Data Science Process for Client Usage Predictor Model Linear Regression Experiment 2

Code and Experiment Report produced by Lang (Ron) Chen November 2021-February 2022 for Lucidity Software

Date updated: 11/10/2022

Data Science Process

1. Took raw data and collected each client's weekly usage of 7 Lucidity Software modules (Actions, Assets, Competency records, Form records, Form templates, Incident, Users) from 2017.7.3 to 2021.8.15. Each client's weekly usage (with their usage of the 7 modules as attributes) are now instances of data known as Client-Weeks.

2. For each individual client, all the Client-Weeks counting backwards to the most recent week where they did not have any activity would be deleted; once the most recent week is reached where they have used at least one module once, this client-week and all that came before it would be retained (even if there were previous weeks with 0 activity over all 7 modules). This is the 'dropping of end zero weeks'.

   This is because if there is no recent activity it is then assumed that the clients have terminated their usage of Lucidity's services, and thus shouldn't be included in either training or testing (including former clients would affect predictions because the final 'increase' 'decrease' prediction is made by ranking the predicted scores of all clients that week and selecting the top 5% and bottom 5% respectively – thus having irrelevant/dummy clients in the mix would not distort the predictions.)

3. The data of the client usage are normalised – with respect to data within the same module of the same client (i.e. the Actions data of client A would be normalised with respects to all the actions data of client A (after dropping the 'end zero weeks' in step 2) only. This is so to allow comparisons between large clients and small clients.

4. Usage Scores were engineered for each Client-Week

5. A linear regression model would be fitted for the overall time-series normalised data vs the engineered Usage Scores.

   For the modules that used the time series data of each module independently, the time series data as a whole would only be used in the final predictor training if the r^2 value of the linear regression model of just that module vs the engineered Usage score surpassed a pre-determined value.

6. The top 5% of all the predicted scores in a particular week would be predicted as 'increase', whilst the bottom 5% would be predicted 'decrease'

Those labelled in yellow are subject to variations in Data Science decisions. See below

Variations in experiment:

The process of data wrangling, target-engineering, and combining the scores into one single predictor allows for many choices to be made; it is impossible to determine which choice is best without actually training them and testing. Utilising modern computational power, scripts were made to train and test models with different combinations of the various Data Science decisions.

Below are the different choices made, and their denotation in the scripts.

1. Rub off the first 26 weeks of all clients?
   Because the first 26 weeks of any clients may include volatility (i.e. start-up usage may be small and thus have an effect on normalisation), a reasonable choice is to discard all the Client-Week rows which came from the first 26 self-weeks of a client's usage of Lucidity.

   Denotation:
   'A': All
   '-26': without the first 26 weeks

2. How many 'Pastweeks' to use for time-series data?

   Predictions should be done with more than 1 week of data, but how many past-weeks to use is a key question. In the scripts, the range of 1 week to 12 weeks of past data for each module were trailed.

   Denotation:
   '1': 1 past week
   '2': 2 past weeks
   …

3. How to engineer client target score for each Client-Week?

   A. Use the sum of Users and Actions
   Denotation: '2_1'

   B. Use the sum of Users, Actions and Form_records
   Denotation: '3_1'

   C. Use the weighted sum of Users and Actions, where the weights are the correlation coefficient between each and the sum of Users and Actions
   Denotation: '2_2'

   D. Use the weighted sum of Users, Actions and Form_records, where the weights are the correlation coefficient between each and the sum of Users, Actions and Form_records
   Denotation: '3_2'

4. What to ultimately use as target for each Client-Week
   A. The engineered score of next week
   Denotation: 'S'

   B. The difference in engineered scores of next week and this week

Denotation: ‘T’

5. What to use as past weeks

   A, B, C, D were only used with ultimate target of ‘S’
   Overall denotation: SS_S, SS_, SsSs_S, SsTs_S

   A. This week = Engineered Scores; Past weeks = Engineered Scores
      Denotation: ‘SS’

   B. This week = Engineered Scores; Past weeks = Past trends
      Denotation: ‘ST’

   C. This week = Observations of each module; Past weeks = Observations of each module
      Denotation: ‘SsSs’
   D. This week = Observations of each module; Past weeks = Trends of each module
      Denotation: ‘SsTs’

   E and F were only used with ultimate target of ‘T’
   Overall denotation: TT_T, TsTs_T

   E. Past week = Trend of score
      Denotation: ‘TT’

   F. Past weeks = Trends of each module
      Denotation: ‘TsTs’

6. Cutoff value

   AKA Feature Selection Cutoff Value (fs_val), it will determine whether or not to ‘accept’ each linear regression model based on their r^2 value.

   Cutoff value choices: 0.2, 0.3, 0.4

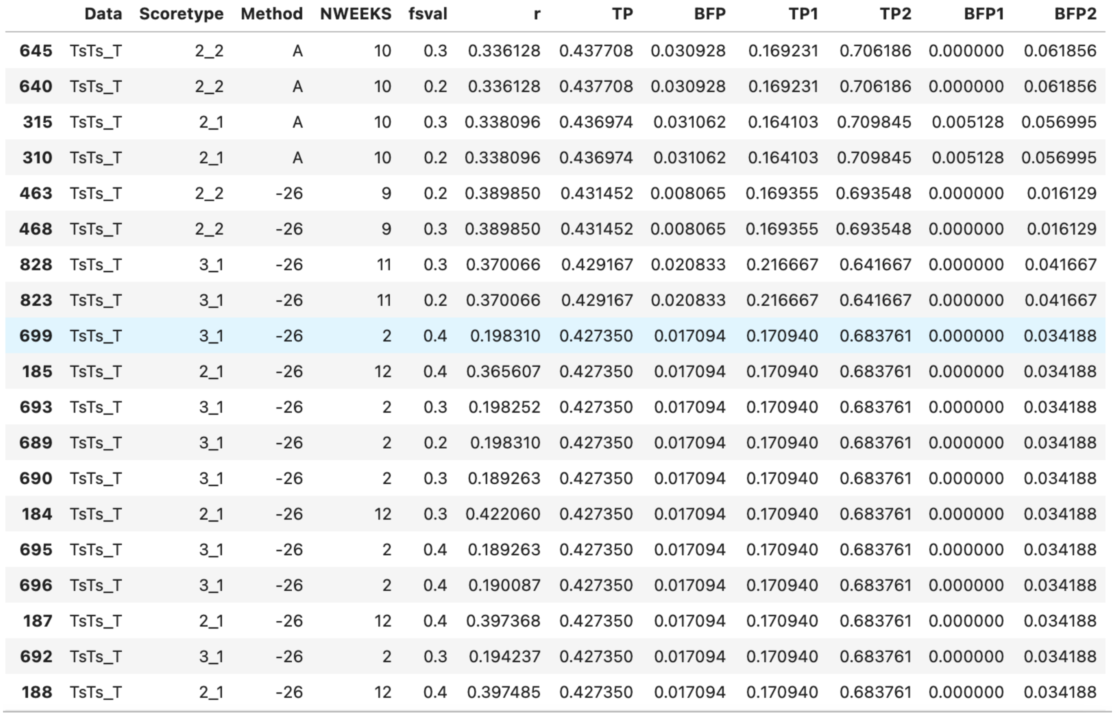
Train-test split and determining best model

A Five-Fold Split (each 80%-20%) were made for each type of data manipulation. The splits were done in terms of weeks rather than Client-Weeks because the final prediction is made in terms of whole weeks of Client-Weeks.

For each of these combinations of choices and their splits, the TP1 - true positive 1 value (#correctly predicted positives/#predicted positive) and TP2 - true positive 2 value (#correctly predicted negatives/#predicted negative) was recorded in a results csv that the script outputted. The average of TP1 and TP2 gives the overall TP - true positive value for this particular way to train the model

The best model would be selected by taking the average of the TP values of all 5 of its splits, and selecting the one with the top TP. Note if any split couldn't return a result because none of its module Linear Regression Models yielded r^2 values above the fsval, then it would not be considered at all. Calculating the combinations, 1056 different models were trained and tested, and taking into account the five splits meant 5280 models trained and tested – all done automatically via script.

The top models was 'TsTs_T', '2_2', 'A', '10', '0.3'. (note here r is the actual r^2 value of the final linear model rather than a variation in experiment. Whilst the TP value was not as good as the LR(diff) models, the top LR(score) model still produced a relatively good score of 43.77%



	Data	Scoretype	Method	NWEEKS	fsval	r	TP	BFP	TP1	TP2	BFP1	BFP2
645	TsTs_T	2_2	A	10	0.3	0.336128	0.437708	0.030928	0.169231	0.706186	0.000000	0.061856
640	TsTs_T	2_2	A	10	0.2	0.336128	0.437708	0.030928	0.169231	0.706186	0.000000	0.061856
315	TsTs_T	2_1	A	10	0.3	0.338096	0.436974	0.031062	0.164103	0.709845	0.005128	0.056995
310	TsTs_T	2_1	A	10	0.2	0.338096	0.436974	0.031062	0.164103	0.709845	0.005128	0.056995
463	TsTs_T	2_2	-26	9	0.2	0.389850	0.431452	0.008065	0.169355	0.693548	0.000000	0.016129
468	TsTs_T	2_2	-26	9	0.3	0.389850	0.431452	0.008065	0.169355	0.693548	0.000000	0.016129
828	TsTs_T	3_1	-26	11	0.3	0.370066	0.429167	0.020833	0.216667	0.641667	0.000000	0.041667
823	TsTs_T	3_1	-26	11	0.2	0.370066	0.429167	0.020833	0.216667	0.641667	0.000000	0.041667
699	TsTs_T	3_1	-26	2	0.4	0.198310	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
185	TsTs_T	2_1	-26	12	0.4	0.365607	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
693	TsTs_T	3_1	-26	2	0.3	0.198252	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
689	TsTs_T	3_1	-26	2	0.2	0.198310	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
690	TsTs_T	3_1	-26	2	0.3	0.189263	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
184	TsTs_T	2_1	-26	12	0.3	0.422060	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
695	TsTs_T	3_1	-26	2	0.4	0.189263	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
696	TsTs_T	3_1	-26	2	0.4	0.190087	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
187	TsTs_T	2_1	-26	12	0.4	0.397368	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
692	TsTs_T	3_1	-26	2	0.3	0.194237	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188
188	TsTs_T	2_1	-26	12	0.4	0.397485	0.427350	0.017094	0.170940	0.683761	0.000000	0.034188