STAT-702 FINAL (GROUP DS-1)

4/29/2018

Miranda Morgan - Miranda.Morgan@trojans.dsu.edu
Krishna Harsha Kosuri - KrishnaHarsha.Kosuri@trojans.dsu.edu
Aakanksha Jaiman - Aakanksha.Jaiman@trojans.dsu.edu
Jennifer Schulte - Jennifer.Schulte@dsu.edu

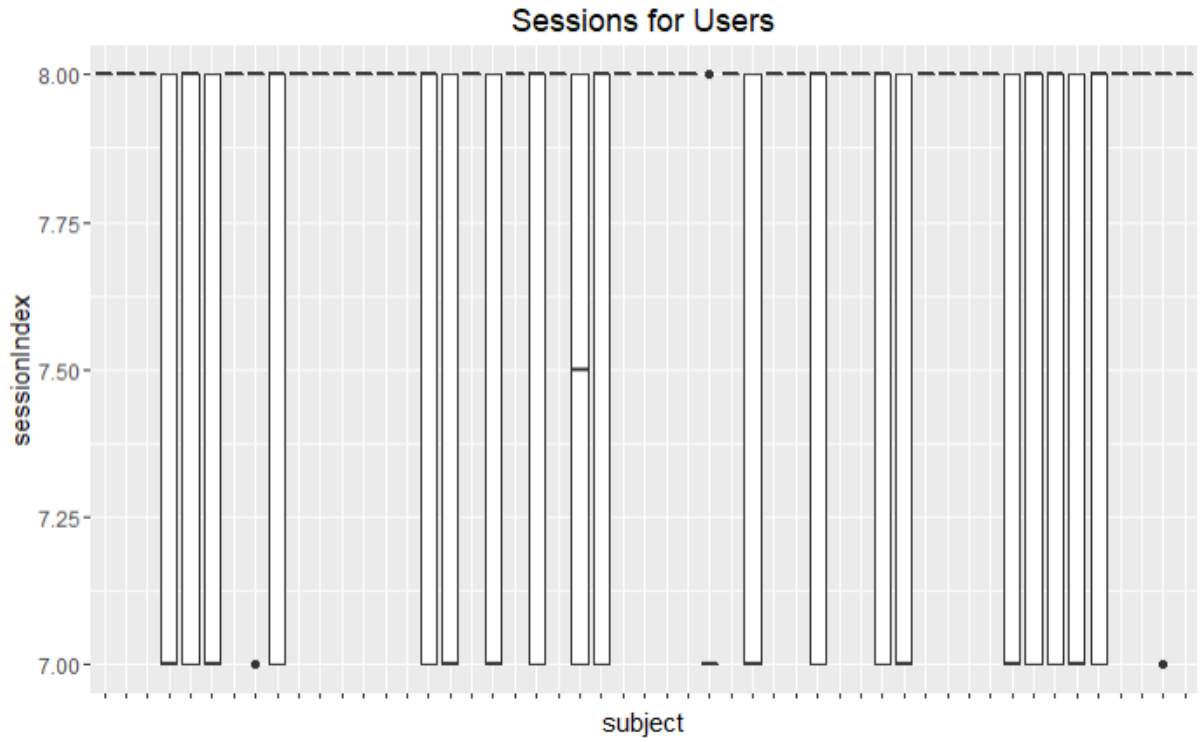
Introduction

About the Data

This is a KeyStroke Dynamics dataset with a group of 51 Individuals who have access to a passcode for a specific system. There are 34 columns in the dataset each describing the typing interval to type in the passcode: .tie5Roanl.

Key Goal of the Paper

Using the Known dataset we are going to evaluate the dataset. This will be done by using 16 classification models to find the best performance, and use this to predict the users in unknown and the questioned dataset.



Summary of the Known Dataset

These predictors describe timing between typings, the session, and the amount of entries into each session (Final Typing 2018, 1-4). Through initial exploration shown in the figure above, we can see that we have two unique session indexes. The most users were in session 8. We also checked for missing values and there were none that appeared.

Exploratory Analysis

In the Exploratory analysis we are going for 4 methods: correlation, VIF function, variable importance using random forest, and histograms for each numeric variable.

Correlation

UD.period.t	UD.i.e	UD.e.five	UD.Shift.r.o
Min. :-0.1770	Min. :-0.12800	Min. :-0.1257	Min. :-0.0615
1st Qu.: 0.0361	1st Qu.:-0.00270	1st Qu.: 0.1103	1st Qu.: 0.0425
Median : 0.0800	Median : 0.02930	Median : 0.1573	Median : 0.0899
Mean : 0.1315	Mean : 0.04965	Mean : 0.2103	Mean : 0.1350
3rd Qu.: 0.1774	3rd Qu.: 0.07460	3rd Qu.: 0.2550	3rd Qu.: 0.1712
Max. : 1.7892	Max. : 1.24790	Max. : 4.8827	Max. : 4.0120

UD.o.a	UD.a.n	UD.n.l	UD.l.Return
Min. :-0.11310	Min. :-0.18290	Min. :-0.175800	Min. :-0.0646
1st Qu.: 0.01220	1st Qu.:-0.01740	1st Qu.: 0.008075	1st Qu.: 0.1051
Median : 0.03860	Median : 0.01140	Median : 0.079650	Median : 0.1455
Mean : 0.05663	Mean : 0.02871	Mean : 0.086764	Mean : 0.1918
3rd Qu.: 0.07000	3rd Qu.: 0.05158	3rd Qu.: 0.129800	3rd Qu.: 0.2312
Max. : 2.43700	Max. : 2.07750	Max. : 1.799000	Max. : 1.6119

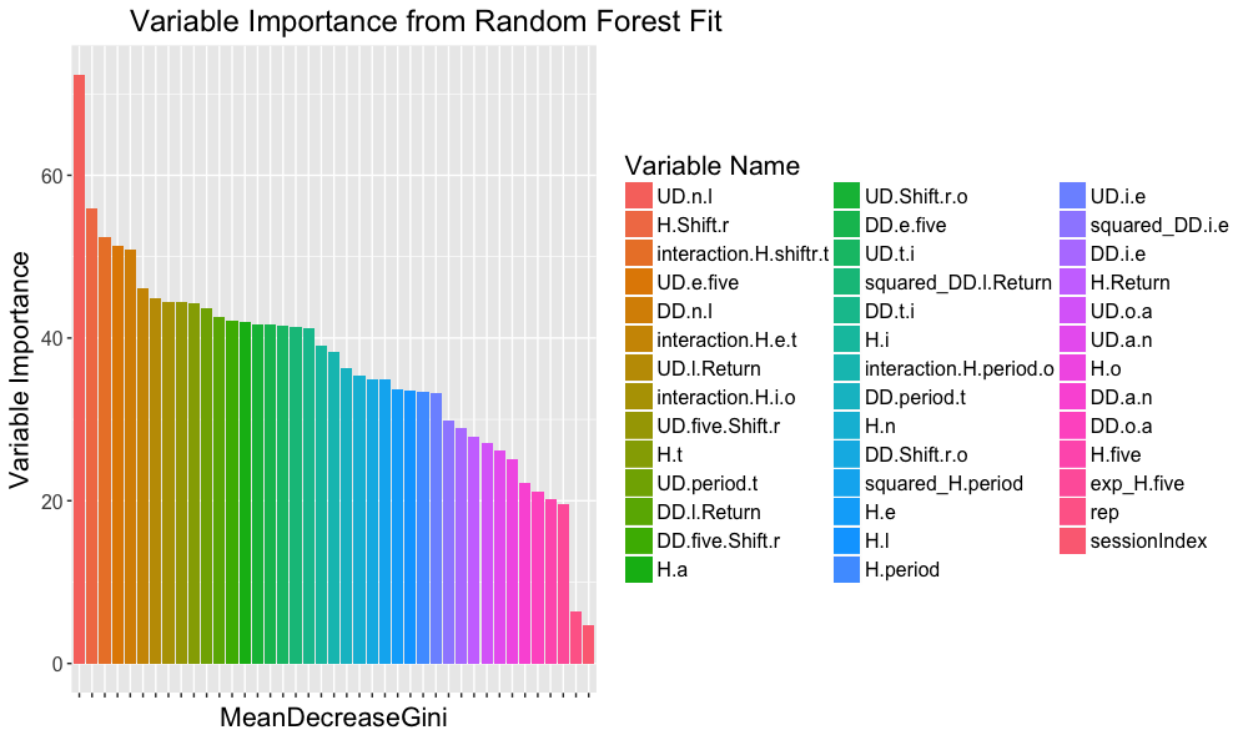
Using correlation we noticed a peculiar pattern among UD variables. These had a negative minimum time over the set of data. The only UD variable this did not occur in was UD.five.Shift.r where the minimum was positive. To understand this better, we would need to know what constituted as a negative time for these variables. It is happening between all of them, which is why it seems as though there is a reason. Due to that and the correlation evidence seen below, these will be excluded in the dataset. We also noticed that these were always less than DD, which seems to be calculating the same thing. This will lead to a collinearity problem later on.

We also ran into some correlations that stood out. There is a strong correlation between H.period and H.o with the highest at almost .69. H.e and H.t have a correlation of .72. H.i and H.o is a higher correlation of .66. H.e and H.period have the second highest correlation at .60 compared to the .72 above. .62 is the correlation between H.five and H.Shift.r. H.shift.r and H.t have a high correlation of .68. H.t and H.a have okay correlations with H.o at around .60, which is one of the lower we have seen for H.o. H.shift.r and H.a are correlated at .63.

VIF Function to see multi collinearity:

VIF is a function to calculate the Variation Inflation Factor to see if variables have multicollinearity or not. We are setting the threshold as 5 so if a variable has less than 5 as VIF then that variable is not multicollinear. The variables that are selected by VIF are part of the correlation analysis above.

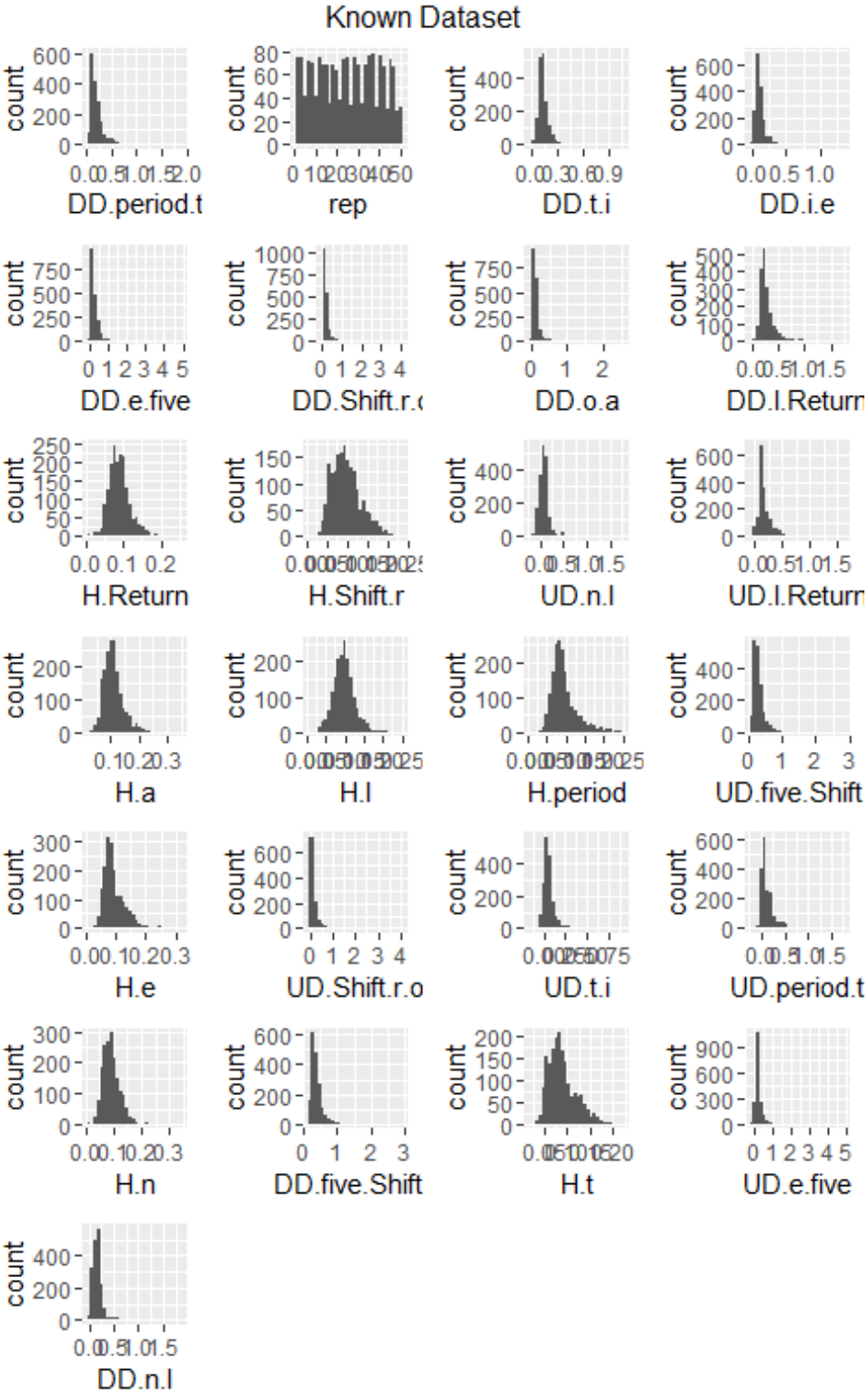
Variable Importance using random forest



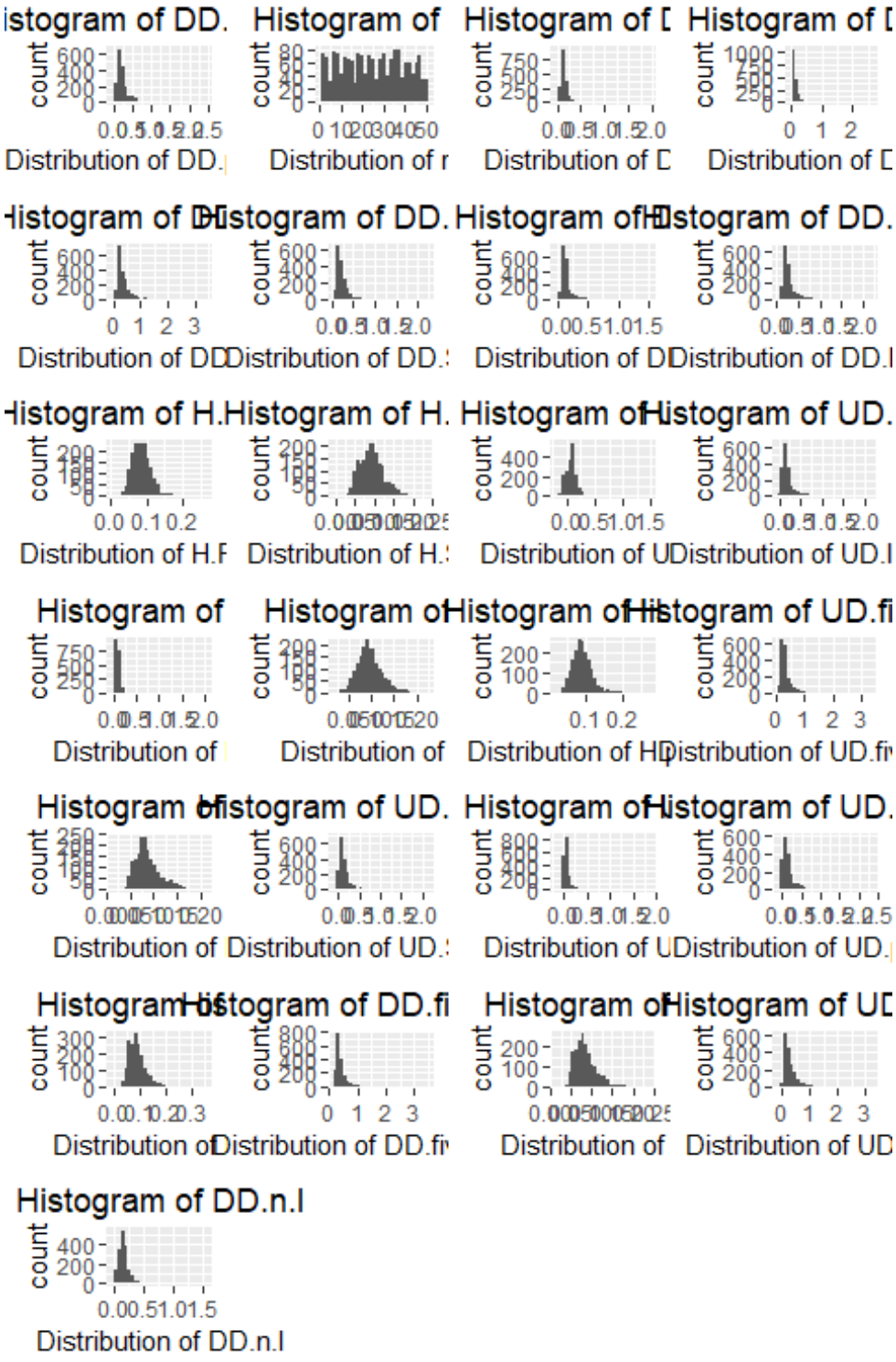
The other method used to fix the predictors is by looking into the variable importance plot from the random forest. Along with the variables selected in correlation we added two other variables from this varimportance plot they are: squared_H.period and squared_DD.i.e.

Histogram:

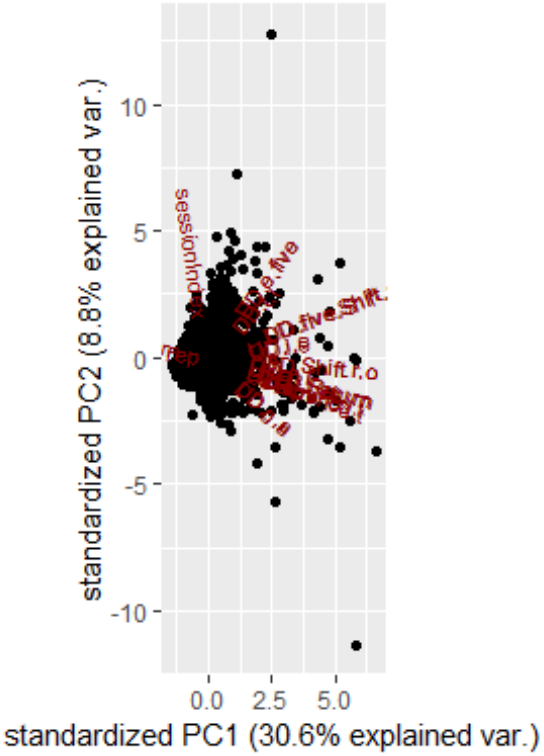
We are plotting each numeric variable to see the distribution of them, to see if they are normally distributed or have skewed distribution. Through this we see that we have skewed distribution in some of the variables like UD.five.Shift.r, DD.period.t etc. To deal with the skewed distribution, it is worth while to do log transformation. Below are plots for both the known and unknown datasets. Overall through initial exploration it seems we may need to use Unsupervised learning to better distinguish the best variables to use.


Known Dataset
count
DD.period.t
rep
DD.t.i
DD.i.e
DD.e.five
DD.Shift.r.(
DD.o.a
DD.I.Return
H.Return
H.Shift.r
UD.n.I
UD.I.Return
H.a
H.I
H.period
UD.five.Shift
H.e
UD.Shift.r.o
UD.t.i
UD.period.t
H.n
DD.five.Shift
H.t
UD.e.five
DD.n.I

Unknown Dataset

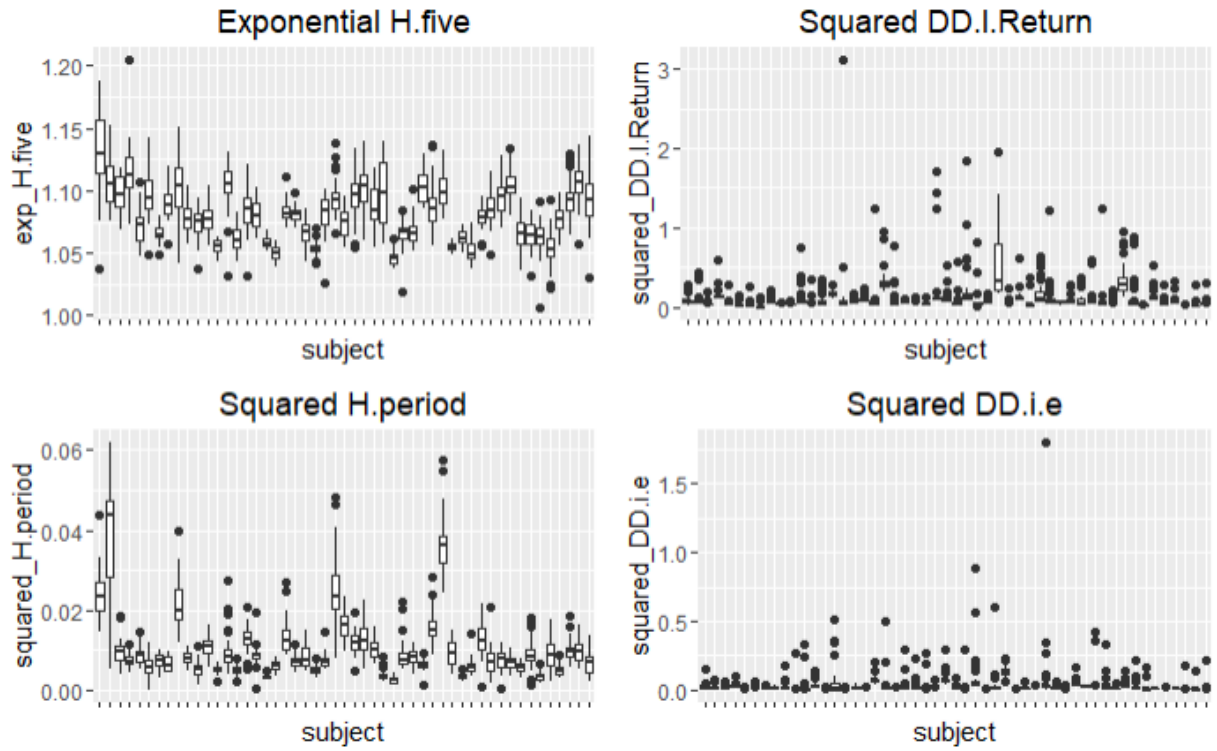


Unsupervised



It appears that both H.a and H.e are good predictor variables based off our tree methods. They are also used in our other training methods that we originally picked. The rest of the values were either showing in first or second principal component but not in both (We see that the DD's are showing more for the second component and H values for the first component). The values in the first component we used completely in training since they explain most of the variance.

Adding New Variables based on Exploratory Analysis



We also explored some transformations of variables. Most helpful seemed to be the exponential transformation of H.five shown above. We can see the individual users start to separate out with this new transformed variable and later it will be used in some of the models in order to distinguish among subjects. It was also noticed that there is an interaction between H.period and H.o, H.e and H.t, H.i and H.o, and finally between H.Shift.r and H.t. We also thought the transformation of DD.l.Return was worth looking at. We can see that some of the subjects that stood out in this set were enhanced when we squared them. Another squared transformation of H.period shows even more separation here with the squared term. There are more separated subjects here than there was before which may yield helpful in analysis. We finally added one more transformation of squared DD.i.e. There are 2-3 subjects that could be separated easier with this variable. There are several outliers that were even more pronounced with the transformation and we can see certain subjects are proving to have more than others as far as observations outside of the normal.

Log transformations on the variables in which had skewed distribution was used in creating a new, log transformed dataset. This will aid in dealing with the skewness to help create better performing models.

Model Evaluation

For all the analysis below the chosen split was a 70/30 split for the train and test set for VSA method. We chose this split, even though it may split up sessions in order to have all users represented in our training. This will allow for us to classify sessions for users in session 8, as well as users in session 7 in one sitting. To completely assess the models and their accuracy, since this approach is variable, we will also add in a 5-fold validation method.

Trees

The first tree model was a regular decision tree. This was done using rpart due to the significant number of variables that included all but the UD, sessionIndex, and rep. A minimum split of 10 was specified in order for smaller splits to be excluded. The testing sets for our sessions was split into two sets: session 7 testing set and session 8 testing set. This includes the whole testing set above, except broken out into two because of the unique session user. Dr. Saunders method, that suggests the maximum reps associated with one user in an event of interest was the chosen typer, was used. Then the two sessions were averaged together to get an overall session accuracy rate. The average session accuracy rate between session 7 and session 8 is an accuracy of 54%. Session 8 had a better accuracy rate overall at 65% where session 7 was at 43%. Then for k-fold, 68% for the average session accuracy, 76% for session 8 and 59% for session 7. So this model is performing okay on our sessions. The variables that the Rpart tree chose for the tree include interaction.H.e.t, DD.Shift.r.o, H.e, interaction.H.period.o, H.Shift.r, DD.five.Shift.r, DD.i.e, H.a, DD.period.t, DD.l.Return, DD.t.i, interaction.H.shiftr.t, DD.n.l, H.t, H.l, DD.e.five, H.period, H.n, H.i, H.o, DD.a.n. We noticed above that the interaction terms may be helpful which we see in the above selection. We also found above that H.a is a good separator of subjects as well.

Bagging

Next a bagging algorithm with the *randomForest()* command was trained. This was given 30 possible predictors to choose from at each split, which excludes all UD variables, sessionIndex, and rep. We then let importance=TRUE to determine important variables (R Help). The model then ran and we predicted the test set using that model. We obtained an error rate of 10.5%, which is an estimated accuracy of 89.5% on the repetitions for each typer. This is predicting highly accurately and seems to be a better fit than our tree algorithm above. We know from class that this method reduces the variance found in our previous model from repeated sampling and training (James et. al, 317). When running 5-fold cross validation the error rate become almost negligent. We have an accuracy of 99% on the repetitions. It appears the model lends 100% accuracy on our datasets with an error rate that was extremely low. These errors were spread out between sessions and remained low enough that for our users, most of their reps were predicting correctly. This leads to our sessions being correctly identified for the proper users. Also, looking at mean decrease in accuracy from the importance function on our model we see that the variables that have the biggest importance is DD.n.l, H.n, and H.Shift.r, and DD.e.five, which we saw above as important variables as well.

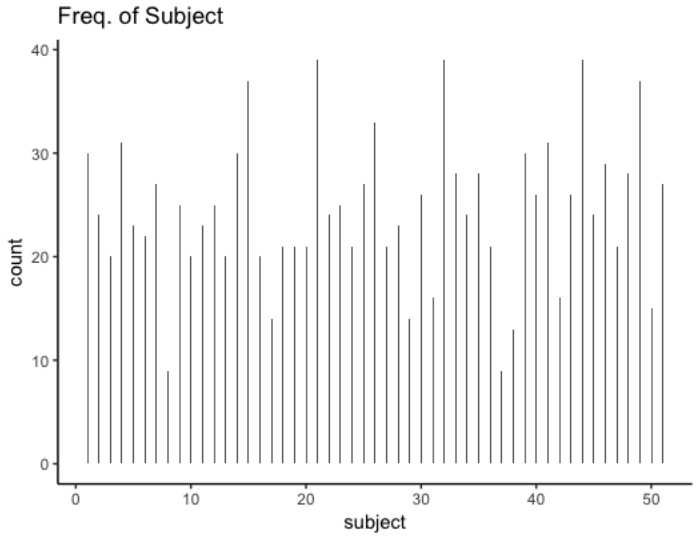
Random Forest

We have an error rate of 7.9% when doing the random forest method classifying reps. I trained a random forest model with all the predictors on the train set excluding rep, UD variables and Session index. We used 14 as the number of possible predictors for a split. We then let importance=TRUE to determine important variables (R Help). The book tells us that the random forest should be better than bagging due to the predictor specification difference at each split (James et. al, 319). This could also suggest similarity between the trees since we saw a good reduction of error when using the method above (James et. al, 320). We then have an accuracy of 92.1% on our test dataset reps. This is a big improvement on all of our models. We again performed 5-fold validation on the model and got 0% when looking at reps. Performing similar to bagging, we are receiving a 100% session accuracy rate for VSA and 99.5% with 5-fold since we predicted so well on the reps themselves. Looking at importance of variables, as far as decrease in accuracy, we have the most important variables as DD.n.l, DD.e.five, and H.Shift.r. These are also some of the important variables we have

seem from above, which solidifies that the model is performing similar to our others with a slight improvement overall.

The remaining models were trained with the following variables that we found in data exploration: H.period, DD.period.t, H.t, DD.t.i, H.i, DD.i.e, H.e, DD.e.five, H.five, DD.five.Shift.r, H.Shift.r, DD.Shift.r.o, H.o, DD.o.a, H.a, DD.a.n, H.n, DD.n.l, H.l, H.Return, DD.l.Return, squared_h.period, squared_dd.i.e, exp_H.five, squared_DD.l.Return, interaction.H.period.o, interaction.H.e.t, interaction.H.i.o, interaction.H.shiftr.t. The UD variables were not included in the models either.

QDA



For QDA class variables where frequency of subject variable >30 have been eliminated. The plot above shows the frequency of each of the class values. This is done because the number of predictors are 29 in total. There were 3 methods created with vsa and k-fold cv. The error rate for VSA model is 84.31373%, which gives an accuracy of about 15.68627% while the 5-Fold CV model gave an error rate of 42.1401% which is not so good with accuracy of 57.8599%. So, Overall the 5-Fold CV method stands out among the three methods.

MclustDA

The error rate for VSA model is 17.36695%, which gives an accuracy of about 82.63305% and thus shows a good performance. The 5-Fold model gave an error rate of 18.58213%, which is good with accuracy of 81.41787%. So, Overall the 5-Fold method stands out among the three methods.

MclustDA - EDDA

The error rate for VSA model is 2.941176%, which gives an accuracy of about 97.05882% and thus shows a very good performance. The 5-Fold model gave an error rate of 2.976651%, which is also very good with accuracy of 97.02335% but slightly less than the VSA model above. So, overall the VSA method stands out among the three methods. For the following models we will add in the log transformation for the models LDA, SVM and Naive Bayes, which made the accuracy better.

LDA

The LDA model was completed with the VSA and 5-fold CV approach. The test error with VSA is .087 and with 5 fold CV is .088. We can say here that LDA performs a little better with VSA in this scenario.

SVM

For the SVM model the test error with VSA is .09 and with 5 fold CV is .10. We can say here that SVM is also performing a little better with VSA. The kernel used for this Model is `Linear` and the different cost values used are: .01,.02,.05,.1,.2,.5,1,2,5,10.

Neural Network

Next was a neural network algorithm because it is a classification problem and an advanced algorithm as compared to logistic regression. The test error with VSA is .36 and with 5 fold CV is .91. The neural network performs better with VSA in this scenario comparatively. Although in both the approaches it is not giving us satisfactory error rates. The size used in the model = 10.

Naive Bayes

Naive Bayes model, that I learned in Dr. Jun Liu's class at DSU, was also used. The test error with VSA is .142 and with 5-fold CV is .14. We can say here that Naive Bayes performs a little better with the CV approach here.

Linear Regression

There are no linear regression models for these datasets. This is because the nature of the dataset does not call in the need for a linear regression. It isn't appropriate due to the fact that a categorical response is needed, not a linear or numerical response.

Logistic Regression

Instead a logistic regression approach was used. The test error for VSA is .1012 which gives an accuracy of 89.88% while the 5-fold CV returned .1043 (89.57% accuracy).

Log Transformation Logistic Regression

The best VSA logistic regression model and logistic regression k-fold cross validation were also done on classifiers that were modified. These modifications included a log transformation as mentioned above. The test error for VSA is .0098, which gives an accuracy of 99% while the 5- fold CV returned again returned .0325, which gives 96.75% accuracy. This shows that performing a log transformation on our variables does in fact produce a better model, and thus result.

KNN

Our last model was K-Nearest Neighbor. The test error for VSA is .2901 and a 70.99% accuracy. While the 5-fold CV returned a much higher test error of .6711.

Unknown Dataset Prediction

We made a final prediction on the random forest model and looked at how it was working on the unknown dataset. It appears that there are different sessionIndex's in there, which identify a

session. The model predicted subjects for each and there was a mixture of results, some subjects were predicted multiple times.

Results

Comparison table for Test Error

Method	VSA	5- Fold CV
LDA	0.0882	0.0878
SVM	0.0976	0.1047
Neural Network	0.3621	0.9195
Naive Bayes	0.1426	0.1402
QDA	0.8431	0.4214
MclustDA	0.1737	0.1858
MclustDA-EDDA	0.0294	0.0298
Trees	0.4622	0.3222
Bagging	0.0000	0.0000
Random Forest	0.0000	0.0048
Multinomial Regression	0.1012	0.1043
Log Multinomial Regression	0.0098	0.0325
KNN	0.2901	0.6711

We chose random forest as the final model after looking at the results table. This performed top, outperforming bagging by rep error since session error was so similar. After choosing this as the final model we ran more analysis on how this was predicting. The model was then trained only on session 8, then tested on session 7 which yielded an accuracy of 86%. Then the model was trained on a 70/30 split of the session 8 itself which yielded a 96% accuracy rate. So we are averaging a 94% accuracy rate across all training methods we have used. The 70/30 split, among both sessions, seemed to yield the best results since it had a representation with enough reps for all the users to be predicted correctly. This shows me the amount of reps per user is a factor in how it predicts.

References:

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An introduction to statistical learning: With applications in R. New York: Springer.

2. https://stackoverflow.com/questions/35090883/remove-all-of-x-axis-labels-in-ggplot

3. https://beckmw.wordpress.com/2013/02/05/collinearity-and-stepwise-vif-selection/

4. https://gist.github.com/ramhiser/6dec3067f087627a7a85

5. https://stackoverflow.com/questions/6578355/plotting-pca-biplot-with-ggplot2

6. Referenced from prevous homeworks from STAT 701 and 702

7. Dr.Saunder's example code on LDA to find session Index error rate