Thomas Mathew

Khanh Ngo

Yelp Data Final Report: Find False Positive Elites

**Note:** The proposed project that was uploaded to Github was unfortunately didn’t work out for us so we decided to take another route and changed the problem statement. We went over the dataset that was given by Yelp and our original problem needed more information to be mined in order to accomplish our task. Our new problem that we are solving is:

Given a dataset of users, determine who should be candidates to become an elite user.

**Motivation and objectives:**

In order for Yelp to give users reliable and trustworthy reviews and ratings, Yelp has created a category of users known as Elite users. Now you might be asking yourself, what it takes to become an elite Yelper. Well there are three basic qualities that Yelp is looking for in Yelpers in order to tag them as Elite, and they are Authenticity, Contribution and Connection. For authenticity, you would have to provide a user profile picture for your account. This gives other users that you stand by what you say. Contributing to the community is measured by the variety of different businesses, numbers of reviews, and user’s helpfulness for other users. Lastly, connection is calculated by the number of friends, votes such as useful, funny, cool, and overall interaction within the community. At the end of the year, “as we get close to December, [yelp will] ask that you re-nominate yourself. The Elite Council spends many a sleepless night with pizza, beer, and 5-Hour Energy shots to pore over individual profiles and figure out who [deserves](http://www.yelp.com/tos/elite_en_us_20131011) another coveted term in office.”(http://www.yelp.com/elite) Now the main problem for Yelp is that according to their website, every year they scour every Yelp profile to determine whether a Yelper is Elite or not. Our team found this to be inefficient and time consuming, so we decided to make it easier for Yelp and narrow down the candidates for becoming Elite. So instead of looking at the entire population, Yelp will only have to look at a select few that could potential be an elite.

The objective is to use various data mining techniques and compare the results with one another. By comparing the results between each method, we will choose the one that outputs the best results. The data mining technique out team will use is through classification of groups. Our team will use a Decision Tree classifier, Naïve Bayes classifier, and K- nearest neighbor (KNN) as different methods.

**Data Mining/Analysis Tasks Tackled:**

The first task that was tackled was to convert the Yelp’s JSON dataset file into a CSV file that we could work with. Luckily, there was code online on another Github user’s account that was also doing the Yelp Dataset Challenge. The following link is what we used to convert the original dataset into data that we could work with: <https://gist.github.com/paulgb/5265767>.

We modified Paul’s code and added in an additional statement for the case of not being either a dictionary or list data type. Also, instead taking the original data and outputting it into the a CSV file, our team took the count of each dictionary or list values and outputted a number instead.

The second task was to familiarize ourselves with a data mining tool called Rapidminer. Rapidminer already has all the methods that we need to implement so we decided to recycle their implementation instead of creating our own. Understanding and using Rapidminer was a huge learning curve but as soon as we understood what we had to do, the program was easy to interact with.

For our third task, we had to compare the various methods and decide which method we would like to present to the Yelp’s deciders.

Lastly, we created a website to present our problem, solution/idea, results, tools and team members. The web development tool that we used to help us create a website is Bootstrap.

**Design of Methods:**

The following is the main program layout:

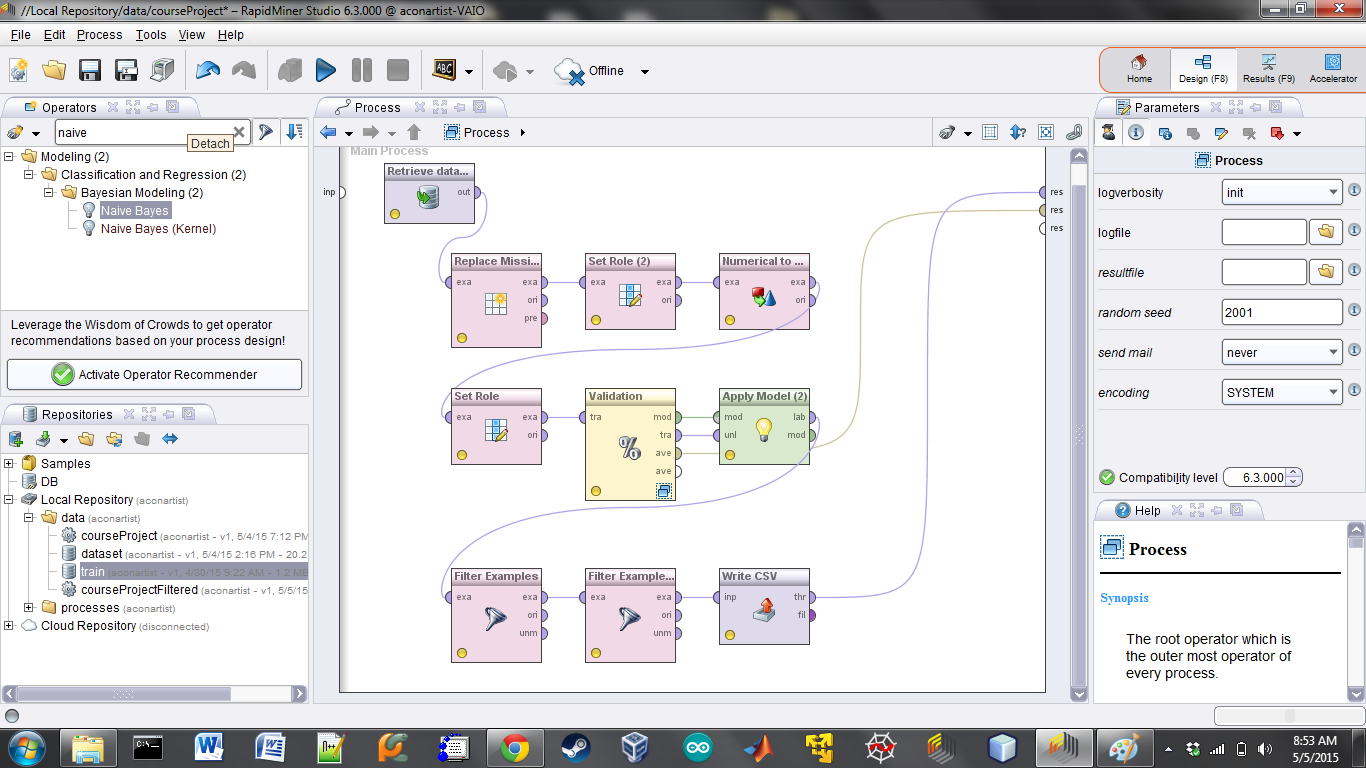


Figure 1: Full diagram process of classification.

The first box is the “Retrieve dataset” which is the dataset that will be passed on to the next element in the list. Following the first box is the “Replace Missing Values” box. This operator switches the missing values in of the selected attributes from the dataset by a specified replacement that we have chosen. The reason for this is to make sure that we do not have any missing values that could change the outcome of the results. The next operator is the set role which targets the specific attribute that we would like to work with next also with the other attributes that will not be affected in the next operator. The fourth operator is going to take the targeted attribute from the past operator and switch the numerical values to binomial values. We have selected true and false as the two values for this new binomial attribute. Everything is then passed to another set role. This set role operator is going to select the “elite” attribute as the label to work on by the next operator. The sixth operator is the validation process. This process contains subparts displayed in the following pictures:

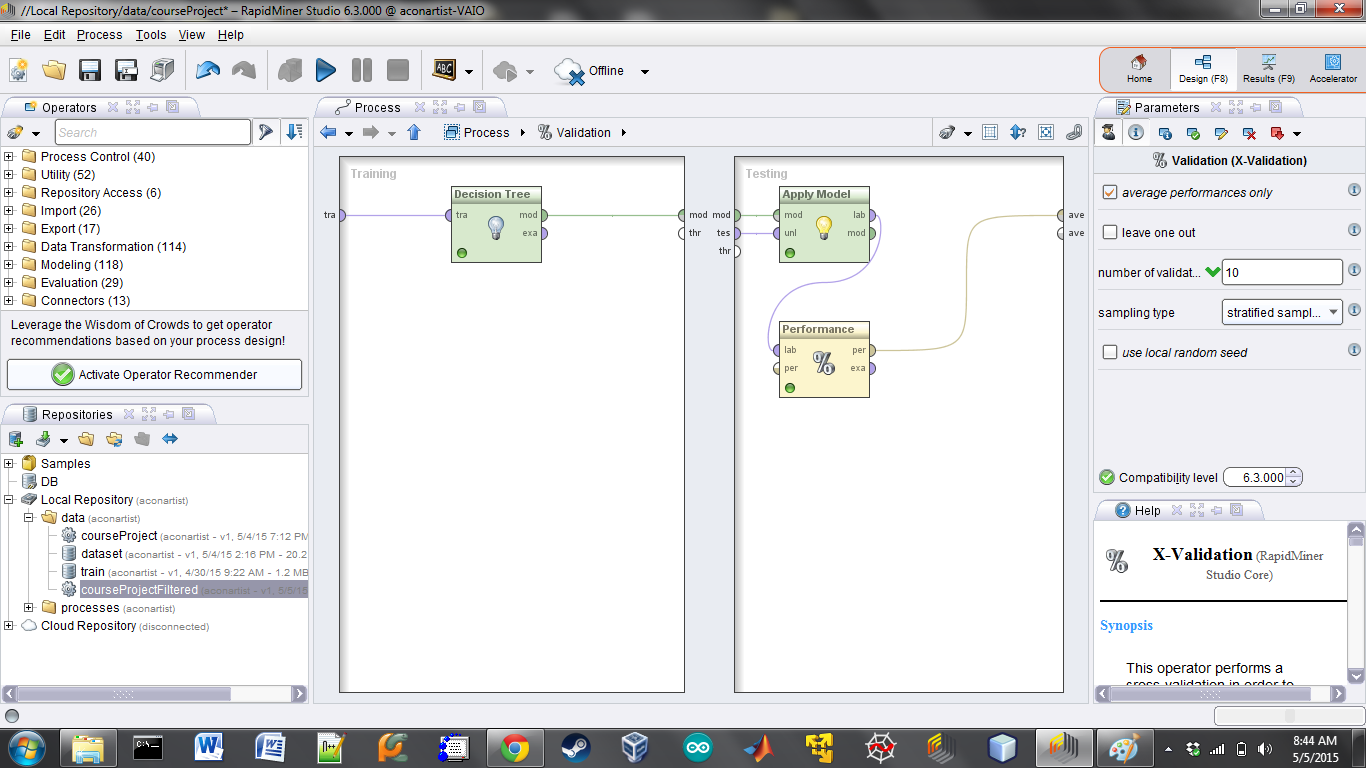


Figure 2: Validation with decision Tree classifier including performance measure.

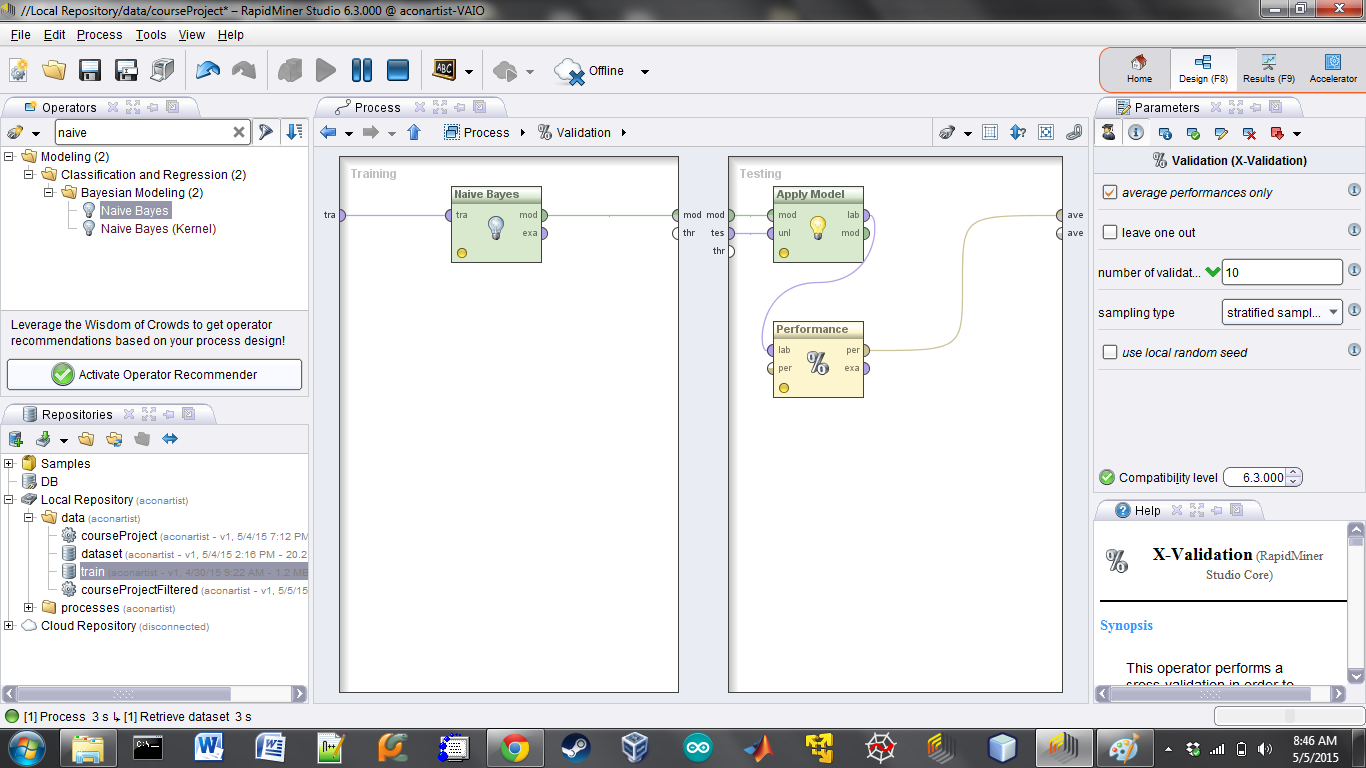


Figure 3: Validation with Naïve Bayes classifier including performance measure.

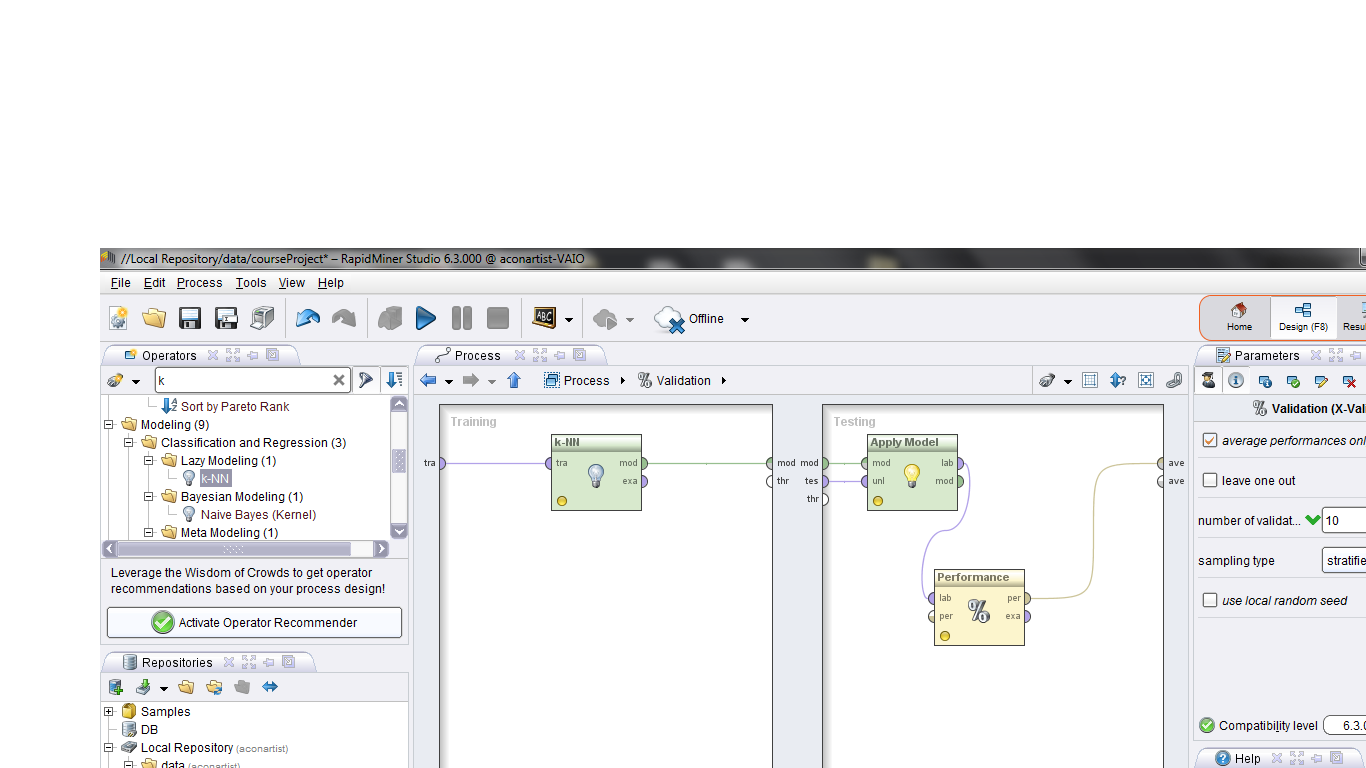


Figure 4: Validation with KNN including performance measure.

Inside the validation is the classifier, apply model and performance operator. After setting up the validation process, we filter the data some more to make sure the output is in the format that we desired.

**Implementation of methods:**

To recall, the three methods that we used was the Decision Tree, Naïve Bayes, and KNN classifier. The idea of classification is the same process for each method. The dataset will be partitioned into training dataset and test dataset. The training set will be put into the learn model algorithm that is already built in Rapidminer. Once the algorithm has learned the training set with the tree induction algorithm that is also built in, a tree model will be used for future instances of test cases. We can then apply the model on the test dataset which was partitioned in the beginning and see how well the test dataset matches up with the tree model that was created.

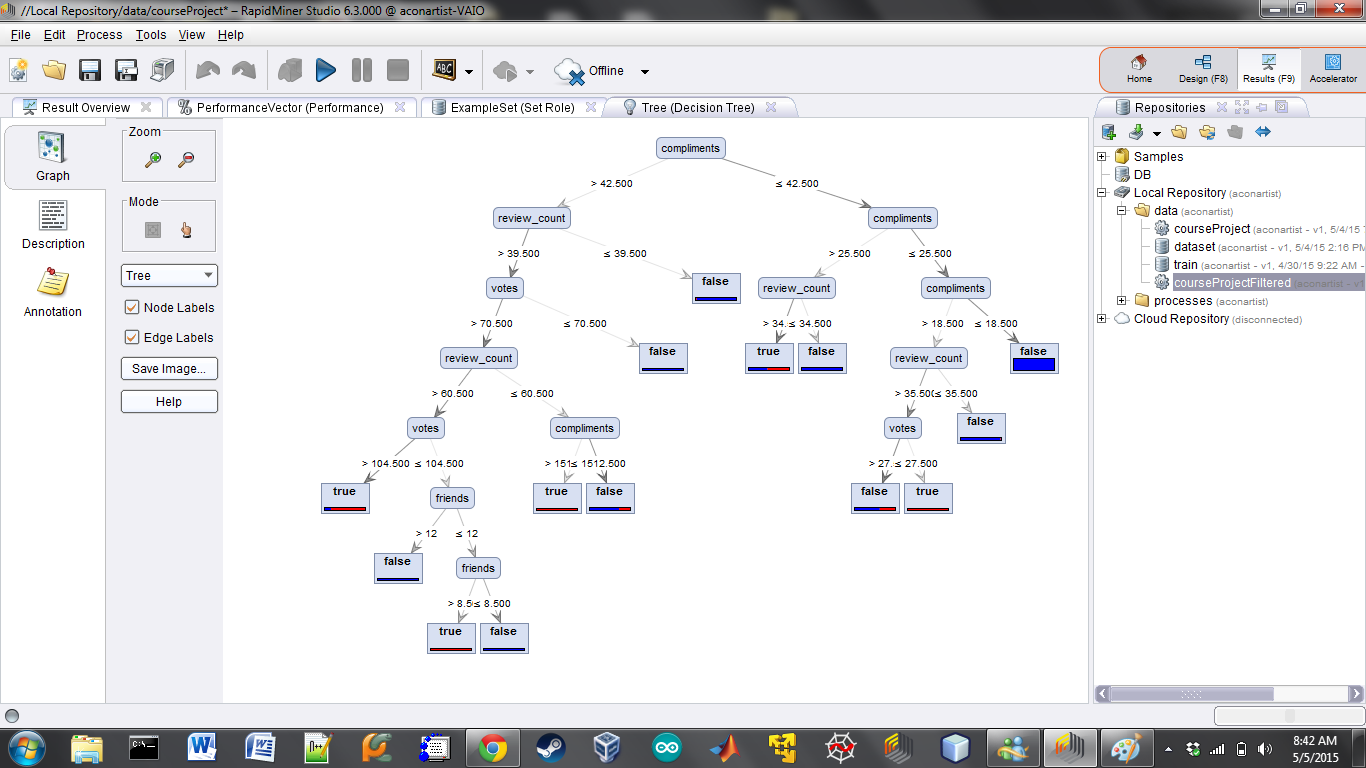


Figure 5: Decision Tree Construction.

For the Decision Tree classifier, the team used both the multiple and binomial split methods and the binomial seemed too turned out the best. We used the k fold cross validation to create the best classification model. K fold will partition the dataset into K different splits and decide which split is the best.

For the Naïve Bayes classifier, probabilities will be calculated to determine which class a new test data falls under. One example is if one test data containing atr1 = yes, atr2 = no, atr3 = yellow, then calculate the probability of P(C|atr1 = yes, atr2 = no, atr3 = yellow) where C is the different type of classes that the new test dataset can fall under. The probability with the highest value is chosen and the test data is most likely will fall under that class category according to Bayes theorem. The problem is that numerator of

P(C|a1,a2,..,aN) = P(a1,a2,..,aN|C)/ P(a1,a2,..,aN) can only be equal to P(a1|C)\*P(a2|C)\*…\*P(aN|C). There, we have to assume that there is independence among the attributes when given a class.

For K nearest neighbor classification, we used Euclidean distance and a chosen k value to determine if a point on a grid falls into a cluster. There are drawbacks to this method because if k is too small, then it becomes sensitive to noise points which are incorrect data values. If k is too large, unnecessary points may be included from other classes. Also, K nearest neighbor is relatively expensive in terms of time and is considered a lazy learner by other data miners. This method was the least fond of for our team.

**Results and Evaluation:**

The evaluation method that we decided to use to determine if one classifier is better than the other was through accuracy measure. The following formula is used to determine how accurate each classifier was:

Accuracy = TP + TN / (TP+TN + FN + FP)

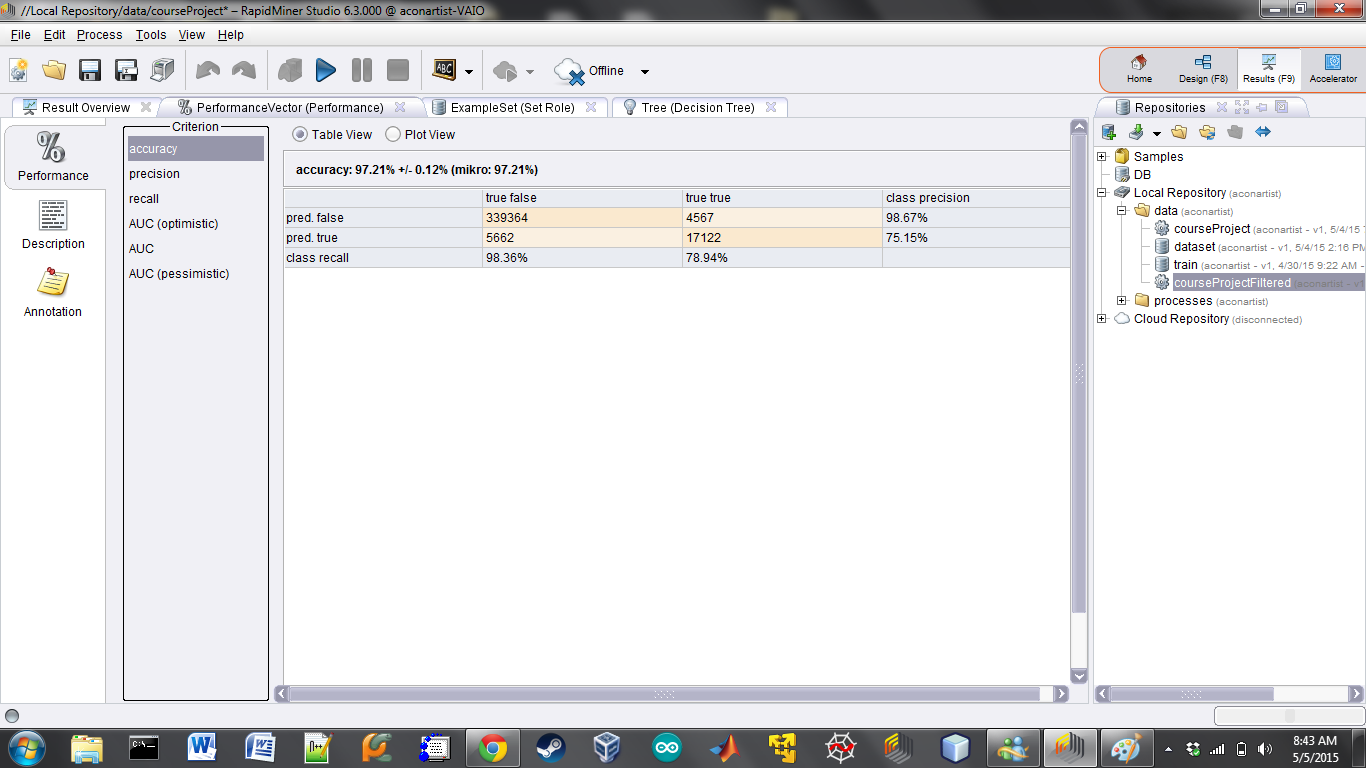


Figure 6: Confusion matrix decision tree.

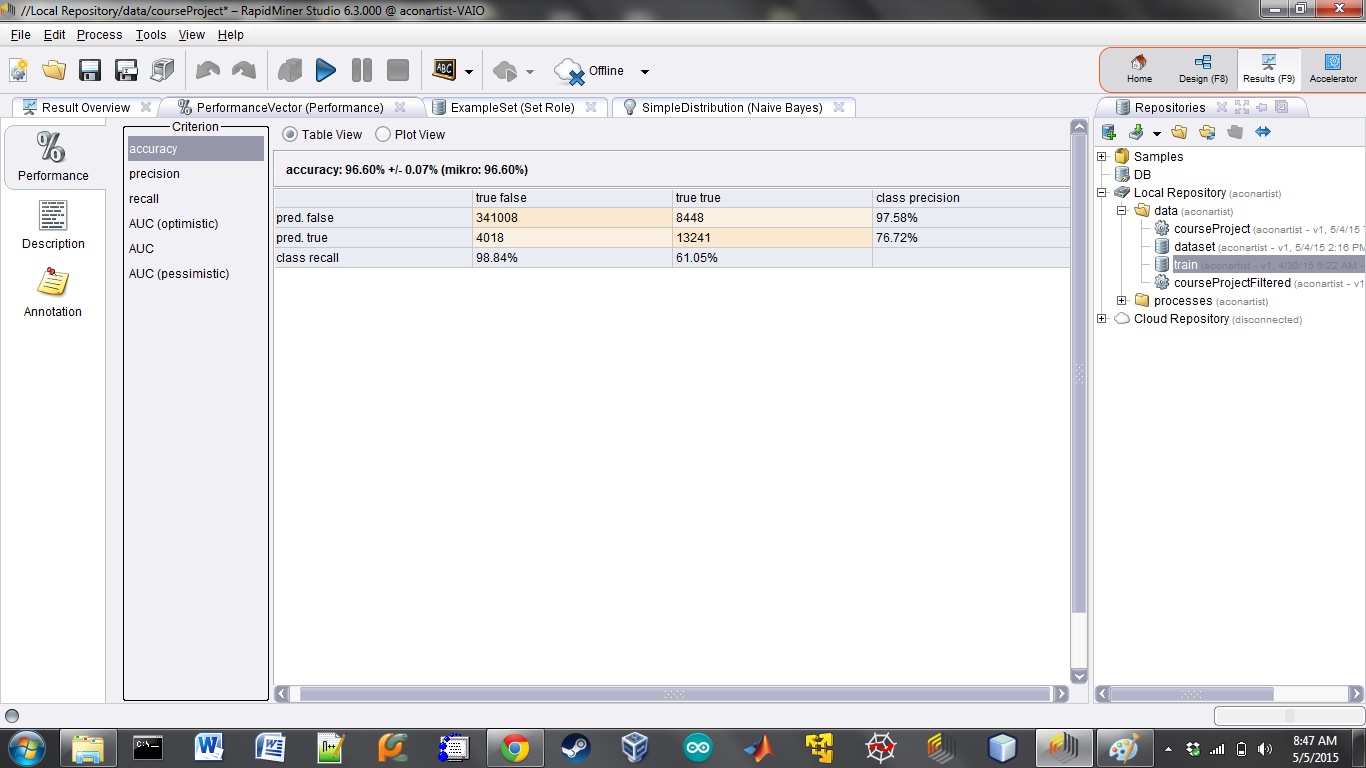


Figure 7: Confusion Matrix Naïve Bayes.

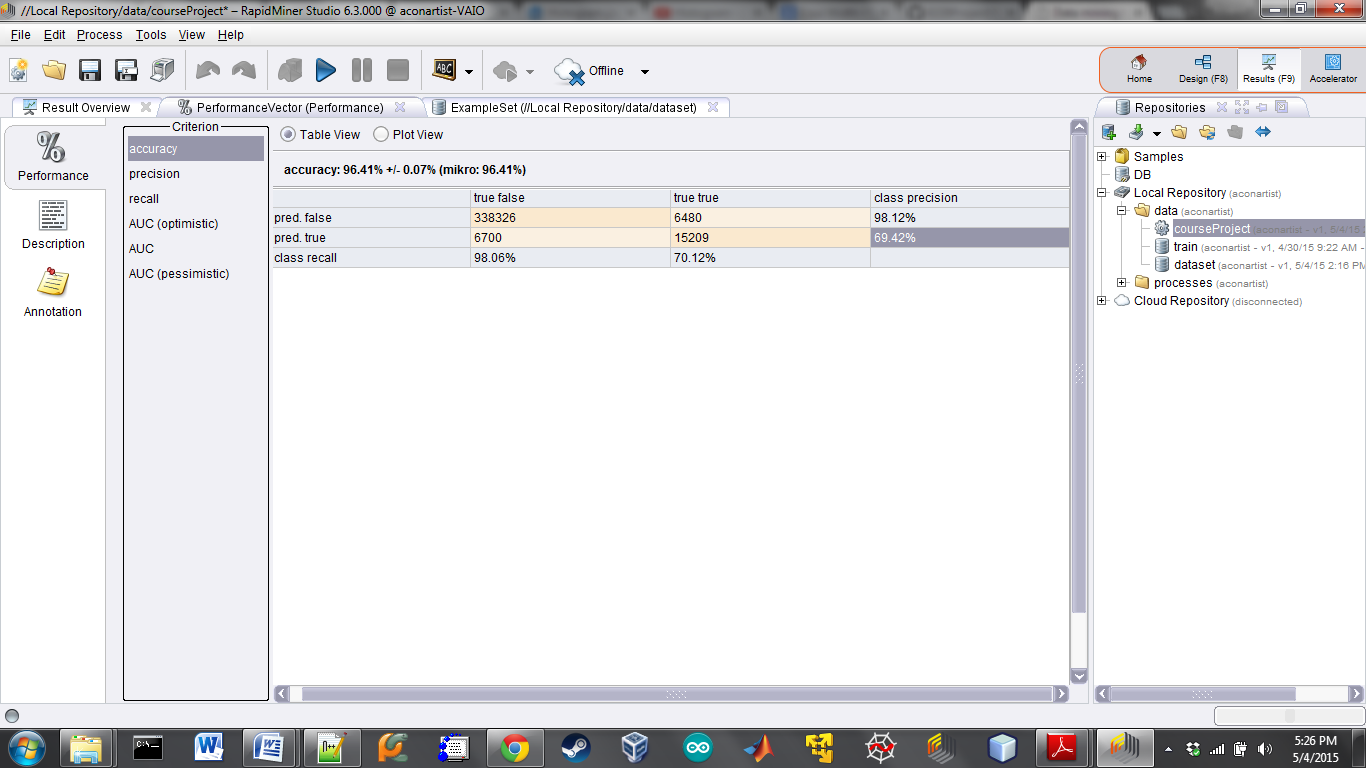


Figure 8: Confusion Matrix KNN.

The decision tree had an accuracy of 97.21 percent, Naïve Bayes with 96.60 percent, and KNN with 96.41 percent. We have decided that the best classifier for our problem is the decision tree classifier. Even though choosing any of the classifiers will give good result in the end, we will stick to our decision to choose the classifier with the best accuracy percentage rate.

Our team had noticed that if a user has many friends, interact with reviews, complements and overall large values for their attribute, then they will most likely be an elite candidate. This goes back to our motivation and objectives section. We have already explained that yelp selected and handed out elite ranks to users according to their Authenticity, Contribution and Connection. Our results visually show that we have followed these requirements. The only assumption that we made is that the authenticity of the dataset that is given only shows users that have profile pictures.

**Presentation/Visualization of the Outcome:**

For the presentation of our results or outcome, we have decided to display our results on a website we have created for demonstration purposes. The website was created using a web tool called Bootstrap. This website is constantly changing according to the device that users are using to visit our website. Our website is divided into several different sections. The sections are: Problem, Our Solution, Tools, Results, and Team member information. The problem section describes our motivation and tasks that we had or hoped to accomplish. Our Solution section contains a description of how we tackled our problem section. Next, our Tools section covers the various tools that we used to mined our data. Following that is the results section that displays several of our results from the mined data. The items that are displayed will be the possible candidates that call in the false positive category. The false positive category is when a user is not an elite but the classifier predicted them to be an elite. These groups are the potential candidates that the Yelp employees can look at to determine if they qualify under further inspection. This way, Yelp employees will not have to look at the entire dataset of users but instead look at partitioned or subset of the dataset that already fit majority of the requirements that they have for a user to become an elite. Again, the goal of this website is to display potential users to the Yelp employees to easily guide them into selecting elites better and faster.

**URL:**

The following website is hosted by omega servers. We choose this because anyone with an omega account will have the ability to have their own website.

<http://omega.uta.edu/~txm6276/>