Computational Data Science Research and Technology Development - State of the Art -*

Prof. Dr. V. David Sánchez A., Ph.D.
Brilliant Brains, Palo Alto, California

January 2023

Abstract

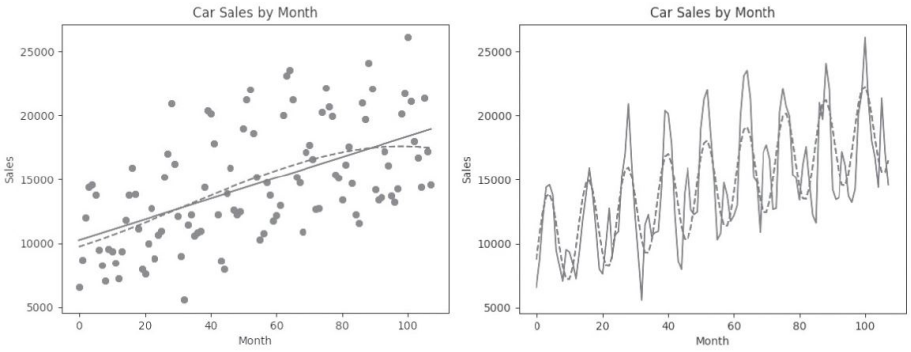
Computational data science is interdisciplinary and merges the knowledge and best practices of different fields including mathematics, statistics, computer and information science, data fusion, complex systems, human computer interaction, computer graphics, data visualization, data mining, database management systems, knowledge discovery, artificial intelligence (AI), machine learning (ML), computational intelligence (CI), high performance computing (HPC), parallel and distributed computational systems, big data, and business. Its main goal is to improve decision making through the analysis of data [1]. Statistics focuses on quantitative data whereas data science handles qualitative data on top. A computational data science project consists of the following stages: data preprocessing, defining the computational data science problem, designing and developing the data-driven solution to the problem, and reporting of findings for final decision making.

Let us briefly introduce basic concepts with a business praxis example [2]. Figure 1 shows two-dimensional charts with the month value as the abscissa and the amount of monthly car sales as the ordinate. The provided raw data appears as highly noisy data due to seasonal variations, time trends, and plain randomness. Figure 1(a) shows the continuous straight and the dashed cubic univariate regression lines determined based on the data provided whose scatterplot is also shown overlayed with the regression lines. These regression lines are intended to be used to forecast the future of monthly car sales. Both lines appear to pass through the middle of the data point cloud without capturing the variation of the data point values, more specifically their monthly cars sales, ordinate values. On the other hand, Figure 1(b) shows again the values provided for the data point cloud, but this time with point-to-point line segment connections which visualize the aforementioned variations and is shown as a continuous line. Shown as dashed line is also the trigonometric regression curve intended to be used to forecast the future of monthly car sales and possessing a lower RMSE (Root Mean Square Error) value than the straight and cubic univariate regression lines since it obviously better captures the variations of the data point cloud values.

Another example, actually a family of use cases includes programs/projects for the intertwined Research and Technology Development (R&TD) within the drug and medical device development process from strategic planning through product commercialization that need to address diverse topics including [3]:

- strategy, reimbursement, and proof of concept

*This abstract has been granted permission for public release. The author has been among several others the Chief Scientist, Editor-in-Chief of an Elsevier Science B.V. Artificial Intelligence (AI) / Machine Learning (ML) journal for 15 years providing advanced computational data science research and associated technology devlopment, e.g., as Chief Technology Officer (CTO) of Thuris Corp., a biotech company engaged in drug development and diagnosis of Central Nervous System (CNS) disorders.



(a) scatterplot of data point cloud + continuous straight linear and dashed cubic regression lines

(b) continuous line plot of data point cloud + dashed trigonometric regression curve

Figure 1: Basic business example – forecasting of monthly car sales [2]

- intellectual property, outcome measures, trials and indications
- quality and medical devices
- diagnostics, biomakers, and wearable devices
- software, target selection and drug discovery, and absorption, distribution and metabolism
- toxicity, formulation, and stability
- parental products and biologics

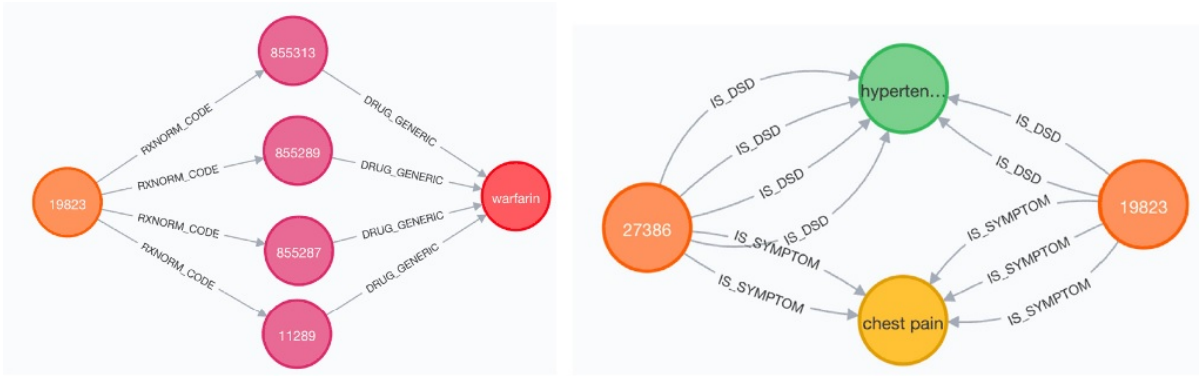
Programming languages and tools are multiple to build computational data science systems including Python and R to mention just two of them [5, 6]. For example, in the landscape of R, computational data scientists can use features to wrangle, i.e., transform datasets into a form convenient for data analysis; program, i.e., develop the data-driven solution to the problem at hand; explore, i.e., examine the data, generate and test hypotheses; model, i.e., provide a low-dimensional summary capturing true signals in the dataset; and communicate, i.e., use R markdown to integrate prose, code, and results. In the Python landscape, computational data scientists can start using the NumPy library to perform underlying computation, aggregration, indexing, and sorting of arrays: basic, structured, record-; the Panda library to define series, dataframe and index objects, data indexing and selection, operating on data, handling missing data, hierarchical indexing, combining datasets (concat & append, merge & join), aggregating and grouping data, using pivot tables, operating vectorized strings, working with time series, and high-performance operations with compound expressions (eval & query); the plotting library Matplotlib and its object-oriented API for visualizing data and results, embedding plots into applications using general-purpose GUI toolkits including TKinter, wxPython, Qt, and GTK, scatterplots, visualizing errors, density and contour plots, histograms, binnings, and density, multiple subplots, text and annotation, custom configurations and stylesheets, three-dimensional plotting, geographic data wirh basemap, and visualization with seaborn; the machine learning library scikit-learn to represent data and use its estimator API, tradeoff hyperparameters and perform model validation, feature engineering, apply different algorithms for classification, regression, support vector machines, decision trees, random forests, principal component analysis, manifold learning, k-means clustering, kernel density estimation; among others. W.r.t. tool integration, e.g., the scikit-learn library integrates well with libraries such as Matplotlib and plotly for plotting, NumPy for array vectorization, Pandas dataframes, SciPy, among others.

On-prem and cloud-based data science environments are currently being used for all aspects and in all stages of the data science lifecycle [4]. Those envionments allow for example real-time deployment of models in any of the popular programming languages along with monitoring and reporting of those models in production. They all have helped encourage the adoption of data science operationalizing it to produce results efficiently

(a) workflow for automating PHI removal

(b) graph schema to retrieve information

Figure 2: Patient cohort architecture based on NLP and KGs [7]



(a) graph -- relationships between prescription codes and medication

(b) graph -- relationships between patient records and medical symptoms

Figure 3: Graphs using NLP to represent relationships between a network of entities [7]

and generate more innovation and business value by assisting computational data scientists develop historical data based models and use them for model-based decisions showing their advantages compared to the use of only intuition and consensus for decision making. More advanced capabilities allow for models to be monitored to detect data drift and to be updated to reflect changes over time. An example in the health care & life sciences area shows the conjunction of the use of diverse techniques to elaborate computational data science solutions as is often the case, e.g., in patent analytics' cohort building using Natural Language Processing (NLP) and Knowledge Graphs (KGs). Figure 2 shows the patient cohort architecture [7]: the end-to-end workflow for automating Protected Health Information (PHI) removal from documents and images to the left (a), and the graph schema to retrieve information based on underlying relationships for querying to the right (b). Figure 3 shows graphs using NLP that represent established relationships between a network of entities. Highly innovative computational data science approaches and solutions are showcased for different business areas. The customization of algorithms for specific applications, required environment configuration, appropriate collaboration among team members in advanced, complex computational data science programs and projects is covered in depth as well as some new promising application domains.

References

[1] J.D. Kelleher and B. Tierney. Data Science, 2018.

[2] B. Tuckfield. Dive Into Data Science – Use Python to Tackle Your Toughest Business Challenges, 2023.

[3] MIT xPRO. Drug and Medical Device Development: A Strategic Approach, 2023.

[4] Domino Data Lab. The Data Science Innovator's Playbook, 2022.

[5] J. VanderPlas. Python Data Science – Essential Tools for working with Data, 2016.

[6] H. Wickham and G. Grolemund. R for Data Science – Import, Tidy, Transform, Visualize, and Model Data, 2017.

[7] Databricks. The Big Book of Data Science Use Cases, 2023.