AutoDC extracts a dataflow plan from the original Python code via instrumentation

```
def combine(patients, ...):
    ...

def create_neural_net():
    ...

def execute_pipeline():
    ...
```

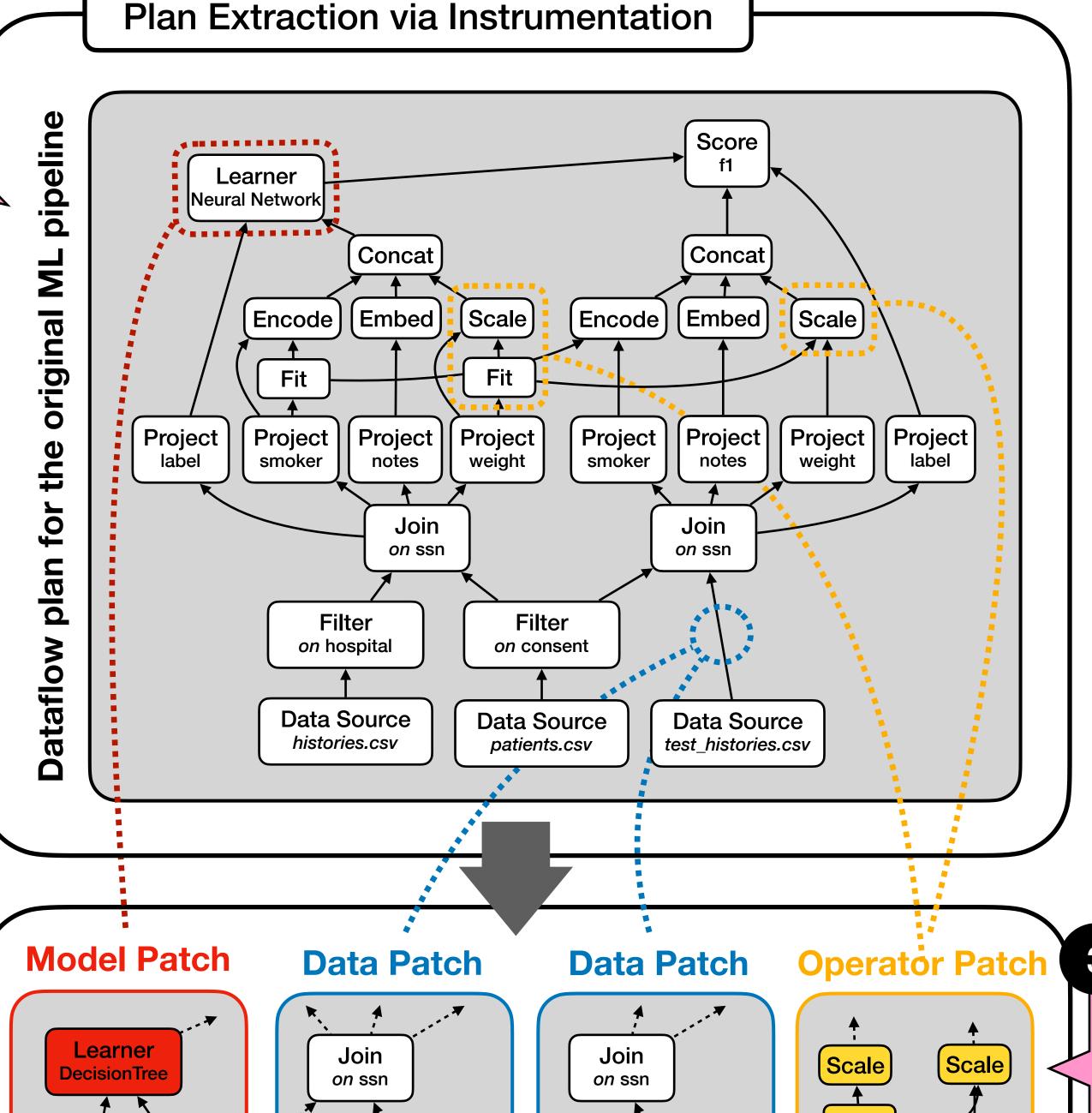
Original ML pipeline code

The data scientist provides the original pipeline code and specifies techniques to apply



```
analyses = [
  Cleanlearn(...),
  OperatorImpact(...),
]
```

Declaratively specified data-centric analysis techniques



Concat

Variant 3

Project label

Filter

on weight

**Data Source** 

test\_histories.csv

Variant 1

Variant Generation with Pipeline Patches

AutoDC generates plan variants for the analyses, based on pipeline patches

Fit

Project weight

Variant 4

Project weight

Project on weight

**Data Source** 

test\_histories.csv

Variant 2

AutoDC generates
a report per analysis
for the data scientist

## Report for operator impact analysis

	operator change	f1_score
>	DecisionTree as model	80.3%
	robust_scaler on weight	66.5%

## Report for cleanlearn analysis

column	error	cleaning	f1_score
weight	outliers	filter	82.1%
weight	outliers	impute	69.7%

Concat

Project

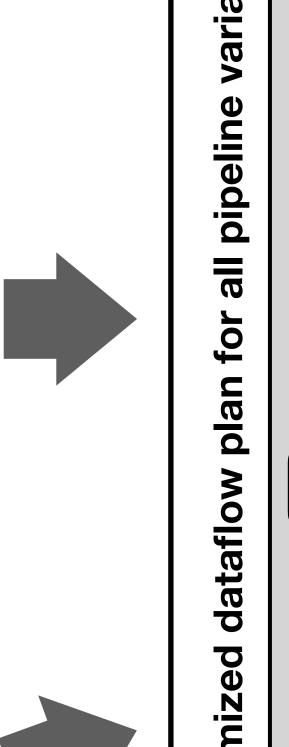
notes

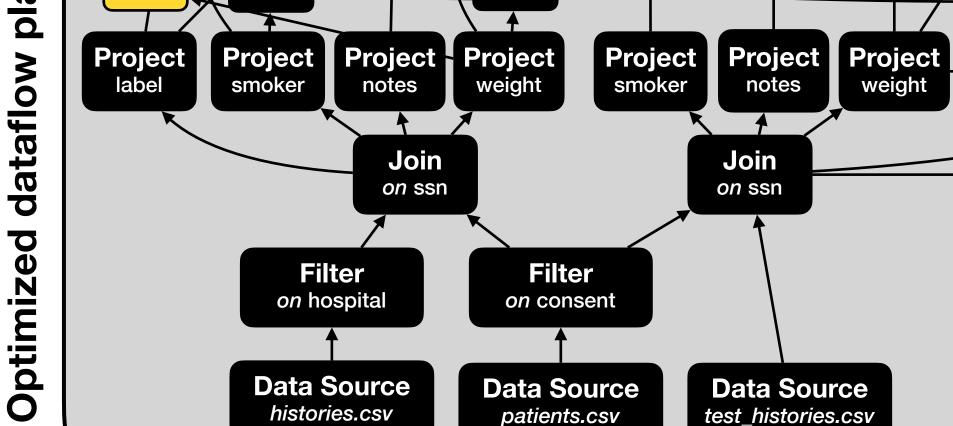
Project

Project

[Encode] [Embed]

Project smoker





Learner

**Multi-Query Optimization** 

Score of

Variant 4

Score

Encode Embed

Learner

Neural Network

Score of

Variant 3

Score

Scale

Score of

Variant 2

Score

Learner

Neural Network

Encode Embed

Data Source patients.csv

Data Source test\_histories.csv

AutoDC merges the plan variants into a single plan for all variants, applies multi-query

Score of

Variant 1

Score

Scale

Project label

Concat

Scale

optimizations and executes it