Predicting University Students' Academic Success and Major using Random Forests

Cédric Beaulac Jeffrey S. Rosenthal

August 23, 2018

Abstract

In this article, a large data set containing every course taken by every undergraduate student in a major university in Canada over 10 years is analysed. Modern machine learning algorithms can use large data sets to build useful tools for the data provider, in this case, the university. In this article, two classifiers are constructed using random forests. To begin, the first two semesters of courses completed by a student are used to predict if they will obtain an undergraduate degree. Secondly, for the students that completed a program, their major is predicted using once again the first few courses they've registered to. A classification tree is an intuitive and powerful classifier and building a random forest of trees lowers the variance of the classifier and also prevents overfitting. Random forests also allow for reliable variable importance measurements. These measures explain what variables are useful to both of the classifiers and can be used to better understand what is statistically related to the students' situation. The results are two accurate classifiers and a variable importance analysis that provides useful information to the university.

Keywords : Higher Education, Student Retention, Academic Success, Machine Learning, Classification Tree, Random Forest, Variable Importance

1 Introduction

Being able to predict if a student is at risk of not completing its program is valuable for universities that would like to intervene and help those students move forward. Predicting the major that will be completed by students is also important in order to understand as soon as possible which program attracts more students and allocate resources accordingly. Since gathering data can be an expensive procedure, it would be useful being able to predict both of these things using data the university already possesses such as student records. Understanding which variables are useful in both of these

predictions is important as it might help understand what drives student in taking specific classes.

Formally, these two prediction problems are classification ones. To solve these, a popular machine learning algorithm is used, a classification tree. A classification tree is an easy to interpret classification procedure that naturally allows interactions of high degree across predictors. The classification tree uses the first few courses attempted and grades obtained by students in order to classify them. To prevent overfitting and to reduce the variance of this classifier, multiple trees are grown and the result is a random forest. A random forest can also be used to assess variable importance in a reliable manner.

The University of Toronto provided a large data set containing individual-level student grades for all undergraduate students enrolled at the Faculty of Arts and Science at the University of Toronto - St. George campus between 2000 and 2010. The data set contains over 1 600 000 grades and over 65 000 students. This data set was studied by Bailey et al. [2] and was used to build an adjusted GPA that considers course difficulty levels. Here, random forest classifiers are built upon this data set and these classifiers are later tested.

The contribution in this article is two-fold. First, classifiers are built and the prediction accuracy of those classifiers exceeds the accuracy of the linear classifiers thus making them useful for universities that would like to predict where their resources need to be allocated. Second, the variable importance analysis reveals the great importance of grades, but more precisely, grades in departments that are considered low-grading departments. This result is interesting as many researchers are still trying to understand the repercussion of the distorted grade inflation that is affecting many universities in recent years.

2 Literature review

2.1 Predicting success

In this article a model is established to predict if a student succeeds at completing an undergraduate program and to predict what major was completed. To classify a student, the algorithm uses information about the first few courses taken by them. The task of predicting student academic success has already been undertaken by many researchers. Recently Kappe and van des Flier [17] tried to predict academic success using personality traits. In the meanwhile, Glaesser and Cooper [12] were interested in the role of parents' education, gender and other socio-economic metrics in predicting high school success.

While the articles mentioned above use socio-economic status and personality traits to predict academic success, many researchers are looking at academic-related metrics to predict graduation rates. Johnson and Stage [15] use High-Impact Practises, such as undergraduate research, freshman seminars, internships and collaborative assignments to predict academic success. Using regression models, they noted that freshman seminars and internships were significant predictors. Niessen and al. [24] discuss the significance of trial-studying test in predicting student dropouts. These tests are designed to simulate a representative first-year course and student would take them before admission. The authors noted that this test was consistently the best academic achievement predictor.

More recently, Aulck and al. [1] used various machine learning methods to analyse a rather large data set containing both socio-economic and academic metrics to predict dropouts. They noted similar performances for the three methods compared; logistic regression, k-nearest neighbours and random forests. The proposed analysis differs from the above-mentioned as it takes on the challenge to predict academic success and major using strictly academic information available in student records. The benefits of having classifiers built upon data they already own is huge for university administrations. It means university wouldn't need to force students to take entry tests or relies on outside firms in order to predict success rate and major which is useful in order to prevent dropout or to allocate resources among departments. As noted by Aulck and al. [1] machine learning analysis of academic data has potential and the uses of random forest in the following article aims at exploiting this potential.

2.2 Identifying important predictors

Identifying and interpreting the variables that are useful to those predictions are important problems as well. The precise effect of grades on a student motivation lead to many debates and publications over the years (more recently [23] [25]). Other than the trivial relation between having good grades and completing a program, understanding the importance of the grades in predicting if a student will complete their program or not is still a relevant question. Random forest mechanisms lead to variable importance techniques that will be useful to understand how grades affect student choices.

Understanding the importance ranking of grades in various departments can also enlighten us regarding the phenomenon of *grade inflation*. This problem and some of its effect has been already discussed in many papers ([26], [16], [3]) and it is consensual that this inflation differs from one department to another. According to Sabot and Wakeman-Linn, [26] this is problematic since grades serve as incentives for course choices for students and

now those incentives are distorted by the grade inflation. As a consequence of the different growths in grades, they noted that in many universities there exist a chasm in grading policies creating high-grading departments and low-grading departments. Economics, Chemistry and Mathematics are examples of low-grading departments while English, Philosophy, Psychology and Political Science are considered high-grading.

As Johnson mentions [16], students are aware of these differences in grading, openly discuss them and this may affect the courses they select. This inconsistency in course difficulty is also discussed by Bailey, Rosenthal and Yoon [2] as they built an adjusted GPA that considers course difficulty levels. The accuracy of that adjusted GPA in predicting uniform test result is a great demonstration that courses do vary in difficulty. It seems important to analyse if the importance of a grade variable is somehow tied to whether it is coming from a high-grading or a low-grading department.

Finally, since some of the High-Impact Practises analysed by Randall Johnson and King Stage [15] are part of the University of Toronto's program, the variable importance analysis should be able to tell us more about those practices.

3 Methodology

3.1 Data

The data set provided by the University of Toronto contains 1 656 977 data points, where each observation represents the grade of one student in one course. A data point is a 7 dimensions observation containing the student ID, the course title, the department of the course, the semester, the credit value of the course and finally the numerical grade obtained by the student. As this is the only data obtained, some pre-processing is required in order for a classification tree to be used. The **first research question** is whether it is possible to design an algorithm which accurately predicts whether or not a student will complete their program. The **second research question** is whether it is possible to design an algorithm which accurately predicts, for students who complete their program, which major they will complete. Those two predictions will be based upon first-year student records.

The data has been pre-processed for the needs of the analyses. At the University of Toronto, a student must complete 20 credits with a GPA of 1.85 or more in order to obtain an Honours B.A. or B.Sc [28]. A student must also either complete 1 Specialist, 2 Majors or 1 Major and 2 Minors. The first five credits attempted by a student roughly represent one year

of courses. Therefore, for each student every semester until the student reaches 5 attempted credits are used for prediction. It means that for some students, the predictors represent exactly 5 attempted credits and for some other students, a bit more. The set of predictors consists of the number of credits a student attempted in every department and the average grade across all courses taken by the student in each department. Since courses were taken by students in 71 different departments, the predictor vector is of length 142. Note that other predictors could have been extracted of the data set, this could be explored in future research project.

To answer the first research question, a binary response indicating whether or not a student completed their program is needed. Since it is assumed that students can take classes in other universities or faculties, every student who completed 18 credits are considered students who completed a program. Students who registered to 5 credits worth of courses, succeeded at fewer than 18 credits worth of courses and stopped taking courses for 3 consecutive semesters are considered students who began a program but did not complete it. This way, students who started a program but did not obtain 18 credits by the fall 2010 semester were removed from the analysis as it is impossible to tell whether or not they ended up completing their program. After this pre-processing was performed, the data set contains 38 842 students of which 26 488 completed an undergraduate program and 12 294 did not.

To answer the second research question a categorical response representing the major completed by the student is required. To do so, the 26 448 students who completed a program are kept. The response will represent the major completed by the student. Since this information is not obtainable, the department in which a student completed the largest number of credits is considered the program they majored in. Therefore, the response variable is a categorical variable that can take 71 possible values. This formatting choice might be a problem for students who completed more than 1 major. Some recommendations to fix that problem can be found in the conclusion.

Regarding the various grading policies of this university it was noticed that Mathematics, Chemistry and Economics are the three departments with the lowest average grades. As grades do vary widely across our data set there is no statistically significant difference between the departments but it is still interesting to observe that departments that were defined as low-grading departments in many papers do appear as the lowest grading departments in this data set too. Finally, the data set was divided in three parts as is it usually done. In order to include as much data as possible in the training set, it contains 90% of the total data set while both the validation and test set contains 5% each.

3.2 Classification Tree

A typical supervised statistical learning problem is defined when the relationship between a response variable and an associated set of predictors is of interest. The response is what needs prediction, such as the program completion, and the predictors are what are being used to predict the response, such as the grades. When the response variable is categorical, this problem is defined as a classification problem. One challenge in classification problems is to use a data set in order to construct a classifier. A classifier is built to emit a class prediction for any new observation with unknown response. In this analysis, classifiers are built upon the data set described in section 3.1 to predict if a new student will complete its program and what major will be completed using information related to its first year of courses.

A classification tree [7] is a model that classifies new observations based on set of conditions related to the predictors. For example, a classification tree could predict a student is on its way to complete a program because it attempted more than 2 Mathematics courses, obtained an averaged grade in Mathematics above 80 and attempted fewer than 2 Psychology courses. The set of conditions leads to a partitioning of the space defined by possible predictors values. Intuitively, a classification tree forms regions defined by some predictors values and assign a response label for new observations that would belong in those regions. Figure 1 illustrates an example of a predictor space partition and its associated classification tree for observations defined by two predictors. The name classification tree comes from the ability to represent the final set of regions as leaves in a tree as represented in Figure 1.

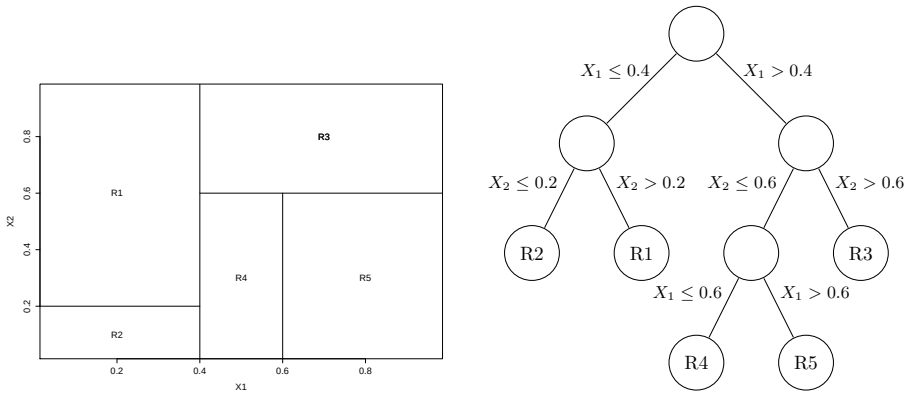

Figure 1: Illustration of the decision tree partitioning process and the resulting tree

Now that the model has been established, an algorithm that creates the classification tree using a training set of labelled observations needs to be defined. The algorithm creates the regions by recursively establishing the conditions. It aims at building regions that contains a high concentration of observations of the same class. Usually a measure of impurity is defined, it measures how mixed a region is. Intuitively, it is desired to obtain a set of conditions under which all students either completed their programs or not. Therefore, the algorithm analyses how mixed are the labels according to all possible conditions and select the condition that minimizes the measure of impurity selected by the researchers. For example, the algorithm will look at all conditions of the form : "did the student attempt more or less than 1 Mathematics course ?" and select the condition that best divides students that completed a program from students that did not.

Once a condition is selected, the training observations are effectively divided in two sets of training observations based upon the condition. The process is repeatedly applied on the two resulting training sets. The algorithm divides the training observations in smaller sets until each resulting set contains few observations. When the partitioning process is completed, each region is labelled with the class representing the majority of observations respecting the conditions defining the region. A more formal definition of the algorithm is included in the appendix.

3.3 Random Forest

By constructing a decision tree, a powerful and easy to interpret classifier is obtained. One way to improve the stability of this classifier and to prevent overfitting is to build a forest of trees using bootstrap samples of the training set.

Bootstrap aggregating (*bagging*) was introduced by Breiman [4] as a way to reduce the variance of unstable classifiers and to prevent overfitting. For the classification problem, the procedure consists of using an ensemble of classifiers that will each cast a vote towards a certain class. In bagging, each classifier in our ensemble is built upon a different bootstrap sample of our training set. Regarding decision trees, the bagging procedure will in fact creates multiple trees. Breiman [6] defines a *random forest* as a classifier consisting of a set of tree-structured classifiers build upon independent identically distributed random vectors where each tree casts a unit vote for the most popular class at one input.

Suppose there is a way to obtain an ensemble of classifiers. The goal is to find a technique that uses the entire set of classifiers to get a new classifier that is better than any of them individually. One method of aggregating the

class predictions is by *voting*: the predicted class for a new observation is the most picked class among individual classifier. A critical factor in whether the bagging procedure will improve the accuracy or not is the stability of the individual classifiers. If a small variation in the training set has almost no effect on the classifier then utilizing a set of classifiers based upon similar training sets will result in a set of almost identical classifiers. For unstable procedures, the classifiers in the set are going to be very different from one another and the aggregation will greatly improve both the stability and accuracy of the procedure. Procedure stability was studied by Breiman [5]; classification trees are unstable and thus, greatly benefit from bagging.

Since having multiple training sets is unusual; bootstrap samples of the data set can be drawn to form our ensemble of training sets. A bootstrap sample is simply a random sample of the original training sets. Each of the samples are drawn at random with replacement from the original training set and are of the same size. Doing so will produce an ensemble of different training sets. For each of these training set a decision tree is fitted and together they form a random forest. Overfitting is a problem caused when a classifier identifies a structure that corresponds too closely to the training set and generalizes poorly to new observations. By generating multiple training samples, via bootstrap, fitting trees on them and building a forest out of multiple tree classifiers it greatly reduces the chances of overfitting.

A random forest classifier is more precise than a single classification tree in the sense that it has lower mean-squared prediction error [4]. By bagging a classifier, the bias will remain the same but the variance will decrease. One way to further decrease the variance of the random forest is by constructing trees that are as uncorrelated as possible. This process might increase the bias of the individual classifiers in exchange. Breiman introduced in 2001 random forests with random inputs [6] which is the most commonly used random forest classifier. The novelty of this random forest model is in the tree-growing procedure. Instead of finding the best condition among all the predictors, the algorithm will now randomly select a subset of predictors and will find the best condition among these.

Random forests are easy to use and are stable classifiers with many interesting properties. One of these interesting properties is that they allow for powerful variable importance computations that evaluate the importance of individual predictors throughout the entire prediction process.

3.4 Variable Importance in Random Forests

A variable importance analysis aims at understanding the effect of individual predictors on the classifier outcome. A predictor with a great effect is considered an important predictor. A random forest provides multiple

interesting variable importance computations. When building a tree, the algorithm picks the split variable that reduces the most of a pre-specified impurity measurement. If a tree has access to all the predictors, the first partitioning is done using the predictor that grants the largest increase in prediction accuracy. Unless predictors are randomly selected, the first few conditions established by the algorithm are based on predictors that give a high amount of information on their own. Compiling the appearances of predictors among the first few partitioning is an easy way to use the mechanisms of classification trees to assess variable importances. A problem regarding this technique is that it doesn't represent appropriately the total effect of a predictor in the model.

Starting with the entire training set, the predictor and split point that produce the largest decrease in impurity measurement is selected by the algorithm. If a predictor isn't producing the largest decrease right away but instead is frequently picked by the algorithm in the deeper levels of a tree to a point where it produces a large total decrease in impurity over multiple partitions, that predictor should be considered an important one. The *Gini decrease importance* sums the total Gini decrease caused by partitioning upon a predictor throughout an entire tree and then computes the average of this measure across all trees in a forest. This technique is tightly related to the construction process of the tree itself and is pretty easy to obtain as it is non-demanding computationally.

Finally, the *permutation decrease importance* was introduced by Breiman in 2001 [6]. Intuitively, if a predictor has a significant effect on the response we should lose a lot of prediction accuracy if the values of that predictor are mixed up in our data set. One way to disrupt the predictors values is by permutation. The procedure computes the prediction accuracy on the test set using the true test set. Then, it permutes the values of one predictor, j, across all observations, run this permuted data through the forest and compute the new accuracy. If the input j is important, we should lose a lot of prediction accuracy by permuting the values of j in the test set. The process is repeated for all predictors, then it is averaged across all trees and the averaged prediction accuracy decreases are compared. The larger the decrease in accuracy the more important the variable is considered.

Storbl & al. [27] published an article in 2007 where these techniques are analysed and compared. According to this paper, the selection bias of the decision tree procedure might lead to misleading variable importance. Numerous papers ([7], [18], [19]) noticed a selection bias within the decision tree procedure when the predictors are of different nature. There exist more potential conditions for a continuous variables, therefore if the predictors are a mix of continuous and categorical variables, there is a higher chance

that a continuous predictor is selected for the partitioning by the algorithm. The permutation decrease importance measure has a higher variance for predictors with a high number of possible conditions but it is still unbiased and with a sufficient number of trees this measurement is reliable.

The simulation studies produced by Storbl & al. [27] show that the Gini decrease importance is not a reliable variable importance measure when predictors are of varying types. Because of the selection bias towards variable with more possible conditions, the algorithm will pick those variable more frequently and the Gini decrease attributed to those will grow very large. In other words, the Gini decrease importance measure tends to overestimate the importance of variable with more possible partitioning. It seems that the permutation decrease importance is more reliable has it is unbiased.

It is shown in [27] that the variable importance techniques described above can give misleading results due to the selection bias within classification trees and the replacements when drawing bootstrap samples. It is recommended that researchers build random forests with bootstrap samples without replacements and use an unbiased tree-building procedure ([22], [18], [21], [14]). If a classic tree-building procedure is used, predictors should be of the same type or only the permutation decrease importance is reliable.

3.5 Algorithms

A classification tree using the Gini impurity as split measurement was coded in the C++ language using the Rcpp library [11]. The code is available upon request from the first author. The algorithm proceeds as explained in Section 3.2, the tree it produces is unpruned and training sets are partitioned until they contain only 50 observations. Three versions of the random forest algorithm are going to be used. **Random forest # 1** consists of 200 trees and can split upon every variable in each region. Bootstrap samples are drawn without replacement and contain 63% of the original training set. **Random forest # 2** fits 200 trees but randomly selects the variable to be partitioned upon in each region.

Finally, the popular R RandomForest package [20] was also used. It is an easy to use and reliable package that can fit random forests and produce variable importance plots. Using this package, **random forest # 3** was built. It contains 200 trees. Once again, bootstrap samples are drawn without replacement and contain about 63% of the size of the original training set. By default, this algorithm randomly selects a subset of inputs for each region. Regarding the impurity measure, the Gini impurity was selected because it has interesting theoretical properties, such as being differentiable,

and has been performing well empirically. Linear models were fit for both of the classification problems serving as benchmarks. In order for the comparison to be as fair as possible the linear model classifiers were constructed upon the same set of predictor.

4 Results

4.1 First research question : Predicting program completion

First, a random forest classifier is built to predict if a student will complete its undergraduate program. Out of the 904 students who completed a program in the test set, the **random forest # 1** predicts accurately that 91.03% completed their program. Among the students who did not complete their program in the test set, the algorithm achieves 49.92% accuracy. The result is a combined 77.75% accuracy on the entire test set.

Variable importance is measured using the permutation decrease since it is more reliable as explained in Section 3.4. The top 15 variables according to the permutation decrease were kept and ordered in Figure 2. Boxplots were also used in order to visualize the variance of these measurements.

In figure 2 and for all the following figures, the variable representing the number of credits in a department is identified by the department code, i.e. the number of credits in Chemistry is identified by CHM. The variable representing the averaged grade in a department is identified by the department code followed by the letter G, i.e CHM G represents the averaged grade in Chemistry. As expected the variance of the grade variables are a bit larger. It seems important to point out that among the top five grades variables are Mathematics (MAT), Finance (COMPG), Economics (ECO) and Chemistry (CHM) which are low-grading departments. Perhaps the strict marking of these departments helps in better distinguishing students among themselves. The ASSEM code represents a special type of first year seminar course. It seems that the students that registers in theses courses are easy to classify as both grades and the number of credits are considered important. This result agrees with the result obtained by Johnson and Stage [15] about the importance of first year seminar courses.

For **random forest # 2**, the algorithm achieves a 94.90% accuracy among the students who completed a program. Among the students who did not complete an undergraduate program in the test set, the algorithm predicts correctly that 42.74% did not complete their program. It achieves a 78.06% accuracy over the entire test set. Using the random forest where

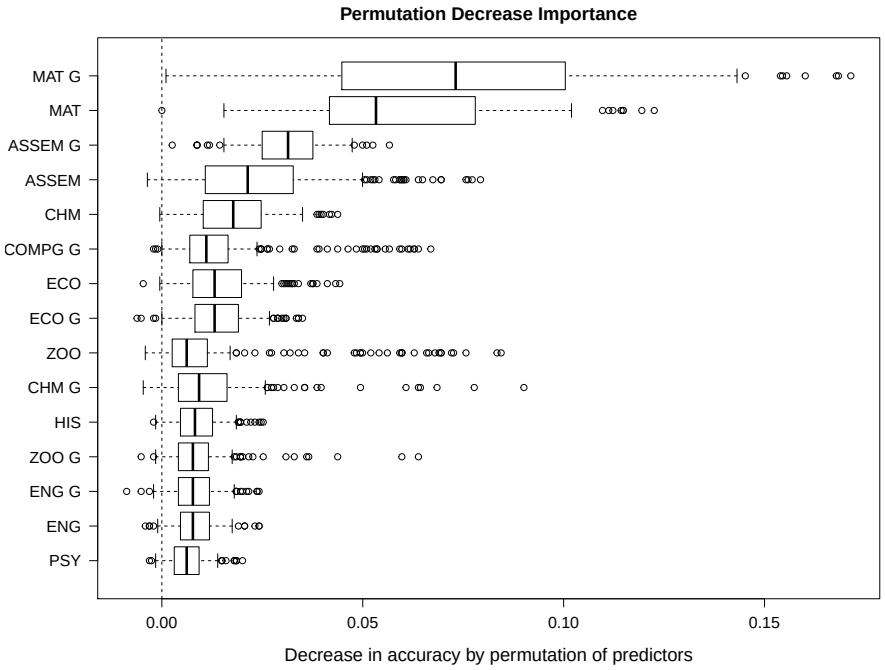

Figure 2: Variables importance boxplots for the **random forest # 1**. The 15 most important predictors are displayed. The importance of a predictor is determined by the average decrease in accuracy in the test set caused by a random permutation of the predictor.

the split variable is randomly selected greatly increases the variance of the permutation decrease of all of the variables, this is noticeable in Figure 3.

According to figure 3, mathematics grades and courses are important but also first year seminars, Chemistry and Zoology/Biology (ZOO). As a matter of fact, the grades in departments are almost always of greater importance than their number of credit counterparts. Once again, most of these are among the low-grading departments of the university.

Finally, the RandomForest package was used for the **random forest # 3**. Among the students who completed their program in the test set, the package achieves a 91.19% accuracy. Out of the 418 students who did not complete their program, the **random forest # 3** achieves a 52.95% accuracy. The combined result it a 78.84% accuracy over the complete test set.

Figure 4 contains the variable importance plot produced by the function included with the package. The grades in first year seminars and Mathe-

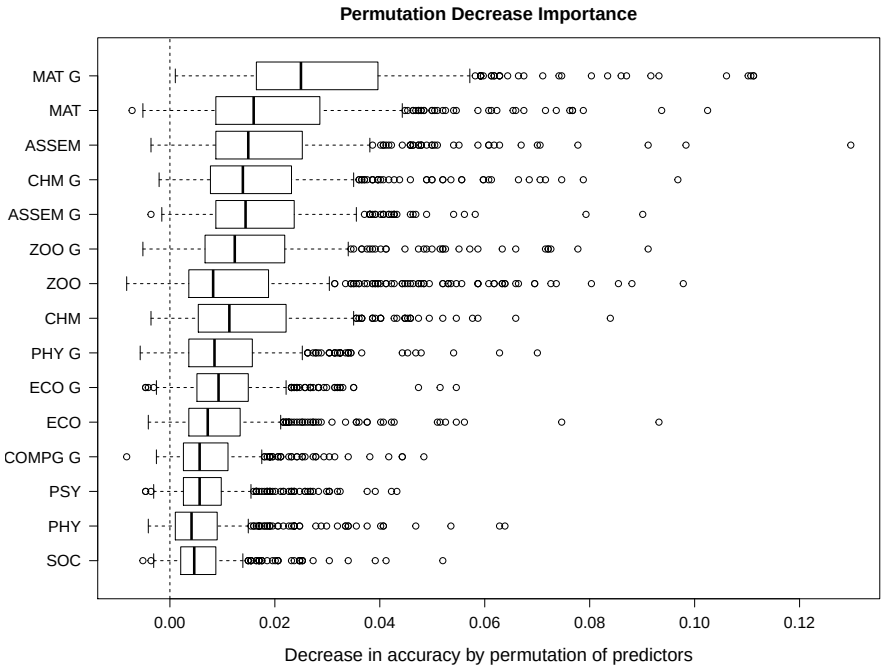

Figure 3: Variables importance boxplots for the **random forest # 2**. The 15 most important predictors are displayed. The importance of a predictor is determined by the average decrease in accuracy in the test set caused by a random permutation of the predictor.

matics are not only the two top grades variables but they are in truth the two most important predictors by a large margin according to this iteration of a random forest. The package does not provide the variance of these measurements.

To summarize, theses different forests all achieve reasonable accuracy located around 78% which is slightly above the 74% accuracy achieved with a logistic regression based upon the same predictors. They all appear to overestimate the probability of completion as they all have great accuracy for predicting who will complete but low accuracy for predicting who will drop out. These predictions can be useful for university administrations that would like to predict the number of second-year students and prepare accordingly. Theses predictions could also be useful in order to target students in need of more support to succeed.

As explained earlier, the analysis performed also contains interesting results with respect to the important predictors. The average grade in a department seems to be almost consistently more important than the amount

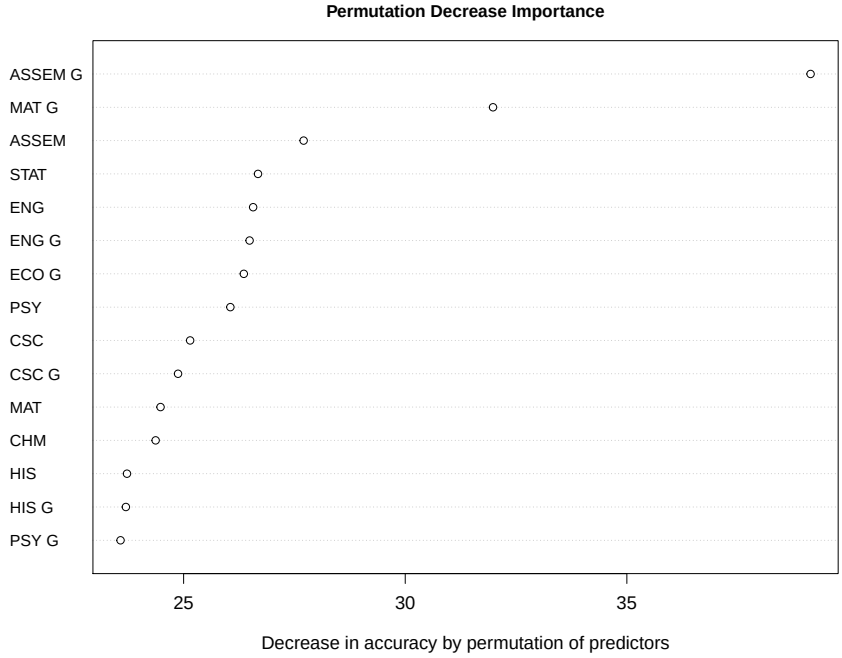

Figure 4: Variable importance plot produced by the RandomForest package for the **random forest # 3**. The 15 most important predictors are displayed. The importance of a predictor is determined by the average decrease in accuracy in the test set caused by a random permutation of the predictor.

of credit in that department. Grades from low-grading departments are always more important than their counterpart and in some cases, they were the only significant grades. A possible explanation is that the grade inflation that suffered the high-grading departments caused the grades to be no longer a reliable tool to distinguish students among themselves. Therefore, universities could use such technique to verify if grades in a department are more important than grades in other departments and act accordingly. The first year seminar courses (ASSEM) were brand new at the University of Toronto and the analysis performed provided evidence of the merit of such courses in order to establish a student's profile and to predict success. In other words, such analysis could help university administrations assess the usefulness of new programs and courses.

4.2 Second research question : Predicting the major

The second task at hand is to build a random forest that predicts the student's major. For the first implementation, **random forest # 1**, the algorithm predicts the correct major completed for 46.31% of the students

in the test set. This appears slightly lower than expected, but considering there are 71 different programs, being able to pin down the right program for about half of the students seems successful. Once again, the variable importance boxplots will be computed and looked at.

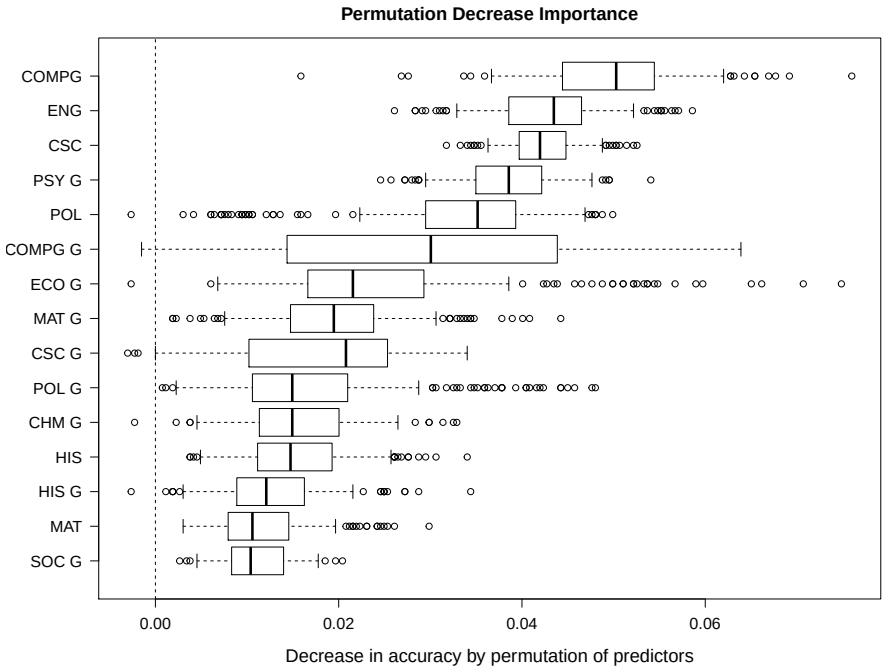

Figure 5: Variables importance boxplots for the **random forest # 1**. The 15 most important predictors are displayed. The importance of a predictor is determined by the average decrease in accuracy in the test set caused by a random permutation of the predictor.

A decrease in importance for the grades variable is noted in Figure 5. This was to be expected because of how the data was formatted. Since the department in which the highest amount of credit was obtained is considered the major completed by the student, these variable importance measures aren't surprising. Actually, if all the courses were included, instead of only the first year, the amount of credit in every department precisely defines the response variable. Considering this weakness in the data formatting, the grades still have a relatively high importance.

Among the 4 most important grade variables, Mathematics, Economics and Finance are found, indicating the high importance of these grades, not only to predict if a student will succeed at a program or not, but to predict what they will major in. The three most important variables are the number

of credits in the Finance department, the English (ENG) department and in the computer science (CSC) department. In the training set, almost all students that attended courses in those departments obtained their major in that very same department. Therefore, these three predictors are really important to classify the students that belong in these respective departments, while grades related variable affect more students but with a lesser importance.

Next, a random forest with random inputs is built as a classifier for the major completed. A 44.57% accuracy was achieved by the **random forest #2** on the test set. Figure 6 contains the 15 largest mean decreases in accuracy after permuting the predictors.

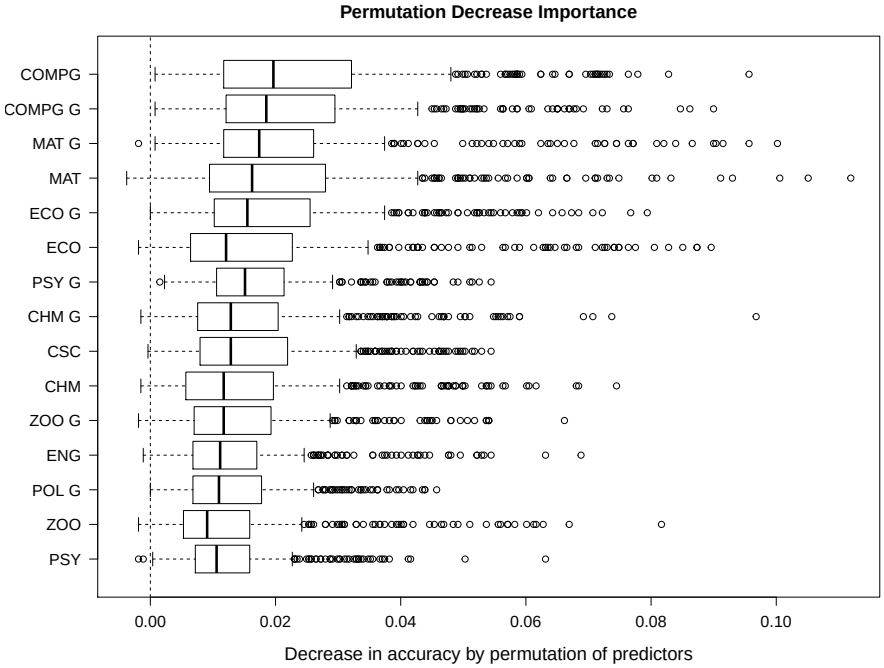

Figure 6: Variables importance boxplots for the **random forest # 2**. The 15 most important predictors are displayed. The importance of a predictor is determined by the average decrease in accuracy in the test set caused by a random permutation of the predictor.

In figure 6, grades in Finance, Mathematics, Economics are the three most important grades. Once again the grade in Psychology (PSY) is the most important grade coming from a department considered high-grading by Sabot & Wakeman-Linn. Like in Figure 3, it seems like the random forest

with random inputs produces variable importance measurements with larger variance making it harder to order the predictors with certainty.

Using the R RandomForest package to produce the **random forest # 3**, a 47.41% accuracy in predicting the major completed is achieved. Once again, one time out of two, the algorithm selects the correct program within the list of 71 programs. Figure 7 contains the variable importance plot produced by the package.

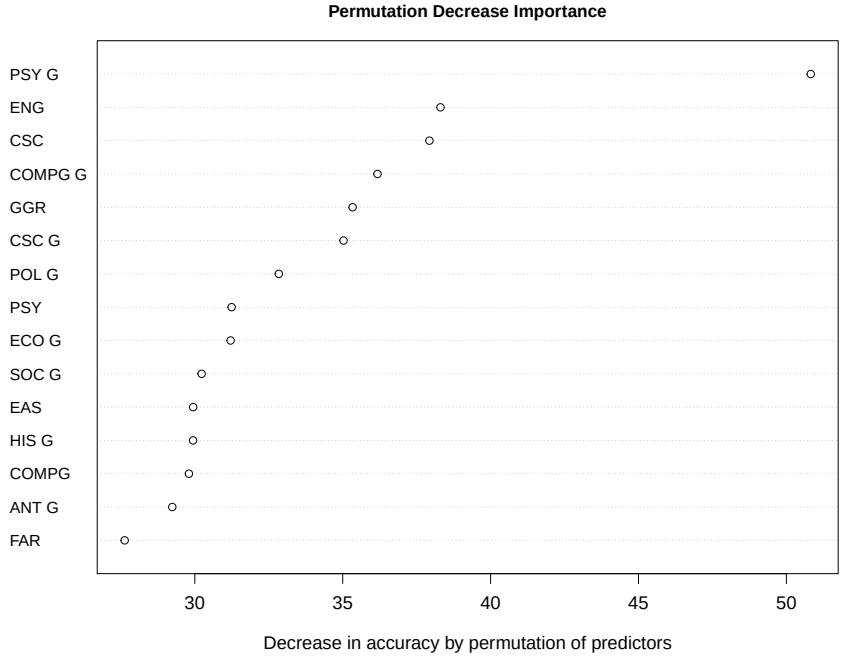

Figure 7: Variable importance plot produced by the RandomForest package for the **random forest # 3**. The 15 most important predictors are displayed. The importance of a predictor is determined by the average decrease in accuracy in the test set caused by a random permutation of the predictor.

Figure 7 contains surprising results. Grades in Psychology is the most important variable while grades in Economics falls low and grades in Mathematics is not even part of the top 15. As in figure 5, the high importance of the number of credits attempted in the English department and in the Computer Science department is observed. Once again, the lack of boxplots or any other detail regarding the variance makes this plot a bit less unreliable.

Overall, the results are less convincing for the prediction of the students' major. A linear multinomial model was fit using the same set of predictors

and its accuracy is 42%. The three different forests had a prediction accuracy located around 46% which is a slight increase compared to the linear model. Some potential improvements regarding these classifiers are located in the conclusion.

These classifiers could help individual departments predict the number of students registering to second, third or fourth year courses and graduate programs. Predicting the major could also help university administrations split the financial resources among the departments or decide the programs that require more advertisements.

The variable importance analysis is once again useful. It was noted that across the three models the grades are slightly less important than they were for the completion prediction. Even though the grades in Mathematics and Economics are still among the most important they are a bit less important than in the previous set of classifiers. Once again it seems that grades in low-grading departments are more useful in order to establish a student's profile. Finally, it seems like for some departments, such as English and Computers Science, it is easy to predict students that will complete a major in those departments by almost solely looking at the number of courses attempted in those departments during the first year.

5 Conclusion

The first year's worth of courses and grades were used to build two classifiers; one that predicts if a student will complete their undergraduate program, the other that predicts the major of a student who completed a program. Random forests were used to build those classifiers. The classifiers can be used for many purposes. For example, to predict the number of students registered in second year courses, to predict the distribution of students across the many programs or to identify students at risk of failing or dropping out.

The importance of each predictor was also evaluated. It was observed in Section 4 that grades were important for predicting if a student will complete their program. Grades in departments that were considered low-grading departments in some grades inflation research articles like Mathematics, Economics, Finance, Biology and Chemistry are consistently among the most important variables. Grades in Psychology were also considered important in a lot of situations. These results indicate that a strong relationship exists between the grades in low-grading departments and the chance of succeeding at an undergraduate program, although this does not necessarily indicate a *causal* connection. Grades were somewhat less important predictors for predicting the students' major but even though they were less important, grades

in Mathematics, Finance, Economics and Psychology were still frequently significantly important.

Finally, for potential improvements in the data analysis, it is to be noted that some students might have completed more than one major or specialization. This might explain the relatively low accuracy for major choice prediction. Allowing for multiple major choices is a potential improvement for this model. This is in fact a multi-label classification problem and some solutions have already been proposed to adapt decision trees to accommodate this more complicated problem ([10], [8], [9]). Some departments also share a great deal of similarities and might be considered equivalent by the university, thus combining some of them might increase our prediction accuracy. The missing values in the predictors were also problematic. Ideally, the algorithm would consider splitting on the grade variables for a certain department only to classify students who took courses in that department. Developing a new decision tree algorithm where new variables are added to the pool of potential split variables depending on previous partitionings should be a great way to improve the actual model in certain scenarios. Overall, implementing a new tree-building procedure where variable are added or discarded based upon previous partitionings and considering a multi-label classifier like suggested by Chen & al. [8] could be great improvements for future work on that data set.

Acknowledgement

We are very grateful to Glenn Loney and Sinisa Markovic of the University of Toronto for providing us with students grade data. The authors also gratefully acknowledge the financial support from the NSERC of Canada.

References

[1] L. Aulck, N. Velagapudi, J. Blumenstock, and J. West. Predicting Student Dropout in Higher Education. *ArXiv e-prints*, June 2016.

[2] M. A. Bailey, J. S. Rosenthal, and A. H. Yoon. Grades and incentives: assessing competing grade point average measures and postgraduate outcomes. *Studies in Higher Education*, 41(9):1548–1562, 2016.

[3] T. Bar, V. Kadiyali, and A. Zussman. Grade information and grade inflation: The cornell experiment. *Journal of Economic Perspectivs*, 23(3):93–108, 2009.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] L. Breiman. Heuristics of instability and stabilization in model selection. *Ann. Statist.*, 24(6):2350–2383, 12 1996.

[6] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

[8] Y.-L. Chen, C.-L. Hsu, and S.-C. Chou. Constructing a multi-valued and multi-labeled decision tree. *Expert Systems with Applications*, 25(2):199 – 209, 2003.

[9] S. Chou and C.-L. Hsu. MMDT: A multi-valued and multi-labeled decision tree classifier for data mining. *Expert Syst. Appl.*, 28(4):799–812, May 2005.

[10] A. Clare and R. D. King. *Knowledge Discovery in Multi-label Phenotype Data*, pages 42–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[11] D. Eddelbuettel and R. Francois. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(1):1–18, 2011.

[12] J. Glaesser and B. Cooper. Gender, parental education, and ability: their interacting roles in predicting gcse success. *Cambridge Journal of Education*, 42(4):463–480, 2012.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

[14] T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.

[15] S. R. Johnson and F. K. Stage. Academic engagement and student success: Do high-impact practices mean higher graduation rates? *The Journal of Higher Education*, 0(0):1–29, 2018.

[16] V. E. Johnson. *Grade Inflation : A Crisis in College Education*. Springer, 2003.

[17] R. Kappe and H. van der Flier. Predicting academic success in higher education: what's more important than being smart? *European Journal of Psychology of Education*, 27(4):605–619, Dec 2012.

[18] H. Kim and W.-Y. Loh. Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96:589–604, 2001.

[19] I. Kononenko. On biases in estimating multi-valued attributes. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1034–1040, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[20] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[21] W.-Y. Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386, 2002.

[22] W.-Y. Loh and Y.-S. Shih. Split selection methods for classification trees. *Statistica Sinica*, 7:815–840, 1997.

[23] J. S. Mills and K. R. Blankstein. Perfectionism, intrinsic vs extrinsic motivation, and motivated strategies for learning: a multidimensional analysis of university students. *Personality and Individual Differences*, 29(6):1191 – 1204, 2000.

[24] A. S. M. Niessen, R. R. Meijer, and J. N. Tendeiro. Predicting performance in higher education using proximal predictors. *PLOS ONE*, 11(4):1–14, 04 2016.

[25] B. Ost. The role of peers and grades in determining major persistence in sciences. *Economics of Education Review*, (29):923–934, 2010.

[26] R. Sabot and J. Wakeman-Linn. Grade inflation and course choice. *Journal of Economic Perspectives*, 5:159–170, 1991.

[27] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):25, 2007.

[28] University of Toronto. Degree requirements (h.b.a., h.b.sc., bcom), 2017.

A Appendix

The following section contains some mathematical notations and definitions for readers who are interested in more a thorough explanation of sections' 3.2 and 3.3 content. Full understanding of the appendix is not needed in order to grasp the essential of the article but it serves as a brief but precise introduction to the mathematical formulation of decision trees and random forests.

Rigorously, a typical supervised statistical learning problem is defined when the relationship between a response variable $\mathbf{Y}$ and an associated m-dimensional covariate vector $\mathbf{X} = (X_1, ..., X_m)$ is of interest. When the response variable is categorical and takes k different possible values, this problem is defined as a k-class classification problem. One challenge in classification problems is to use a data set $D = \{(Y_i, X_{1,i}, ..., X_{m,i}); i = 1, ..., n\}$ in order to construct a classifier $\varphi(D)$. A classifier is built to emit a class prediction for any new data point $\mathbf{X}$ that belongs in the feature space $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_m$. Therefore a classifier divides the feature space $\mathcal{X}$ into k disjoint regions such that $\cup_{j=1}^{k} B_l = \mathcal{X}$, i.e. $\varphi(D, \mathbf{X}) = \sum_{j=1}^{k} j\mathbf{1}\{\mathbf{X} \in B_j\}$.

As explained in section 3.2 a classification tree [7] is an algorithm that forms these regions by recursively dividing the feature space $\mathcal{X}$ until a stopping rule is applied. Most algorithms stop the partitioning process whenever every terminal node of the tree contains less than β observations. This β is a tuning parameter that can be established by cross-validation. Let p_{rk} be the proportion of the class k in the region r, if the region r contains n_r observations then :

$$p_{rk} = \frac{1}{n_r} \sum_{x_i \in R_r} \mathbf{1}\{y_i = k\}. \tag{1}$$

The class prediction for a new observation that shall fall in the region r is the majority class in that region, i.e. if $\mathbf{X} \in R_r$, $\varphi(D, \mathbf{X}) = \text{argmax}_k(p_{kr})$. When splitting a region into two new regions R_1 and R_2 the algorithm will compute the total impurity of the new regions ; $n_1Q_1 + n_2Q_2$ and will pick the split variable j and split location s that minimizes that total impurity. If the covariate j is continuous, the possible splits are of the form $X_j \leq s$ and $X_j > s$ which usually results in $n_r - 1$ possible splits. For a categorical predictor having q possible values, we usually consider all of the $2^{q-1} - 1$ possible splits. Hastie & al. [13] introduces many possible region impurity measurements Q_r, in this project, the *Gini index* has been chosen :

$$Q_r = \sum_{j=1}^{k} p_{rj}(1 - p_{rj}). \tag{2}$$

Here is a pseudo-code of the algorithm :

Algorithm : DT(D,β)

1. Starting with the entire data set D as the first set of observations r.
2. Check ($n_r > \beta$).
3. **if** (false) :
   Assign a label to the node and exit.
   else if :
   for (j in all predictors):
   for (s in all possible splits) :
   Compute total impurity measure.
   Select variable j and split s with minimum impurity measure and split the set r into two children sets of observations.
   Repeat steps 2 & 3 on the two resulting sets.

Since decision trees are unstable procedures [5] they greatly benefit from bootstrap aggregating (bagging) [4]. In classifier aggregating, the goal is to find a way to use an entire set of classifiers $\{\varphi(D_q)\}$ to get a new classifier φ_a that is better than any of them individually. One method of aggregating the class predictions $\{\varphi(D_q, \mathbf{X})\}$ is by *voting*: the predicted class for the input $\mathbf{X}$ is the most picked class among the classifiers. More precisely, let $T_k = |\{q : \varphi(D_q, \mathbf{X}) = k\}|$ then, the aggregating classifier becomes $\varphi_a(\mathbf{X}) = \text{argmax}_k(T_k)$.

On way to form an ensemble of classifiers is to draw bootstrap samples of the data set D which forms an ensemble of learning sets $\{D_B\}$. Each of the bootstrap samples will be of size n drawn at random with replacement from the original training set D. For each of these learning set a classifier $\varphi(D_b)$ is constructed and the resulting set of classifiers $\{\varphi(D_b)\}$ can be used to create an aggregating classifier. If the classifier is an unpruned tree then the aggregating classifier is a random forest.

A random forest classifier is more precise than a single classification tree in the sense that it has lower mean-squared prediction error [4]. By bagging a classifier, the bias will remain the same but the variance will decrease. One way to further decrease the variance of the random forest is by construction trees that are as uncorrelated as possible. Breiman introduced in 2001 random forests with random inputs [6]. In these forests, instead of finding the best variable and partitioning among all the variables, the algorithm will now randomly select $p < m$ random covariates and will find the best condition among those p covariates.