Lecture #15

Carnegie Mellon University

Special Topics:

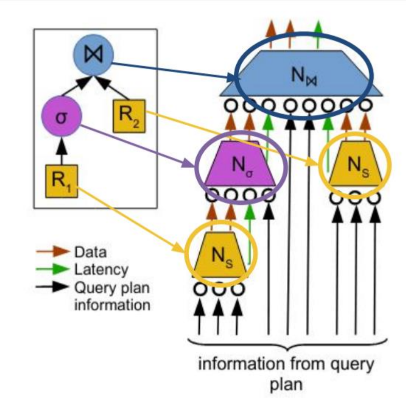
Self-Driving Database Management Systems

Training Data Collection

@Dhruv_Arya // 15-799 // Spring 2022

LAST CLASS

QPPNet
→ Hierarchical Neural Networks for behaviour modeling
→ Query plan modeled at both the operator-level and the plan-level
→ Easy to transfer to new queries as long as the query uses operators that have already been learnt



Challenges

→ Lots of data needed to train models

→ Current data sources -> Manually configured templates for synthetic data generation or Real-world databases

→ Even when we have a benchmark -> generating labels (e.g. execution time, latency) is expensive

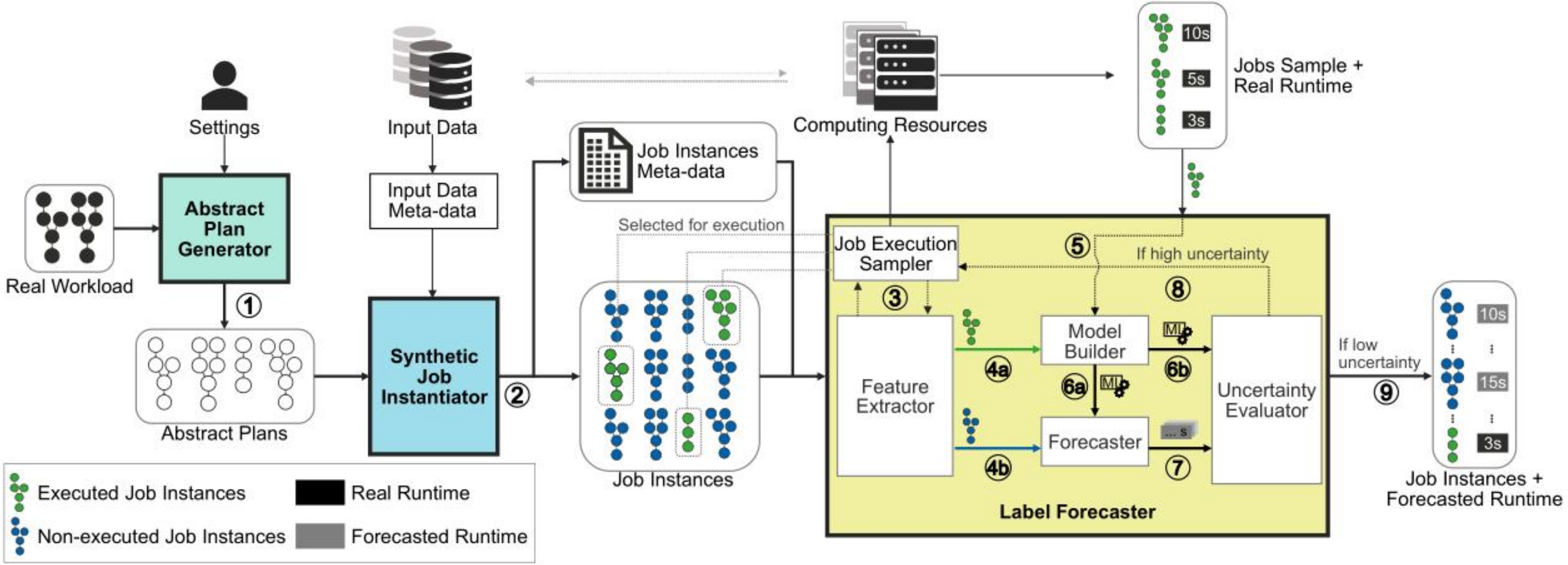
TODAY'S AGENDA

→ Lots of data needed to train models
→ **Synthetically generate more data**

→ Current data sources -> Manually configured templates for synthetic data generation
→ **Automatically learn workload structures**

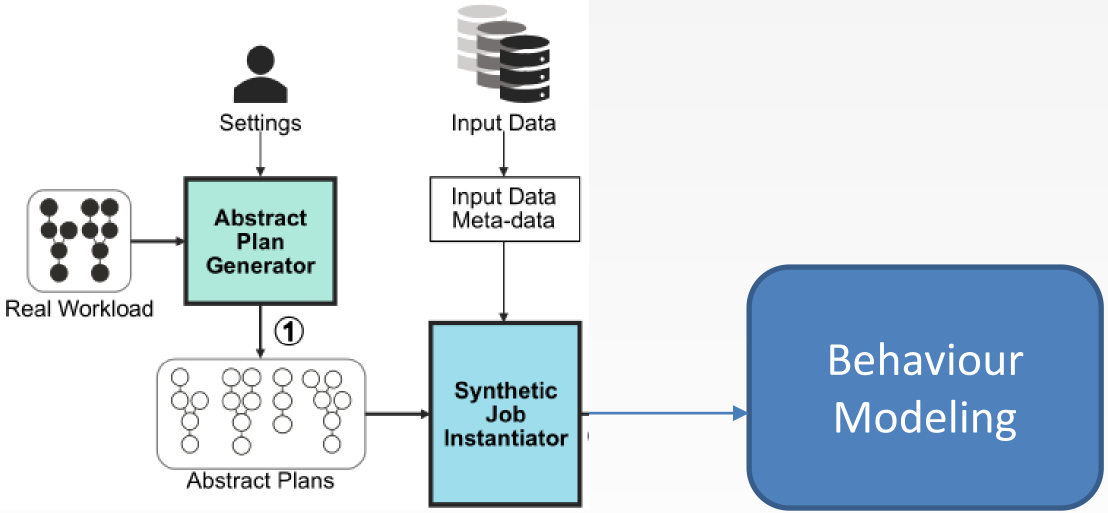
→ Even when we have a data source -> generating labels (e.g. execution time, latency) is expensive
→ **Forecast labels for data**

DataFarm – A High Level Overview



EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

High Level Overview



I. Abstract Plan Generator:

Generates a generic plan.

II. Synthetic Job Instantiator:

Picks the tables and the parameters.

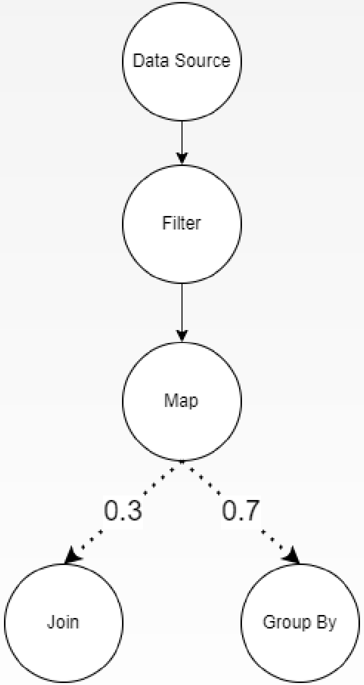
III. Behaviour Modeling:

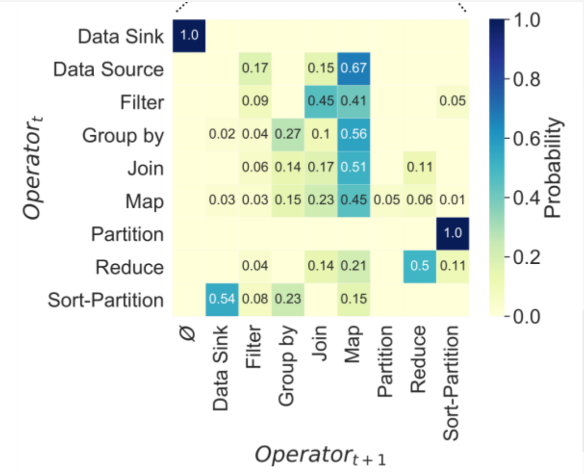
Predict query execution time.

EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Abstract Plan Generator

- → Model the plan as a Markov decision process
- → Build a transition probability graph from a given operator to child operators

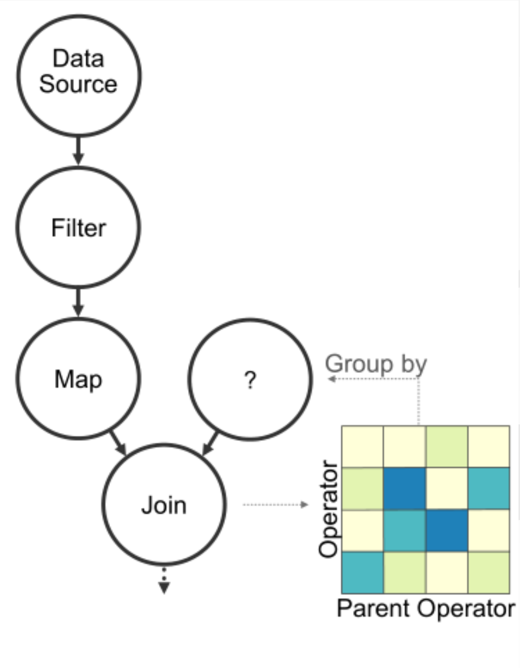




EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

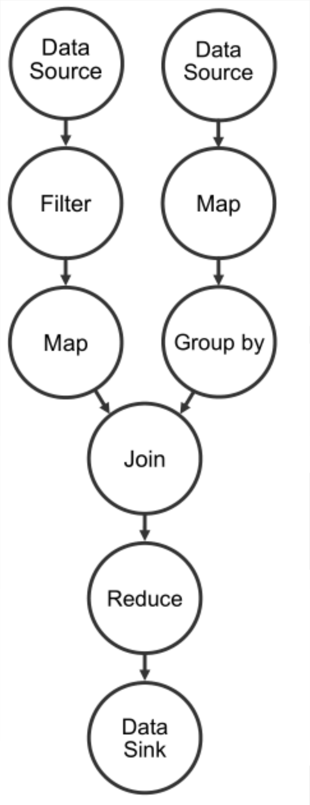
Abstract Plan Generator

- → Model the plan as a Markov decision process
- → Another transition probability matrix for transitioning into a parent operator
- → Comes into play when the Join operator is encountered
- → **Parent branches can't have additional joins**

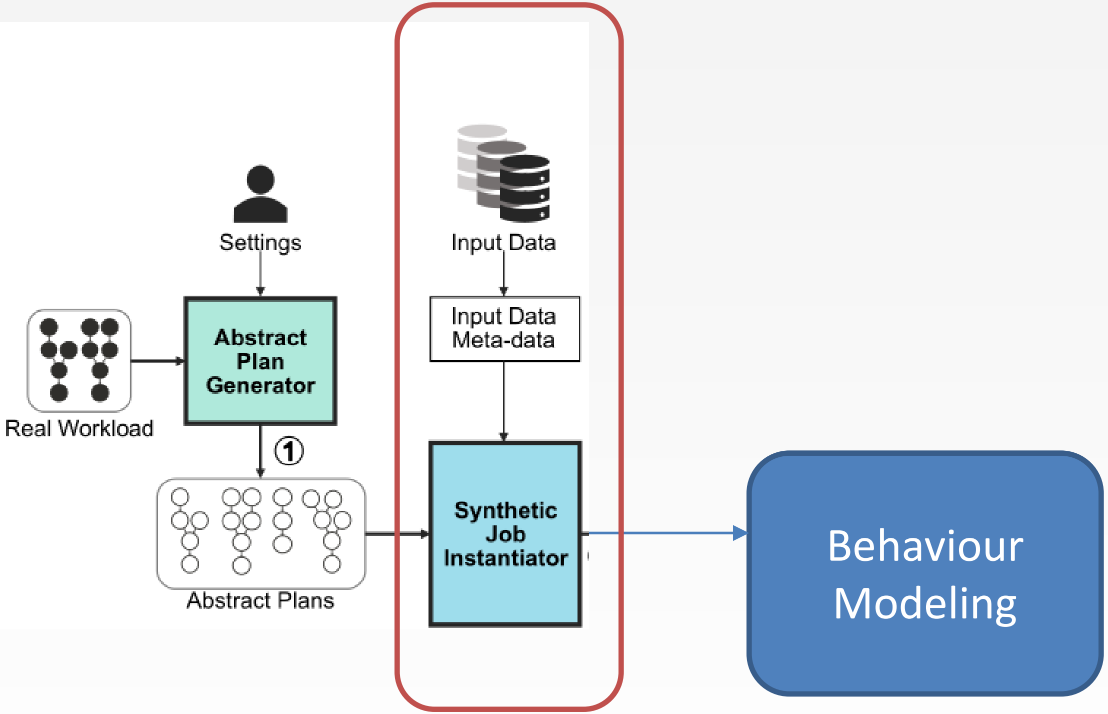


Abstract Plan Generator

→ Does not specify the tables nor the selectors and the parameters.
→ The abstract plan generator does not know the structure of the data



Plan Instantiation



Use meta-data to instantiate the abstract plans.

EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Plan Instantiator --- Input Data

→ Table Schema
→ Fields used for joins
→ Cardinality
→ Parameter values distribution

```
{
  "dataBaseID": "IMDB",
  "tables": [
    {
        "tableName": "title.rating",
        "rawCardinality": 99381,
        "typeSchema": "(String, float, int)",
        "fields": [
        "titleId",
        "averageRating",
        "numVotes"
        ],
        "filterFieldValue": {
            "releaseYear": {
                "selectivity": "0.95"
            },
            "values": [
                "1869.0"
            ]
        }
    }
  ]
}
```

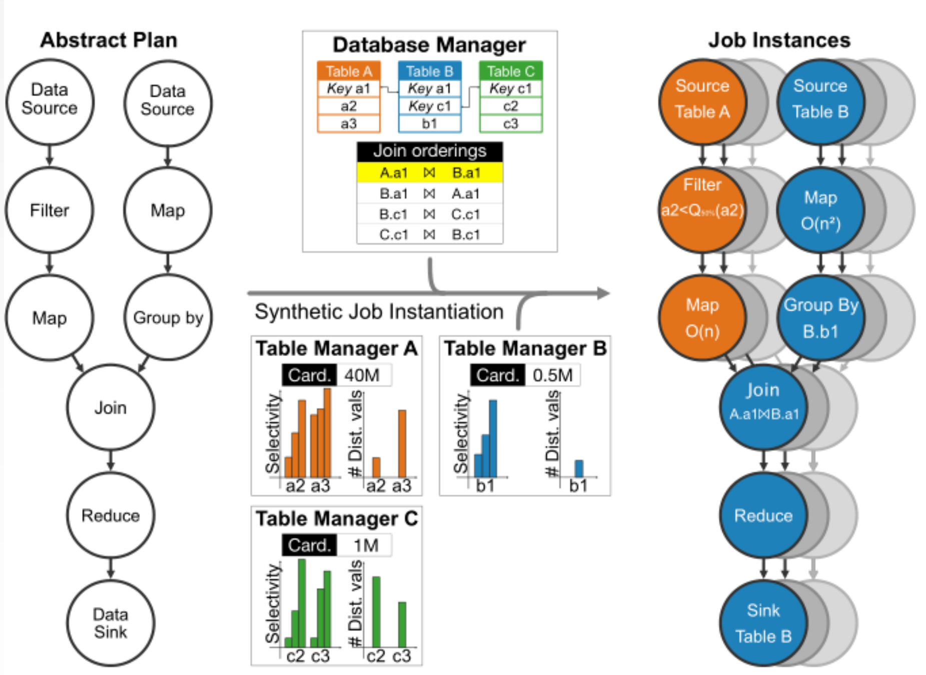
Plan Instantiator --- Input Data

Two Interfaces for the users:
→ Database Manager: Add new relations
→ Table Manager: Specify the table-specific statistics


```
{
  "dataBaseID": "IMDB",
  "tables": [
    {
        "tableName": "title.rating",
        "rawCardinality": 99381,
        "typeSchema": "(String, float, int)",
        "fields": [
        "titleId",
        "averageRating",
        "numVotes"
        ],
        "filterFieldValue": {
            "releaseYear": {
                "selectivity": "0.95"
            },
            "values": [
                "1869.0"
            ]
        }
    }
  ]
}
```

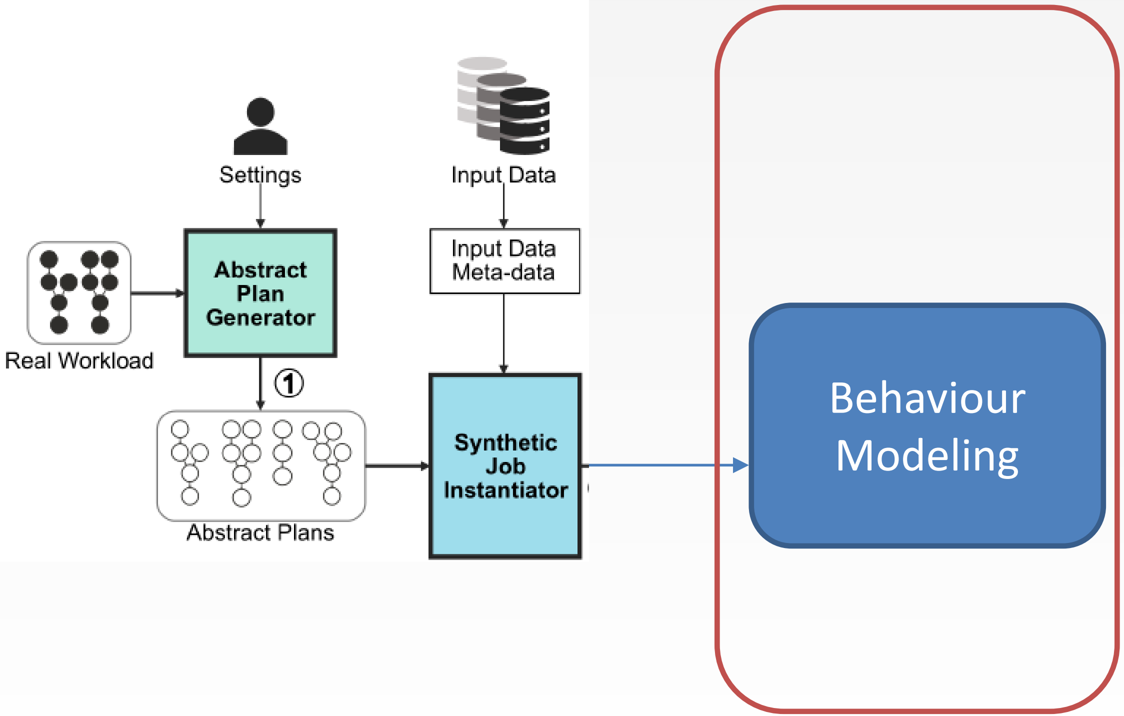
Plan Instantiator

→ Iteratively instantiate nodes from the Data Source to the Data Sink
→ Instantiate other sub-branch on encountering Join
→ Use statistics from Table Manager to instantiate parameter values



EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

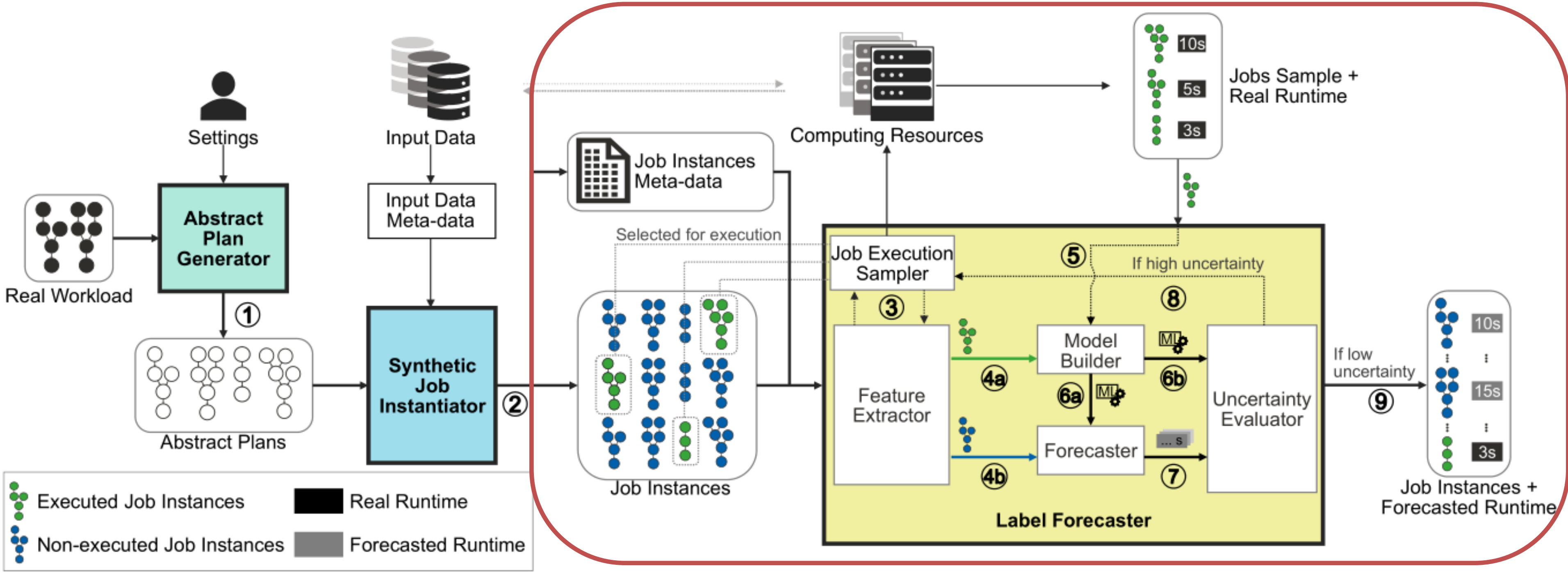
Behaviour Modeling



Efficiently predicting some label for the given data.

Some label = execution time

Behaviour Modeling/Label Forecaster



EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Label Forecasting – Feature Extraction

→ Tables used -> represented as a one-hot encoding
→ Operator-level statistics
→ Complexity of operators computed
→ Map operator complexity = **Number of rows scanned x Number of fields**
→ Cardinality estimates based on user-supplied data
→ Principal Component Analysis for identifying features with most relevant variance

Label Forecasting – Model Training

→ Train a Quantile Regression Forest Model
→ Gives the distribution of the output variable given the input variable

→ Prediction value: L

→ Prediction Interval: $[u_{low}(\hat{l}), u_{high}(\hat{l})]$

Algorithm 1: Active labeling.

Input : jobs instances J; computational resources R; number of init. jobs κ; threshold η; early stopping threshold λ; max iterations MAX_i
Output : set of labels L

```
L ← ∅; L_ex ← ∅; L̂_noex ← ∅; J_ex ← ∅; J_noex ← ∅; U ← ∅; i ← 0;
F ← FeatureExtractor.transform(J);
S_ex ← JobExecSampler.initialize(F, κ);
while not earlyStop(U, λ) or i < MAX_i do
    L_ex ← L_ex ∪ R.submit(S_ex);
    J_ex ← J_ex ∪ S_ex;
    J_noex ← J \ J_ex;
    M ← ModelBuilder(F[J_ex], L_ex);
    L̂_noex ← Forecaster(F[J_noex], M);
    u ← UncertaintyEstimator(L̂_noex, M);
    U ← U ∪ {u};
    S_ex ← JobExecSampler.nextExecs(J_noex, u, η);
    i ← i + 1;
end
L ← L_ex ∪ L̂_noex;
return L
```

EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Model Training

→ Active Learning for the training:

→ Start by executing initial few jobs to get ground truth labels

→ Iteratively select jobs such that the distribution of high-, medium-, and low-cardinality jobs is similar

Algorithm 1: Active labeling.

Input : jobs instances J; computational resources R; number of init. jobs κ; threshold η; early stopping threshold λ; max iterations MAX_i

Output : set of labels L

$L \leftarrow \emptyset$; $L_{ex} \leftarrow \emptyset$; $\hat{L}_{noex} \leftarrow \emptyset$; $J_{ex} \leftarrow \emptyset$; $J_{noex} \leftarrow \emptyset$; $U \leftarrow \emptyset$; $i \leftarrow 0$;
$F \leftarrow$ FeatureExtractor.transform(J);
$S_{ex} \leftarrow$ JobExecSampler.initialize(F, κ);
while *not earlyStop*(U, λ) **or** $i < MAX_i$ **do**
    $L_{ex} \leftarrow L_{ex} \cup R$.submit($S_{ex}$);
    $J_{ex} \leftarrow J_{ex} \cup S_{ex}$;
    $J_{noex} \leftarrow J \setminus J_{ex}$;
    $M \leftarrow$ ModelBuilder($F[J_{ex}]$, L_{ex});
    $\hat{L}_{noex} \leftarrow$ Forecaster($F[J_{noex}]$, M);
    $u \leftarrow$ UncertaintyEstimator($\hat{L}_{noex}$, M);
    $U \leftarrow U \cup \{u\}$;
    $S_{ex} \leftarrow$ JobExecSampler.nextExecs(J_{noex}, u, η);
    $i \leftarrow i + 1$;
end
$L \leftarrow L_{ex} \cup \hat{L}_{noex}$;
return L

Experimental Setup

- → Apache Flink
- → 4 x Intel Xeon 2.40GHz CPU +16 GB RAM
- → IMDB Dataset – 5 Tables
- → TPC-H:
  - a. 1GB, 5GB, 10GB, 50GB ---
  - b. Q1, Q3, Q11, Q13, Q17, and Q21 only
  - c. Implemented in Flink

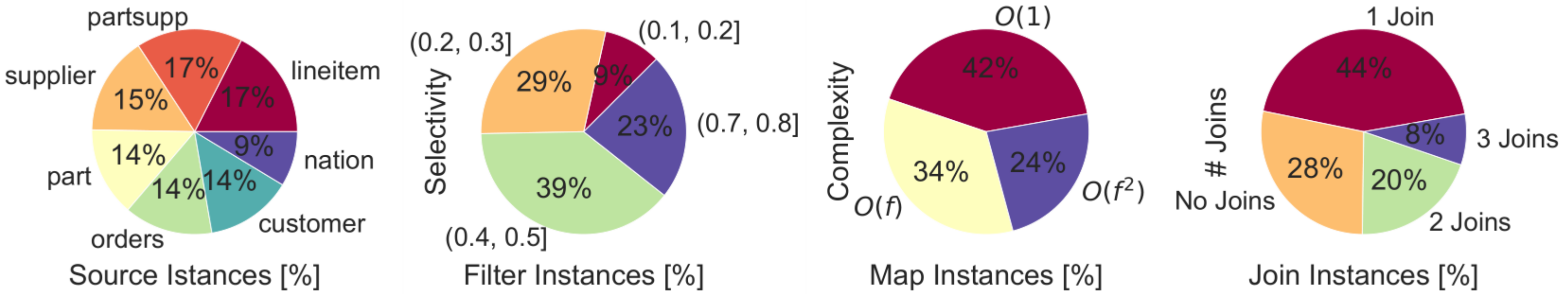
https://github.com/agora-ecosystem/data-farm/blob/main/data/input_workload_exec_plan/20200310_170524-TPC_H_Q1-SQL.json

```
select
    l_returnflag,
    l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
    sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
from
    lineitem
where
    l_shipdate <= date '1998-12-01' - interval ':1' day (3)
group by
    l_returnflag,
    l_linestatus
order by
    l_returnflag,
    l_linestatus;
:n -1
```

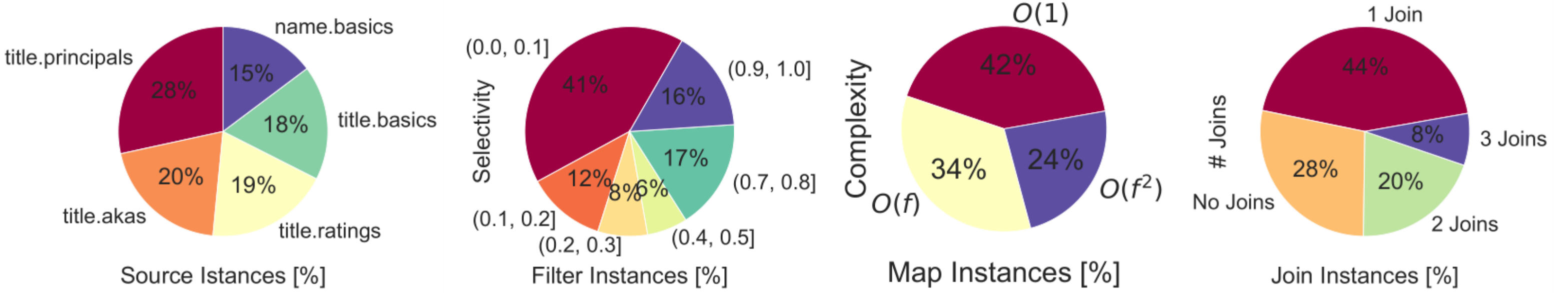
Generated Workloads

- → W1: 2000 jobs based on TPC-H<- 50 Abstract plans per dataset -> 10 instantiations each * four TPC-H datasets
- → W2: 1000 jobs based on IMDB <- 50 Abstract plans -> 20 instantiations each

Characteristics of the Generated Data



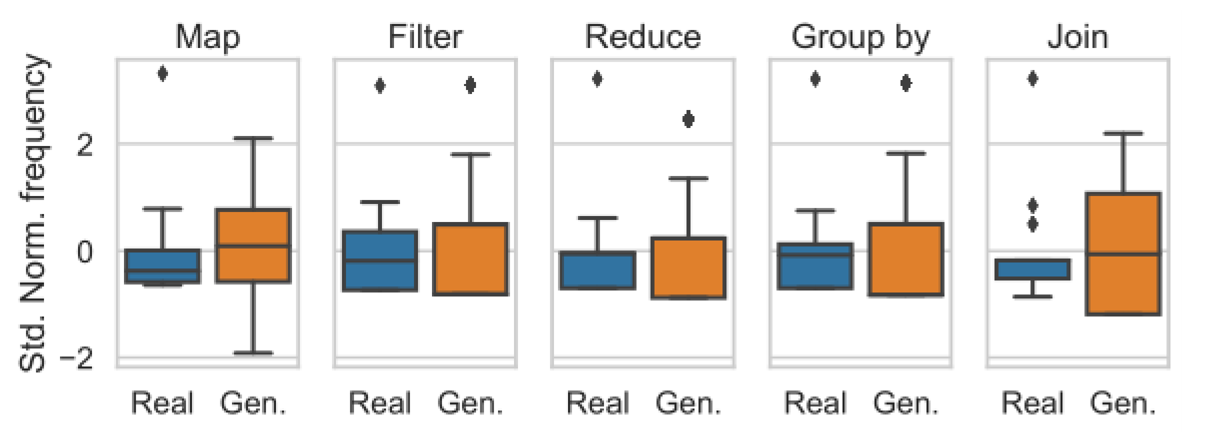
(a) W1



(b) W2

EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

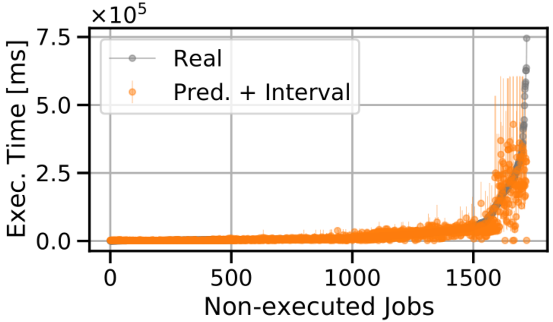
Operator Frequency Comparison



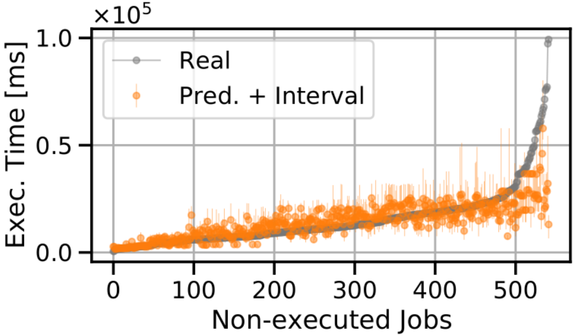
EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Prediction Accuracy

Generated Workload	# Gen. Jobs	# Exec. Jobs	Non-Exec. Jobs [%]	R^2 (Our)	R^2 (TDGen)
W1	2,000	142	93	**0.67**	<0
		532	73	**0.75**	0.65
W2	1,000	141	86	**0.16**	0.07
		418	58	**0.52**	0.08



(a) W1 – 142 executed jobs.

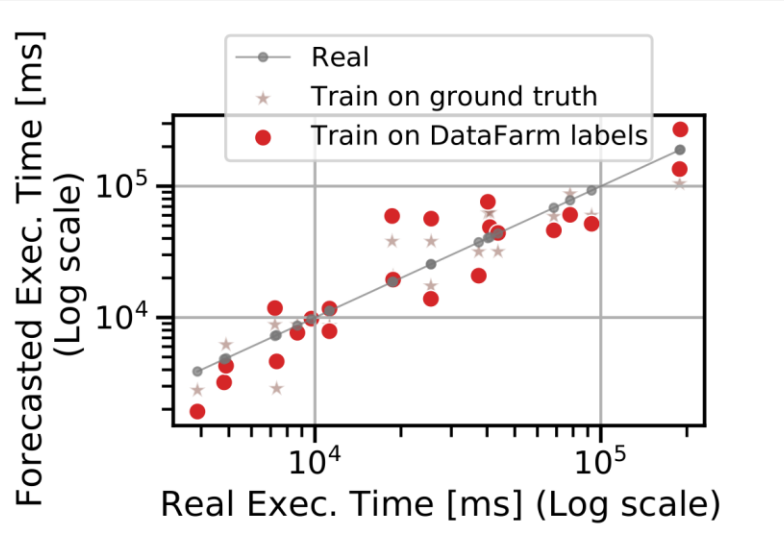


(b) W2 – 418 executed jobs

EXPAND YOUR TRAINING LIMITS!
GENERATING TRAINING DATA
FOR ML-BASED DATA
MANAGEMENT
SIGMOD '21

Effectiveness

→ Comparison of Random Forest Regressors
→ Trained on Predicted Labels vs Ground Truth
→ Ground Truth model only 1% more accurate --- 74% vs 73% variance explained

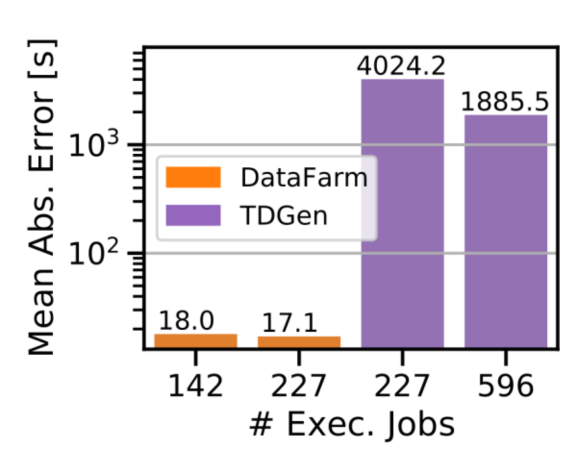


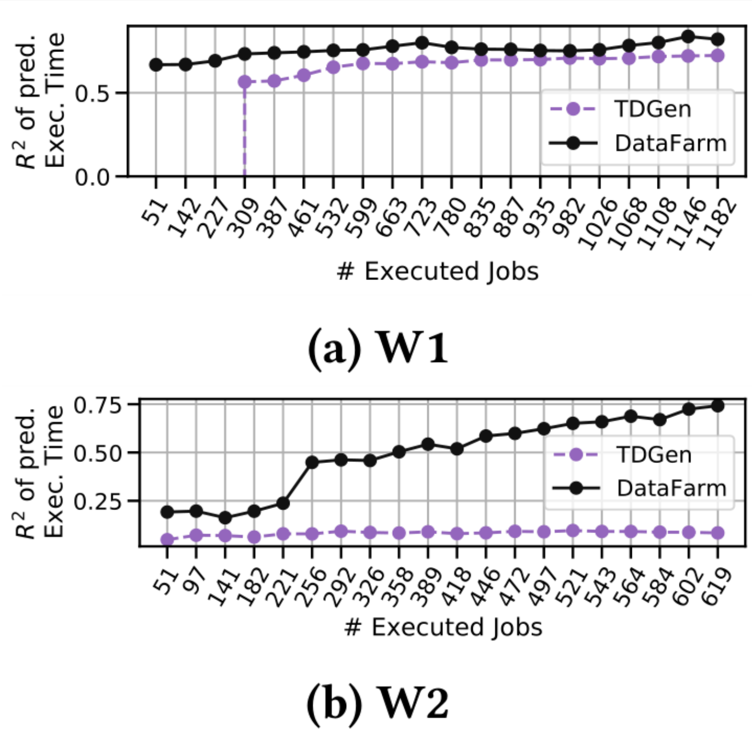
EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

DataFarm vs TDGen

→ Execution Time estimation model trained on jobs generated by DataFarm and TDGen

→ TDGen – doesn't consider input data distribution





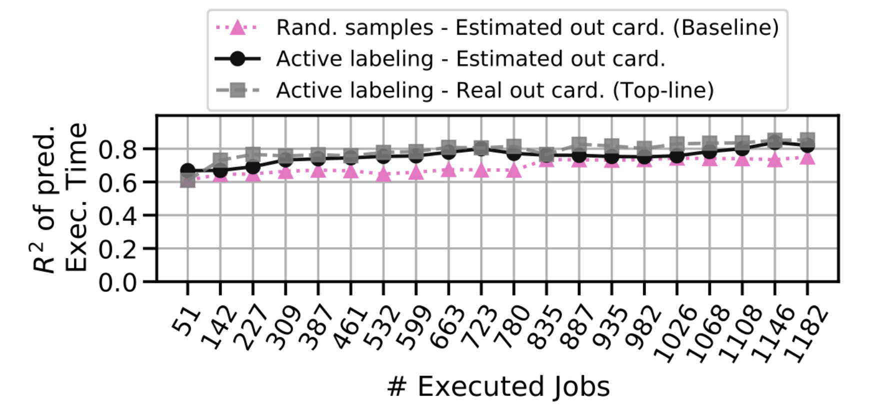
(a) W1

(b) W2

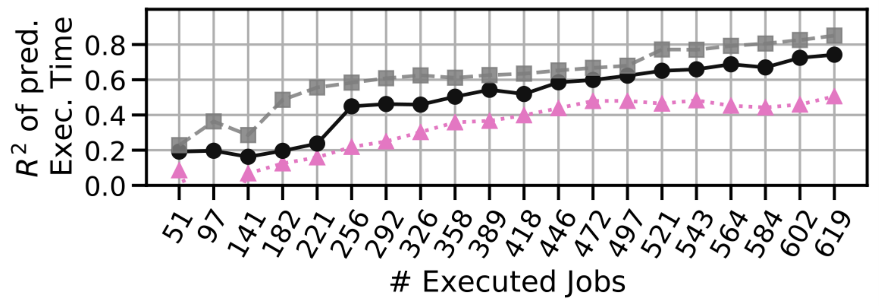
EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Active Labeling and Cardinality Estimates Effectiveness

→ Real Cardinality numbers extracted from Flink
→ Chosen method is significantly better only for W2



(b) W1 - R^2 validation scores.



(d) W2 - R^2 validation scores.

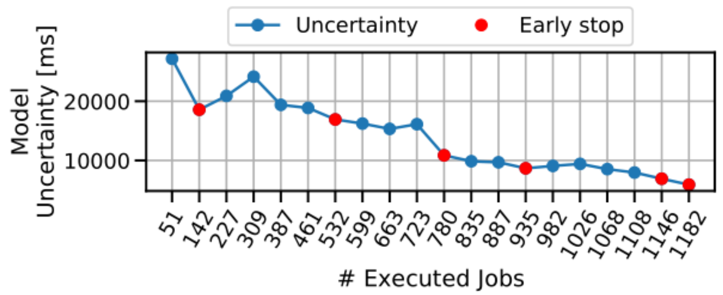
EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Model Uncertainty vs Training Data Size

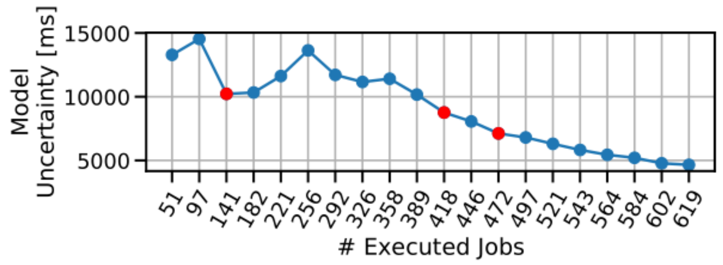
→ Early stop --- when there is a significant drop in uncertainty

$$\bar{u} = \frac{\sum_{\hat{l}}^{\hat{L}_{noex}} u_{high}(\hat{l}) - u_{low}(\hat{l})}{|\hat{L}_{noex}|}$$

→ u_{high} – 75% percentile prediction
→ u_{low} – 25% percentile prediction
→ Only on the unexecuted jobs



(a) W1 - Model's uncertainty.

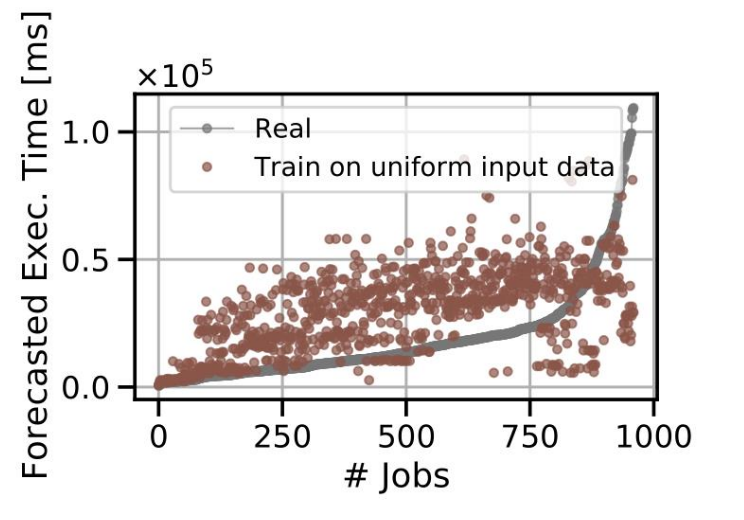


(c) W2 - Model's uncertainty.

EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

Evaluation

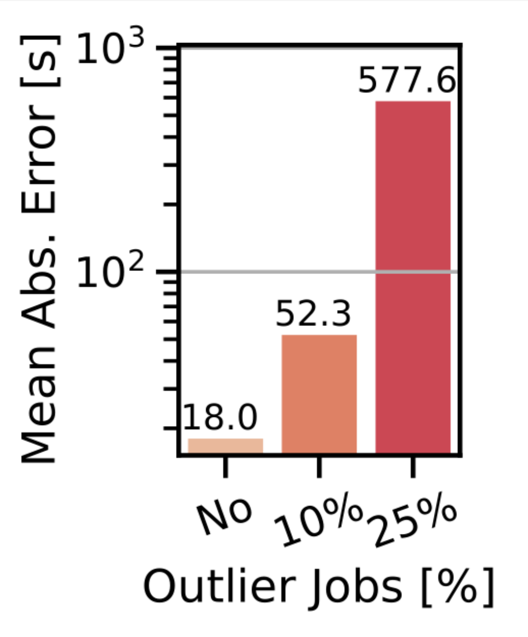
→ Data sampled uniformly during the generation phase
→ $R^2 = -0.40$



Input Data Importance – W2

Outliers

→ Ground Truth of X% of data points in W1 changed during the training process
→ The uncertainty value of the QRF model also becomes high



EXPAND YOUR TRAINING LIMITS! GENERATING TRAINING DATA FOR ML-BASED DATA MANAGEMENT
SIGMOD '21

PARTING THOUGHTS

→ This approach does not fully simulate a workload --- timing/frequency aspects not covered

→ Need better evaluation metrics

→ Only OLAP data right now?

References

1. F. Ventura, Z. Kaoudi, J. Quiané-Ruiz, and V. Markl. Expand your Training Limits! Generating and Labeling Jobs for ML-based Data Management. In SIGMOD, 2021.
2. Z.Kaoudi, J.Quiané-Ruiz, B.Contreras-Rojas, R.Pardo-Meza, A.Troudi, and S. Chawla. 2020. ML-based Cross-Platform Query Optimization. In 2020 IEEE 36th International Conference on Data Engineering (ICDE).