TRAINING LOGIT AND RANDOM FOREST MODELS TO PREDICT IT SPENDING

THESIS

Jacob P. Batt, Captain, USAF

AFIT-ENS-MS-22-M-117

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-22-M-117

TRAINING LOGIT AND RANDOM FOREST MODELS TO PREDICT IT SPENDING

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Masters of Science in Operations Research

Jacob P. Batt, B.A.M.

Captain, USAF

March 24, 2022

AFIT-ENS-MS-22-M-117

TRAINING LOGIT AND RANDOM FOREST MODELS TO PREDICT IT SPENDING

THESIS

Jacob P. Batt, B.A.M.
Captain, USAF

Committee Membership:

Raymond Hill, Ph.D
Chair

Maj Phillip Jenkins, Ph.D
Member

AFIT-ENS-MS-22-M-117

Abstract

The ever-present need to modernize is imperative for the Air Force, but the distribution of funds for technology remains tight. To this end, the Air Force Audit Agency is looking to utilize machine learning techniques to enhance their capabilities. This research explores Logistic Regression and Random Forest modeling to streamline data collection and cost classification. The final Logistic Regression model identified 4 significant attributes out of the 36 given and was 85% accurate in predicting whether a purchase amount was over or under $10,000. To expand beyond binary classification, a six-category classification Random Forest model was developed. It identified 6 significant attributes and was 34% accurate in predicting whether a purchase was within 1 of 6 amount categories. Due to the class imbalance of the given data, it was necessary to use a class weighting and over-sampling technique to enhance the Random Forest model. The final class-balanced model identified the same 6 significant attributes but was 78% accurate in predicting whether a purchase was within 1 of 6 amount categories. However, no models were able to predict whether a purchase should be classified as an information technology purchase of not.

Table of Contents

Page

Page

List of Tables

Table Page

TRAINING LOGIT AND RANDOM FOREST MODELS TO PREDICT IT SPENDING

I. Introduction

1.1 Motivation

Managing government spending continues to be an important task for agencies at every level. The Air Force Audit Agency is looking for accurate and efficient ways to predict Information Technology (IT) spending across the Air Force. Governmental budgeting relies on spending projections to build yearly requirements, models that provide spending insight are beneficial to this end. The fast pace in which technology grows requires constant updating and modernization of the Air Force's technological infrastructure. To rise to this challenge, IT spending will increase across the board. With limited government resources, it is imperative to carefully plan and distribute IT equipment as need dictates.

The Audit Agency has determined it needs to explore machine learning techniques to assist in its data collection, purchase predictions and spending classifications. Data collection and storage can be optimized by understanding the important attributes of the data. Removing features that do not add significant detail or help models make predictions lessens the amount of data needed, saving time and money for the data collection process and database maintenance. When beginning an audit, it is helpful to have estimations that drive expectations of the results. Knowing whether a purchase should be IT or not and the approximate amount of that purchase allows the Audit Agency to more quickly identify possible discrepancies while also providing

preemptive snapshots of activities to leaders up and down the chain of command. Both Logistic Regression (LOGIT) and Random Forest (RF) techniques are efficient at identify important variables, classifying observations and have utility in predictive modeling, making them prime candidates for this research.

1.2 Research Objectives

There are three primary goals of this research. First, it is important to verify that the Audit Agency is collecting the right data to answer the questions they are posing. Insufficient data will not yield useful results and too much data could be a wasteful use of resources. This will be addressed by finding the important variables contained within the given data, first with the data pre-processing and then by allowing the models to select the most informational attributes. The second goal is to show whether or not machine learning models are effective for addressing the Audit Agency's needs. If they are not suitable, other options will need to be explored.

Lastly, it is necessary to verify whether LOGIT and RF models specifically are viable options for the Audit Agency. This goal is measured by establishing whether or not the models can predict specific purchase amounts and the characterization of a purchase, whether it is IT or not. If the models cannot predict purchase amounts, this research will explore how precise a spending amount is able to be predicted and if that is useful for the Audit Agency. Being able to accurately predict whether a purchase falls into a specific expense range provides a useful groundwork for budgeting.

1.3 Research Overview

The organization for this research is as follows. Chapter II discusses the background research on LOGIT and RF modeling techniques and the efficacy measures that will be used to evaluate the results of this research. It will also include various

examples of RF modeling beyond the scope of this project discussing similarities and differences between each approach. Chapter III contains the data description and pre-processing before detailing the construction methodology and results of the LOGIT modeling performed in this study. Chapter IV focuses on the RF modeling used in this research. It covers the methodology for binary and multi-category RF models, the enhancement and excursion techniques performed and discusses the results of each. Chapter V discusses the overall results and implications for the research. It details the limitations of this research and lays the groundwork for future research on this topic. Finally, a generic guide for repeating this research is included to help the Agency duplicate the models.

1.4 Tools

This research is conducted in R Studio, which was chosen for its power, simplicity and open source availability. The LOGIT modeling was conducted using the "caret" package, which is a powerful tool for data processing, model training and variable importance detection [1]. For the RF modeling, "H2O" is the primary package that will be used. This package is powerful machine learning tool available for R and Python. It using its own self-contained data type for seamless integration, but is easily transferable to other R data types [2]. The Audit Agency uses SAS as its primary analysis tool and does currently have access to R. While the R code is not directly transferable to SAS syntax, the process laid out by this research should be repeatable using available SAS packages. For the full list of R packages considered and the code used for each model can be found in the Appendices B - D.

II. Background and Literature Review

2.1 Machine Learning Algorithms

While typical modeling or programming involves construction and manual adjusting designs, Machine Learning is a field of study built around an algorithm's ability to improve autonomously. In this way, an algorithm will get better at performing its task through a training process that increasing the efficiency of the model over time. Training a machine learning algorithm requires training and validation data sets, the former helps the algorithm teach itself how to properly classify the data while the later provides a check for how well that data was classified. Training and validation sets can either be separate data sets describing the same system, or partitioned subsets of the original data. This section includes brief overviews of the two machine learning techniques applied in this research and their key elements.

2.2 Logistic Regression Algorithm

Logistic Regression (LOGIT) is a subset of generalized least squares linear regression focused primarily on binary relationships. The LOGIT model uses the Natural LOG of the odds ratio (chance of an event happening or not) as its prediction function [3]. The advantage to using LOGIT for this research is that predictor variables do not need to be linearly related and normally distributed with equal within-group variance [4]. This research is interested in predicting specific spending amounts, which must be more precise than the binary categorization offered by LOGIT. However, LOGIT models provide useful insight for the relationships between variables. This research runs several models over different data partitions to compare all the variables and ascertain which provide worthwhile insight. Two metrics are used to determine the validity of a LOGIT model; model accuracy, which is displayed via confusion matrices

and goodness-of-fit described by McFadden's Psuedo R^2.

2.3 Random Forest Algorithm

The Random Forest (RF) algorithm is an ensemble method based off of a Decision Tree structure. A Decision Trees utilizes a sequential decision process to start at a "root" and evaluate the "branches" until final "leaf" is reached, identifying the final classification target [5]. A root is an observation from the data set and the branches at each level are values of a particular attribute (or predictor variable). The algorithm will iterate through each attribute attempting to select the best fit for the observation by minimizing the impurity of the choice at each level and runs until it is unable to make any improvement [6].

For decision trees, impurity is a ratio between the number of variables belonging to a class and assigned to a node [5]. This research uses the Gini-Impurity index, which looks at the probability an observation belongs to a particular class. Each observation here forms a root and splits are determined by the importance of predictor variables, see below for the formulation calculated by $p_i = |C_i, D|, |D|$, where p_i is the probability an observation in D belongs to class C_i [6].

$$\text{Gini (Node D)} = 1 - \sum_{i=1}^{m} p_i^2 \tag{1}$$

While decision trees are simple to understand, they are prone to data irregularities and usually have poor prediction accuracy. An imbalanced data set or slight changes to given data can have a large impact of the prediction results of these models. The tendency it has to over-fit data or introduce bias contributes to the prediction errors.

A RF ensemble method expands the capability of Decision Trees by generating and aggregating groups of trees into a single model. Different trees are built from randomly selected subsets of predictor variables, and the majority of classifications are

recorded [6]. Iterations of RF models are tuned using hyperparameters based on the resulting important variables and accuracy of previous iterations. RF models suffer from high complexity with longer run times and are susceptible to imbalanced data. They are also difficult to interpret and explain circumstances of resulting accuracy. Despite these disadvantages, RF models are highly accurate and less likely to over-fit. RF modeling was used in this research because it is accurate and the disadvantages were mitigated by the data set's relatively small size and artificially balanced classes.

2.3.1 Hyperparameters

RF models use several hyperparameters to enhance performance. In this research the hyperparameters tuned were: mtry, sample size, replacement, node size, and number of trees (ntrees). This section uses p to represent the number of variables and n as the number of observations. mtry represents the number of randomly drawn variables for each split when growing a tree. Low mtry builds distinct and stable trees, but these trees often perform worse because of the variable variety in their construction. It is best for mtry to be balanced, a typical start point is $\sqrt{p}$. Sample size determines the number of observations per tree. It is closely related to mtry and has a similar impact on the model. Using more samples per tree increases RF accuracy, n is often used because it maximizes the sampling size while minimizing the risk of over-fitting. Replacement couples with sample size to ensure randomness by selecting some observations multiple times and not selecting others [7].

Node size sets the number of observations within a terminating node or leaf. For classification problems, a node size of one is necessary to ensure that a single observation is placed into a single category. Finally, the number of trees is the size of the forest. More trees will smooth out inconsistencies within the model, but increase run time. The number of trees used depends on the size of the data and processing

power, it simply must be large enough to converge. The data for this research is small, so large numbers of trees is not a problem for the model's run time [7].

This research utilizes K-fold cross validation to estimate true error and build more accurate models. Cross validation is the process of dividing a data set into randomly constructed training and validation sets, where the training set builds the model and the validation set checks the model accuracy. K-fold cross validation is repeating the cross validation process k-times (with k different randomly constructed training and validation sets) then taking the average of those models [8]. For efficiency, H2O uses a setting called stopping tolerance that stops modeling if improvement is below a certain threshold.

2.3.2 Splitting Rule

The splitting rule determines how the model chooses the best variable for each new branch. Decision Trees commonly seek to maximize information gain when splitting its branches. For classification problems, and this research, RF models minimize Gini-impurity because it uses classification probabilities specifically. Information gain and Gini-impurity are both measures of model entropy (randomness), which is the measure of distinction between correct and incorrect classifications. Cleaner or more obvious distinctions makes choosing the correct classification easier. Equation 2 represents entropy in this context, with p_i representing the probability distribution of items in class i with m classes [9].

$$\text{Entropy (Node D)} = -\sum_{i=1}^{m} p_i log(p_i) \tag{2}$$

2.4 Efficacy Measures

For this research, model effectiveness is measured using confusion matrices and McFadden's Pseudo R^2. Traditional R^2 goodness-of-fit measurements do not apply to LOGIT models, so pseudo R^2 values were developed to provide a similar measure for the log likelihood. McFadden's Pseudo R^2 is based off a "likelihood ratio index" that compares a model with no predictors to a model with all the predictors and is defined in Equation 3 where LL_{fitted} is the log likelihood value of the fitted model and LL_{null} is the log likelihood value of the empty model [10].

$$R^2 = 1 - LL_{fitted}/LL_{null} \tag{3}$$

Confusion matrices are useful for observing the predictive value of a model. They display how many observations the model chooses correctly against its incorrect choices. The three measures of concern for this research are the accuracy ($TN + TP/TN+FP+FN+TP$), sensitivity ($TP/TP+FN$) and specificity ($TN/TN+FP$), referencing Table 1. For classification models, the diagonal of the confusion matrices indicates the number of correct predictions for each class.

Table 1: Confusion Matrix Example

	Predicted: No	Predicted: Yes
Actual: No	True Negative (TN)	False Positive (FP)
Actual: Yes	False Negative (FN)	True Positive (TP)

While confusion matrices and Psuedo R^2 values were generated for each LOGIT model, the purpose of the full LOGIT models is sufficiently encapsulated with its Psuedo R^2. When determining the important variables to keep, the model's prediction capability explained by the confusion matrix is less important than how well the variable fit the data. The reduced LOGIT models constructed from all the impor-

tant variables generally have a stronger predictive capability and thus provides more insight. The results of the RF models are displayed using confusion matrices as they provide sufficient detail on a model's predictive capability.

2.5 Machine Learning Applications

There is a plethora of literature on LOGIT and RF modeling, but this literature review has not uncovered any sources that use these techniques as they are applied to auditing, like this research. The following is a compilation of RF modeling studies that either shares aspects with this research or highlights other useful applications. Due to the high density of research covering linear and logistic regression, this section focuses on RF modeling approaches with the goal of establishing credibility for its use in various applications.

2.5.1 Imbalanced Data

Many machine learning techniques perform better with balanced data because it provides an easier opportunity for the algorithm to train on all the possible outcomes. Realistically, imbalanced data is commonplace as research is often more interested in deviations from the norm, which occur rarely. Thus, it is necessary to develop techniques that help circumvent imbalance.

Ruiz-Gazen and Villa (2008) used LOGIT and RF modeling to predict storms with an imbalanced data set. They used two methods for overcoming data imbalance; re-balancing with weights then over/under-sampling and creating thresholds that help group final answers. These thresholds function similar to re-sampling techniques by setting an expected probability of an event occurring and inserting that into the model. For example, if the probability of a storm occurring is 5%, a threshold of 0.05 is used to indicate to the model how often to expect an irregular occurrence. The focus

of their research was to compare the accuracy of LOGIT and RF models in storm prediction. They concluded that the accuracy of the RF models was skewed due to overfitting and the interpretability and speed of LOGIT models was likely more useful to meteorologists [11]. The research presented in this thesis, shares common themes with the work of Ruiz-Gazen and Villa [11], but has different goals and methodology. This thesis uses LOGIT and RF models jointly in an iterative process to provide information on the data by narrowing down important variables and attempting to provide precise dollar amount predictions. While there is a comparison between the binary RF model and the LOGIT models in this research, the RF models were the primary focus as they provided the needed precision when predicting spending amounts with smaller and more numerous classes.

Medical data are a common source of imbalanced and much of the available work of balancing data for RF modeling is performed in medical research. Khalilia, Chakraborty and Popescu (2011) published a disease risk prediction study that compared the results of both support vector machine and RF models. The data set was highly imbalanced, so a random sub-sampling method was employed. This method divided the data set into sub-samples consisting of randomly chosen "active" and "inactive" observations such that each sub-sample was 70% inactive and 30% active. They then trained the model on each sub-sample. The resulting RF model outperformed the SVM model in seven of the eight the disease categories. They concluded that the RF model was not only more accurate in modeling, but was overall more useful as it provided additional information on the important variables [12].

Zhu et al. (2018) provide another example of the RF algorithm being applied to imbalanced medical data. In this study, a class weight voting RF algorithm (CWsRF) was employed to combat the data imbalance. The "votes", as used in this paper, are essentially what an observation indicates the classification should be for an input

and traditionally the votes of the majority class have greater bearing on the models outcome. The CWsRF was constructed by building a traditional RF model, using its results to identify majority and minority classes, calculating weights for each class based on score and accuracy of an observation, and then calculate "votes" based on criteria established in the designated voting methodology. In this case, threshold voting was chosen over simple majority. The resulting CWsRF model showed better accuracy over the traditional RF model it was compared against [13].

All of the aforementioned articles used a variation of weighting to combat imbalanced data. Two of them combined weighting with over/under-sampling, which is the approach of this project's excursion models. However, in contrast to these methods, this research provides an opportunity to artificially balance the data. Because the range of each class is unimportant for the Audit Agency, the classes can be constructed such that they similar in size.

2.5.2 Other Random Forest Applications

For a demonstration of RF predicting specific dollar amounts, Antipov and Pokryshevskaya (2012) used a combination of continuous and categorical data to appraise apartments in Russia. They used three different accuracy measures to validate their model: average sales ratio (estimated value over the actual sales price), coefficient of dispersion (percentage deviation of the sales ratio from the median value), and the mean average percentage error (MAPE) between the observed and predicted value. The results of this study produced an appraisal error rate between 9% and 20%, depending on the apartment district [14].

A similar study was done by Hong, Choi, Heeyoul and Kim (2020) were the accuracy of the hedonic pricing model is compared to a RF model for valuing South Korean houses. Both studies used MAPE as the primary appraisal measure and

shared many common attributes. However, this study included only continuous data and thus did not need to reconcile categorical data for the models. This study produced an approximate 5.5% error rate with the RF model and an approximate 20% with the hedonic pricing model [15]. In contrast to the research presented in this paper and the data from the Audit Agency, the data for both the Russia and South Korea housing studies had several measures of price to compare and significantly smaller price intervals. But despite this difference, the work done in these studies provide an indication of the utility of RF models for predicting specific dollar values, which is the goal of the Audit Agency sponsored research.

There is literature available that demonstrates use of RF models in market and purchase predictions. To help retailers understand online customers and develop marketing strategy, Joshi et al. (2018) developed a RF model to categorize customer purchases and predict behavior based on preferences. The data used for this study were collected from a survey that used a mix of categorical and Likert scaled-response questions based on attitude, motive and intent of purchases. A separate RF model was run for each of the eight purchase categories surveyed, with the intent of establishing relationships between the various factors and predicting buying behavior. The results of this study revealed the top three important variables for each model and error rates ranging from approximately 1% to 35% [16].

Baati and Mohsil (2020) similarly attempted to predict online shopper intent by using categorical data that a website gathers on visitors, such as region, day of week and browser. They sought to compare the accuracy of a Naïve Bayes, C4.5 classifier and RF model when attempting to predict an online shopper's intent. Because the original data collected was imbalanced, each model type was run twice. The first set of models used the unchanged imbalanced data and the second set used oversampling to artificially balance the data. With imbalanced data, the Naïve Bayes model performed

the best with 90% accuracy and the RF performed worst with an approximate 84% accuracy. With oversampling, all three models performed about the same with an approximate 87% accuracy [17]. Both of these examples show the utility of RF models in behavior prediction with categorical data, which is a similar purpose and data type used for the modeling in this research.

Another application of RF modeling is found in manufacturing. Instead of a predictive model, Liu et al. (2021) uses RF models for classifying feature importance and correlation for lithium battery manufacturing. This research is important to improving the manufacturing process due to the high complexity and inter-dependencies of the different components. The process used for this research included analysing feature importance with Gini Impurity, analyzing feature correlation with predictive measures of association (PMOA) and reconstructing classifications with the reduced feature set. This data provided useful insight to the manufacturers for the important features and pinpointed focus areas to improve the manufacturing process [18].

III. Data Cleaning and Logistic Regression Modeling Results

3.1 Data Pre-Processing

The data provided by the Audit Agency contained 6000 data entries, each with 36 attributes, that detailed the purchases from a sample of 30 different bases. To reduce the initial attribute count to something more manageable, input from the Audit Agency and dimensionality reduction techniques were applied. The variables removed from this initial reduction are detailed in Table 2.

Table 2: Removed Attributes

Attribute Name	**Status**	**Reason**
Unique ID	Removed	Uninformative
Primary ID	Removed	Uninformative
CRIS Appropriation	Removed	Uninformative
CRIS PEC	Removed	Uninformative
OAC OBAN	Removed	Uninformative
CRIS Expense Only	Removed	Uninformative
Contract No Long	Removed	Uninformative
Document No	Removed	Uninformative
DOV Voucher	Removed	Uninformative
Post Date/CRIS Report Date	Combined (Date Difference)	Uninformative
ITPECNEW(1and 2)	Removed	Blank or redundant
FT Is FSRM	Removed	Uninformative
FT Is ZZEEIC	Removed	Miscellaneous
FT Is ITBPAC	Removed	Imbalanced data
FT Is ITEEIC	Removed	Redundant
FT Is ITNAICS	Removed	Redundant
FT ITPSC	Removed	Uninformative
FT Known	Removed	Uninformative
Sub Cost Pool	Removed	Redundant
IT Tower	Removed	Redundant
Sub IT Tower	Removed	Redundant
Service Type	Removed	Uninformative
Date Diff	Added	Relevant

Removed attributes that were labeled uninformative add nothing to the data. That is, the attribute assigns a unique value to each data point or the information is more significantly encapsulated in another attribute. There were also some attributes removed for having imbalanced data. These attributes were too skewed to a particular value to add significantly to the models. The Date Diff attribute was created to combine the post and report dates for each expense. The rationale behind this addition was to see if longer lengths of time between post and report dates, which would be the processing time, gave an indication of the amount purchased.

In addition to the removed attributes, the Post Date column had 206 blanks. Because of the relatively small number of blanks, the cells were filled with dates representing the first day of the same month and year of the CRIS Report Date. If a CRIS Report Date was labeled 2019-08-20, the assigned Post Date would be 2019-08-01. While this imputation method is imperfect, all the report CRIS Report Date were the last day of the month so this method gave the largest possible time differential between Post Date and CRIS Report Date. Thus, the bias and smoothing downsides of constant value imputation are mitigated by the relatively low volume of blanks. The benefit of having those data points available without blanks made the models in R easier to work with. Finally, the Tier column had a number of NA's; these were changed to 0 values to ease the categorical modeling.

The two attributes used as independent variables to measure the success are IT Expenditure (x_9) and Expense Amount Validated (y). The IT Expenditure was converted into a binary attribute where all affirmative IT Purchases were conditioned "Yes", while all negative or unknown IT purchases were conditioned as "No". The Expense Amount Validated attribute was used to predict the amount spent on a purchase, with values that ranged from \$0 to over \$162 million. Thus, to improve the efficiency of the model without compromising the integrity of its values, Expense

Amount Validated was converted to a nominal data type categorizing the dollar amounts into ranges. It was confirmed with the Audit Agency that there are no particular purchase amounts of interest, so the value ranges were structured with the purpose of evenly binning the data and thus avoiding imbalanced data.

3.2 Logistic Regression Modeling

Introduction

The use of LOGIT models served a dual purpose. First was to identify any insignificant attributes that remained from the data pre-processing and remove them to enhance the parsimony of the model. Additionally, this will help the Audit Agency focus on attributes important in future data collection. The second purpose was to provide the Audit Agency with a potential analytical method for exploring the data and helping them reach insightful conclusions. LOGIT models tend to be easier to build and run, and are a nice alternative to competing machine learning models.

Each LOGIT series was run twice. A baseline model, which identifies the significant attributes over the given range, and a reduced model, which uses only the significant attributes from the baseline model over the same range. This two-step process helps evaluate the best possible model at each range by accounting for differences between ranges and how they may affect significant values. The baseline models 1, 2 and 3 all use the same list of attributes (Table 3) and regression equation (Equation 4). The goodness-of-fit for all baseline models is measured with McFadden's Psuedo R^2.

$$y = \sum_{i=1}^{10} b_i x_i \tag{4}$$

Table 3: Baseline Attributes

Attribute	Variable	Data Type	Number of Categories
Audit Location	x_1	Nominal	28
Cost Pool	x_2	Nominal	8
Subservice Category	x_3	Nominal	22
Date Difference	x_4	Continuous	—
FT is Contract	x_5	Logical	2
FT ITPEC	x_6	Nominal	3
ITPECNEW3	x_7	Nominal	2
FT ITRCCC	x_8	Nominal	4
IT Expenditure	x_9	Nominal	2
Tier	x_{10}	Nominal	3
Expense Amount Validated	y	Logical	Variable

Note: Expense Amount Validated is continuous, but was converted to nominal for each series. The "Number of Categories" column displays the number of possible categories for which an observation could be placed within that specific attribute.

3.2.1 Series 1 Methodology and Results

Both the full and reduced models in this series used the full set of 6000 observations. Expense Amount Validated was converted into a logical value that was split at the $\$5,000$ mark to be used as the dependent variable. Anything $\geq \$5,000$ was labeled true and less than $\$5,000$ were false. The $\$5,000$ pivotal amount was chosen to give an approximate 50/50 split of the data. The reduced models were run with a 70/30 train/test ratio, and model validity was measured using resulting confusion matrices and accuracy.

Any values with a p-value < 0.05 were regarded as significant and used to construct the reduced model. If any of the categories found within a certain attribute were significant, then that entire attribute was retained in the reduced model. The baseline model identified four significant attributes, as listed in Table 4, with a McFadden's Pseudo R^2 value of ≈ 0.5067.

Table 4: Series 1 Significant Attributes

Attribute	**Category**	**Estimate**	**P-value**
Cost Pool	Internal Labor	-2.57	$2e^{-}16$
Cost Pool	NA	-4.71	$2.65e^{-}16$
Subservice Category	Client Computing	-1.56	0.032
Subservice Category	Comm and Collab	-1.72	0.013
FT is Contract	TRUE	0.617	$9.47e^{-}11$
Tier	N1	-0.338	0.036
Expense Amount Validated	—	—	—

Before running the reduced model, Sub-service Categories was removed because it is an imbalanced attribute set and caused model estimation problems. Removing this attribute had no significant impact on the resulting model. Model 1 reduced, produced a validation set accuracy of 85.48% with a sensitivity of 81.52% and a specificity of 91.95%. The training set had an accuracy of 85.08% with a sensitivity of 80.63% and a specificity of 92.47%.

Table 5: Series 1 Reduced Model Training Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	2114	119	5.63%
TRUE	508	1461	34.77%

Table 6: Series 1 Reduced Model Validation Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	909	55	5.71%
TRUE	206	628	24.70%

3.2.2 Series 2 Methodology and Results

Of the 6,000 original observations, 2,455 had $0 expense amounts. With nearly half of the total observations having the same value, there is a chance that the resulting model could have over-inflated accuracy. To verify whether or not this had an effect on model performance, Series 2 models use just the 3,545 non-zero observations. Additionally, the pivotal amount was raised from $\$5,000$ to $\$10,000$. Anything $\geq \$10,000$ was labeled true, all else were false. This maintains the approximate 50/50 split of the data around 'expense amount validated'. This series used the same p-value and significance guidelines for the keeping attributes as Series 1. The baseline model for this series identified four significant values and had a McFadden's Pseudo R^2 of ≈ 0.1499.

Table 7: Series 2 Significant Attributes

Attribute	**Category**	**Estimate**	**P-value**
FT is Contract	TRUE	-1.22	$2e^{-}16$
FT ITRCCC	N1	-0.694	$1.83e^{-}9$
IT Expenditure	N1	0.906	0.028
Tier	N1	-0.952	$4.39e^{-}12$
Expense Amount Validated	—	—	—

The reduced models ran a 70/30 train/test split for training and validation. The third category for Tier was redundant, but every other category had a significant p-value and contributed to the model. Overall, the model had an validation accuracy of 69.87%, a sensitivity of 59.31% and a specificity of 78.15%. The model's training accuracy was 66.53% with a sensitivity of 64.72% and a specificity of 68%.

Table 8: Series 2 Reduced Model Training Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	719	439	61.06%
TRUE	392	933	42.02%

Table 9: Series 2 Reduced Model Validation Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	277	130	32.94%
TRUE	190	465	29.01%

3.2.3 Series 3 Methodology and Results

The original data set contained some high expense values that were removed for the Series 3 models. These large values are near outliers for the Expense Amount Validated variable. This series used all observations that ranged between 0 and $\$500,000$ yielding 3,343 observations. The pivotal amount changed to $\$15,000$ to maintain a 50/50 split, where values $\geq \$15,000$ are labeled true and all others are false. Using the same significance criterion as Series 1 and 2, the baseline model identified five significant values with a McFadden's Psudeo R^2 of ≈ 0.1826.

The Sub-service Categories and Cost Pool attributes were removed from the data set to allow the 70/30 train/test split. Removing these attributes allowed the model to be run without losing much data, as less than half of the possible Sub-service Categories (8/22) and Cost Pool (2/8) categories were identified as significant by the baseline model. The resulting reduced model produced an validation accuracy of 61.1% with a sensitivity of 44.58% and a specificity of 76.53%. The training accuracy was 69.47% with a sensitivity of 51.10% and a specificity of 83.06%.

Table 10: Series 3 Significant Attributes

Attribute	**Category**	**Estimate**	**P-value**
Cost Pool	Internal Labor	-2.64	$2e^{-}16$
Cost Pool	Telecom	-1.08	0.031
Subservice Category	Data	1.79	0.027
Subservice Category	Depot	1.73	0.041
Subservice Category	Development	1.63	0.016
Subservice Category	Network and Connect	1.62	0.033
Subservice Category	Other	2.08	0.006
Subservice Category	Security and Compliance	1.35	0.046
Subservice Category	Storage	1.69	0.033
Subservice Category	Weapon System	2.01	0.003
FT is Contract	TRUE	-0.716	$1.4e^{-}13$
FT ITRCCC	N1	-0.427	$2.29e^{-}4$
Tier	N1	-0.556	$1.14e^{-}3$
Expense Amount Validated	—	—	—

Table 11: Series 3 Reduced Model Training Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	509	228	44.79%
TRUE	487	1118	43.56%

Table 12: Series 3 Reduced Model Validation Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	259	146	36.05%
TRUE	322	476	40.35%

3.2.4 Series 4 Methodology and Results

One of the goals of the Audit Agency was to predict whether or not a purchase was IT. To accomplish this, the Series 4 models tested the IT Expenditure attribute against all others within Table 3. This model was run over each of the ranges used in Series 1, 2 and 3. The baseline model for each of the ranges produced no results. None of the variables were significant based on the criteria used up to this point.

3.3 Discussion

This section examines the results of the LOGIT models by comparing them side-by-side. This helps draw conclusions about that data, assumptions and testing methodology. Table 13 displays the important results for each series of models.

Table 13: LOGIT Model Comparisons

Model	**Accuracy**	**Sensitivity**	**Specificity**
Series 1 Reduced	85.48%	81.52%	91.95%
Series 2 Reduced	69.87%	59.31%	78.15%
Series 3 Reduced	61.1%	44.58%	76.53%
Series 4	No Results	No Results	No Results

Each series produced worse accuracy, sensitivity (how well the model accurately predicted the proper category of a value) and specificity (how exact a category was predicted) than its predecessor. The only difference between the full models for each series were the ranges used, where Series 1 used all 6,000 observations, Series 2 used 3,545 and Series 3 used 3,343 observations. Each full model produced a different McFadden's Pseudo R^2 (Series 1 Full = 0.5067, Series 2 Full = 0.1499 and Series 3 Full = 0.1826) and set of significant variables (see Tables 4, 7, and 10). The McFadden's Pseudo R^2 from the full models follow a similar trend to the reduced model's accuracy, which indicates that the range used for a model is more impactful

than the variables and that these models do not struggle with over-fitting.

Comparing the confusion matrices of the reduced models (Tables 6, 9, and 12), each series accumulates more error (5.71%, 32.94% and 36.05%, respectively) while attempting to predict values less than the pivotal value (FALSE). Thus we know that the models predict \$0 expenses well, but after those values are removed, the models struggle with getting better than 70% overall accuracy. Note that since the \$0 expenses make up over 1/3 of the total values, they could be inflating the model and creating a false accuracy. Removing the \$0 expenses for Series 2 created a more realistic model because it has better class balance. The Series 3 models are not useful as they are the least accurate and provide little to no additional information than the Series 2 models.

Overall, the Series 2 reduced model can predict whether a purchase will be above or below \$10,000 with 70% accuracy. The micro-purchase limit for military card holders is \$10,000, so this model may be valuable if the Audit Agency is trying to predict whether a purchase is considered a micro-purchase or not [19]. The Series 4 models produced no usable results, these data are not suited for the predicting whether a purchase will be IT or not. Each variable in the model produced either a singularity or a p-value of one.

IV. Random Forest Modeling and Results

The Audit Agency data were also modeled using random forest methods. The binary predictions of Logistic Regression (LOGIT) models provide only broad expense predictions which are not sufficiently detailed for the Audit Agency's purpose. The Audit Agency's mission necessitates a model with narrower prediction intervals. To this end, seven random forest models were trained over five different expense ranges. First, two binary models were built to compare the results of the previous LOGIT models. Next, models were built at three, four, five and six expense ranges to test how many are necessary to maintain model accuracy. Certain attributes caused the LOGIT models problems due to lack of presence in the training data. In an attempt to combat this, all of the Random Forest (RF) models use a 75/25 train/test split of the data.

To minimize potential over-fitting and reduce model size, each random forest model is run twice, once with the full set of variables identified in Table 3 and again with only the variables which had more than 5% importance percentage in the full model. H2O calculates variable importance by looking at how often a variable is chosen when building a tree and then calculating the degree to which the squared error increased or decreased [2]. Both the full and reduced models are measured by the classification error rate and confusion matrix of the training set, but the reduced models will additionally test the accuracy of the validation set to verify consistency within the final results. Random Forest models suffer from data imbalance because the decision trees used in the forests are measured by information gain and forests themselves use Gini-splitting criteria which are both susceptible measures to data skew [20]. Since the \$0 purchase amounts makes up over a third of the observations, only values $>$ \$1 are used to build the models.

4.1 Binary Models

4.1.1 Series 1 Full and Reduced Models

The binary RF models were built similarly to the LOGIT models. Model 1 Full takes the variables from Table 3 over all the non-zero expense values using ten folds of cross validation. To keep an approximately even split of the data over the dependent variable, any expense greater than the pivotal value $\$10,000$ was labeled true, all others were false. This model identified six variables that had an importance rating of over 5% (see Table 14). It resulted in a Root Mean Square Error (RMSE) of approximately 0.46, a training set classification error rate of 30.17% and a validation set classification error rate of 25.59%; see Table 15 and Table 16 for the confusion matrices.

Table 14: Full Binary Model 1 Important Variables

Attribute	**Importance Percentage**	**Data Type**
Audit Location	20.51%	Nominal
Date Diff	19.53%	Continuous
Subservice Category	14.92%	Nominal
FT Is Contract	14.31%	Nominal
Tier	11.68%	Nominal
Cost Pool	9.93%	Nominal

Table 15: Full Binary Model 1 Training Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	459	726	61.27%
TRUE	76	1397	5.16%

Model 1 Reduced used the six variables from Table 14 with the same range, pivotal expense amount and train/test split with ten folds of cross validation. This model had

Table 16: Full Binary Model 1 Validation Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	152	244	61.61%
TRUE	33	458	6.72%

similar results to the full model with a RMSE of approximately 0.47 and a training set classification error rate of 31.38%. The resulting confusion matrices for this model are given in Tables 17 and 18. The validation set produced an error rate of 34.05%, which is consistent with these results. The results of this model are not significantly different from the full model.

Table 17: Reduced Binary Model 1 Training Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	447	738	61.28%
TRUE	96	1377	6.52%

Table 18: Reduced Binary Model 1 Validation Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	113	283	71.46%
TRUE	19	472	3.87%

4.1.2 Series 2 Full and Reduced Models

The Series 2 binary full and reduced models only use expense values that fall between \$0 and \$500,000. To account for this new range, and keep an approximately even split of the data, the pivotal expense value was changed to \$15,000. As such, expense values over \$15,000 were labeled true and all else labeled false. The Full Model for this series identified the same six important variables from Table 14, with

only small differences in importance percentage. Utilizing ten folds of cross validation, the resulting RMSE was approximately 0.44 with confusion matrices displayed in Tables 19 and 20, a training classification error rate of 29.99% and a validation classification error rate of 34.21%.

Table 19: Full Binary Model 2 Training Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	656	667	50.42%
TRUE	85	1099	9.18%

Table 20: Full Binary Model 2 Validation Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	213	251	54.09%
TRUE	35	337	9.41%

The reduced model for this series utilized the six most important variables while maintaining the same range, pivotal expense value and ten folds of cross validation. This model resulted in a RMSE of approximately 0.47, a training set classification error rate of 31.38% and a validation set error rate of 34.45%. As seen in the confusion matrices 21 and 22, the results of this model are not significantly different from the other binary models.

Table 21: Reduced Binary Model 2 Training Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	447	738	62.28%
TRUE	96	1377	6.52%

Table 22: Reduced Binary Model 2 Validation Confusion Matrix

	FALSE	TRUE	ERROR
FALSE	203	261	56.25%
TRUE	27	345	7.26%

4.2 Multi-Categorical Models

Initial models treated the expense values as continuous data. However, the range of the data was so vast that the resulting models produced unusable results. Thus, multi-categorical models were used to test the capability to categorize and predict expense values for smaller intervals. The models for this section were constructed in three series using three, four and five expense categories. All models were constructed following the pattern of the binary models, where full models were constructed to identify any variables that had an importance score over 5%.

Each of the models was constructed over all non-zero expenses. Each series used the same train/test split percentage and ten folds of cross validation. The full model for each series yielded the same six important variables as all the other models (see Table 14). The expense categories for each model along with the resulting RMSE and error rate are displayed in Table 23. The results of the models show a clear trend of growing RMSE and error rate with each additional category. For brevity, only the validation and training confusion matrices for the five category model are displayed. Due to the downward trend, adding more categories is unnecessary.

Table 23: Model Expense Categories

Category	**Reference**	**Ranges**	**RMSE**	**Error**	**Validation**
3	ONE TWO THREE	$\$5K \geq x > \$0$ $\$50K \geq x > \$5K$ $x > \$100K$	0.5752	42.89%	Null
4	ONE TWO THREE FOUR	$\$3K \geq x > \$0$ $\$15K \geq x > \$3K$ $\$100K \geq x > \$15K$ $x > \$100K$	0.6536	52.74%	Null
5	ONE TWO THREE FOUR FIVE	$\$1K \geq x > \$0$ $\$5K \geq x > \$1K$ $\$20K \geq x > \$5K$ $\$100K \geq x > \$20K$ $x > \$100K$	0.7421	65.61%	Null

Table 24: Reduced 5 Category Training Confusion Matrix

	ONE	TWO	THREE	FOUR	FIVE	ERROR
ONE	140	101	38	56	53	63.92%
TWO	88	205	62	118	99	64.16%
THREE	33	74	96	125	115	78.33%
FOUR	32	72	88	254	193	61.63%
FIVE	36	95	70	196	219	63.07%

Table 25: Reduced 5 Category Validation Confusion Matrix

	ONE	TWO	THREE	FOUR	FIVE	ERROR
ONE	57	33	12	27	22	62.25%
TWO	29	81	22	27	26	56.22%
THREE	16	29	22	28	36	84.40%
FOUR	20	29	22	67	67	67.32%
FIVE	14	19	22	62	88	57.07%

4.3 Model Enhancements

4.3.1 Gridsearch

A random forest algorithm uses a set of hyperparameters to fine-tune its performance. H2O's default hyper-parameters were sufficient for all the previous models built, adding ten folds of cross validation was the only change. Given the poor model results thus far, it is unlikely that adjustments in hyper-parameters will render more useful models, however, the tuning process is examined regardless. Gridsearch methodology is a function that constructs a Cartesian plane of test values for a given variable and trains a separate model on each value. The Caret package accomplishes this with its "expand.grid" and "train" functions [1].

The full list of inputs H2O uses to build its random forest models are found within the online manual [2]. The hyper-parameters used focus on are mtry, sample size, node size, number of trees and splitting rule. For classification random forest models, defaults for node size (1), sample size (n), and the splitting rule (Gini impurity) are sufficient. The number of trees for a model determines the size of the sampling forest, when large enough, this parameter converges and does not add anything new to the model. Since the data set here is relatively small, and run time is not an issue, 1000 trees is defined. Lastly, mtry determines the number of variables in each split and is the focus of this gridsearch operation. A typical value for mtry in a classification model is $\sqrt{p}$ (where p is the number of predictor values) [7]. The gridsearch run uses a range of values from 2-12 to ensure a thorough exploration of the parameter. The gridsearch models were run using the full range of variables identified in Table 3, ten folds of cross validation and repeated three times. The results are displayed in Table 26, which reveals that a mtry value of 11 produced the model with the highest accuracy. The difference between the accuracy of the gridsearch and default models is marginal.

Table 26: Gridsearch Results for 5 Category Model

mtry	Accuracy	Kappa
2	36.39%	0.1803812
3	38.45%	0.2093152
4	39.5%	0.2241431
5	40.11%	0.2330828
6	40.66%	0.2411802
7	40.81%	0.2442773
8	41.13%	0.2491956
9	41.16%	0.2504337
10	41.39%	0.2539497
11	41.4%	0.2545487
12	41.2%	0.2524946

4.3.2 Imbalanced Data Correction Methods

Operating under the assumption that the random forest algorithm is susceptible to imbalanced data, the models were constructed to exclude all $0 expenses, which makes up over a third of the data set. The remaining values were artificially categorized into near even categories so the models could be run without the need for imbalance fixes. However, removing such a large quantity of observations can have an adverse impact on how well the model can be trained. This section explores balancing data with H2O.

When working with classification random forest models, the effects of imbalanced data can be mitigated by changing the sampling method [21]. Sampling adaptations of this nature are achieved by weighting either individual values or entire classes such that the algorithm will balance its selections from each class. The "balance classes" option weights appropriately and over-samples the minority classes to create an artificially even distribution of each class within the model [2]. This option inflates the overall sample space producing more training opportunity for the model.

Two models test the utility of this option; the five category model from Section 4.2

and a six category model that includes all the $0 expenses that were removed as the sixth category. The full range of variables identified in Table 3 were used for this section in order to provide the most complete picture of the data. Apart from the aforementioned changes, all other factors are identical to the previous random forest models. The results and confusion matrices for these models are displayed in Tables 27, 28, 29, 30, and 31. The accuracy for the five category model increased by over 10% and the six category model nearly doubled the accuracy of any of the previous random forest models.

Table 27: Enhanced Categorical Models

Categories	**Reference**	**Ranges**	**RMSE**	**Error**	**Validation**
5	ONE	$\$1K \geq x > \$0$	0.7421	65.61%	Null
	TWO	$\$5K \geq x > \$1K$			
	THREE	$\$20K \geq x > \$5K$			
	FOUR	$\$100K \geq x > \$20K$			
	FIVE	$x > \$100K$			
6	ONE	$x = \$0$	0.4611	21.63%	Null
	TWO	$\$1K \geq x > \$0$			
	THREE	$\$5K \geq x > \$1K$			
	FOUR	$\$20K \geq x > \$5K$			
	FIVE	$\$100K \geq x > \$20K$			
	SIX	$x > \$100K$			

Table 28: Enhanced 5 Category Training Confusion Matrix

	ONE	TWO	THREE	FOUR	FIVE	ERROR
ONE	381	90	38	65	69	40.75%
TWO	88	302	54	108	85	52.59%
THREE	31	64	326	120	101	49.22%
FOUR	33	82	90	244	187	61.64%
FIVE	24	53	56	185	304	51.13%

Table 29: Enhanced 5 Category Validation Confusion Matrix

	ONE	TWO	THREE	FOUR	FIVE	ERROR
ONE	174	26	15	14	12	27.80%
TWO	14	144	17	43	22	40.00%
THREE	9	15	122	44	35	45.78%
FOUR	8	32	31	95	69	59.57%
FIVE	4	17	21	67	125	46.58%

Table 30: Enhanced 6 Category Training Confusion Matrix

	ONE	TWO	THREE	FOUR	FIVE	SIX	ERROR
ONE	1844	0	0	1	0	0	0.11%
TWO	7	1295	281	29	133	97	29.7%
THREE	3	147	1339	80	174	106	27.58%
FOUR	10	52	102	1190	296	202	35.75%
FIVE	4	28	55	85	1461	221	21.2%
SIX	1	30	11	48	195	1564	15.41%

Table 31: Enhanced 6 Category Validation Confusion Matrix

	ONE	TWO	THREE	FOUR	FIVE	SIX	ERROR
ONE	608	0	1	1	1	1	0.65%
TWO	0	536	34	20	13	7	12.13%
THREE	0	20	524	29	25	16	14.66%
FOUR	0	18	22	497	51	28	19.32%
FIVE	0	6	18	29	524	39	14.94%
SIX	0	7	16	22	41	526	14.05%

4.4 Discussion

Overall the random forest algorithm used for this paper shows potential utility for the Audit Agency and provides some useful insights about the data. The binary LOGIT and RF models demonstrated similar prediction accuracy, but a binary categorization of expenses is too broad to be helpful to the Audit Agency. The subsequent three, four, and five category models further break down the expenses into smaller categories, but with each additional category the accuracy drops significantly (see Table 32). The three, four and five category models are not effective as none of them have 60% or more accuracy, making them little better than a random guess. The five category gridsearch model did theoretically increase accuracy, however the mtry (number of features chosen per split) recommended by the model is not feasible because the data cleaning removed all but ten dependent features.

Table 32: Random Forest Model Comparisons

Model	**Error**	**Validation**
2 Category (1)	31.38%	70.96%
2 Category (2)	31.38%	76.56%
3 Category	42.89%	56.98%
4 Category	52.74%	40.97%
5 Category	65.61%	24.83%
5 Category Gridsearch	42.4%	—
5 Category Class Balanced	65.61%	33.93%
6 Category Class Balanced	21.63%	68.23%

Note: The class balancing algorithm in H2O does not include the validation set. Without balancing the validation set accuracy was 36.47%. The validation set was then used to train the model, generating 100% accuracy. The validation percentage presented for the 6 Category Class Balance model is an average of these two values.

The systematic decline in accuracy between models highlights the limitations of the given data to predict specific expense amounts. The data set is small and does not have many duplicate values. Outside of the $0 expense values, there are only 181

observations with the same expense values. This means that machine learning models do not have many reference points to compare and categorize similarly. This work attempted to combat this by utilizing classification categories instead of continuous amounts to give the model larger pools of observations to sort and compare. However, as indicated previously, this method has minimal impact.

The class balanced models presented a second approach for getting around the data problem using weights and oversampling. Categories were weighted in proportion to their size and then over-sampled until each category had the same number of observations. This method created a data set with 11,070 points (six categories with 1,845 observations each) with nearly 80% training and 68% validation accuracy. Because H2O only weighted the categories and used random sampling with each category, this indicates that larger data sets will likely produce more accurate models and supports the theory that the variables recorded in the data are sufficient. Also potentially indicating that the small size of the data caused the previous model inadequacies. Overall, with the given data, the 6 Category Class Balance model could be adequate for the purposes of the Audit Agency.

V. Conclusion

5.1 Research Results and Implications

This work shows promising results for the Audit Agency. Both of the Logistic Regression (LOGIT) and Random Forest (RF) produced usable models with $\approx 70-80\%$ prediction accuracy and produced very similar lists of important variables. Models this accurate make good contextual reference frameworks for budgeting and verification of the Audit Agency's purchase data. However these models are categorical and limited by the parameters and data used to build them, they are not accurate enough to provide answers detailing specific expense amounts or purchase types.

Many attributes from the original data-set were identified as unnecessary and removed (see Table 2). To save time and effort, it would behoove the Audit Agency to focus data collection efforts on the attributes identified in Table 3 as they brought the most utility to the modeling process. To maximize efficiency, only collect data on the identified attributes for the model of choice, LOGIT - (Table 7) and RF - (Table 14).

5.2 Limitations

This work is limited to the scope of the needs of the Audit Agency and the data provided. The techniques used within this paper are exploratory and not exhaustive of all possible machine learning or regression analytical tools. As such, these models are subject to the disadvantages of their respective techniques as discussed in Chapter II. The RF models used classification criteria in construction, and as such, limits the effects of certain hyperparameters (node size, sample size, and splitting rule) being tuned and the information that could be gained by them from the models.

IT procurement is a slow process and one that is constantly in motion because

of the speed technology advances. The data provided only covers expenses made in 2018 and 2019. This leaves gaps where a few more years of data could enhance the results or at least provide more observations for the models to utilize. The small data set created a problem for the RF models with not enough observations in some of the variables. This was remedied with class balancing via oversampling to build a larger artificial data set.

Lastly, this work utilized commercial random forest packages within R, H2O and caret. As such, the algorithm was not tailored to the data set. Commercial software is useful, but can be hard to troubleshoot or observe specific parts of the model if it becomes necessary. For example, H2O uses replacement when oversampling and thus does not maintain the expanded data frame at the end of the algorithm. This made it challenging to observe how the sampling was done and identify possible sources of data errors.

5.3 Future Research

There are several directions one could expand upon this work. Machine learning techniques outside of LOGIT and RF could be applied to this problem. Support Vector Machines (SVMs) is one possible classification algorithm that could be applied, however, it grows in complexity with more categories to parse. It is unlikely to out-perform random forests. Neural networks could be employed for classifying or predicting specific dollar amounts from the data. These networks are often more complex and take longer to run, but can be trained and re-trained to obtain better results each iteration. This is a promising avenue if the Audit Agency wishes to expand the number of expense categories. Balanced data are important for all of these algorithms, as such, either more actual data is needed or artificial data will have to be generated based on inputs from subject matter experts.

In addition to different algorithms, the problem can also be approached in different ways. For instance, the pivotal expense values used to create the categories were chosen to create artificially balanced groups, but these expense values have no other significance. Exploring different pivotal expenses could provide more detail on the data and create tighter classification intervals. Another possible approach is to move away from classifications and towards regression. By rounding each expense value to the nearest 100th or 1000th dollar amount, the similarities between expense values could become more obvious and make it easier for an algorithm to train. It should be noted that however that there are many expense between $0 - $100 and $0 - $1000, and rounding could remove some important data points, making the answer less precise.

5.4 Summary

Overall, this work was able to provide useful answers to the research questions posed. The Audit Agency is collecting a lot of data for their questions; the variables in Table 2 could be removed from the data collection without much statistical consequence to the end results. Machine learning models can efficiently model the data and provide answers, but their accuracy is vulnerable to the small and imbalanced data set. Outside of providing insight on significant variables, LOGIT is not useful for the Audit Agency. While maintaining fair accuracy, binary classification is not informative enough to provide other useful insights. By contrast, RF models allow for more classes and thus more precise expense predictions. While still limited, the ability to place a cost prediction into increasingly smaller intervals is a useful tool for the Audit Agency. Neither the LOGIT or RF were able to predict with any amount of accuracy whether a particular purchase was IT related.

Table 33: Specifications of the Final Balanced Class Random Forest Model

Setting/Parameter	**Value**
Seed	29
Training set	75%
Validation set	25%
Cross-validation folds	10
Node size	1
Sample size	n
Splitting rule	Gini impurity
ntrees	100
mtry	Use all features
Stopping tolerance	0.001

While this work was done in R, the Audit Agency must be able to reproduce this in SAS. While SAS does have a random forest algorithm, HPFOREST, with a balanced classes option, it does not over-sample the same way that H2O does. To remedy this, it is necessary to artificially duplicate observations to create a larger data frame with similar characteristics to the one created by H2O's class balancing option. When balancing, H2O sampled enough to bring all the classes to the same size as the largest class sample, use the following steps to build an artificial balanced class data frame in Microsoft Excel.

1. Assign each expense value observation into a class.

2. Calculate the size difference between the largest class and each smaller class.

3. For each observation within a smaller class, assign and random number such that all the random numbers for that class add up to its difference calculated in Step 2.

4. Duplicate each observation equal the value of its random number assigned in Step 3.

Appendix

Appendix A: Important Data Name Descriptors

Below are the names and meanings for some important indicator random variables or attributes.

- FT is ZZEEIC - Expense code
- FT ITBACK - Budget program activity code - the code for the major program has something that suggests IT (cyber, IT, etc.).
- FT ITEEIC - EEIC (Element of Expense Identification Code) is the expense code (investment codes tells what the investment is for) - EEIC suggests IT expense.
- FT ITNAICS - NAICS (North American Industry Classification System) code is contract specific - shows if "FT is contract" is true. NAICS code contains something that suggests IT.
- FT ITPEC - PEC - Program element code name of program suggests IT spending.
- FT ITPSC - PSC is the product service code - suggests IT spending.
- FT ITRCCC - RCCC - Responsibility cost center code - transaction name at unit level - suggests IT spending.

Appendix B: R Code for LOGIT Models

```
#Packages
library(readxl)
library(car)
library(caret)
library(MASS)
library(FactoMineR)
library(caTools)
library(ggplot2)
library(aod)
library(pscl)

## Overall test of all attributes
dta <- read_excel("c://R/RData_Thesis1.xlsx",3)    #all amounts
dta_1 <- cbind.data.frame(dta[3:13])
sapply(dta_1, class)
dta_1 <- transform(dta_1,
                   Audit.Location=as.factor(Audit.Location),
                   Cost.Pool=as.factor(Cost.Pool),
                   SubService.Category=as.factor(SubService.Category),
                   Date.Diff.Cont=as.integer(Date.Diff.Cont),
                   FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                   FT.ITPEC.N=as.factor(FT.ITPEC.N),
                   ITPECNEW3.N=as.factor(ITPECNEW3.N),
                   FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                   IT.Expenditure.N=as.factor(IT.Expenditure.N),
                   Tier.N=as.factor(Tier.N),
                   Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

## Reduced Data with no 0 entries
no_0 <- read_excel("c://R/RData_Thesis1.xlsx",4)       #no zeros left
no_0_1 <- cbind.data.frame(no_0[3:13])
sapply(no_0_1, class)
no_0_1 <- transform(no_0_1,
                    Audit.Location=as.factor(Audit.Location),
                    Cost.Pool=as.factor(Cost.Pool),
                    SubService.Category=as.factor(SubService.Category),
                    Date.Diff.Cont=as.integer(Date.Diff.Cont),
                    FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                    FT.ITPEC.N=as.factor(FT.ITPEC.N),
                    ITPECNEW3.N=as.factor(ITPECNEW3.N),
                    FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                    IT.Expenditure.N=as.factor(IT.Expenditure.N),
                    Tier.N=as.factor(Tier.N),
                    Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

## Reduced Data with no 0 entries or entries >500000 around $15K
no_0or500 <- read_excel("c://R/RData_Thesis1.xlsx",5) #no zeros or above 500K
no_0or500_1 <- cbind.data.frame(no_0or500[3:13])
sapply(no_0or500_1, class)
no_0or500_1 <- transform(no_0or500_1,
                         Audit.Location=as.factor(Audit.Location),
                         Cost.Pool=as.factor(Cost.Pool),
```

```
                        SubService.Category=as.factor(SubService.Category),
                        Date.Diff.Cont=as.integer(Date.Diff.Cont),
                        FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                        FT.ITPEC.N=as.factor(FT.ITPEC.N),
                        ITPECNEW3.N=as.factor(ITPECNEW3.N),
                        FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                        IT.Expenditure.N=as.factor(IT.Expenditure.N),
                        Tier.N=as.factor(Tier.N),
                        Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

################################################################################

##Series 1 Train/test splits
trn_m1 <- createDataPartition(dta_1$Tier.N, p=0.7, list=FALSE)
tr1 <- dta_1[ trn_m1, ]
tst1 <- dta_1[ -trn_m1, ]

#Overall Model 1 baseline
m1_base <- glm(Expense.Amount.Validated.N ~ Audit.Location + Cost.Pool +
               SubService.Category + Date.Diff.Cont + FT.Is.Contract.N +
               FT.ITPEC.N + ITPECNEW3.N + FT.ITRCCC.N + IT.Expenditure.N + Tier.N,
               data=dta_1, family="binomial"); summary(m1_base)
#McFadden's R2
m1b_MF <- pR2(m1_base);m1b_MF

#Overall Model 1 reduced
m1_red <- train(Expense.Amount.Validated.N ~ Cost.Pool + FT.Is.Contract.N + Tier.N,
                data=tr1, method="glm", family = "binomial");summary(m1_red)
m1r_pr <- predict(m1_red, newdata=tst1)
#Model 1 reduced confusion matrix
confusionMatrix(m1r_pr,tst1$Expense.Amount.Validated.N)

##Series 2 Train/test splits
trn_m2 <- createDataPartition(no_0_1$Tier.N, p=0.7, list=FALSE)
tr2 <- no_0_1[ trn_m2, ]
tst2 <- no_0_1[ -trn_m2, ]
#Overall Model 2 baseline
m2_base <- glm(Expense.Amount.Validated.N ~ Audit.Location + Cost.Pool +
               SubService.Category + Date.Diff.Cont + FT.Is.Contract.N +
               FT.ITPEC.N + ITPECNEW3.N + FT.ITRCCC.N + IT.Expenditure.N + Tier.N,
               data=no_0_1, family="binomial"); summary(m2_base)
#McFadden's R2
m2b_MF <- pR2(m2_base);m2b_MF
#Overall Model 2 reduced
m2_red <- train(Expense.Amount.Validated.N ~ FT.Is.Contract.N + FT.ITRCCC.N +
                IT.Expenditure.N + Tier.N,
                data=tr2, method="glm", family="binomial");summary(m2_red)
m2r_pr <- predict(m2_red, newdata=tst2)
#Model 2 reduced confusion matrix
confusionMatrix(m2r_pr, tst2$Expense.Amount.Validated.N)

##Series 3 Train/test splits
trn_m3 <- createDataPartition(no_0or500_1$Tier.N, p=0.7, list=FALSE)
tr3 <- no_0_1[ trn_m3, ]
```

```
tst3 <- no_0_1[ -trn_m3, ]
#Overall Model 3 baseline
m3_base <- glm(Expense.Amount.Validated.N ~ Audit.Location + Cost.Pool +
                SubService.Category + Date.Diff.Cont + FT.Is.Contract.N +
                FT.ITPEC.N + ITPECNEW3.N + FT.ITRCCC.N + IT.Expenditure.N + Tier.N,
                data=no_0or500_1, family="binomial");summary(m3_base)
#McFadden's R2
m3b_MF <- pR2(m3_base);m3b_MF
#Overall Model 3 reduced
m3_red <- train(Expense.Amount.Validated.N ~ FT.Is.Contract.N + FT.ITRCCC.N + Tier.N,
                 data=tr3, method="glm", family="binomial");summary(m3_red)
m3r_pr <- predict(m3_red, newdata=tst3)
#Model 3 reduced confusion matrix
confusionMatrix(m3r_pr, tst3$Expense.Amount.Validated.N)

#Series 4 - Overall Model 4 Baseline
m4_base <- glm(IT.Expenditure.N ~ Audit.Location + Cost.Pool +
               SubService.Category + Date.Diff.Cont + FT.Is.Contract.N +
               Expense.Amount.Validated.N + FT.ITPEC.N + ITPECNEW3.N +
               FT.ITRCCC.N + IT.Expenditure.N + Tier.N,
               data=dta_1, family = "binomial"); summary(m4_base)
#McFadden's R2
m4b_MF <- pR2(m4_base);m4b_MF
```

Appendix C: R Code for Random Forest Models

```
# Packages
library(readxl)
library(car)
library(MASS)
library(FactoMineR)
library(randomForest)
library(caTools)
library(h2o)
library(jsonlite)
library(dplyr)
library(ggplot2)
library(tidyr)
library(caret)


                                        ### Binary Series ###
## Binary around $10000 and no zeros
bin1 <- read_excel("c://R/RData_Thesis1.xlsx",4)
binary1 <- cbind.data.frame(bin1[3:13])
sapply(binary1, class)
binary1 <- transform(binary1,
                     Audit.Location=as.factor(Audit.Location),
                     Cost.Pool=as.factor(Cost.Pool),
                     SubService.Category=as.factor(SubService.Category),
                     Date.Diff.Cont=as.integer(Date.Diff.Cont),
                     FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                     FT.ITPEC.N=as.factor(FT.ITPEC.N),
                     ITPECNEW3.N=as.factor(ITPECNEW3.N),
                     FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                     IT.Expenditure.N=as.factor(IT.Expenditure.N),
                     Tier.N=as.factor(Tier.N),
                     Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

sample1 = sample.split(binary1$Tier.N, SplitRatio = 0.75)
train1 = subset(binary1, sample1 == TRUE)
test1 = subset(binary1, sample1 == FALSE)
h2o.init(nthreads = 20, max_mem_size = "6g")

trn1_f <- as.h2o(train1)
tst1_f <- as.h2o(test1)
rf1_f <- h2o.randomForest(y="Expense.Amount.Validated.N",
                          training_frame=trn1_f, stopping_rounds = 5, stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
                              10)

trn1_r <- as.h2o(select(as.data.frame(trn1_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
tst1_r <- as.h2o(select(as.data.frame(tst1_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
rf1_r <- h2o.randomForest(x=1:6, y="Expense.Amount.Validated.N",
                          training_frame=trn1_r, validation_frame=tst1_r, stopping_rounds = 5,
                              stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
```

```
                                10)

perf1_f <- h2o.performance(rf1_f);perf1_f
h2o.varimp(rf1_f)
print(h2o.auc(rf1_f, valid = TRUE))
print(h2o.auc(rf1_r, valid = TRUE))


## Binary around $15000 and no zeros or >$500K
bin2 <- read_excel("c://R/RData_Thesis1.xlsx",5)
binary2 <- cbind.data.frame(bin2[3:13])
sapply(binary2, class)
binary2 <- transform(binary2,
                     Audit.Location=as.factor(Audit.Location),
                     Cost.Pool=as.factor(Cost.Pool),
                     SubService.Category=as.factor(SubService.Category),
                     Date.Diff.Cont=as.integer(Date.Diff.Cont),
                     FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                     FT.ITPEC.N=as.factor(FT.ITPEC.N),
                     ITPECNEW3.N=as.factor(ITPECNEW3.N),
                     FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                     IT.Expenditure.N=as.factor(IT.Expenditure.N),
                     Tier.N=as.factor(Tier.N),
                     Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

sample2 = sample.split(binary2$Tier.N, SplitRatio = 0.75)
train2 = subset(binary2, sample2 == TRUE)
test2 = subset(binary2, sample2 == FALSE)
h2o.init(nthreads = 20, max_mem_size = "6g")

trn2_f <- as.h2o(train2)
tst2_f <- as.h2o(test2)
rf2_f <- h2o.randomForest(y="Expense.Amount.Validated.N",
                          training_frame=trn2_f, stopping_rounds = 5, stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds = 10)

trn2_r <- as.h2o(select(as.data.frame(trn2_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
tst2_r <- as.h2o(select(as.data.frame(tst2_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
rf2_r <- h2o.randomForest(x=1:6, y="Expense.Amount.Validated.N",
                            training_frame=trn2_r, validation_frame=tst2_r, stopping_rounds = 5,
                                stopping_tolerance = 0.001,
                            stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
                                10)

perf2_f <- h2o.performance(rf2_f);perf2_f
h2o.varimp(rf2_f)
perf2_r <- h2o.performance(rf2_r);perf2_r
h2o.varimp(rf2_r)
print(h2o.auc(rf2_r, valid = TRUE))
```

```
                                    ### No zeros ###
## 5 cost groups
cat6 <- read_excel("c://R/RData_Thesis1.xlsx",8)
cat6_1 <- cbind.data.frame(cat6[3:13])
sapply(cat6_1, class)
cat6_1 <- transform(cat6_1,
                    Audit.Location=as.factor(Audit.Location),
                    Cost.Pool=as.factor(Cost.Pool),
                    SubService.Category=as.factor(SubService.Category),
                    Date.Diff.Cont=as.integer(Date.Diff.Cont),
                    FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                    FT.ITPEC.N=as.factor(FT.ITPEC.N),
                    ITPECNEW3.N=as.factor(ITPECNEW3.N),
                    FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                    IT.Expenditure.N=as.factor(IT.Expenditure.N),
                    Tier.N=as.factor(Tier.N),
                    Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

sample6 = sample.split(cat6_1$Tier.N, SplitRatio = 0.75)
train6 = subset(cat6_1, sample6 == TRUE)
test6 = subset(cat6_1, sample6 == FALSE)
h2o.init(nthreads = 20, max_mem_size = "6g")

trn6_f <- as.h2o(train6)
tst6_f <- as.h2o(test6)
rf6_f <- h2o.randomForest(y="Expense.Amount.Validated.N",
                          training_frame=trn6_f, stopping_rounds = 5, stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds = 10)

trn6_r <- as.h2o(select(as.data.frame(trn6_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
tst6_r <- as.h2o(select(as.data.frame(tst6_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
rf6_r <- h2o.randomForest(x=1:6, y="Expense.Amount.Validated.N",
                          training_frame=trn6_r, validation_frame=tst6_r, stopping_rounds = 5,
                              stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
                              10)

perf6_f <- h2o.performance(rf6_f);perf6_f
h2o.varimp(rf6_f)
perf6_r <- h2o.performance(rf6_r);perf6_r
h2o.varimp(rf6_r)

############################ Build up cost groups ################################
## 3 cost groups
cat3 <- read_excel("c://R/RData_Thesis1.xlsx",9)
cat3_1 <- cbind.data.frame(cat3[3:13])
sapply(cat3_1, class)
cat3_1 <- transform(cat3_1,
                    Audit.Location=as.factor(Audit.Location),
                    Cost.Pool=as.factor(Cost.Pool),
                    SubService.Category=as.factor(SubService.Category),
                    Date.Diff.Cont=as.integer(Date.Diff.Cont),
```

```
                    FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                    FT.ITPEC.N=as.factor(FT.ITPEC.N),
                    ITPECNEW3.N=as.factor(ITPECNEW3.N),
                    FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                    IT.Expenditure.N=as.factor(IT.Expenditure.N),
                    Tier.N=as.factor(Tier.N),
                    Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

sample3 = sample.split(cat3_1$Tier.N, SplitRatio = 0.75)
train3 = subset(cat3_1, sample3 == TRUE)
test3 = subset(cat3_1, sample3 == FALSE)
h2o.init(nthreads = 20, max_mem_size = "6g")

trn3_f <- as.h2o(train3)
tst3_f <- as.h2o(test3)
rf3_f <- h2o.randomForest(y="Expense.Amount.Validated.N",
                          training_frame=trn3_f, stopping_rounds = 5, stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
                              10)

trn3_r <- as.h2o(select(as.data.frame(trn3_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
tst3_r <- as.h2o(select(as.data.frame(tst3_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
rf3_r <- h2o.randomForest(x=1:6, y="Expense.Amount.Validated.N",
                          training_frame=trn3_r, validation_frame=tst3_r, stopping_rounds = 5,
                              stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
                              10)

perf3_f <- h2o.performance(rf3_f);perf3_f
h2o.varimp(rf3_f)
perf3_r <- h2o.performance(rf3_r);perf3_r
h2o.varimp(rf3_r)


## 4 cost groups
cat4 <- read_excel("c://R/RData_Thesis1.xlsx",10)
cat4_1 <- cbind.data.frame(cat4[3:13])
sapply(cat4_1, class)
cat4_1 <- transform(cat4_1,
                    Audit.Location=as.factor(Audit.Location),
                    Cost.Pool=as.factor(Cost.Pool),
                    SubService.Category=as.factor(SubService.Category),
                    Date.Diff.Cont=as.integer(Date.Diff.Cont),
                    FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                    FT.ITPEC.N=as.factor(FT.ITPEC.N),
                    ITPECNEW3.N=as.factor(ITPECNEW3.N),
                    FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                    IT.Expenditure.N=as.factor(IT.Expenditure.N),
                    Tier.N=as.factor(Tier.N),
                    Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

sample4 = sample.split(cat4_1$Tier.N, SplitRatio = 0.75)
```

```
train4 = subset(cat4_1, sample4 == TRUE)
test4 = subset(cat4_1, sample4 == FALSE)
h2o.init(nthreads = 20, max_mem_size = "6g")

trn4_f <- as.h2o(train4)
tst4_f <- as.h2o(test4)
rf4_f <- h2o.randomForest(y="Expense.Amount.Validated.N",
                          training_frame=trn4_f, stopping_rounds = 5, stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
                              10)

trn4_r <- as.h2o(select(as.data.frame(trn4_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
tst4_r <- as.h2o(select(as.data.frame(tst4_f), Audit.Location, Date.Diff.Cont, SubService.Category
    , FT.Is.Contract.N, Tier.N, Cost.Pool, Expense.Amount.Validated.N))
rf4_r <- h2o.randomForest(x=1:6, y="Expense.Amount.Validated.N",
                          training_frame=trn4_r, validation_frame=tst4_r, stopping_rounds = 5,
                              stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = FALSE, nfolds =
                              10)

perf4_f <- h2o.performance(rf4_f);perf4_f
h2o.varimp(rf4_f)
perf4_r <- h2o.performance(rf4_r);perf4_r
h2o.varimp(rf4_r)

##############################################################################
## Examples for predicitions and Confusion Matrices
#Confusion Matrices
h2o.confusionMatrix(rf3)
h2o.confusionMatrix(rf4)

#Predictors, use as needed
pred4 <- h2o.predict(rf4_r, newdata=as.h2o(tst4_r))
h2o.exportFile(pred4, path = 'c://Users/Ajax/Desktop/rfexcurvalid.csv', force = TRUE)
h2o.exportFile(predbc2, path = 'c://Users/Ajax/Desktop/val.csv', force = TRUE)
```

Appendix D: R Code for Random Forest Model Enhancement and Excursion

```
## Gridsearch Packages and Code ##
library(readxl)
library(car)
library(MASS)
library(FactoMineR)
library(randomForest)
library(caTools)
library(h2o)
library(jsonlite)
library(dplyr)
library(ggplot2)
library(tidyr)
library(caret)

### No zeros ###
## 5 cost groups gridsearch
gs <- read_excel("c://R/RData_Thesis1.xlsx",7)
gs1 <- cbind.data.frame(gs[3:13])
sapply(gs1, class)
gs1 <- transform(gs1,
                 Audit.Location=as.factor(Audit.Location),
                 Cost.Pool=as.factor(Cost.Pool),
                 SubService.Category=as.factor(SubService.Category),
                 Date.Diff.Cont=as.integer(Date.Diff.Cont),
                 FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                 FT.ITPEC.N=as.factor(FT.ITPEC.N),
                 ITPECNEW3.N=as.factor(ITPECNEW3.N),
                 FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                 IT.Expenditure.N=as.factor(IT.Expenditure.N),
                 Tier.N=as.factor(Tier.N),
                 Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

#Control function for training with 10 folds of cross validation
control <- trainControl(method='repeatedcv',
                        number=10,
                        repeats=3,
                        search='grid')

tunegrid <- expand.grid(.mtry=(1:12))
re_gridsearch <- train(Expense.Amount.Validated.N~.,
                       data=gs1,
                       method='rf',
                       metric='Accuracy',
                       tuneGrid=tunegrid)
print(re_gridsearch)

## RF Packages and Code Excursion Model for Balanced Classes ##
library(readxl)
library(car)
library(MASS)
```

```
library(FactoMineR)
library(randomForest)
library(caTools)
library(h2o)
library(jsonlite)
library(dplyr)
library(ggplot2)
library(tidyr)
library(caret)

## Excursion Balanced Classes
## Six Category RF model, full range
bc <- read_excel("c://R/RData_Thesis1.xlsx",6)
bc1 <- cbind.data.frame(bc[3:13])
sapply(bc1, class)
bc1 <- transform(bc1,
                 Audit.Location=as.factor(Audit.Location),
                 Cost.Pool=as.factor(Cost.Pool),
                 SubService.Category=as.factor(SubService.Category),
                 Date.Diff.Cont=as.integer(Date.Diff.Cont),
                 FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                 FT.ITPEC.N=as.factor(FT.ITPEC.N),
                 ITPECNEW3.N=as.factor(ITPECNEW3.N),
                 FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                 IT.Expenditure.N=as.factor(IT.Expenditure.N),
                 Tier.N=as.factor(Tier.N),
                 Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

samplebc = sample.split(bc1$Tier.N, SplitRatio = 0.75)
trainbc = subset(bc1, samplebc == TRUE)
testbc = subset(bc1, samplebc == FALSE)
h2o.init(nthreads = 20, max_mem_size = "6g")

trnbc <- as.h2o(trainbc)
tstbc <- as.h2o(testbc)
rfbc <- h2o.randomForest(y="Expense.Amount.Validated.N",
                         training_frame=trnbc, validation_frame=tstbc ,stopping_rounds = 5,
                             stopping_tolerance = 0.001,
                         stopping_metric = "AUC", seed = 29, balance_classes = TRUE, nfolds = 10,
                         mtries = -2, ntrees=100)

perfbc <- h2o.performance(rfbc);perfbc
h2o.varimp(rfbc)

#Predictors, use as needed
predbc <- h2o.predict(rfbc, newdata=as.h2o(trainbc));predbc
predbc2 <- h2o.predict(rfbc, newdata=as.h2o(testbc));predbc2
h2o.exportFile(predbc2, path = 'c://Users/Ajax/Desktop/rfexcurvalid.csv', force = TRUE)
h2o.exportFile(h2o.getFrame("trainbc_sid_a493_9"), path = 'c://Users/Ajax/Desktop/rfexcur.csv',
     force = TRUE)

##############################################################################
## 5 cost groups, no zero
cat6 <- read_excel("c://R/RData_Thesis1.xlsx",8)
```

```
cat6_1 <- cbind.data.frame(cat6[3:13])
sapply(cat6_1, class)
cat6_1 <- transform(cat6_1,
                    Audit.Location=as.factor(Audit.Location),
                    Cost.Pool=as.factor(Cost.Pool),
                    SubService.Category=as.factor(SubService.Category),
                    Date.Diff.Cont=as.integer(Date.Diff.Cont),
                    FT.Is.Contract.N=as.logical(FT.Is.Contract.N),
                    FT.ITPEC.N=as.factor(FT.ITPEC.N),
                    ITPECNEW3.N=as.factor(ITPECNEW3.N),
                    FT.ITRCCC.N=as.factor(FT.ITRCCC.N),
                    IT.Expenditure.N=as.factor(IT.Expenditure.N),
                    Tier.N=as.factor(Tier.N),
                    Expense.Amount.Validated.N=as.factor(Expense.Amount.Validated.N))

sample6 = sample.split(cat6_1$Tier.N, SplitRatio = 0.75)
train6 = subset(cat6_1, sample6 == TRUE)
test6 = subset(cat6_1, sample6 == FALSE)
h2o.init(nthreads = 20, max_mem_size = "6g")

trn6_f <- as.h2o(train6)
rf6_f <- h2o.randomForest(y="Expense.Amount.Validated.N",
                          training_frame=trn6_f, stopping_rounds = 5, stopping_tolerance = 0.001,
                          stopping_metric = "AUC", seed = 29, balance_classes = TRUE, nfolds = 10)

perf6_f <- h2o.performance(rf6_f);perf6_f
h2o.varimp(rf6_f)
```

Bibliography

1. Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, RCoreTeam, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan, and Tyler Hunt. *Package 'caret'*, 2021.

2. H2O.ai. *H2O.ai 3.34.0.3 Documentation*, 2021.

3. Michael P. LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

4. Andrew Worster, Jerome Fan, and Afisi Ismaila. Understanding linear and logistic regression analyses. *Canadian Journal of Emergency Medicine*, 9(2):111–113, 2007.

5. Bonaccorso Giuseppe. *Machine Learning Algorithms : Popular Algorithms for Data Science and Machine Learning, 2nd Edition.*, volume 2nd ed. Packt Publishing, 2018.

6. Habibollah Esmaily, Maryam Tayefi, Hassan Doosti, Majid Ghayour-Mobarhan, Hossein Nezami, and Alireza Amirabadizadeh. A comparison between decision tree and random forest in determining the risk factors associated with type 2 diabetes. *Journal of Research in Health Sciences*, 18:412–419, 2018.

7. Philipp Probst, Marvin N. Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3), 2019.

8. Marcel Neunhoeffer and Sebastian Sternberg. How cross-validation can go wrong and what to do about it. *Political Analysis*, 27:101–106, 2019.

9. Carl Kingsford and Steven L. Salzberg. What are decision trees? *Nature Biotechnology*, 26(9):1011–1013, 2008.

10. Bo Hu, Jun Shao, and Mari Palta. Pseudo-r 2 in logistic regression model. *Statistica Sinica*, 16:847–860, 07 2006.

11. Anne Ruiz-Gazon and Nathalie Villa. Storms prediction: Logistic regression vs random forest for unbalanced data. *arXiv preprint arXiv:0804.0650*, 2008.

12. Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11(1):1–13, 2011.

13. Min Zhu, Jing Xia, Xiaoqing Jin, Molei Yan, Guolong Cai, Jing Yan, and Gangmin Ning. Class weights random forest algorithm for processing class imbalanced medical data. *IEEE Access*, 6:4641–4652, 2018.

14. Evgeny A Antipov and Elena B Pokryshevskaya. Mass appraisal of residential apartments: An application of random forest for valuation and a cart-based approach for model diagnostics. *Expert Systems with Applications*, 39(2):1772–1778, 2012.

15. Jengei Hong, Heeyoul Choi, and Woo-sung Kim. A house price valuation based on the random forest approach: the mass appraisal of residential property in south korea. *International Journal of Strategic Property Management*, 24(3):140–152, 2020.

16. Rohit Joshi, Rohan Gupte, Palanisamy Saravanan, et al. A random forest approach for predicting online buying behavior of indian customers. *Theoretical Economics Letters*, 8(03):448, 2018.

17. Karim Baati and Mouad Mohsil. Real-time prediction of online shoppers' purchasing intention using random forest. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 43–51. Springer, 2020.

18. Kailong Liu, Xiaosong Hu, Huiyu Zhou, Lei Tong, Dhammika Widanalage, and James Marco. Feature analyses and modelling of lithium-ion batteries manufacturing based on random forest classification. *IEEE/ASME Transactions on Mechatronics*, 2021.

19. Federal Register. Federal acquisition regulation: Increased micro-purchase and simplified acquisition thresholds, 2020.

20. PA Flach. The geometry of roc space: understanding machine learning metrics through roc isometrics. pages 194 – 201, United States, 2003. AAAI Press. Conference Proceedings/Title of Journal: Proc. 20th International Conference on Machine Learning (ICML'03).

21. Anjali S. More and Dipti P. Rana. An experimental assessment of random forest classification performance improvisation with sampling and stage wise success rate calculation. *Procedia Computer Science*, 167:1711–1721, 2020.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2022	Master's Thesis	Sept 2020 — Mar 2022

4. TITLE AND SUBTITLE

Training LOGIT and Random Forest Models to Predict IT Spending

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Batt, Jacob P, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-22-M-117

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Audit Agency
Statistician, Daniel Rosenblatt, PhD
470 I St East B745, JBSA-Randolph, TX 78150
DSN 487-0727, COMM (210)-652-0727
Email: daniel.rosenblatt@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFAA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Air Force must modernize, but the distribution of funds for technology remains as tight as ever. To this end, the Air Force Audit Agency is looking to utilize machine learning techniques to enhance their capabilities. This research explores Logistic Regression and Random Forest modeling to streamline data collection and cost classification. The final Logistic Regression model identified 4 significant attributes out of the 36 given and was 85% accurate in predicting whether a purchase amount was over or under $10,000. To expand beyond binary classification, a six-category classification Random Forest model was developed. It identified 6 significant attributes and was 34% accurate in in predicting whether a purchase was in 1 of 6 amount categories. Due to the class imbalance of the given data, it was necessary to use a class weighting and over-sampling technique to enhance the Random Forest model. The final class balanced model identified the same 6 significant attributes but was 78% accurate in predicting whether a purchase was in 1 of 6 amount categories. No models were able to predict whether a purchase should be classified as an information technology purchase of not.

15. SUBJECT TERMS

audit, random forest, logistic regression (LOGIT), machine learning, predictive modeling, bootstrapping, R programming, imbalanced data

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Raymond R. Hill, AFIT/ENS
U	U	U	UU	64	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x7469; rhill@afit.edu