Machine learning algorithms for anomaly detection on Computer-Based Testing

Soo Ingrisone, Pearson

James Ingrisone, Pearson VUE

Paper presented at the National Council of Measurement in Education (NCME) annual meeting, virtual: June 2021

Abstract

The machine learning (ML) algorithms for anomaly detection on CBT is explored. Hierarchical agglomerative clustering is used for automatically labelling unlabeled data. Random forest ensembles are used to evaluate the accuracy of the clustering. Actual data from a certification exam are used to validate ML classification results.

Keywords: Innovative research that gets implemented in practice and helps promote informed decisions, Credentialing (certificates, certification, licensure), Test Security

Test security is essential to make valid interpretations from test scores and its subsequent actions based on those interpretations. The *Standards for Educational and Psychological Testing* (AERA, APA, & NCME, 2014) refer broadly to the topic of test security in Standard 8.9:

> "Disclosure of confidential testing material for the purpose of giving other test takers advance knowledge interferes with the validity of test score interpretations; and circulation of test items in print or electronic form may constitute copyright infringement. In licensure and certification tests, such actions may compromise public health and safety. In general, the validity of test score interpretations is compromised by inappropriate test disclosure" (AERA, APA & NCME, 2014, p.136).

With the advancements in technology, there is an enhanced ability to collect and share test material which results in additional challenges to maintain the security of test content and to detect cheating behavior. Threats to the test security can occur before, during or after test administration. It could involve examinees, test administrators, other staff at the testing sites, testing program managers, and operations' vendors (Ferrara, 2017).

Stealing test forms or items pose major threats to the validity of test score interpretation and the credibility of large-scale assessment programs (Olson & Fremer, 2013). Item harvesting occurs when test takers collect test questions for future distribution. Test collusion is when two or more examinees deliberately share test content or answers to test items for potential gain in test performance before, during or after a test. Test collusion may include cases where test items are exposed by brain-dumping via website or social media; inappropriate test preparation is provided; and secure material is taught by instructors, teachers, and trainers. It also extends to proxy testing, using cheat sheets, and security breaches at the testing centers. (Maynes, 2017; Ferrara, 2017; Belov & Wollack, 2018). Studies have shown that the most common cause of

irregularities prior to testing is the compromise of test materials (Ferrara, 2017; U.S. Dept of Ed., ISE, & NCES, 2013).

In licensure and certification testing, threats to test security may be particularly concerning because it may impact public health and safety. In addition, individuals may be personally motivated to gain an advantage as test results can have real impact on jobs, promotions, or income (Ferrara, 2017).

Testing agencies should consider a comprehensive framework for dealing with test security issues that include prevention, detection, investigation, and resolution (Ferrara, 2017). Testing irregularities can have far-reaching consequences, making early detection of test security breaches essential to preventing widespread compromise (US Dept of Ed., ISE, & NCES, 2013). Therefore, assessment programs need to be proactive about test security issues to promote confidence and ensure integrity in the overall testing program (Olson & Fremer, 2013).

In general, claims that a test security violation has taken place and subsequent action is justified cannot be adequately supported by any one piece of evidence (Kingston and Clark, 2014). Thus, multiple data forensic methods are suggested to corroborate findings where groups of aberrant test takers have been detected (Cizek & Wollack, 2017). Machine learning is one of the various methods that can be used to detect unusual responses. According to Kim et al (2017) and Wollack and Maynes (2017), clustering analysis used in machine learning algorithms appears promising to detect group collusion and suggested more research in this area.

Machine learning is a collection of methods that enable computers to automatically learn and improve their predictions over time from data. There are two types of machine learning (ML) learning: unsupervised and supervised learning. The unsupervised machine learning algorithm finds inherent clusters of data points. In other words, it can review large volumes of

data and discover specific trends and patterns that would not be apparent to humans. The supervised machine learning algorithm trains a model by using labeled data to classify data or predict outcomes. As a fully trained ML algorithm gains experience with data, it keeps improving in accuracy and efficiency with speed of information processing that greatly surpasses that of humans (Mitchell, 2017; Molnar, 2020). From fraud and security threat detection to flagging abnormalities in healthcare imaging data, there are countless business applications for automatic identification of abnormal data (Hastie, et al. 2017). Consequently, ML algorithms show potential for detecting aberrant groups in licensure and certification testing.

In an era of data-driven decision making, the purpose of this study is to explore how machine learning (ML) algorithms can be employed to detect group collusion in Computer-Based Testing (CBT). During a test administration window, a group of test items on a certification exam were wrongfully harvested and shared as exam preparation material. The stems, options, and keys for over 100 questions were discovered, and about 20% of the questions were found to be mis-keyed which is not uncommon. Collusion had occurred and widespread content disclosure was suspected as it appeared that pass rates increased dramatically. Consequently, this research is to investigate how well ML algorithms detect collusion.

This study is divided into the three parts: In Part I, the clustering analyses results based on machine learning algorithms, such as hierarchical agglomerative clustering (HAC), are examined. The characteristics of the clusters are explored. In Part II, random forest ensembles are used to validate the recovery of the clusters from HAC. Also, important features contributing to the model predictions are reviewed. In Part III, the collusion cluster detected by HAC is compared with the potential collusion groups identified by the exact response matches on incorrectly keyed items in the exam preparation material. The level of overlap between the two

methods are compared to examine how well the clustering algorithm identified members of the collusion group.

Methods

In this study, both unsupervised machine learning and supervised machine learning algorithms are utilized on data of about 1,000 candidates that took a certification exam. Studies have found that the predictive accuracy of classes can be improved by leveraging clustering techniques before applying classification algorithms on data (Gao, et al, 2013; Chakraborty, 2014). The analysis is performed using the Scipy and Scikit-learn libraries from the programming language Python 3.7 via JupyterLab. The following steps are carried out in the study:

Part 1. Hierarchical agglomerative clustering (HAC)

Unsupervised machine learning takes a dataset with no labels and attempts to find some latent structure within the data. Hierarchical agglomerative clustering (HAC) is known to excel at discovering embedded structures/clusters in the data when we do not know in advance how many clusters we want. The goal of cluster analysis is to partition the observations into distinct groups (“clusters”) so that the observations within each group are quite similar to each other. HAC is one such algorithm and it is a bottom-up approach. Starting with each observation as its own cluster, at each step the closest pair of clusters is merged as one and moves up the hierarchy until one cluster remains. It produces a tree-like visual representation called a dendrogram which represents a graphical summary of a series of joins resulting from the data clustering (Hastie, et al. 2017).

Distance metric: Gower's similarity coefficient

HAC requires defining the distance metric that can be used to measure the proximity between two observations. The current dataset contains mixed types, such as continuous and binary data types. Therefore, a distance metric that is capable of handling different types of variables is needed. The standard Euclidean measures of distance are inappropriate for assessing the dissimilarity between two observations because the variables of interest are not continuous in nature (Finch, 2005). Gower's distance metric is specified in this study. The strength of Gower's distance metric is that it is a composite measure. It computes the similarity by dividing features into two subsets, one for categorical features and the other for numeric features. For continuous variables, Manhattan distance is used and for binary variables, Jaccard coefficient is used to calculate the distance matrix. In essence, it is a weighted average of the distances on the different variables. It is scaled to fall between 0 and 1 (Gower, 1971).

Clustering algorithm selection: Ward's method

Once the distance matrix is derived according to Gower's distance metric, the Cophenetic Correlation Coefficient (CPCC) is used to evaluate which type of hierarchical clustering technique is best for the data. It is a measure of how accurately the dendrogram represents the dissimilarities among data points, thus the goodness of fit of the clustering. The cophenetic distance between two data points is represented in a dendrogram by the height of the link at which those two data points are joined. That height is the distance between the two subclusters that are merged by that link. The value close to 1 indicates a good fit of the clustering to the data. In the study, CPCC is used to compare alternative clustering solutions by four different clustering techniques, single, complete, average and Ward's method. Previous studies have shown that in a mixed-type data context, Ward's method was capable of correctly clustering

observations (Hummel, et al, 2017; Miyamoto, et al, 2015). Therefore, Ward's method is selected for the calculation of between-cluster distances.

Ward's method (see Equation 1) assumes that a cluster is represented by its centroid, but it measures the proximity between two clusters in terms of the increase in the error sum of squares (SSE) that results from merging the two clusters into a single cluster. In other words, it attempts to choose the successive clustering steps so as to minimize the increase in error sum of squares at each step (Hand, et al., 2001).:

$$\Delta(\mathrm{A},\mathrm{B}) = \sum_{i\in A\cup B} \|\vec{x_i} - \vec{m}_{A\cup B}\|^2 - \sum_{i\in A} \|\vec{x_i} - \vec{m}_A\|^2 - \sum_{i\in B} \|\vec{x_i} - \vec{m}_B\|^2 = \frac{n_A n_B}{n_A + n_B} \|\vec{m}_A - \vec{m}_B\|^2 \tag{1}$$

where $\vec{m}_j$ is the center of cluster *j*, and n_j is the number of points in it. Δ is called the merging cost of combining clusters *A* and *B*.

Number of clusters: Silhouette coefficient

Since the "ground truth" labels of groups are not known in the current data, Silhouette coefficients are used to determine an optimal number of clusters for HAC. Silhouette analysis provides the separation distance between the clusters. The Silhouette Coefficient for a point *i* is defined as follows:

$$S(i) = \frac{b(i) - a(i)}{\mathrm{max\,a(i),b(i)}} \tag{2}$$

where *b(i)* is the smallest average distance of point *i* to all points in any other cluster and *a(i)* is the average distance of *i* from all other points in its cluster.

The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess the optimal number of clusters visually. The silhouette coefficient value ranges from -1 to 1. The value near 1 indicates that the

sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster. The resulting clusters are assumed to be the observed labeled classes in the following classification model.

The clusters are visualized via t-Distributed Stochastic Neighbor Embedding (t-SNE). It visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map (van der Maaten & Hinton, 2008).

Part 2. Random Forest

The classification accuracy for each of the classes identified by HAC are evaluated and validated by random forests. Random forests are simple and easy to use, and often perform well. It requires data with a categorical dependent variable (target variable) and a set of independent variables (features) which are used in predicting the classes. It is attractive because it handles mixed-type variables very well and can capture complex interaction structures in the data (Hastie, et al. 2017). Random forests build a large collection of de-correlated trees by using a random subset of predictors at each split, and then averages them. Thus, it makes the average of the resulting trees less variable and more reliable. When used for classification, a random forest takes a class vote from each tree, and then assigns (predicts) the class using majority vote. This process is referred to as ensemble techniques because they use a collection of results to make a final decision. It generally has much better predictive accuracy than a single decision tree (Breiman, 2001).

K-fold Cross Validation

Cross validation is a technique for assessing the performance of machine learning models on unseen data. Namely, it uses a limited sample to estimate how the model is expected to

perform when used to make predictions on data not used during the training of the model. It is a popular method because it generally results in a less biased estimate of the model performance.

The data are divided into a 70/30 split resulting in a train and test set, respectively. During the modeling phase, a repeated stratified k-fold cross-validation (CV) process is applied on the train set. This k-fold CV process partitions the training data set into k equal subsets where one-fold is treated as a validation set. The machine learning model (random forest) is fit on the remaining $k - 1$ folds. Then, the model is used to make predictions of the hold-out fold (validation set) and calculates the mean squared error (MSE) on the validation set. The MSE is to measure how well the class predictions made by the model match the observed class that was not previously seen. The closer the model class predictions are to the observed class, the smaller the MSE will be. The MSE as a model evaluation score is retained and the model is discarded. This procedure is repeated k times by treating a different k^{th} group as a validation set with k estimates of the MSEs. The k-fold CV estimate is computed by averaging these values (James, et al., 2013). In this study, three repeats of 10-fold cross-validation are applied to the train set, i.e., 30 different models are fitted on the train set and evaluated. Thus, the mean and standard deviation of the MSE across all repeats and folds are observed to examine the trained model accuracy. Next, a fully trained machine learning model is applied to the test set to make class predictions from the new instances. Classification performance evaluation measures are inspected to determine the level of agreement between the predicted classes and observed classes.

Feature importance selection: SHAP

Furthermore, the magnitude of feature attributes is examined via SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2016). In SHAP, features with large absolute Shapley values are considered important. Shapley values indicate the contribution of each feature

to the prediction. In this study, tree SHAP algorithms are employed to explain the output of ensemble tree models. Tree SHAP is a fast and exact method to estimate SHAP values for tree models and ensembles of trees, under several different possible assumptions about feature dependence (Lundberg, 2018). The sum of absolute Shapley values per feature across the data is obtained as the global importance for a classification random forest (see Equation 3).:

$$I_j = \sum_{i=1}^{n} \left| \phi_j^{(i)} \right| \quad (3)$$

where ϕ_j are the Shapley values. The selected important features are investigated to understand the nature of the most contributing features to the random forest trained model for predicting the classes.

Part 3. Collusion Agreement

The collusion cluster detected by HAC is compared against the potential collusion identified by the group of candidates with exact response matches on the incorrectly keyed items appearing in the exam preparation material. About 20% the items found in the exam preparation material are mis-keyed These mis-keyed items are utilized to validate the collusion cluster results. An obvious indication of test collusion among test takers is that they have unusually high response matching on items involved in the collusion (Belov & Wollack, 2018). Also, studies articulated that consistent incorrect response patterns reflect some evidence of wrongdoing, such as working together before the exam to share test content (i.e. acquiring pre-knowledge) or receiving disclosed answers to test questions by instructors, teachers, and trainers (Maynes, 2017; Holland, 1996; Bellezza & Bellezza, 1989). Therefore, the incorrectly keyed items are used to uncover candidates who may have benefitted from the exam preparation material. Candidates whose item responses matched exactly with the incorrectly keyed items in the exam preparation material were identified as potentially being part of the collusion group. The

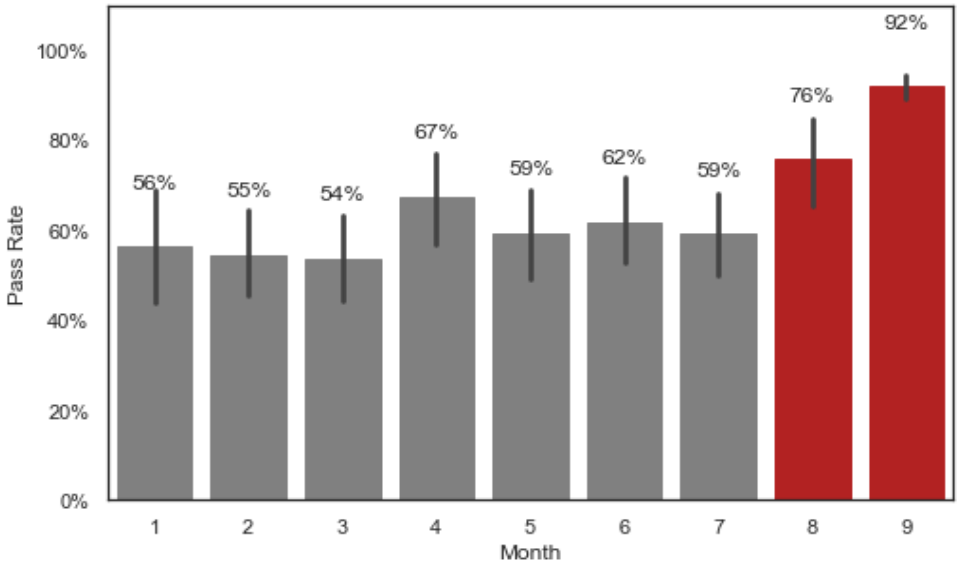
agreement rate between the collusion groups identified by HAC and by exact response matches on the incorrectly keyed items are investigated.

Results

Preliminary analysis

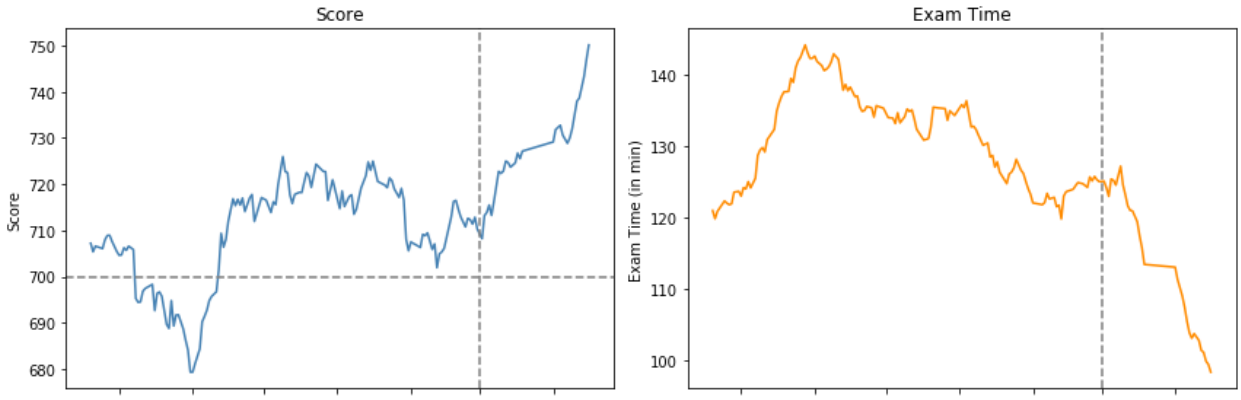
Figure 1 displays the monthly pass rates over the entire administration window. The pass rates for the first three months range from 54% to 56%. There is a noticeable increase in month 4 to 67% followed by a drop in months 5-7 with pass rates ranging from 59% to 62%. A substantial jump in pass rates appears to emerge in months 8 and 9, 76% and 92%, respectively. Given that the average pass rate for the previous year's administration was about 66%, and approximately 60% for the current administration months 1-7, the pass rates in months 8 and 9 look unusually high and warrant additional investigation as they suggest that a group of candidates may have engaged in collusion.

Figure 1. Pass Rate by Month



Moving averages of scores and exam time are displayed in Figure 2. The horizontal line in the Figure 2 score plot refers to the passing scaled score of 700. The vertical lines of both score and exam time plots in Figure 2 indicate the 8th month of the testing window where unusual patterns generally appear to emerge: Starting in month 8, the moving average of scores noticeably starts increasing, while the moving average of the exam time clearly begins decreasing. These unusual scores and times are consistent with collusion behavior.

Figure 2. Moving Average of Scores and Exam Time



Forensic analysis using machine learning methods is employed as clustering analysis used in machine learning algorithms show potential for detecting group collusion.

Part I. Hierarchical Agglomerative Clustering (HAC)

Four different types of commonly used hierarchical clustering techniques were employed, i.e., single, complete, average, and Ward's method. The hierarchical structures produced by the four techniques are evaluated using the cophenetic correlation coefficient (CPCC). In Table 1, the CPCC reveals that the hierarchical clustering produced by single and complete linkage techniques fit the data less well than the average linkage technique or Ward's method. Therefore, either technique may be chosen to perform the clustering. However, since Ward's method has

been shown to cluster observations correctly in the mixed-type data context, Ward's method is chosen.

Table 1. Cophenetic Correlation Coefficient (CPCC) and Four Agglomerative Hierarchical Clustering Techniques

Technique	CPCC
Single Link	0.499
Complete Link	0.640
Average Link	0.750
Ward's Method	0.731

Based on the inspection of the Silhouette coefficient curve in Figure 3, k = 2 clusters exhibit the highest value at 0.469 and suggest the optimal number of the clusters for the data. Consequently, two clusters are accepted as the final solution from HAC.

Figure 3. Silhouette Coefficient Plot by Number of Clusters

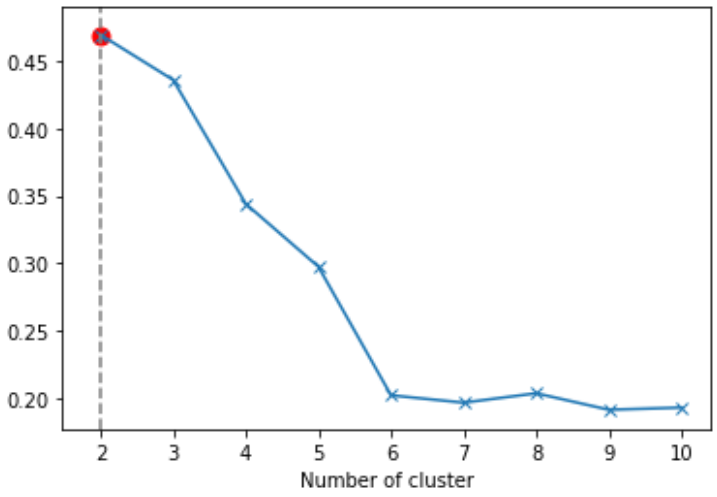
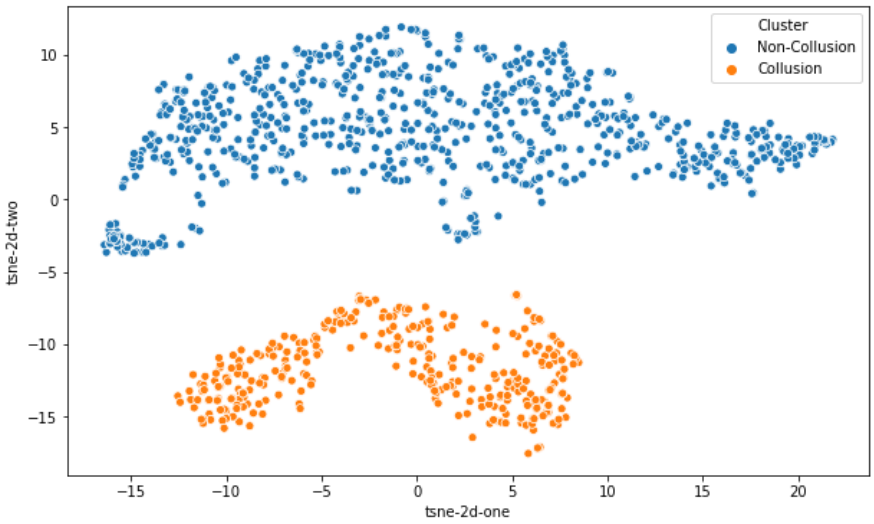


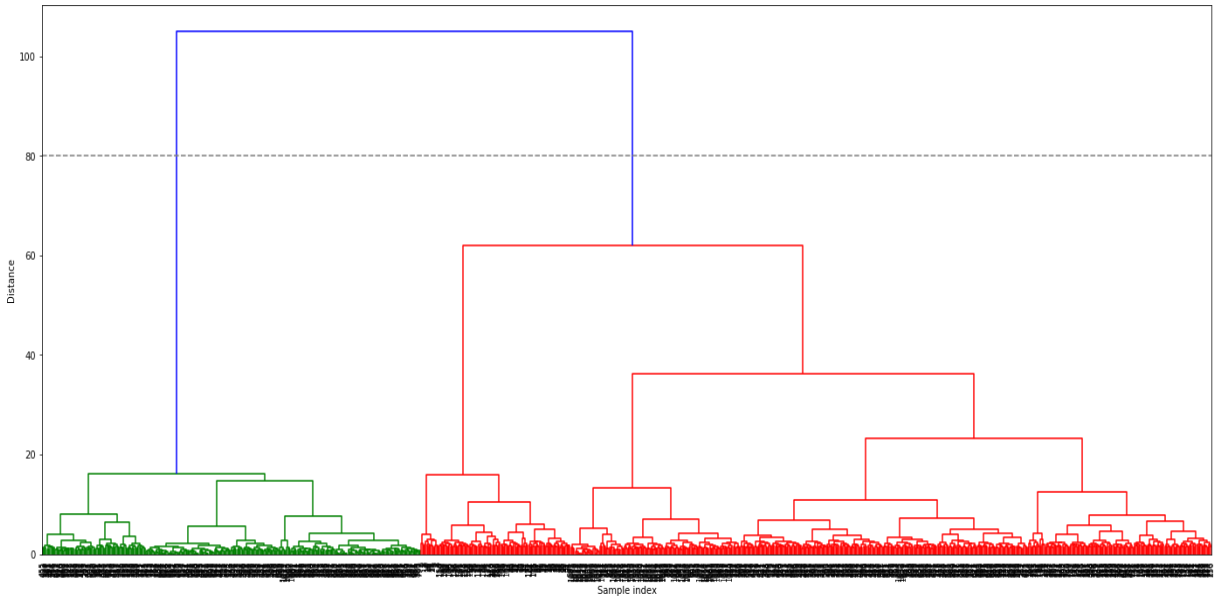
Figure 4 presents a visualization of the clusters via t-Distributed Stochastic Neighbor Embedding (t-SNE). The t-SNE plot reveals two well-separated clusters that HAC detected in the data.

Figure 4. Scatterplot of t-SNE



The tree representing the hierarchical merging of clusters is visualized as a dendrogram in Figure 5. The dendrogram shows a graphical summary of the clustering structure of the data. The location of the horizontal line through the dendrogram shows where the "tree is cut" and represents the two clusters that emerged from the data.

Figure 5. Dendrogram



Next, the characteristics of the two clusters found by HAC, collusion and non-collusion are examined. Table 2 shows score and exam time descriptive statistics for each cluster. The

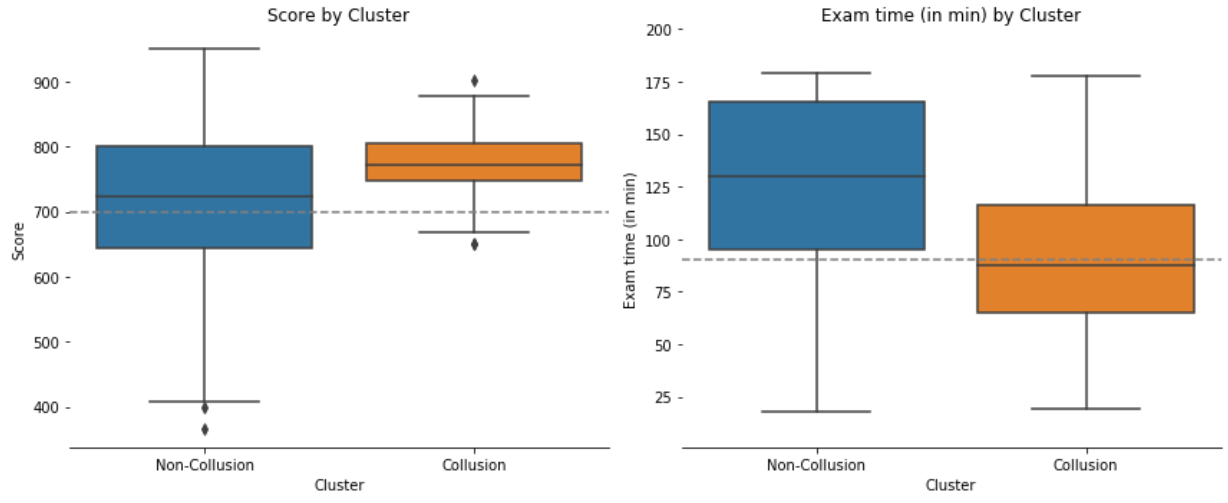
mean scaled score of the collusion group is significantly higher than that of the non-collusion group (t=12.433*, p < .001), and the average exam time in minutes is significantly faster for the collusion group than for the non-collusion group (t=14.353*, p < .001).

Table 2. Descriptive Statistics of Score and Exam Time

	Cluster	Min	25th	Median	75th	Max	Mean	SD
Score	Collusion	651	748	772	805	902	773.98	40.99
	Non-Collusion	367	643	724	801	951	716.30	110.96
Exam Time	Collusion	19.42	64.88	87.54	116.15	117.95	93.32	35.62
	Non-Collusion	18.25	95.22	129.93	165.31	179.13	127.49	38.94

The boxplots in Figure 6 depict both the score distribution and exam time distribution for each cluster described in Table 2 above. The horizontal lines cross 700, the scaled score cut, and 90 minutes, half the exam time, respectively. The boxplots graphically illustrate the distinct score and exam time distributions of each cluster. The median score of the collusion cluster is substantially higher than that of the non-collusion cluster (772 vs. 724), and the score range is much narrower. Furthermore, the median exam time of the collusion cluster is substantially faster than that of the non-collusion cluster (87.54 vs. 129.93).

Figure 6. Boxplot of Score and Exam Time per Cluster



* Significance at 0.05 α – level.

Figure 7 presents the kernel density estimate plots by score and exam time and reveals two very different cluster contours. For the non-collusion group, the shape of the dispersion of score and exam time is consistent with what is typically seen in licensure and certification testing. That is, a wide range of scores along with examinees using most of the exam time. However, for the collusion group the shape of the dispersion is unusual. For the collusion group, the exam time values are concentrated around 90 minutes or less whereas, for the non-collusion group, the values are concentrated around 165 minutes. In terms of the score values, they are concentrated around 750 to 825 for the collusion group and around 650 to 800 for the non-collusion group. Higher scores with narrow score ranges and faster times are typical signals for collusi*on* group behavior.

Figure 7. Kernel Density Estimate Plots for Score and Exam Time per Cluster

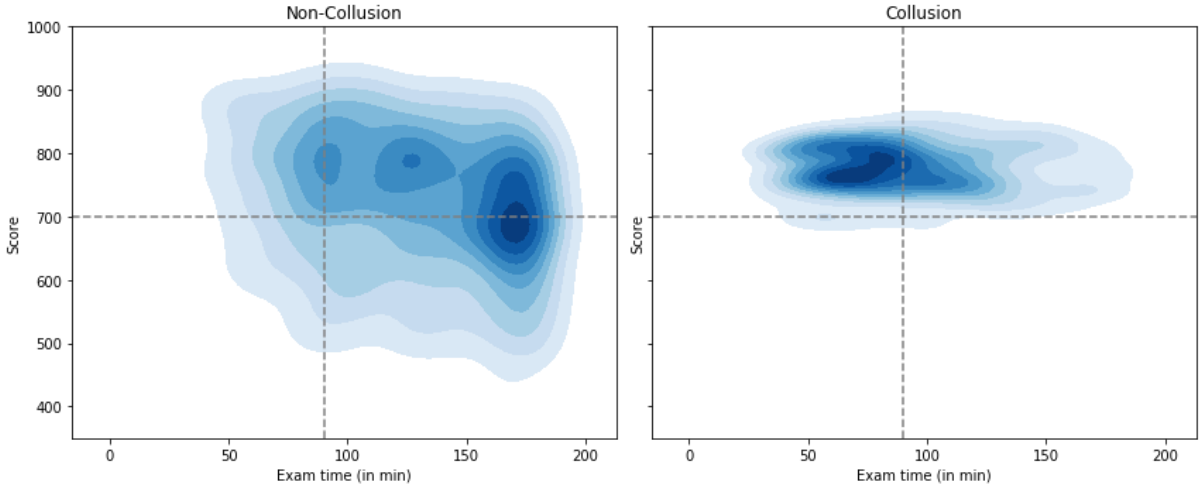
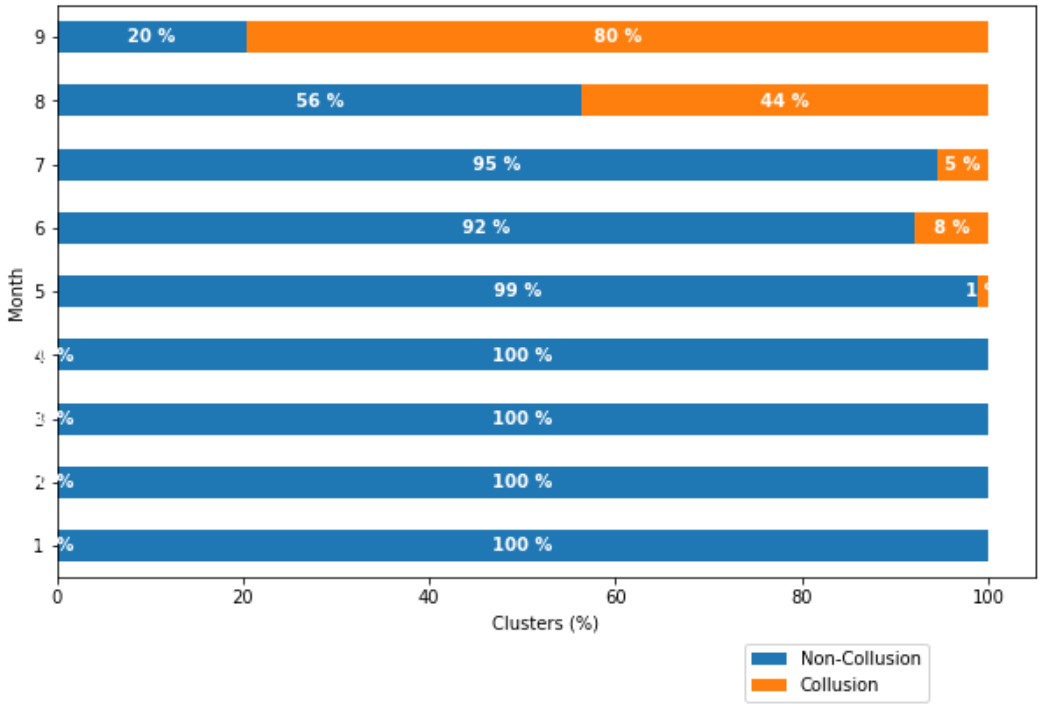


Figure 8 shows the clusters by month. In the first four months of testing, only the *non*-collusion group exists. However, in months five through seven the collusion group slowly emerges (1%, 8%, and 5%, respectively). By month eight a rapid increase is noticed where 44% of the testers are in the collusion group. This dramatic rise continues with 80% of the examines identified as members of the collusion cluster in month nine. The results suggest that the exam

preparation material may have been narrowly available starting in month five and more widely available in months eight and nine.

Figure 8. Clusters by Month

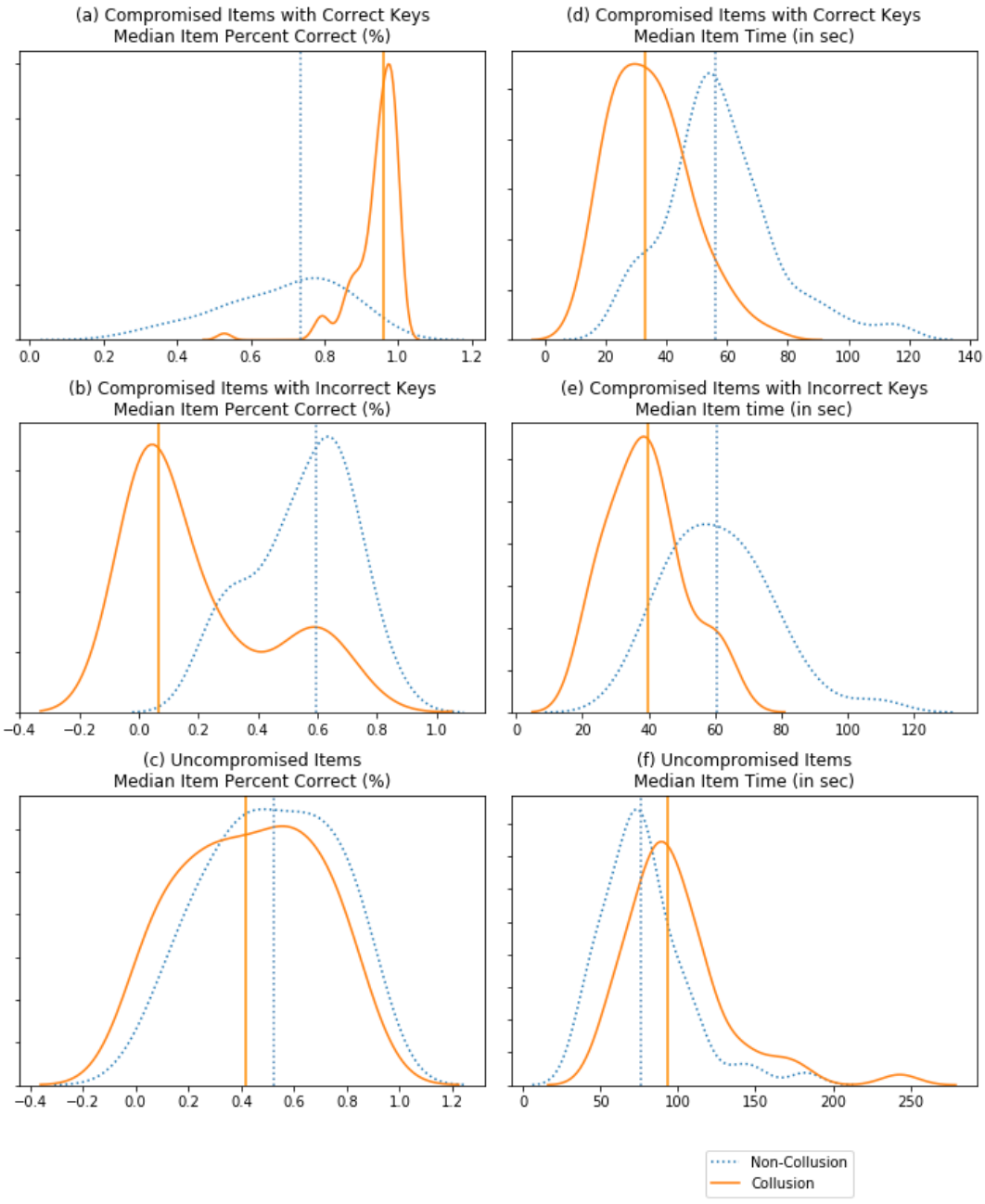


As previously mentioned, exam preparation material was discovered. While it contained over 100 compromised items, over 50 items on the exam were not compromised. Regarding the compromised items approximately 20% were incorrectly keyed. Figures 9 (a) – (f) display how each cluster group performed on the compromised and uncompromised items and provide additional evidence that the collusion group benefitted from the available exam preparation material.

Although the collusion group performed exceptionally well on the correctly keyed compromised items (a), the non-collusion group outperformed them on both the incorrectly keyed compromised items (b) and the uncompromised items (c). In terms of median item response time, the collusion group answered both the correctly (d) and incorrectly (e) keyed compromised items substantially faster than the non-collusion group but responded slower on the

uncompromised items (f). This shows that when the collusion group had access to an answer key they performed very well when the key was correct and poorly when it wasn't, and they responded more quickly whether the key was flawed or not. However, when they came across an item they had not seen before they took longer to respond and did not do as well as the non-collusion group.

Figure 9. Compromised Items vs Non-compromised Items Performance by Cluster



Part II. Random Forest

The random forest ensembles are used to validate the recovery of the clusters derived from HAC. The data are divided into a 70/30 split resulting in a train and test set, respectively. During the modeling phase, the train model accuracy using train set is 0.989 and its associated standard deviation is 0.015. The model has learned how to predict the classes with 98.9% accuracy. The mean absolute error (MAE) and the mean squared error (MSE) were 0.009 and 0.009, respectively. Namely, the average estimate is off by 0.009 degrees. A single tree from the forest is exhibited in Appendix A to illustrate how a trained model predicts classes.

The obtained model accuracy (f1-score) using the test set was 0.991. The model has predicted the classes with 99.1% accuracy. The average estimate is off by 0.009 degrees (MAE). The obtained MSE, RMSE and Hamming loss were 0.009, 0.096, and 0.009, respectively. Using the selected evaluation measures, Table 3 shows how well the random forest model predicted the classes. All measurement values are high and close to 1, indicating that the model predicted the classes well and obtained a near optimal result. It validates that the two clusters derived from HAC satisfy the assumption that members belonging to the same class are more similar than members belonging to a different class.

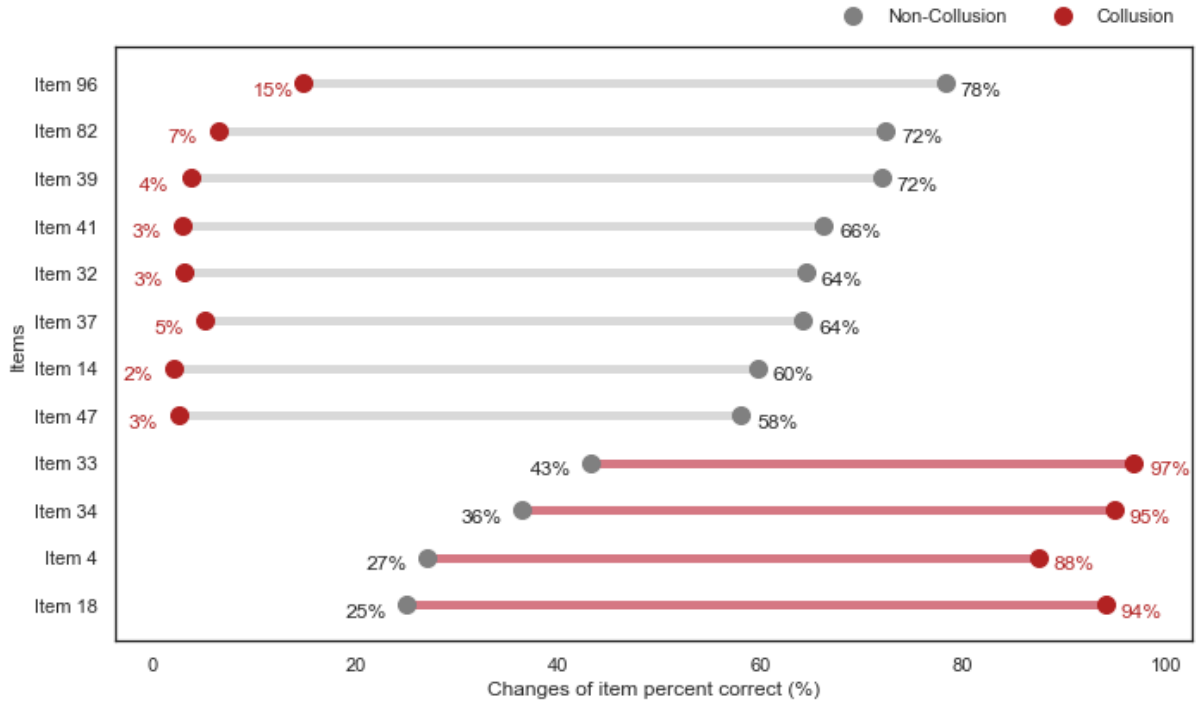
Table 3. Classification Performance Evaluation

Measure	Value
Adjusted Rand Index	0.963
Adjusted Mutual Info	0.918
Normalized Mutual Info	0.918
Homogeneity Score	0.917
Completeness Score	0.920
V-measure Score	0.918
Fowlkes-Mallows Score	0.984

The plot in Appendix B, based on SHAP, ranks the importance of the variables that contributed to the random forest model from most to least. The advantage of using SHAP feature

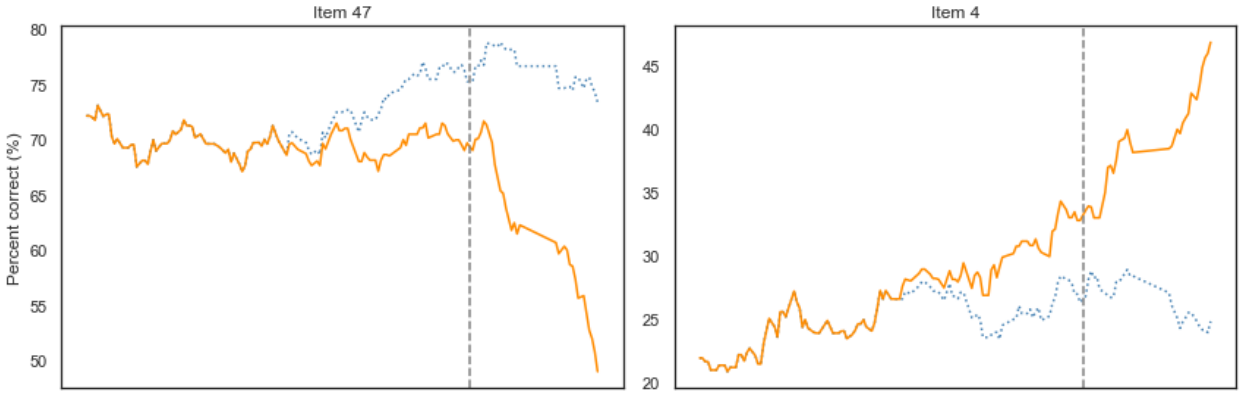
importance is that it identifies the variables that contribute most to class assignment. In Figure 10, the 12 items with the highest contribution to the model are examined. The percent correct of each item is computed separately for the collusion and non-collusion groups. The results show the stark difference in item performance between groups. The first eight items in Figure 10 illustrate group performance on the compromised items with incorrect keys where performance differences range from 55% to 68%. For example, for item 96, the item proportion correct for the collusion group was 15% whereas it was 78% for non-collusion group. The next 4 items in Figure 10 illustrate group performance on the compromised items with correct keys where performance differences range from 54% to 69%. For example, for item 18, the item proportion correct for non-collusion group was 25% whereas it was 94% for the collusion group. Based on the SHAP feature importance, it is clear that item performance differences between the two groups play an important role in the class assignment and items with the largest disparities contribute the most to the model.

Figure 10. Item Percent Correct Comparison by Groups



Out of 12 items above, two items are selected as an example to exhibit the trend of their percent correct moving averages across the administration window, i.e., items 47 and 4. In Figure 11, the vertical lines represent the 8th month of the administration window. The blue dotted line shows the percent correct moving average for the non-collusion group only, and the orange solid line shows the percent correct moving average including both the non-collusion and collusion groups. The item trends presented in Figure 11 illustrate the impact the collusion group has on the item performance. Consistent with previous results, it appears that the collusion group's influence started in about month 5 and increased substantially in months 8 and 9. This provides further evidence that the collusion group affects test scores and the validity of test score interpretations. Furthermore, the members of the collusion group should be removed from the item calibration.

Figure 11. Item Trend



Part III. Collusion Agreement

Belov & Wollack (2018) noted that a clear signal of test collusion among test takers is that they have unusually high response matching on items involved in the collusion. Furthermore, consistent incorrect response patterns reflect some evidence that test takers engaged in test collusion (Maynes, 2017). Therefore, the incorrectly keyed items found in the

exam preparation material are used to identify test takers that have uncommonly high exact response matching and may have benefitted from the exam preparation material.

The level of overlap between the collusion group derived from HAC and the candidates that have exact response matches on the incorrectly keyed items is examined. The collusion group consists of 352 candidates where 336 passed the exam. The heatmap in Figure 12 displays the magnitude of the collusion group members by score and exam time. The intensity of the observations is depicted by numbers in a cell as well as by the color saturation.

Figure 12. Heatmap of HAC Collusion Group by Score and Exam Time (N=352)

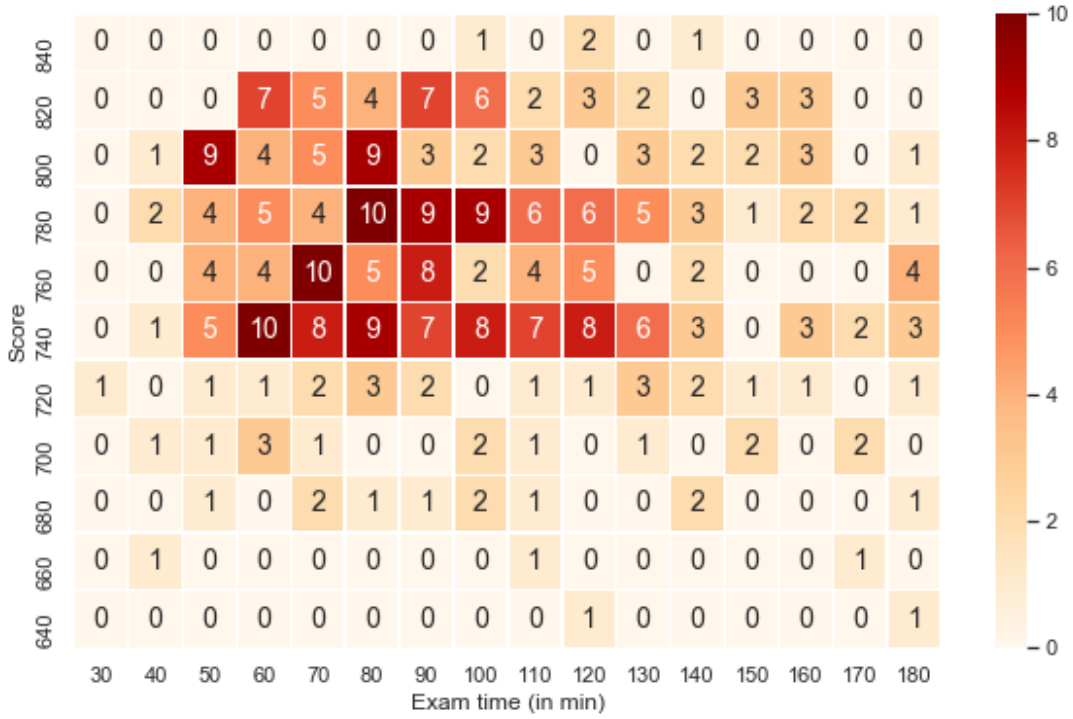


Table 4 shows the exact response match rate on the incorrectly keyed items from 50% to 100% and the number of candidates in each associated exact response match (ERM) group. It also includes the total overlap and agreement rate between the collusion group and each ERM group as well as the overlap and agreement rate for those that passed the exam. For example, the ERM group corresponding to 67% exact response match includes 349 candidates where the total

overlap between the ERM group and collusion group is 340 candidates, which is 97% agreement. In addition, out of 349 ERM candidates, 324 passed the exam. All of these 324 candidates are members of the collusion group, resulting in 100% overlap. The overlap agreement rate between the collusion group and each ERM group ranges from 94% to 100% for all test takers, and 99% to 100% for those that passed the exam. The level of overlap illustrates how well the HAC algorithm identified members of the collusion group. In this case, HAC combined with exact response match rates on the incorrectly keyed items provides a range of collusion group options for authorized bodies to consider when contemplating disciplinary action and sanction enforcement. HAC is a promising technique to help authorized bodies resolve test security violations.

Table 4. Overlap and Agreement Rate between HAC and ERM

Exact Response Match Rate on Incorrectly Keyed Items (%)	ERM Group (N)	Total Overlap of ERM and HAC (N)	Agreement Rate (%)	Total Overlap of ERM and HAC Pass Only (N)	Agreement Rate Pass Only (%)
50%	375	352	94%	336	99%
55%	365	352	96%	336	100%
60%	359	347	97%	331	100%
65%	351	342	97%	326	100%
67%	**349**	**340**	**97%**	**324**	**100%**
70%	333	327	98%	312	100%
75%	319	315	99%	301	100%
80%	302	299	99%	285	100%
85%	276	275	100%	266	100%
90%	224	224	100%	221	100%
100%	60	60	100%	60	100%

Discussion

Test security is crucial for valid interpretations and fairness of test scores. Studies have shown that the most common responses by licensure and certification organizations to test security violations are conducting security investigations and invalidating test scores (ATP, 2015). Machine learning (ML) algorithms are useful for providing data forensic evidence to support these decisions. Furthermore, as part of regular test security monitoring, ML algorithms could help identify trends before other evidence emerges.

This study used machine learning (ML) algorithms to examine candidate response data to detect test taker collusion. The results show that ML algorithms are quite promising for aberrant group detection. More specifically, a combination of random forest ensembles and exact response matches on incorrectly keyed items provided evidence that ML using hierarchical agglomerative clustering (HAC) algorithms detected the collusion group quite effectively. HAC is a useful approach for collusion group identification which may then lead to further investigation and possible invalidation of test scores.

While ML algorithms are extremely powerful and promising, there are some potential limitations. For example, HAC requires large storage space, and can be computationally intensive. This is especially true when working with big data. In addition, it is unfeasible to build models with ML algorithms using data with missing values. To overcome this disadvantage, it is recommended to impute values or perform listwise deletion (Hastie, et al., 2017). Furthermore, the ideal use case for clustering involves continuous data. However, item- and exam-level data are mixed-type data where continuous and categorical features coexist. Applying clustering techniques to mixed-type data is complex.

There are multiple clustering techniques that have been successfully employed in anomaly detection. In future studies, the performance across various clustering methods will be compared to determine which methods are most efficient for detecting aberrant behavior under different conditions. The studies could be extended to examine techniques that require less storage space, are less computationally intensive, and that can best handle missing data and mixed-type data.

References

American Educational Research Association (AERA), American Psychological Association (APA), and National Council on Measurement in Education (NCME) Joint Committee on Standards for Educational and Psychological Testing. (2014). *Standards for Educational and Psychological Testing*. Washington, D.C.: American Educational Research Association.

Association of Test Publishers (ATP). (2015). *Association of Test Publishers security report*: November 2015. Washington, DC: Association of Test Publishers.

Bellezza, F. S. & Bellezza, S. F. (1989). Detection of cheating on multiple-choice tests by using error-similarity analysis. *Teaching of Psychology,* 16, 151–155.

Belov, D. I. & Wollack, J. A. (2018). *Detecting groups of test takers involved in test collusion as unusually large cliques in a graph*, Law School Admission Council Research Report 18-01.

Breiman, L. (2001). Random forests, *Machine Learning*, 45(1), 5 – 32.

Chakraborty, T. (2014). EC3: combining clustering and classification for ensemble learning, *Journal of Latex Class Files*, 13(9). 1-14.

Cizek, G.J. & Wollack, J.A. (2017), *Handbook of Quantitative Methods for Detecting Cheating on Tests*, New York, NY: Routledge.

Ferrara, S. (2017). A framework for policies and practices to improve test security programs: prevention, detection, investigation, and resolution (PDIR), *Educational Measurement: Issues and Practice*, 36 (3), pp. 5-23.

Finch, H. (2005). Comparison of distance measures in cluster analysis with dichotomous data, *Journal of Data Science*, 3, 85-100.

Gao, J., Liang, F., Fan, W., Sun, Y., & Han, J. (2013). A graph-based consensus maximization approach for combining multiple supervised and unsupervised models, *IEEE TKDE*, 25(1), 15–28.

Gower J.C. (1971). A general coefficient of similarity and some of its properties, Biometrics, 27, 857-872.

Hand, D., Manilla, H. & Smyth, P. (2001). *Principles of data mining*, The MIT press.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of statistical learning, data mining, inference and prediction* (2nd Ed.), New York, NY: Springer.

Holland, P. W. (1996). *Assessing unusual agreement between the incorrect answers of two examinees using the K-index*. ETS Research Report No. 96–7, Princeton, NJ: ETS.

Hummel, M., Edelman, D., & Kopp-Schneider, A. (2017). Clustering of samples and variables with mixed-type data, *PLoS One*. 12(11).

Impara, J., Kingsbury, G., Maynes, D. & Fitzgerald, C. (2005). *Detecting Cheating in Computer Adaptive Tests Using Data Forensics*, Paper presented at the 2005 Annual Meeting of the National Council on Measurement in Education and the National Association of Test Directors, Montreal, Canada

Lundberg, S. M., & Lee, S. (2017). *A unified approach to interpreting model predictions*. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Lundberg, S. (2018). *SHAP*, https://shap-lrjball.readthedocs.io/en/docs_update/index.html

Maynes, D. D. (2017). Detecting potential collusion among individual examinees using similarity analysis, In G. J. Cizek & J. A. Wollack (Eds.), *Handbook of Quantitative Methods for Detecting Cheating on Tests*, (pp. 47–69). New York, NY: Routledge.

Mitchell, T.M. (2017). *Machine learning*, New York, NY: McGraw Hill.

Miyamoto, S., Abe,R., Endo, Y., Takeshita, J.(2015). *Ward method of hierarchical clustering for non-Euclidean similarity measures*, Paper presented at 2015 Seventh International Conference of Soft Computing and Pattern Recognition.

Molnar, C. (2020). *Interpretable machine learning*, Leanpub book.

Olson, J., & Fremer, J. (2013). *TILSA Test Security Guidebook: Preventing, Detecting, and Investigating Test Security Irregularities*. Washington, DC: CCSSO.

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12, pp. 2825-2830.

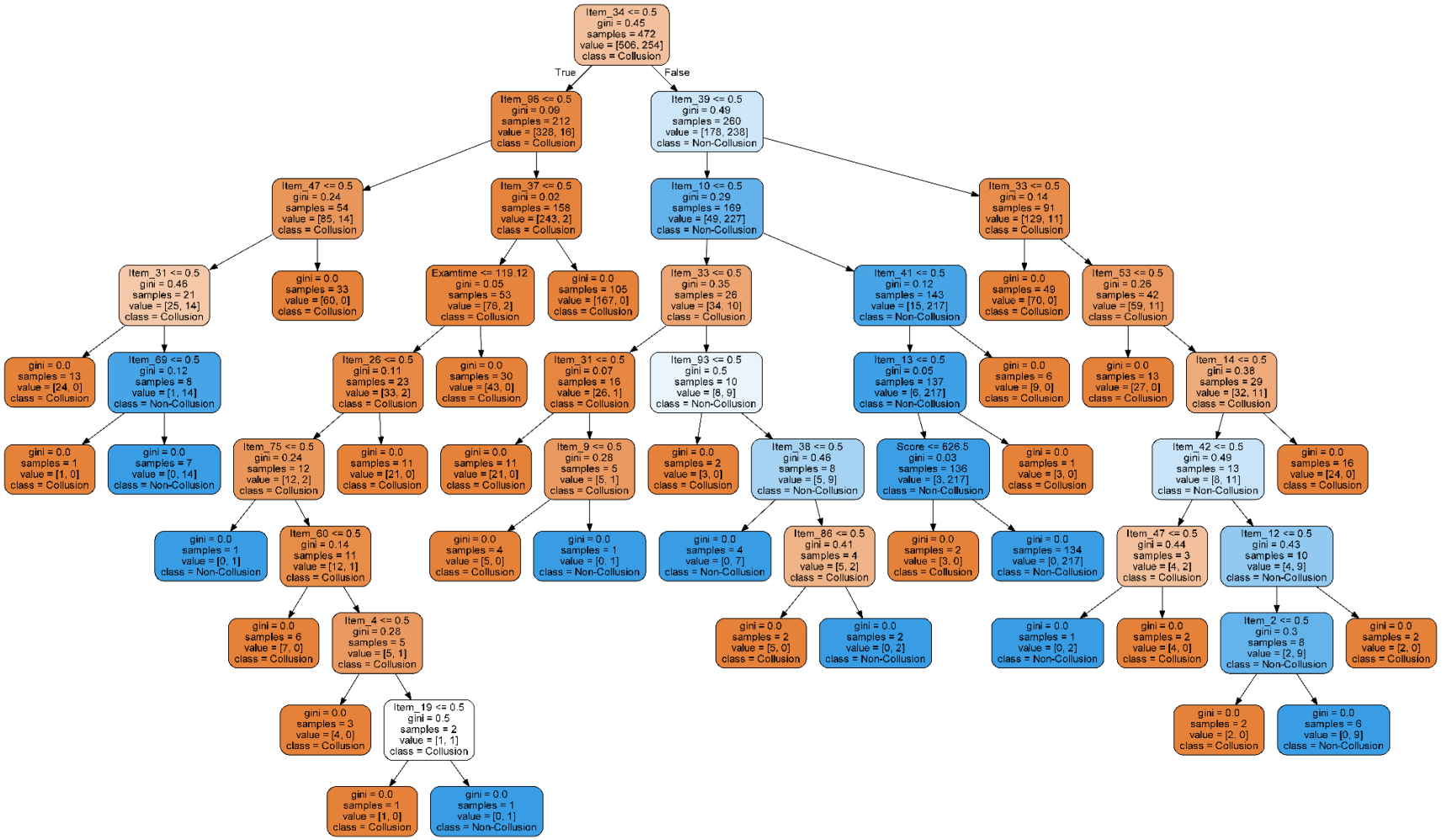
U.S. Department of Education, Institute of Education Sciences, National Center for Education Statistics (2013). Testing integrity symposium: Issues and Recommendations for Best Practice, October 2012.

Van der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE, *Journal of Machine Learning Research*, 9, pp. 2579—2605.

Wollack, J. A., & Maynes, D. (2017). Detection of test collusion using cluster analysis. In G. J. Cizek & J. A. Wollack (Eds.), *Handbook of quantitative methods for detecting cheating on tests* (pp. 124–150). New York, NY: Routledge

Appendix

Appendix A. Visualizing a Single Tree



Note: Gini coefficient – evaluates the degree of the heterogeneity of the collection of clusters, which is useful to explain how well the cluster collection reveal the underlying true cluster patterns. It ranges from 0 (completely equality) to 1 (complete inequality).

Appendix B. Feature Importance using Random Forest Classification based on SHAP