دانشکده مهندسی کامپیوتر

اصول طراحی کامپایلر

پروژه data engineer

استاد

دکتر سعید پارسا

طراحان

محسن رحیمی

رضا بوذرجمهری

آیدین اصل زعیم

رضا حقیقت گو

بهار ۱۴۰۳

مقدمه

DSL یا زبان خاص دامنه (Domain-Specific Language) نوعی زبان برنامه‌نویسی است که برای یک حوزه خاص یا یک وظیفه ویژه طراحی شده است. برخلاف زبان‌های عمومی برنامه‌نویسی مثل ،Python Java یا C++ که برای حل مسائل مختلف و عمومی استفاده می‌شوند، DSLها به منظور ساده‌سازی و بهینه‌سازی کدنویسی در یک دامنه خاص استفاده می‌شوند.

هدف این پروژه توسعه یک زبان دامنه خاص (DSL) برای عملیات پاکسازی داده‌ها است. DSL پاکسازی داده‌ها به کاربران اجازه می‌دهد تا وظایف پاکسازی داده‌ها را به صورت مختصر و قابل خواندن مشخص کنند که سپس به کد قابل اجرای پایتون ترجمه می‌شود. این پروژه شامل تعریف گرامر برای DSL، ایجاد یک درخت نحو انتزاعی (AST) و تولید کد پایتون معادل برای پاکسازی داده‌ها است.

اجزای پروژه

این پروژه را می‌توان در بخش های زیر دنبال کرد:

تعریف گرامر

گرامر برای DSL پاکسازی داده‌ها با استفاده از ANTLR (Another Tool for Language Recognition) تعریف شده است. این گرامر قوانین و ساختار DSL را مشخص می‌کند، از جمله انواع مختلف جملاتی که می‌توانند برای عملیات پاکسازی داده‌ها استفاده شوند.

قوانین کلیدی گرامر

- قانون شروع: نقطه ورود DSL که ساختار کلی یک برنامه را تعریف می‌کند.
- قانون برنامه: یک برنامه شامل یک load Statement و صفر یا بیشتر statement های پاکسازی داده‌ها است.
- جملات: عملیات‌های مختلفی مانند حذف ردیف‌های دارای مقدار گمشده، پر کردن مقادیر گمشده، نرمال‌سازی داده‌ها، استانداردسازی داده‌ها، اعمال تبدیل لگاریتمی، دسته‌بندی خودکار داده‌ها، تقسیم داده‌ها به مجموعه‌های آموزشی، اعتبارسنجی و آزمایش، حذف داده‌های تکراری، حذف ردیف‌ها یا ستون‌ها، یکپارچه‌سازی داده‌های ناسازگار، کدگذاری داده‌ها و مدیریت داده‌های خارج از محدوده.

درخت نحوی انتزاعی (AST)

AST یک نمایش درختی از ساختار نحوی انتزاعی کد منبع نوشته شده در DSL است. هر گره از درخت یک سازه را در کد منبع نشان می‌دهد.

توابع کلیدی

- ساخت AST: با استفاده از تابع make_ast_subtree ، AST از جملات DSL تجزیه‌شده ساخته می‌شود. این تابع هر جمله را پردازش کرده و زیردرخت متناظر را در AST ایجاد می‌کند.
- پیاده سازی listener: یک Listener سفارشی (CustomDataCleanerListener) از Listener پایه‌ای که توسط ANTLR تولید شده است گسترش می‌یابد. این Listener متدهایی را برای مدیریت خروج هر قاعده تعریف شده در گرامر نادیده می‌گیرد و AST را هنگام پردازش کد DSL ورودی می‌سازد.

تولید کد میانی

تولیدکننده کد (DataCleanerCodeGenerator) درخت AST را مرور کرده و کد پایتون برای عملیات پاکسازی داده‌ها تولید می‌کند. این تولیدکننده عملیات مختلف مشخص‌شده در DSL را مدیریت کرده و آن‌ها را به کد معادل پایتون با استفاده از کتابخانه‌هایی مانند pandas و numpy ترجمه می‌کند.

عملیات‌های کلیدی

• بارگذاری داده‌ها: جمله load را برای خواندن یک فایل CSV به یک DataFrame pandas ترجمه می‌کند.

- حذف ردیف‌های دارای مقدار گمشده: کدی را برای حذف ردیف‌های دارای مقادیر گمشده در ستون‌های مشخص‌شده تولید می‌کند.
- پر کردن مقادیر گمشده: کدی را برای پر کردن مقادیر گمشده در ستون‌های مشخص‌شده با استفاده از روش‌هایی مانند میانگین، میانه یا مد تولید می‌کند.
- نرمال‌سازی داده‌ها: کدی را برای نرمال‌سازی داده‌ها در ستون‌های مشخص‌شده به محدوده داده‌شده تولید می‌کند.
- استانداردسازی داده‌ها: کدی را برای استانداردسازی داده‌ها در ستون‌های مشخص‌شده تولید می‌کند (یعنی میانگین صفر و واریانس واحد).
- تبدیل لگاریتمی : کدی را برای اعمال تبدیل لگاریتمی به ستون‌های مشخص‌شده تولید می‌کند.

- دسته‌بندی خودکار: کدی را برای دسته‌بندی داده‌ها در ستون‌های مشخص‌شده با استفاده از تکنیک‌های خوشه‌بندی مانند K-means تولید می‌کند.
- تقسیم داده‌ها: کدی را برای تقسیم داده‌ها به مجموعه‌های آموزشی، اعتبارسنجی و آزمایش تولید می‌کند.
- حذف داده‌های تکراری: کدی را برای حذف ردیف‌های تکراری از داده‌ها تولید می‌کند.
- حذف ردیف‌ها و ستون‌ها: کدی را برای حذف ردیف‌ها یا ستون‌های مشخص‌شده تولید می‌کند.
- یکپارچه‌سازی داده‌های ناسازگار: کدی را برای جایگزینی مقادیر ناسازگار در ستون‌های مشخص‌شده تولید می‌کند.
- کدگذاری داده‌ها: کدی را برای کدگذاری متغیرهای دسته‌ای با استفاده از روش‌هایی مانند کدگذاری یک‌گرمی تولید می‌کند.
- مدیریت داده‌های خارج از محدوده : کدی را برای مدیریت داده‌های خارج از محدوده در ستون‌های مشخص‌شده با استفاده از روش‌هایی مانند دامنه بین چارکی (IQR) تولید می‌کند.

مرور و مصورسازی AST

کلاس PostOrderASTTraverser برای مرور AST به صورت پس‌نظم و آماده‌سازی مرور برای تولید کد استفاده می‌شود. علاوه بر این، NetworkX برای ترسیم و مصورسازی AST استفاده می‌شود و یک نمایش گرافیکی واضح از ساختار درخت را ارائه می‌دهد.

توابع کلیدی

- ساخت گراف: یک نمایش گرافی از AST با استفاده از NetworkX ساخته می‌شود.
- مرور: گره‌های AST را به صورت postorder مرور کرده و ویژگی‌های گره را جمع‌آوری می‌کند تا تولید کد تسهیل شود.
- مصورسازی :NetworkX برای ترسیم AST استفاده می‌شود و فهم ساختار و روابط بین گره‌ها را آسان‌تر می‌کند. این نمایش بصری در دیباگ و تایید صحت AST کمک می‌کند.

جمع بندی

پروژه DSL پاکسازی داده‌ها شامل تعریف یک زبان سفارشی برای مشخص کردن عملیات پاکسازی داده‌ها، ساخت AST از جملات DSL و تولید کد پایتون برای اجرای این عملیات است. اجزای کلیدی پروژه شامل تعریف گرامر، ساخت AST، تولید کد و مرور و مصورسازی AST می‌شود. این پروژه نشان‌دهنده کاربرد اصول کامپایلر برای توسعه یک ابزار عملی برای پیش‌پردازش داده‌ها است.

در نهایت یک نمونه فایل ورودی به برنامه دادیم و حال خروجی آن را مشاهده می‌کنیم

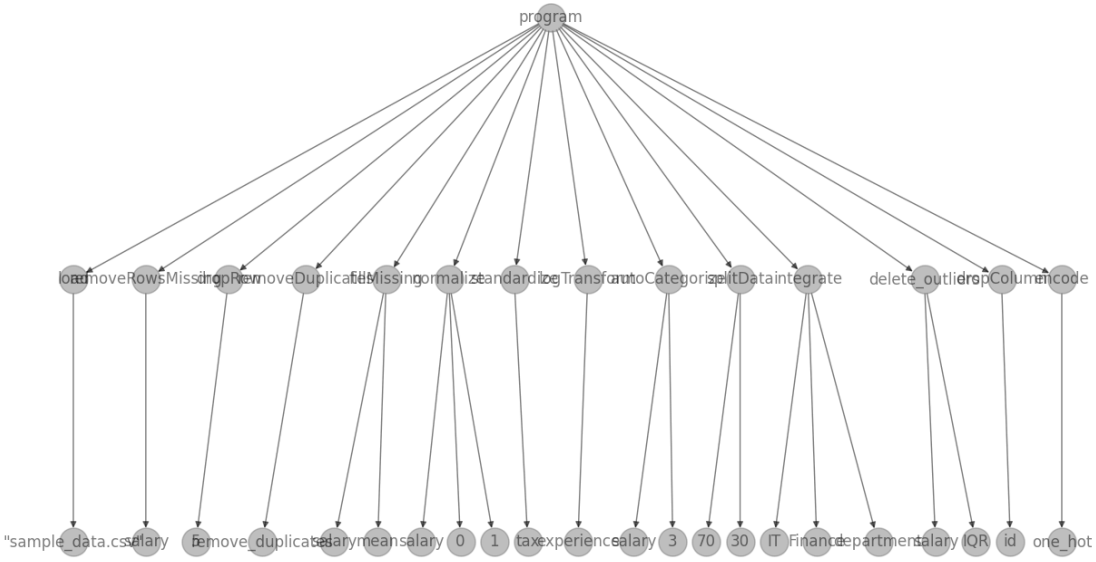
1. ابتدا یک فایل csv حاوی داده تست برای بررسی کد خود آماده می‌کنیم

	A	B	C	D	E	F	G
1	id	age	salary	tax	department	experience	
2	1	23	50000	100	HR	1	
3	2	45	52000	444	IT	3	
4	3	25		554	IT	4	
5	4	29	1000000	2344	HR	5	
6	5		55000	445	HR	7	
7	6	35	56000	453	Finance	9	
8	7	40	34900	876	Finance	10	
9	8		58000	900	IT	6	
10	9	50	59000	980	Finance	13	
11	10	55	60000	977	IT	15	
12	11	60	61000	545	HR	17	
13	12	65	5600	445	Finance	19	
14	13	70	76655	555	Finance	20	
15	14	75		550	HR	21	
16	15	80	65000	778	IT	23	
17	16	34	66000	899	HR	25	
18	17	85	67000	890	Finance	27	
19	18	90	77889	665	HR	30	
20	19	95	69000	990	IT	35	
21	20	100	70000	998	Finance	40	
22	6	35	56000	453	Finance	9	
23	11	60	61000	545	HR	17	
24	21	44	555500000	0	HR	34	
25	22	45	67	66	Finance	23	
26							

2. در یک فایل ورودی حاوی دستور های dsl خود را وارد می‌کنیم

```
// Load the sample data
load "sample_data.csv"

// Remove rows with missing values in 'salary' column
remove_rows_missing salary

// Drop specific rows (e.g., 2nd and 5th rows)
drop_row 5

// Remove duplicate rows
remove_duplicates

// Fill missing values in 'salary' column with mean
fill_missing salary with mean

// Normalize 'salary' column to range 0 to 1
normalize salary to_range(0, 1)

// Standardize the 'tax' column
standardize tax

// Logarithm transform the 'experience' column
log_transform experience

// Auto categorize the 'salary' column into 3 clusters
auto_categorize salary n_clusters=3

// Split data into 70% training, 30% testing sets
split_data train=70 , test=30

// Integrate inconsistent data in 'department' column
integrate IT to Finance in department

// Delete outliers in 'salary' column using IQR method
delete_outliers salary with IQR

// Drop specific columns (e.g., 'id' and 'department')
drop_column id

// Encode all columns with one-hot encoding
encode all with one_hot

```

3. حال فایل main.py را اجرا می‌کنیم و مشاهده می‌کنیم که درخت ast زیر تولید می‌شود



4. در نهایت حاصل این عملیات ایجاد یک فایل csv جدید به این شکل است

	A	B	C	D	E	F	G
1	age	salary	tax	experience	department_Finance	department_HR	
2	23	0	-1.23731647119456	0.693147180559945	False	True	
3	45	0	-0.548025999792248	1.38629436111989	True	False	
4		0	-0.546022248421892	2.07944154167984	False	True	
5	40	0	0.317594592201357	2.39789527279837	True	False	
6		0	0.365684625089891	1.94591014905531	True	False	
7	50	0	0.525984734718336	2.63905732961526	True	False	
8	55	0	0.51997348060727	2.77258872223978	True	False	
9	60	0	-0.345647111386335	2.89037175789616	False	True	
10	65	0	-0.546022248421892	2.99573227355399	True	False	
11	70	0	-0.32560959768278	3.04452243772342	True	False	
12	80	0	0.121226957906512	3.17805383034795	True	False	
13	34	0	0.363680873719535	3.25809653802148	False	True	
14	85	0	0.345647111386335	3.3322045101752	True	False	
15	90	0	-0.105196946943667	3.43398720448515	False	True	
16	95	0	0.546022248421892	3.58351893845611	True	False	
17	100	0	0.562052259384737	3.71357206670431	True	False	
18	35	0	-0.529992237459047	2.30258509299405	True	False	
19	45	0	-1.30544401778665	3.17805383034795	True	False	
20							