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

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

[1] Université Concordia University
[2] Technische Universität Berlin

1. Motivation

- Data-centric ML pipelines are crucial ➜ labor-intensive.
- AutoML systems ➜ struggle with large datasets.
- LLMs demonstrate strong capabilities in coding ➜ struggle on unseen data.
- LLM-based pipeline generation ➜ lacks tailored dataset context.

➠ Need for efficient, scalable pipeline generation for diverse data.

Raw Dataset

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

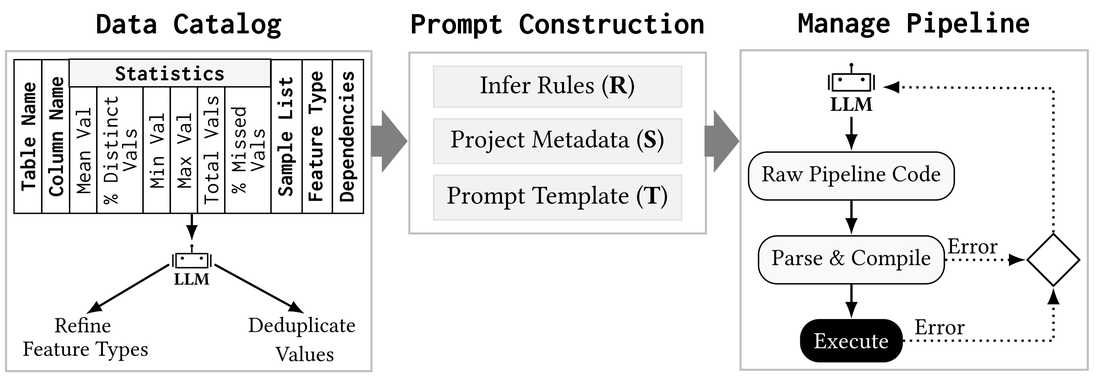
```
import pandas as pd
import SimpleImputer
import OneHotEncoder
import ColumnTransformer
import Pipeline
import RandomForestRegressor
import r2_score

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)
```

2. CatDB Overview



3. Data-centric ML pipelines w/ [Metadata-only & LLM] vs. [CatDB & LLM]

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

Pipeline DAG with Metadata & LLM

❶ Load Dataset
❷ Hash Categorical Feature
Gender: {M, F, Male, Female}
❸ Train Model

Accuracy = 39.2%

Pipeline DAG with CatDB & LLM

❶ Load Dataset
❷ Deduplicate Categorical Values
Experience: Replace(12 Months:1 year, ..)
Skills: Replace(CPP:C++, .Net:C#, ..)
Gender: Replace(F:Female, M:Male)
❸ Decompose Features
Address: Split to State and ZipCode
❹ Infer Feature Type
Skills: Set as List Type
❺ Feature Selection
Address: Keep ZipCode & Remove State
❻ Feature Hashing
❼ Train Model

Accuracy = 91.8%

4. An Example of a Prompt (Instructions and Meta- data)

System Rules (Instructions)

Task: Generate a data science pipeline in Python 3.10.

Input: A dataset in CSV format, a schema that describes the columns and data types of the dataset, and a data profiling info that summarizes the statistics and quality of the dataset.

Output: A pipeline code that performs the following steps:

#1: Import the necessary libraries and modules.

#2: Load the training and test datasets. For the training data, utilize the variable "train_data.csv", and for the test data, employ the variable "test_data.csv".

#3: The user will provide the Schema, and Data Profiling Info of the dataset with columns appropriately named as attributes.

#4: Perform missing value imputation for features 'Address' and 'Zip'.

#5: Perform feature extraction (dataset contains categorical values).

#6: Perform feature selection (redundant columns e.g., 'Address' and 'Zip').

#7: Select an appropriate ML algorithm.

#8: Assign a default value to a particular hyperparameter.

#9: Evaluate the model.

Catalog Data

Dataset Description: A The dataset was obtained from multiple sources, including surveys, job posting sites, and other publicly available sources.A total of 6704 data points were collected.The dataset included fivevariables: age, experience, job role, and education level and salary.

Schema, and Data Profiling Info:

```
# Experience (string), distinct [ 60% ], missing [ 0% ], categorical-vals [1 year, 2 years, 3 years]
# Gender (string), distinct [ 40% ], missing [ 0% ], categorical-vals [ Male, Female ]
# Address (string), distinct [ 40% ], missing [ 20% ], categorical-vals [ CA, TX ]
# Zip (string), distinct [ 40% ], missing [ 40% ], categorical-vals [ 7050, 7871 ]
# Salary (int, target feature), min-max vals [ 100, 310 ], total-vals [ 5 ]
```

Categorical Features: Experience, Gender, Address, Zip

5. CatDB Data-Centric ML Pipeline Generation APIs

❶
```
# Install Data Profiling
!pip install -U git+https://github.com/CoDS-GCS/CatDB.git@dataprofiling --quiet

# Install CatDB
!pip install -U git+https://github.com/CoDS-GCS/CatDB.git --quiet
```

```
  Preparing metadata (setup.py) ... done
  Preparing metadata (setup.py) ... done
```

❷
```
# Import CatDB APIs
from catdb import config, prepare_dataset, create_report, generate_pipeline

# Import Data Profiling API
from dataprofiling import build_catalog
```

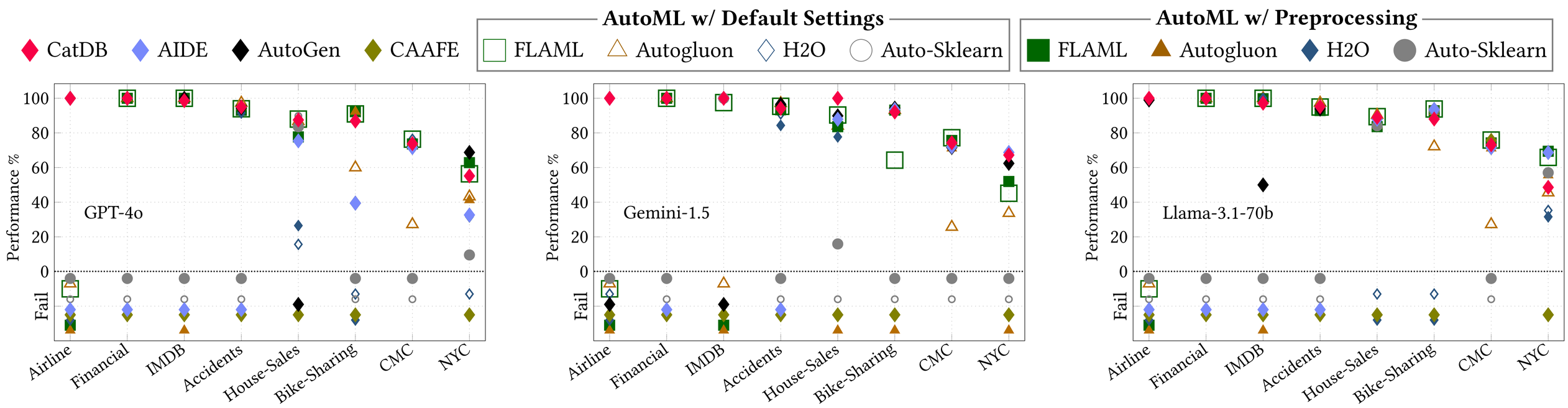
❸
```
cfg = config(model="gpt-4o", API_key="KEY", iteration=7)
```

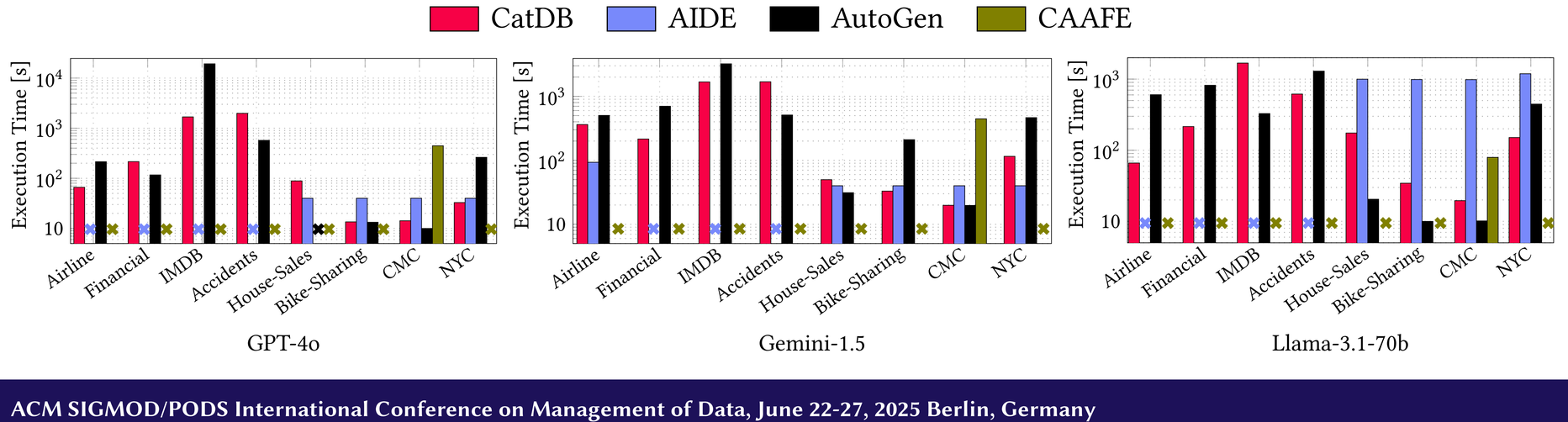
❹
```
data = prepare_dataset(path="adult.csv", task_type="binary",
                       target_attribute="income")
```

❺
```
catalog = build_catalog(data=data)
create_report(catalog)
```

❻
```
pipeline = generate_pipeline(catalog, cfg)
create_report(pipeline)
```

6. Performance Comparison of 8 Datasets





- CatDB's runtime includes: Data loading, Metadata projection, Rule definition, Pipeline generation, Validation, Error management, and Execution.
- CatDB performs efficiently across all datasets.
- Executed successfully on all datasets.
- Achieved performance gains by: Guiding LLMs to generate only necessary primitives and Parallelizing code execution.

saeed.fathollahzadeh@concordia.ca