

The DB2Night Show Episode #256

Machine Learning Optimizer Feature in Db2 **Calisto Zuzarte**

Fri, Jun 23, 2023 10:00 AM - 11:15 AM CDT

Agenda

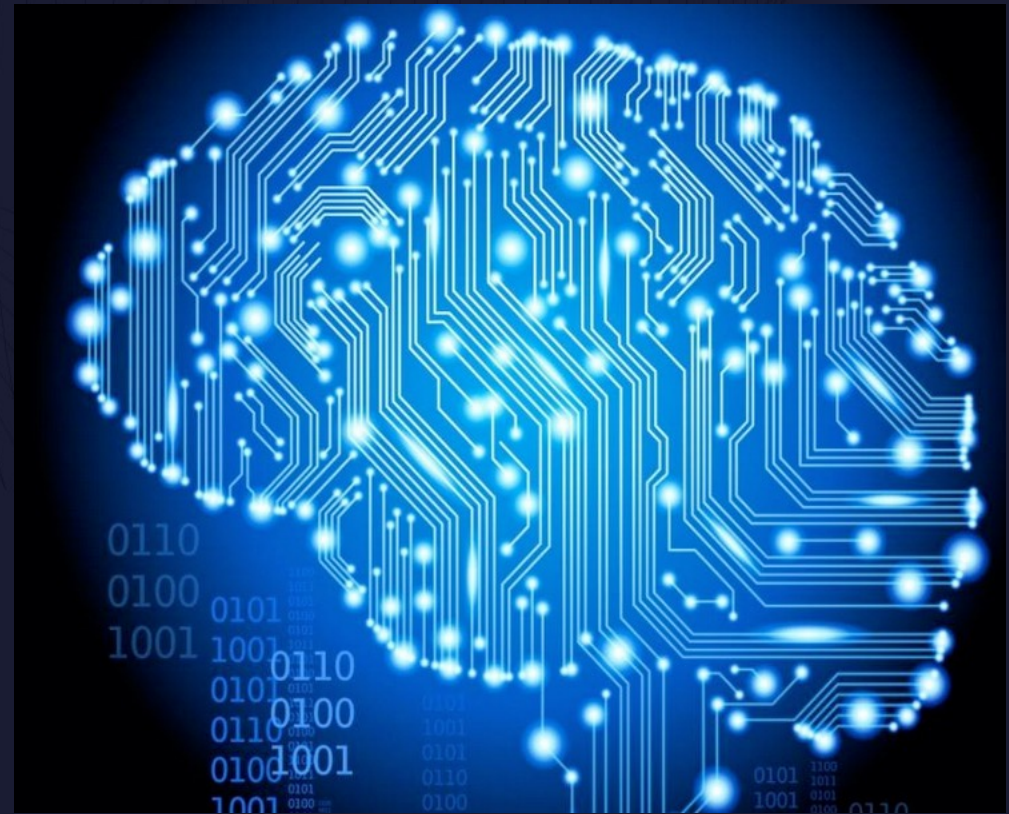
- Motivation
- Tech Preview
- Plan for Productization



Motivation

Artificial Intelligence (AI)

Artificial Intelligence is the simulation of human intelligence in machines that are programmed to think like humans.



Machine Learning (ML)

Machine Learning provides AI systems the ability to automatically learn and improve from experience without being explicitly programmed.

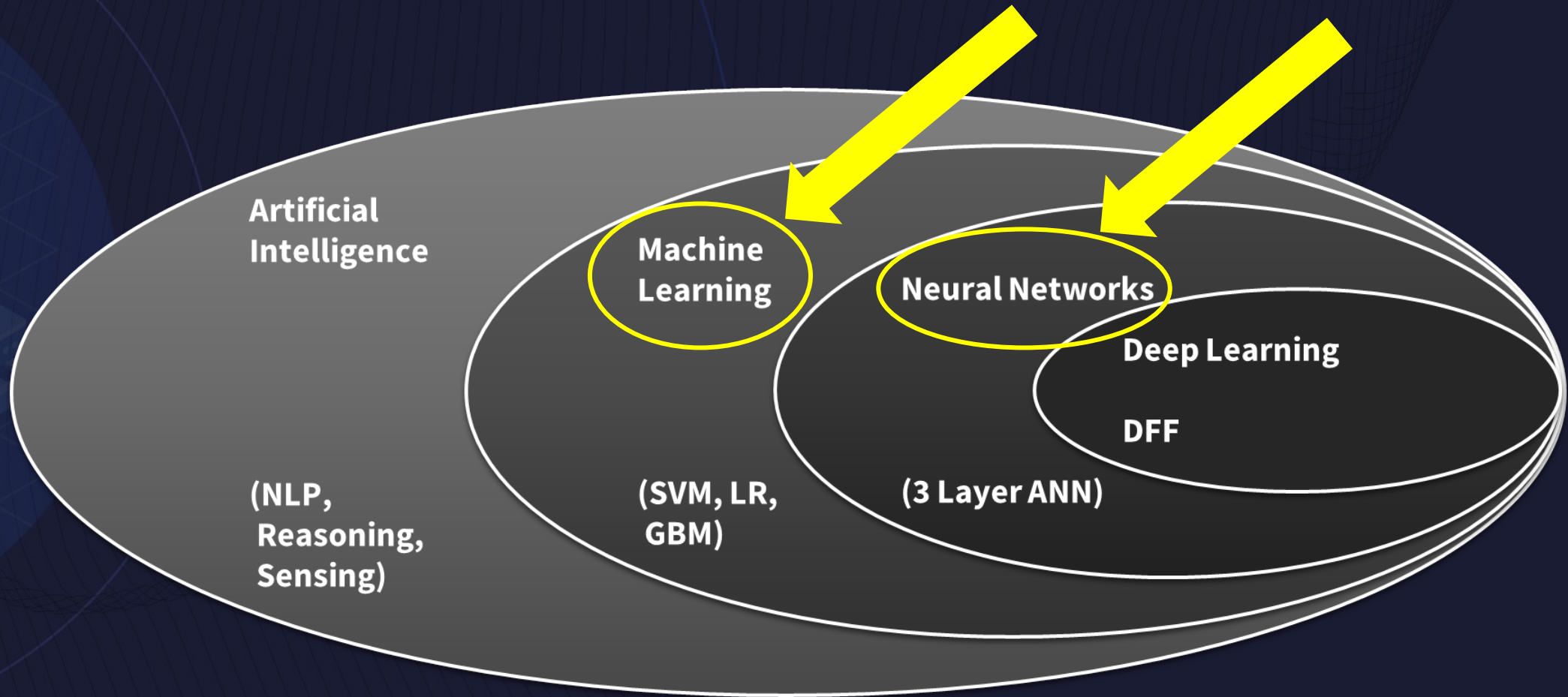


Neural Network (NN)

A Neural Network is a series of algorithms that tries to recognize underlying relationships in a set of data using interconnected nodes much like neurons in a human brain



Putting it Together



Motivation – Optimizer Challenges

- Stability with default statistics and simplifying assumptions
- Tuning effort with advanced statistics
- Optimizer Development effort

Motivation – Benefits of Machine Learning

- Adaptability to specific customer
 - Data
 - Workload
 - Environment
- Can exploit optimizer and run time feedback to improve the model.

Motivation – ML Optimizer Goals

- Automate everything
- Achieve reliable performance
- Simplify the optimizer development
- Infuse ML gradually

Motivation – Infuse AI Gradually

- Cardinality estimation for common **Local Predicates**
 - Equality, Range, BETWEEN, IN, OR
- Join cardinality estimation
 - Pairwise Joins Equality, Multiple Join Predicates, Multiple Joins
- Enhance cardinality estimation
 - Commonly used expressions, Parameter Markers, Group By etc.,
- Join planning
- Other aspects

Motivation - Why Start With Cardinality Estimation?

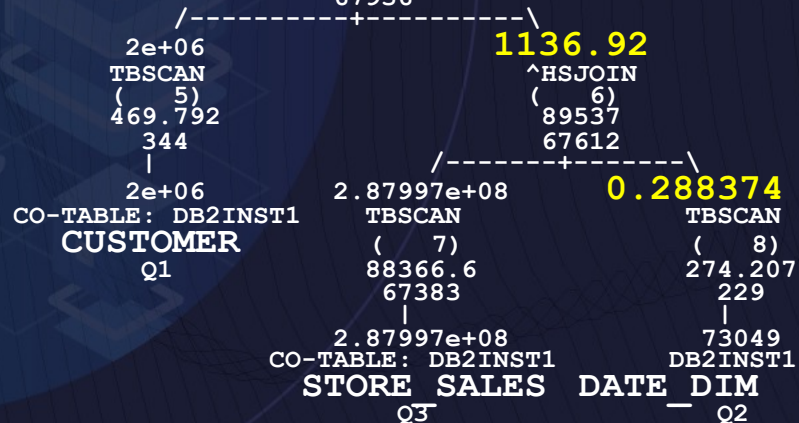
- Cardinality Estimation is the number of rows input to or output from an operator
- Critical for cost-based optimizers
- Primary source of query performance problem issues from customers

Motivation - Tuning For Good Cardinality Estimates (4 | 4)

Actual : 10,113,972

1136.92

1000X off !!!



Default Statistics

457723

20X off !!!

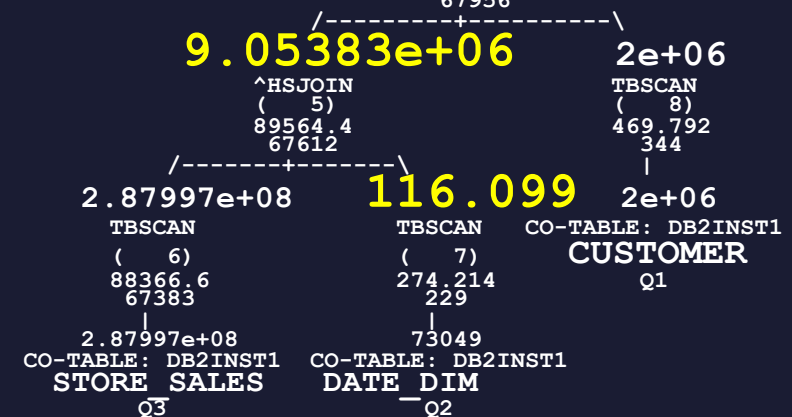


With additional Column Group Statistics

Close Estimate and Better Plan



9.05383e+06

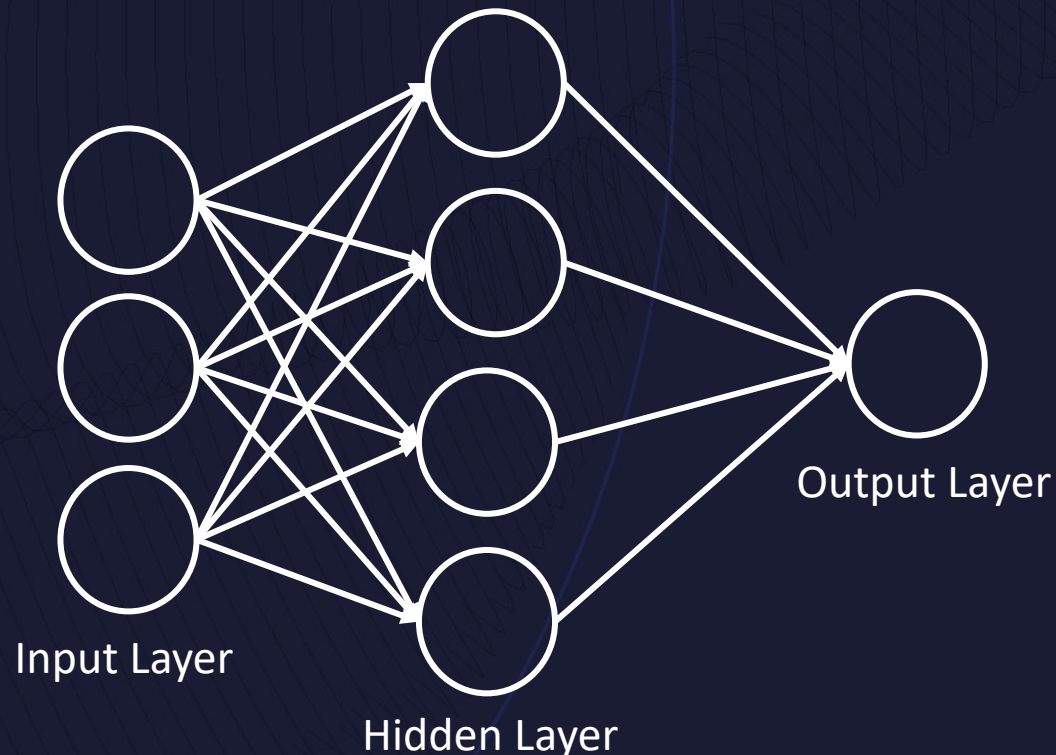


With additional Statistical Views

Can a Model Learn Cardinality Estimation

“Cardinality Estimation Using Neural Networks” CASCON 2015: 53-59

- Henry Liu, Mingbin Xu, Ziting Yu, Vincent Corvinelli, Calisto Zuzarte
- <https://dl.acm.org/citation.cfm?id=2886453>



The background is a dark blue gradient with a subtle pattern of thin, curved lines. On the left side, there is a circular inset containing a faint, light blue geometric pattern of squares and lines. The text "Tech Preview" is centered in the lower half of the image.

Tech Preview

Tech Preview - Documentation


- <https://www.ibm.com/support/pages/machine-learning-optimizer-technology-preview-db2-1156>
- Send questions and feedback to calisto@ca.ibm.com

Tech Preview – Try this on a Test System

- **Enabling the ML Optimizer**
 - `db2set DB2_ML_OPT="ENABLE:ON"`
 - `db2 -tf MLOptimizerCreateTables.ddl`
 - Needs Auto-RUNSTATS
- **Toggle to look at how the traditional Optimizer does:**
 - `db2set -im DB2_SELECTIVITY="ML_PRED_SEL OFF"`
- **Disabling the ML Optimizer**
 - `db2set DB2_ML_OPT="ENABLE:OFF"`
 - drops all the models

Tech Preview – Try this on a Test System


```
SELECT * FROM T1, T2
WHERE
  T1.C1 = 'abc' AND
  T1.C6 IN (5, 3, 205) AND
  T1.C2 BETWEEN 5 AND 10 AND
  T2.C3 <= 120 AND
  ((T1.C4 > 5 AND T1.C5 < 20) OR
   (T1.C4 < 2 AND T1.C5 = 100)) AND
  T1.C0 = T2.C0 AND
  T1.C3 = ? AND
  MOD(T1.C4, 10) = 1;
```



Local Predicates with Equality,
Range, Between , IN, OR



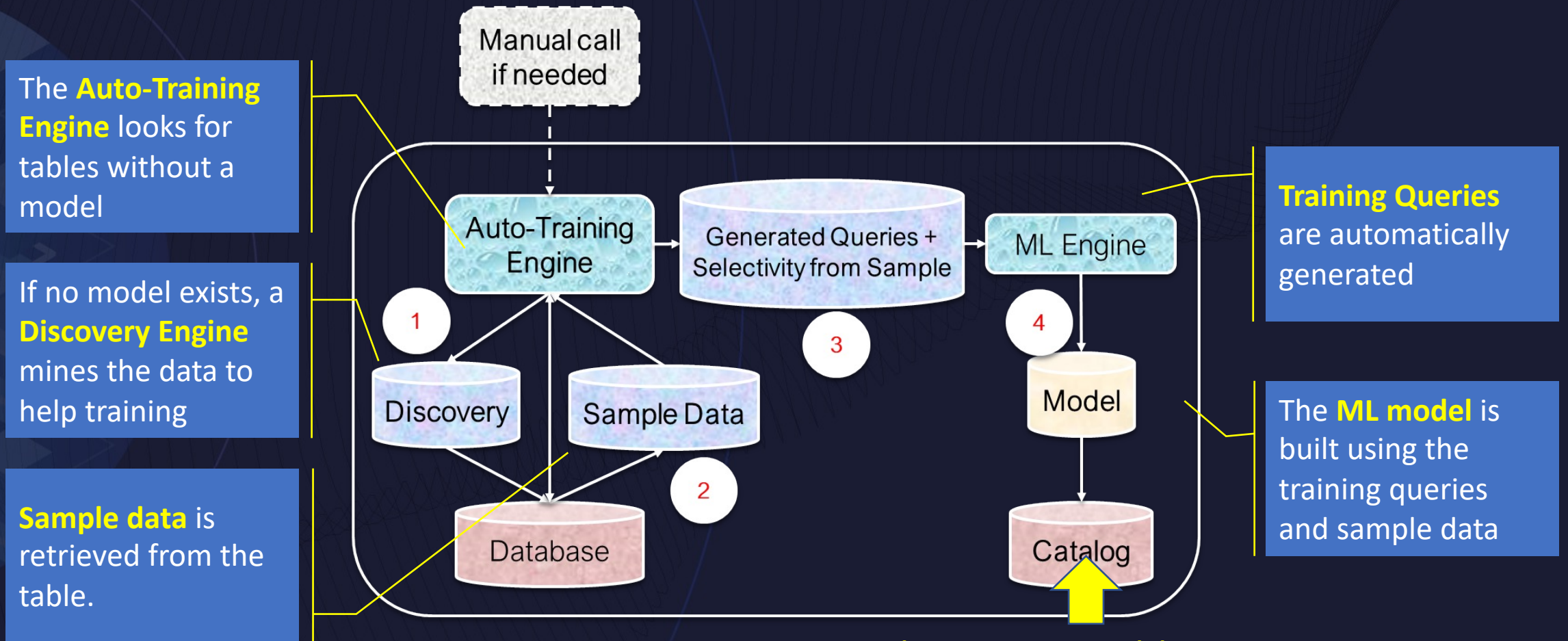
Pair-Wise Join Predicates



Predicates With Parameter Markers
Predicates With Expressions

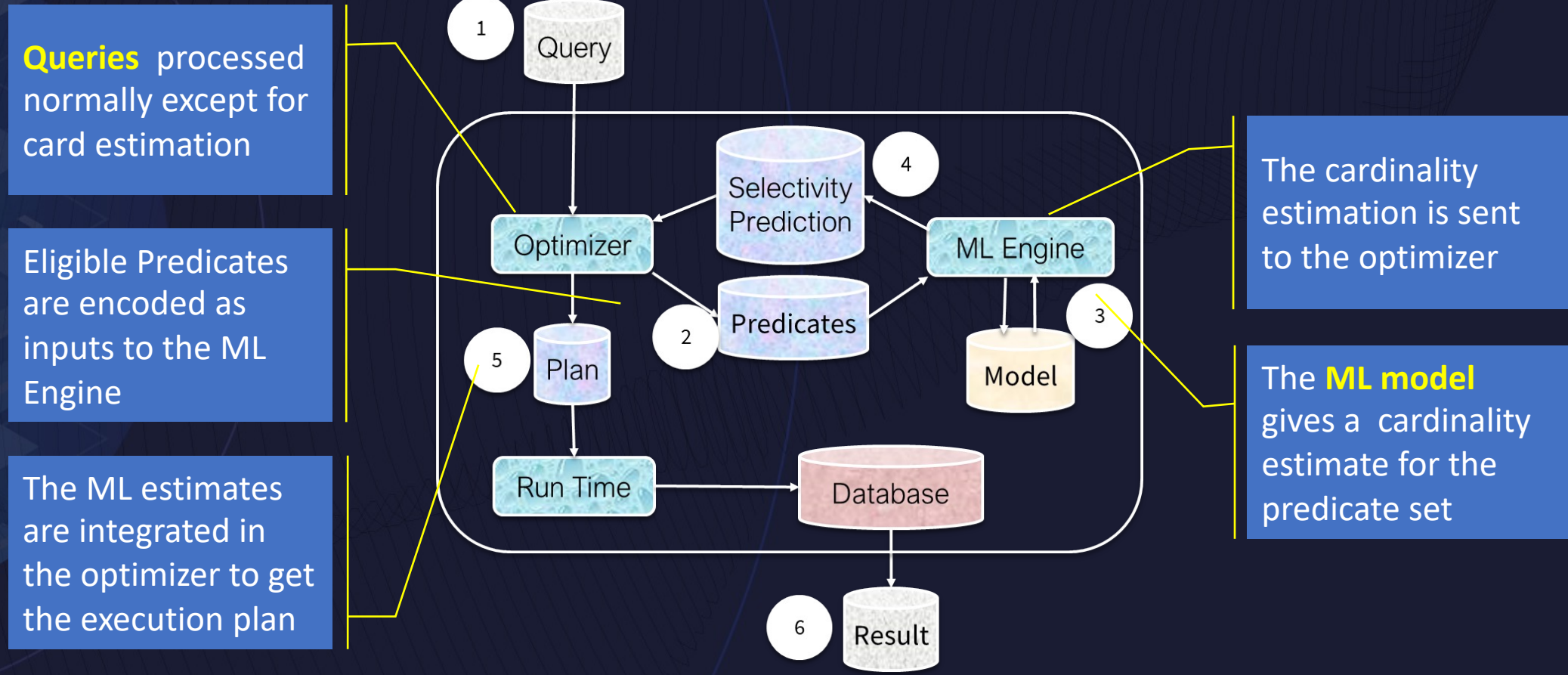
- These will be processed by the traditional optimizer

Tech Preview – Automatic Training



Tech Preview: Table in SYSTOOLS

Tech Preview – Cardinality Estimation Using the Model



Queries processed normally except for card estimation

Eligible Predicates are encoded as inputs to the ML Engine

The ML estimates are integrated in the optimizer to get the execution plan

The cardinality estimation is sent to the optimizer

The **ML model** gives a cardinality estimate for the predicate set

Tech Preview – Automatic Feedback

- Like Auto-RUNSTATS, table data change counters are used to trigger retraining
- No optimizer or run time feedback in the Tech Preview

Model Size and Training Time

NN **Model Size** is significantly better than with LGBM

NN Model size is 1000X better !
30KB versus 30MB

Accuracy, (not shown here) is a little better with LGBM than with NN

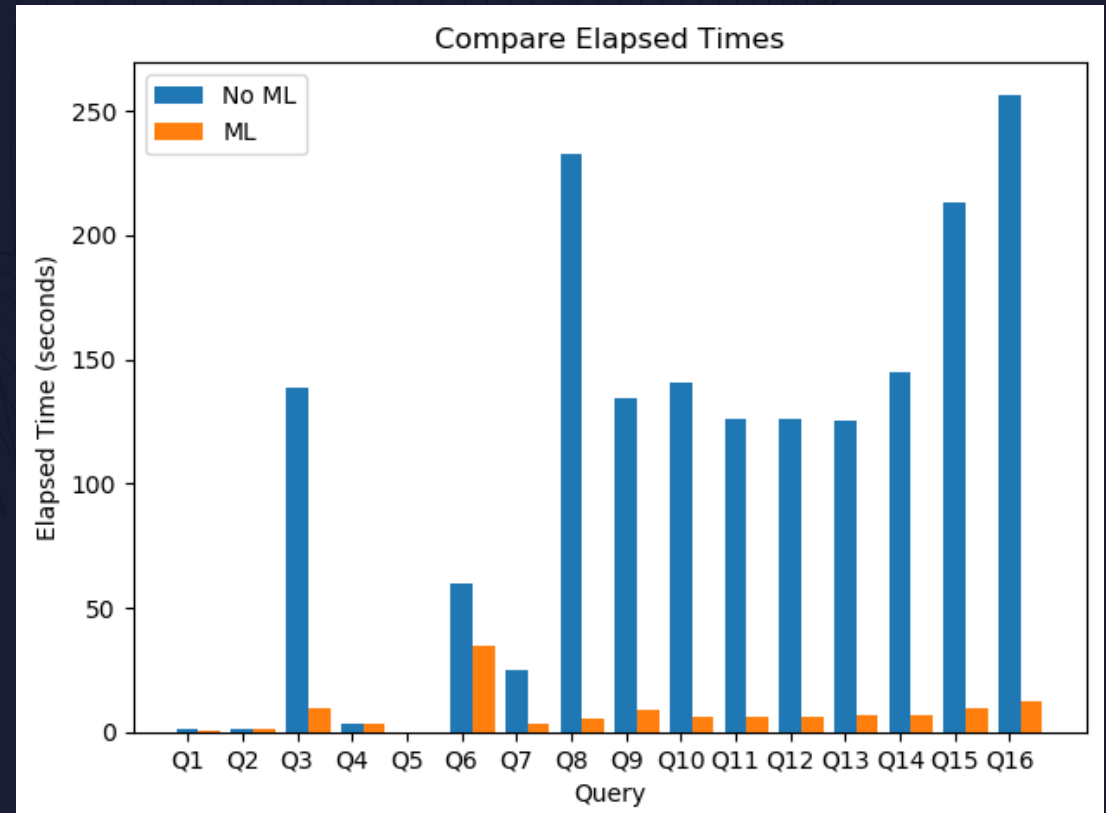
TABLENAME	MODEL SIZE (MiB)		TRAINING TIME (S)	
	NN	LGBM	NN	LGBM
CALL_CENTER	0.021	0.003	0	2
CATALOG_PAGE	0.022	33.401	60	94
CATALOG_RETURNS	0.037	32.742	67	358
CATALOG_SALES	0.037	32.745	103	376
CUSTOMER	0.024	33.147	37	358
CUSTOMER_ADDRESS	0.023	33.717	34	89
DATE_DIM	0.037	33.176	43	362
INCOME_BAND	0.021	0.066	1	2
ITEM	0.030	6.432	68	307
PROMOTION	0.022	13.707	480	14
REASON	0.021	0.146	9	1
SHIP_MODE	0.021	0.182	28	2
STORE	0.022	0.422	46	2
STORE_RETURNS	0.024	32.763	47	361
STORE_SALES	0.037	32.865	68	342
TIME_DIM	0.022	1.861	34	80
WAREHOUSE	0.021	0.003	0	1
WEB_PAGE	0.022	7.889	40	3
WEB_RETURNS	0.037	32.767	82	347
WEB_SALES	0.037	32.757	82	368
WEB_SITE	0.024	2.650	46	6

Training Time is also better with NN compared to LGBM

Training time is 5X less than LGBM
5 m versus 1 m

Tech Preview – Problem Scenarios

- Real-world problematic queries reported by a couple of customers
- 10X benefit in some of these scenarios simulated in-house
- Note that in practice the average benefit will be less
- The goal is to get more reliable performance.



Tech Preview – Problem Scenario (Q10)

```
SELECT
  IH.AMOUNT,
  CHD.COMMENTS
FROM
  DEMO.PURCHASE_HISTORY PH,
  DEMO.INSURANCE_HISTORY IH,
  DEMO.CREDIT_HISTORY_DATA CHD,
  DEMO.SENTIMENT_SCORE_DATA SSD,
  DEMO.POLICE_DATA PD
LEFT OUTER JOIN
  (SELECT EMAILID
   FROM DEMO.PURCHASE_HISTORY PH1
   WHERE PH1.PURCHASE_DATE BETWEEN '2018-12-30' and '2018-12-31') X
ON PD.EMAILID = X.EMAILID
WHERE
  PH.INSURANCE_ID = IH.INSURANCE_ID AND
  PH.PURCHASE_DATE BETWEEN '2014-01-01' AND '2019-12-31' AND
  PD.EMAILID = PH.EMAILID AND
  PD.CRIMINAL_RANK > .4 AND
  PD.EMAILID = SSD.EMAILID AND
  SSD.SCORE < .7 AND
  PH.EMAILID = CHD.EMAILID AND
  CHD.PAY_0 BETWEEN 0 AND 2 AND
  CHD.PAY_2 BETWEEN 0 AND 2 AND
  CHD.PAY_3 BETWEEN 0 AND 2 AND
  CHD.PAY_5 BETWEEN 0 AND 2 AND
  CHD.PAY_6 BETWEEN 0 AND 2 AND
  CHD.PAY_4 BETWEEN 0 AND 2 AND
  CHD.BILL_AMT1 BETWEEN 150 AND 746814 AND
  CHD.BILL_AMT2 BETWEEN 0 AND 743970 AND
  CHD.BILL_AMT3 BETWEEN 0 AND 689643 AND
  CHD.BILL_AMT4 BETWEEN 0 AND 706864
```

```
CHD.PAY_0 BETWEEN 0 AND 2 AND
CHD.PAY_2 BETWEEN 0 AND 2 AND
CHD.PAY_3 BETWEEN 0 AND 2 AND
CHD.PAY_5 BETWEEN 0 AND 2 AND
CHD.PAY_6 BETWEEN 0 AND 2 AND
CHD.PAY_4 BETWEEN 0 AND 2 AND
CHD.BILL_AMT_1 BETWEEN 150 AND 746014 AND
CHD.BILL_AMT_2 BETWEEN 0 AND 743970 AND
CHD.BILL_AMT_3 BETWEEN 0 AND 689643 AND
CHD.BILL_AMT_4 BETWEEN 0 AND 706864
```


Tech Preview – Interesting Scenarios

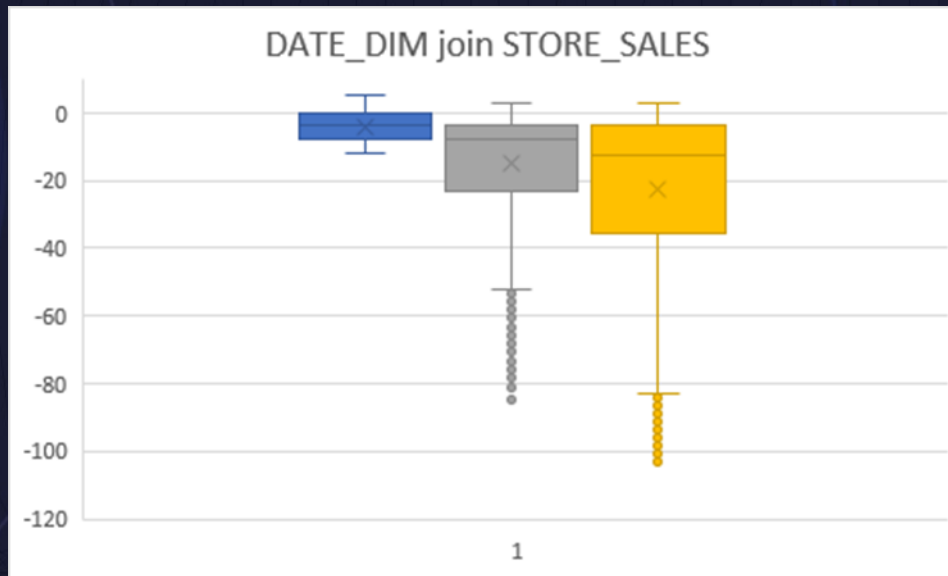
- ```
SELECT GUEST_LAST_NAME, ARRIVAL_DATE, DEPARTURE_DATE
FROM HOTEL_DB
WHERE (ARRIVAL_DATE <= '2019-12-25' and DEPARTURE_DATE >= '2019-12-25') OR
 (ARRIVAL_DATE <= '2018-12-25' and DEPARTURE_DATE >= '2018-12-25') OR
 (ARRIVAL_DATE <= '2017-12-25' and DEPARTURE_DATE >= '2017-12-25')
```
- ```
SELECT GUEST_LAST_NAME, ARRIVAL_DATE, DEPARTURE_DATE
FROM HOTEL_DB
WHERE DATE_COL BETWEEN '2019-08-01' and '2019-08-31') AND
      COMPANY = 'IBM'
```

Join Cardinality – Single Table Model

For both plots : (1) Closer to 0 is better (2) Thinner box is better

ML

No ML, Basic Statistics + CGS



N:1 JOIN - ONE JOIN PREDICATE



M:N JOIN - THREE JOIN PREDICATES



Plan For Production

Plan for Productization – Infrastructure Enhancements

- New system catalog table to store system AI models
 - SYSAIMODELS
 - Appropriate SYSCAT view for individual model type
- Security / access control
- Audit

Plan for Productization - Usability Enhancements

- Explain support
- Activity logging
- Better model management
- Better configuration management
- Appropriate error messages
- DDL to manage models
- Appropriate dependency management

Plan for Productization – Model Enhancements

- Improved ML model size : **~ 20 Kb to 30 Kb per table**
- Increased number of columns (**up to 20 instead of 10**) strongly correlated columns
- Improved Training Time : **~ 1 to 2 Minutes**
 - Dependent on number and characteristics of the columns included
 - Not so dependent on the table size

Plan for Productization – Configuration

Automatic maintenance

Automatic database backup

Automatic table maintenance

Automatic runstats

Real-time statistics

Statistical views

Automatic sampling

Automatic column group statistics

Automatic reorganization

Automatic AI maintenance

Machine Learning Optimizer

Automatic Model Discovery

(AUTO_MAINT) = ON

(AUTO_DB_BACKUP) = OFF

(AUTO_TBL_MAINT) = ON

(AUTO_RUNSTATS) = ON

(AUTO_STMT_STATS) = ON

(AUTO_STATS_VIEWS) = OFF

(AUTO_SAMPLING) = ON

(AUTO_CG_STATS) = OFF

(AUTO_REORG) = OFF

(AUTO_AI_MAINT) = ON

(AUTO_ML_OPTIMIZER) = ON

(AUTO_ML_DISCOVER) = ON

Summary

- Infusing AI in the Db2 Optimizer is strategic
- Please try out the Tech Preview on your test system
 - <https://www.ibm.com/support/pages/machine-learning-optimizer-technology-preview-db2-1156>
 - Send questions and feedback to calisto@ca.ibm.com
- ML cardinality estimation plan for GA (vNext)
 - Initially with the local predicate model only and
 - Appropriate infrastructure for all future AI models for use within Db2

Thank You

Speaker: Calisto Zuzarte

Company: IBM

Email Address: calisto@ca.ibm.com