Feature Importance in predicting L2 word learning accuracy

Arun Jose (jose4@wisc.edu)
Department of Computer Sciences, 1210 W. Dayton Street
Madison, WI 53706 USA

Abstract

Why does a Machine Learning algorithm choose one output over another? Can the feature importance parameter in Decision Trees provide some insight into this? Is there any empirical relation between these feature importances and the regression coefficients in Linear Regression? This project attempts to tackle these questions by running multi-model regression analyses on a large scale Duolingo L2 word learning dataset.

Keywords: big data; second language learning; machine learning; feature importance

Introduction

Most of the cutting-edge research today is on how to improve the accuracy of machine learning models, down to the least significant decimal point. Not much insight is available as to why a particular algorithm made a particular decision. However, as machine learning becomes more and more intertwined with critical applications in the real world, this can no longer be ignored. This project attempts to peek inside the black box of machine learning and obtain some human-interpretable reasons as to why some decisions were made over others.

The goal of this paper is to explore how varying the model used impacts the importance of the predictors of L2 word learning accuracy. Linear regression analyses work well in the case of linear decision boundaries, but fail to capture any non-linear relations in the data. Would the results be statistically different on using a non-linear classifier like neural nets or decision trees? In particular, the goal is to experiment with the state-of-the-art non-linear machine learning models like gradient boosted decision trees (Chen & Guestrin, 2016) and measure the importance of the predictors on the same dataset to see how different the results are. What predictors remain model-agnostic in terms of importance?

The paper starts with an overview of the prior work done in both these disparate approaches. This is then followed by the description of the methodology in the project which includes a description of the dataset, predictors used, the machine learning algorithms used, and the feature importance measures employed. Following this, a visualization of the results obtained and interpretations for the same are provided. The paper concludes by outlining a few limitations with the approach, the scope for future work and final remarks on the project.

Previous Work

Recent studies (Paxton & Griffiths, 2017) in the Cognitive Science literature have shown that using big and naturally occurring datasets to supplement traditional experimental paradigms is a powerful tool to understand human behavior and cognition. One recent application in this regard was an investigation (Hopman, Thompson, Austerweil, & Lupyan, 2018) to predict human word learning accuracy of a second language (L2) from a Duolingo big data set. The paper models ten word-level parameters as predictors of L2 word learning accuracy and measures their importance using linear regression. The measure to gauge importance used in this paper was to look at the standardized coefficients(β values) upon fitting the data and use that as a measure of how much influence a predictor has on the overall accuracy of the model.
A fundamentally different approach to obtain the same indicator of the influence of a predictor is the feature importance from a decision tree based algorithm. There has been a good amount of research done (Breiman, Friedman, Olshen, & Stone, 1984; Kira & Rendell, 1992; Grabczewski & Jankowski, 2005) to perform feature selection based on some impurity measure like the Gini index within decision trees. Using such measures, one can obtain a quantitative measure of how important a particular feature is in the overall prediction.

The results from both these contrasting approaches can be compared to find similarities/dissimilarities in the feature importances. This could be used to provide a robust set of potential predictors to model L2 word learning accuracy.

Which of these is "the cheese"?

Figure 1: The Duolingo app interface

Method

The first step in the project was to work on getting the prediction model set up as per the Hopman et al. (2018) paper. There needed to be some modifications to the Duolingo dataset used (Settles & Meeder, 2016). There have also been a few modifications to the predictors that the authors have been working on and these have been discussed in detail in the following subsections. A sample UI screen from the Duolingo app is shown in Figure 1 for reference.

The Duolingo Dataset

The original Duolingo dataset (Settles & Meeder, 2016) contains a total of 12.9 million instances of data. However, these data points relate to one individual user on a single lexeme during a single session. Since different lexemes could refer to the same word in most cases (e.g. cat vs. cats), and since most predictors did not differentiate between such lexemes, the dataset was aggregated by word. This dataset was further streamlined to capture the data for English speakers and learners for only three languages: Spanish, Italian, and Portuguese. One implicit assumption made here was that the user's UI language obtained from the dataset is the user's L1, i.e. their native language. Further, data points for users with less than 41 data instances, and words which were practiced less than three times per individual user were removed. All this led to a total of 3.17 million data points. Among these data points, there were around 8237 entries with missing values for human concreteness. For example, proper nouns like 'America' had no defined value for human concreteness. These words were imputed with the mean value for this predictor. Finally, the remaining data points with NaN values for any predictor were dropped to obtain a total of 3.15 million data points.

Since word level predictors might be better suited to a dataset grouped by words, another dataset was formed from this as well which aggregated all the data points per word. This dataset had a total of 8793 data points. In the rest of the paper, the first dataset will be referred to as the 'complete dataset' and the word-aggregated dataset will be referred to as the 'agg-by-word dataset' for simplicity.

The Word Level Predictors

A short description of each of the predictors used has been described below. The corresponding column names used in the model are also mentioned in brackets for reference.

- **Word experience** (*wordexperience_log*): The total number of times a user saw a given word. This has been logged to make the values comparable with other predictors.

- **User experience** (*userexperience_log*): The sum of word experiences for all words that a user has experienced. This has been logged to make the values comparable with other predictors.

- **Human Concreteness** (*human_conc*): The human-rated concreteness of the English word of a translation pair.

- **Levenshtein distance** (*ld*): The normed Levenshtein distance between the translation and the keyword.

- **Average local alignment** (*local_alignment_avg*): The average of the local alignments calculated in both directions of the word-translation pair. The procedure to calculate this value has been described in detail in Thompson, Roberts, and Lupyan (2018).

- **Semantic density** (*semantic_density_l1/l2*): The density of the semantic neighborhood of the L1/L2 word.

- **Word frequency** (*l1/l2_freq_wfzipf*): The word frequency measure from Python's wordfreq package. This is logged since word frequency generally follows a Zipfian distribution.

- **Distinct meanings** (*l1/l2_nrsynsets_wn*): The number of different meanings the word has on wordnet.

The ML Regression Algorithms

This section provides a rough description of the Machine Learning models used in the regression analysis for predicting L2 word learning accuracy. The first model described is a simple linear regression model that captures the coefficients of regression, similar to what was used in the Hopman et al. (2018) paper. The other three models below were chosen for two main reasons. Firstly, all three of them are able to capture non-linearities in the data points. And secondly, all of them have a fairly intuitive computation for feature importance, which is different from the methodology used in the linear regression model.

Linear Regression A linear regression model assumes that there exists a linear relationship between the dependent variable and the predictors. It attempts to fit this relationship using the equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \varepsilon \qquad (1)$$

where y is the predicted value; x_1, x_2,..., x_p are the p predictors; β_1, β_2,..., β_p are the corresponding regression coefficients; and ε captures the noise.

Decision Tree A decision tree is a machine learning model which constructs a tree-like inference structure from the data, split by the predictors. It provides a simple and interpretable model to the user. Since the target variable here takes continuous values, regression trees are used.

Random Forest Random forests (Breiman, 2001) work on the principle of bootstrap aggregation, also known as *bagging*. Bagging involves taking random samples from the dataset and creating B different training sets, each of size n and running a regression decision tree to compute the prediction f_b. These predictions are then averaged out to get the final prediction $\hat{f}$ for unseen samples x' as:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x') \qquad (2)$$

Random forests add on top of the idea of bagging to select only a random subset of the features at each candidate split. This small tweak provides a way to decorrelate the trees.

Gradient Boosted Decision Tree *Gradient boosting* works similar to bagging in the sense that a set of decision trees are used to make a prediction. The difference is that the ensemble of trees are built sequentially (as opposed to independently in the case of random forests), and the results are combined along the way. **XGBoost** (Chen & Guestrin, 2016) is an open-source library that implements gradient boosted decision trees and has gained a lot of popularity over the last few years as the algorithm of choice for many winning teams in machine learning competitions.

Measures of Feature Importance

In a linear regression analysis, the sign of the regression coefficient (β) indicates whether there is a positive or negative correlation between the dependent variable and each predictor. Moreover, the magnitude of this coefficient is also a quantitative measure of how influential the corresponding predictor is in contributing to the overall model accuracy. For example, if a predictor p has a β value of 0.2, it means that a 1 Standard Deviation increase in the value of p results in a 0.2 Standard Deviation increase in the overall accuracy of the model.

Decision trees provide a fundamentally different approach to the same problem, i.e. to get a measure of how important a feature is to the model. Each node in a decision tree is a condition to split values with a single feature. This condition is based on the difference in impurity measures after the split, for instance the Gini impurity or variance measures (Breiman et al., 1984).

The Scikit-Learn package in Python provides a way to capture the significance of this split condition via a *feature_importances_* parameter. It offers to calculate this measure by using different *importance_type* measures such as 'weight'(the number of times a feature is used to split the data across all trees) and 'gain'(the average gain across all splits the feature is used in). The default *importance_type* of 'weight' has been used throughout in this paper. In the case of ensemble models like random forests and gradient boosted decision trees, these impurity values are averaged over all the constituent trees to calculate the feature importance.

Scikit-Learn also provides another feature importance measure called 'Permutation feature importance' using its *eli5* package. This approach directly measures feature importance by observing how the model performance is influenced by random re-shuffling (thus preserving the distribution of the variable) the values of each predictor.

Results

All the feature importance measures described above were implemented in Python and the results on both the datasets are shown in Figure 2 and Figure 3. In the case of linear regression, the coefficients of regression (β values) are used as a measure of feature importance. Since this paper focuses only on the quantitative feature importance, the absolute values of β were taken. This was then divided by the sum of the beta values for each predictor to obtain a normalized feature importance value for Linear Regression.

For the other three models, the feature importance was calculated using the 'weight' importance type, i.e. the number of times the feature was used to split the data. I also computed the corresponding values for 'gain' importance type (not shown), but the results obtained were similar. Finally, the permutation feature importance of each predictor was also computed for each of the three models. To align these values with the obtained results, they were also scaled from 0 to 1 by normalizing them. The feature importance could take negative values, which implies that adding the predictor performed arbitrarily worse than a random permutation. Such values were set to zero feature importance in the results shown.

Since, three of the four models shown are non-linear, and prior research (Spiess & Neumeyer, 2010) has shown that the R^2 is not a valid measure for non-linear regression models, Mean Squared Error (MSE) has been used as a measure of goodness of fit in this paper. The MSE values for the four models have been outlined in Table 1.

Table 1: MSE for the models

Model	MSE
Linear Regression	0.013285
Decision Tree	0.026004
Random Forest	0.016003
XGBoost	0.033280

Discussion

Upon examining Figure 2, user experience is one predictor that stands out as a high-importance predictor in almost all the plots. So it is highly likely that a user who has spent a lot of time learning a language on Duolingo is better at that language and has a higher probability of success, as opposed to newer users. The other predictors that show a good frequency of being high-importance are the Levenshtein distance and the semantic density of the L1 word. This indicates that word pairs that are closer in terms of translation distance and words, whose translations in the user's L1 are spread out from its semantic neighbors, are likely to be easier to learn than others. The only predictor that is consistently low-importance in all the models is the number of distinct meanings for the word. So this might not be a good predictor for the dataset and could be removed on further analyses. Other predictors like human concreteness and the average local alignment show inconsistent behavior and these need to be investigated with larger datasets or by incorporating human data to get a better understanding.

In Figure 3, a spike in the word experience predictor is observed as compared to Figure 2. This is expected, since the dataset now aggregates the data points by individual word for

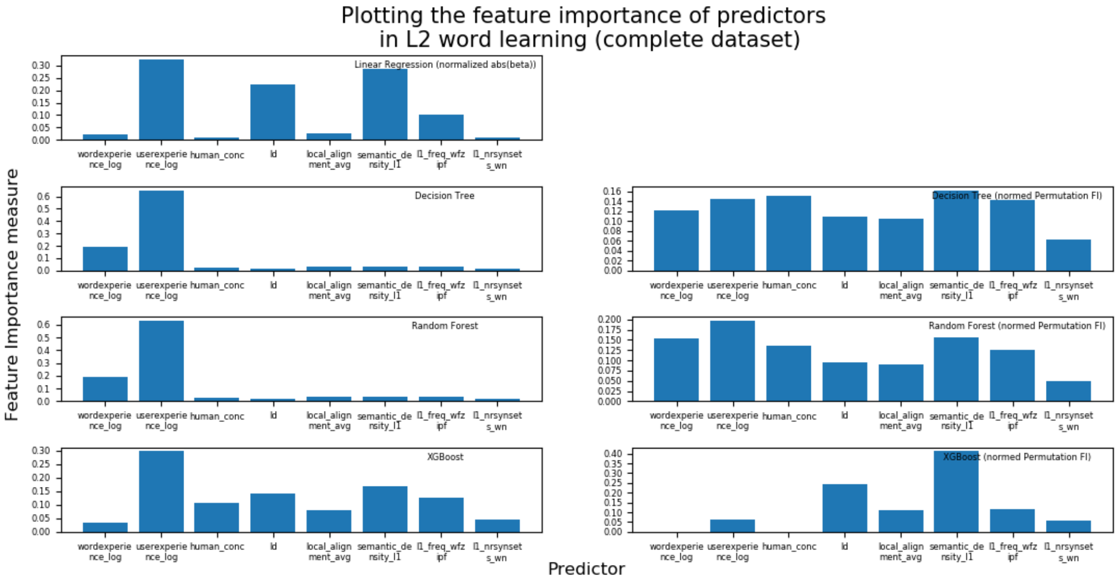

Figure 2: The feature importances for the four models: Linear Regression, Decision Tree, Random Forest, and XGBoost on the 'complete dataset'. Absolute values of the linear regression coefficients were normalized to obtain a comparable FI measure. The permutation feature importances were also computed for the other three models and normalized before plotting. See subsection *The Word Level Predictors* for a description of the predictors.

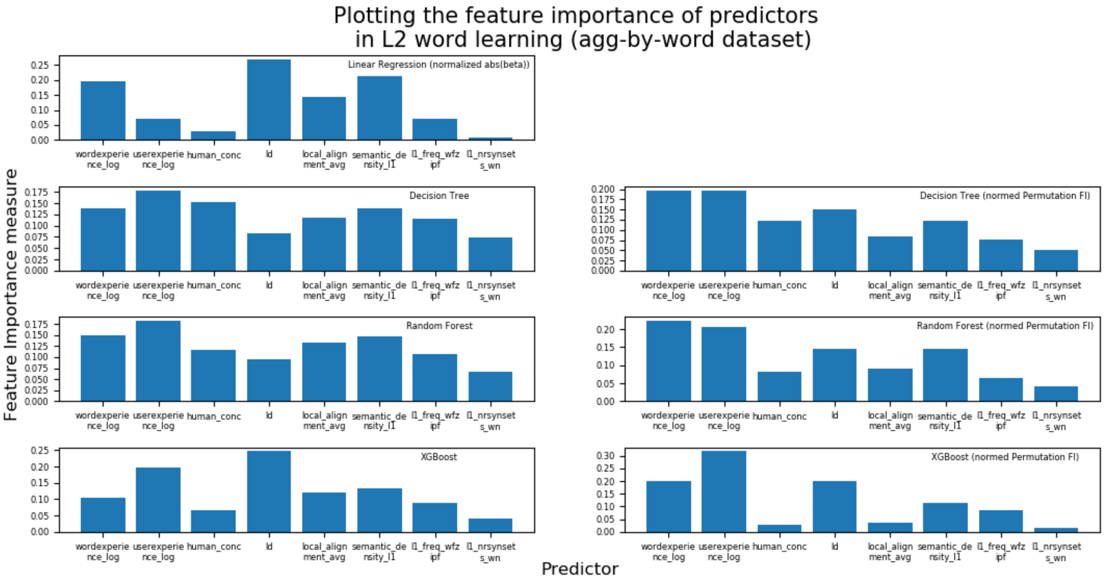

Figure 3: The feature importances for the four models: Linear Regression, Decision Tree, Random Forest, and XGBoost on the 'agg-by-word dataset'. Absolute values of the linear regression coefficients were normalized to obtain a comparable FI measure. The permutation feature importances were also computed for the other three models and normalized before plotting. See subsection *The Word Level Predictors* for a description of the predictors.

all users and hence the effect of seeing the word a number of times would be higher. As the authors mention in Hopman et al. (2018), this dependence is actually inversely related since Duolingo's algorithm is not random and asks users questions with words, for which they had failed before, more often than other words. Apart from this, the three predictors mentioned above, namely user experience, Levenshtein distance, and the L1 semantic density, all continue to show fairly high importance. As a result, these three are likely to be the most influential of all the predictors used according to this dataset. Once again, the number of distinct meanings for the word appears to be a poor predictor and could be removed from the list of predictors. In this figure, however, human concreteness and the average local alignment shows more consistent high importance as compared to Figure 2, and the aggregation by words seems to have an influence on the importance of these predictors. I could not think of any particular reason that this was happening, and further research could be done to analyze these predictors.

Limitations

The dataset has a few limitations already described in Hopman et al. (2018): the accuracies are skewed to values close to 100%, the Duolingo algorithm is non-random, and there is an implicit assumption made to select the UI language as the user's L1.

The project attempts to decipher a little bit of the reason why a machine learning algorithm makes a particular decision based on the training data provided. This is a way of attempting to interpret the results of the model by looking at the most significant predictors, and provide some justification for the same. However, there is still no real way to explain the inconsistencies between feature importances among the various models. The models just build their respective tree structures to minimize prediction error and only the final formed tree nodes are observed.

This project makes no further attempt to investigate the differences between the feature importance values among the models. Instead, it offers a robust indicator to which of the predictors could be reliable to model the difficulty in learning a new language.

Future Work

A few action items for future work to build on this has been outlined below:

- Only decision tree based algorithms have been implemented in this paper. Recent research (Ish-Horowicz, Udwin, Flaxman, Filippi, & Crawford, 2019) on other non-linear models, such as neural networks, offer a good starting point to extend this idea to a more diverse set of machine learning models.

- The current results capture the data for all three languages (Spanish, Italian, Portuguese) in both directions w.r.t. English. A separate analysis could be performed to investigate any difference in predictors between English learners and English speakers.

- Parts of Speech (POS) is another parameter that can be used either as a predictor or as a control variable to see how the result varies.

Concluding Remarks

The project allowed me to explore an area of big data and machine learning that I am really curious about, which is to interpret the output of a model in some meaningful manner. As the results show, there is a consensus among most algorithms on certain predictors of L2 word learning: namely the user experience in learning the language, the translation distance of the word from the user's L1, and the how densely populated the word's semantic neighborhood is. These predictors are a robust indicator of the existence of some meaningful pattern in the dataset that can be modeled using these predictors. Finally, as the authors mention in Hopman et al. (2018), these results shouldn't be used as a stand-alone metric to see which parameters are important. Rather, they need to be combined with a more controlled study on actual human behavior, for which such studies provide a useful starting point.

Acknowledgments

The author is indebted to Elise Hopman for taking time out of her schedule to meet on a weekly basis, share the updates that she has been working on since the Hopman et al. (2018) paper, and discuss approaches to be taken at each stage of the project. The author would also like to thank Prof. Austerweil for the constant encouragement and suggestions throughout the project. Last but not least, the author is thankful to his fellow classmates in the *Computational Cognitive Science* course and the members of the Austerweil Lab who offered useful suggestions like evaluating the Permutation feature importance.

References

Breiman, L. (2001). Random forests. *Machine Learning*, *45*.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. CHAPMAN HALL/CRC.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Grabczewski, K., & Jankowski, N. (2005). Feature selection with decision tree criterion. In *Fifth international conference on hybrid intelligent systems (his'05).* IEEE.

Hopman, E. W. M., Thompson, B., Austerweil, J. L., & Lupyan, G. (2018). Predictors of l2 word learning accuracy: A big data investigation. In *Proceedings of the 40th annual meeting of the cognitive science society.* Austin, TX: Cognitive Science Society.

Ish-Horowicz, J., Udwin, D., Flaxman, S., Filippi, S., & Crawford, L. (2019). Interpreting deep neural networks through variable importance.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In *Machine learning proceedings 1992* (pp. 249–256).

Paxton, A., & Griffiths, T. L. (2017). Finding the traces of behavioral and cognitive processes in big data and naturally occurring datasets. *Behavior Research Methods*, *49*.

Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the acl.*

Spiess, A., & Neumeyer, N. (2010). An evaluation of r^2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a monte carlo approach. *BMC Pharmacol*, *10*.

Thompson, B., Roberts, S., & Lupyan, G. (2018). Quantifying semantic similarity across languages. In *Proceedings of the 40th annual conference of the cognitive science society.*