BUDA 535 Final

Group 5

4/27/2020

### Problem 1 (30 points)

Classification problems focus on trying to sort the available data into two or more discrete categories (e.g., tumor malignancy or political party). There are many possible methods to perform this; each with its own strengths and weaknesses.

Linear Discriminant Analysis (LDA) models look at all the variables in a dataset as having the same covariance matrices and the same data cloud shapes. This means there is a linear boundary between variable classes that separate what predicted variables classes will fall into. This type of model is easy to implement/code and good for making quick prediction models. This is a good method to use to look at a prediction and if it has a high accuracy to research deeper. If the method does not predict well, you can move on to another method without having wasted a ton of time. The drawback of the LDA model is that the larger the dataset gets, the harder to predict under the assumption all the data follows the same data clouds and covariances. When the variable classes have very different covariance matrices, the model gets introduced to a high level of bias. With this assumption behind the model, it is not the best to use to explain the inner workings to a stakeholder but would be good for being able to run quick prediction accuracies.

Quadratic discriminant analysis (QDA) models are the opposite assumption of the LDA model. The QDA model assumes the variables have different covariance matrices and all the data clouds have different means, shapes, and centers. QDA models use a non-linear decision boundary lines to predict. This means the lines try to pick up different means in the data when separating variables. As with the LDA model, QDA is also good for quick turnaround time to make predictions with its simple coding function. The most optimal time for a QDA would be with a non-linear relationship model. As with LDA, if this model does not predict well, you would not have wasted a lot of time and can move on to a different method quickly. QDAs are not optimal for large datasets as the more parameters the QDA model needs to estimate, the higher the variance of the model is. This model is also not the easiest to explain the workings behind the math to a stakeholder but is easy to run fast prediction models if needed.

There is also the idea of regularized discriminant analysis (RDA) models. This type of model is hybrid between LDA and QDAs. RDAs use a tuning parameter of alpha between 0 and 1. The closer to 0 alpha is, the more LDA leaning the model will be (covariances pooled). The closer to 1 alpha is, the more QDA leaning the model will be (covariances independent). RDAs work best when the data set is large as you can account better for the linear and non-linear relationship trade offs by using the tuning parameters to your advantage. Since the tuning parameters are added to RDAs, this makes the coding more complex than the LDA and QDA methods. This would also increase the difficulty of explaining to a stakeholder as the complexity increases. But as with the previous LDA and QDA methods, RDAs are able to give you a good prediction accuracy measurement and insight to research data deeper.

Naïve Bayes models assume that all the variables are not correlated (independent) to each other. This type of model, as with LDA and QDAs, are simple coding models and good for real-time predictions. This model works best with categorical variables. When independence assumptions are true in the data, this model prediction is the most accurate compared to other models. The drawback of this model though is that if a category variable is not in the training data, it will assign that variable a 0 probability and not predict it. It also assumes all predictors are independent of each other when in reality this would be a very uncommon assumption. This would also not be the easiest to explain to a stakeholder as with the LDA and QDA methods but would give quick prediction outcomes.

Another set of concepts we learned in BUDA 535 regarding model evaluating was ridge, lasso, and elastic net. Ridge regression helps us lower the complexity of models by removing certain variable’s effect on the response. This is done by penalizing the sum of the coefficients and setting alpha in our model to 0. This method does not do variable selection but is able to give insights on the variables and is known to increase prediction accuracy. This is best when there is a large number of parameters that have similar weights on influencing the response. Lasso in regression penalizes the sum of the absolute values of the coefficients and sets alpha equal to 1. This can set some coefficients to 0 which helps with variable selection. Lasso does better when there is a small number of significant parameters that influence the response. Elastic net is a combination of the two penalties. Elastic net aims to minimize the loss function by finding an alpha between 0 and 1.

The multinomial logistic regression is an extension of binomial regression for conditions where more than two outcomes are possible. The example in the text referred to the nes96 data where the response was that each person could belong to one of three political parties. The multinomial logit model can be used to predict classification on new data very well. Unlike Naïve Bayes and other methods, there is no need for the independent variables to be statistically independent from each other. Multinomial logistic regression often provides better insight into how and which predictors affect the classification. This is invaluable in predicting how a model will behave in new circumstances or how predictors are added or modified. AIC and step may also be used to identify important variables and simplify the model. However all of this comes with a cost. The complicated output of the model may be very accurate, but not easily interpreted. This may make customer meetings very difficult and complicate their buy-in to your prediction.

Tree based classifiers use a series of yes and no questions on a set of variables at different points (called nodes) to predict a response variable. This allows us to divide up the space of the predictor variables so that we can get better predictions. They operate very much like an if then statement with a binary response. They work well on very simple datasets that do not have many interactions between variables. This allows for easy interpretability and explanation to end users. For example, when using a CART model, we can plot the entire model and show the criterion used at each node. When using Bagging and Random Forest models we lose some of this functionality, but we can output the variable importance to see which variables have the largest impact on the generated trees. Data requirements for building a tree based model does not require scaling or data normalization.

Tree based classifiers can be highly dependent on data and small changes in the training set can have large implications on the overall model. Another consideration when using a tree based classifier is the computational cost and time as they are expensive to run compared to other methods. It can take significantly longer to train a tree based model than other methods. Finally, calculations can sometimes be far more complex than other algorithms.

In this course we went over three different tree based methods: recursive partitioning, Bagging, Random Forests, and Boosting. Recursive partitioning begins by making a prediction based upon a rule which predicts the response. The following subsets of data are split again until all data points have been predicted or the splitting no longer improves prediction. A downside to this method is that the observations themselves are being split and then predicted, and it does not estimate what is happening to the population. This causes slight changes in the data to sometimes have a profound impact on the tree due to the data dependency. Bagging addresses some of the data dependency that is created in using recursive partitioning. In bagging we create a bootstrap sample of our data with replacement then fit trees on those samples, and then take the average of the predictions of all the trees. With this method we lose the “pretty” picture that we get with recursive partitioning since we are averaging over our trees to produce the prediction. Computationally this method takes more time than the recursive partitioning. Random Forests attempts to speed up bagging by creating multiple trees using bootstrap samples then using a randomly selected subset of variables at each node. This also improves the robustness of the model since no two trees would be the same. We then average over all the trees to get a better prediction. Boosting works like bagging, but trees are grown sequentially. We iterate over our prediction in order to shrink the residuals to grow out the trees. Boosting can overfit and is highly dependent on tuning the model correctly.

### Problem 2 (20 Points)

Clustering refers to several techniques for finding similarities among data points by sorting them into subgroups. Clustering the data allows the user to make assumptions about the entire subgroup leading to simpler and quicker analysis. A typical example of clustering is a market segmentation effort to identify people more likely to purchase a product. Both K-Means and Hierarchical Clustering use distance in order to separate observations into different groups. However, there are differences in these methods that are important in deciding when to use each method.

K-means clustering is a type of unsupervised learning, which is used to find similar groups within the data. The variable K represents the number of groups which is defined by the user. The K-means algorithm partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. K-means is a very simple yet effective method of cluster analysis. In K-means, the number of clusters is predetermined by the user.

Hierarchical clustering is an unsupervised clustering algorithm that involves creating clusters that have a predetermined ordering from top to bottom. Hierarchical clustering starts with k = N clusters and proceeds by merging the two closest objects into one cluster, obtaining k = N-1 clusters. The process of merging two clusters to obtain k-1 clusters is repeated until the desired number of clusters K is reached. Euclidean distance is used to find which clusters to merge. There are two types of hierarchical clustering, Divisive and Agglomerative. Hierarchical clustering utilizes a dendrogram, which is a type of tree diagram showing the relationships between similar sets of data. Branches in the dendrogram represent the similarities among the data; the shorter the branch, the greater similarity of the data. This has the advantage of providing an attractive, graphical representation of the data.

Hierarchical clustering works especially well with smaller data sets. On the other hand, K-means works very well with larger data sets. This is because the time complexity of K Means is linear while that of hierarchical clustering is quadratic. Also Hierarchical clustering is able to work with a variety of data while K-means is only appropriate for numeric data.

Agglomerative clustering known as AGNES (Agglomerative Nesting) works in a bottom-up method. Each object is initially considered as a single-element cluster. At each step of the algorithm, the two clusters that are the most similar are combined into a new larger cluster. This procedure is iterated until all points are members of just one single big cluster. The result is a tree which can be plotted as a dendrogram. The AGNES function also provides the agglomerative coefficient, which measures the amount of clustering structure found.

Divisive hierarchical clustering known as DIANA (Divise Analysis) works in a top-down method. The DIANA algorithm constructs a hierarchy of clusters, starting with one large cluster containing all n observations. Clusters are divided until each cluster contains only a single observation.

Fuzzy Analysis Clustering known as FANNY computes a fuzzy clustering of the data into k clusters. Fuzzy clustering is considered soft clustering, in which each element has a probability of belonging to each cluster.

Divisive Hierarchical Clustering of Binary Variables known as MONA allows us to cluster on binary variables. Mona is a hierarchical algorithm that is divisive, but it is referred to as monothetic, which means the clustering is only determined one variable at a time. It allows distance to be measured a single variable at a time.

Clustering of large applications known as CLARA relies on the sampling approach to handle large data sets.

One potential disadvantage to K-means clustering is that it requires us to pre-determine the number of clusters.

Hierarchical clustering algorithms can be rather complex in relation to K-means algorithms and have large storage requirements and are computationally expensive. Hierarchical clustering can also yield worse results for certain types of data with obvious clustering (e.g., the nationality and gender example in the text).

The gap statistic, which compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data, may be used to determine the number of clusters. The estimate of the optimal clusters will be the value that maximizes the gap statistic. This means that the clustering structure is far away from the random uniform distribution of points.

The elbow method uses the sum of squares at each number of clusters. This is calculated and graphed. The user then searches for a change of slope from steep to shallow (an elbow) to determine the optimal number of clusters.

The NbClust package provides 30 indices for determining the relevant number of clusters. This method proposes the best clustering scheme from the different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods.

The Clustree package shows how samples change groupings as the number of clusters increases. This is useful for showing which clusters are distinct and which are unstable. It doesn’t explicitly tell you which choice of optimal clusters is but it is useful for exploring possible choices.

### Problem 3 (20 Points)

#### Training Data

We’ve separated approximately 500 rows to be used as the test data set. The remaining 4096 rows will be used for the training set. Further analysis of the data set indicates that 1614 (or 39%) of the emails in the training set are spam. We also confirmed that 39% of the emails in the test set are spam. Thus the test set aligns proportionally to the training data.

#### Generalized Linear Model

The best tuned GLM model yielded an accuracy of 93.6% against the training data and 91.5% against the test data. The glm used all of the variables, but with differing weights to show the impact. However, it was clear that at least 12 variables were highly significant.

#### Random Forest

The random forest model yielded an accuracy of 95.2% against the training data and 94.1% against the test. The model focused on 30 variables. However, the weighting was very different from the GLM. Of the top 12 variables in the GLM, only five were prioritized in the random trees. The remainder were present in the random forest model, but at much lower priority. This smaller model provided a list of easily explainable terms to advise the client on how to detect spam.

#### Elastic Net

The elastic net model set alpha to 0.55 and lambda to 0.0005. The model selected all variables but this time selected only five of the 12 variables prioritized by the random forest method. The ridge regression model yielded an accuracy of 93.6% on the training data and 91.5% against the test data. However, the model used all of the variables making interpretation and explanation challenging.

#### Naive Bayes

The naive bayes model yielded an accuracy of 89.4% on the training model but improved on the test data with an accuracy of 91.9%. However, this model was much worse than the others. It gave twice the number of false positives (email identified as spam) compared to the best performing models.

#### SVM

The SVM model yielded an accuracy of 93.8% on the training data and 92.9% against the test data.

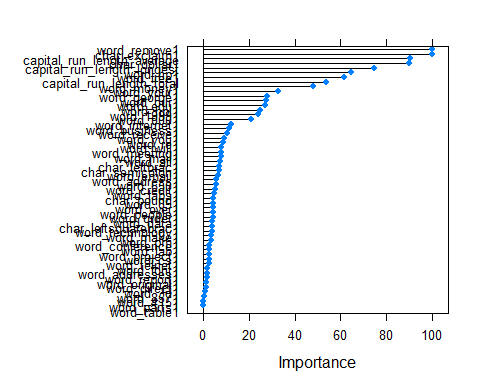
#### LDA

The LDA model provided an accuracy of 92.3% against the training data and 92.3% against the test data.

#### Statistics

## train Accuracy test Accuracy test Sensitivity  
## GLM 0.9355465 0.9148515 0.8643216  
## Random Forest 0.9519059 0.9405941 0.9095477  
## Elastic Net 0.9357934 0.9148515 0.8643216  
## Naive Bayes 0.8944891 0.9029703 0.8743719  
## SVM 0.9384751 0.9188119 0.8592965  
## LDA 0.9230801 0.9108911 0.8442211  
## test Specificity Precision  
## GLM 0.9477124 0.9148936  
## Random Forest 0.9607843 0.9378238  
## Elastic Net 0.9477124 0.9148936  
## Naive Bayes 0.9215686 0.8787879  
## SVM 0.9575163 0.9293478  
## LDA 0.9542484 0.9230769

## $title  
## [1] "Confusion Matrix"  
##   
## $GLM\_model  
## Reference  
## Prediction email spam  
## email 290 27  
## spam 16 172  
##   
## $randomforest\_model  
## Reference  
## Prediction email spam  
## email 294 18  
## spam 12 181  
##   
## $Ridge\_model  
## Reference  
## Prediction email spam  
## email 290 27  
## spam 16 172  
##   
## $NaiveBayes\_model  
## Reference  
## Prediction email spam  
## email 282 25  
## spam 24 174  
##   
## $SVM\_model  
## Reference  
## Prediction email spam  
## email 293 28  
## spam 13 171  
##   
## $LDA\_model  
## Reference  
## Prediction email spam  
## email 292 31  
## spam 14 168



#### Conclusion

Based on the varImp plot, we see that several variables were very important in determining whether an email was spam or not. Among these were the overuse of symbols such as char\_exclaim1, char\_dollar1; the frequency of capitals (capital\_run\_length\_average and capital\_run\_length\_longest) and the use of some key words (word\_free1, word\_money1 and word\_remove1). These seemed to be indicators that an email was spam.

The random forest model provided the simplest and most accurate model. It had the highest accuracy for both the training and test data. The model also had the highest specificity and sensitivity of 96% and 91% respectively. That means 96% of observations were correctly identified as emails and 91% correctly identified as spam.

It significantly reduced the number of variables providing cleaner interpretation. Finally, it yielded the fewest false positive indications. In this instance, falsely identifying an actual email as spam has higher potential for lost revenue or negative customer satisfaction than falsely identifying spam as an email. The potential for lost sales, missed critical communication or impact to potential business is great. Reducing false positives must be a critical consideration and the random forest model did that the best.

### Problem 4 (30 Points)

#### Elastic Net

The Elastic Net was one of our strongest models with 84% prediction accuracy on the test data. The model had 224 false positives and 215 false negatives. The best tune for the model was at an alpha of .1 and lambda of .0005

#### Ridge

The Ridge model had a prediction accuracy of 83% on our test data. The model had 290 false positives and 181 false negatives. The best tune selected a lambda of .1.

#### Lasso

The lasso model did not perform well compared to our other models with a prediction accuracy of 79% on our test data. The model had 375 false positives and 199 false negatives. The only variables that were kept in the final model was year and productivity.

#### LDA

The LDA model performed well with a prediction accuracy of 84% on the test data. The model had 218 false positives and 219 false negatives.

#### QDA

The QDA model only had a prediction accuracy of 75% on our test data. The model had only 69 false positives but 619 false negatives.

#### Rpart

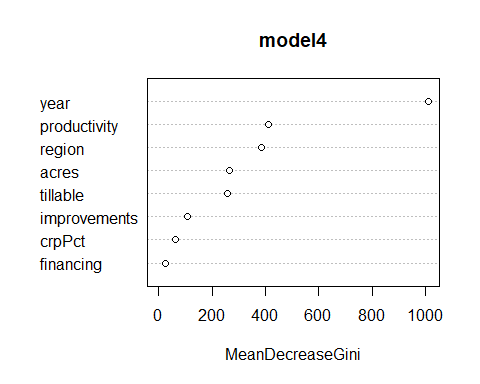
The Rpart model only had a prediction accuracy of 77% on our test data. The model had 518 false positives and 102 false negatives.

#### Model Selection

## train Accuracy test Accuracy test Sensitivity  
## Elastic Net 0.8486757 0.8415162 0.8096856  
## Ridge 0.8354960 0.8299639 0.7536109  
## Lasso 0.7923325 0.7927798 0.6813934  
## LDA 0.8419972 0.8422383 0.8147833  
## QDA 0.7401810 0.7516245 0.9413764  
## Rpart 0.7956644 0.7761733 0.5598980  
## RandomForest 0.8511581 0.8472924 0.7799490  
## test Specificity Precision  
## Elastic Net 0.8650345 0.8159247  
## Ridge 0.8863779 0.8305243  
## Lasso 0.8750785 0.8011988  
## LDA 0.8625235 0.8140917  
## QDA 0.6114250 0.6415750  
## Rpart 0.9359699 0.8659658  
## RandomForest 0.8970496 0.8484288

We used a variety of classification models and based our selection upon two criteria prediction accuracy and interpretability. The three best models from a prediction accuracy standpoint were LDA (84%), GLM with Elastic Net (84%), and Random Forest(85%). We also wanted to choose a model that would allow us to easily explore the variables to see if any were driving accuracy levels. The model we chose that best satisfied this criteria was the random forest model.

#### Analysis



Using the random forest model, we achieved a prediction accuracy of 85% with all the predictor variables in the model. We were able to see that the year variable almost doubled the next closest variable (productivity) in importance. If we used the same random forest model and removed the year variable, the prediction accuracy dropped to 69% which confirmed that year is our biggest driver of the model. When digging into the year variable a little deeper. We noticed that from the years 2002-2007, approximately 77% of the sales had below the mean acre prices. From the years 2008-2011, that number dropped to approximately 25%. The random forest model also predicted more false negatives as it leaned towards predicting more things below the mean acre price as there were. This led us to believe that our model undervalued the mean acre price. This could have two possible impacts. Either the seller using the model would set the sale price too low or the buyers using the model wouuld not be prepared to bid enough to win the sale.

Looking at the misclassified data points we noticed that we missed a signficant number in the South Central region. We also observed that 99 out of the 119 misclassified data points were title transfers. Out of all of our misclassifed data points we missed the most on the title transferred properties.

The random forest model is a good model to choose for this type of problem because of its multiple sampling of trees that the model runs. The random forest model runs many randomized samples at each node which creates numerous amounts of trees. From those trees, it averages out to get the best prediction. With all the randomness of the tree building and how it averages out to a high prediction accuracy, this method was the strongest to use. This type of model also provides variable importance values which are something we can show and explain to a business stakeholder along with the percentages.

In the end, we are very comfortable using this random forest model compared to the others we tried. The prediction accuracy, variable importance, and tree randomization are all present for this model and can be explained in a way to a business stakeholder that they would understand and trust.