# JST BI Blitz

## Tutorial

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

# Tutorial

JST BI Blitz is designed to help users get accustomed to manipulating and interpreting business intelligence models. In this tutorial you will learn how to operate the workbook in the following ways:

1. Importing and cleaning datasets
2. Partitioning Datasets
3. Training KNN on the partitions
4. Scoring KNN with other datasets
5. Training NB on the partitions
6. Scoring NB with other datasets
7. Analyzing Results

The tutorial will present these procedures in the order listed above. There are a couple items to note before getting started:

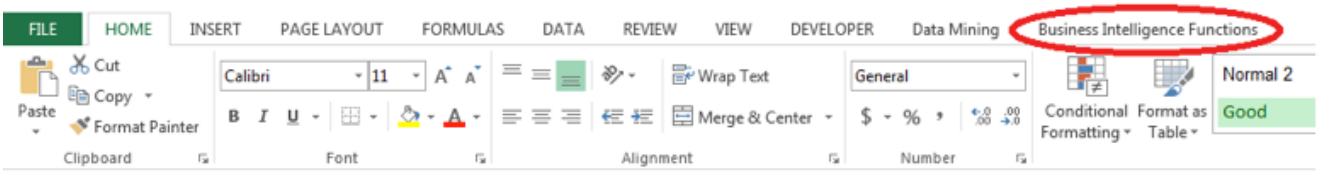
- Sample data is included within this workbook for you to practice on:
  - The "Flight Delays" dataset can be used for Naive Bayes model construction and scoring.
  - The "Boston Housing" dataset is to be used with kNN model construction and scoring.
- The "Help" buttons in the user forms (pop-up windows) contain tips and advice for effective use of the Blitz's functions.
- Make sure that you have cleaned and partitioned the data before training or using the models (instructions on how to do this are detailed in the next section).
- Before partitioning a dataset, make sure that you are on the page which contains the data to be partitioned.
- Before attempting to use the algorithms to score additional data, make sure that you are on the page which contains the additional data

## 1. Importing and Cleaning Datasets

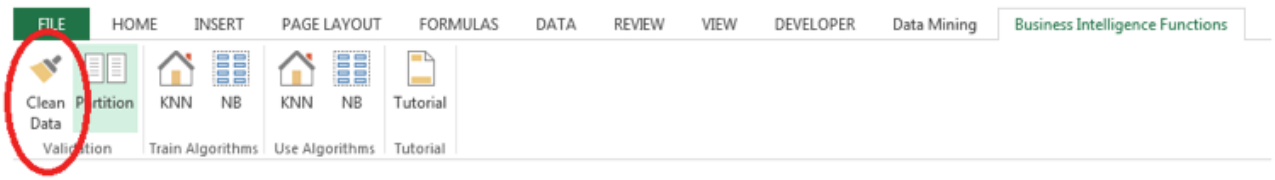
In order to perform statistical analysis on the datasets, it is important to make sure that the data to be used is in the right format. To import the dataset you want to test on, simply copy all of the values and paste them into the sheet labeled "Data" in this workbook. Step-by-step instructions to copy the data are below:

1. Switch to the workbook that has your data in it
2. Click somewhere inside the page where the desired data lies
3. Press the Control(Ctrl) (Cmd for Macs) button and the "A" button simultaneously to select all of the data
4. Press Ctrl/Cmd and "C" simultaneously or right-click on the highlighted data and click "Copy" to copy the data
5. Switch back to the Blitz workbook and select cell "A1" by clicking on it
6. Once cell A1 is selected, press Ctrl/Cmd and "V" or right click on cell "A1" and select "Paste"
7. Your data should now be in the sheet

Now you have your data where you want it, but in order for the algorithms to work correctly on the data, you must make sure that there is not any missing data which could throw off the calculations. To take care of this, you need to clean the data. In the upper-right hand corner of the screen in excel, you'll see a tab called "Business Intelligence Functions". Select that tab.



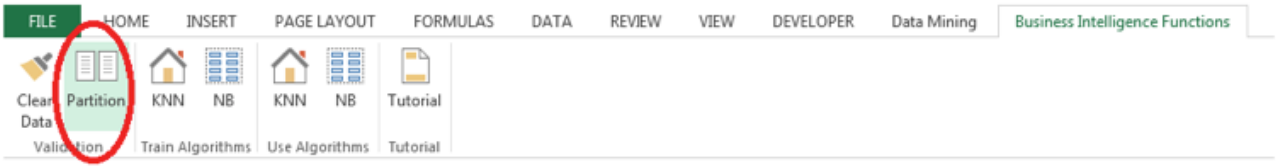

A new row of buttons appears on the ribbon in the left-hand corner. Select "Clean Data" to remove any rows, or records in the data which contain blank values. The workbook will create a new page called "Cleaned Data" Now your data is ready to be partitioned!

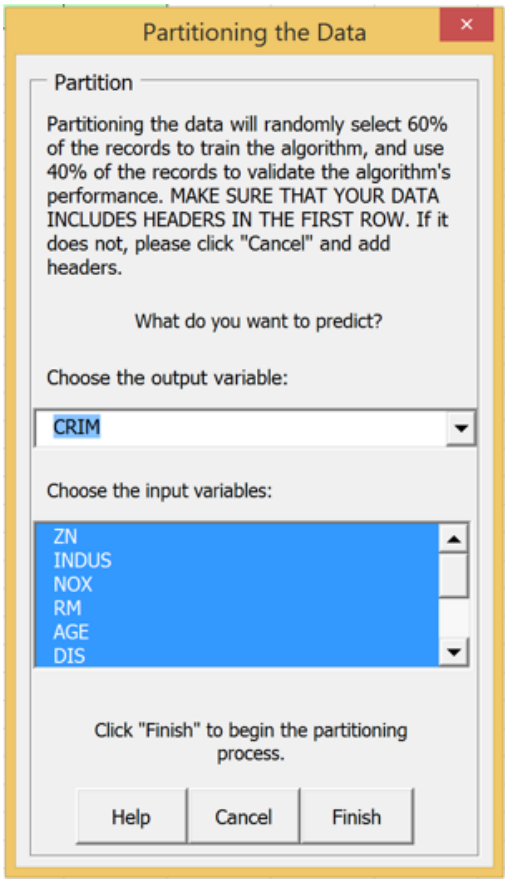


## 2. Partitioning DataSets

Business Intelligence models require at least two portions of data to be built upon. These are called partitions. In this workbook, the dataset will be randomly broken up into two partitions; one for training the model and another for validating the model.

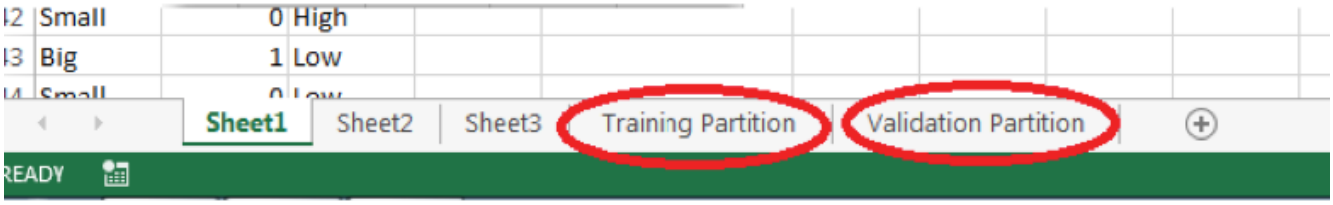
1. Select the "Business Intelligence Functions" tab in the upper-right hand corner of the screen if it isn't already selected.
2. Select the page containing the data that you want to partition (This should be the "Cleaned Data" page.
3. Select the "Partition" Button on the Ribbon




A form will pop up on the screen, asking you to first choose an output variable. Select an output variable. The output variable is the variable that you eventually would like to predict based on the input variables. The remaining variables will appear in a list at the bottom of the form and all selected by default. In other words, the program assumes that you want to include all of the variables. Click "Finish" when you have selected the appropriate variables.

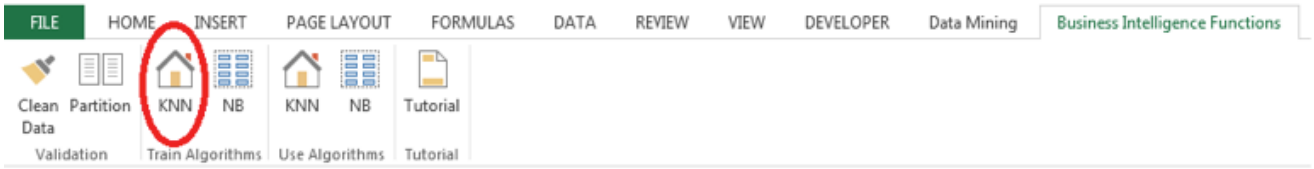
There are now two more worksheets. One is called "Training Partition" and the other one is called "Validation Partition". If you click on either of these pages, you will see a portion of the data in each. The training partition has approximately 60% of the data while the validation partition has the remainder(roughly 40%).




At this point, you are ready to build a BI model with the partitions!

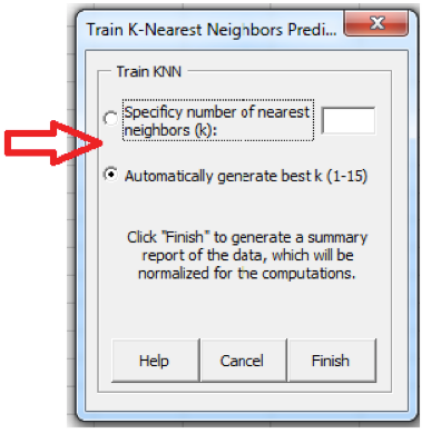
### 3. Training KNN on the Partitions

k-Nearest Neighbors is a model which makes predictions on a record based on its "neighbors", or records with similar attributes. On the "Business Intelligence Functions" tab, click on the "KNN" button in the "Train Algorithms" group.




PLEASE NOTE: kNN uses mathematics to build a model and is only able to work on variables which have **strictly numerical values**. If the dataset contains variables with text values (words such as "high" or "low"), the model will not run.

A form will appear giving you two options: 1) Specify a certain k to build the model on, or 2) Have the model find the best k between 1 and 15. If you choose the first option, specify a value for k that is between 1 and 15 and is an integer (whole number).




PLEASE NOTE: kNN must normalize all values to be able to compare them to each other, so each row takes a significant amount of time to compute, so building the model will take some time. In addition, the larger the range of k's selected, the longer it will take for the program to calculate performance for each k.

Choose your desired option and click "Finish".

A progress bar will appear.

When this is finished, a new "KNN_Results" sheet will be present (this will be "KNN_Results1", "KNN_Results2", etc. if you have previously run the algorithms). This sheet shows several different numbers which are relevant to the model you have just created. You'll see either statistics for the *k* you selected or statistics for the *k* we were able to determine was the best**.** The "best" k represents the optimal amount of closest records with similar attributes to use for predicting a particular record. For example, if k=5, then the five "closest" records to it on a graph would be used by the model to make a prediction.

You will also see three respective variable names and values below them.

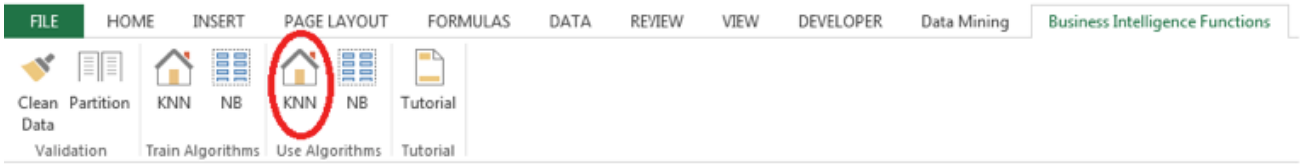
- RMSE - Root Mean Squared Error
- TSSE -Total Sum of Squared Errors
- AvgError - Average Error

In general, the lower to numbers you get, the stronger and more reliable the model is. Keep in mind, however, that large numbers don't necessarily mean that the model is weak or inaccurate. kNN error rates will naturally be higher in larger datasets or in data which contains large numerical values. RMSE is usually the number to use to compare different model performances.

## 4. Scoring KNN with other data

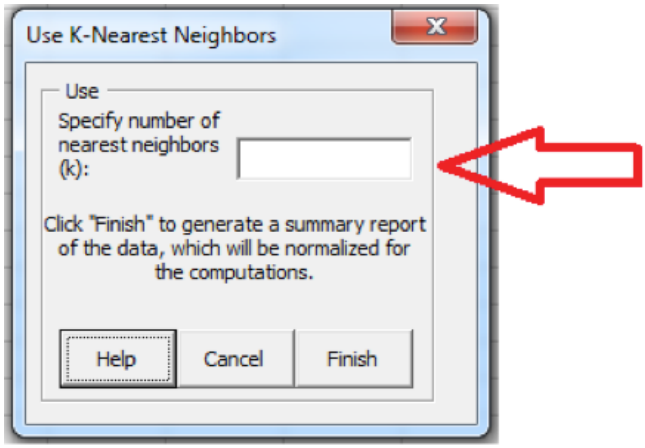
Now that you have made a kNN model, you are able to use it on new data! Please select data from another workbook or data source of your choice and paste it into any empty tab beginning at cell A1.

Then click the "KNN" button in the "Use Algorithms" group on the ribbon




A form will appear asking you to specify a certain k to build the model on. Just like the KNN training algorithm, make sure you specify a value for k that is between 1 and 15 and is an integer (whole number). Choose your desired option and click "Finish". If you ran the "automatically determine kNN" function from before, enter the best k in this box.

A progress bar will show the program's status as the new data is run through the model. when this is done a new page called KNN_Score will be present**.** This page contains all of the original variables with a new column called "Outcome" added on the left side of the data. The outcome column takes the values of that row (record) and compares them with the nearest k neighbors from the Training Data sheet to predict the value of the specified outcome variable.



## 5. Training NB on the Partitions

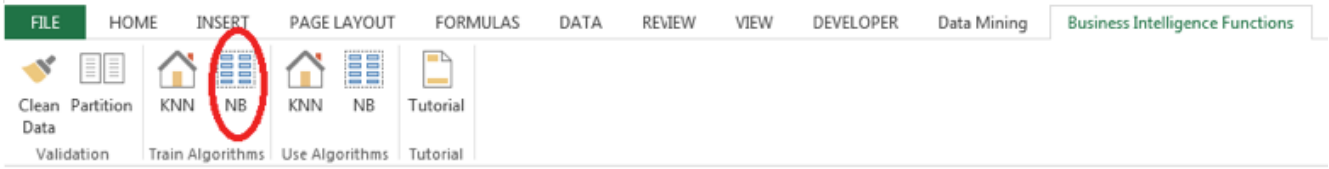
Naive Bayes is another business intelligence algorithm which calculates probabilities from categorical data in a dataset and uses these probabilities for future prediction on records with the same variables.

Thus, this algorithm is two-fold. The first step is to "train" your data by calculating the probability of occurrence for each of your input variables. An important thing to remember is that your output variable must have values of "0" or "1". ("0" means the input variable doesn't occur for a record in your data set, and "1" means the given variable does occur for a record in your dataset.) The probabilities that are calculated will be used in the formulas for scoring Naive Bayes with other data (section 6).

PLEASE NOTE:

1. Naive Bayes needs records with **categorical** and not numeric values.
2. Make sure your data is **cleaned** the correct way for this algorithm.
3. Make sure that you have **partitioned** the dataset which you are about to run the model on.

To start, select "NB" from the "Train Algorithms" group on the ribbon. It does not matter which sheet is active when you click the button.




A progress bar appears, but it may disappear quickly as this algorithm is able to run much faster. A new sheet with the name "NB_Training" or "NB_Training1" (depends on the number of times you have run this algorithm previously) has been generated and contains the probability of the output variable being true ("1") and the probability of the output

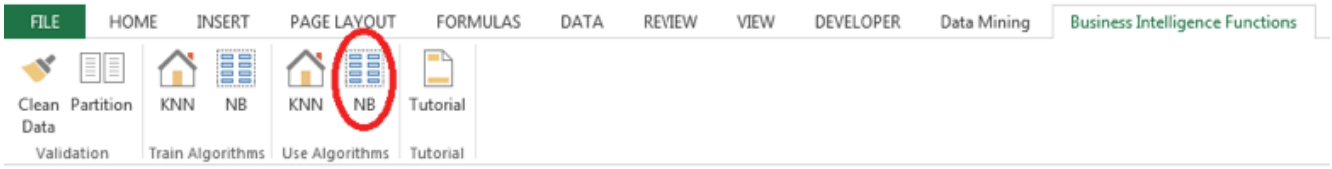
| Prob of 1 | 0.487179487 | Prob of 0 | 0.512820513 |
|---|---|---|---|
| | | | |
| Input Variables | Categories | Probability of 1 | Probability of 0 |
| Size | Small | 0.461538462 | 0.41025641 |
| | Big | 0.526315789 | 0.6 |
| Income | Low | 0.641025641 | 0.487179487 |
| | High | 0.342105263 | 0.525 |

variable being false ("0"). This information is shown on the first row of the table. Also included are specific probabilities for each of the data's input variables. High probability values indicate that when that particular variable value is present in a record, there is greater likelihood that the outcome will be a "1".

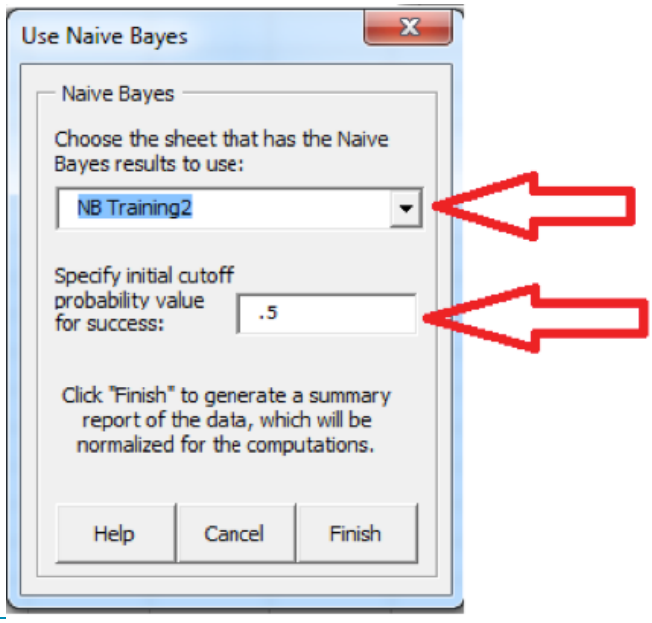
### 6. Scoring NB with other data

Once the partitioning of the data and the NB training have been done, you may now import and use another dataset to "score" it on the NB model. That is, the model will take the new data and calculate probabilities based on the probabilities calculated when the NB Training algorithm was run.

**Important:** Make sure you are on the sheet with your new imported data that you want to score on (i.e. make sure that sheet is active). Then, click the "NB" Button in the "Use Algorithms" Group.




A form will come up asking you to specify the sheet that has the probability calculations from running NB training. Select the sheet that was generated in the previous procedure (NB Training) from the drop-down list and specify a cutoff value. For each input variable, the probability score will be compared to the specified cutoff value. If it is below the cutoff value, the output is "0", and if it is above, the output is "1". The cutoff value must be a number **between 0 and 1**. The default value is .5, which you may choose to use. Click "Finish".




A progress bar will appear and when it is finished, and another "NB_Score" page will have been generated. On this page are the rows and

columns of the new data, with an additional column named “Outcome” on the left-hand side. The NB model has predicted positive outcomes as 1’s (outcome is true) and negative outcomes as 0’s (outcome is false). In this sense, you can see a record, its respective values in that row, and the model’s prediction on the outcome variable. This makes it easier to compare records with different variable values against each other.