JST BI Blitz

Advanced Spreadsheet Automation Final Project

Sean Fisher, Tahna Black, Joe Burwasser - April 17, 2014

Executive Summary

The Brigham Young University Information Systems program currently provides students taking their introductory courses with opportunities to explore business intelligence. Unfortunately, this capability is only provided through Microsoft SQL Server and students must either install this software on their personal computers or use the school's desktops, which are limited in quantity. We have provided a solution to this situation: enabling students to utilize the basic business intelligence functions by including them in a downloadable workbook called JST BI Blitz.

The objective of this project is to provide business intelligence algorithms for free to students in a form doesn't require Microsoft SQL Server.
These algorithms include:

- K-Nearest Neighbors(KNN)
- Naive Bayes (NB)

The workbook displays the results of these algorithms in an intuitive manner which can be easily analyzed.

Aside from the algorithms, this workbook aims on providing other Business Intelligence-related capabilities to users. A customized ribbon includes the following options:

- Cleaning the data before it's analyzed (removing rows with blank values)
- Partitioning the data into a training and a validation set
- Tutorial for effectively utilizing the workbook

With the Blitz workbook, students are able to gain experience with statistical analysis in a convenient manner on their own laptops. Students can learn to build BI models without any additional software or financial strain.

Implementation Documentation

At the start of the project, our team brainstormed capabilities to be included in the solution. The tasks and sub procedures were delegated between the three team members. The components of our solution are described below, including their role in the overall task. The components include the following:

- Functions
- Data Cleaning
- Partitioning
- Data Mining Algorithms
  - K-Nearest Neighbors
  - Naive Bayes
- Features
- User Forms
- Storage Medium
- User Ribbon

After the implementation documentation, we have provided a full-fledged tutorial to walk students through the process of using JST BI Blitz. This tutorial is provided separately.

Functions

The modules and sub procedures were all coded in native VBA. We focused on providing efficient, simple methods for performing the algorithms. When we first conceptualized the program, we broke it up into different parts for each of the capabilities, as discussed below.

Data Cleaning

One of the areas of focus was to make sure that the model and algorithms implemented would be able to be run on the data which was provided. We developed a sub procedure for cleaning the data which removes all records from the dataset which have missing values. In real-world data mining, there are several options for recovering from missing values, including placing a default value in the cell or an average of the rest of the values in the column. Because this is a simplified version of a business intelligence tool, we opted to simply remove the missing values, which should have minimal impact on the educational value of the tool. We anticipate that the tool will be used with the provided data sets or other prepared data, making full-fledged data preparation features unnecessary.

Partitioning

As we learned more about constructing effective business intelligence models, it became apparent that at minimum two partitions are necessary for verifying model accuracy. We decided to provide functionality for the user to be able to partition the data and coded a separate function from the cleaning module. The code uses the random number generator to separate the data into two groups (partitions) which are displayed in separate worksheets. This function allows students to better grasp the concept of partitioning by providing a visual representation of what it does.

Data Mining Algorithms

We decided to implement two data mining algorithms: k-Nearest Neighbors (kNN) and Naive Bayes (NB). k-Nearest Neighbors provides numeric prediction capabilities, while Naive Bayes provides binary classification capabilities. Details on each algorithm and how they work are provided below.

K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm is intended for use when you have all numeric columns and a numeric output variable. For example, if you were trying to predict somebody's BMI from their weight, pounds of food consumed per day, and miles they walked every day, kNN would be perfect. Each input variable and the output variable are numeric.

The algorithm works by calculating the Euclidean distance between the given record and every other record in the data set. The Euclidean distance algorithm we have provided automatically normalizes (standardizes) the data given by subtracting the mean and dividing by the standard deviation. Then it averages the output variable for the closest *k* records to provide a predicted value for the target record. You can experiment with the quantity of *k* needed to give you good prediction: sometimes it is 1, sometimes it is 15. We built into the program a way to automatically determine the best *k* value, which you can then use to predict new data. kNN gives three measures of how well it works: Root Mean Squared Error (RMSE), Total Sum of Squared Errors (TSSE), and Average Error (AvgError). RMSE is the square root of TSSE, and TSSE is the sum of all of the difference between predicted and actual values that kNN gives by running each validation record on the training set. Average error is simply the average of the difference between predicted and actual values. RMSE is typically the measure that is most useful to judge on.

Naive Bayes (NB)

Naive Bayes is an algorithm which calculates probabilities for specified outcome variables with given input variables. It is meant to be used with categorical input and output variables. For example, if you were trying to predict whether a flight will be delayed or not (delayed = 1, not delayed = 0) based on the size category of the airport, the type of peanuts served on the plane, and the size class of the airplane, NB would be ideal. The output variable should be either a 1 or a 0, but the input variables can be various categories.

Naive Bayes uses a generated table of probabilities to predict new records. We generate these probabilities by counting the number of occurrences in the data set for a particular category, then calculating the percentage that the particular category appears for both 1 and 0 outcomes. We then use the Naive Bayes formula to calculate a probability that an incoming record will be classified as a 1. If the probability is over a cutoff value (typically .5), we classify the incoming record as a one. After the Naive Bayes algorithm has been trained, students may then take another dataset (with the same input variables) and predict outcomes using the stored calculations from the trained dataset.

Features

We consider user experience to be one of the top priorities for this project, so providing an intuitive and simple interface for users was imperative. Students must be able to know how to manipulate the program's capabilities to achieve their needs and learn about the data mining process.

User Forms

Users are able to input their specifications, including data, through the forms which have been constructed. Our goal was to create forms that would teach the audience about the information being inputed and how it would contribute to the algorithm. Along with input data, each form contains a "Help" button to guide the user in using the form. These forms were constructed in VBA, and the functions of each form are explained in detail (along with screen shots) in our tutorial.

Storage Medium

We originally considered providing the program as an add-in to an open excel workbook, but we decided on making a unique workbook which could be downloaded and utilized without any pre configuration besides macro enablement. Users have the capability of using the provided datasets that come with the workbook, or they can import other data sets into the workbook. How the datasets are tested with our algorithms are explained in detail in our tutorial.

User Ribbon

To achieve ease of use, we decided that it would be most beneficial to break up all of the capabilities of the program into individual parts. We achieved this by creating a separate button for each process on the Excel ribbon for the user to choose. Each process, including how and when to render the capability, is explained in detail in our tutorial.

Discussion of Learning and Conceptual Difficulties Encountered

The most intimidating and difficult task of the project was deciding on what external resources to use for actually implementing the logic needed for the business intelligence models. After consulting with an external data mining expert, we attempted to implement the Accord.net framework. Accord.net is an open source .NET framework used for statistical applications and a variety of other functions as well including machine learning and signal processing. After a total of almost 10 hours spent on trying to implement the framework and get it working in the desired fashion, our team concluded that it would be more efficient and beneficial to develop the program logic through VBA. The primary problems arose in using C# to program the DLL needed to run the algorithm, then register the DLL through the COM service and distribute the necessary DLLs with the workbook, then register them with COM on each individual computer. This was deemed as infeasible.

We also encountered some difficulty with getting the data-cleaning function to work. Originally we coded the function to remove records if any cell in that row was blank. This lead to the discovery that some workbooks are formatted to make cells with the value 0 to appear as blank cells. The code was modified accordingly to account for this.

Another task which was difficult was developing the logic for the Business Intelligence Algorithm(s) which we provided for the users. While each team member had obtained experience using these models and was able to understand them conceptually, no one had previously tried to code these model(s**)**. The best remedy for this was the ability to conceptually discuss and review our progress with the program logic. The ability to bounce ideas of other team members and have different perspectives on approaching problems helped us to discover errors and overcome obstacles which arose.

Assistance

Upon establishing the project objectives and goals, we started communicating with two renowned BI specialists, Dr. Dean and Mr. Anderson. Douglas Dean is a professor at Brigham Young University who currently teaches a course on Business Intelligence. Dr. Dean assisted us in identifying other pre-existing BI models and steering us in the direction where we could make a valuable contribution. He also put us in contact with Russ Anderson. Russ Anderson introduced us to the Accord.net framework and recommended certain algorithms for us to focus on. While we didn’t end up using the framework, the exposure to it and the knowledge gained about the algorithms was invaluable.

We also received assistance from BYU Professor Gove Allen, who took on the role of project sponsor for our team. Professor Allen assisted in gaining the support of other external resources, including Dr. Douglas Dean (mentioned above). Professor Allen also helped to narrow the project scope so that the finished product would be both attainable and effective.