**Individual Assignment by Nitesh Shanbhag**

As part of this Individual Assignment, A Telco Dataset was chosen to Predict Churn of the Customers. Once the Base table was created the Important Variables were selected using Pearson Correlation, Forward Selection, Backward Selection, Hybrid Selection etc. and final list of variables were obtained. Once Variables were Finalized 3 Models were chosen to predict the Churn, Logistic Regression, Decision Trees and Random Forest. Then we see that there is class imbalance in data, So Under sampling and Mixed Sampling was applied along with Bagging and 5\*2 Cross validation Procedure and Final Output Measures for all the models were obtained.

Below the explanation of each Statistical model used for modelling.

**Logistic Regression:**

For predicting Churn or any Classification problem. Logistic regression helps in finding the probability that that row/customer has to each class.

In the example that I have chosen whether to predict Churn for Telco companies. P (Churn = Yes / No based-on Predictor Variables)

So, the Output will be any value between 0 and 1 and we need to choose the cutoff and decide that for example if p>0.5 we take it as 1 and less than or equal we take it as 0 this will depend on the business problem.

The Logistic Model:

P(Churn) = B0 + B1 x1+ B2 x2 + B3 x3 + Bn xn

n = number of predictors.

At first the equation (Logistic Function) will be

P(Churn) = ( e^(B0 + B1x1+B2x2+….Bnxn) )/

(1+ e^(B0 + B1x1+B2x2+….Bnxn))

To Fit the above model The Maximum Likelihood is used. To explain maximum likelihood shortly it optimizes all the values of the coefficient in order to make it most optimal solution to the given set of predictors and Target variables in train.

In any statistical Problems odds are first thing that comes to mind. What are the odds that the Churn is going to happen and what are the odds that churn is not going to take place.

In Maths Odds are calculated by:

P(X) / (1-P(X) ) = e^(B0 + B1x1+B2x2+….Bnxn)

Odds = Discussed above gives P values.

Taking log on both sides

Log(P(X) / (1-P(X)) = B0 + B1x1+B2x2+….Bnxn

Left hand side is called Logit function. We see that Logit function of x1,x2 and all the predictors.

In Linear Regression if x1 is doubled the y also is affected 2 times but in Logistic regression as there is Log value applied on both sides. For each change in x1 the target variable is affected log times and not directly.

For example, if x1 increases, target or logit is affected by log odds of B1. Or equivalently e^ B1

So, depending on Value of B1, if Positive B1: the increasing x1 will lead to increasing P(X1) and if B1 is negative the increasing x will lead to decreasing y or P(X1)

Explaining how the Coefficients are calculated:

Based on the value of predictors in training dataset

B0 ,B1 ….Bn are calculated keeping in mind that the optimal value of each coefficients will lead to Value close to 1 for actual churners which is known from train dataset and 0 for non-churners in train dataset. This is obtained with the help of maximum likelihood function.

One very good method followed for maximum likelihood is to reduce the error. Here error can be defined as the actual V/s predicted for each case. So, the best set of coefficients will be chosen as final one which leads to minimum error and close fit to actual data.

This can also be achieved using Optim function in R. The logistic Regression function that we use as in-built package to predict can also be used with self-written function of optim in R and it generally should give the same results.

**Making Predictions:**

Once the coefficients are calculated using the above method. The Values in coefficients and inputs are put and output of probability is obtained which is now considered the actual prediction.

When there are multiple variables the results obtained from one predictor is different from when there is multiple predictors. This method is called as confounding.

**Decision Trees:**

This method is applied both for Classification and Regression Problems.

Inorder to Predict a target this segments and stratifies the predictor variables into simple regions with the help of mean or mode of the training observations.

**Input Attributes:**

D features / attributes or predictors X1,X2 ……XD

Each Xj has domain Oj

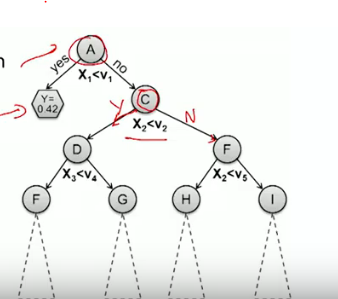
* Categorical Oj = {Internet,No Internet etc}
* Numerical Hj = {0,1,10 etc}

Y is the Output with domain Oy : Categorical Classification and Numerical Regression.

Data D:

N examples (Xi, yi) where xi is a d-dim feature vector yi belongs to Oy is the Output variable.

Decision tree is a tree structured plan of set of attributes to test inorder to predict the output.



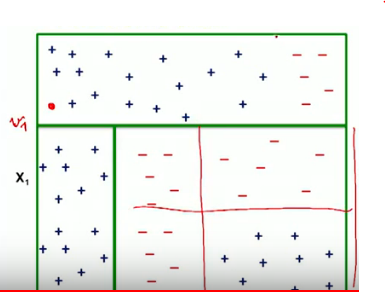
Here A is the Internal decision node the one denoted in Hexagon Y are Prediction nodes. Internal nodes havesplit values and leaf nodes make the predictions.

The Algorithm will be to Drop xi until it hits a leaf node.

In above example if x1 < v1 then it moves to y and it returns a value of 0.42 else it goes on to check for other nodes until it hits its prediction nodes and returns a value. All nodes will be having an end meaning a prediction value.

Imagine the First line is The First Split, Similarly if Yes then another line is drawn which is same as the above decision trees,

So If we see the below picture without having multiple decision lines it wouldn’t be possible to split the data into + and – hence the decisions lines helps a lot.



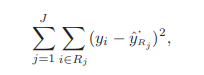
Regression Trees:

For Predicting the value for Regression trees below is the Algorithm that is followed.

1. The predictor space—that is, the set of possible values for X1, X2...,Xp—into J distinct and non-overlapping regions, R1, R2,...,RJ.
2. For every observation that falls into the region Rj, same prediction is made, which is simply the mean of the response values for the training observations in Rj.

Suppose there are 2 regions R1 and R2 and response mean of the value of Output in R1 of train is 10 then for test the response outout for all rows belonging to R1 will get a value of 10. And similarly if it belongs to R2 it will get the mean of R2 of train for all observations in test,

This is carried out by dividing the variable space with the help of high dimensional rectangles. The goal is to find R1,R2 Rj boxes by minimum RSS(error) given by



Where yRj is the mean response for the training observations within the jth box. As its not feasible to consider every partition into J boxes hence a Top Down approach is followed known as Recursive Binary Splitting.

Top because first split begins at the top of the tree and Greedy because it always chooses the best one for the split.

So inorder to do this at each split is considered which leads to maximum reduction in RSS.

So if first split leads to R1 and R2 then the cut off point is selected such that the sum of RSS of both regions remain minimum.

Then after the Split R1 and R2 are again split into smaller regions again keeping in mind minimum RSS of all 3 regions together.

This will continue until a stop point is reached. For example until each regions contain not more than 5 observations.

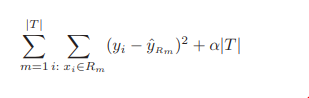
Once the Split at train is achieved. Observations from test set is mapped first to the regions R1 to Rj and once its mapped the mean value of train is assigned to all the observations of test belonging to that region.

**Pruning:**

If there are lots of trees and nodes then it would fit perfectly on train but will give bad results on test dataset. Which leads to overfitting so cross validation may be one approach but this will be cumbersome hence Pruning is a technique that is used.

Cost Complexity Pruning is Used to perform Pruning and this can be explained using a simple algorithm below.

1. By using above Procedure which we already discussed of creating a full and final decision tree.
2. Once decision tree is created Cost Complexity Pruning is applied to obtain sequence of best subtrees given by as a function of α.
3. Use K-fold cross-validation to choose α. That is, divide the training observations into K folds. For each k = 1,...,K: (a) Repeat Steps 1 and 2 on all but the kth fold of the training data. (b) Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α. Average the results for each value of α, and pick α to minimize the average error.
4. Return the subtree from Step 2 that corresponds to the chosen value of α



Basically, the Above Formula is used to Cost Complexity Pruning. Simple explanation to the above formula is: The RSS + α\*Number of Nodes should be minimum.

To explain it in simpler way, for example if there were 10 nodes at a tree and there was minimum RSS even and there was slightly higher RSS with just 3 nodes then the 3 nodes would be selected as it gives the sum of above formula to minimum.

So, with the help of this approach: There are not very small trees but just the right optimum number which helps solve over fitting and better predictions at test.

**Classification Tree:**

Like Regression tree but used to predict Qualitative response instead of qualitative one.

Growing Classification Tree is same as Regression tree but instead of RSS Classification Error rate is used.

the classification error rate is simply the fraction of the training observations in that region that do not belong to the most common class:

E = 1 − max k (ˆpmk).

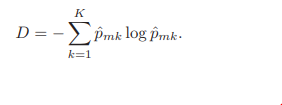
As Classification Error rate is not the optimum solution Gini Index is used.

Gini Index is given by :



If value of pmk’s are closer to 0/1 then Gini index takes a smaller value which is considered good. Which means that region contains observations from single class.

One more alternative is Cross Entropy



Since 0 ≤ pˆmk ≤ 1, it follows that 0 ≤ −pˆmk log ˆpmk. One can show that the cross-entropy will take on a value near zero if the ˆpmk’s are all near zero or near one. Therefore, like the Gini index, the cross-entropy will take on a small value if the mth node is pure. In fact, it turns out that the Gini index and the cross-entropy are quite similar numerically.

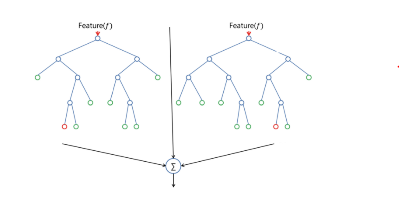
So If the each region is Pure and when the test observation falls into that region the prediction correctness is higher.

**Random Forest:**

Random Forest is a supervised learning algorithm. The forest it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

It can be used for both Classification and regression problems.



Instead of using decision trees and later adding bagging we can directly use the Random Forest.

Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Another good quality of random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction.  measures a features importance by looking at how much the tree nodes, which use that feature, reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results, so that the sum of all importance is equal to 1.

In a decision tree each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).A node that has no children is a leaf.

Through looking at the feature importance, decision is taken which features it may want to drop, because they don’t contribute enough or nothing to the prediction process. This is important, because a general rule in machine learning is that the more features we have, the more likely our model will suffer from overfitting and vice versa.

In decision trees, A fresh sample of m predictors is taken at each split, and typically we choose m ≈ √p—that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors.

So Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors. Then in the collection of bagged trees, most or all of the trees will use this strong predictor in the top split. Consequently, all of the bagged trees will look quite similar to each other. Hence the predictions from the bagged trees will be highly correlated. Unfortunately, averaging many highly correlated quantities does not lead to as large of a reduction in variance as averaging many uncorrelated quantities. This means that bagging will not lead to a substantial reduction in variance over a single tree in this setting.

Random forests overcome this problem by forcing each split to consider only a subset of the predictors. Therefore, on average (p − m)/p of the splits will not even consider the strong predictor, and so other predictors will have more of a chance. We can think of this process as decorrelating the trees, thereby making the average of the resulting trees less variable and hence more reliable.

Using a small value of m in building a random forest will typically be helpful when we have a large number of correlated predictors.

**Important Hyperparameters:**

1. Increasing the Predictive Power
2. Increases the Model Speed.

**Steps Followed:**

A Telco Churn Dataset was chosen for this Project and a Classification problem was taken.

First Step was cleaning the data and preparing a basetable.

The Code for Cleaning and preparation of Basetable can be found in the Rcode submitted “**1.Rcode.R”**

Once the basetable was created.

Variable Selection Methods were run to select the best set of Variables.

**Forward Step Wise Subset Selection:**

In this method 1 strong variable which is having a high correlation with Churn is First chosen and then the second variable is chosen such that the combination of 1st and 2nd should be the best result similarly with each addition of variable the AUC AIC etc are all noted and the best set of variable can be taken as the final one.

From this we found below variables to be important

**Senior Citizen, gender, Dependents, tenure, Multiple Lines, Paperless Billing, Payment Method, Total Charges, MTM, Oneyear**

**Backward Step Wise Subset Selection:**

Here the approach is same as Forward Selection but here first prediction is done with all the predictors and then 1 by 1 each predictor is removed and the output measures are checked. So instead of adding 1 by one it deletes 1 by 1

Tenure , Streaming TV , DSL, Fiberoptic, MTM were considered very important

Multiple Lines, Online Security, Tech Support, Streaming Movies, Paperless Billing, Payment Method, Monthly Charges, Total Charges were considered second important

Oneyear and SeniorCitizen were considered lesser important with 1 star

**Hybrid Method Subset Selection**

Here this is combination of both Forward and Backward method. Here variables are added in sequence and also removed as and when seems not important

This also gave similar results as Backward Stepwise Selection.

This was the Final order of the variables

[1] "MTM" "Fiberoptic" "tenure"

[4] "DSL" "Online Security" "Streaming Movies"

[7] "Tec Support" "Paperless Billing" "Total Charges"

[10] "Online Backup" "Payment Method" "Senior Citizen"

[13] "One year" "Streaming TV" "Multiple Lines"

[16] "Monthly Charges" "Dependents" "gender"

[19] "No" "Two-year" "Partner"

[22] "Phone Service" "Device Protection"

**Pearson Corelation Method for Variable Selction**

Corelation between each Individual Variables and Churn Variable is Found out and only the variables having Correlation coefficient < .01 is taken as good predictors.

Below are the variables considered with this method.

"SeniorCitizen" "Partner" "Dependents" "tenure"

"OnlineSecurity" "OnlineBackup" "DeviceProtection" "TechSupport"

"StreamingTV" "StreamingMovies” “PaperlessBilling" "PaymentMethod"

"MonthlyCharges" "TotalCharges" "DSL"

"FiberOptic" "No" "MTM" "Oneyear"

"Twoyear"

Taking into consideration variables from Pearson Correlation:

AUC For Train and Test is Run and we get the AUC of 84.66 for Train and 84.55 for Test.

**Modelling:**

For Modelling, Logistic Regression, Random Forest and Decision Trees have been used.

First all the models have been applied without any sampling techniques and All the Performance Measures have been calculated.

In this dataset there is not a class imbalance as there is 0.33 ratio between Churners and non-churners.

But For Learning Purpose Sampling has been applied as in real life datasets there is going to be huge imbalance in data.

3 Sampling Methods have been applied.

**Undersampling:**

Under sampling means that suppose there are churners 70% and 30% non-churners. While training the model understands 70 times about Churners but only 30% about non-churners. Hence for the model to understand Both equally. Number of rows are reduced such that number of churners and non-churners are based on the under-sampling rate chosen.

In my case I have choosen Undersampling rate as 0.4 and 0.5 which means 0.5 means 50% Churners and 50% non churners randomly selected.

**Oversampling**

Oversampling is a concept where train Non churners too many times and Churners very less so when the model looks a little different features it suddenly understands and remembers.

This was not giving good results hence discarded

**Mixed Sampling:**

Perfect mix of Undersampling and oversampling is Mixed sampling. This is done so that we don’t miss out any information while Under sampling.

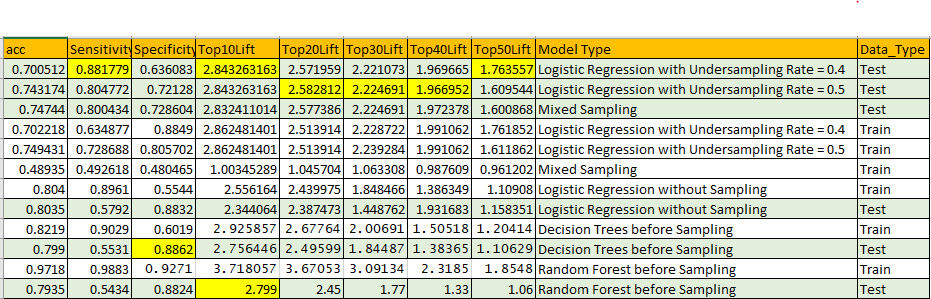
**Bagging with Logistic Regression:**

Bagging is generally a concept which is used in decision trees to avoid over fitting where if there are lots of cuts in train data of decision tree it tends to overfit and predict badly on test hence bagging is used so to select data randomly again and again from existing set so to avoid over fitting.

But when there is class imbalance also its necessary and helpful to do Bagging because when we do underfitting lots of information from non-churners is lost so its better to train with full sample of train dataset but by maintaining the under sampling ratio. Same applies for Mixed sampling.

So once when we get probabilities from all train. The average value of churn is taken into consideration. Also same for all the Prediction Measures.

**Output Measures:**



**Sensitivity:**

The Output of the model is displayed below.

To select which model to finalize depends on the Business.

If we take sensitivity into consideration where in the business would like to know what is the percentage of churners / non churners predicted right at the whole model. Here we are talking about (high **S**e**n**sitivity, **N**egative result = rule **out**).

A test that is 100% sensitive will identify *all* churners who have churned. It’s extremely rare that any test is 100% sensitive. A test with 88.2% sensitivity will identify 88.2% of churners who have got churned, but will miss 12% of chruners.

So if a business has the budget to look at whole data then Sensitivity is a good option and for this as we can see **Logistic Regression with Undersampling rate 0.4 is best**

**Specificity:**

The specificity of a test (also called the True Negative Rate) is the proportion of non churners who will have a **negative**result. In other words, the **specificity** of a test refers to how well a test identifies non churners. A test that has 100% specificity will identify 100% of non churners. A test that is 90% specific will identify 90% of non churners.

Tests with a high specificity (a high true negative rate) are most useful when the result is positive.

If based on Specificity if business wants to take decisions then Decision Trees before Sampling give better results

**Lift Values:**

In real life its impossible for the business to take action on all the customers and its important to target only a certain percentage of the customers.

That’s when Lift values come in handy.

If say business has the budget only for 10% of the customer, then Top10 lift values needs to be used. If its 20% then Top20 Decile lift needs to be used.

Similarly Top30 Top40 and Top50 can also be used.

**Top10 Lift**

If business says that it wants to target the 10% of the customers and they want maximum number of churners in Top 10 itself then Logistic Regression with Undersampling rate 0.4 is used.

**Top20 Lift**

If business says that it wants to target the 20% of the customers and they want maximum number of churners in Top 20 itself then Logistic Regression with Undersampling rate 0.5 is used

**Top30 Lift**

If business says that it wants to target the 30% of the customers and they want maximum number of churners in Top 30 itself then Logistic Regression with Undersampling rate 0.5 is used

**Top40 Lift**

If business says that it wants to target the 40% of the customers and they want maximum number of churners in Top 40 itself then Logistic Regression with Undersampling rate 0.5 is used

**Top50 Lift**

If business says that it wants to target the 50% of the customers and they want maximum number of churners in Top 50 itself then Logistic Regression with Under sampling with 0.4 under sampling rate is used

**Cumulative Gains Curve**

Cumulative Gain Chart for Logistic Regression with Undersampling rate 0.4, 0.5 and Logistic Regression with Mixed Sampling is shown below. It almost follows the same trend and deeper lift values can be studied from table above.

