AccuCampus

RISK SCORING EXPLAINED

Abstract

The Risk Scoring system is a key tool in AccuCampus that helps detect students at risk of dropping out. Using AI technology, AccuCampus analyzes student data from its database and from other sources to try to detect patterns and keep staff members in contact with those students.

In this document, we review how Risk Scoring works in-depth. We will look at what the recommended configurations are for it and its limitations.

Contents

Introduction

The risk scoring system in AccuCampus is an internal system that helps detect which students might be at risk. It uses advanced AI methods to detect what factors or combination of factors contribute to an increased or decreased risk of attrition for each student.

Assessing risk is difficult, subjective, and requires a professional, multidisciplinary, in-depth analysis of the situation of each student. Furthermore, students can also have a high risk of dropping out as a result of psychological, environmental, or economic factors.

Professional risk assessment is, thus, the best tool we have. It is an excellent response in certain situations but does not scale: it is impossible to assess the personal situation of thousands of different students in a timely fashion.

To analyze thousands of students, we need to use proxy indicators of what the situation of each student could be. In addition, while risk scoring is not perfectly accurate, it can help reduce the number of students that need further analysis. Here is where the AccuCampus risk scoring system comes in. In this guide, we explain how it works, how to set it up, its limitations, and the best practices when using it.

The process uses some of the concepts described in the paper *"A comparative analysis of machine learning techniques for student retention management"* (Delen, 2010) and other standard AI techniques adapted to student risk analysis.

How it Works

The Risk Scoring system works by analyzing all the information that is in your AccuCampus account and tries to detect meaningful patterns for those students who have dropped or continued to enroll term after term.

There are several steps taken in the process: first, AccuCampus sends the data to the risk-scoring engine; then it undergoes a transformation to a usable form on which we can perform statistical analysis; and finally, it trains the AI to work with the institution's data. Once this process trains the AI, it is now usable for prediction. Below is a detailed description of how each step works.

Data Extraction

Let us start by understanding what data you would send through the system. For each semester, we collect basic student information, records of student visits to each service, and all the extended profile information imported into the system. Other user actions are recorded indirectly: appointments, for example, are recorded when the student actually visits the center or attends an online session for that appointment.

This has two advantages over other systems. First, since it includes student visits, it is dynamic and constantly evolving; a student with a high risk score today can have a lower score tomorrow if an instructor created an alert for the student and he visited one of the institution's centers. Second, we can further extend the system's capabilities since we can extract data from other systems and add it to the students' profiles.

Data Pre-Processing and Analysis

When we setup a "model" in AccuCampus, it internally extracts all that historical information and does some calculation to find patterns.

The initial analysis is descriptive; AccuCampus will show a graph of the distribution of each field analyzed and its values. See "Setup and Configuration" for examples on how this can help improve the results.

Before the analysis, it is necessary for the system to do some clean up. With all the information compiled in the same place, several transformations take place internally to adapt the data so that the AI algorithm better understands it in the future. These transformations include:

- **Conversion of Text Fields:** "Freshman", "Sophomore", "Junior" and "Senior" are just letters from a computer point of view. To give sense to that type of information, the system groups all the students based on those qualitative attributes. In statistical terms, it creates dummy variables[1] to process such attributes.

- **Missing Values Treatment:** Sometimes information is not complete, but it can still be useful. When that happens, the system might decide to compensate the absence of information with some internal estimates for those specific students. This way, AccuCampus can make a more accurate analysis even with incomplete data.

- **Fixing Skewed Data and Normalization:** To aid the system and allow for better understanding of what the numeric values mean, it is necessary to see where they fall in comparison with the rest of the data. This way, all numbers are more representative of their intrinsic meaning and comparable across the entire data set. These changes are necessary to allow a better statistical analysis and produce a more accurate output. We use advanced statistical transformations like Box-Cox[2] to achieve this.

[1] Dummy variables, more information here: https://www.statisticssolutions.com/dummy-coding-the-how-and-why/

[2] Box-Cox Transformation, more information here: https://towardsdatascience.com/box-cox-transformation-explained-51d745e34203

Model Building and Selection

At this point, we have our data set with historical information analyzed, selected, transformed and ready to insert into our AI engine. Not all institutions are the same though: they may have a different way of importing the information, their students could have different backgrounds from a typical institution, and their centers may offer different services with different results in each case.

> "A **MODEL** IS THE OUTPUT OF AN ALGORITHM THAT HAS BEEN TRAINED TO RECOGNIZE CERTAIN TYPES OF PATTERNS. YOU TRAIN A MODEL OVER A SET OF DATA, PROVIDING IT AN ALGORITHM THAT IT CAN USE TO REASON OVER AND LEARN FROM THOSE DATA" (Microsoft, 2019)

The goal of the AccuCampus risk scoring engine is to find patterns in the institution's data, understanding the specific characteristics of each institution. Current AI technology is not a one-size-fits-all and AccuCampus is no exception.

To find the best model, AccuCampus tries dozens of different standard algorithms with different parameters to see which one adjusts best. Currently, five different algorithms are available[3]: Decision Trees, Logistic Regression, Neural Networks, Random Forest, and Support Vector Machine (SVM). For each of those algorithms, many adjustments are made, resulting in a wide range of possible models.

Once we build the potential models, it is necessary to see which one adjusts best to the data. However, that is not a straightforward task; if the model adjusts too much to the historical data then it loses its forecasting power. For example, the GPA might improve the accuracy of the prediction, but knowing that students with specific card numbers dropped that last year will only help for the historical data and has no prediction power. The opposite is also possible when the model is too generic. The resulting issues that arise are "overfitting" and "underfitting" respectively[4].

In order to select the best model, we need to test them all and see how they perform. Described in basic terms, AccuCampus splits the data in 2 parts: with the first and largest data set, the system tries to understand what the relevant attributes are and find patterns. It uses the second data set to analyze that model's prediction power.

Evaluating the models against our control data, we use an indicator called AUC-ROC: "*The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes...*" (Bhandari, 2020).

An explanation of how to understand the AUC metric is found in Understanding the AUC-ROC Result.

Prediction and Feedback Path

Now we run the model selected by the system against the current students. A calculated risk score is now available for each student using this model. AccuCampus allows users to setup reports and automatic alerts based on the risk score.

[3] Some algorithms might not always run depending on the structure or size of the data being analyzed.
[4] Overfitting and underfitting explained: https://medium.com/@sanidhyaagrawal08/what-is-overfitting-in-machine-learning-ab519d921610

When faculty create an action plan using the risk scoring, this creates a feedback path for the model. A student with a high score might be sent to counseling; going to counseling might reduce their chances of dropping out; not dropping out is a good signal and will be used by the prediction model to adjust the student's risk score.

This feedback path makes the system evolve and adjust to new changes, for the good or for the bad, in the way centers work with their students.

Setup and Configuration

To start using the risk-scoring module in AccuCampus, go to the *Institutional Research* item on the menu and then click on *Risk Scoring*. Click on *Create your first model* to get started. Give it a name, select the features to exclude and select the correct semester order.

Preparing the Data

Date Handling

The AI engine is prepared to handle text and numbers. Dates, however, we need to give special attention as they can have multiple meanings. For instance, converting dates to "time since today" is strongly recommended. Here are some examples:

- **Birthdates:** Instead of entering the birthdate as "12/15/2000" or "December 15, 2000", we can enter the age of the student as a number.

- **High School Graduation year:** While this can be included as a number (i.e. "2019"), it will likely be more useful to include as 'years since high school graduation'. As an example, if a student enters college in 2017 who graduated high school in 2015 (2 years prior) and another student enters college in 2020 who graduated in 2018 (also 2 years prior), then in both cases, the student graduated high school 2 years before they entered college. The factor is now better quantified as 'the number of years (or gap) between high school graduation and college entry' and should be entered as "2".

The exception is when the date or year is a different qualitative group and, in that case, keeping the date as a text field will be fine for that behavior.

Please note, when fields include text and/or slashes and dashes, they will be treated as text by the system rather than a number in the analysis.

Number Handling

Numbers typically are used by the models correctly and do not require any special treatment. The only exception is when the numbers used are contextual groups.

The use of zip codes is a good example of this special type of number handling. While there can be some relationship between the zip code of a person and their likelihood to drop out, the link is not because of its number value. In other words, a zip code starting with 90 does not make the student 9 times more/less likely to drop out than a student with a zip code starting with 10, or 3 times more or less likely than one with a zip code starting with 30.

In those cases, it is better to change the zip code to another format, such as “zip60630” or “zip606” to have larger groups.

Semester Order

AccuCampus automatically selects the semester order based on the dates. Our recommendation is to remove semesters that are optional for the students to attend, such as summer semesters. The system expectation is that students will move from one semester to the other until graduation and it would consider a student as a dropout if they were not in a semester included.

Grouping Similar Attributes

Having too much detail might look like noise. Whenever possible if there are hundreds or thousands of possible options for one field then we recommend grouping or using an exclusion. We noticed that Grouping is an option in most cases. For example, specific street addresses are not helpful; instead, grouping by zip code, city, or distance to campus will be more useful.

Detecting Early Issues

During the initial analysis we can detect cases where one specific attribute needs to be corrected or removed, some examples include:

1) **Data imported incorrectly:** Looking at the graphs, we easily spot issues in our import process. If most or all students have an attribute and we, from experience, know that including this confounds the analysis, we can review the data being imported, identify possible issues, and propose modifications.

2) **Attributes with extremely limited data:** It is possible that the information loaded into the system is incomplete for one attribute. In that case, we find explicitly excluding this data helps prevent the system from considering it and causing issues.

3) **Attributes with irrelevant information:** There is information that is in no way relevant to estimate whether a student can be at risk or not. For example, card numbers are not an indicator of success and can be excluded from the risk assessment. While the system will not be able to find any pattern from there, it is better to keep only what could be relevant to reduce the analysis time.

4) **Sensitive information:** In some cases, the use of certain attributes can improve the accuracy of the results but also have important ethical and even legal consequences. This is usually the case with race or ethnicity fields, which work as a proxy of other environmental factors that are not part of the calculation. It is up to the institution to decide whether they should be included in the analysis or not.

Feature Importance

After the analysis, AccuCampus will report the 'feature importance (estimate)' along with descriptive graphs for each attribute.

Due to the complexity of the information, it is not possible to say what is important and what is not. That is simply because it is usually a combination of multiple factors that increases or reduces the risk.

However, AccuCampus runs a statistical regression to have an estimation of the importance of each attribute. If an attribute has high importance or relevance compared to others, it means it will likely affect the risk score, for better or worse.

Because of the limitations mentioned above, it is possible that an attribute with high importance does not affect the score for a specific student.

Building the Model

Once the analysis is completed and the information is corrected, the model can be built. The AI engine will start working in the background to try to understand the patterns hidden in the institution's data.

This process can take several hours. Once complete, it will also calculate the current scores for the students and provide an indicator of how good the results are, shown as "Quality".

Understanding the AUC-ROC Result

There is no need to understand how the AUC-ROC result works mathematically to use it in AccuCampus. However, it is useful to know whether the model is accurate or not. The score is a number that goes from 0 to 1, with a number closest to 1 as most desirable. If the score is 0.5 it will work as well as flipping a coin; a score of 0.75 or higher is desirable.

This is how the scale works:

- **AUC-ROC lower than 0.6:**
  The model has very little prediction power and needs adjustment.

- **AUC-ROC between 0.6 and 0.7:**
  The model has a 'fair' prediction power. Either the model needs more data, or improvements are needed to the existing data.

- **AUC-ROC between 0.7 and 0.8:**
  The model will produce helpful results. It is always possible to try other combinations and options, but AccuCampus will be able to identify students at risk fairly well.

- **AUC-ROC between 0.8 and 0.9:**
  The model will produce largely accurate results, with less false positives and less false negatives.

- **AUC-ROC higher than 0.9:**
  The model is excellent. This is score ideal but is unlikely achievable since the system is analyzing human behavior. Moreover, this is difficult for any system to understand.

Best Practices

We recommend following these best practices for optimal results:

1) Make the changes described in Preparing the Data before importing the students' information.

2) Review the list of common issues under Detecting Early Issues. Exclude attributes with 500+ different options.

3) Use AccuCampus extensively over 3 full semesters before building the model. This will give the AI engine enough data to learn from it.

4) If the AUC-ROC value is lower than 0.75, then consider reviewing the data or disabling the AI engine.

5) Upload key grades in the students' profiles. It is not necessary to upload all the grades, but uploading key ones, or an average of the student's grades, or the values of the lowest and highest grades will likely help understand how the student is doing.

In addition, as always, reach out to the Engineerica implementation team for questions and help on setting up the Risk Scoring system.

Limitations

There are some known limitations in the AccuCampus' risk scoring system. The first and most obvious one is that AccuCampus does not know the individual situation of each student; the analysis is being done based on the information available and there are several factors that cannot be represented within a database. The understanding of the student situation will only be as good as the information available.

The engine currently works to detect whether the student will drop out or not. The engine design was not for nor can it predict a specific outcome on the performance in a specific course or in specific learning areas.

Furthermore, information from a few semesters is required to begin training the model in AccuCampus. The longer the institution uses AccuCampus, the more accurate the prediction will be. Information from action items and student activity on the AccuCampus website is not currently affecting the risk score.

About Engineerica

Engineerica Systems, Inc. is a Florida Corporation founded in 1994 by College of Engineering and Computer Science UCF alumni. Engineerica's flagship product, AccuTrack, has been in use by academic institutions since 1998.

Currently, Engineerica offers several attendance tracking systems including desktop software, client-server applications, cloud-based solutions, Apple iOS and Android apps. These systems include academic center management software, classroom attendance applications, conference, and event tracking systems, and more.

Engineerica's strives to provide the best possible products and services to its clients. This focus on high quality products and services is the driver behind its continuous growth.

www.engineerica.com