OurPath: Predicting Whether a User Churns After Six Weeks

By Ryan Singh

25 September 2019

0.1 Introduction

This report will communicate my findings on the most discriminant features within the OurPath data with regards to whether programme users will '*churn*' or not. Predictions for 10 users for which churn status is unknown will also be presented.

Using a very basic a basic Machine Learning model called a Decision Tree, we can identify that the sentiment of user's messages, the number of emojis used along with their height and BMI are amongst the most useful data in predicting whether or not a user will churn.

Ideally, this information will enhance the ability to identify users who are likely to quit. However, I believe more data is needed before we can have full confidence in this model.

0.2 The Data

The Data available consists of basic information on each user such as their height, weight, age etc., their answers to diagnostic questions such as 'What's the main reason you'd like to make a change?'. As well as these, some summary data on the content of their messages, and the different events they had participated in. Some of this data requires manipulation before it may be used in machine learning models. For example, determining the BMI of each user is potentially more useful than height or weight alone (see the technical section for more details).

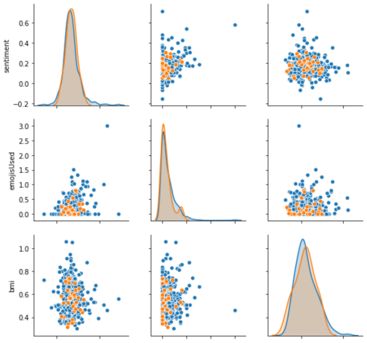


Figure 1: A snapshot of some of the data, orange indicates the user churned. We can see here the churned users seem to be completely interspersed with the non-churned users.

0.2.1 *Technical Details:* Preprocessing and Feature Transformations

There were two main preprocessing challenges with this data. Firstly, the message data had a temporal dimension; a user might send a low sentiment message in week 1 and a high sentiment message in week 6, our model should be able to take into account this change over time. Secondly, we have lots of non-ordinal categorical data i.e. answers to any of the OurPath quiz flow questions, this kind of data usually needs to be transformed before a statistical model can utilise it.

The second challenge has a standard solution, one-hot encoding. In our case this works by creating a new binary feature for each answer to the quiz flow. For example, a specific answer for the motivation questions is 'looks'. One-hot encoding will create a new feature called 'looks' and then for each user who answered 'looks' attribute a 1 and for any other answer attribute a 0. The drawback of this approach is it drastically increases the dimensionality of the dataset.

The first challenge is more subtle and there are a variety of different ways to approach it. The solution used here was to use Linear Regression. We take the sentiment of each message along with the week it was sent, this allows us to find a line of best fit, tracking sentiment over time. Clearly, this approach assumes the sentiment of a user changes in a linear fashion, which may be an over simplification.

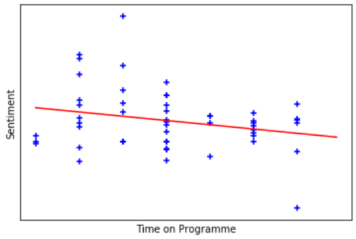


Figure 2: A visualisation of a line of best fit of sentiment over time for one user, as you can see it is a crude approximation, but serves as a heuristic for the users declining sentiment.

Finally, there were some adjustments to features which using my prior knowledge I thought could be beneficial. For example, I created a feature for BMI which can be a better indicator of how a person feels about their weight (since it is scaled by the square of height). After observing a sample of the message frequencies, I also believed that the number of messages users sent after week 3 could be important.

0.3 The Model: Decision Trees

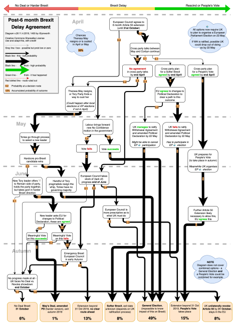


Figure 3: A brexit flow chart, essentially similar to a decision tree.

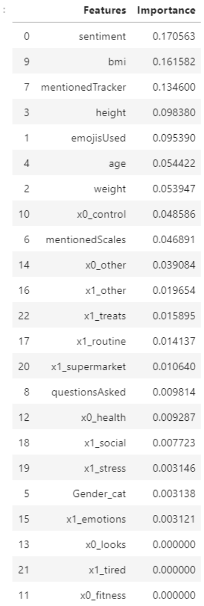
We have all seen a decision tree at some point in our lives, whether in the back of a magazine or in a doctor's waiting room. In a classic decision tree you answer some basic questions about yourself and based on your answers you follow a route through the diagram (see figure 3) ending up at some final category i.e. 'You're a workaholic' or 'You don't need a flu jab'.

A machine learning decision tree, tries to take the data available and form one of these charts, which then allows it to categorise new (unseen) data. Decision trees are considered a 'white-box' method, because they allow you some insight into why the model is behaving the way it is. This is especially useful for us, as it enables us to identify the most important features in the data.

0.3.1 *Technical Details:* GridSearch, Cross Validation and Parameter Tuning

One important aspect of the data is it has a major 'class imbalance', there are around 9 users who remain on the programme for every user who churns. Out of the box decision trees are not great at dealing with class imbalance. Without adjusting for the imbalance, the decision tree algorithm would place little importance on correctly classifying churned users. The algorithm would be 'happy' that it had predicted almost all of the non-churning users correctly. This would be extremely unhelpful, one could easily just predict every single user does not churn, and achieve a 99 percent accuracy.

There are a couple of steps we take to remedy the situation. In the first instance we can adjust a parameter in the algorithm called 'class weights'. In the second we can measure our performance in more helpful ways than naive accuracy. One such



	Features	Importance
0	sentiment	0.170563
9	bmi	0.161582
7	mentionedTracker	0.134600
3	height	0.098380
1	emojisUsed	0.095390
4	age	0.054422
2	weight	0.053947
10	x0_control	0.048586
6	mentionedScales	0.046891
14	x0_other	0.039084
16	x1_other	0.019654
22	x1_treats	0.015895
17	x1_routine	0.014137
20	x1_supermarket	0.010640
8	questionsAsked	0.009814
12	x0_health	0.009287
18	x1_social	0.007723
19	x1_stress	0.003146
5	Gender_cat	0.003138
15	x1_emotions	0.003121
13	x0_looks	0.000000
21	x1_tired	0.000000
11	x0_fitness	0.000000

Figure 4: Initial Feature Importance

performance metric 'f1 score' is known to be particularly good for balancing precision and recall.

In fact we can combine both of these remedies by performing a parameter grid search. We create a grid of possible class weights and execute the algorithm, measuring f1 score on each fold in a 3-fold cross validation split of the test data. This allows us to select the best parameters as measured by the f1 score.

0.4 Results

After learning the above model, we can easily view the importance of each feature, it quickly becomes clear some features have negligible importance (figure 4). For the sake of model clarity these features are dropped from the model, and then the decision tree algorithm is run again.

After the second pass through the data the remaining features and their importance are as below.

Rank	Feature	Importance
1	BMI	0.31
2	M.After 3	0.27
3	Age	0.12
4	(Mean) Emojis Used	0.11
5	Sentiment Coefficient	0.07
6	Sentiment Intercept	0.02

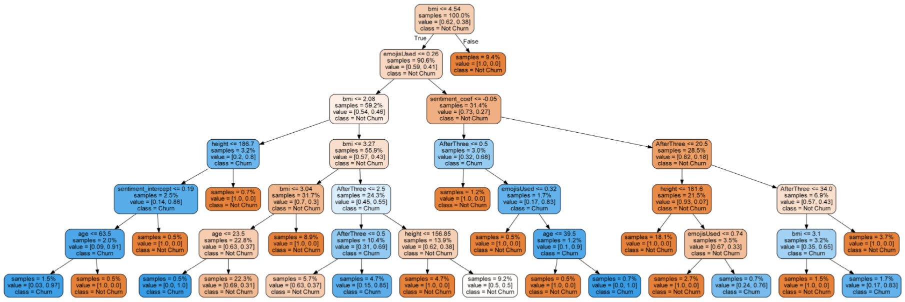


Figure 5: The final decision tree after being trained on the data. The more orange the box is the more likely the user ending up in that box is to churn.

We can use the tree above to predict only user '5cd89558f2f52212d094bd8e'(lets call them Andrea) will churn after 6 weeks. In order to illuminate why the tree predicts this, we can pretend we are Andrea with the information as described below. Following the tree down from the top we see that the algorithm has noticed we are of middling BMI and we sent only a few (but not none) messages after week 3. According to the model that places us in a category where we are likely to churn after six weeks.

Andrea	Info
BMI	3.46
M.After 3	2.00
Age	31
(Mean) Emojis Used	0.00
Sentiment Coefficient	0.06
Sentiment Intercept	-0.14

0.4.1 *Slightly Technical* Discussion

Although this analysis using decision trees managed to identify some key discriminating features and make a prediction for who out of the 10 test cases will churn, I believe further analysis is needed before fully trusting this model.

It is probable that the reasons for churning are varied, an old user whose motivation is health has very different reasons to a young person whose motivation is looks (i.e. users who churn come from a range of different distributions). Thus, identifying chunks (hypercubes or hypercuboids) of the feature space which are densely churn populated, as decision trees do, is a suitable approach. However, due to the low number of instances, we may only have 1 or 2 samples from each distribution. Trying to fit hypercubes around these invokes the proverb 'needle in a haystack'. A decision tree of five layers can produce a maximum of 32 hypercubes, which would on average contain around 10 data points each. Ideally, with enough data, we could stratify this space, for example by age, and make our task much easier.

As it turned out, BMI was judged as very important by the decision tree, more so than height or weight. The decision to use BMI is an example of where domain specific knowledge could help improve the performance of the decision tree. I also believe, not enough use was made of temporal data. The linear regression approach used for sentiment, could be extended to other features such as emoijis used or frequency of messages. Going further, it is likely non-linear regression techniques could better approximate the changes over time (see Figure 2).

Lastly, decision Trees, were used here for their clarity alongside their aforementioned ability to cope with varying generating distributions. However, other more complex methods could, perhaps, be more successful in locating obscure patterns in the data.