

Automated Exploratory Data Analysis of Databases

Diego Arenas
@darenasc

04 November 2021



Diego Arenas

- Computer scientist; MSc in Data Science, University of Edinburgh; and EngD © in Computer Science, University of St Andrews.
- 15+ years of experience working in data science and data engineering projects
- Co-host of the podcast “escuchAI” about AI (<http://escuchai.com>)
- Personal webpage <https://darenasc.github.io>

Agenda

- Motivation
- Data Exploration
- What is aeda
- How to use aeda
- Demo
- Use cases
- Q&A

Do you know the
content of *all* the
databases in your
organisation?

Data exploration






- Important initial phase in any data related project
- Analyse the content of a data source
- Making sense of the data
- Some data sources have hundreds of tables
- Data quality
- Data governance

Why automate the data exploration phase?

- Series of ***repetitive*** steps we follow to try to understand the data we are working with
- Time of manual data exploration is often linear 10 new tables will take 10x additional time to process
- Multiple data sources?
- Standardised way to explore data sources
- We want to see trends, distributions of data, data quality assessment, etc.
- Working with big tables can be optimised from GB to KB of data and waiting times

What is aeda?

- AEDA stands for Automated Exploratory Data Analysis
- *It uses two databases: a Source database and a Metadata database*
- Extracts metadata from the source and creates a data catalogue of metadata

Database	SOURCE	METADATA
 MySQL SOURCE / METADATA	✓	✓
 MariaDB SOURCE / METADATA	✓	✓
 Postgres SOURCE / METADATA	✓	✓
MS SQL Server SOURCE / METADATA	✓	✓
 SQLite3 METADATA		✓
 Snowflake METADATA		✓

What metadata?

- | | | |
|---|----|--|
| Basic information | 1. | Server name, table catalog, table schema, table names, column names, ordinal position of the columns, and column data types . |
| Project size | 2. | number of tables in the database. |
| | 3. | number of rows per table. |
| | 4. | number of columns per table. |
| | 5. | number of unique values per column. |
| Discrete data type \Rightarrow | 6. | number of null values per column. |
| Continuous data type \Rightarrow | 7. | frequency number per data value per column. |
| Datetime data type \Rightarrow | 8. | timewise aggregation of the data. |
| Continuous data type \Rightarrow | 9. | univariate summary statistics for all the numeric data types: mean, standard deviation, variance, maximum, minimum, percentiles (0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.975, 0.99), interquartile range, range, kurtosis, and skewness. |

How to use aeda?

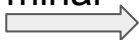
[my-source-database]

```
db_engine = <A-SUPPORTED-DB-ENGINE>  
host = <IP-OR-HOSTNAME-SOURCE-DATABASE>  
schema = <SCHEMA-SOURCE-DATABASE>  
catalog = <CATALOG-SOURCE-DATABASE>  
user = <SOURCE-USER>  
password = <SOURCE-PASSWORD>  
port = <SOURCE-PORT>
```

[my-metadata-database]

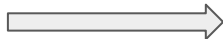
```
db_engine = <A-SUPPORTED-DB-ENGINE>  
host = <IP-OR-HOSTNAME-METADATA-DATABASE>  
schema = <SCHEMA-METADATA-DATABASE>  
catalog = <CATALOG-METADATA-DATABASE>  
user = <METADATA-USER>  
password = <METADATA-PASSWORD>  
port = <METADATA-PORT>  
metadata_database = yes
```

From the terminal



```
python aeda_.py explore my-source-database my-metadata-database
```

Data exploration
GUI



```
streamlit run aeda_app.py
```

Demo

- Creation of the connections to databases
- Running an exploration
- Examine the data

SERVER_NAME

☐ (Blank)☐ ...☐ ...☐ ...

TABLE_NAME

☐ (Blank)☐ ...

TABLE_NAME

COLUMN_NAME

DISTINCT_VALUES

NULL_VALUES

DATA_TYPE

TABLE_NAME	COLUMN_NAME	DISTINCT_VALUES	NULL_VALUES	DATA_TYPE
...	0.00	int
...	0.00	bigint
...	0.00	bigint
...	0.00	bigint
...	0.00	int
...	0.00	int
...	0.00	int
...	0.00	bigint
...	0.00	bigint
...	0.00	datetime
...	0.00	datetime
...	0.00	int
...	0.00	bigint
...	0.00	bigint
...	0.00	bigint
...	0.00	bigint
...	0.00	datetime
...	0.00	datetime
...	0.00	datetime
...	0.00	datetime
...	0.00	datetime
...	0.00	int
...	0.00	bigint
...	0.00	datetime
...	0.00	datetime
...	0.00	int
...	0.00	int
...	0.00	datetime
...	0.00	int
...	0.00	float
...	1.00	bigint
...	0.00	int
...	0.00	float
...	0.00	bigint
...	0.00	bigint
...	0.00	int

5bn

N_ROWS

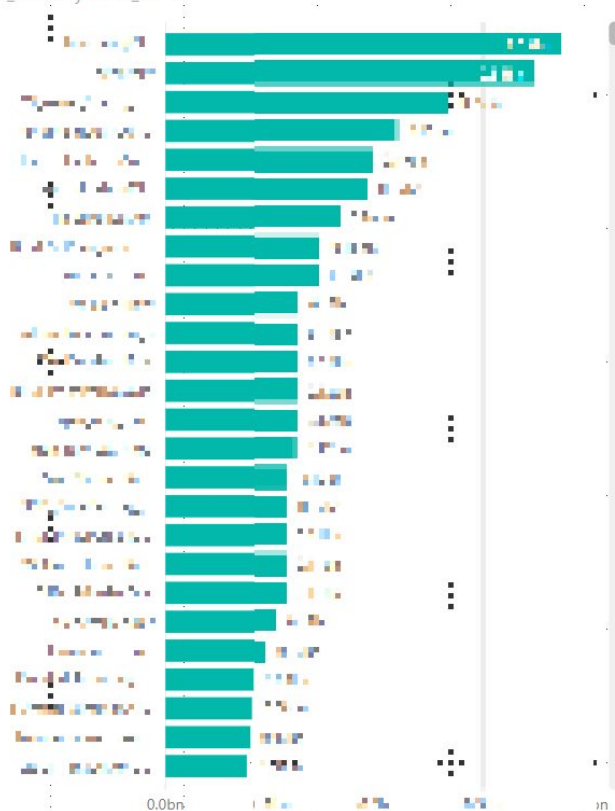
14.80K

N_COLUMNS

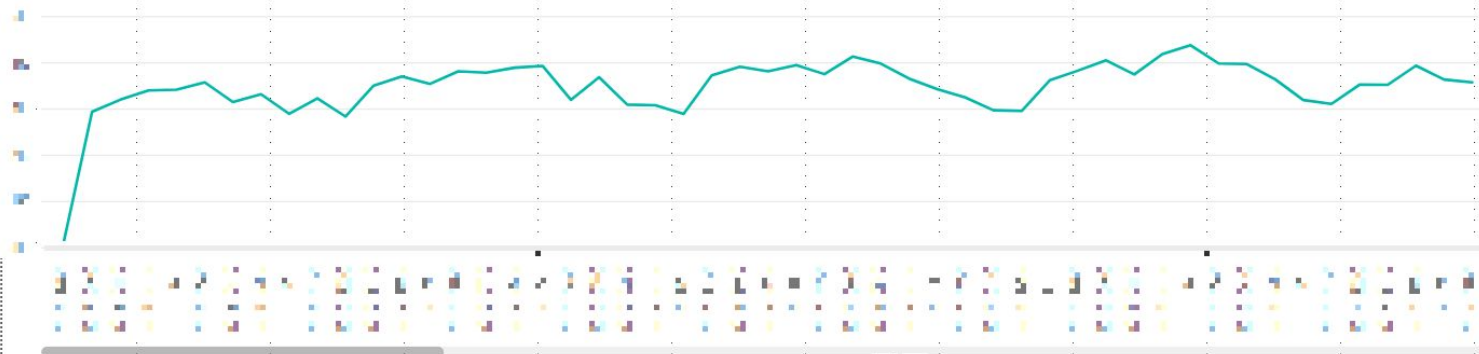
823

N_TABLES

N_ROWS by TABLE_NAME



.....
: FREQUENCY_NUMBER by DATA_VALUE
:

[illegible][illegible]

Use cases

1. New project and requires inspection of the new data sets
2. Creation of dev database from a live database
3. Data quality inspection
4. New CDC or ETL processes from new systems
5. Machine learning project
6. Find data visible in applications
7. “Searching for a needle in a haystack”, tracing data values in data sources.
8. Link data by value domain
9. Rapid growing database of IoT devices?
10. Multiple data sources to explore?
11. ...

Excellent libraries in Python

- Pandas Profiling <https://github.com/pandas-profiling/pandas-profiling>
- Soda SQL <https://github.com/sodadata/soda-sql>
- Sweeviz <https://github.com/fbdesignpro/sweetviz>
- Datapane <https://github.com/datapane/datapane>
- Great expectations
https://github.com/great-expectations/great_expectations
- Superset <https://github.com/apache/superset>
- Metabase <https://github.com/metabase/metabase>

Questions?

Get in touch

<https://calendly.com/darenasc/>

Link to the library:

<https://github.com/darenasc/aeda>



Metadata schema

There are 6 tables in the metadata schema.

Table name	Description
columns	Each row is a column of a table
tables	Each row is a table of a database
uniques	Similar to the columns table adding the number of rows and null values
data_values	Each row is a unique data value of a column
dates	Similar to data_values for date related columns
stats	Each row presents statistics of a numeric column of a table