

**ASSIGNMENT FRONT SHEET**

**Course Name: ALY6020 20906 Predictive Analytics**

**Professor Name: Dr. Yevgeniy Frenkel**

**Student Name: Dong Quoc Tuong (Lukas)**

**Student Class: Fall 2019 CPS Term: B. 2020**

|  |
| --- |
| **Module 4: Random Forests**  **Completion Date: May 3rd Due Time:12:00am** |

**Statement of Authorship**

*I confirm that this work is my own. Additionally, I confirm that no part of this coursework, except where clearly quoted and referenced, has been copied from material belonging to any other person e.g. from a book, handout, another student. I am aware that it is a breach of Northeastern University’s regulations to copy the work of another without clear acknowledgement and that attempting to do so renders me liable to disciplinary procedures. To this effect, I have uploaded my work onto Turnitin and have ensured that I have made any relevant corrections to my work prior to submission.*

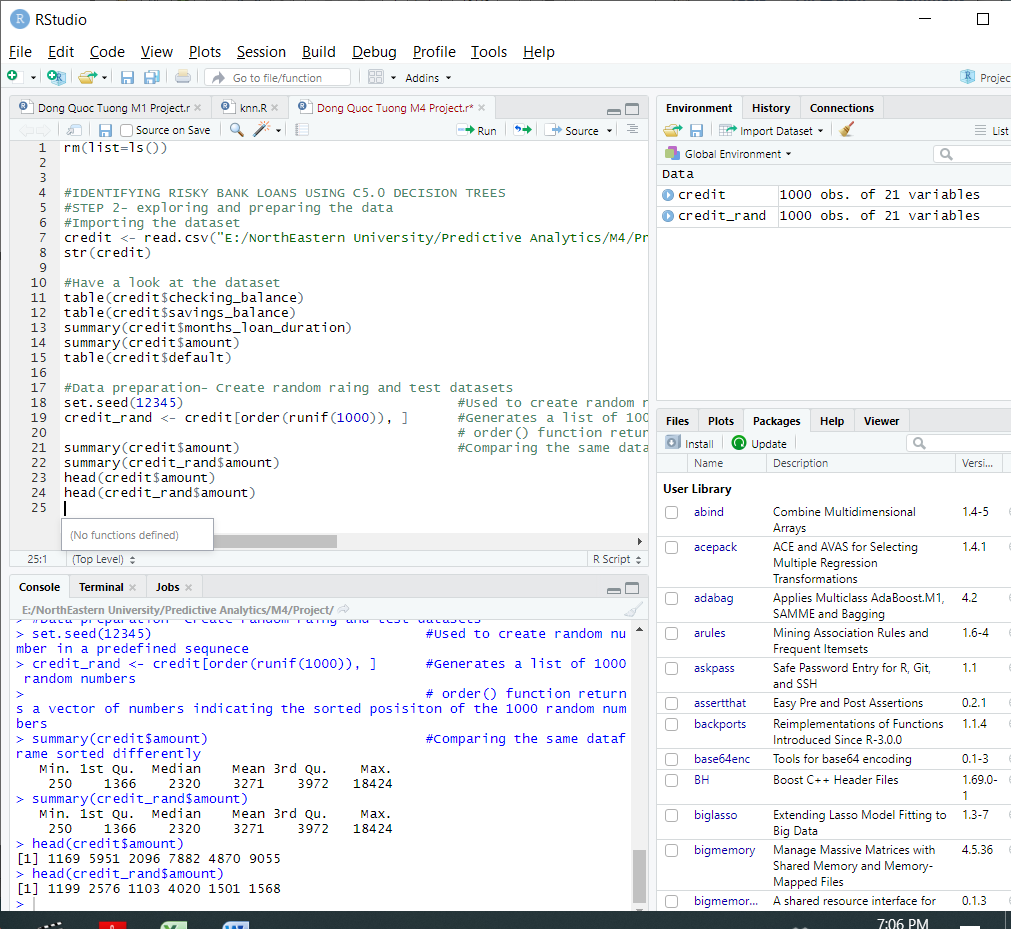
**Tick here** to confirm that your paper version is identical to the version submitted through Turnitin

**Executive Summary**

Decision Trees are widely used in the banking industry thanks to their high accuracy to ability to formulate a statistical model in plain language. (Gupta, 2017) There is a wide array of implementation of decision trees, but the most well-known is the C5.0 algorithm. In this paper, we will have a look at developing a simple credit approval model using the C5.0 algorithm and then later compare the result with the one we get from the Random Forest Model.

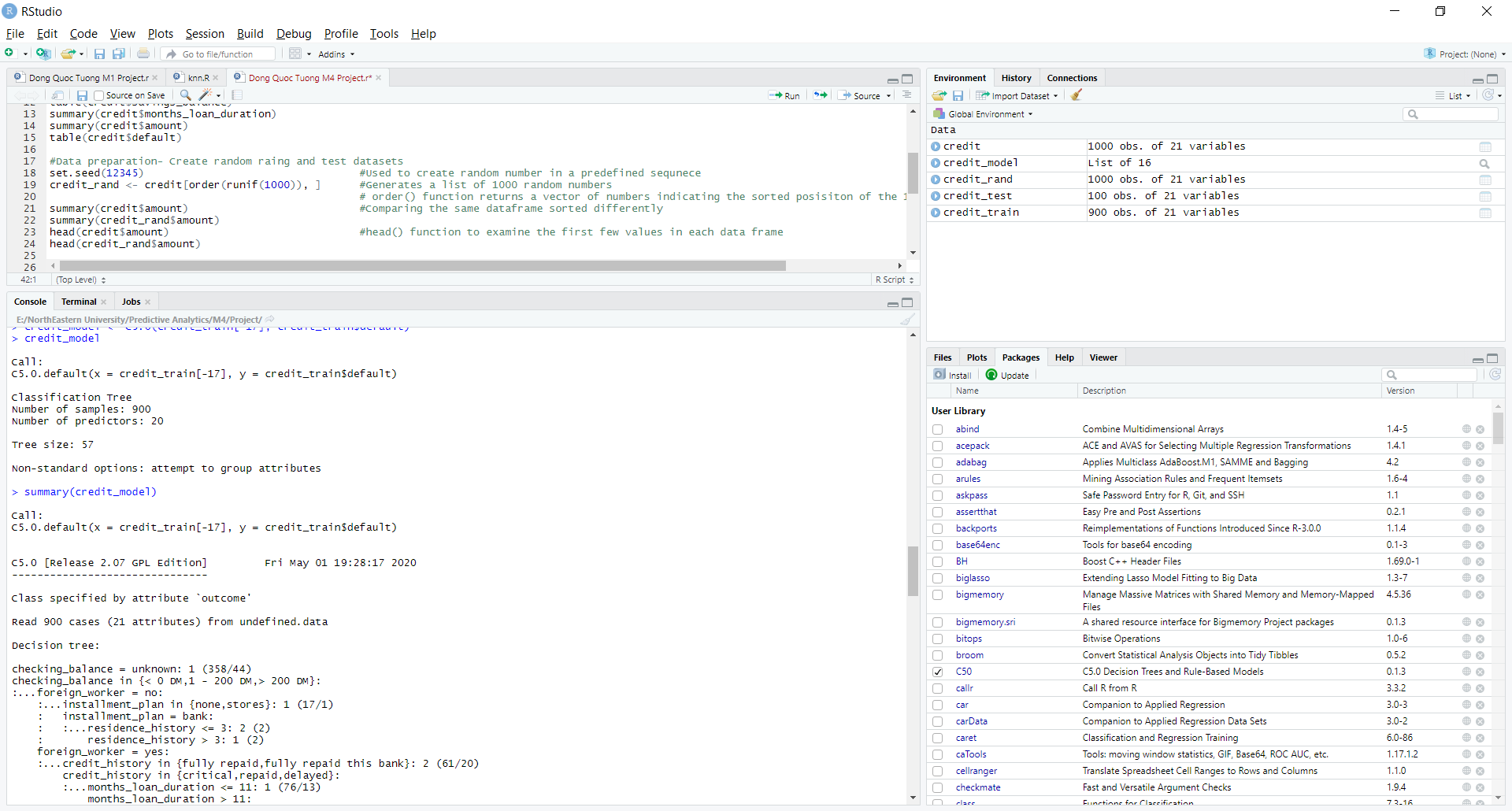
**Analysis**

The credit dataset comprises 1000 inputs of loans, plus a set of numeric and nominal features indicating characteristics of the loan and the loan candidates. The result that we need to pull is whether the loan went into default. The loan amounts ranged from 250DM to DM across terms of fours to 72 months. The median amount is 2320DM and the median duration of 18 months. There is a total of 30 % of the loans in the dataset went into default (2)



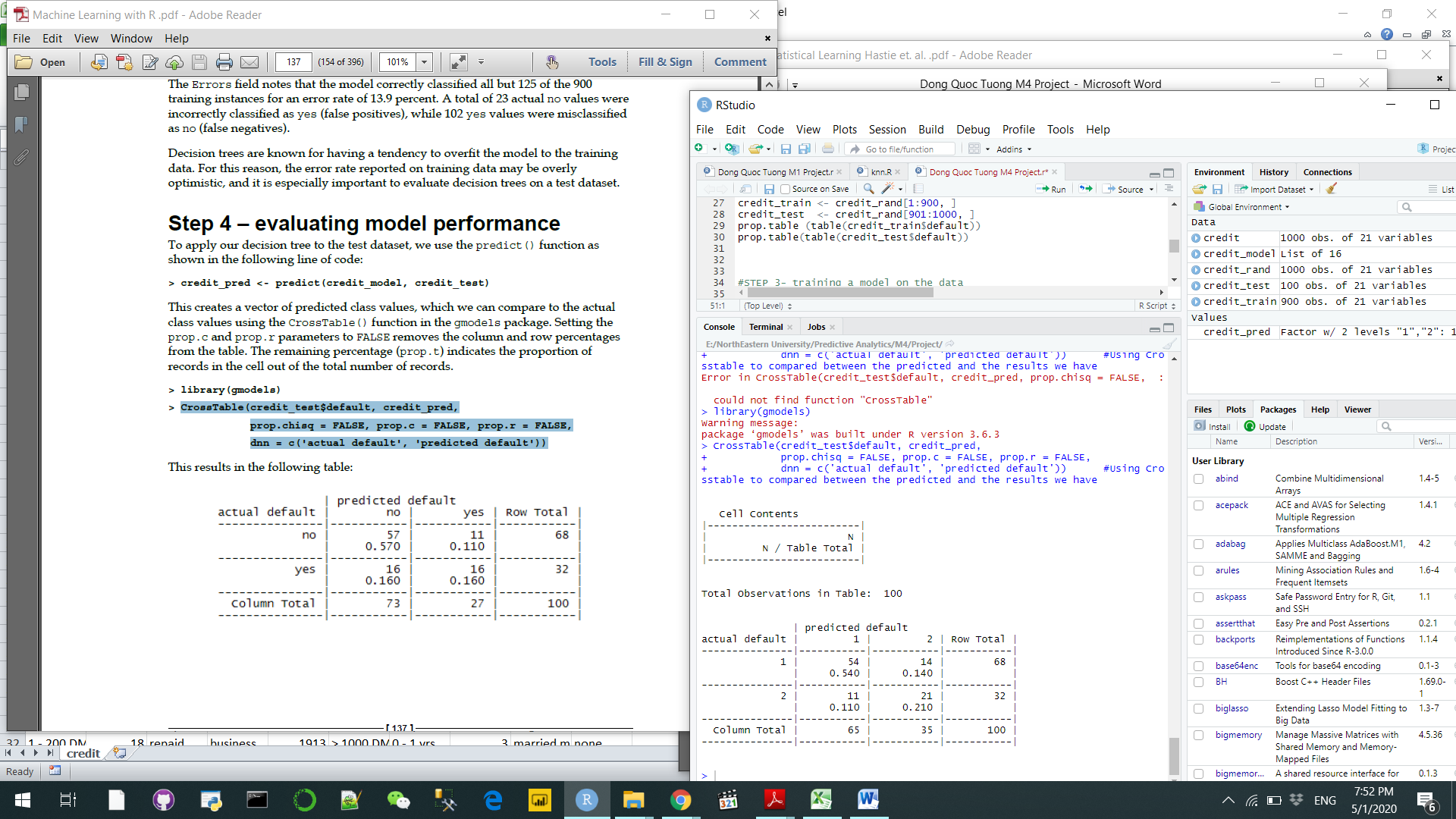
The below text indicated some simple fact about the trees, including the function call that generated it, the number of features (labeled predictors), and examples (labed samples) used to grow the tree. The tree size is 57, which means that our decision tree has 57 decision steps- slightly greater than the example trees that we had so far. The first four lines shows some of the initial determinants in our decision tree, for example:

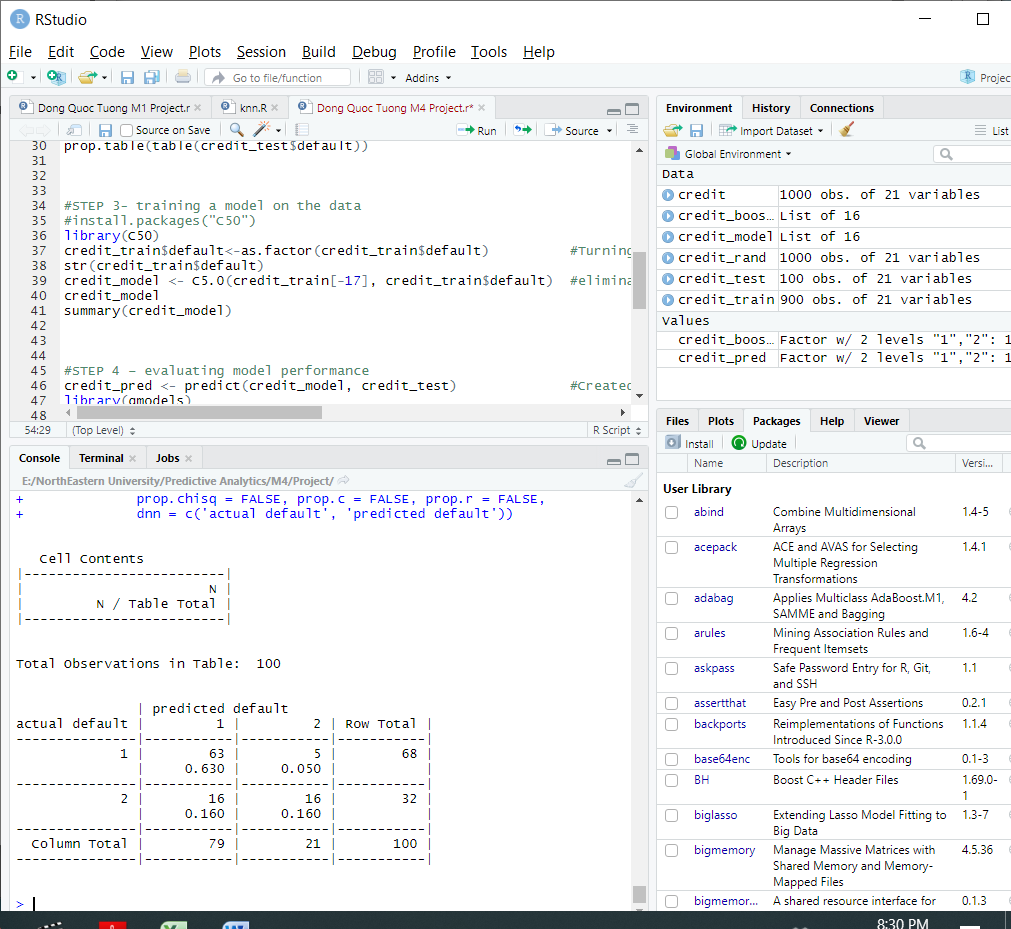
1. If the checking account balance is unknown, then we will categorize it as **not likely to default**.
2. If not, then we move on to step 2 which is if the checking account balance is less than zero DM, between one and 200 DM, or greater than 200 DM and…
3. The credit history is very good or perfect, and…
4. There is more than one dependent, then we will categorize that input as **likely to default**.



**Q1 +Q2**

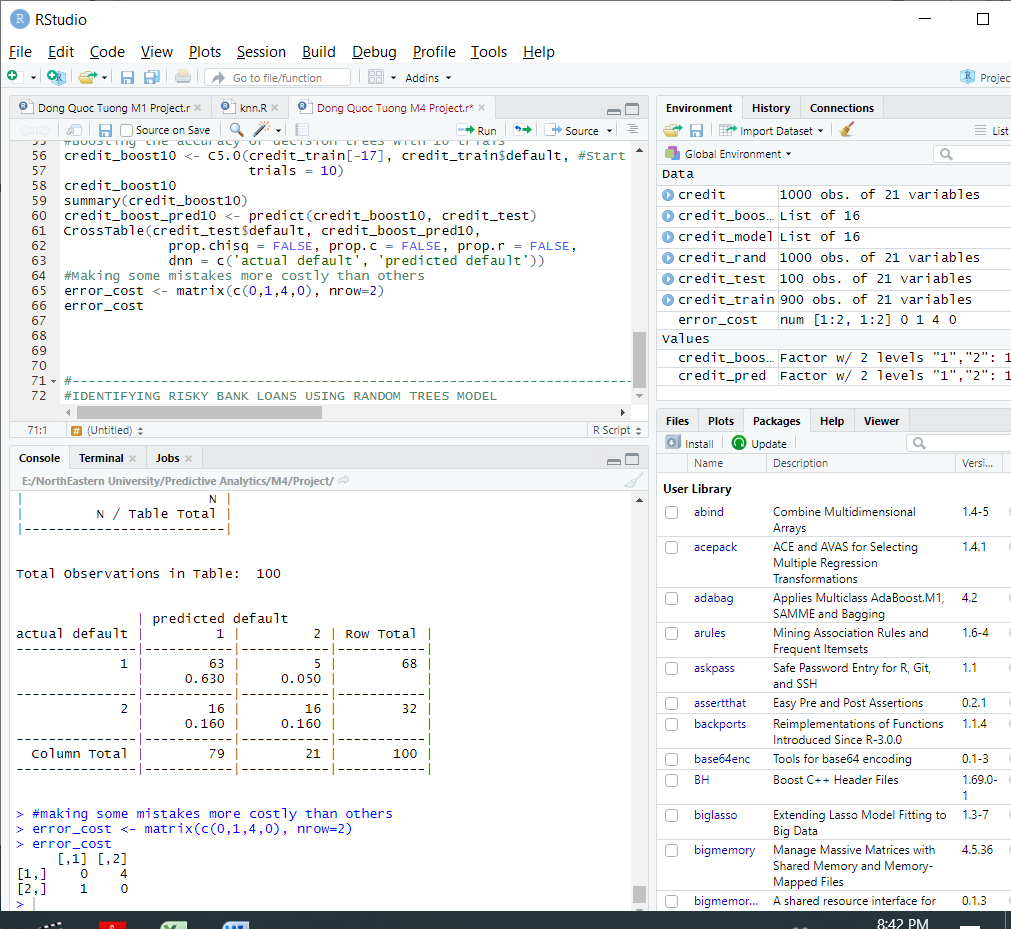
The metrics that I will using is the CrossTable to evaluate the model. From the previous paper that we did in the past, we know that the normal Precision and Recall numbers or even Cross table are good enough to evaluate our model already. Out of the 100 loan applications, our models predict correctly 54 cases did not default and 21 cases did, resulting in the accuracy of 75% and an error rate of 25%. The precision rate would be 0.79 while the recall rate would be 0.83. These are high results and the problem that we are trying to solve is not life and death so we can settle for this figure or improve them using the Boosting Trials 10 and Amaking some mistakes more costly than others.



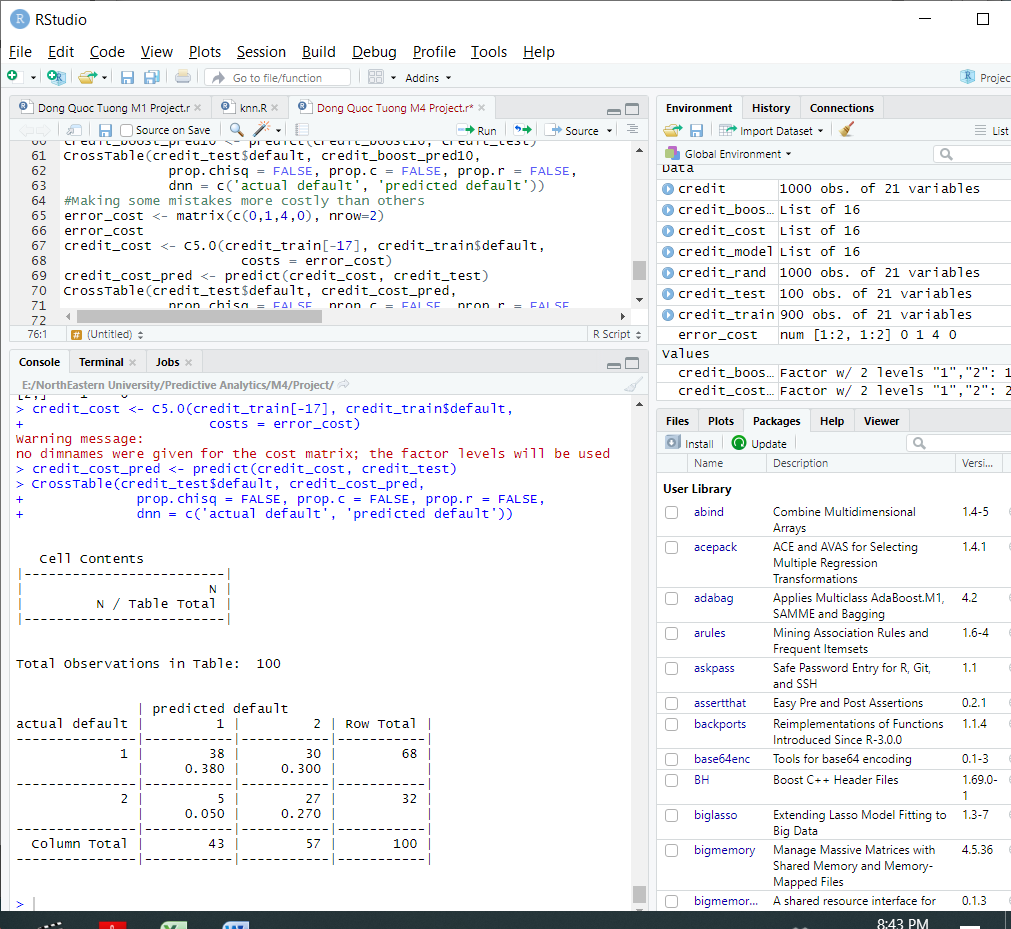
After applying the Boosting Trials 10, we reduced the error range from 25% to only 21%. It does not seem much but the Precision rate increased from 0.79 to 0.92 while the recall rate dropped from 0.83 to 0.8. This is the tradeoff between the precision and recall rate and such action was worth it because the precision significantly changed for the better while we just need to scarify the 0.3 for the recall rate. That said, we cannot apply boosting by default to every decision tree due to two reasons. Firstly, the decision tree building process normally takes a quiet amount of computation times and running them multiple times is nearly impractical. Secondly, if the training data is very noisy, the boosting might not yield better results.

As defined by this matrix, there is no cost assigned when the algorithm categorizes a

No (1)or yes (2) accurately, but a false negative (2) equals to has a cost of 4 versus a false positive's cost of 1. We will examine the how this affects the underlying classification by using applying it to our decision tree with the cost parameters

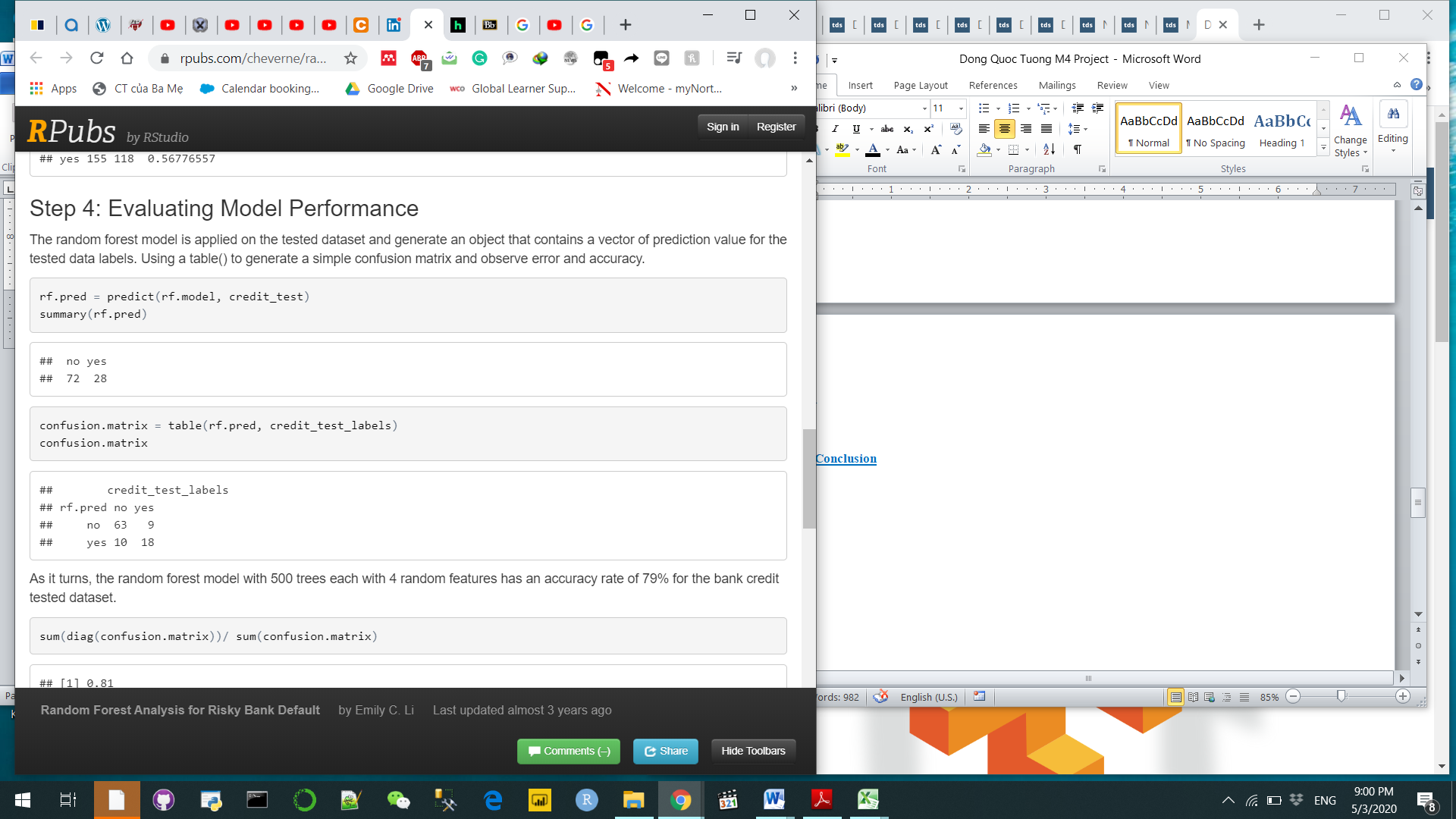


Unfortunately, this version makes more mistake than the two aforementioned models. The Precision rate dropped to only 0.55 the Compared to our best boosted model, this version makes more mistakes overall while the recall rates skyrocketed to 0.88. Thus, we will settle with the Boosting Trails 10 as our best model for Decision Tree using C5.0 algorithm

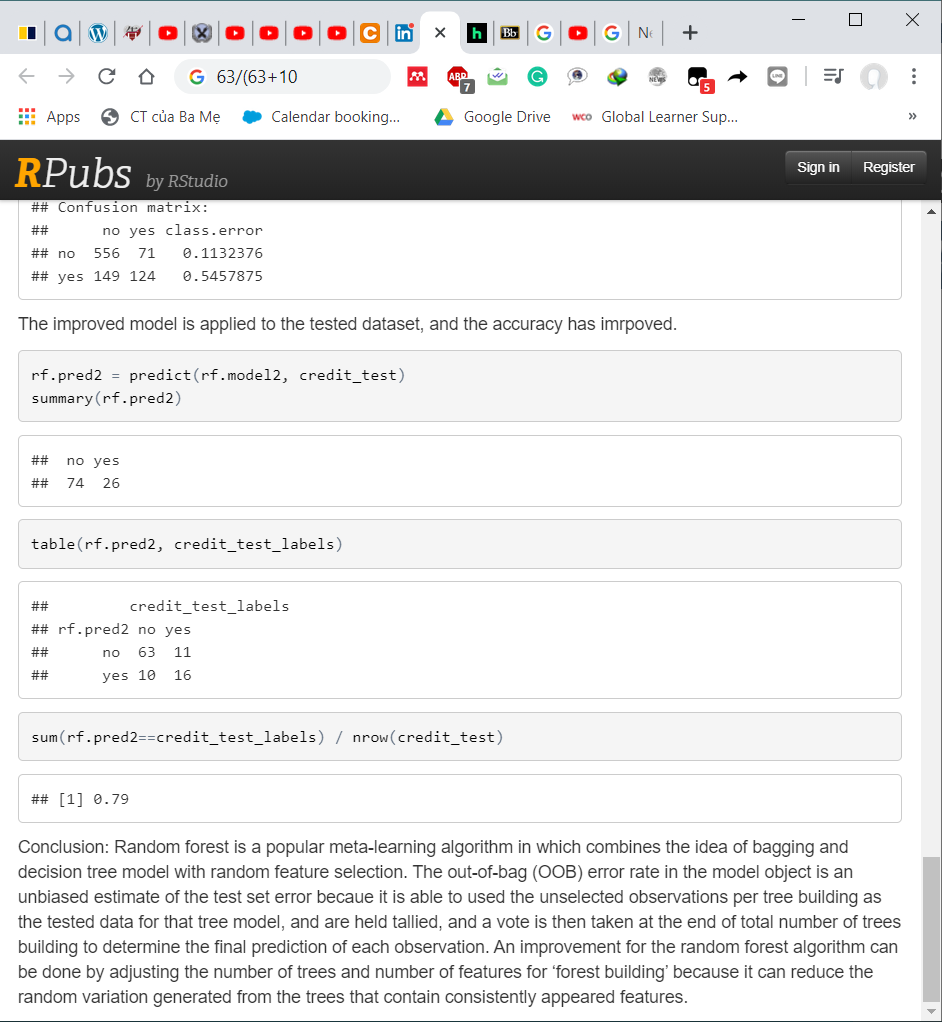


**Q3 + Q4**

In this part of our paper, we will use the randomForest() function to conduct our prediction model. (Yiu, 2019) The function is asked to generate 500 random trees to generate a “forest: with each tree using a square root of the number of factors as a convention. The model displays the fact that we have the accuracy rate of 75.6% and error rate of 24.44%. The Precision rate is about 0.87 while the recall rate is 0.83. These figures are higher compared to the C5.0 Decision tree model and much more balanced, but they are less impressive when compared to the BoostedDT .



.Then after that, we create the random forest by changing the ntree from 500 to 1000 and select 8 features instead of 4 at a time. This way, because there are only 16 total factors and haf of them were selected for building a tree each time. The improved model contains 77.2% accuracy and 22.78% error. The precision rate decreased to 0.85 and recall rate increased to 0.86



With this table below, I would argue the best parameters for random forest is ntree=500 while the mtry=4 because they have higher precision rate and situation is not that dire to start with. However, if we look at all the model as a whole we would go with the Decision tree that was boosted after 10 trials to get the maximum impact.



**Conclusion**

In the end, we see that the both Random Forest and Decision Tree are very deciding whether or not an loan applicants are worthy of a loan based individuals ‘ credits ratings. Despite the fact that standard Random Forest has greater result than the standard Decision, ultimately, it was the Boosted Decision Tree that took the center stage.

**References**

Gupta, P. (2017). Decision Trees in Machine Learning. Retrieved from <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>

Yiu, T. (2019). Understanding Random Forest. Retrieved from <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>