2018 IEEE Big Data Congress (BigData 2018), San Francisco, USA, 2-7 July (c) IEEE 2018

A Survey of Current End-user Data Analytics Tool Support

Hourieh Khalajzadeh[1], Mohamed abdelrazek[1], John Grundy[2], John Hosking[3], Qiang He[4]
[1]*School of Information Technology, Deakin University, Australia*
[2]*Faculty of Information Technology, Monash University, Australia*
[3]*Faculty of Science, University of Auckland, New Zealand*
[4]*School of Software and Electrical Engineering, Swinburne University of Technology, Australia*
{h.khalajzadeh, mohamed.abdelrazek}@deakin.edu.au, john.grundy@monash.edu,
j.hosking@auckland.ac.nz, qhe@swin.edu.au

Abstract— There is a large growth in interest in big data analytics to discover unknown patterns and insights. A major challenge in this domain is the need to combine domain knowledge – what the data means (semantics) and what it is used for – with data analytics and visualization techniques to mine and communicate important information from huge volumes of raw data. Many data analytics tools have been developed for both research and practice to assist in specifying, integrating and deploying data analytics and visualization applications. However, delivering such big data analytics application requires a capable team with different skillsets including data scientists, software engineers and domain experts. Such teams and skillset usually take a long time to build and have high running costs. An alternative is to provide domain experts and data scientists with tools they can use to do the exploration and analysis directly with less technical skills required. In this paper we present an overview and analysis of several current approaches to supporting the data analytics for end-users, identifying key strengths, weaknesses and opportunities for future research. Our key findings include: the existing tools focus on the machine learning modeling and implementation aspect of the software development lifecycle of AI-powered systems. Furthermore, they are complicated for a domain expert with no data science and programming background to conduct initial exploration or discovery of the problem and are not designed to enable collaboration between software engineers, data scientists and domain experts involved in the development of the AI-powered software systems.

Keywords-data analytics; data visualization; machine learning; domain specific visual languages; big data

I. INTRODUCTION

Using “big data” to improve decision-making has recently become a highly active research and practice area [1, 2]. Gartner’s technical professional advice [3] for machine learning applications recommended six stages: classifying the problem, acquiring data, processing data, modeling the problem, validation and execution, and finally deploying. Classifying the problem describes how to categorize the problem or the research question to solve – objectives, success criteria, assess constraints, etc. Acquiring data identifies where to find the data to support the problem. Processing data involves how to prepare data for further analytics tasks. The steps in this stage include data transformation, normalization and cleansing, as well as the selection of training and test sets for supervised learning. Modeling the problem determines the range of machine learning algorithms to be used given the category of the problem and the type and volume of the available datasets. Validation and execution involve validating the results, determining the platform on which to execute the models and algorithm, and tuning and refining results. Finally, the output of the machine learning process is deployed to provide some form of business value. This phase entails determining where and how to deploy the results e.g. visualizing the results to aid decision making.

To be effective, a big data analytics software needs to support: diverse data ingestion, wrangling and cleansing; data integration and querying for very large data volumes; feature extraction and selection; tailoring and combination of diverse data analytics techniques, software and services; and communication of findings and integration with existing IT solutions. This gets more complicated when addressing quality of service attributes including: scalability, privacy, security, reliability and adaptability to changes in the target environment.

Recently, a number of data analytics and machine learning tools have become increasingly popular, providing packaged data sourcing, integration, analysis and visualization toolkits oriented towards end users such as Azure ML Studio [4], Amazon AWS ML [5], Google Cloud ML [6], and BigMl [7]. Many of these tools do not require programming language knowledge and are based on using simple drag and drop interfaces. However, while end user configuration is desirable, these tools suffer from lack of extensibility, scalability and ability to integrate diverse third party solutions; capacity to handle complex problems; still focus on the machine learning algorithms wiring (implementation) and sometimes one-click deployment, but lack domain knowledge and business problem capturing, modeling, traceability to the solution and validation of the solution against the problem; lack of explanation of the technical model – i.e. from end-user perspective, is this the best model they can get give their dataset and more importantly why they are getting a certain answer when they fed in an example input. These tools would be very useful for end-users and data scientists, for discovery and exploration phase, to be able to model the problem, extract insights/patterns, develop predictive and clustering models if it is feasible before they can get software engineers involved. This paper discusses the pros and cons of several existing tools, and identify key gaps and requirements.

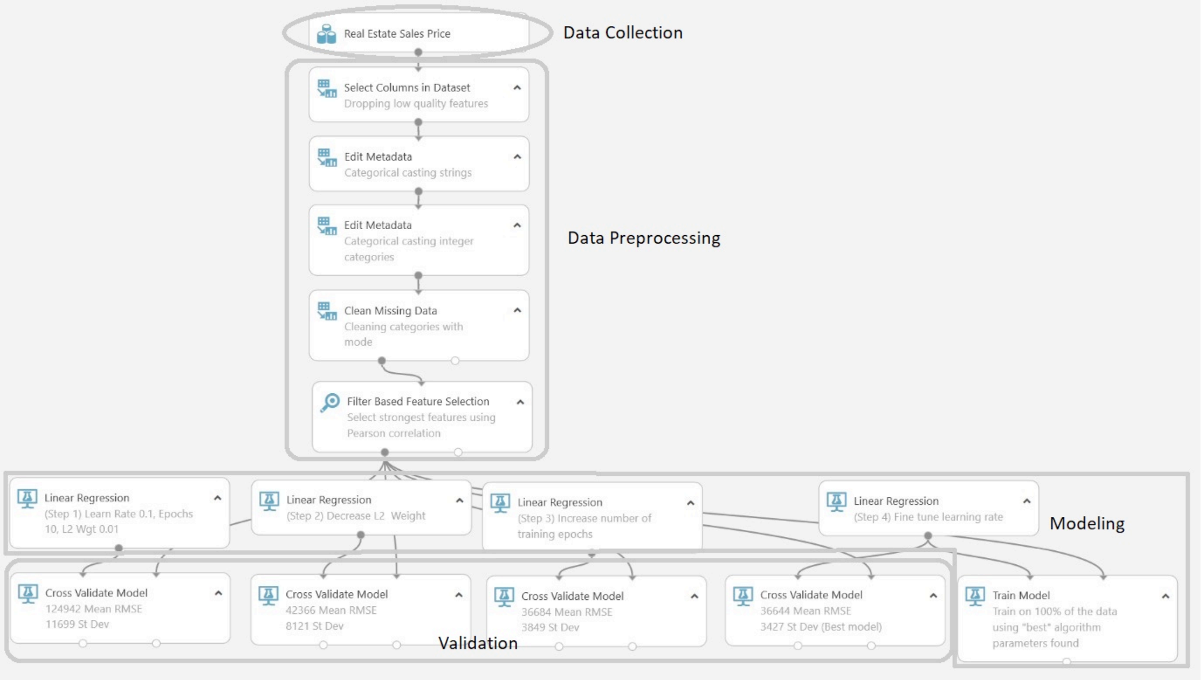

Real Estate Sales Price
Data Collection
Select Columns in Dataset
Dropping low quality features
Edit Metadata
Categorical casting strings
Edit Metadata
Categorical casting integer categories
Data Preprocessing
Clean Missing Data
Cleaning categories with mode
Filter Based Feature Selection
Select strongest features using Pearson correlation
Linear Regression
(Step 1) Learn Rate 0.1, Epochs 10, L2 Wgt 0.01
Linear Regression
(Step 2) Decrease L2 Weight
Linear Regression
(Step 3) Increase number of training epochs
Linear Regression
(Step 4) Fine tune learning rate
Modeling
Cross Validate Model
124942 Mean RMSE
11699 St Dev
Cross Validate Model
42366 Mean RMSE
8121 St Dev
Cross Validate Model
36684 Mean RMSE
3849 St Dev
Cross Validate Model
36644 Mean RMSE
3427 St Dev (Best model)
Train Model
Train on 100% of the data using "best" algorithm parameters found
Validation

Figure 1. Real estate sales price prediction project in Azure ML Studio

The rest of this paper is organized as follows. In Section 2, the motivation and background of this research are described. Current end user tools are analyzed and compared in Section 3 while the strengths and weaknesses are discussed in Section 4. The future research directions are presented in Section 5. Finally, the paper is concluded in Section 6.

II. MOTIVATION AND BACKGROUND

Consider a property price prediction problem based on the solution developed in Azure ML Studio and based on the Ames Housing dataset [8] as shown in Figure 1. In this scenario, a real estate agency wants to improve agents' focus and outcomes by looking for key patterns in a large amount of real estate, government and financial data. The company employed a technical team of software engineers and data analytics to work on the problem. In this figure, the user drags the "Real Estate Sales Price" module to upload input data, "Select Columns in Dataset", "Edit Metadata", "Clean Missing Data", and "Filter Based Feature Selection" modules to prepare and clean data, "Linear Regression" and "Train Model" modules to apply linear regression and train the model, and finally "Cross Validate Model" module to validate the model. To choose the modules and change the properties, user needs to have data science knowledge. If user wants to use a preprocessing method or apply a model which does not exist in the list of modules, knowledge of programming languages such as Python and R is required to add some features to a certain extent.

However, the company realized that the team lacks understanding of domain knowledge. The company appointed a senior real estate agent, who is highly experienced with the nuances in the domain, to help the team building the analytics solution. The team is struggling with the lack of a common dialect between engineers, scientists and domain experts. Eventually, the team realized that the company is not ready due to issues in the available dataset that need to be rectified. It took a long time for the team to design and build a working analytics solution. A few months later, after the team was adjourned, the company wanted to add new features – build a new model to predict who will be willing to sell their property in a given suburb. In addition to, the company started to notice degradation in the performance of the existing property price prediction model, which need to be updated. The company recruited a new team which is struggling to develop models for the new capability – due to lack of business knowledge - in addition to updates to the existing model and requirements to integrate both.

In order to solve this problem, using many conventional approaches, such an end user would need to have a basic knowledge of a data analytics programming languages such as Python and R and also basic knowledge of data science. The end user would choose the features based on the quality of the features, apply data type casting and data cleansing, and finally filter the features to build the dataset as an input for the prediction model. Then the user needs to choose the best model, algorithm, and validation method and adapt the characteristics based on the problem requirements. Moreover, if the user has programming knowledge, then e.g. R or Python code can be embedded to improve some of the steps.

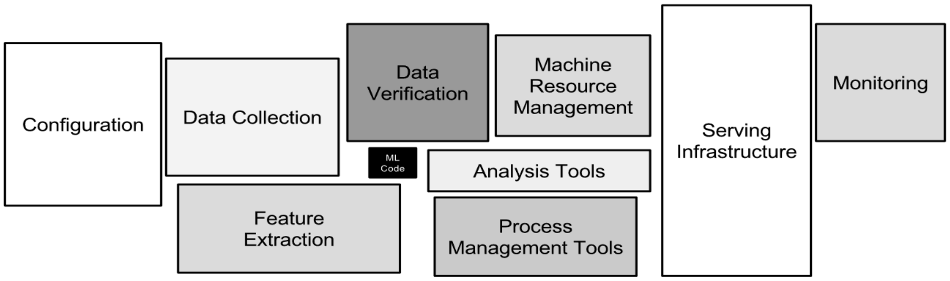


Figure 2. Artificial intelligence systems development building blocks [9]

However, these tasks are only small component in the building blocks necessary to build AI systems, as shown by the small black box in the middle of Figure 2, covering the Machine Learning (ML) code, and the box in the bottom covering feature extraction. There are other major activities that need support, including designing the analytics pipeline, collecting and cleansing data, determining features from the dataset needing extraction, managing machines and processes for large scale analytics, presenting and using the resultant information, and monitoring the processes.

Hence, key requirements that we have identified for such end user data analytics tools include the need:

- To cover all data preprocessing operations such as cleaning, wrangling, anomaly detection and so on.
- For tools to be easy to understand and use for all groups of users including domain experts, data scientist and so on. Users with no data science and programming knowledge should be able to use them.
- For tools to cover a variety of the algorithms for each stage of data processing, modeling and evaluation processes e.g. different data integration, analysis and visualization algorithms and approaches.
- For experienced users such as data scientists to have the ability to add their own features to the tool and incorporate their own specialized functions and algorithms. Therefore, the tool needs to offer flexible options for the users in case they want to expand the features based on their requirements.
- To cover the AI-SDLC stages including business problem description, requirements, design, implementation, testing and deployment.
- For tools to be industry ready in order to be used for large scale industry-based projects, including scaling to very large datasets and leveraging large compute platforms.
- Cost effectiveness is another required characteristic for the data analytic tools. In terms of deployment, users need to have the ability to deploy their solutions on the cloud or on premises or both.

We use these requirements below to analyze several research and commercial toolsets, to compare and contrast them, and identify key strengths, weaknesses and gaps.

III. A Review of Current End User Data Analytics Tools

A traditional software development lifecycle (SDLC) including the following activities: analysis of the requirements, design, implementation, testing, maintenance. The current practices and tools do not cover most of the activities in the analysis and design. The existing tools do not cater for the business requirements and focus mainly on the implementation phase.

According to Google technical debt of AI systems in [9], only a small component of real-world ML systems is the ML model. The required surrounding infrastructure is vast and complex. There are a variety of tools developed to automate the ML code as well as the data verification and feature extraction phases.

We group these components (building blocks of an AI-powered system) into three groups: data-related activities (DataOps); artificial intelligence and machine learning -related activities (AIOps); and development and deployment activities (DevOps). The DataOps activities include data collection/ingestion, data validation cleansing, wrangling, filtering, union, merge, etc. AIOps covers feature engineering and model selection, model training and tuning. Finally, DevOps covers model integration and deployment, monitoring and serving infrastructure.

From the perspective of DataOps, there are many tools such as Tableau [10], Plotly [11], and Trifacta [12] that focus on the data operations such as visualization, data cleaning, data wrangling, and so on.

As a well-known and popular example, Tableau provides end user support for visualizing, analyzing, and understanding the data. End users are typically any business person wanting to explore data insights. It offers an interactive visual analysis toolbox which allows its users to obtain insights that support their business. An example of Tableau in use is shown in Figure 3. In this example the user wants to gain some insights on housing sale and profit within different cities. Users need to choose and connect to their data stored in a file or a server. By loading the dataset, the features are accessible for the users and they can drag and drop different features and generate different charts and add more fields to obtain the right level of detail based on their need. Then, the users are ready to begin focusing on their results and use filters and colors to help them explore their data visually. For example, the users might want to know sales and profit in places which have some problems and therefore build a map view to leverage this issue and find a solution for it. In this example, Tableau has automatically assigned the proper geographic roles to different countries and the states, cities, and postal code fields within different countries. Finally, once all the visualizations are within the desirable extent by including all the details, users can build a dashboard to keep and use their insights and finally to create a story in Tableau to present their insights and walk the viewers through their data discovery process.

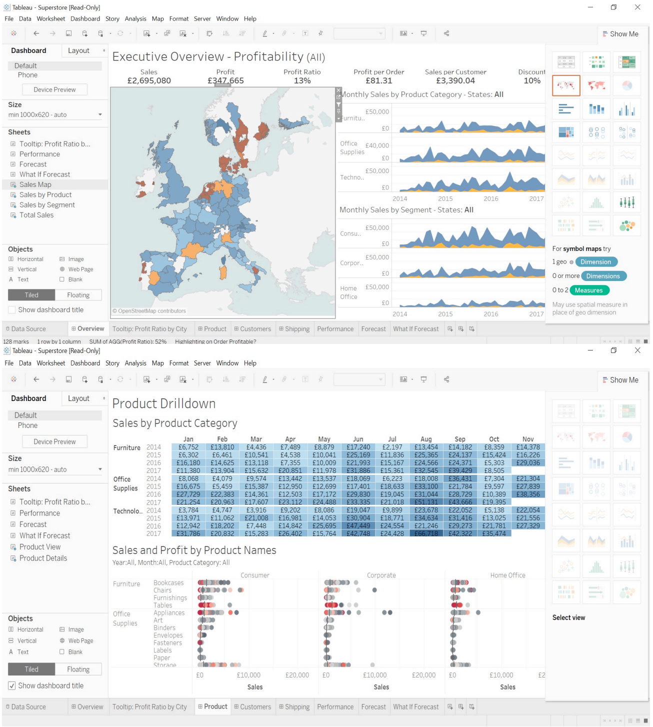


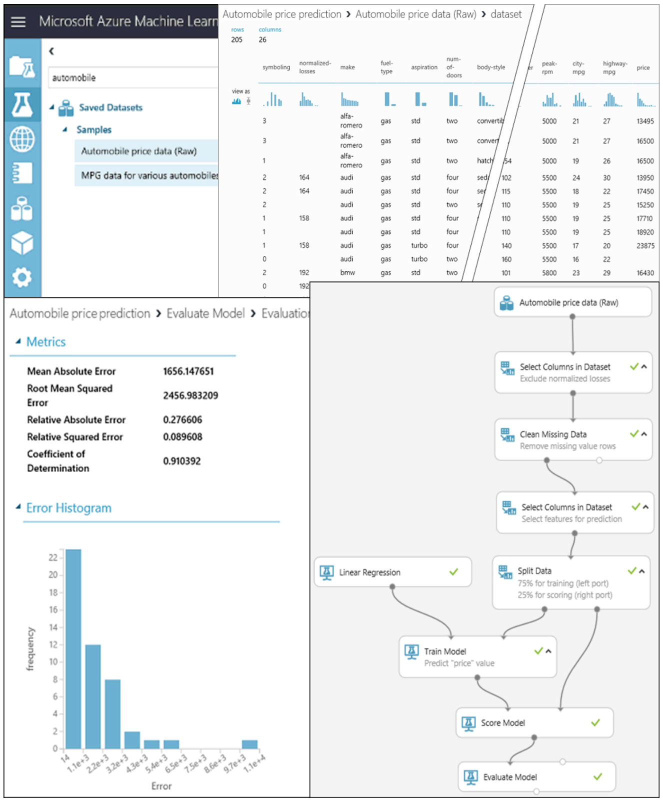
Tableau - Superstore [Read-Only]
Dashboard
Layout
Executive Overview - Profitability (All)
Sales £2,695,080
Profit £347,665
Profit Ratio 13%
Profit per Order £81.31
Sales per Customer £3,390.04
Discount 10%
Monthly Sales by Product Category - States: All
Monthly Sales by Segment - States: All
Sheets
Objects
Show Me
Product Drilldown
Sales by Product Category
Sales and Profit by Product Names
Consumer
Corporate
Home Office
Sales

Figure 3. An example of Tableau in use for real estate data analysis.

From the viewpoint of AIOps, there are a large number of tools focusing on the artificial intelligence and machine learning operations. Most of the tools in the market belong to this group. Some examples are Azure ML Studio [4], Amazon AWS ML [5], Google Cloud ML [6], BigMl [7], Weka [13], Rapidminer [14], IBM Watson ML [15], SAS [16], KNIME [17], and Tensorport [18]. The tools in this group also often cover DataOps to some extent.

A representative example of such tools is Azure ML Studio. This tool provides different data visualization and preprocessing, and machine leaning algorithms with an interactive drag and drop method. Target end users include business analysts but also experienced data scientists. An example of this tool in use is shown in Figure 4. In this example, price estimation for vehicles, users create data analysis experiments by dragging different modules to the canvas to do the analysis. To start, users upload their own datasets or find a related dataset to their problem from a list of available datasets and drag the chosen dataset to the experiment canvas. Then, they visualize and prepare their data. To preprocess the data some modules are available such as “Select Columns in Dataset”, “Clean Missing Data”, “Apply Filters”, “Add Columns”, “Add rows”, and many other modules with different editable properties. Different modules can be connected, and the experiment can be run at any stage.

After reading, visualizing, and preprocessing the data, data needs to be split to training and testing parts by dragging the “Split Data” module to the canvas and modifying the properties. Then, a variety of learning algorithms such as logistic regression, decision forest, neural network, decision tree, and support vector machine, can be applied to the dataset by choosing the appropriate modules. Then, the dataset can be trained, tested, and evaluated and the results can be visualized simultaneously. Finally, users can iteratively change and add or delete new modules to improve the model. For expert users, R and Python scripts can also be embedded to the model.

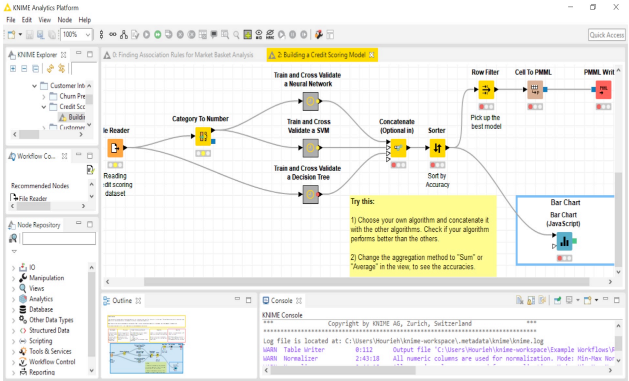

Microsoft Azure Machine Learn
automobile
Saved Datasets
Samples
Automobile price data (Raw)
MPG data for various automobiles
Automobile price prediction › Automobile price data (Raw) › dataset
Automobile price prediction › Evaluate Model › Evaluation
Metrics
Error Histogram
frequency
Error
Automobile price data (Raw)
Select Columns in Dataset
Exclude normalized losses
Clean Missing Data
Remove missing value rows
Select Columns in Dataset
Select features for prediction
Linear Regression
Split Data
75% for training (left port)
25% for scoring (right port)
Train Model
Predict "price" value
Score Model
Evaluate Model

Mean Absolute Error	1656.147651
Root Mean Squared Error	2456.983209
Relative Absolute Error	0.276606
Relative Squared Error	0.089608
Coefficient of Determination	0.910392

Figure 4. An example of Azure ML Studio in use

Finally, from the DevOps point of view, some of the available tools focus on the deployment of the solutions on the cloud or on premises as well as building industry ready solutions. This group covers some of the tools within the second group. Some examples of the tools preparing industry ready solutions deployable on both cloud and on premises are Rapidminer, IBM Watson ML, SAS, and KNIME.

An example of such a tool is KNIME. This provides an industry ready open source data analytics solution covering all the data, AI, and deployment operations. Target end users are data scientists and experienced end users wanting to build and deploy comprehensive AI-based solutions. Nodes can be added in KNIME by expanding the "Node Repository" and dragging and dropping different nodes to the workflow editor window. As shown in Figure 5, the "File Reader" icon is used for uploading datasets by expanding the "IO" and the contained "Read" category. A variety of nodes such as classification and clustering algorithms, manager node as well as interactive table and scatter can be used to visualize data and build different models. Then, nodes can be connected together in order to get the data flowing. Finally, nodes can be iteratively configured and executed.



Try this:

1) Choose your own algorithm and concatenate it with the other algorithms. Check if your algorithm performs better than the others.

2) Change the aggregation method to "Sum" or "Average" in the view, to see the accuracies.

```
***                    Copyright by KNIME AG, Zurich, Switzerland                    ***
*******************************************************************************************
Log file is located at: C:\Users\Hourieh\knime-workspace\.metadata\knime\knime.log
WARN  Table Writer          0:112       Output file 'C:\Users\Hourieh\knime-workspace\Example Workflows\F
WARN  Normalizer            2:43:18     All numeric columns are used for normalization. Mode: Min-Max Nor
```

Figure 5. An example of KNIME in use

IV. Strengths and Weaknesses

We have selected a few representative tools within the three groups of DataOps, AIOps, and DevOps and their key strengths and weaknesses are compared. We use several viewpoints including being ready and efficient to be used in industry; the cost, being easy to use, comprehensiveness in terms of the algorithms and models they cover, flexibility by offering the ability to embed code in different languages such as Python and R by the end user, and finally being usable for non-data science experts with no knowledge of data analytics and science. A summary and comparison of selected tools is shown in table 1. Some AI-SDLC phases as discussed in Section 3 are used in this table to reflect what phases are covered by each of the tools. The AI-SDLC phases are: business problem description, requirements, design, implementation, testing and deployment and are then classified to a group of broader phases of Dataops, AIOps, and Devops as introduced in Section 3.

4.1. DataOps Tools

Within the first group of DataOps, Tableau, Plotly, and Trifacta wrangler are chosen and compared in Table 1. Tableaue is a very comprehensive and easy to use tool used by many organizations to visualize and understand data. No knowledge of data analytics and programming is required to use this tool. At the same time, users can embed their own code to extend the usability of this tool, however to some extent only. Plotly provides open source Matlab, Python and R libraries for data scientists to create interactive charts in Matlab, Python, and R. It also offers different plans on library-specific support options including code review, hands-on assistance with implementation via email, chat and phone for developers, teams, and non-experts. Trifacta wrangler automatically profiles the uploaded data, shows the percentage of the missing, mismatching, or inconsistent values and also gives the data's distribution using histograms. Users can instantly gain some insights into their data quality using this tool but it is not usable on large datasets. All these tools, however, are only helpful for visualizing, preprocessing, and gaining insights on data. For using the gain insights to create a model and build solution, we need the next group of the tools.

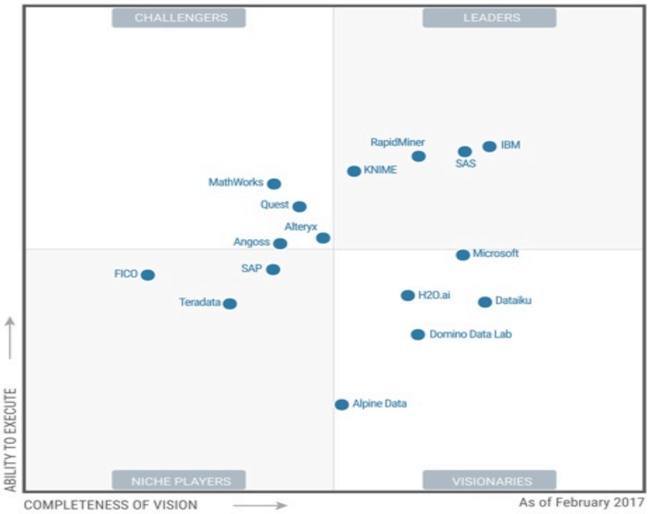
4.2. AIOps Tools

Most of the existing tools cover AIOps. The first three tools chosen in this group are the well-known Cloud based machine learning tools, Azure ML Studio, Amazon AWS ML, and Google Cloud ML. Azure ML Studio is very efficient and easy to understand and use. Also, for the benefit of data scientists or users familiar with programming, Python and R codes can be added to expand features where required. However, for non-expert users a basic knowledge of data science is required. Amazon AWS ML is also easy to understand and use but does not support embedding scripts and code in different languages such as R or Python for data scientists same as Azure ML Studio. Moreover, it needs a basic knowledge of the data science as well. Google Cloud ML offers more features than Azure and Amazon, but good knowledge of the TensorFlow [19] library is required to use the features, so, it is not very easy to use for non-expert users as a deep technical knowledge of ML is mandatory.

BigML is chosen as a both visual and programmatic machine learning tool for developers. It is easy to use BigML but it is not possible to manipulate and change the algorithms. Therefore, it is not possible to use any algorithms and models other than the few algorithms it is provided. Weka, is a very easy to use tool which is mostly used for research and learning. Weka does not cover all the algorithms and it is not possible to manipulate and add code to change the functionality of any the algorithms. It is mostly suitable for small datasets and simulations, not large-scale projects. Finally, TensorPort, the last tool chosen in this group, is a machine learning platform covering AIOps only to make use of Tensorflow, an open-source software library widely used for dataflow programming across a range of tasks. Tensorport is very comprehensive and the programmers have the freedom to change everything based on their needs. However, a deep knowledge of data analytics and machine learning is needed in order to use this tool. Using these tools users can build their data analytics solutions. Some of these tools cover DataOps as well. However, not all of them are industry ready and usable for large scale problems. Industry-based tools fit in the next group of the tools.

TABLE I. COMPARISON OF DIFFERENT DATA ANALYTICS TOOLS

End Users Tools	SDLC phases								Tool usability						
	Business problem description	Requirements	Design	Implementation		Testing	Deployment								
							DevOps			Cost					
				DataOps	AIOps		Cloud based	On premises	Industry ready	Free trial/for limited access	Plan based/pay as you go	Usability	Comprehensiveness	Flexibility	No Data science knowledge required
Tableau				✓			✓	✓	✓	✓	✓	✓	✓		✓
Plotly				✓			✓	✓		✓	✓		✓	✓	
Trifacta				✓			✓			✓		✓			✓
Azure ML Studio				✓	✓		✓			✓	✓	✓		✓	
Amazon AWS ML				✓	✓		✓				✓	✓			
Google Cloud ML				✓	✓		✓			✓	✓		✓		
BigML				✓	✓		✓	✓		✓	✓	✓		✓	
Weka				✓	✓			✓		✓		✓			
Rapidminer				✓	✓		✓		✓	✓	✓				
IBM Watson ML				✓	✓		✓	✓	✓		✓	✓	✓	✓	
SAS				✓	✓		✓	✓	✓		✓		✓	✓	
KNIME				✓	✓		✓		✓	✓		✓	✓	✓	
TensorPort					✓			✓	✓	✓			✓	✓	



Source: Gartner (February 2017)

Figure 6. Gartner 2017 Magic Quadrant for Data Science Platforms [20]

4.3. DevOps Tools

Finally, in the DevOps tool category, a variety of tools have been produced, such as Rapidminer, IBM Watson ML, SAS, and KNIME. These were selected due to being reported as the leaders within industry-based tools according to [20], based on completeness of vision and ability to execute, as shown in Figure 6.

These tools provide the ability to develop industry ready solutions deployable on the cloud or on premise, and they cover DataOps as well as AIOps. However, Rapidminer is not very easy to use and does not have an efficient user interface. It is also very slow with big datasets. IBM Watson ML is easy to use and it offers both visual drag and drop for non-experts and code based modelings for experts. However, knowledge of data science is required to use this tool. SAS is very comprehensive for industry-based usage. However, it is expensive and knowledge of programming is required. Finally, KNIME is very comprehensive and easy to use. It is possible to change and manipulate the algorithms based on every ones needs. It is also very efficient and quick. However, like the other tools, knowledge of data analytics is required in order to use this tool.

Based on these comparisons, ML Azure studio and KNIME are among the most comprehensive and easiest tools to use. However, they still need considerable knowledge of data analytics and data science as well as a basic knowledge of programming to be applicable to a broader range of domains. The problem with most of these tools is that they are complicated for a non-expert to make use of. At the same time, they are over-simplified for expert data scientists. These experts usually do not use them because they find writing their own programs using programming languages such as Python and R more comprehensive and applicable to different domains. Moreover, they only cover the few phases of data verification, feature extraction, and machine learning based on Figure 2.

In terms of the data verification phase, a dataset containing the features and examples needs to be collected and provided before getting started with any of these tools. Based on [21], the common issues with tools such as Weka, R [22], Orange [23], KNIME, Scikit learn [24], Mahout [25], and Spark Mllib [26] are that they are single threaded and provide only desktop applications for small datasets. They are also over complicated for non-experts, over-simplified for real world problems, and poorly engineered for real world or high scale usage. Additionally, commercial tools such as SAS and SPSS [27] inherit the same issues and are expensive.

V. RESEARCH DIRECTIONS

Based on the strength and weaknesses of different tool discussed in this paper, there are several key directions for the future research in developing tools for data analytics problems:

At present most tools focus on low-level data analytics process design, coding and basic visualization of results. There is a need to better capture requirements, changes in the requirements, and adaptation of the specified process. In many ways these AI-based systems are no different in terms of their need for good requirements and high-level design models from the other traditional systems. So saying, we want to better support domain expert end users in their requirements management for AI-based systems, providing approaches to capture their requirements not so much about the software solution but domain problem, available data and business intelligence needed to solve it.

Most current tools assume data is in a form more or less amendable to processing by ML algorithms. However, many real-world datasets are not "clean" nor "integrated", and great effort is needed to source the data, integrate and harmonize it, pre-process and cleanse it, organize it according to required algorithm needs, and then provide it for processing. Complex, large datasets introduce many issues that need to be supported, including handling partial and incomplete datasets.

Many existing tools need support to be expandable based on the user' requirements. Only a few of these tools, such as Azure ML Studio, BigML, IBM Watson ML, SAS, KNIME, and TensorPort offer the ability for the data science expert to embed new code and expand the algorithms and usability based on their own needs. On the other hand, including this feature must not make the tool over complicated to remain usable for non-expert users. Therefore, both simplicity for non-experts with no data science and programming knowledge, and support for expansion and tailoring for data science experts both need to be provided.

Most current tools only cover the DataOps, AIOps, and DevOps parts of the data analytics life cycle. Some, such as Tableau and Tensorport, cover one phase and some such as Rapidminer and KNIME, do try and cover all the phases. However, as discussed above, data verification, feature extraction, and ML code are only small parts of the whole systems development life cycle. At present there is no tool covering all the phases such as analysis of the requirements, design, implementation, testing, and maintenance. Therefore, there is a gap in providing tool features to capture the requirements and changes in requirements as well as adapting the solution based on these changes.

Many real-world problems require large datasets to be processed, require computationally expensive algorithms, and thus require deployment of solutions on complex, powerful computing infrastructure. Some tools hide all of these details from the user but provide limited configuration and flexibility; some tools require detailed knowledge of users in configuring and deploying solutions; and many only support limited single desktop solutions that do not scale. Supporting scaling and distribution is required for many real-world applications, but balancing this against limited end user knowledge of computing platforms.

Many tools provide a variety of visualization support to show results to end users to support business decision making with the information produced by AI and ML algorithms. However these are typically limited to built-in visualization options or the specification of complex visualizations require programming knowledge. Some visualizations allow end user information exploration while others are static displays. Further enhancement of information visualization capabilities including interactive exploration and end user specification of complex visualisations for their target domain are needed.

VI. Conclusions and Future Work

Many new data analytics tools covering different data analytics phases, including DataOps, AIOps, and DevOps, have been developed for both research and industry usage. We have introduced, summarized and compared a number in this paper. Such tools help data scientists to specify, integrate and apply complex data analytics and visualization techniques to build a range of big data applications. Based on our analysis of these tools we see that most of these tools currently focus on the data analytics and machine learning modeling and implementation phase. This is only a small part of the AI-software development life cycle. Furthermore, they are complicated for a domain expert with no data science and programming background, and are not designed to allow for collaboration between the key stakeholders (team members) involved in the development of the AI-powered software systems. In our future research, we will be looking at alternative approaches to provide domain experts with the tools they can directly use to cover the whole lifecycle of AI-based systems development.

ACKNOWLEDGEMENTS

Support for this research from ARC Discovery grant DP170101932 is gratefully acknowledged.

References

[1] I. Portugal, P. Alencar, D. Cowan, “A Preliminary Survey on Domain-Specific Languages for Machine Learning in Big Data”, IEEE International Conference on Software Science, Technology and Engineering (SWSTE), 2016.

[2] S. Landset, T. M. Khoshgoftaar, A. N. Richter, T. Hasanin, “A Survey of Open Source Tools for Machine Learning with Big Data in the Hadoop Ecosystem”, Journal of Big Data volume 2 issue 24, 2015.

[3] C. E. Sapp, “Preparing and Architecting for Machine Learning”, Gartner Technical Professional Advice, 2017.

[4] Microsoft Azure Machine Learning Studio, https://studio.azureml.net/

[5] Machine Learning at AWS - Amazon AWS, https://aws.amazon.com/machine-learning/

[6] Predictive Analytics - Cloud Machine Learning Engine | Google Cloud, https://cloud.google.com/products/machine-learning/

[7] BigML.com is Machine Learning Made Easy, https://bigml.com/

[8] D. D. Cock, “Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project”, Journal of Statistics Education, Volume 19, Number 3 (2011)

[9] D. Sculley, et al, “Hidden Technical Debt in Machine Learning Systems”, Advances in Neural Information Processing Systems 28 (NIPS), 2015

[10] Tableau Software: Business Intelligence and Analytics, https://www.tableau.com/

[11] Plotly: Modern Visualization for the Data Era, https://plot.ly/

[12] Trifacta: Data Wrangling Tools & Software, https://www.trifacta.com/

[13] Weka 3 - Data Mining with Open Source Machine Learning Software, https://www.cs.waikato.ac.nz/ml/weka/

[14] RapidMiner: Data Science Platform, https://rapidminer.com/

[15] Watson Machine Learning - Overview | IBM Cloud, https://www.ibm.com/cloud/machine-learning

[16] SAS: Analytics, Business Intelligence and Data Management, https://www.sas.com/en_au/home.html

[17] KNIME - Open for Innovation, https://www.knime.com/

[18] TensorPort · GitHub, https://github.com/tensorport

[19] Tensorflow, https://www.tensorflow.org/

[20] https://www.kdnuggets.com/2017/02/gartner-2017-mq-data-science-platforms-gainers-losers.html

[21] https://www.slideshare.net/bigml/the-past-present-and-future-of-machine-learning-apis

[22] R: The R Project for Statistical Computing, https://www.r-project.org/

[23] Orange – Data Mining Fruitful & Fun, https://orange.biolab.si/

[24] Scikit-learn: Machine Learning in Python, http://scikit-learn.org/stable/

[25] Apache Mahout, https://mahout.apache.org/

[26] Apache Spark™ - Lightning-Fast Cluster Computing, https://spark.apache.org/mllib/

[27] IBM SPSS - IBM Analytics, https://www.ibm.com/analytics/au/en/technology/spss/