Knowledge-Driven Framework for Designing Visual Analytics Applications

Madhushi Bandara
School of Computer Science & Engineering
University of New South Wales
Sydney, Australia
k.bandara@unsw.edu.au

Fethi A. Rabhi
School of Computer Science & Engineering
University of New South Wales
Sydney, Australia
f.rabhi@unsw.edu.au

***Abstract*— Machine learning and data analysis are becoming an essential part of the decision-making process in modern organizations. Even though new and improved analytics algorithms are developed frequently, organizations are struggling to develop analytics applications that can stay up-to-date with changing business requirements and technology innovations. The rapid development of ad-hoc programs to conduct machine learning tasks at hand has resulted in creating more expenses and efforts in the long term, a phenomenon referred to as technical debt in literature. This paper addresses the technical debt associated with data analytics applications by proposing a knowledge repository that captures analytics-related knowledge, which can be developed and maintained separately from the organization's IT infrastructure and used to design analytics applications with visual interfaces. This way, organizations can develop dynamic and adaptable analytics applications with easy-to-follow front-ends and can accommodate new data sources or machine learning models. We evaluate the proposed approach by conducting a case study that develops an application for the acquisition and management of high-frequency financial market data.**

Keywords—knowledge-driven, analytics process, data processing, visual analytics

I. Introduction

Modern industries of all domains and scales are looking to incorporate machine learning (ML) techniques and big data repositories to get better insights for their decision-making processes. Even though the area of machine learning has been widely researched among academics and researchers, deploying high quality, end-user friendly and long-lasting enterprise system incorporating machine learning techniques presents many challenges and obstacles. The literature report that analysts spend more than 80% of their time collecting, exploring, cleaning and pre-processing datasets, and only 20% of their time to spend on data analysis, modelling and interpretation [1]. Therefore, it is crucial to simplify data processing activities incorporating knowledge visualization, which will lead to producing high-quality datasets to be used in machine learning models. We also recognize the need of such activities to be automated and integrated with existing enterprise system infrastructure.

An ML model needs to be adaptable and refinable with new data and insights without degrading its quality, performance or simplicity. An ML application should also have the capability to incorporate new raw data which needs to go through cleaning, pre-processing and transformation stages so that the model can be retrained. Furthermore, new features need to be introduced with a minimum of modifications. To be able to leverage most existing ML software packages in a context that requires frequent user interactions, an enterprise information system needs to support the following data processing capabilities:

1. Ability to easily integrate new data sources
2. Switch between data sources that provide similar data
3. Easily update data pre-processing services to handle changes of data sources
4. Easily transform the structure and format of datasets to suit an ML model
5. Easily update or replace ML model features

Often, organizations rely on the programming and domain expertise of the data analysts to carry out these tasks manually via ad-hoc scripts. Resulting scripts are not reusable, understandable or maintainable over time. This is the reason that organizations need a better approach to managing data acquisition and pre-processing and several big IT vendors (e.g. IBM and SAP) already offer large and complex products that support integrated data management and analytics activities.

Instead of acquiring an expensive platform, we advocate a more pragmatic approach [2]. Organisations should aim at capturing the complex *knowledge* related to analytics operations, underlying assumptions and connections to the IT infrastructure so that they can be reused and maintained without necessarily involving the analysts who created them in the first place. To support this new approach, this paper presents the design of an innovative data processing platform that has the following characteristics: (1) It captures and stores the experience and knowledge accumulated by analysts related to data acquisition and pre-processing in a *knowledge repository* using semantic modelling principles, where ontologies are the basic building blocks. (2) This knowledge repository maintains a record of the relationships amongst data sources, dataset types, datasets, measures and variables, together with their transformation history. (3) This repository sits on top of an organisation's IT infrastructure and therefore, does not disrupt existing work practices or introduce dependencies on specific IT vendors. (4) This repository can be used to develop interactive, visual analytics applications that reduce cognitive burden of the analyst. Organizations can maintain and update the knowledge base over time with new data sources, structures and operations.

978-1-7281-9134-8/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00089

The remainder of this paper is organized as follows: Section II introduces related work. Section III discusses our proposed approach and Section IV provides a case study which demonstrates how to use the proposed approach to create analytics applications. Section V evaluates the approach by conducting data processing operations with high-frequency trading data related to the case study. The paper concludes with a summary and future work in Section VI.

II. Background and related work

Even though there are numerous papers on data analytics model development, surprisingly only a limited number of studies are focused on providing adequate analytics infrastructure such as knowledge repositories to assist the operations in the serving environment [3]. In a traditional machine learning project, once a data set is cleaned, processed and used to train the ML model, any knowledge about the process of creating the datasets and model, trade-off decisions made along the way and the rationale behind those decisions is lost. When a model has to be re-trained or updated to reflect new dataset characteristics, the whole trial and error process needs to be repeated. Skully et. al [4] concluded that ML applications carry significant *technical debt*. Due to lack of clear abstraction boundaries with specific intended behaviours, they summarise the behaviour of a machine learning component as "*Change Anything – Changes Everything*".

According to Google's analysis of the technical debt of AI systems [5], only a small component of real-world ML systems is the actual ML model. The required surrounding infrastructure is vast and complex and includes a variety of tools developed to automate the ML code as well as the data verification and feature extraction phases. To understand these problems further, let us look at the process of applying machine learning techniques over high-frequency financial data (or tick data). Tick data consists of real-time data produced from exchanges, with each tick corresponding to a discrete market event such as a quote, bid, sale or price movement. With the massive move of securities trading to online electronic platforms, huge volumes of such data have become available, providing detailed market insights such as the interplay between order flow, liquidity and price dynamics. This level of analysis is not at all possible with conventional daily data. Few characteristics of high-frequency data that elaborate its complexity are irregular temporal spacing, discreteness, diurnal patterns (different levels of volatility over time), and temporal dependence. Raw high-frequency datasets are characterized by their massive size and high dimensionality and in need of further processing before they can be used. Empirical results have confirmed that analysing hundreds of low-level indicators with machine learning methods is impractical and can cause a severe drop in performance of trading systems as well as the build-up of significant technical debt [6].

Existing approaches to address the ML technical debt include the use of meta-learning [7] or Service-oriented Architecture (SOA) and workflow-based platforms (e.g. ADAGE [8]) to conduct data processing. However, these approaches lack a sound information model that is sufficient to capture details of the data acquisition or transformation process in order to assist users' operations of the future [3].

To fill that gap, we identify the potential of visual analytics and knowledge modelling approaches. The knowledge modelling approach is supported by semantic technology as advocated by Berners-Lee et al. [9]. This new approach for modelling data and their semantics using ontologies has a well-developed set of standards and notations. These standards are supported by different tools for modelling, storing, querying and inferencing a knowledge repository. Also, they allow organisations to manage heterogeneous datasets and provide visualizations while still having the ability to infer information (via a reasoning engine) [10].

To identify how semantic technology is used to aid data analysts, we conducted a systematic literature survey [11]. We observed there is no semantic model that can capture multiple aspects of the data processing pipeline's expert knowledge, ranging from variables and data transformation process to data sources, with the ability to answer questions raised by analysts [12]. Our early research efforts have resulted in the Research Variable Ontology (RVO), which has been developed to address this gap [13]. Nonetheless, there is a need for more case studies to demonstrate the applicability of this approach in areas that require the development of ML models over several heterogeneous data sources and which require frequent model experimentation and refinements.

The visual analytics principles have the potential of turning knowledge accumulated in organization via semantic web technologies into an insightful analytics experience. As information visualisation has changed our view of databases, the goal of visual analytics is to make our way of processing data and information transparent for the analytics discourse [14]. The visualisation of related information and activities can provide the means of understanding related processes, instead of being left with only the results for comprehension by analysts and end-users. Visual analytics can foster the constructive evaluation, correction and rapid improvement of processes and models that can ultimately lead to improvements in knowledge management and decision making [14]. In relation to this area, Loom [15] and TUORIS [16] show the potential of interactive visual exploration techniques to increase our understanding about analytics related artifacts such as large data sets, especially in big data environments. At the same time, organisations like their user interfaces to be flexible enough to accommodate any changing user requirements [17] which has led to research in End-User Development (EUD).

The approach we propose in this paper will help to capture and represent analytics knowledge related to data processing activities via semantic web technologies and to utilize them in developing visual analytics tool.

III. Proposed approach

We propose the use of a knowledge repository that captures all knowledge related to data processing which can be developed and maintained separately from an organisation's IT infrastructure (see Fig. 1). This enables the development of visual data analytics applications that can leverage accumulated knowledge acquired through previous development projects for conducting various types of data processing operations including data acquisition and transformation. This architecture is flexible, enabling knowledge sharing between different analytics applications. It is based on previous work in the area of building architectures for data intensive science [13].

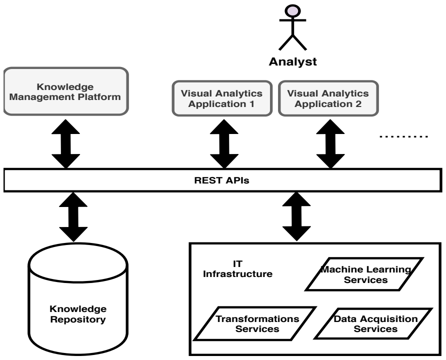

Analyst
Knowledge Management Platform
Visual Analytics Application 1
Visual Analytics Application 2
....
REST APIs
Knowledge Repository
IT Infrastructure
Machine Learning Services
Transformations Services
Data Acquisition Services

Fig. 1. Proposed Architecture that incorporates Knowledge Repository, and existing IT Infrastructure as services via REST APIs to develop knowledge-driven visual analytics applications.

A. Knowledge Repository

The knowledge repository is the key component of our architecture that elevates our approach from traditional service-oriented architecture. The knowledge repository is designed to store and link domain knowledge related to data, facts and findings related to ML models and information about available resources (i.e., data sources, ML models and transformations) together. The data analysts can interrogate the knowledge repository to learn and get recommendations from accumulated knowledge and use this knowledge for generating high-quality data for their ML models.

A snapshot of the ontology or the schema used for the knowledge repository, published as RVO [13] is shown in Fig. 2. By creating instances of these concepts that are adapted to their particular context, RVO is designed to inherently support knowledge integration. Any organization can integrate external ontologies and extend the capability of RVO to represent complex knowledge and use that knowledge to develop analytics platforms.

In our prototype, the knowledge repository has been implemented using RDF/OWL standards to encode and store knowledge in a Marklogic[1] graph database. This (non-SQL) database management system provides a SPARQL query language-based REST API to access and query information in the knowledge repository.

B. IT Infrastructure

The IT infrastructure in the proposed architecture represents any software program or API that can be used by the Analytics Applications to assist with data processing operations. Machine Learning, Data Transformation and Data acquisition services are particularly important.

C. Visual Analytics Applications and Knowledge Management Platform

The Visual Analytics Applications represent a number of end-user applications developed and provided for analysts through a graphical user interface (GUI) to utilize the IT platform and knowledge repository for quality data acquisition and processing. Such analytics applications can be developed to serve different needs of the analysts in the organisations. Few example functionalities they can provide are: 1. Study different data sources, services and ML models available 2. Download raw datasets from a selected data source, 3. Transform raw data into desired measures in specific format 4. Save and visualize datasets. Analytics applications can access the Knowledge repository via its REST API and run queries to get information about available *data sources, datasets, ML models and transformations.*

The Knowledge Management platform represents the user interface for managing the Knowledge repository of the organization by adding details about new IT infrastructure and domain knowledge. The Knowledge Management platform makes it possible for end-users to query and visualise that knowledge stored in the repository or insert new knowledge as in any conventional database.

IV. Case Study

A. Oveview

We conducted a case study to demonstrate and evaluate our proposed approach as part of a project collaboration between research teams in the School of Finance and the School of Computer Science and Engineering at the University of New South Wales, Australia. This project aims at developing analytics functions related to acquiring, pre-processing and managing high-frequency financial time-series data coming from multiple data sources in heterogeneous formats and standards.

In this case study, the objective is to develop an end-user application called Timeseries Builder that can create and

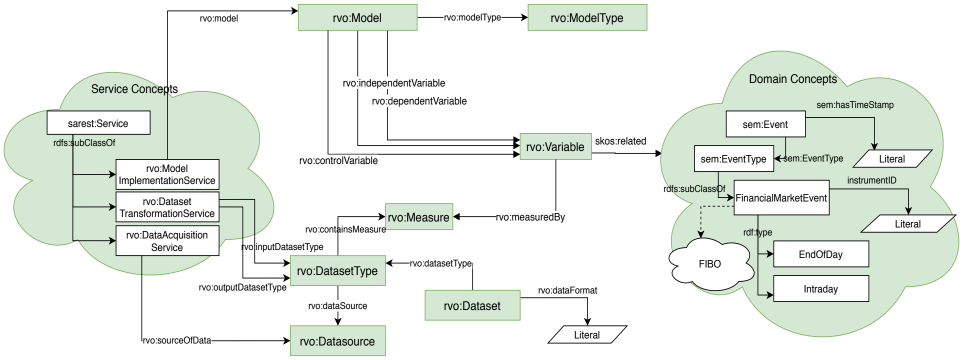


Fig. 2. The Research Variable Ontology that is used to create the knowledge repository, accumulating and integrating analytics related domain and service concepts with concepts related to variables, measures, ML models and datasets.

[1] https://www.marklogic.com

manage high-quality time series data related to Australian equity markets. This will help determine if the knowledge created as part of this project can be made available to other projects. The benefits will be reducing the set-up costs associated with developing new applications that require the analysis of high frequency Australian financial market data and make it easier to translate existing research to the Australian setting and to engage with industry in Australia.

In preparation for the evaluation, the following steps need to be performed:

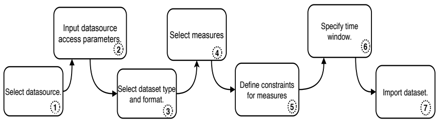
- Defining the Timeseries Builder end-user requirements
- Populating the knowledge repository
- Extending the IT infrastructure with data processing services
- Building the Visual Analytics Application Front-End

The evaluation should determine whether the knowledge repository is able to represent data coming from multiple data sources, with heterogeneous data representation formats and standards. Furthermore, a large set of standard *measures* for market liquidity, trading activity and price volatility at a stock-day level needs to be catalogued and organized around *variables* they represent, so that the data can be used for visualization and time series analysis. The knowledge repository needs to be able to identify the relevant services in the IT infrastructure to transform raw data into time series data with custom time series windows.

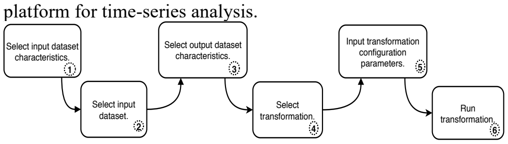
B. Timeseries Builder Basic Requirements

This application requires the following use cases to be supported:

- to import data from multiple data sources: visually browse through multiple data sources and formats provided by different data providers and select and download the desired dataset that contains a selected set of measures filtered by constraints and specific time window. This process is illustrated in Fig. 3.
- to transform a dataset in one structure and format to a standard structure and format: the analysts will need a user-friendly data transformation platform with visualizations to perform the conversion specifying characteristics of input and output datasets, selecting and configuring a suitable transformation function. This process is illustrated in Fig. 4.


Select datasource.
1
Input datasource access parameters.
2
Select dataset type and format.
3
Select measures
4
Define constraints for measures
5
Specify time window.
6
Import dataset.
7

- Figure 3: Steps for *scenario 1- Import data* through visual analytics platform for time-series analysis.


Select input dataset characteristics.
1
Select input dataset.
2
Select output dataset characteristics.
3
Select transformation.
4
Input transformation configuration parameters.
5
Run transformation.
6

- Figure 4: Steps for *scenario 2- Transform data* through visual analytics platform for time-series analysis.

C. Populating the Knowledge Repository

The first step is creating instances in the knowledge repository is capturing expert knowledge coming from financial researchers, analysts and literature. Fig. 5 illustrates a visualization of the instances stored in the knowledge repository for this case study. It shows how classes in RVO are used to organize and link information about *data sources, dataset types, measures, variables, datasets,* as well as different *services*. Also, domain knowledge related to financial markets is used to provide the context of different *measures*.

D. Building the Visual Analytics Application Front-End

Finally, the *Timeseries Builder* [2] front-end was implemented via R Shiny [3] framework. This analytics application has two graphical user interfaces (GUI) as separate tabs to conduct two use cases we described in section IV.B: *Import Data* and *Transform Data*.

V. Evaluation

The evaluation process consists of checking the following:

- Illustrate how a data analyst uses the Timeseries Builder end-user application (Sections V.A, and V.B)
- Evaluate the capabilities of the proposed architecture to represent organizational analytics knowledge and utilize them in reducing the technical debt (Section V.C)

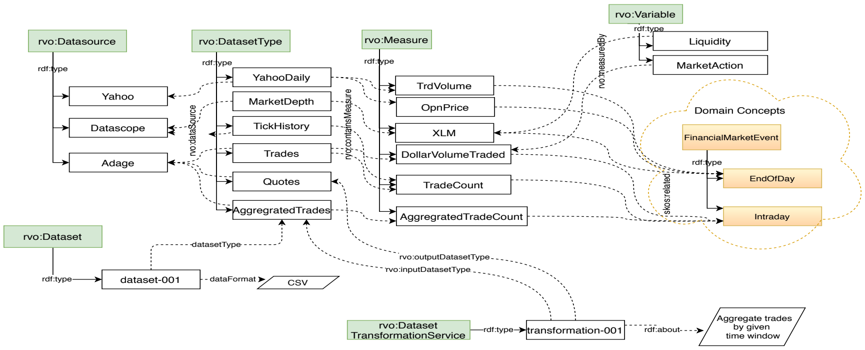


Fig. 5. Visualizing a snapshot of the knowledge repository for financial data for case study. Concepts from RVO (Fig. 2) are shown in green.

[2] http://adage2.cse.unsw.edu.au:3838/unsw-intradaymeasures-project-test/

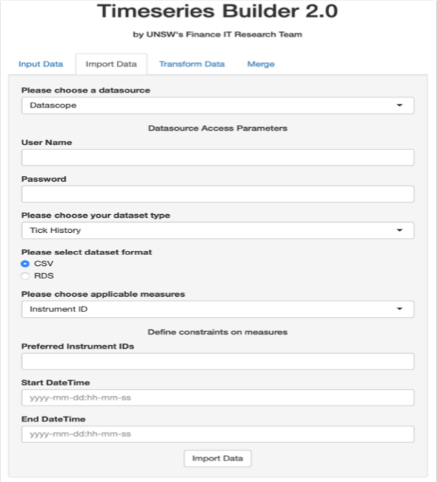
[3] https://shiny.rstudio.com

A. Use Case 1: Import Dataset

Fig. 6 illustrates the GUI that supports Scenario 1 (the *Import Data* tab). This GUI is organized from top to bottom, to support all the steps. As the first step analysts will select the data-source they need to access data from. When new *data sources* are added into the organization's IT infrastructure and modelled in the knowledge repository, this drop-down list is automatically updated via associated query. Based on the selected *data source,* in Step 2, another query will fetch the access parameters related to that *data source* from the knowledge repository and populate the Datascope Access Parameters section in GUI. GUI. For example, in Fig. 6, when *Datascope* is selected as the *data source*, the analyst is prompted to enter the username and password, because those are the authentication details required by the Datascope API.

Step 3 is to select the dataset type and format necessary for the analyst. In this example, the analyst has selected Tick History and CSV, respectively. In Step 4, the analyst should specify *measures* applicable to the imported dataset. This is a multi-select drop-down list, where analysts can select any *measure* they are allowed to filter or customize before downloading data. In Datascope *(Tick History dataset type),* the analyst can filter and download datasets by preferred instrument ids. So, in Fig. 6 analyst selects *Instrument ID* as the applicable *measure*.

Step 5 is to define constraints on the *measures* selected in step 4. Therefore, the GUI section for step 5 is dynamically created based on what *measures* are selected by the analyst. In our example, as *Instrument ID* has been selected as the *measure* in step 4, the analyst can enter a list of instrument ids at step 5. Step 6 is to specify a time window by providing the start and end date and time for the dataset. Finally, the analyst can click the *Import Data* button, and the resulting dataset will be ready to download on the right-side panel.



Timeseries Builder 2.0

by UNSW's Finance IT Research Team

Please choose a datasource: Datascope

Datasource Access Parameters

User Name

Password

Please choose your dataset type: Tick History

Please select dataset format
● CSV
○ RDS

Please choose applicable measures: Instrument ID

Define constraints on measures

Preferred Instrument IDs

Start DateTime: yyyy-mm-dd:hh-mm-ss

End DateTime: yyyy-mm-dd:hh-mm-ss

Fig. 6: Time Series Builder illustrating the *Scenario-1 - Import Data*

B. Use Case 2: Transforming Datasets

To conduct Scenario 2, the analyst will use the *Transform Data* tab. Fig 10 shows all the steps associated with it.

In Step 1, the analyst will specify input *dataset* characteristics. The analyst will select the data-source from the drop-down list. Options for the *dataset* structure drop-down list and format type list are dynamically populated based on the *data source* selected by the analyst.

Step 2 is to select the *dataset*. When the analyst clicks the Browse button, the platform will show available *datasets* for the analyst to select and upload. The third step is to specify the output characteristics of the *dataset*. Step 4 is to select the appropriate transformation.

A SPARQL query associated with *Select transformation* step will access the knowledge base and fetch available transformation implementations that match the input and output *dataset* characteristics specified by the analyst and populate the drop-down list for the analyst to select from. As this transformation does not require any configuration parameters, the analyst can skip step 5. Alternatively, if the analyst selects a transformation that requires additional configuration parameters, a query will fetch the list of parameters and GUI for *Input transformation configuration parameters* step will be dynamically generated. For example, in an alternative instance (shown in Fig. 7 right side), the analyst selects *Datascope Tick History to ADAGE Aggregated* as transformation and GUI prompts analyst to select a time window for aggregation.

The last step is for the analyst to run the transformation by clicking the button *Run Transformation*. This execution operation will conduct the transformation and display result dataset for the analyst, together with a download button.

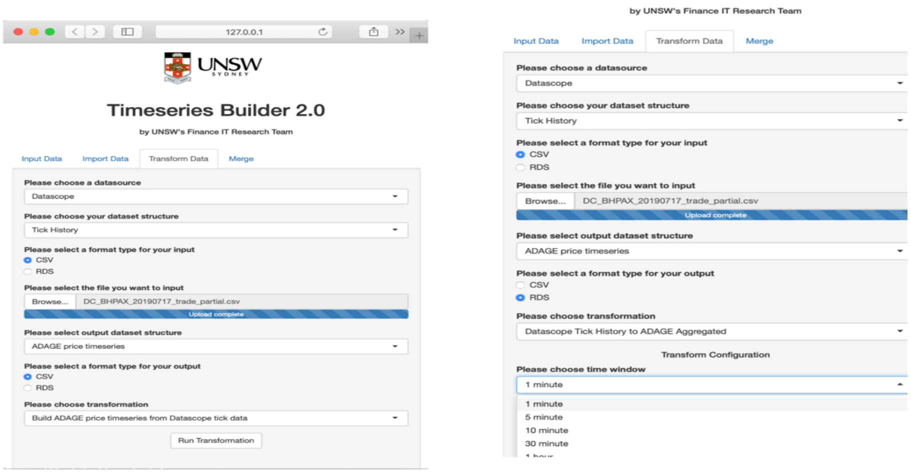
C. Discussion

Our experience showed that it was much easier for a software engineer to develop a visual analytics application backed by the knowledge repository. Also, end-users do not have to be aware of any underlying infrastructure or write software programs. Furthermore, we observe that our ontology and the knowledge repository can capture sufficient information to successfully support data acquisition and transformation activities that result in quality data that can be fed into ML models.

To evaluate if data processing knowledge from this project has been adequately captured, we assume that financial researchers want to start a new project that plans to analyse a new *variable* (e.g. VAR1). The first thing they want to do is find *datasets* that represent VAR1. With the knowledge repository in place, they can easily find out whether the related to VAR1. All they have to do is run a SPARQL query on the knowledge repository. The output of this query will list all *datasets* that have a *measure* for VAR1. Such recommendations can save the time and effort of researcher otherwise spent on literature surveys, navigating past reports or data archives.

To demonstrate how the proposed architecture can help to reduce technical debt by reducing long term maintenance effort, we conducted a few hypothetical changes related to case study implementation as described below and observed how the analytics applications are affected by them.

First is a change in the organizational IT infrastructure. we look at implications on the analytics platform when there is a requirement to add a new data source by adding a data acquisition service. The only action needed for this data source to be used within use case 1: Import Datasets is updating the knowledge repository with new *data source* details (name, access parameters, related *dataset type* etc.). When analysts use the application afterwards, a query will



127.0.0.1

Timeseries Builder 2.0

by UNSW's Finance IT Research Team

Input Data | Import Data | Transform Data | Merge

Please choose a datasource: Datascope

Please choose your dataset structure: Tick History

Please select a format type for your input: ● CSV ○ RDS

Please select the file you want to input: Browse... DC_BHPAX_20190717_trade_partial.csv — Upload complete

Please select output dataset structure: ADAGE price timeseries

Please select a format type for your output: ● CSV ○ RDS

Please choose transformation: Build ADAGE price timeseries from Datascope tick data

Run Transformation

by UNSW's Finance IT Research Team

Input Data | Import Data | Transform Data | Merge

Please choose a datasource: Datascope

Please choose your dataset structure: Tick History

Please select a format type for your input: ● CSV ○ RDS

Please select the file you want to input: Browse... DC_BHPAX_20190717_trade_partial.csv — Upload complete

Please select output dataset structure: ADAGE price timeseries

Please select a format type for your output: ○ CSV ● RDS

Please choose transformation: Datascope Tick History to ADAGE Aggregated

Transform Configuration

Please choose time window: 1 minute

1 minute
5 minute
10 minute
30 minute
1 hour

Fig. 7. Time Series Builder Platform illustrating the *Scenario-2* - *Transform Data*, without and with transform configurations .

fetch all available *data sources* from the knowledge repository to populate *data source* list in Fig. 9. Analysts can select the *data source,* and the rest of the GUI will be updated to reflect parameters related to it. Analysts can provide the inputs and get datasets they need from new source seamlessly. There is no need to update the analytics application program logic or GUI or train the analysts on handling new data source.

Second change is introducing a new ML service to the IT infrastructure. In a traditional analytics platform, a completely new application from back end services to GUI has to be written from scratch to transform existing data into the *dataset type* and *format* required this ML service. However, in the proposed architecture organization only needs to add a new data transformation *service* into the organization IT infrastructure and include its details in the knowledge repository. Then analysts can go to the same Transform Data GUI (shown in Fig. 7) and the new transformation will be available for selection from the drop-down list.

VI. Conclusion

This paper presents an approach for organizations to design their data processing operations for ML systems that reduce technical debt and ensure maintainability over a long period of time. We understand that organizational domain knowledge, analytics application requirements and technologies used for analytics can change frequently and design our solution so that organizations can cater to these challenges with minimum effort. We used high-frequency financial data analytics case study to evaluate the proposed approach.

As future work, we identify the need to evaluate the proposed approach incorporating end-user feedbacks, possibly applied to different analytics domains. Furthermore, use cases discussed in Section 5 only utilize data acquisition services and transformation services. We need to add machine learning services and study how they can be incorporated into end-user analytics applications as well.

References

[1] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989. [1] A. Gabernet, and J. Limburn, "Breaking the 80/20 rule: How data catalogs transform data scientists' productivity," 2017/ Available from: https://www.ibm.com/cloud/blog/

[2] F.A. Rabhi, M. Bandara, A. Namvar, and O. Demirors. "Big data analytics has little to do with analytics," in: Beheshti, A., Hashmi, M., Dong, H., Zhang, W.E. (eds.) ASSRI 2015/2017. LNBIP. Springer, Cham, vol. 234, 2018, pp. 3–17.. https://doi.org/10.1007/978-3-319-76587-7 1

[3] D. Crankshaw, J. Gonzalez, and P. Bailis, "Research for practice: prediction-serving systems," Communications of the ACM. vol. 61(8), 2018, pp. 45–49. https://doi.org/10.1145/3190574

[4] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, and M. Young, "Machine Learning: The High-Interest Credit Card of Technical Debt", 2014.

[5] D. Sculley et. al, "Hidden Technical Debt in Machine Learning Systems," Advances in neural information processing systems, 2015, pp. 2503-2511.

[6] N. Huck, "Large data sets and machine learning: Applications to statistical arbitrage," European Journal of Operational Research, vol. 278 (1), 2019, pp. 330-342.

[7] P. Brazdil, C. G. Carrier, C. Soares, and R. Vilalta, "Metalearning: Applications to data mining." Springer, Heidelberg, 2008.

[8] L. Yao, and F.A. Rabhi, "Building architectures for data-intensive science using the adage framework," Concurrency and Computation: Practice and Experience. vol. 27(5), 2015, pp. 1188-1206.

[9] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," Scientific American. vol. 284(5), 2001, pp. 28-37.

[10] A. Myłka, A. Myłka, B. Kryza, and J. Kitowski. "Integration of heterogeneous data sources in an ontological knowledge base," Computing and Informatics. vol. 31, 2012, pp. 189-223.

[11] M. Bandara, and F.A. Rabhi. "Semantic modeling for engineering data analytic solutions," Semantic Web. vol. 11(3), 2020, pp. 525-547.

[12] G. Shmueli, and O.R. Koppius. "Predictive analytics in information systems research," MIS Quarterly, vol. 35, 2011, pp. 553 -572.

[13] M. Bandara, A. Behnaz, and F.A. Rabhi, "RVO- The Research Variable Ontology," European Semantic Web Conference, 2019 pp. 412-426.

[14] D. Keim, G. Andrienko, J.D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, "Visual analytics: Definition, process, and challenges," Information Visualisation. 2008, pp. 154-175.

[15] J. Brook, F. Cuadrado, E. Deliot, J. Guijarro, R. Hawkes, M. Lotz, R. Pascal, S. Sae-Lor, LM. Vaquero, J. Varvenne, and L. Wilcock. "Loom: Complex large-scale visual insight for large hybrid IT infrastructure management," Future Generation Computer Systems. vol. 80, 2018, pp. 47-62.

[16] V. Martínez, S. Fernando, M. Molina-Solana, and Y. Guo. "Tuoris: A middleware for visualizing dynamic graphics in scalable resolution display environments," Future Generation Computer Systems. vol. 106, 2020, pp. 559-571.

[17] M. Bandara, A. Behnaz, F. A. Rabhi, and O. Demirors. "From requirements to data analytics process: An ontology-based approach," Lecture Notes in Business Information Processing, Business Process Management Workshops, BPM 2018 vol. 342, 2018