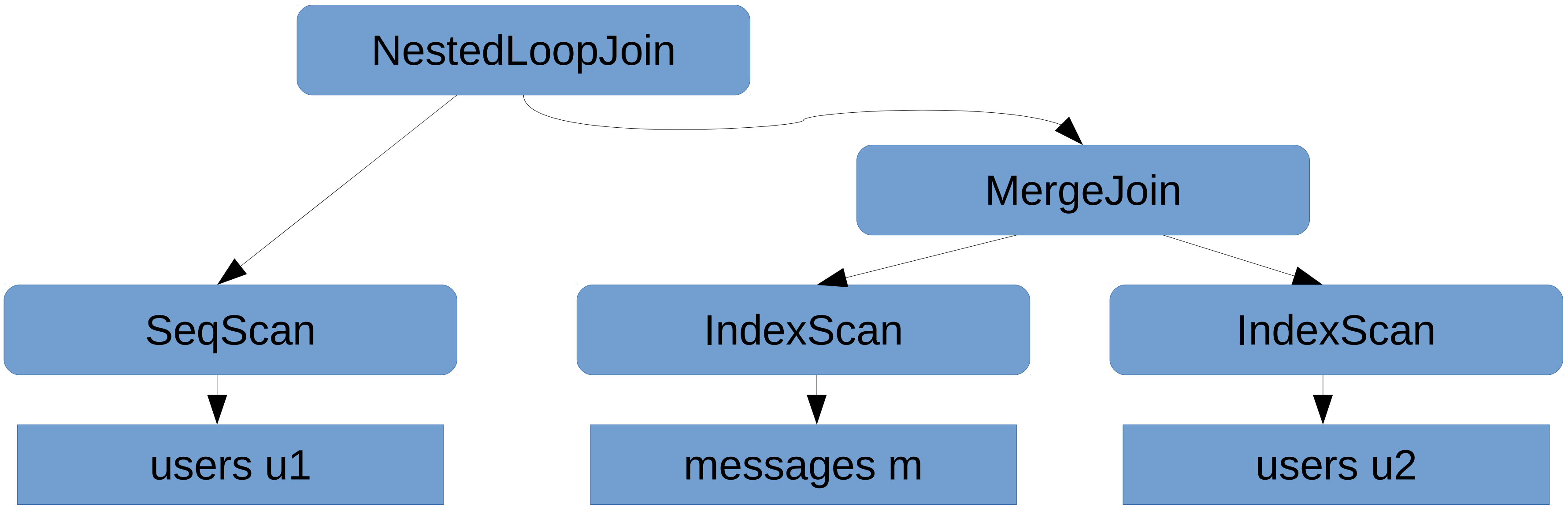


Query execution plan

```
EXPLAIN SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
                                   QUERY PLAN
-----------------------------------------------------------------------------------
 Hash Join  (cost=540.00..439429.44 rows=10003825 width=27)
   Hash Cond: (m.receiver_id = u2.id)
   ->  Hash Join  (cost=270.00..301606.84 rows=10003825 width=23)
         Hash Cond: (m.sender_id = u1.id)
         ->  Seq Scan on messages m  (cost=0.00..163784.25 rows=10003825 width=19)
         ->  Hash  (cost=145.00..145.00 rows=10000 width=4)
               ->  Seq Scan on users u1  (cost=0.00..145.00 rows=10000 width=4)
   ->  Hash  (cost=145.00..145.00 rows=10000 width=4)
         ->  Seq Scan on users u2  (cost=0.00..145.00 rows=10000 width=4)
(9 rows)
```

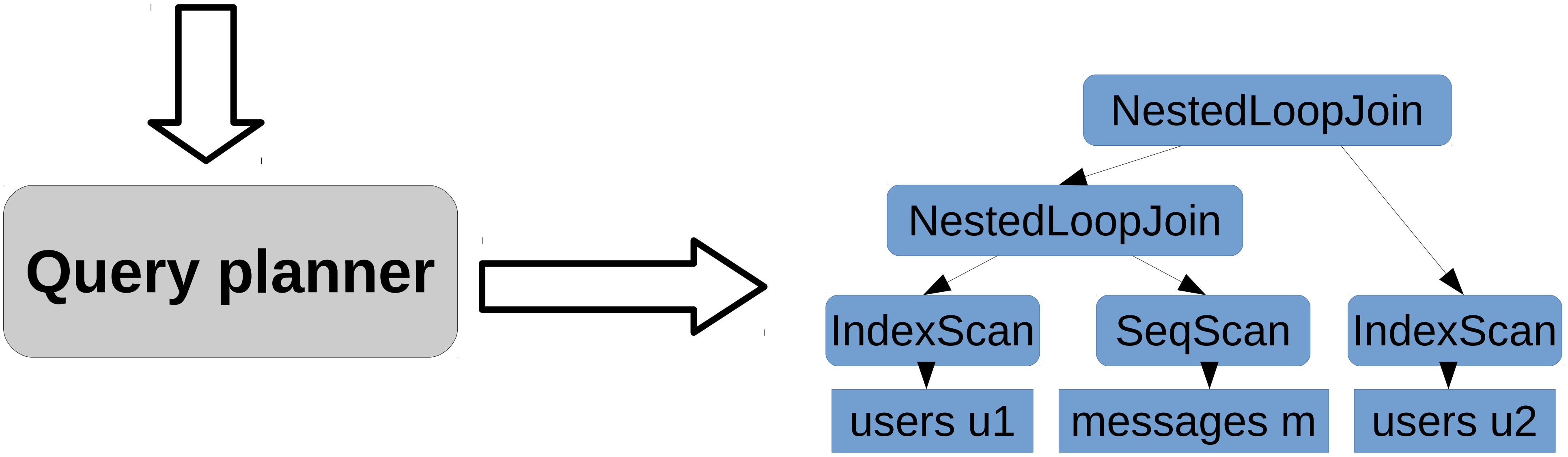
Query execution plan

```
SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
```



Motivation

```
SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
```



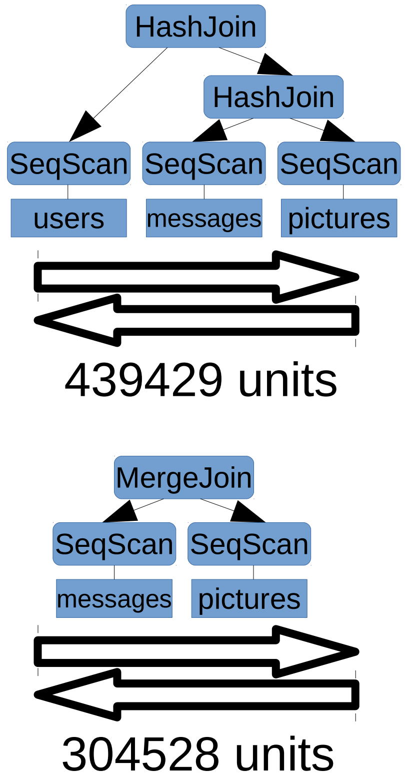
How to choose execution plan?

Optimization method

Dynamic programming

or

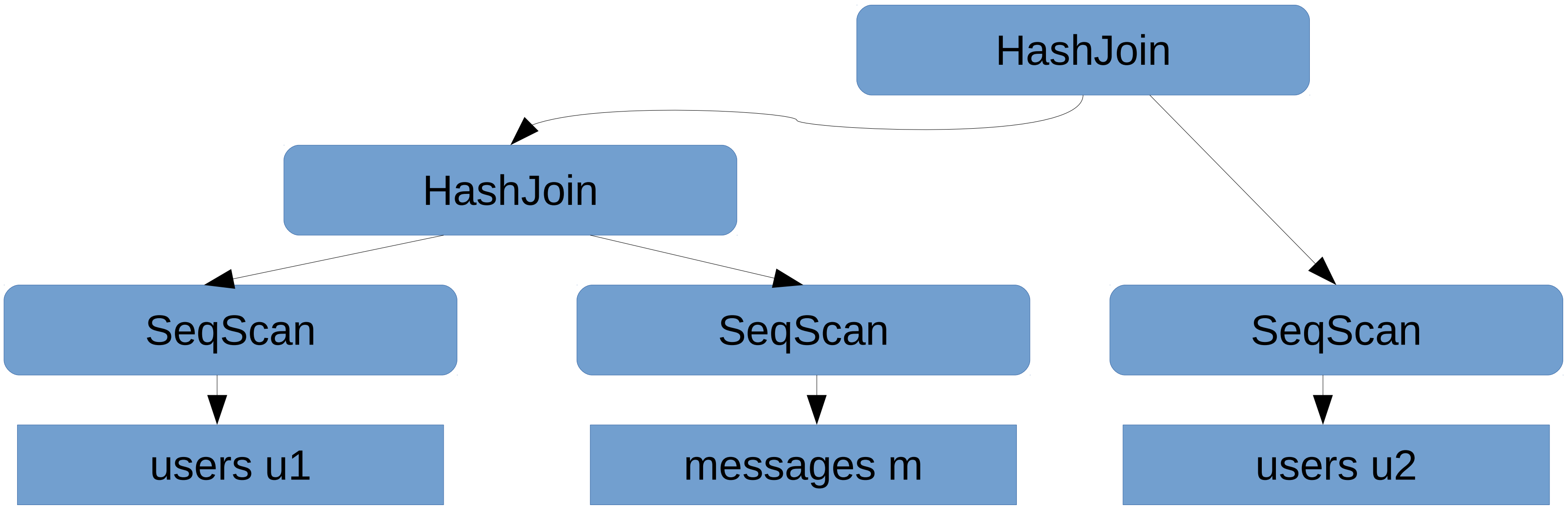
Genetic algorithm



Plan's cost estimation

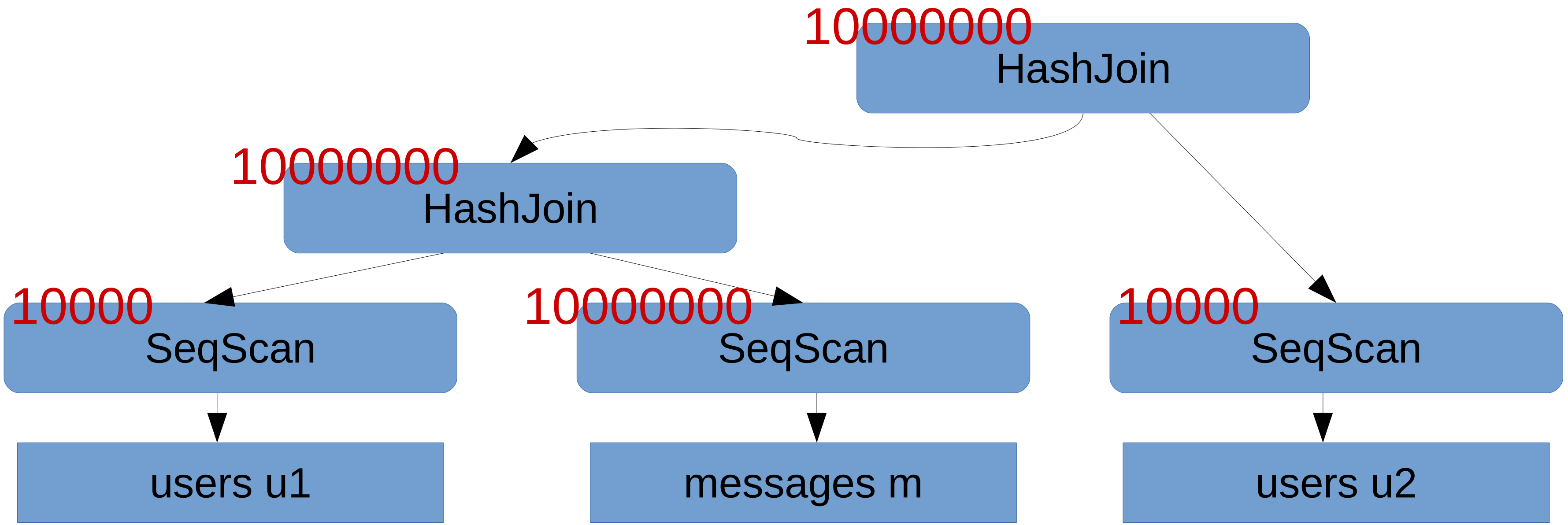
Cost estimation

```sql
SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
```



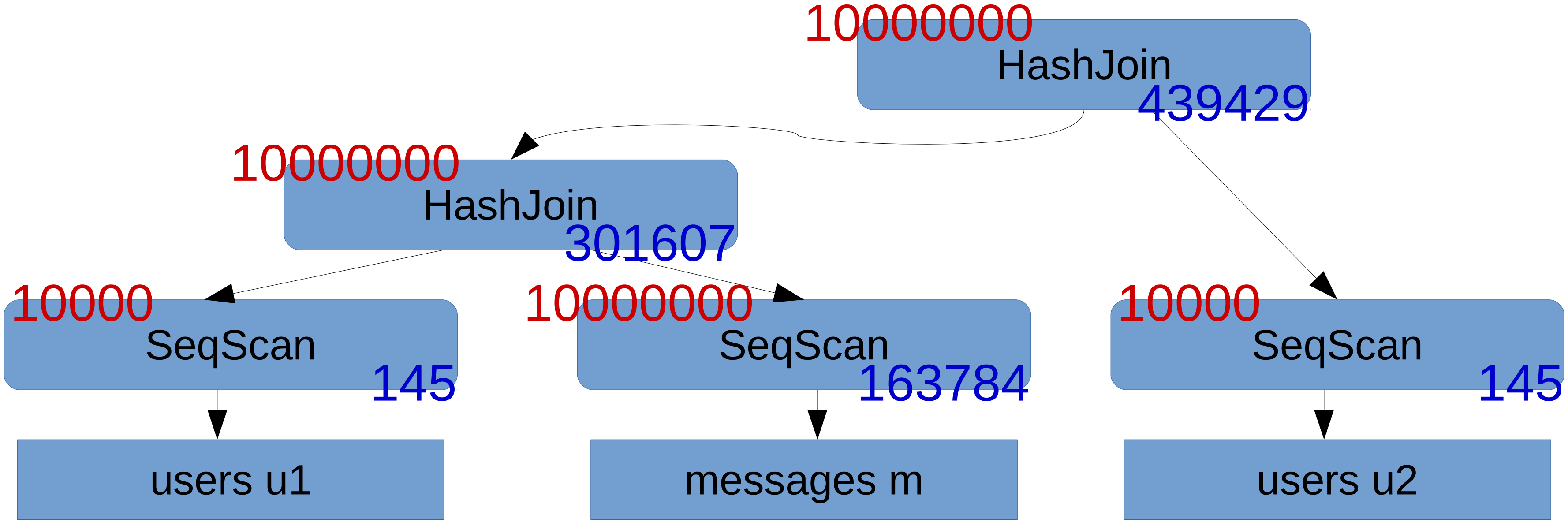
Number of tuples estimation

```
SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
```

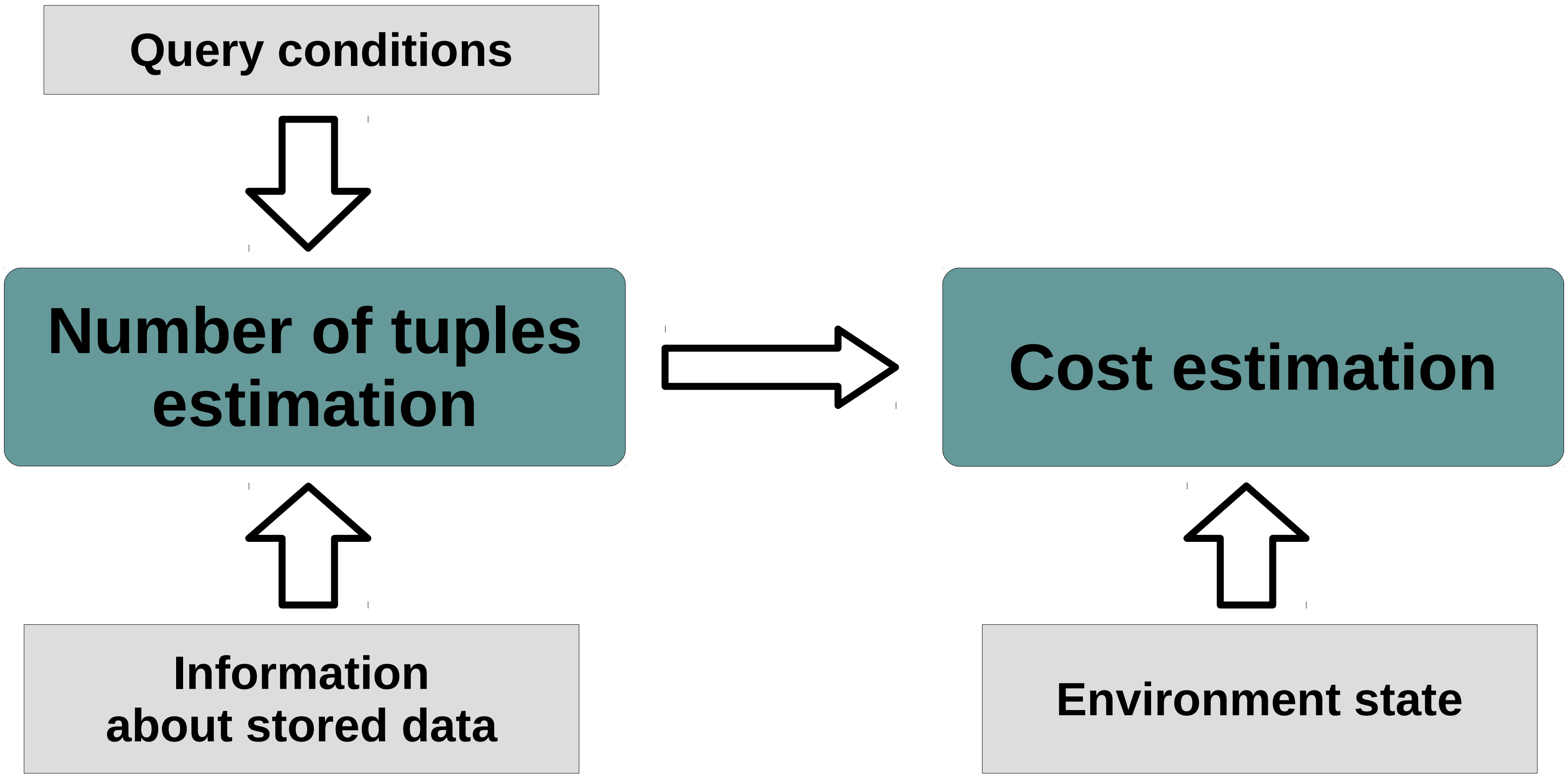


Cost estimation

```
SELECT *
FROM users AS u1, messages AS m, users AS u2
WHERE u1.id = m.sender_id AND m.receiver_id = u2.id;
```



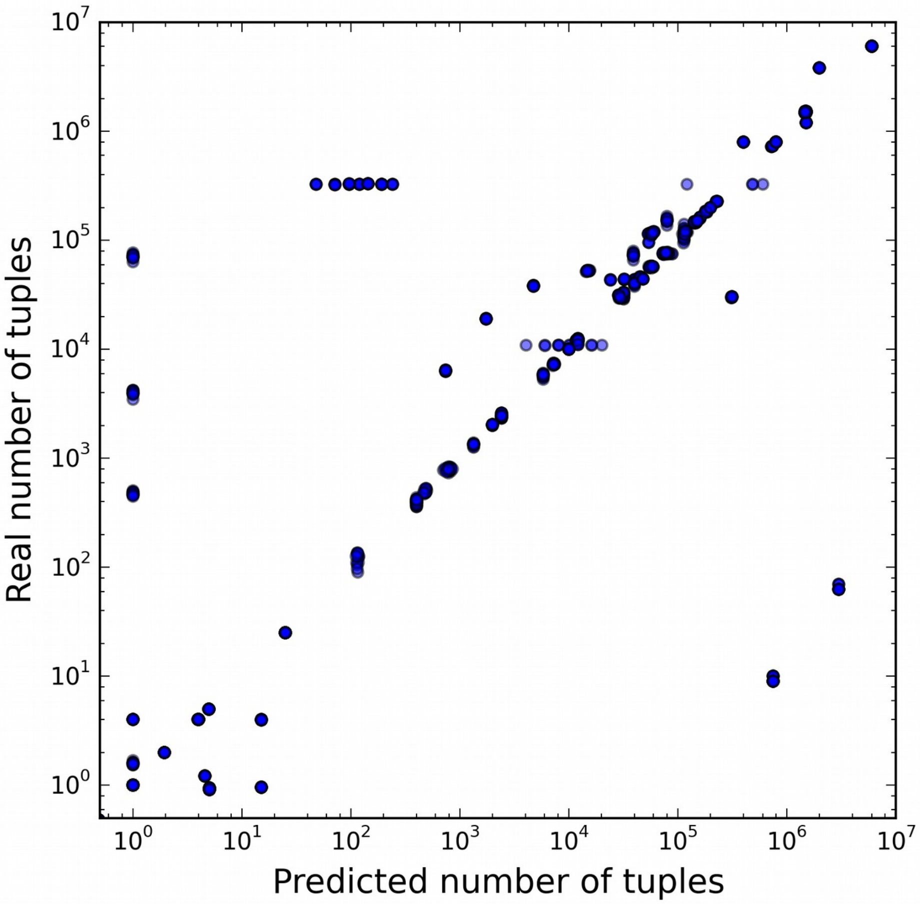
Cost estimation



Cost estimation

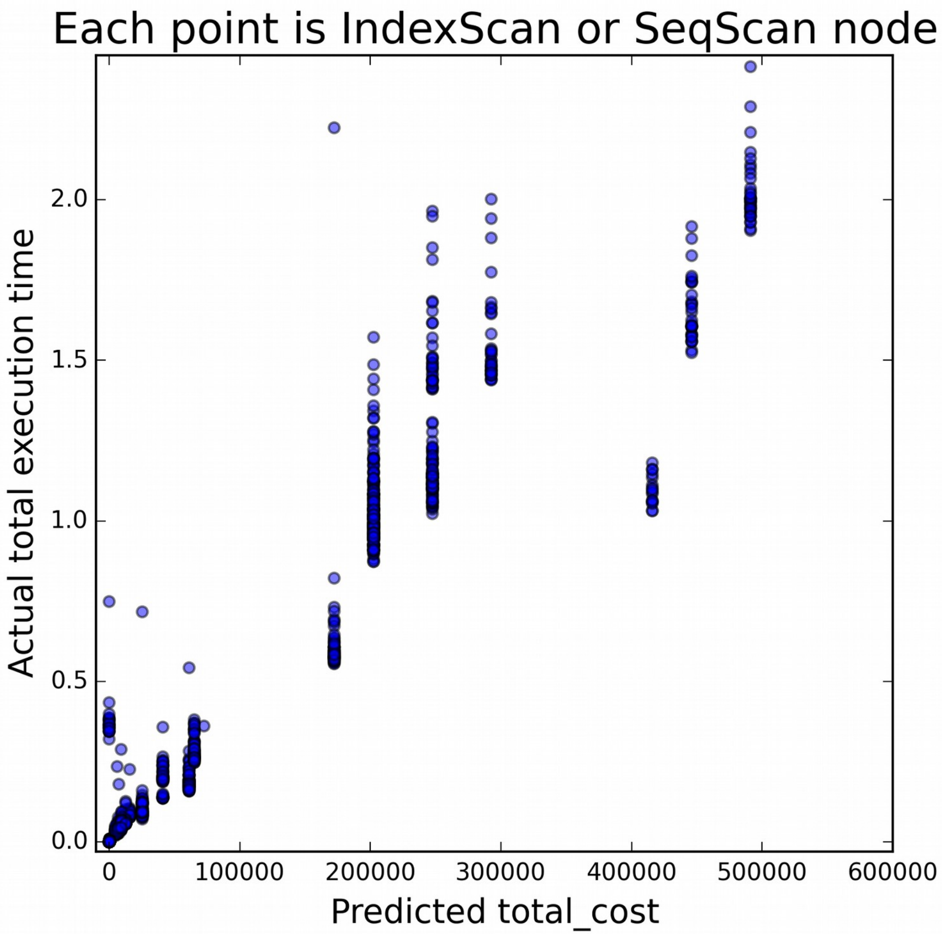
Dataset:
The TPC Benchmark™H (TPC-H)
http://www.tpc.org/tpch/

Number of tuples estimation



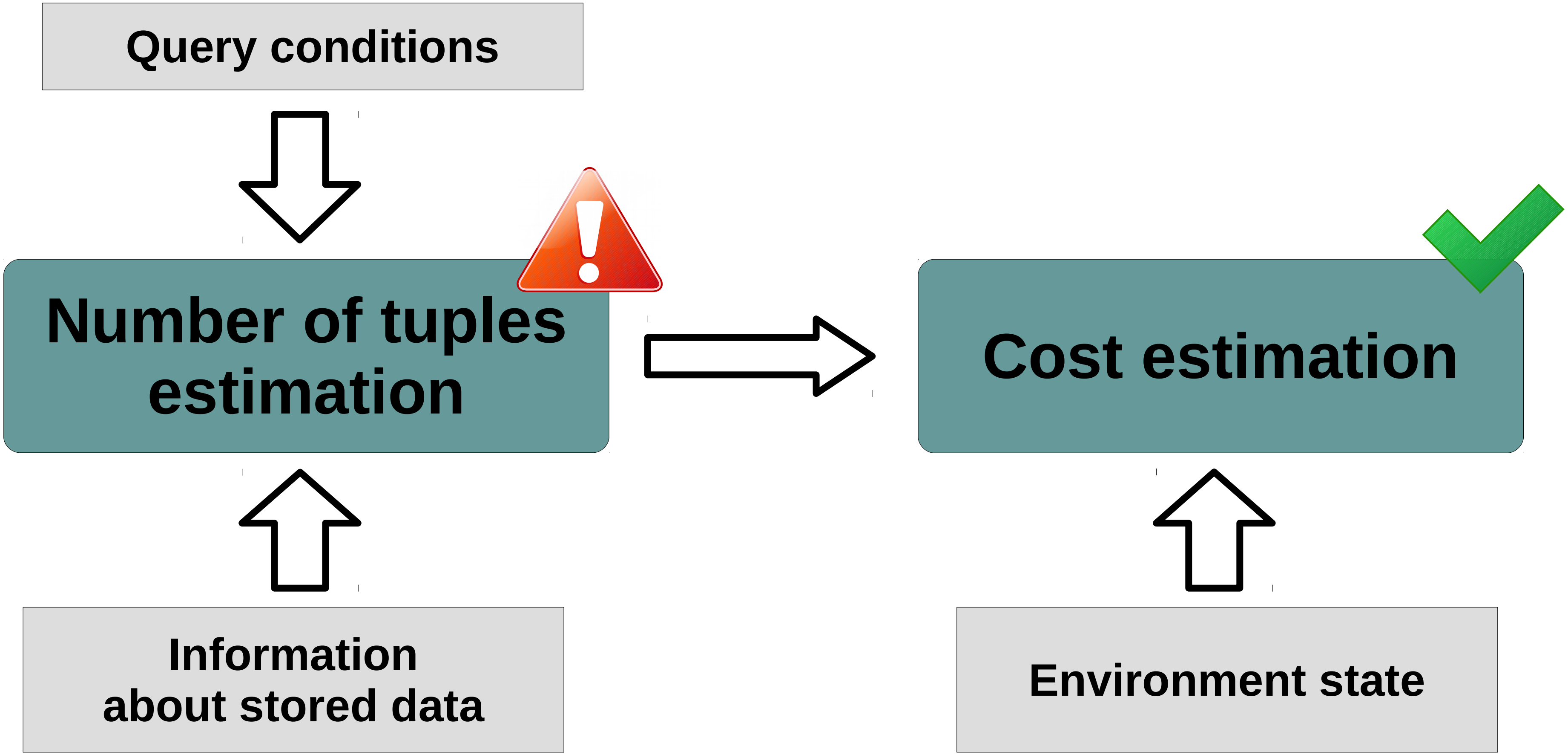
Dataset:
The TPC Benchmark™H (TPC-H)
http://www.tpc.org/tpch/

Cost estimation



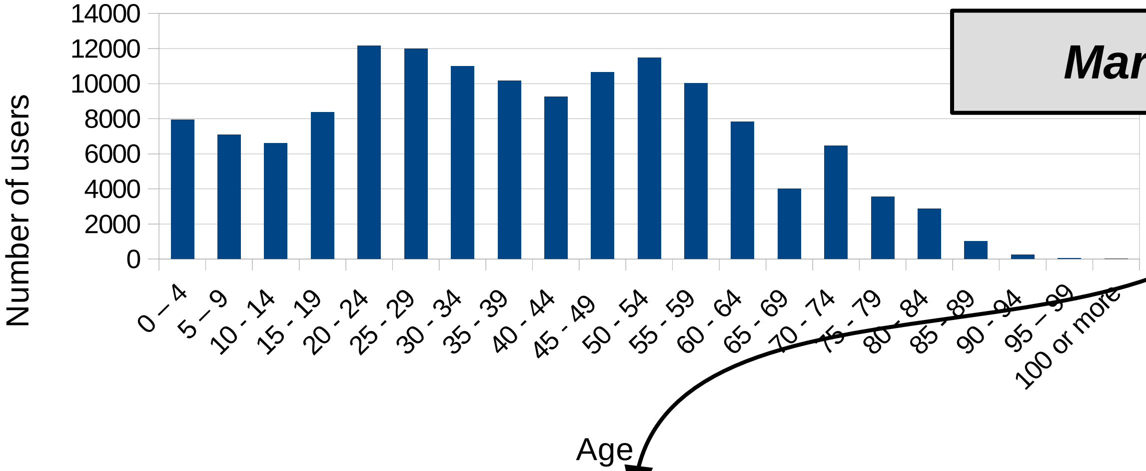
Dataset:
The TPC Benchmark™H (TPC-H)
http://www.tpc.org/tpch/

Cost estimation



Number of tuples estimation

```
SELECT * FROM users
WHERE age < 25;
```



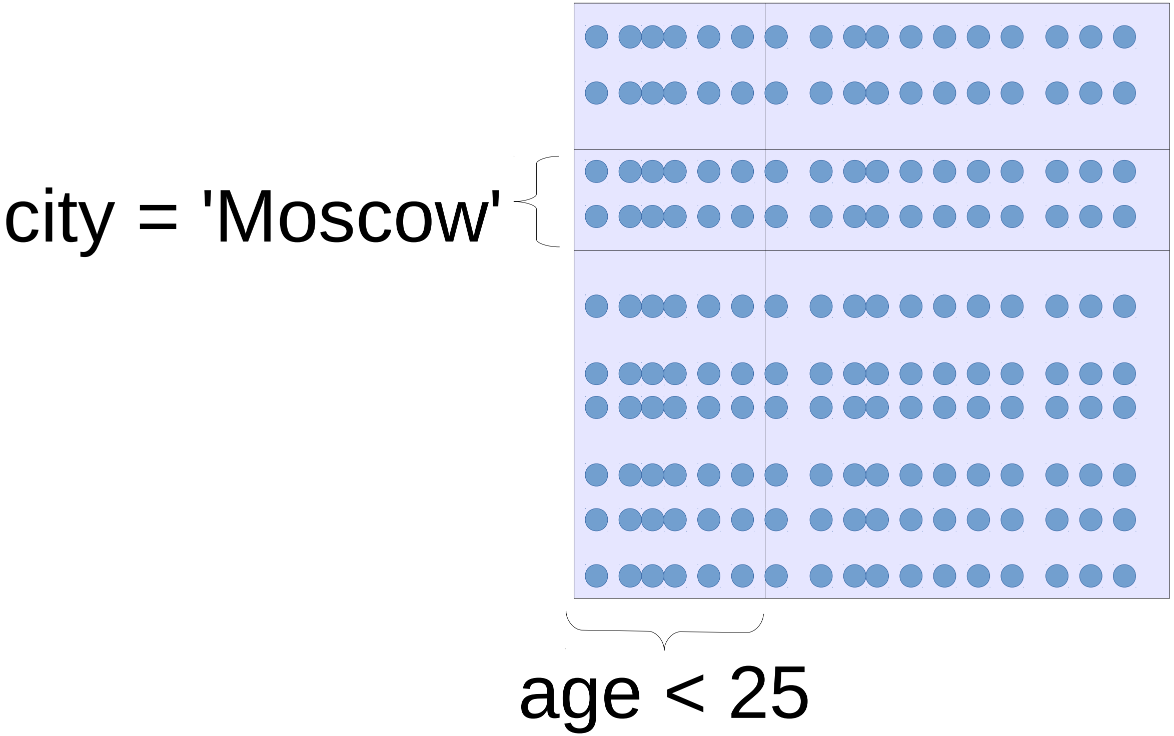
Marginal selectivity

$$Selectivity \simeq 0.3$$

$$Cardinality = Tuples \cdot Selectivity$$

Joint selectivity

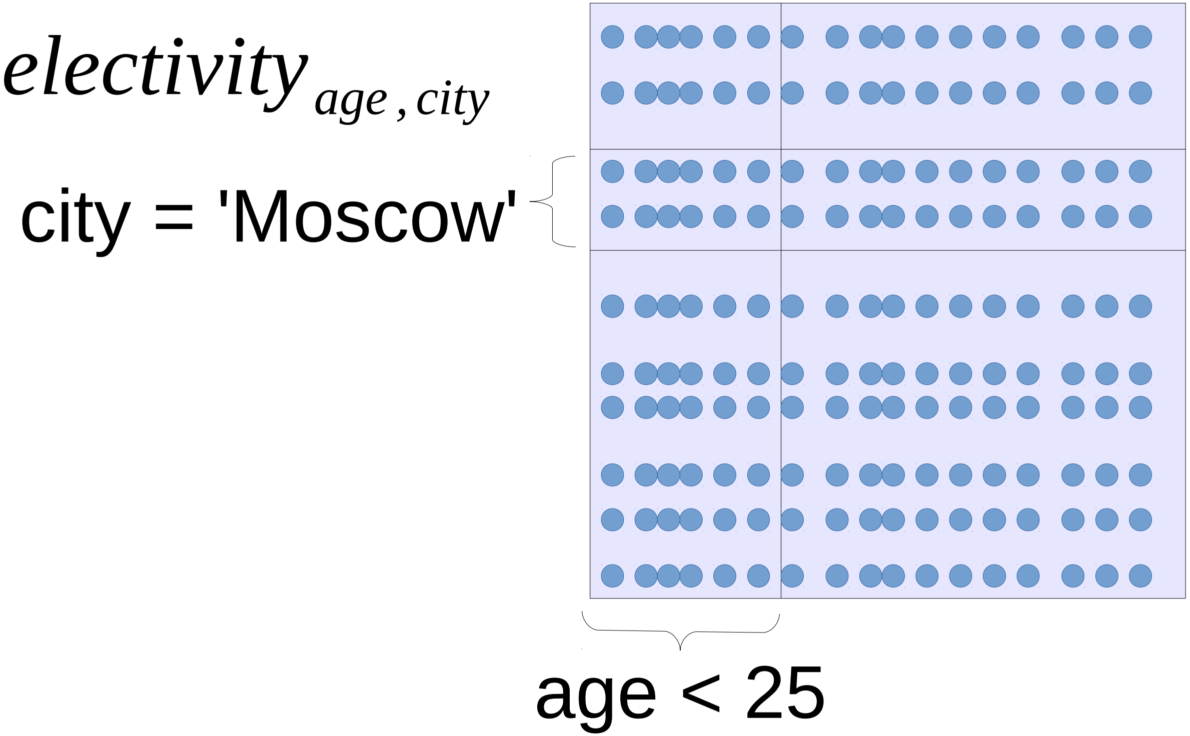
```
SELECT * FROM users
WHERE age < 25 AND city = 'Moscow';
```



Joint selectivity

```
SELECT * FROM users
WHERE age < 25 AND city = 'Moscow';
```

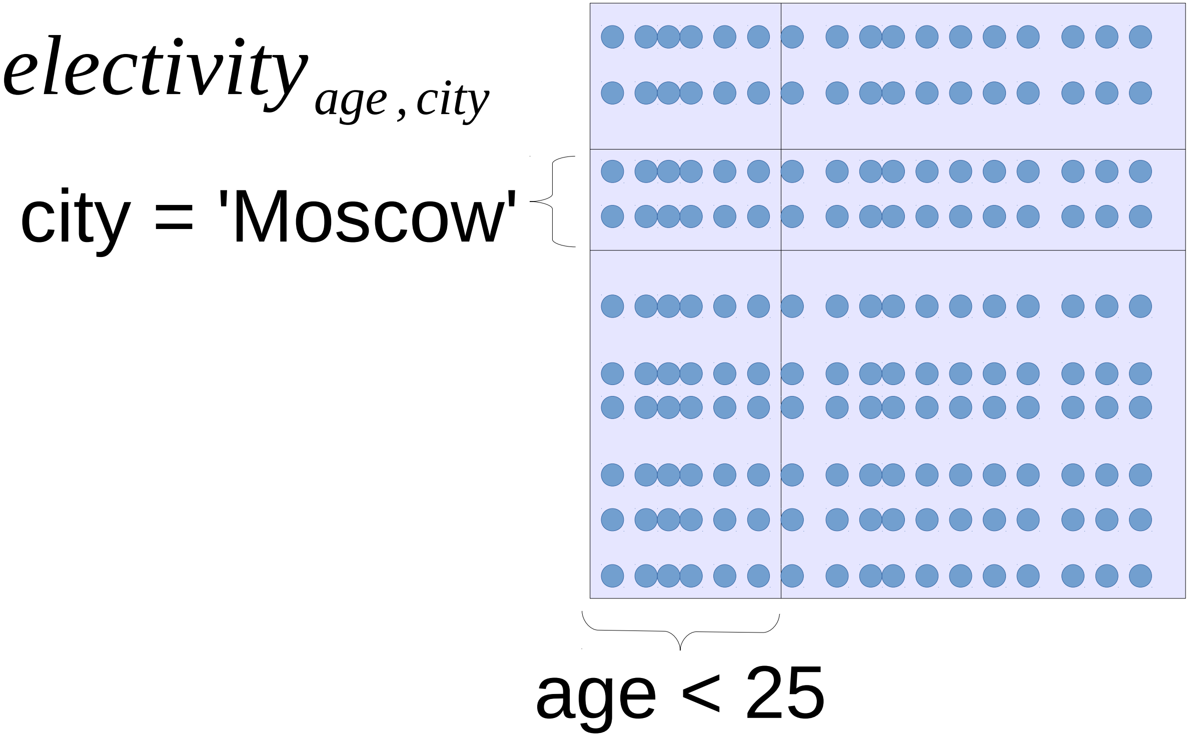
$$Cardinality = Tuples \cdot Selectivity_{age,city}$$



Joint selectivity

```
SELECT * FROM users
WHERE age < 25 AND city = 'Moscow';
```

$$Cardinality = Tuples \cdot Selectivity_{age,city}$$



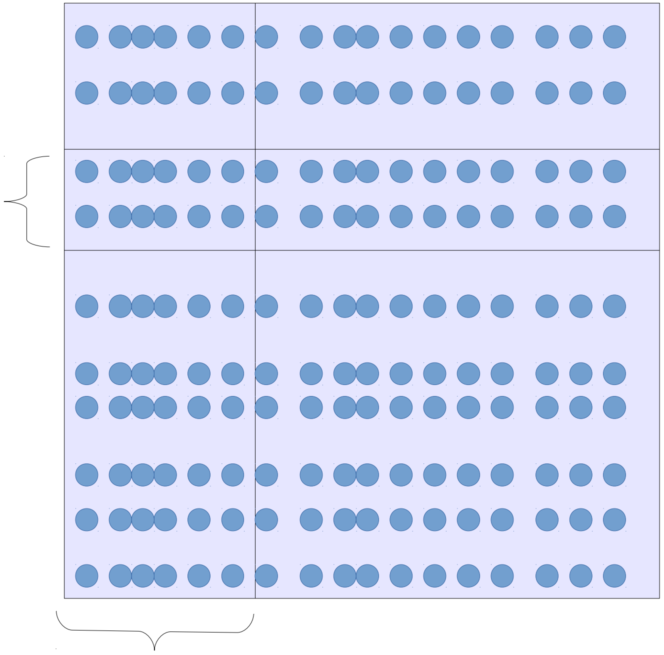
$$Selectivity_{age} \simeq 0.3$$

$$Selectivity_{city} \simeq 0.14$$

Joint selectivity

```
SELECT * FROM users
WHERE age < 25 AND city = 'Moscow';
```

$$Cardinality = Tuples \cdot Selectivity_{age,city}$$



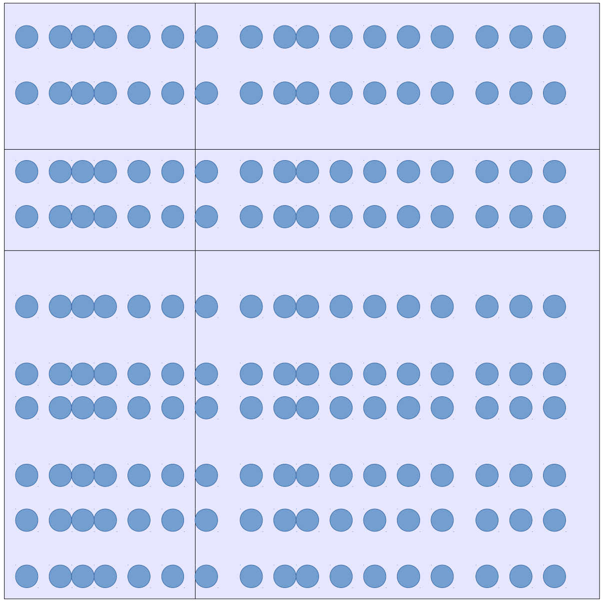
$$Selectivity_{age} \simeq 0.3$$
$$Selectivity_{city} \simeq 0.14$$
$$Selectivity_{age,city} = Selectivity_{age} \cdot Selectivity_{city}$$

Joint selectivity

```
SELECT * FROM users
WHERE age < 25 AND city = 'Moscow';
```

$$Cardinality = Tuples \cdot Selectivity_{age\,,city}$$

city = 'Moscow'



$$Selectivity_{age} \simeq 0.3$$

$$Selectivity_{city} \simeq 0.14$$

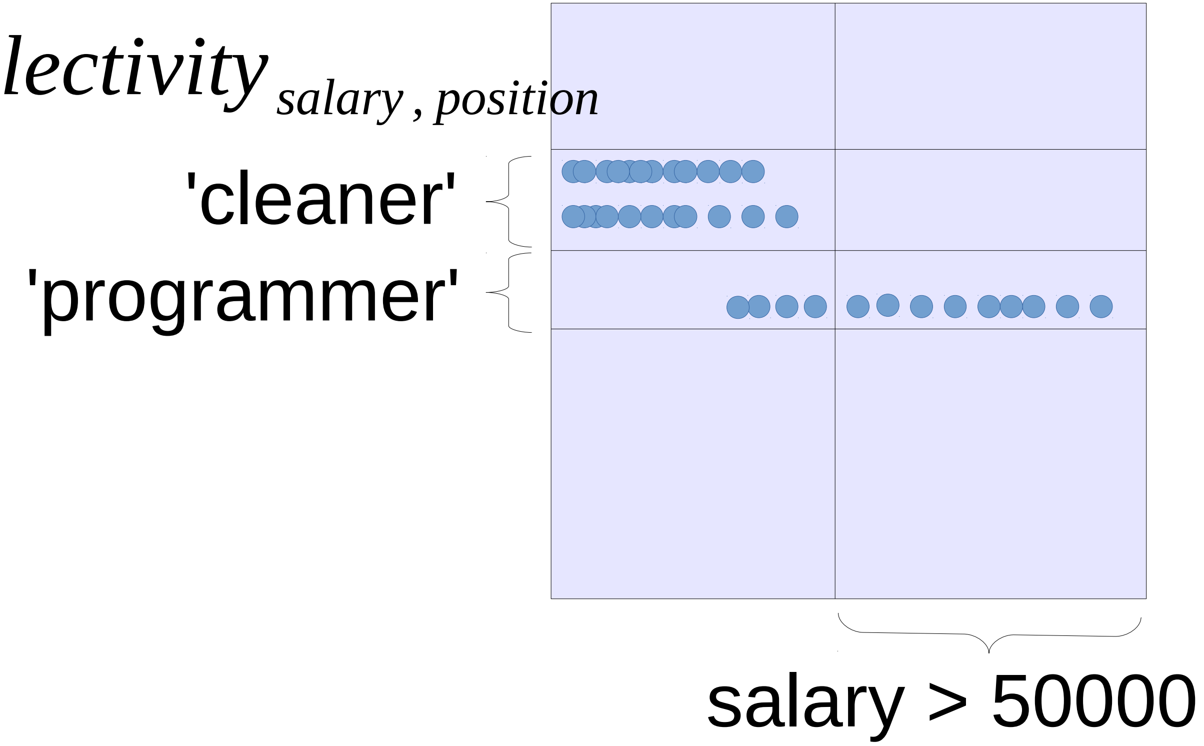
$$Selectivity_{age\,,city} = Selectivity_{age} \cdot Selectivity_{city}$$

Excluding $Selectivity_{25<age\ \mathrm{AND}\ age<57} = Selectivity_{25<age<57}$

Joint selectivity

```
SELECT * FROM users
WHERE position = 'cleaner' AND salary > 50000;
```

$$Cardinality = Tuples \cdot Selectivity_{salary,\, position}$$



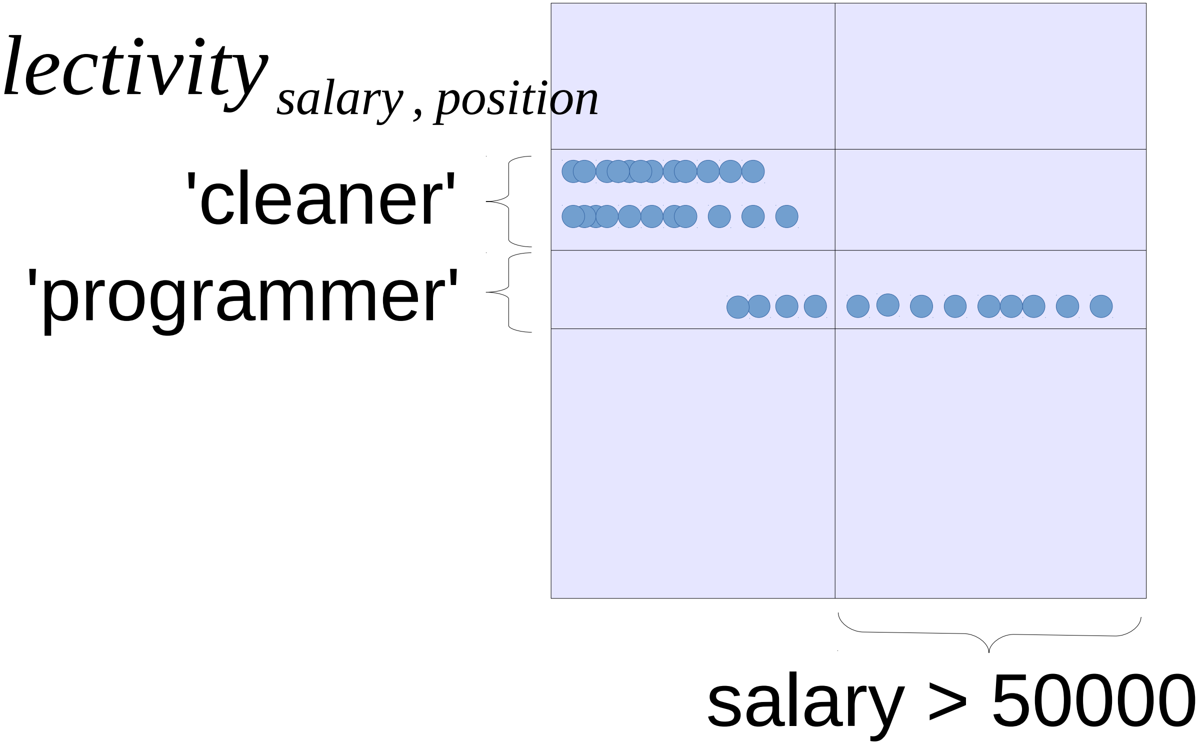
Joint selectivity

SELECT * **FROM** users
WHERE position = 'cleaner' **AND** salary > 50000;

$$Cardinality = Tuples \cdot Selectivity_{salary,\, position}$$

$$Selectivity_{cleaner} \simeq 0.2$$

$$Selectivity_{salary} \simeq 0.3$$



Joint selectivity

```
SELECT * FROM users
WHERE position = 'cleaner' AND salary > 50000;
```

$$Cardinality = Tuples \cdot Selectivity_{salary,\,position}$$

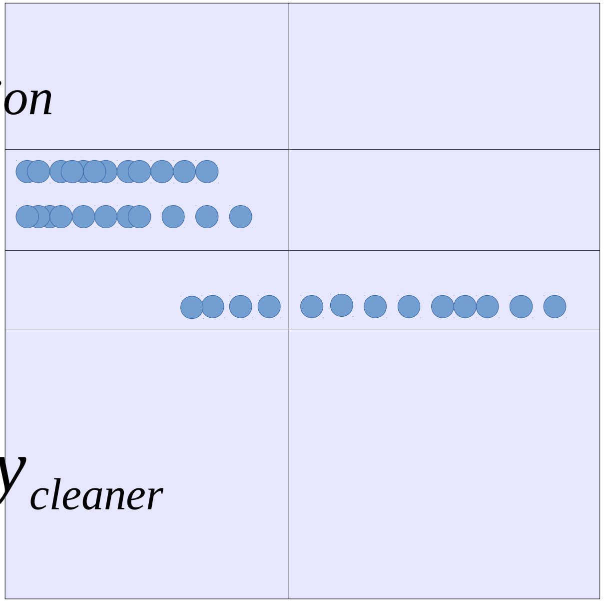
$$Selectivity_{cleaner} \simeq 0.2$$

$$Selectivity_{salary} \simeq 0.3$$

$$Selectivity_{salary,\,cleaner} \simeq Selectivity_{salary} \cdot Selectivity_{cleaner}$$

'cleaner'

'programmer'



salary > 50000

Joint selectivity

```
SELECT * FROM users
WHERE position = 'cleaner' AND salary > 50000;
```

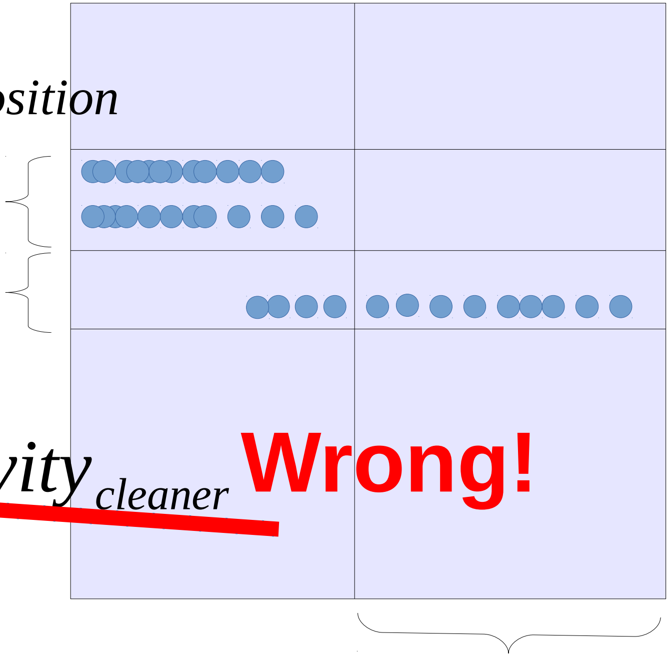
$$Cardinality = Tuples \cdot Selectivity_{salary,\, position}$$

$$Selectivity_{cleaner} \simeq 0.2$$

$$Selectivity_{salary} \simeq 0.3$$

$$\cancel{Selectivity_{salary,\, cleaner} = Selectivity_{salary} \cdot Selectivity_{cleaner}}$$

Wrong!



Joint selectivity

```
SELECT * FROM users
WHERE position = 'cleaner' AND salary > 50000;
```

$$Cardinality = Tuples \cdot Selectivity_{salary,\,position}$$

$$Selectivity_{cleaner} \simeq 0.2$$

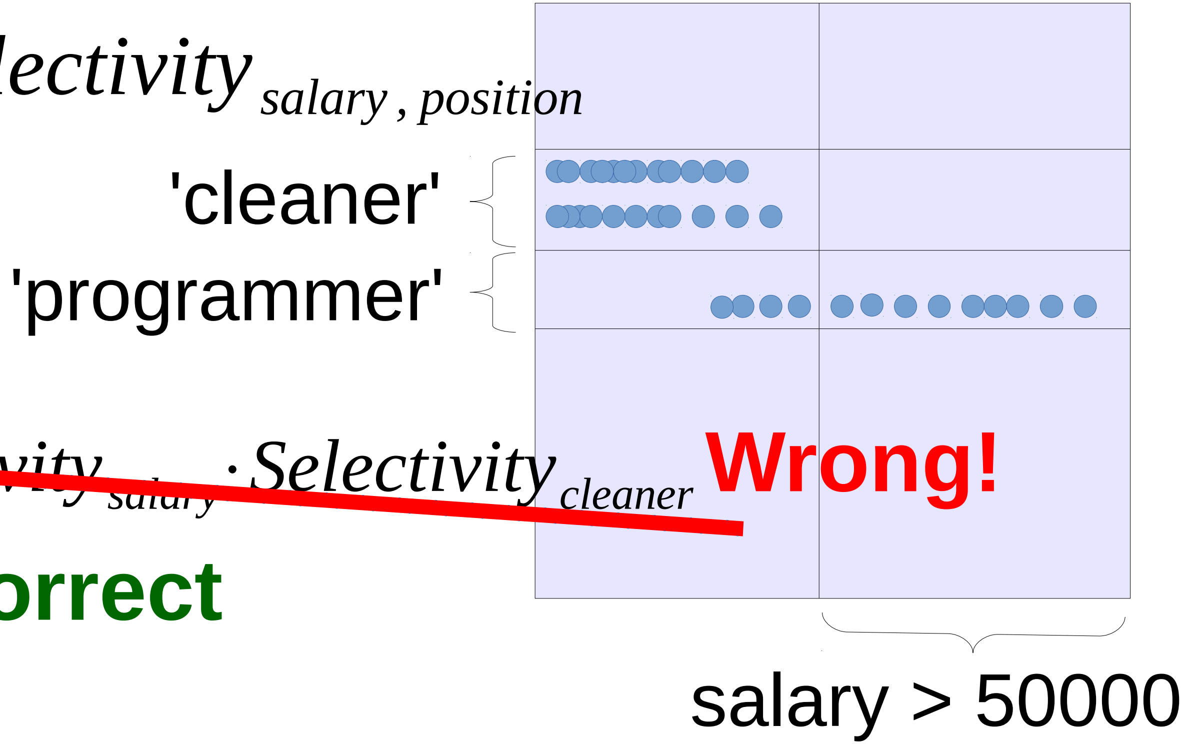
$$Selectivity_{salary} \simeq 0.3$$

$$\cancel{Selectivity_{salary,\,cleaner} = Selectivity_{salary} \cdot Selectivity_{cleaner}}$$

Wrong!

$$Selectivity_{salary,\,cleaner} \simeq 0$$

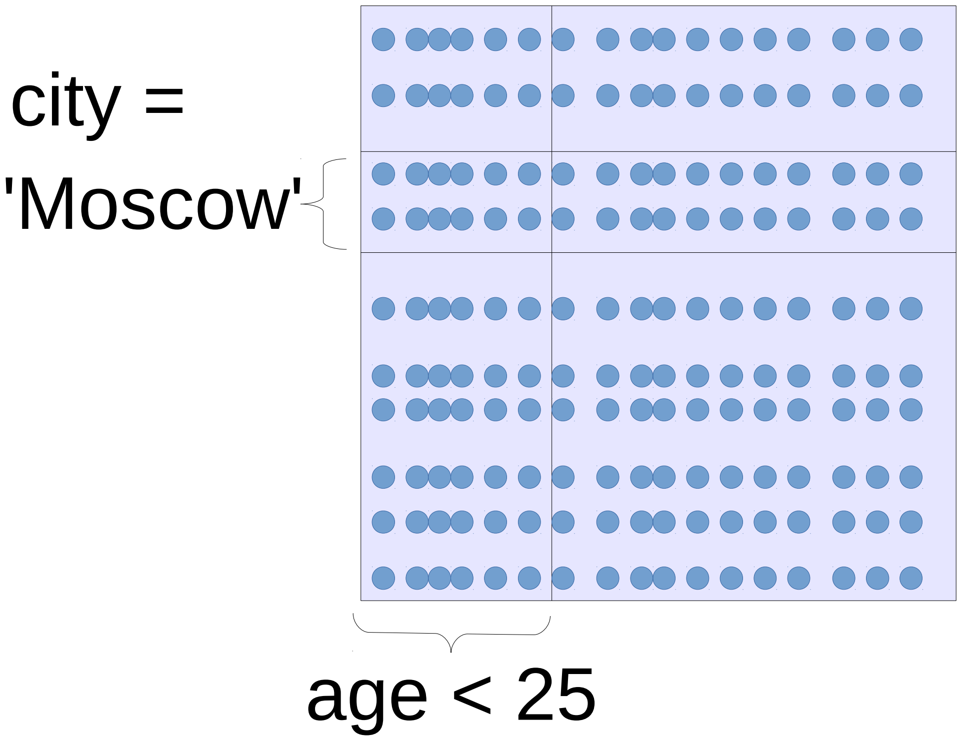
Correct



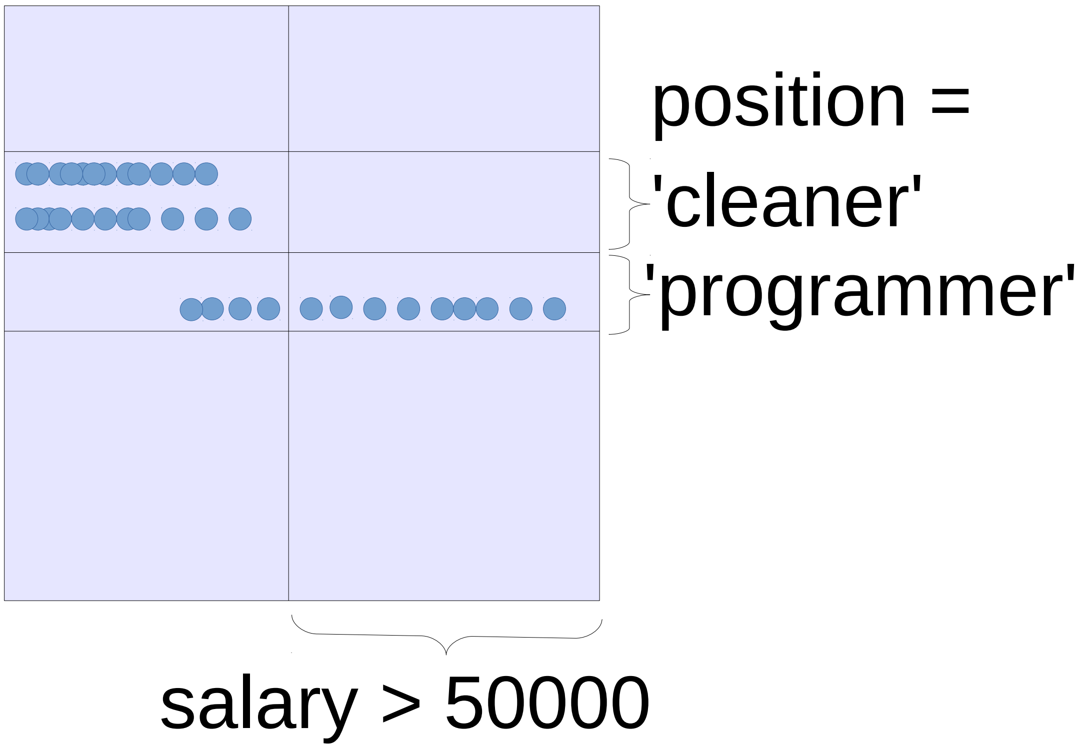
Dependency of conditions

$$Selectivity_{1,2} \simeq Selectivity_1 \cdot Selectivity_2 \,?$$

Independent conditions



Dependent conditions



Problem statement

Marginal selectivities:

1. 0.0001
2. 0.78
3. 0.23
4. 0.4
5. 0.5

l_partkey = p_partkey
AND
l_shipdate >= date '1995-12-01'
AND
l_shipdate < date '1995-12-01' + interval '1' month
AND
l_commitdate < l_receiptdate
AND
l_shipdate < l_commitdate

Joint selectivity

Information about data

List of conditions

Selectivity is 0.25!

Machine learning

Machine learning

- Machine learning tries to find regularities in data.
- Data is a set of objects.
- Each objects has a set of observed variables (features) and hidden variables.
- The goal is to find the way of predicting the hidden variables for a new object given the values of features.

Credit scoring

Return time	Age	Salary	Married	Number of children	Has high education
14	25	40000	0	0	1
12	47	100000	1	2	1
9	55	100000	1	2	1
10	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

K nearest neighbours

1. Define similarity between two objects:

$$\mathrm{dist}(\vec{x}_1, \vec{x}_2) = \ldots \qquad \mathrm{sim}(\vec{x}_1, \vec{x}_2) = \frac{1}{1 + \mathrm{dist}(\vec{x}_1, \vec{x}_2)}$$

2. Define K.

3. Find the K nearest objects and compute their weights:

$$w_i = \frac{\mathrm{sim}(\vec{x}_{new}, \vec{x}_{(i)})}{\mathrm{sim}(\vec{x}_{new}, \vec{x}_{(1)}) + \ldots + \mathrm{sim}(\vec{x}_{new}, \vec{x}_{(K)})}$$

4. Return weighted combination of their hidden variables:

$$y_{new} = w_1 y_{(1)} + \ldots + w_K y_{(K)}$$

K nearest neighbours

Return time	Age	Salary	Married	Number of children	Has high education
14	25	40000	0	0	1
12	47	100000	1	2	1
9	55	100000	1	2	1
10	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

K nearest neighbours

Return time	Age	Salary	Married	Number of children	Has high education
14	25	40000	0	0	1
12	47	100000	1	2	1
9	55	100000	1	2	1
10	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

$$\operatorname{dist}(\vec{x}_1, \vec{x}_2) = |a_1 - a_2| + \frac{|s_1 - s_2|}{10000} + |m_1 - m_2| + |c_1 - c_2| + |e_1 - e_2|$$

K nearest neighbours

Return time	Age	Salary	Married	Number of children	Has high education
14 5	25	40000	0	0	1
12 26	47	100000	1	2	1
9 34	55	100000	1	2	1
10 8	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

$$\operatorname{dist}(\vec{x}_1, \vec{x}_2) = |a_1 - a_2| + \frac{|s_1 - s_2|}{10000} + |m_1 - m_2| + |c_1 - c_2| + |e_1 - e_2|$$

K nearest neighbours

Return time	Age	Salary	Married	Number of children	Has high education
14 5	25	40000	0	0	1
12 26	47	100000	1	2	1
9 34	55	100000	1	2	1
10 8	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

$K=2$

$$\mathrm{sim}\left(\vec{x}_{new}, \vec{x}_{(1)}\right) = \frac{1}{6} \qquad \mathrm{sim}\left(\vec{x}_{new}, \vec{x}_{(2)}\right) = \frac{1}{9}$$

K nearest neighbours

Return time	Age	Salary	Married	Number of children	Has high education
14 5	25	40000	0	0	1
12 26	47	100000	1	2	1
9 34	55	100000	1	2	1
10 8	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

$$K=2$$

$$w_1=\frac{1/6}{1/6+1/9}=\frac{3}{5} \qquad w_2=\frac{1/9}{1/6+1/9}=\frac{2}{5}$$

K nearest neighbours

Return time	Age	Salary	Married	Number of children	Has high education
14 5	25	40000	0	0	1
12 26	47	100000	1	2	1
9 34	55	100000	1	2	1
10 8	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

$$K=2$$

$$y_{new} \simeq w_1 y_{(1)} + w_2 y_{(2)} = \frac{3}{5} \cdot 14 + \frac{2}{5} \cdot 10 = 12.4$$

Ridge regression

1. Model:

$$y_i \simeq w_1 \cdot x_{i,1} + \ldots + w_D \cdot x_{i,D} + b = f(\vec{x}_i, \vec{w}, b)$$

2. Fitting parameters:

$$L(\vec{w}, b) = \sum_{i=1}^{l} (f(\vec{x}_i, \vec{w}, b) - y_i)^2 + \lambda \sum_{i=1}^{D} w_i^2 \rightarrow \min_{\vec{w}, b}$$

3.Make predictions:

$$y_{new} \simeq f(x_{new}^{\rightarrow}, \vec{w}^{min}, b^{min}) = w_1^{min} \cdot x_{new,1} + \ldots + w_D^{min} \cdot x_{new,D} + b^{min}$$

Ridge regression

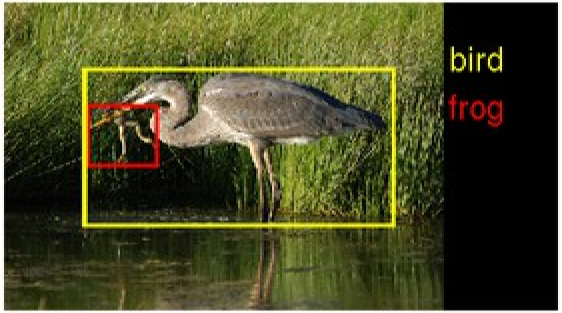
Return time	Age	Salary	Married	Number of children	Has high education
14	25	40000	0	0	1
12	47	100000	1	2	1
9	55	100000	1	2	1
10	32	80000	1	1	1
...	...	...	...	...	...
???	28	50000	1	0	1

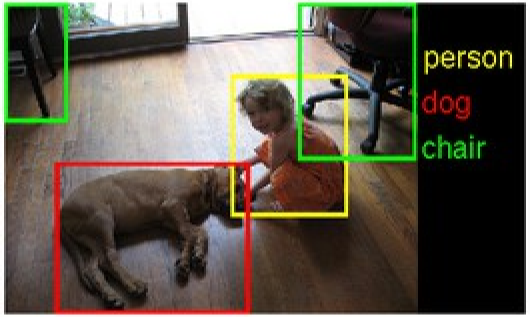
$$y_{new} \simeq 15.9 - 1.4 \cdot 10^{-2} \cdot age - 5 \cdot 10^{-5} salary - 0.5 \cdot married - 0.2 \cdot children$$

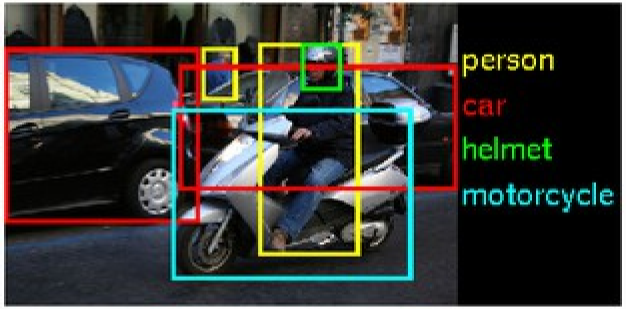
$$y_{new} \simeq 12.4$$

Modern methods

1. Random forest
2. Gradient boosting
3. Graphical models
4. Bayesian methods
5. Deep learning







PROFESSIONAL Postgres

Problem statement

Marginal selectivities:

1. 0.0001
2. 0.78
3. 0.23
4. 0.4
5. 0.5

l_partkey = p_partkey
AND
l_shipdate >= date '1995-12-01'
AND
l_shipdate < date '1995-12-01' + interval '1' month
AND
l_commitdate < l_receiptdate
AND
l_shipdate < l_commitdate

Joint selectivity

Information about data

List of conditions

Selectivity is 0.25!

Machine learning

Problem statement

Marginal selectivities:
1. 0.0001
2. 0.78
3. 0.23
4. 0.4
5. 0.5

l_partkey = p_partkey
AND
l_shipdate >= **const**
AND
l_shipdate < **const**
AND
l_commitdate < l_receiptdate
AND
l_shipdate < l_commitdate

Joint selectivity

Information about data

List of conditions

Selectivity is 0.25!

Machine learning

Problem statement

Selectivity	users.age > const	users.city = const	messages.sender_id = users.id
0.25	0.25	-	-
0.23	0.25	0.6	-
0.3	0.5	0.6	-
0.0005	-	0.5	0.001
...	...	...	...
???	0.5	0.5	-

Problem statement

LogSelectivity	users.age > const	users.city = const	messages.sender_id = users.id
-1.386	-1.386	0	0
-1.470	-1.386	-0.511	0
-1.204	-0.693	-0.511	0
-7.600	0	-0.693	-6.908
...	...	...	...
???	-0.693	-0.693	0

PostgreSQL model

$$Joint\_selectivity = \prod_{c \in conditions} selectivity_c$$

$$\log Joint\_selectivity = \sum_{c \in conditions} \log selectivity_c$$

A special case of ridge regression:

$$\log Joint\_selectivity = \sum_{c \in conditions} w_c \log selectivity_c$$

The tried techniques

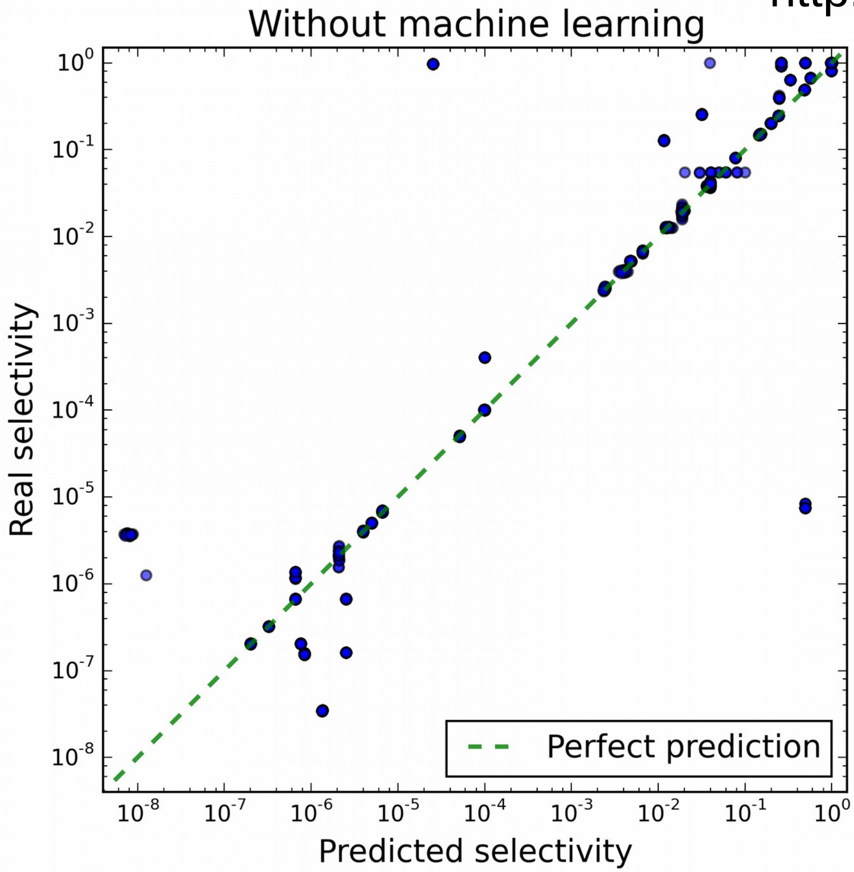
- Ridge regression
  - stochastic gradient descent
- Composition of ridge regressions
  - stochastic gradient descent
  - the exact solution of linear algebraic equation system by Gauss
- K Nearest Neighbours
  - K = 1

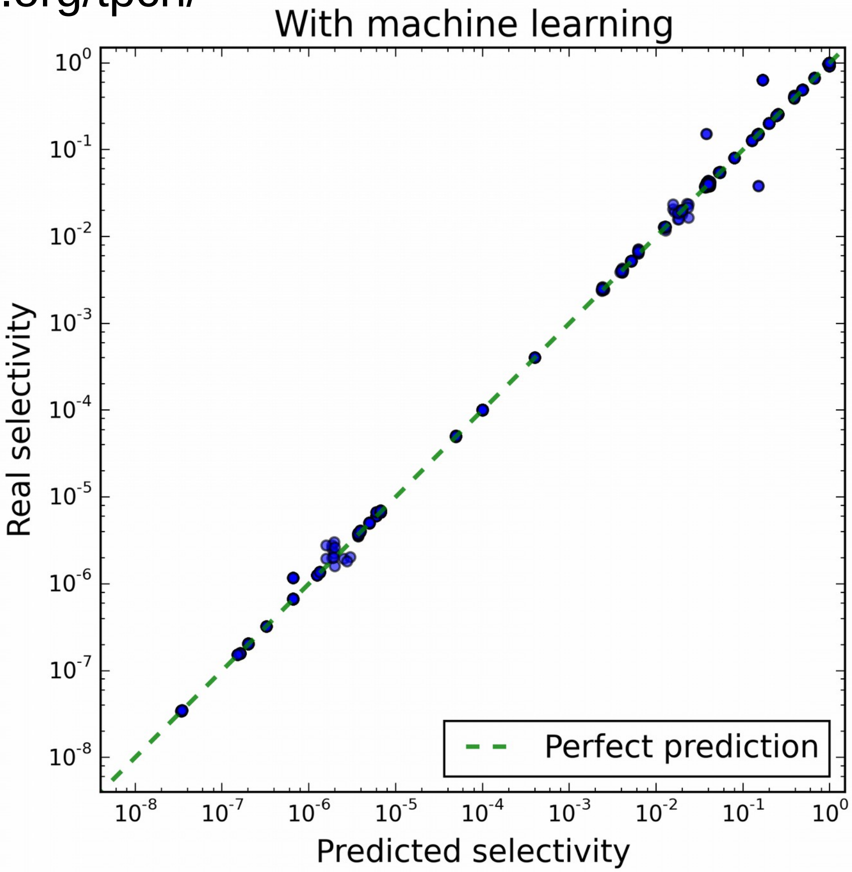
Obtained results: selectivity

Dataset:

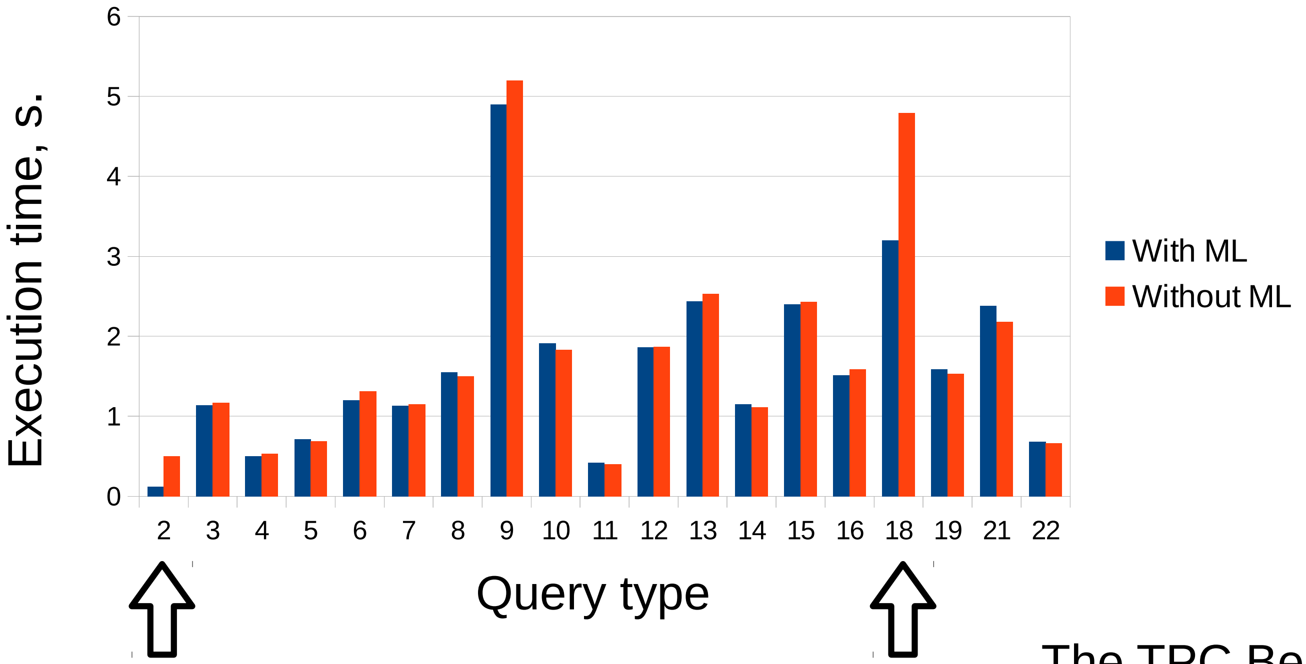
The TPC Benchmark™H (TPC-H)

http://www.tpc.org/tpch/





Obtained results: performance

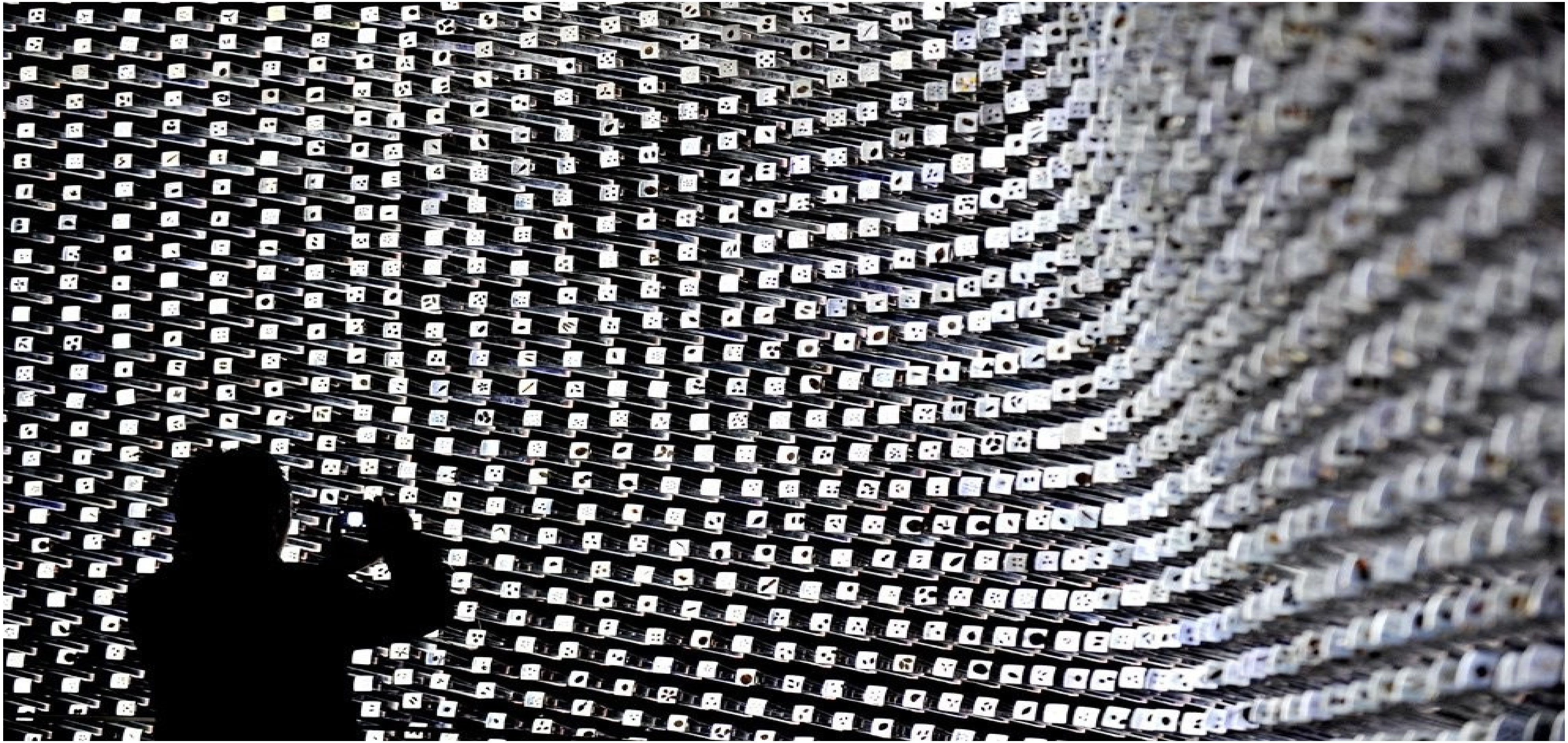


Dataset:
The TPC Benchmark™H (TPC-H)
http://www.tpc.org/tpch/

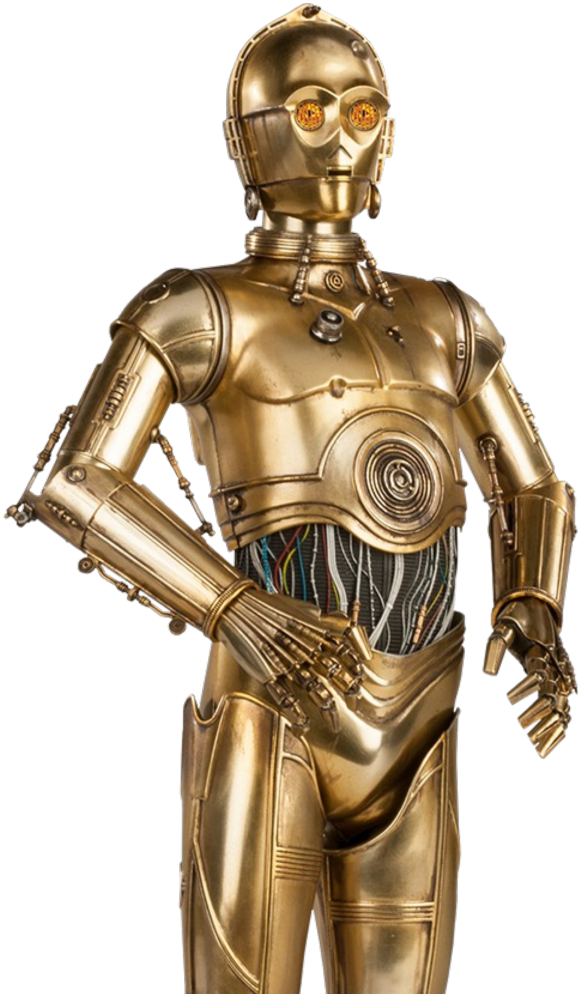
Possible implementations

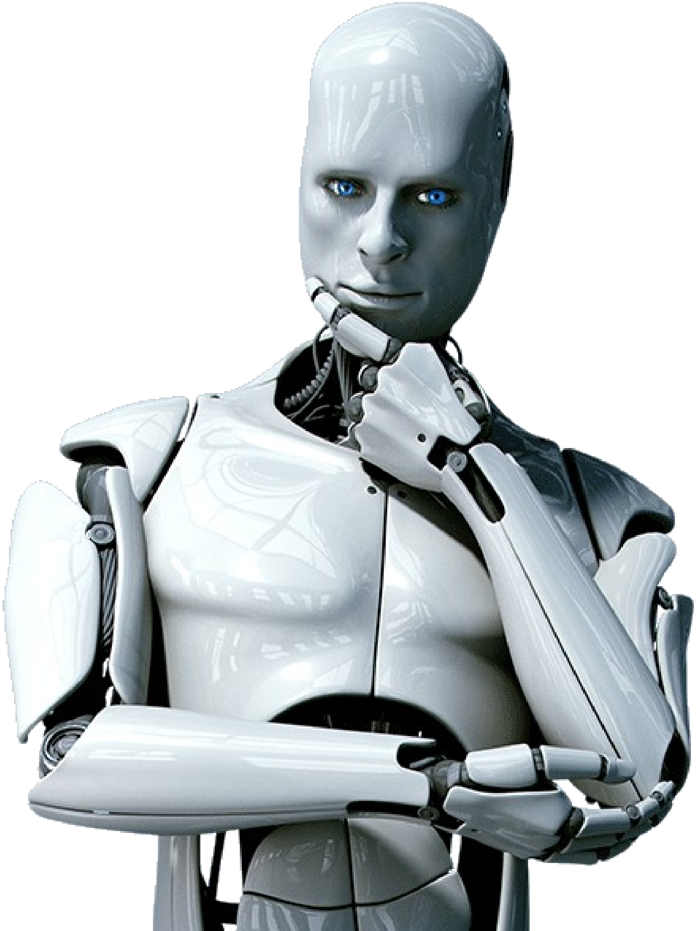
1. Online learning
2. Background learning
3. Smart **PREPARE**

Sample selection



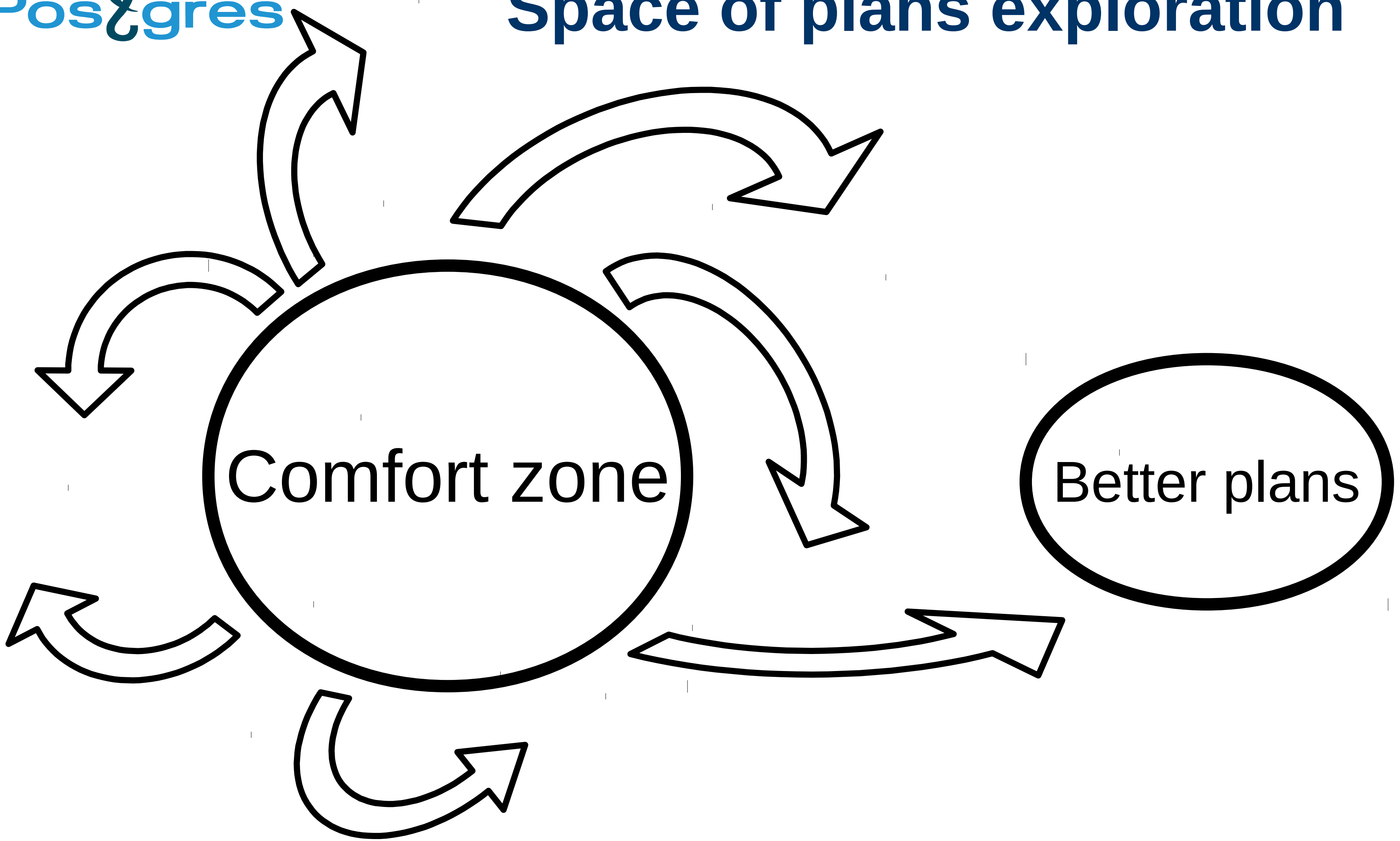
New machine learning approaches





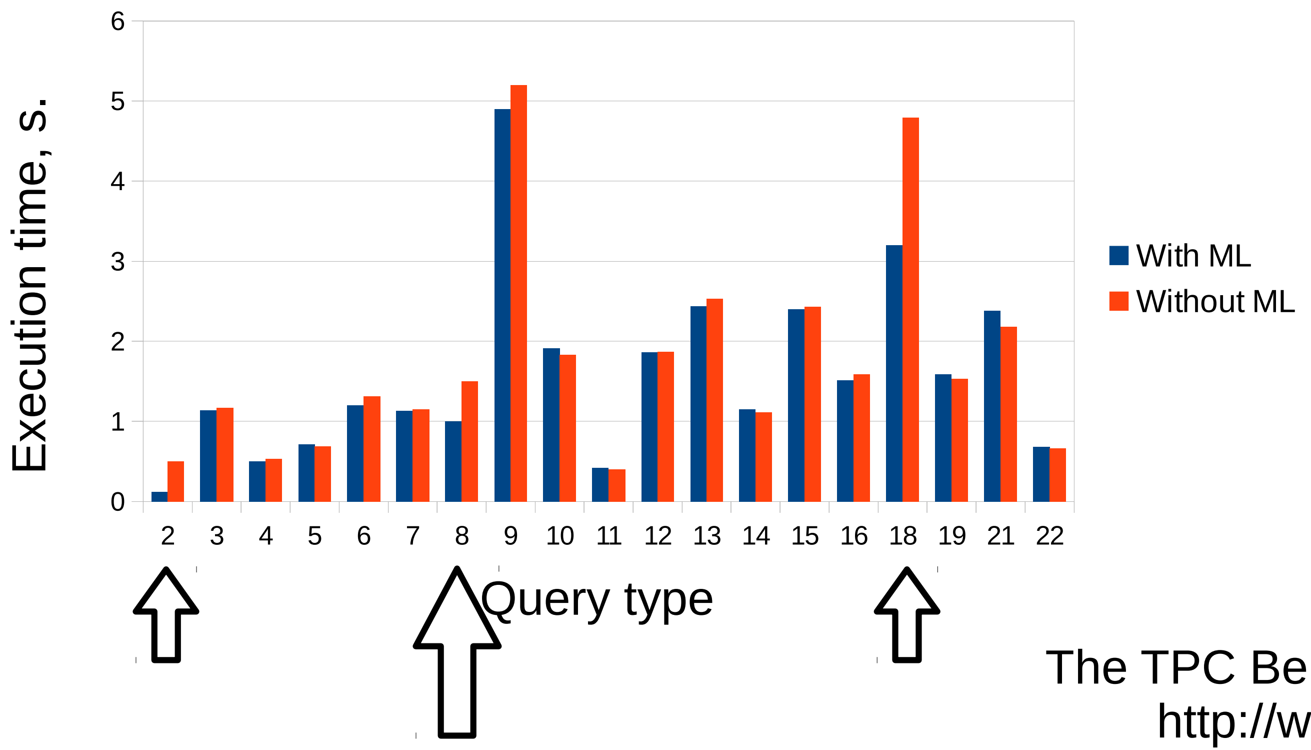
Realtime adaptation

Space of plans exploration



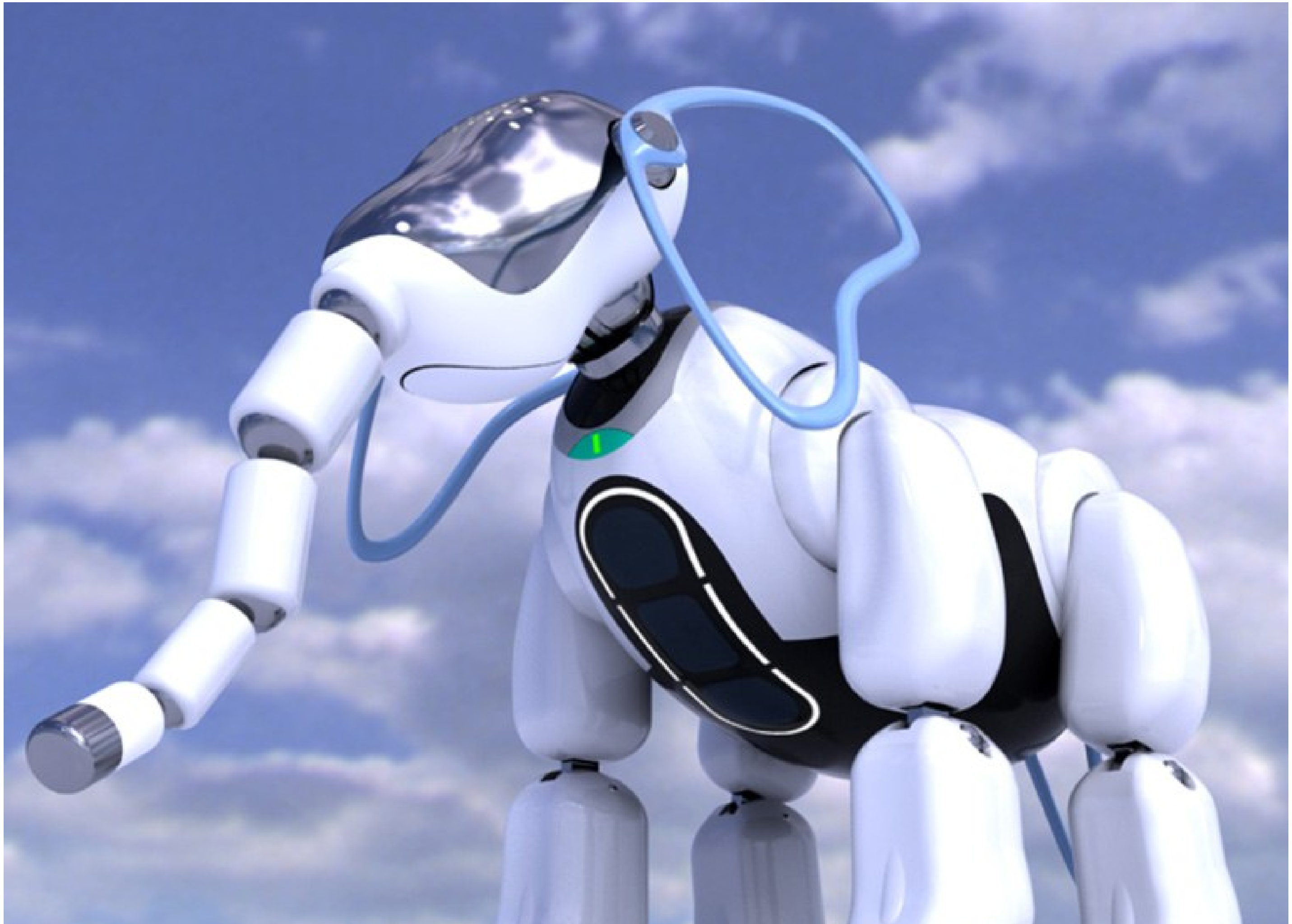
Space of plans exploration

Obtained results: performance acceleration



Dataset:
The TPC Benchmark™H (TPC-H)
http://www.tpc.org/tpch/

DBMS + ML = Better DBMS



Questions?

http://habrahabr.ru/company/postgrespro/blog/273199/
http://tigvarts.livejournal.com/691.html

Contacts:

o.ivanov@postgrespro.ru

+7 (916) 377-55-63