Demonstrating CatDB: LLM-based Generation of Data-centric ML Pipelines

June 22-27
2025

Saeed Fathollahzadeh [1], Essam Mansour [1], and Matthias Boehm [2]

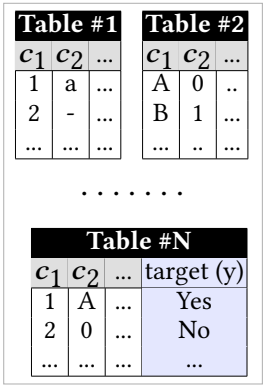
1 Université Concordia University

2 berlin

ACM SIGMOD/PODS
(demo track)

Data-centric ML Pipeline Generation via CatDB

Raw Data



Data-centric ML Pipeline Generation via CatDB

Raw Data

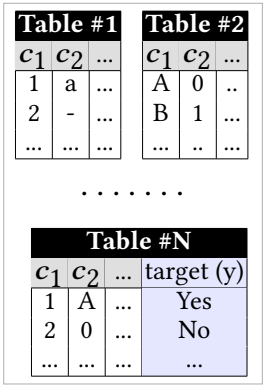


Table #1

c_1	c_2	...
1	a	...
2	-	...
...	...	...

Table #2

c_1	c_2	...
A	0	..
B	1	...
...	..	...

. . . . . .

Table #N

c_1	c_2	...	target (y)
1	A	...	Yes
2	0	...	No
...	...	...	...

ML Pipeline

```python
import pandas as pd
import SimpleImputer
import OneHotEncoder
import ColumnTransformer
import Pipeline
import RandomForestRegressor
import r2_score

trai = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")

ca = ["Experience", "Gender"]
cat = Pipeline(steps=[
      ("imputer", SimpleImputer(....
      ("onehot", OneHotEncoder(...
preprocessor = ColumnTransformer(
      transformers = [("cat",...)]
model = RandomForestRegressor(...)
p = Pipeline(steps=[ ....])

p.fit(X_train, y_train)
y_test_pred = p.predict(X_test)
```

Data-centric ML Pipeline Generation via CatDB

Raw Data

Table #1

c_1	c_2	...
1	a	...
2	-	...
...	...	...

Table #2

c_1	c_2	...
A	0	..
B	1	...
...	..	...

. . . . . .

Table #N

c_1	c_2	...	target (y)
1	A	...	Yes
2	0	...	No
...	...	...	...

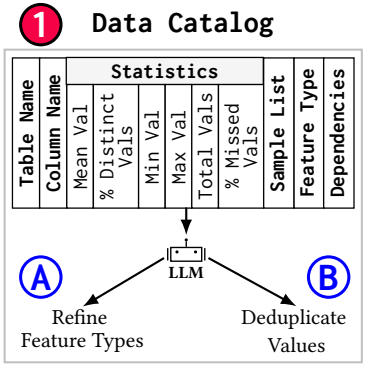
Pipeline Generation via CatDB

```
from catdb import config, generate_pipeline
from dataprofiling import build_catalog

cfg = config(llm_model='gpt-4o')
cat = build_catalog(path='raw_dataset')
p = generate_pipeline(catalog=cat, config=cfg)
```

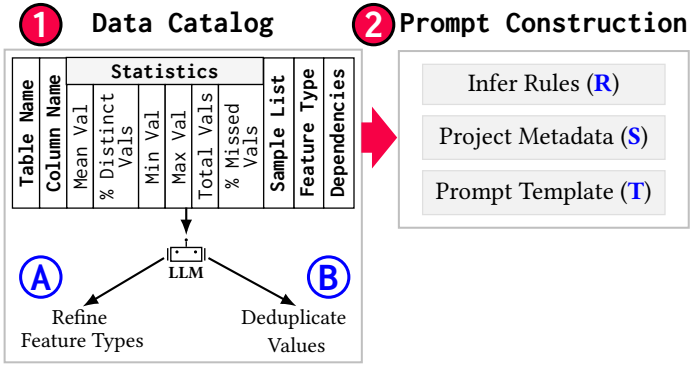
ML Pipeline

```
1:  import pandas as pd
2:  import SimpleImputer
3:  import OneHotEncoder
4:  import ColumnTransformer
5:  import Pipeline
6:  import RandomForestRegressor
7:  import r2_score

8:  trai = pd.read_csv("train.csv")
9:  test = pd.read_csv("test.csv")

10: ca = ["Experience", "Gender"]
11: cat = Pipeline(steps=[
        ("imputer", SimpleImputer(....
        ("onehot", OneHotEncoder(...
12: preprocessor = ColumnTransformer(
        transformers = [("cat",...)]
13: model = RandomForestRegressor(...)
14: p = Pipeline(steps=[ ....])

15: p.fit(X_train, y_train)
16: y_test_pred = p.predict(X_test)
```

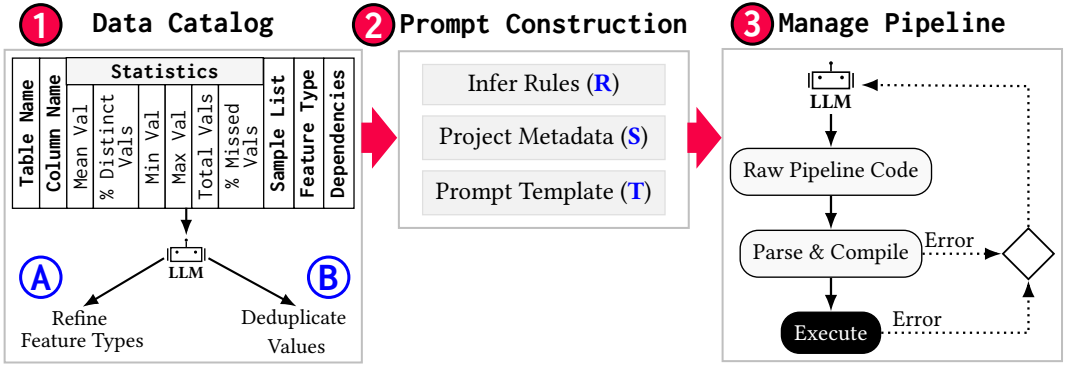
ML Pipeline Generation Workflow in CatDB



ML Pipeline Generation Workflow in CatDB



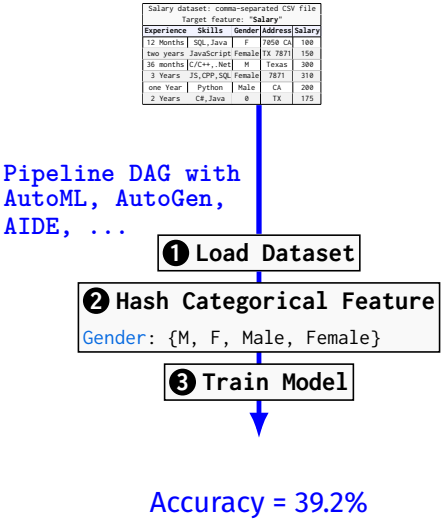
ML Pipeline Generation Workflow in CatDB



Comparison of Pipeline DAG in SOTA vs. CatDB

Salary dataset: comma-separated CSV file Target feature: "**Salary**"				
Experience	**Skills**	**Gender**	**Address**	**Salary**
12 Months	SQL,Java	F	7050 CA	100
two years	JavaScript	Female	TX 7871	150
36 months	C/C++,.Net	M	Texas	300
3 Years	JS,CPP,SQL	Female	7871	310
one Year	Python	Male	CA	200
2 Years	C#,Java	0	TX	175

Comparison of Pipeline DAG in SOTA vs. CatDB



Comparison of Pipeline DAG in SOTA vs. CatDB