Option #1: Education BI Use Case: Student Retention & Reducing Costs

Scott Miner

Colorado State University – Global Campus

**Option #1: Education BI Use Case: Student Retention & Reducing Costs**

The assignment asks the student to submit a business intelligence framework for a university with one online and two physical campuses. The framework must provide easy and secure real-time access to historical student data combined over the last 15 years from multiple university departments. Additionally, the framework must be capable of predicting student attrition rates, as well as presenting results via BI dashboards.

**BI Dashboards**

Eckerson (2006) describes a dashboard as a “full-fledged business information system that is built on a business intelligence and data integration infrastructure” (p. xiii) composed of three parts: (a) a *monitoring application* that displays critical information quickly via graphs and charts, (b) an *analysis application* that allows users the ability to drill-down into performance measures, and (c) a *management application* that empowers executives by giving them continuous feedback across a range of key metrics.

**Architectural Framework**

Muntean et al. (n.d.) describe how now, more than ever, universities need consistent and timely access to data to make more short-term decisions, plan for the long term, create opportunities for students, and retain students. University management is much like corporate management in that multiple divisions comprise a university as multiple business units comprise a corporation. Examples of divisions within universities include finance, research, human resources, student, and teaching and learning divisions. The architectural framework of a performance management solution encompasses several key components: (a) a data warehouse that serves as a single source of knowledge, (b) a reporting layer that supports transactional reporting and ad hoc queries, (c) an analytical layer that supports multidimensional analysis, (d) a monitoring layer with personalized dashboards and key performance indicators (KPIs), and (e) a University Web portal that brings all services needed by students and faculty into one location. Figure 1 displays this architectural framework for a performance management system for universities.

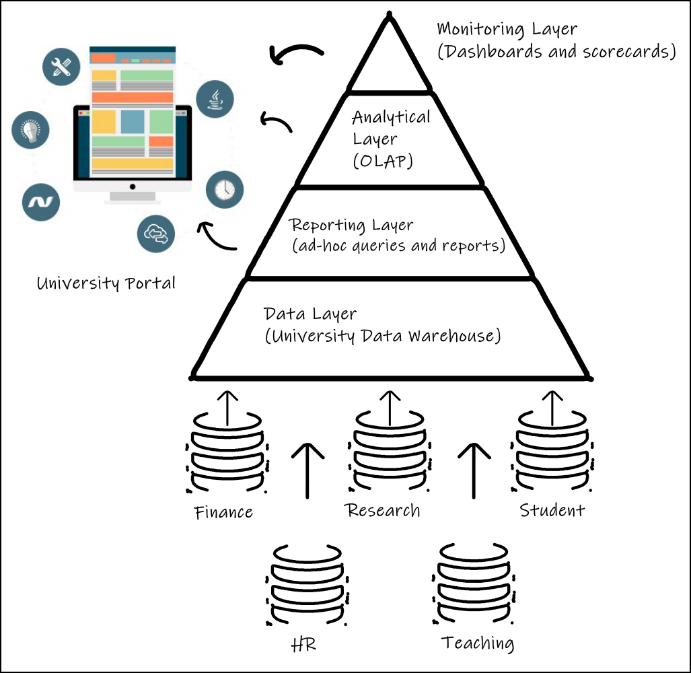


Figure 1. Data sources from different departments are consolidated in the DW layer, which provides the foundations for the reporting, analytical, and monitoring layers accessed by the University Portal

**Predictive Capabilities**

As mentioned, the framework provides an analytical layer capable of predicting students likely to drop out. Thammasiri et al. (2014) compared four popular classification methods, including logistic regression (LR), decision trees (DT), support vector machines (SVM), and artificial neural networks (ANN), along with three balancing techniques, including over-sampling, under-sampling and synthetic minority over-sampling (SMOTE) to determine which model had the best predictive accuracy when applied to the student attrition problem. More specifically, which model had the best predictive accuracy for the minority class (e.g., students likely to drop out), which is often the area of focus in classification problems. The SVM using the SMOTE balancing techniques achieved the best performance with 90.24% A screenshot of a social media post

Description automatically generatedpredictive accuracy on the minority class. The study also found the most important factors in predicting dropout rates, the most important of which was Fall GPA, followed by the number of credit hours earned compared to credit hours registered, followed by spring scholarship status. Figure 2 shows the SAS code to generate an SVM model to predict student attrition. The script begins by creating a connection to Cloud Analytic Services (CAS) (SAS Help Center: Example: Support Vector Machine, n.d.). Once we establish a connection and load the data into a CAS table, the PROC CAS procedure runs the SVM algorithm on the student data set. The table parameter names the data set to be analyzed. The nominals parameter lists nominal values along with the output variable. The inputs parameter lists all independent variables of all data types. Lastly, the target parameter specifies the dependent variable (e.g., “enrollment\_status”) (SAS Help Center: Support Vector Machine Action Set, n.d.). Because the SVMSMOTE ranked highest in predictive accuracy (0.902) and specificity (0.958), I recommend implementing this model to predict student attrition.

Figure 2. SAS code to generate an SVM model to predict student attrition

**Secure and Real-Time Data Access**

The system must furnish real-time responses to ad hoc analytical queries. With real-time data warehousing, the information in the DW is never more than a few seconds old. Real-time data warehousing empowers the university by enabling executives to take advantage of opportunities before competitors. Moreover, faculty can identify students more likely to withdrawal sooner and intervene more quickly to increase retention rates and reduce costs. Oliver (2013) describes the event-and-action data warehouse architecture, capable of near real-time data warehousing with data latency measured in seconds. For organizations reliant on legacy systems, the architecture provides four key benefits, namely (a) **timely BI** because the DW provides real-time information for key decisions, (b) **efficiency** because the architecture is easy to implement and understand, (c) **reduced risk** because no legacy application code changes are required, and (d) **performance** because there is minimal impact on operation performance. Event-and-action architecture is composed of three key components, (a) *Changed Data Capture (CDC),* (b) *a CDC Store*, and (c) *DW action*. Figure 3 illustrates the event-and-action architecture.

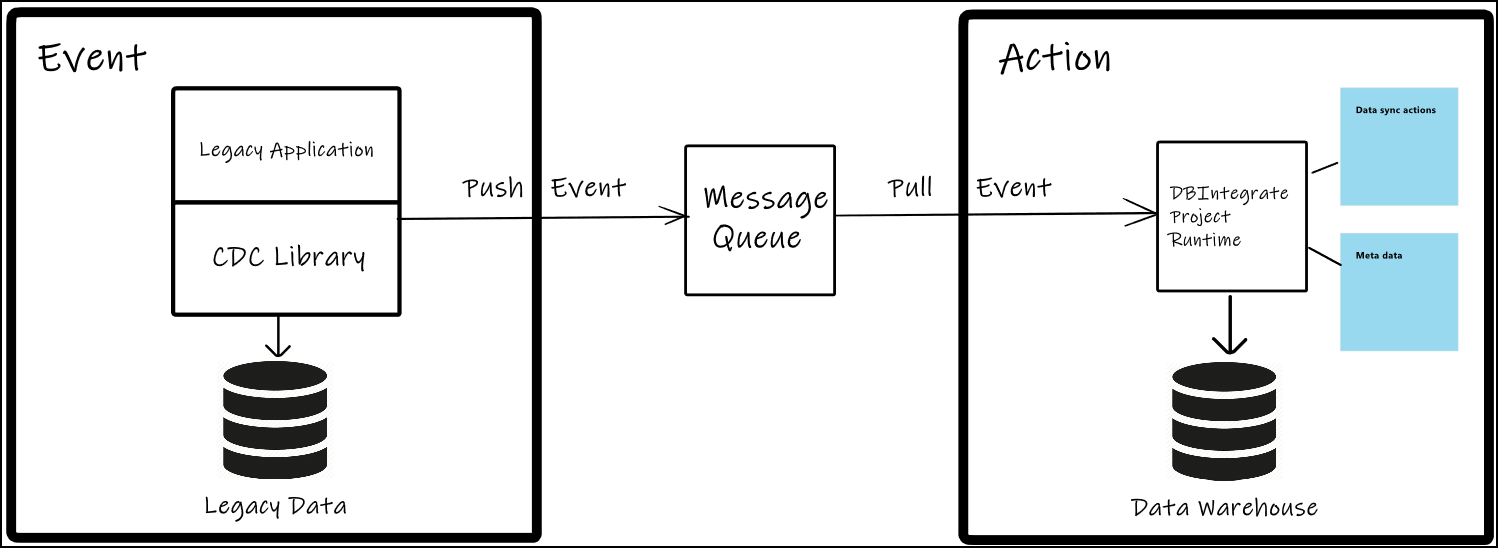
In Figure 3, we see a CDC library installed on a legacy system to intercept change events. The change events are loaded into a CDC store, represented by the message queue. Oliver (2013) indicates many legacy runtimes provide an API to accomplish such a task. The message queue preserves the sequencing of events, and as soon as they are written, a pull action from the framework grabs the event from the queue to update the DW. The whole process takes no longer than 5.221 seconds, allowing the system to achieve near real-time processing.

Figure 3. Event-and-action architecture, data sync actions combined with pull events update the DW in less than five seconds.

Further, the framework will provide for secure access to the data from multiple university divisions. Warigon (1998) describes a framework to enhance security composed of seven phases: (a) identifying data, (b) classifying data, (c) quantifying the value of data, (d) identifying data security vulnerabilities and their costs, (e) selecting cost-effective security measures, and (f) evaluating the effectiveness of security measures. Data can be classified into three categories: (a) public data, (b) confidential or moderately sensitive data, and (c) top secret or most sensitive data. With an access control policy based on “least privilege,” end-users and faculty staff only have access to the data for which they have legitimate privileges. Such an architecture enhances data security.

**Conclusion**

Stocker (2012) emphasizes that the education sector is under constant pressure to meet student retention targets. The framework described in this paper uses a real-time DW architecture that combines data sets across university departments to analyze historical data from the past 15 years to predict students likely to drop out using SVM and BI dashboards that allow for secure, real-time data access. Such an architecture will allow the university to assess patterns for their students and do intervention programs more effectively based on up-to-date information (Hildreth, 2012). SAS (2011) demonstrates the benefit achieved by the University of Texas, who implemented a similar architecture to provide the most cost-effective means to produce graduates and provide quality education. The system decreased IT staff burden by providing simple Web access, all while enduring data integrity and streamlining processes. For these reasons, my framework proposes similar benefits.

**Part B: SAS Code – Conditional Situations**

The second part of the assignment asks the student to read records from a file named ‘EXAMPLE.DAT’ that contains 50 observations across nine attributes. The directions instruct the student to create a new data set named MYDATA and to populate the data set with five of the nine attributes (e.g., ID, GP, AGE, TIME1, and TIME2) from the ‘EXAMPLE.DAT’ file. The student is to only read certain records from the data file, including only individuals who are ten years of age or younger. To subset the data, we make use of a subsetting IF statement in the DATA step. The subsetting IF statement populates the MYDATA data set with the corresponding attributes from the ‘EXAMPLE.DAT’ file (Elliott & Woodward, 2016). Figure 4 shows the SAS code used to read in the file. By examining output either from the results viewer, the output tab, or the log file, we can verify we subsetted the data correctly. Figure 5 displays the output from the results viewer after we run the PROC PRINT procedure to display the records to the viewer in HTML format. The MYDATA data set contains 22 of the 50 records from the original file. We subsetted the data according to the requirements. We can verify this by viewing and confirming the ages of the 22 subjects, output via the PRINT procedure in Figure 5.



Figure 5. Contents of the WORK.MYDATA dataset

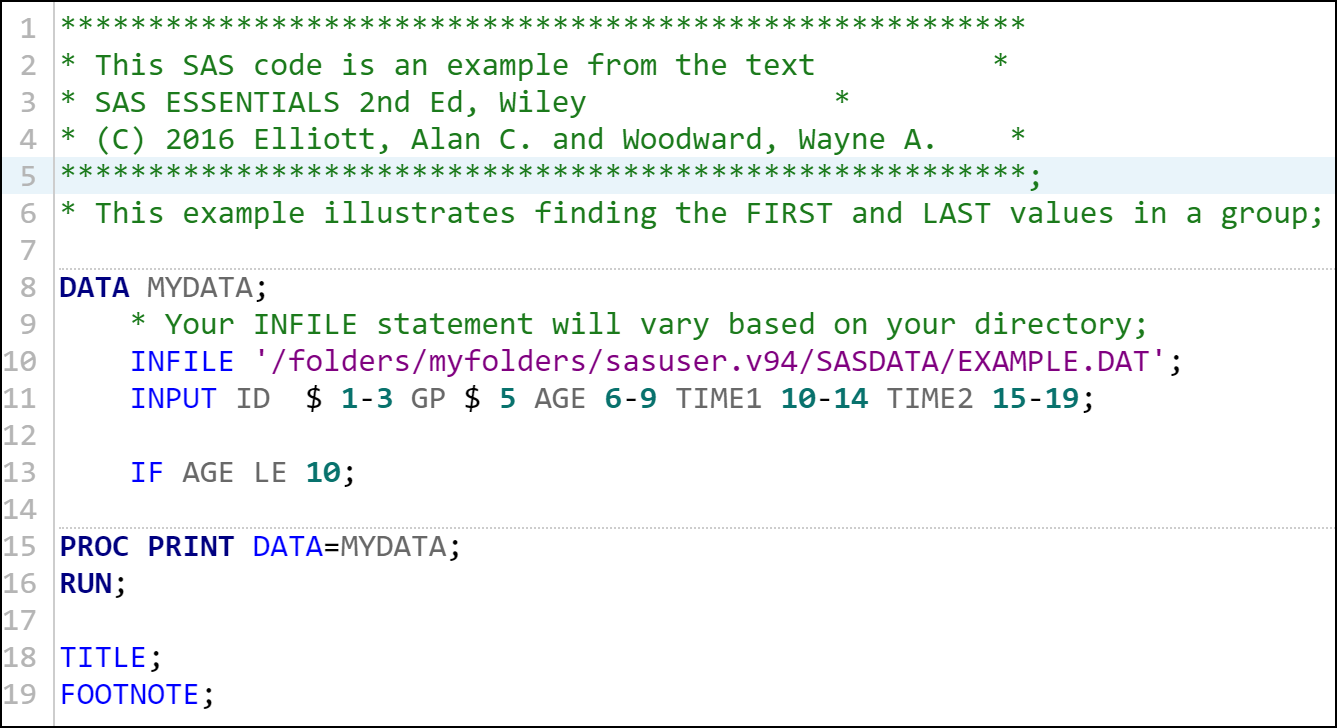


Figure 4. SAS Code to read records from 'EXAMPLE.DAT' input file

References

Eckerson, W. W. (2006). Performance Dashboards: Measuring, Monitoring, and Managing Your Business. Wiley. http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=142545&site=ehost-live

Elliott, A. C., & Woodward, W. A. (2016). SAS essentials: Mastering SAS for data analytics (Second edition). John Wiley and Sons, Inc.

Hildreth, S. (2012). Minnesota DEPARTMENT OF EDUCATION. Computerworld, 46(15), 21.

Muntean, M., Sabau, G., Bologa, A.-R., Surcel, T., & Florea, A. (n.d.). Performance Dashboards for Universities. 7.

Oliver, F. (2013). Real-Time Data Warehousing for Legacy Applications. Business Intelligence Journal, 18(3), 29–35.

SAS. (2011). SAS, University of Texas System Launch Public Dashboard to Share Progress, Performance. Business Wire (English). http://search.ebscohost.com/login.aspx?direct=true&db=bwh&AN=bizwire.c38645533&site=ehost-live

SAS Help Center: Example: Support Vector Machine. (n.d.). Retrieved January 19, 2020, from <https://documentation.sas.com/?docsetId=webeditorref&docsetTarget=n1r3xo1gzo2tfyn1x7re11rswt7w.htm&docsetVersion=3.71&locale=en>

SAS Help Center: Support Vector Machine Action Set. (n.d.). Retrieved January 19, 2020, from <https://documentation.sas.com/?docsetId=casactml&docsetTarget=casactml_svm_example01.htm&docsetVersion=8.2&locale=en>

Stocker, R. (2012). The Role of Business Intelligence Dashboards in Higher Education. Credit Control, 33(1), 37–42.

Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. Expert Systems with Applications, 41(2), 321–330. <https://doi.org/10.1016/j.eswa.2013.07.046>

Warigon, S. (1998). Data warehouse control & security. Internal Auditor, 55(1), 54.