**Dropout prediction**

**LABEL ENCODING:** [**https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd**](https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd)

[**https://machinelearningmastery.com/calculate-feature-importance-with-python/**](https://machinelearningmastery.com/calculate-feature-importance-with-python/)

[**https://towardsdatascience.com/interpreting-coefficients-in-linear-and-logistic-regression-6ddf1295f6f1**](https://towardsdatascience.com/interpreting-coefficients-in-linear-and-logistic-regression-6ddf1295f6f1)

[**https://ema.drwhy.ai/featureImportance.html**](https://ema.drwhy.ai/featureImportance.html)

[**https://towardsdatascience.com/a-simple-interpretation-of-logistic-regression-coefficients-e3a40a62e8cf**](https://towardsdatascience.com/a-simple-interpretation-of-logistic-regression-coefficients-e3a40a62e8cf) **(USEFUL EXAMPLE)**

**https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-interpret-odds-ratios-in-logistic-regression/**

**https://stackoverflow.com/questions/48508127/how-to-get-coefficients-of-multinomial-logistic-regression**

[**https://machinelearningmastery.com/multinomial-logistic-regression-with-python/**](https://machinelearningmastery.com/multinomial-logistic-regression-with-python/)

**Dropout definition**

I defined three types of dropout behaviour:

1. Early dropout i.e. dropped out within the first three weeks
2. Late dropout i.e. dropped out within the last two weeks
3. No dropout i.e. attempted the last module

**Dropout label assignment process**

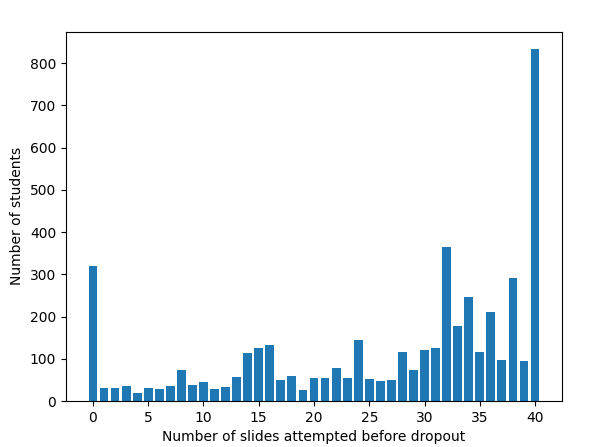
**Si = 00001010010101010 PF 001001001 NP .... 0101010101 NNNNNNNNNNNNNN**

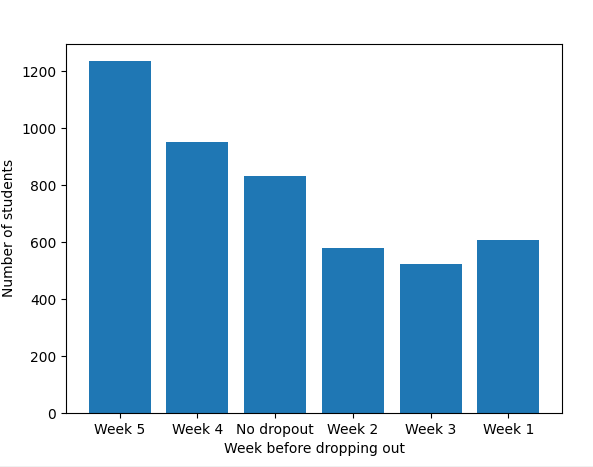
**Pi = PPFPFPFPFPFPNNNNNNNNN**

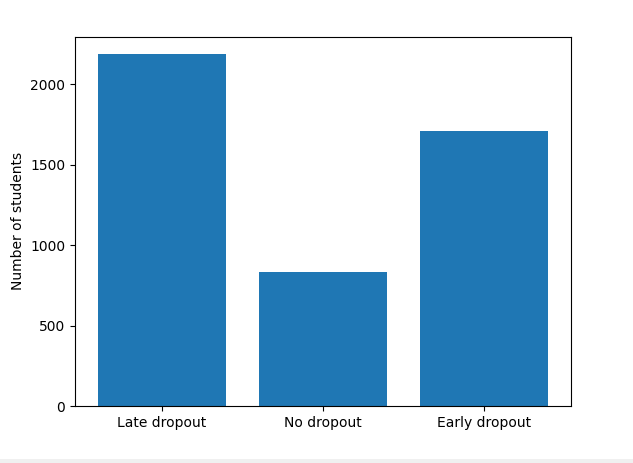
The procedure I used to classify a student’s dropout behaviour is as follows:

1. Generate the interaction sequence for the student across all modules (i.e. for the entire challenge). This interaction sequence will contain a series of 0s and 1s indicating slide completion, as well as Ns, Fs and Ps indicating problem slide interactions.
2. Calculate a new interaction sequence by removing all 0s and 1s in the interaction sequence , leaving behind the problem slides only.
3. Compute the quantity , which represents the number of problem slides attempted before the student disengages from the course (which is represented by trailing Ns). Note that where is the number of trailing Ns in the sequence
4. Using the quantity , calculate the week that the student dropped out using slide number thresholds e.g. if the student completed at most 8 slides before dropping out, or , then the student dropped out in Week 1. This is because each week has two modules, each with 4 problems, making 8 problems in total. If , then note this as a no-dropout student.
5. Using the quantity , determine the dropout behaviour for the student (i.e. no dropout, early dropout or late dropout), noting the rule that early dropouts dropout in weeks 1-3, late dropouts dropout in weeks 4-5, and no dropouts do not drop out at all.

**Distribution of dropout outcomes**







**Density plot instead of scatter plots**

**Dropout classification**

1. Six features were identified (shown in the table below) and extracted. The extraction process is straightforward – simply count the number of Ps or 1s in the interaction sequence, or a subset of the sequence representing the first X weeks of the challenge, where .
2. The output labels were extracted using the process described in an earlier section.
3. Logistic regression classifier was used with a 33/67 split for testing and training data. Note: emphasise that it was randomised – use 10-fold cross-validation with stratification.

Really used one feature in the logistic regression model. Be clear. Need to calculate the correlation between the features and variables. Numbers are good – just one feature. Distribution of three classes.

3 classes: for each example you’re predicting the majority class – 0R classifier. Look at the training data, if you always predict late dropout. Compare to 0R to make sure that it is lower than 76%.

|  |  |
| --- | --- |
| **Problem completion features** | **Score** |
| Number of problems completed in the first week of the challenge (# of Ps in subset of sequence). | 0.76 |
| Number of problems completed in the first two weeks of the challenge (# of Ps in subset of sequence). | 0.76 |
| Number of problems completed in the first three weeks of the challenge (# of Ps in subset of sequence). | 0.82 |
| Number of slides completed in the challenge (# of Ps in sequence) | 0.89 |
| **Slide completion features** | |
| Number of slides completed in the first week of the challenge (# of 1s in subset of sequence). | 0.76 |
| Number of slides completed in the first two weeks of the challenge (# of 1s in subset of sequence). | 0.76 |
| Number of slides completed in the first three weeks of the challenge (# of 1s in subset of sequence). | 0.78 |
| Number of slides completed in the challenge (# of 1s in sequence) | 0.80 |

**Conclusion 1:** This shows that simply using a student’s interactions for the first week (i.e. the interaction sequence up to the first week), we can predict what type of dropout behaviour they will exhibit (i.e. early, late or no dropout) with considerable accuracy that is comparable to if we had taken the entire interaction sequence rather than a subset.

This also shows that considering more data, such as the first two and three weeks of the challenge or even the whole challenge (i.e. whole interaction sequence), does not necessarily lead to a significant increase in accuracy. What this means is that student dropout behaviour is highly determined by their slide interactions in the first week of the challenge.

**Conclusion 2:** Problem and slide completion both offer similar performance. These are likely to be correlated attributes – so worth exploring.

Predicting the dropout looks very promising – compared to predicting the score. Score is not a particularly useful measure. It is better to predict dropout than how student will perform in particular case.

**Other things to predict:** How many attempts taken to complete a problem. More useful than final score. Potentially, if you’re going to do score, then you can do discretise. Dropout is better thing to be pulling out. Interesting that first week is such a strong predictor of outcomes for the course.

**TO DO:** Look at how this varies for different datasets. Advanced might not be so useful (small course), but intermediate would be interesting to look out.

What additional features can be used to improve prediction:

**Overall:** Positive results, potentially publishable. Aim to submit a paper (conference deadline is in October. You can attach the paper to the thesis as an appendix. In your contribution, you say that you submitted a paper in this conference based on your results – will make good impression on examiner. **Check examiners.**

**Things to do**

* **Send email about thesis examiners and plan out rest of the weeks.**
* **Code clean-up and results**
  + Clean-up and commit all code to Git. Ensure that it is commented neatly.
  + Save all results, and label them appropriately into a results folder, as well as relevant subfolders if needed.
* **Continue with dropout prediction**
  + Test against the majority class (0R classifier).
  + Do a proper evaluation – look at different evaluation metrics.
  + Perform 10-fold cross validation with stratification to ensure results are robust.
* **Continue with outcome prediction – but discretise the scores.**
* **Focus on writing things out** 
  + Realise where you went wrong, and things that you might have missed.
  + **Start writing out the discussion part of the thesis now:** Add in the background/theory/intro/comparisons with similar work and context later **– start writing out the discussion part of the thesis now.**
* **Slide value importance**
  + Focus only on the slides that matter (we have this information from behaviour – also attach this to the appendix), and extract features. Compare performance (dropout and final performance) with the case of considering all slides.
  + Narrow down on specific slides and combinations of slides, and groups of slides (e.g. weeks like all slides in w1p1, and type of slides e.g. last slide of every module). Compare performance with each other.

**Other things to do:**

* **Extract time-related features:** look at Sophia’s thesis for inspiration.
* **Active vs passive features**
  + **Active features:** Number of submissions made. This also might be harder to extract, so focus on write-up first. This would represent the **effort** exerted by a student.
  + **Passive features:** slide views (put less emphasis on this, and focus on write-up first). This is likely to have quite a lot of noise.
  + **Note:** Making too many submissions make you lose points. Not sure how much students are aware of that (they should be aware, but they are told – some people might not care). There is some structure in place to prevent this.
* **Performance prediction** 
  + **Sequence prediction:** Whether they passed, failed or made no submission to a problem in the future – sequence prediction, if we focus solely on problem slides.
  + **Score prediction:** Score taken out of 400.

|  |  |
| --- | --- |
|  |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |