Karlsruhe Institute of Technology

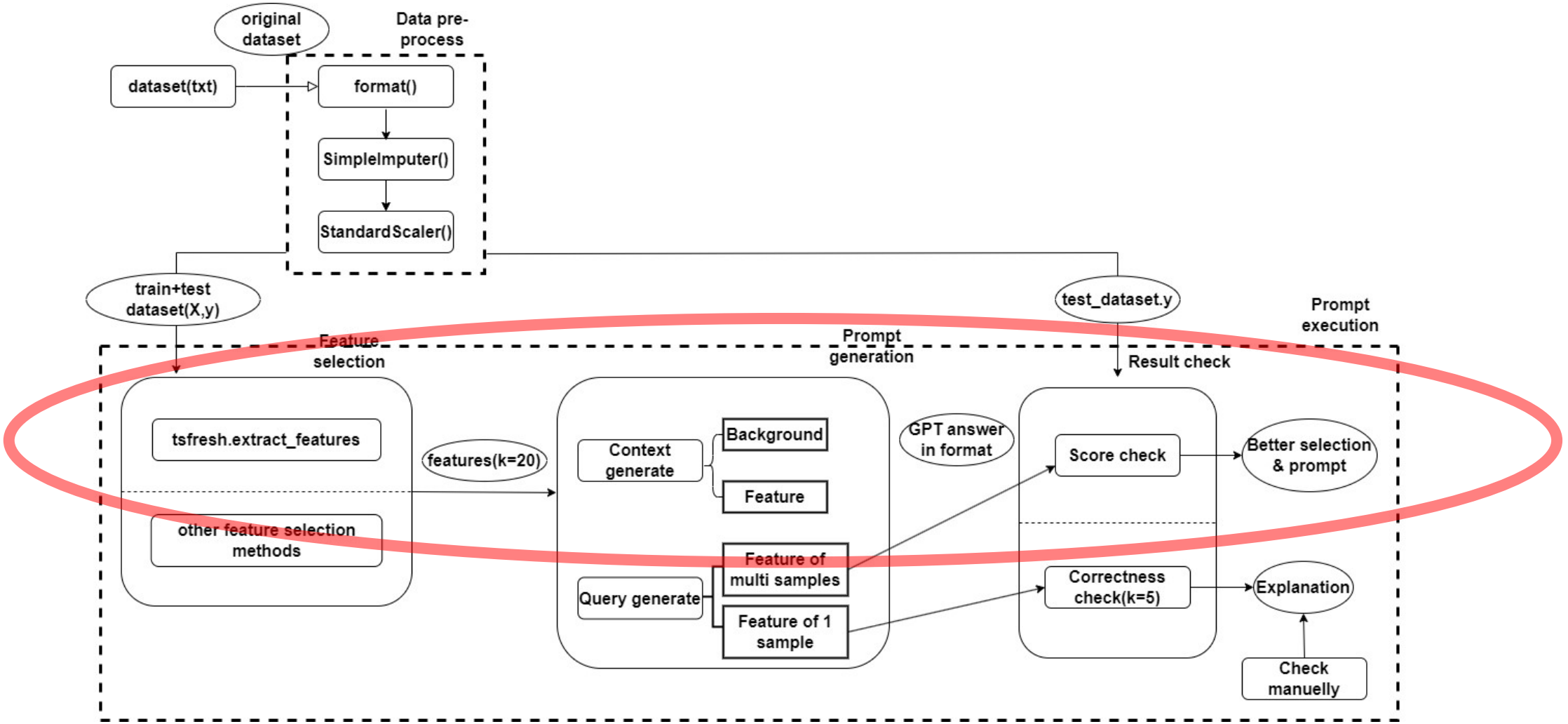
TECHNOLOGY FOR PERVASIVE COMPUTING

Explainable model for time-series via ChatGPT

Xue Zhechang

CHAIR FOR PERVASIVE COMPUTING SYSTEMS, INSTITUTE OF TELEMATICS, DEPARTMENT OF COMPUTER SCIENCE

pipeline


original dataset
Data pre-process
dataset(txt)
format()
SimpleImputer()
StandardScaler()
train+test dataset(X,y)
test_dataset.y
Prompt execution
Feature selection
Prompt generation
Result check
tsfresh.extract_features
other feature selection methods
features(k=20)
Context generate
Background
Feature
Query generate
Feature of multi samples
Feature of 1 sample
GPT answer in format
Score check
Better selection & prompt
Correctness check(k=5)
Explanation
Check manuelly

Outlines

- Feature selection
- Prompt generation
- Result check

Feature selection

2 week ago

- Method: Use 3 different but similar datasets to get common important features

- Problem:
  - 3 datasets not enough -> not common
  - Hard to count

- Proof:
  - Use Decision Tree, Gradient Boosting Trees & Random Forest
  - Select top 20 features after model fit
  - Way different (20%)

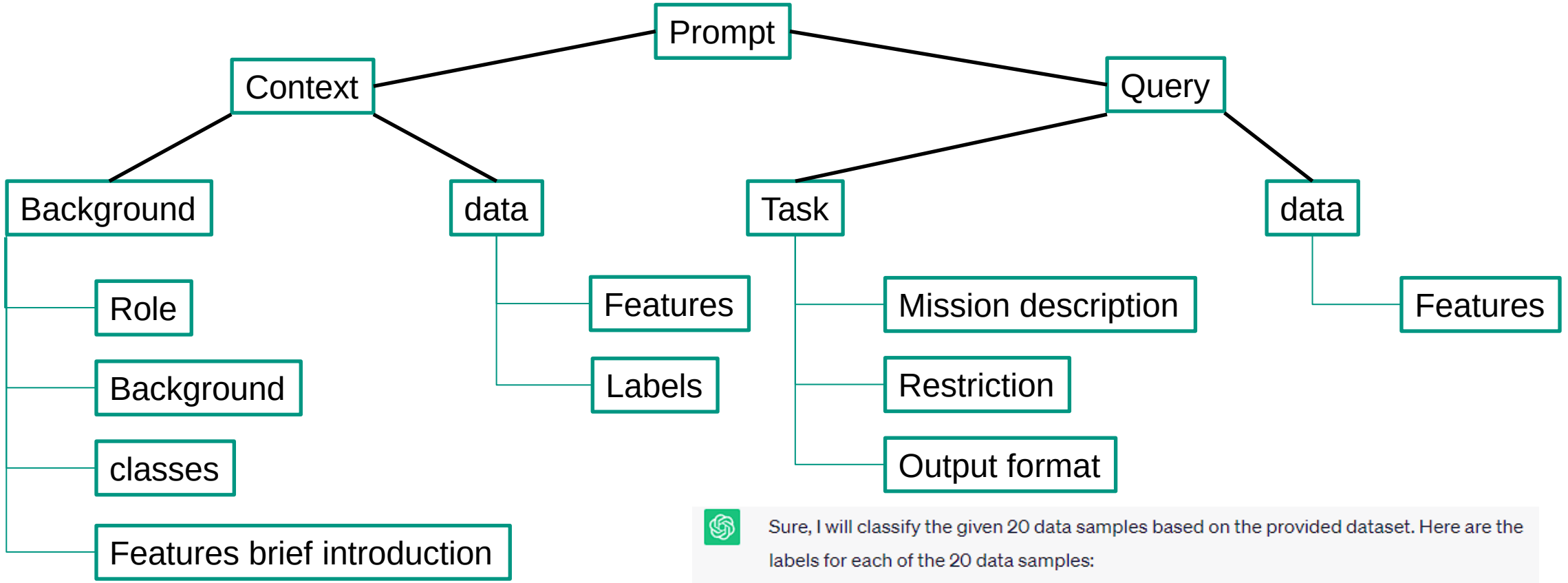
Feature selection

Improvement

- Select features by models
  - Decision Tree, Gradient Boosting Trees & Random Forest
- Experiment: Select top 20 features
  - Have feature importance as proof
  - Different by each

Prompt generation

Structure



Sure, I will classify the given 20 data samples based on the provided dataset. Here are the labels for each of the 20 data samples:

[3, 4, 1, 1, 1, 2, 2, 3, 1, 1, 1, 3, 2, 2, 3, 1, 1, 3, 1, 1]

There are exactly 20 labels in the answer.

Prompt generation

Token limit

- Dataset
  - 300 samples(10 classes * 30 samples each class)
  - 20 features
  - Total: 6000 numbers, but limit 4096 tokens!

- Data volume = features * labels * samples of each label
  - Features ?
  - Labels ↓
  - Samples of each label ↑

→ **Performance ↑**

Prompt generation

Start from 10 * 10 * 30

Karlsruhe Institute of Technology

Decline significant figures

GPT-3 Codex

```
'0.2089817' '0.008413343' '0.1047324' '45'
 '0.432897' '0.4555556' '3.879']['2' '1.315007' '8' '119.8487'
'0.2839363' '0.008106622' '0.1425059' '58'
 '0.4281772' '0.4634146' '2.862714']['2' '1.840393' '4' '-172.5804'
'0.1692497' '-0.01921115' '0.08476608'
 '64' '0.8177784' '0.4033613' '3.022143']['2' '0.6511865' '6'
'136.8141' '0.215617' '0.000254107' '0.110453' '189'
 '0.2871086' '0.3309353' '2.428714']['3' '1.407508' '2' '107.0058'
```

Clear Show example

Tokens	Characters
20,524	34212

GPT-3 Codex

```
'0.3046' '23' '-0.139'
 '0.4688' '0.489']['10' '0.1811' '0' '-129' '0.6874' '-0.0097' '0.1985'
'8' '0.2788'
 '0.3846' '0.3683']['10' '0.0743' '0' '-13.11' '0.3402' '-0.0002'
'0.2237' '127' '0.2516'
 '0.4478' '0.3243']['10' '0.1555' '0' '131.1' '0.4135' '0.0011'
'0.1914' '10' '0.2687'
 '0.4756' '0.5056']
```

Clear Show example

Tokens	Characters
17,562	26555

Prompt generation

Tips

Karlsruhe Institute of Technology

Better separation

GPT-3 Codex

```
['5' '0.3214' '2' '-50.29' '0.6615' '-0.0069' '0.2788' '4' '0.0319'
 '0.2449' '0.9889']
```

Clear Show example

Tokens	Characters
59	87

GPT-3 Codex

```
[5,0.3214,2,-50.29,0.6615,-0.0069,0.2788,4,0.0319,0.2449,0.9889]
```

Clear Show example

Tokens	Characters
46	64

GPT-3 Codex

```
[5 0.3214 2 -50.29 0.6615 -0.0069 0.2788 4 0.0319 0.2449 0.9889]
```

Clear Show example

Tokens	Characters
38	64

Prompt generation

End with 5 * 10 * 15

Decline data volume

GPT-3 Codex

```
[10 0.1252 0 -135.8 0.3202 -0.0033 0.1192 27 0.5408 0.4156 0.2473]
[10 0.157 0 -127.2 0.4033 -0.0048 0.1243 33 0.5331 0.5833 0.3297]
[10 0.1553 0 -52.35 0.4158 -0.0007 0.2392 39 0.4707 0.6393 0.5166]
[10 0.237 0 -16.06 0.6272 -0.001 0.3046 23 -0.139 0.4688 0.489]
[10 0.1811 0 -129 0.6874 -0.0097 0.1985 8 0.2788 0.3846 0.3683]
[10 0.0743 0 -13.11 0.3402 -0.0002 0.2237 127 0.2516 0.4478 0.3243]
[10 0.1555 0 131.1 0.4135 0.0011 0.1914 10 0.2687 0.4756 0.5056]
```

Clear Show example

Tokens	Characters
11,263	19955

GPT-3 Codex

```
[5 0.3628 0 -22.73 0.3735 0.0009 0.1732 53 0.3671 0.6489 0.8791]
[5 0.0463 0 -120.6 0.4524 -0.0003 0.2146 49 0.1901 0.4302 0.1317]
[5 0.035 0 154.8 0.5807 -0 0.1513 20 -0.0179 0.5497 0.0824]
[5 0.1559 0 37.19 0.5137 0.0001 0.3009 21 0.0769 0.3333 0.6154]
[5 0.0809 0 5.747 0.5282 0.0011 0.2762 4 0.3929 0.48 0.231]
[5 0.2772 0 143.2 0.5893 0.0097 0.3576 20 0.2531 0.32 0.5657]
[5 0.262 0 -86.96 0.3245 -0.0152 0.163 1 0.8631 0.2353 0.7031]
```

Clear Show example

Tokens	Characters
2,944	5508

Result check (50 times)

	Decision Tree	Gradient Boosting	Random Forest**
Average	51.9%	54%	40.2%
Highest	90% (5)	100% (1)	60%(2)
Lowest	20% (2)	25% (1)	20%(1)