School of Computer and Information Sciences
University of Hyderabad

Using Customer Behavior Data to Improve Customer Retention

(Under the esteemed guidance of Prof V. Ravi)

A project report submitted by

Vangala Sarveswara Rao

18MCMI18, M.Tech AI

Business Data Analytics | 06 – 11- 2018

Table of Contents

1. Abstract

Churn prediction is essential for businesses as it helps you detect customers who are likely to cancel a subscription, product or service. We have developed various machine learning models for churn prediction using the Telco Customer Churn dataset from IBM. We have performed the univariate feature selection, and selected 8 best performing features from the 20 features for predicting the churners. Then, we compared the performance of the various Classification approaches using the metrics such as Accuracy, Precision, Recall, and Area under ROC Curve (AUC) and choose the Classification and Regression Trees (CART) as the best model for the churn prediction.

2. Introduction

- Churn prediction can be extremely useful for customer retention and by predicting in advance customers that are at risk of leaving.
- You can do so by filtering out customers who don't seem to be using your product as much as they used to previously.
- Once you identify this group of customers, you can send out trigger emails to them based on their inactivity levels.
- Keeping the existing customer easier than the finding a new customer to a service or product.

3. Problem Statement

- Churn prediction aims at customer retention as it easier than acquiring a new customer.
- We can easily predict the customers who are going to churn if they are not using the product as often as before.
- It can be applied to any business which deals with customers and wants to retain them.
- Churn prediction comes under the classification task as we have to predict either customer will churn or not based on the previous data.
- Applying feature selection will improve the performance of the classification model as it reduces the dimensionality of the dataset.
- We will select the best model based on performance as well as human comprehensibility.

4. Data mining task identification

In churn prediction, we have to predict whether a customer will churn or not. This comes under **binary classification problem**.

5. Data understanding

5.1 DATA ACQUIRING

The Telco Customer Churn data set is taken from this url:

Name	format	Size in KB	To download
Telco Customer churn dataset	csv	954	Click Here

5.2 DATA SET DESCRIPTION

1. Number of Instances: 7043
2. Number of Attributes: 20

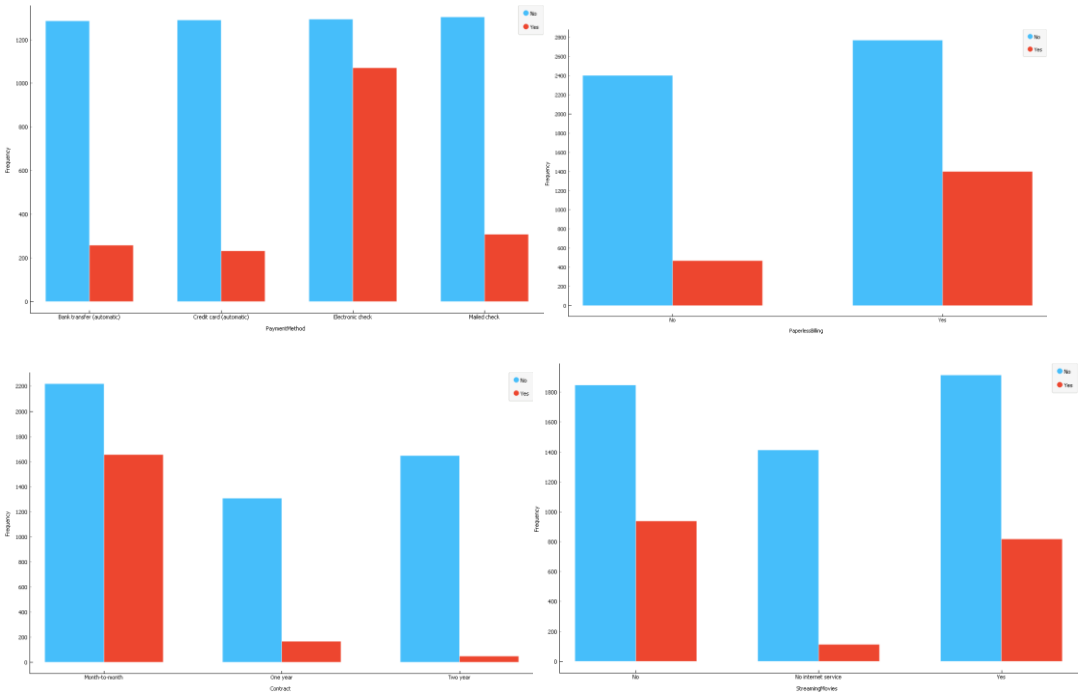
Feature	Description		Feature	Description
Customer ID	Unique to Every customer		Online Backup	Yes / No / No internet Service
Gender	Male/female		Device Protection	Yes / No / No internet Service
Partner	Yes/no		Tech Support	Yes / No / No internet Service
Senior Citizen	Yes/no		Streaming TV	Yes / No / No internet Service
Dependents	Yes/no		Streaming Movies	Yes / No / No internet Service
Tenure	An integer		Contract	Month-to-Month / One Year / Two Year
Phone Service	Yes/no		Paperless billing	Yes / No
Multiple lines	No phone service, Yes, No		Payment Method	Electric Check / Mailed Check / Bank Transfer(Automatic)/ Credit Card

Internet Service	DSL, Fiber Optic, No		Monthly Charges	Continuous value
Online Security	No internet Service, Yes, No		Total Charges	Continuous value

3. Number of Classes: 2
    - Churn = Yes
    - Churn = No
4. Missing values: The dataset contains missing values.

5.3 EXPLORING THE DATA

The below bar graphs shows the class distribution for each possible categories of the features.



6. Data Preparation

Data miners spend most of their time (60 – 70%) on the data preparation stage of CRISP-DM process. Hence, it is a very important phase in the CRISP-DM process.

6.1 DATA CLEANING

Data cleaning includes fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.

6.1.1 Handling missing values (Dropping the rows)

Missing values occurs in the dataset because some variables are not applicable, or not recorded. Either we can replace the missing values or delete the records which contains missing value.

In IBM Churn Dataset, only 11 rows contains the missing values out of 7043 rows. As it is a very small number, we can drop those rows. In our dataset, the following rows contains the missing values, and we removed those rows from the dataset.

```
[488, 753, 936, 1082, 1340, 3331, 3826, 4380, 5218, 6670, 6754]
```

6.2 DATA STANDARDIZATION (Z-SCORE)

Standardization of a dataset is a common requirement for many machine learning algorithms, as they except the data to have Gaussian distribution such as logistic regression. Standardize the features by removing the mean and scaling to unit variance.

$$z = \frac{x - \mu}{\sigma}$$

where:

μ is the mean of the population.

σ is the standard deviation of the population.

In our data set we have to standardize only some of the features such as tenure, monthly charges, and total charges.

```
col_names = ['tenure', 'MonthlyCharges', 'TotalCharges']
features = df[col_names]
scaled_features = StandardScaler().fit_transform(features)
df[col_names] = scaled_features
print(df[col_names])
```

	tenure	MonthlyCharges	TotalCharges
0	-1.280248	-1.161694	-0.994194
1	0.064303	-0.260878	-0.173740
2	-1.239504	-0.363923	-0.959649
3	0.512486	-0.747850	-0.195248
4	-1.239504	0.196178	-0.940457

The figure shows Standardized values for above mentioned features

6.3 DATA NORMALIZATION (MIN-MAX NORMALIZATION)

Min max normalization transforms features by scaling each feature to a given range.

The transformation is given by:

```
X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
X_scaled = X_std * (max - min) + min
```

where min, max = feature_range.

```
from sklearn.preprocessing import MinMaxScaler

col_names = ['tenure', 'MonthlyCharges', 'TotalCharges']
features = df[col_names]
scaled_features = MinMaxScaler().fit_transform(features)
df[col_names] = scaled_features
print(df[col_names])
```

	tenure	MonthlyCharges	TotalCharges
0	0.000000	0.115423	0.001275
1	0.464789	0.385075	0.215867
2	0.014085	0.354229	0.010310
3	0.619718	0.239303	0.210241
4	0.014085	0.521891	0.015330

3 feature values after applying min-max normalization

6.4 FEATURE SELECTION

Feature selection used to select important features to either improve the model performance or boost their performance in higher dimensional datasets. Statistical tests can be used to select those features that have the strongest relationship with the output variable.

6.4.1 F – Statistic

Here is what `f_regression` does, on input matrix X and array y. For every feature $X[:, i]$ it computes the correlation with y:

$$\rho_i = \frac{(X[:,i] - mean(X[:,i])) * (y - mean(y))}{std(X[:,i]) * std(y)}.$$

Then it computes the F-statistic

$$F_i = \frac{\rho_i^2}{1 - \rho_i^2} * (n - 2),$$

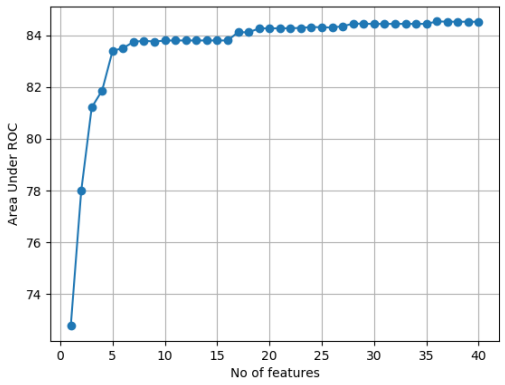
This is done in 2 steps:

1. The correlation between each regressor and the target is computed, that is, ((X[:, i] - mean(X[:, i])) * (y - mean_y)) / (std(X[:, i]) * std(y)).
2. It is converted to an F score

```
from sklearn.feature_selection import f_classif
test, _ = f_classif(X, y)

for i in range(len(X.columns)):
    print("%s F-value: %.3f" %(X.columns[i], test[i]))
```

```
gender F-value: 0.513
SeniorCitizen F-value: 163.012
Partner F-value: 161.776
Dependents F-value: 192.189
tenure F-value: 1007.509
PhoneService F-value: 0.961
PaperlessBilling F-value: 267.488
MonthlyCharges F-value: 271.577
TotalCharges F-value: 291.345
MultipleLines_No F-value: 7.504
MultipleLines_No phone service F-value: 0.961
MultipleLines_Yes F-value: 11.285
InternetService_DSL F-value: 110.036
InternetService_Fiber optic F-value: 733.952
InternetService_No F-value: 383.982
OnlineSecurity_No F-value: 932.622
OnlineSecurity_No internet service F-value: 383.982
OnlineSecurity_Yes F-value: 212.445
OnlineBackup_No F-value: 542.225
OnlineBackup_No internet service F-value: 383.982
OnlineBackup_Yes F-value: 47.949
DeviceProtection_No F-value: 476.931
DeviceProtection_No internet service F-value: 383.982
DeviceProtection_Yes F-value: 30.937
TechSupport_No F-value: 899.938
TechSupport_No internet service F-value: 383.982
TechSupport_Yes F-value: 196.052
StreamingTV_No F-value: 117.909
StreamingTV_No internet service F-value: 383.982
StreamingTV_Yes F-value: 28.240
StreamingMovies_No F-value: 122.596
StreamingMovies_No internet service F-value: 383.982
StreamingMovies_Yes F-value: 26.135
Contract_Month-to-month F-value: 1375.798
Contract_One year F-value: 230.628
Contract_Two year F-value: 703.210
PaymentMethod_Bank transfer (automatic) F-value: 99.500
PaymentMethod_Credit card (automatic) F-value: 129.884
PaymentMethod_Electronic check F-value: 702.709
PaymentMethod_Mailed check F-value: 58.406
```



The above figure shows the Features selected by ANOVA F-Value Vs their Area under ROC curve

Best Features according to ANOVA F-Statistic are:

Tenure	PaymentMethod_Electronic check	InternetService_Fiber Optic	OnlineSecurity_No
TechSupport_No	Contract_month-to-month	Contract_Two year	

6.4.2 Mutual information

Mutual information (MI) quantifies the amount of information obtained about one random variable through another random variable. MI is a non-negative value, it is equal to zero if two random variables are independent, and higher values means higher dependency.

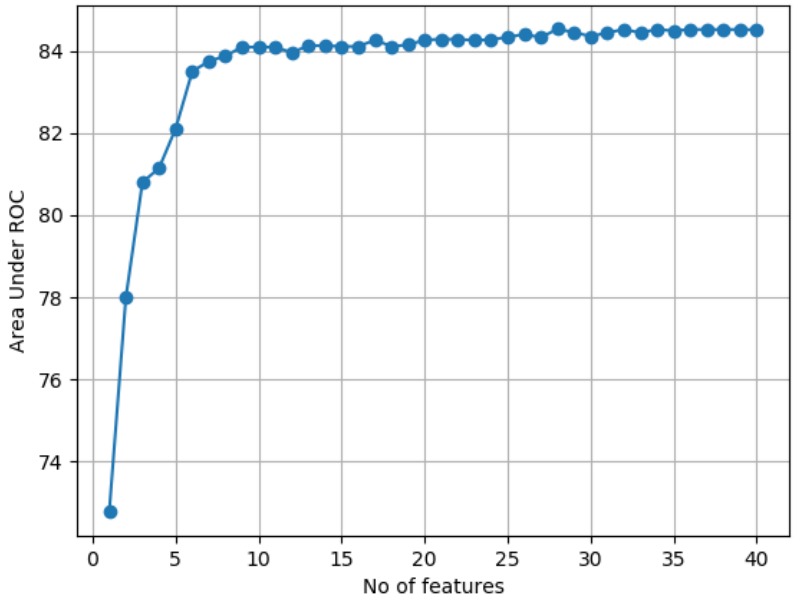
$$\mathrm{I}(X;Y) = \int_{\mathcal{Y}} \int_{\mathcal{X}} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right) dx\,dy,$$

where $p(x,y)$ is now the joint probability *density* function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y respectively.

```
from sklearn.feature_selection import mutual_info_classif
test = mutual_info_classif(X, y)

for i in range(len(X.columns)):
    print("%s Mutual Information Value: %.3f" %(X.columns[i], test[i]))
```

```
gender Mutual Information Value: 0.005
SeniorCitizen Mutual Information Value: 0.010
Partner Mutual Information Value: 0.021
Dependents Mutual Information Value: 0.015
tenure Mutual Information Value: 0.071
PhoneService Mutual Information Value: 0.000
PaperlessBilling Mutual Information Value: 0.022
MonthlyCharges Mutual Information Value: 0.046
TotalCharges Mutual Information Value: 0.043
MultipleLines_No Mutual Information Value: 0.000
MultipleLines_No phone service Mutual Information Value: 0.002
MultipleLines_Yes Mutual Information Value: 0.000
InternetService_DSL Mutual Information Value: 0.008
InternetService_Fiber optic Mutual Information Value: 0.046
InternetService_No Mutual Information Value: 0.029
OnlineSecurity_No Mutual Information Value: 0.069
OnlineSecurity_No internet service Mutual Information Value: 0.031
OnlineSecurity_Yes Mutual Information Value: 0.016
OnlineBackup_No Mutual Information Value: 0.039
OnlineBackup_No internet service Mutual Information Value: 0.027
OnlineBackup_Yes Mutual Information Value: 0.004
DeviceProtection_No Mutual Information Value: 0.037
DeviceProtection_No internet service Mutual Information Value: 0.034
DeviceProtection_Yes Mutual Information Value: 0.000
TechSupport_No Mutual Information Value: 0.053
TechSupport_No internet service Mutual Information Value: 0.034
TechSupport_Yes Mutual Information Value: 0.016
StreamingTV_No Mutual Information Value: 0.004
StreamingTV_No internet service Mutual Information Value: 0.028
StreamingTV_Yes Mutual Information Value: 0.005
StreamingMovies_No Mutual Information Value: 0.002
StreamingMovies_No internet service Mutual Information Value: 0.026
StreamingMovies_Yes Mutual Information Value: 0.003
Contract_Month-to-month Mutual Information Value: 0.084
Contract_One year Mutual Information Value: 0.014
Contract_Two year Mutual Information Value: 0.064
PaymentMethod_Bank transfer (automatic) Mutual Information Value: 0.009
PaymentMethod_Credit card (automatic) Mutual Information Value: 0.015
PaymentMethod_Electronic check Mutual Information Value: 0.038
PaymentMethod_Mailed check Mutual Information Value: 0.008
```



Features Selection using the Mutual Information

Best Features according to Mutual Information are:

Tenure	TotalCharges	InternetService_Fiber Optic	OnlineSecurity_No
TechSupport_No	Contract_monto-to-month	Contract_Two year	

6.4.3 Chi2 Test

$$\chi^2 = \frac{1}{d}\sum_{k=1}^{n}\frac{(O_k - E_k)^2}{E_k}$$

Where d is the number of samples,

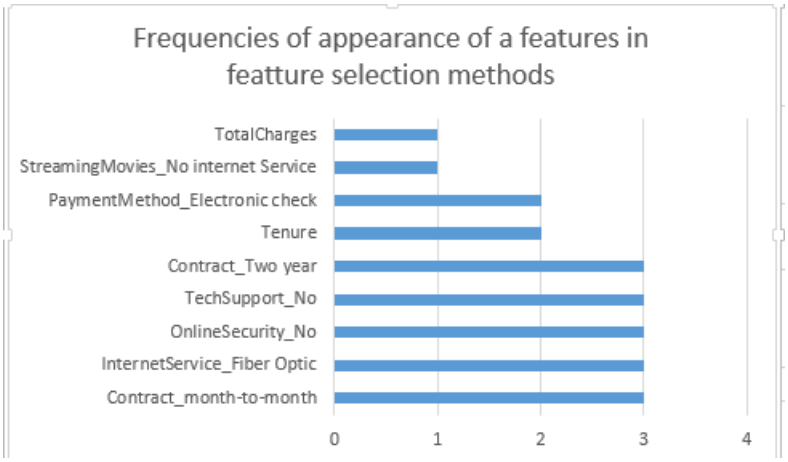
Ok , Ek are the observed value and Expected values for kth sample

Best Features according to Chi2 are:

StreamingMovies_No internet service	PaymentMethod_Electronic check	InternetService_Fiber Optic	OnlineSecurity_No
TechSupport_No	Contract_month-to-month	Contract_Two year	

6.4.4 Common Features Selected by various feature selection methods

Features Common to ANOVA-F-Value, Mutual Information, and Chi2 are:



6.5 DIMENSIONALITY REDUCTION

Dimensionality reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables.

6.5.1 Principal Component Analysis

Principal component analysis performs a linear mapping from the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.

Step 1: Standardize the data (Z)

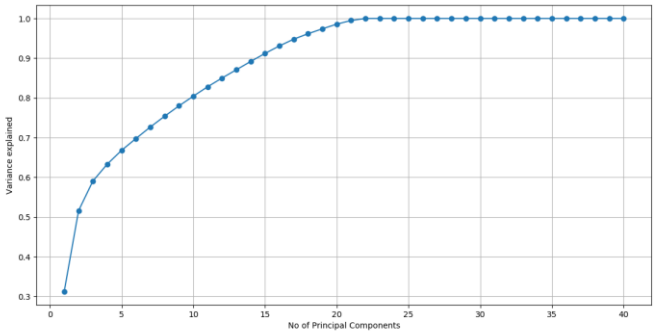
Step 2: Calculate the Covariance matrix for the data (ZTZ).

Step 3: Calculate Eigen values and Eigen vectors (E) for the covariance matrix.

Step 4: Sort the Eigen vector based on their corresponding Eigen values.

Step 5: Get the Principal Components (P) by transforming the original data(X) with the Eigen vectors (E).

$$P = XE$$
$$P = [P1\ P2\ \ldots\ldots\ Pn]$$

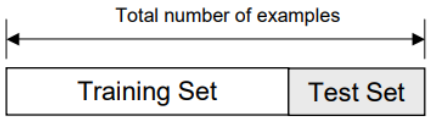


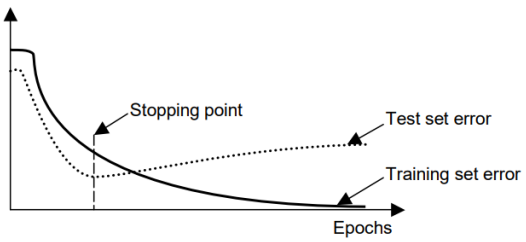
Number of PC's Vs Variance explained by them

6.6 CROSS VALIDATION

6.6.1 Hold Out Cross Validation

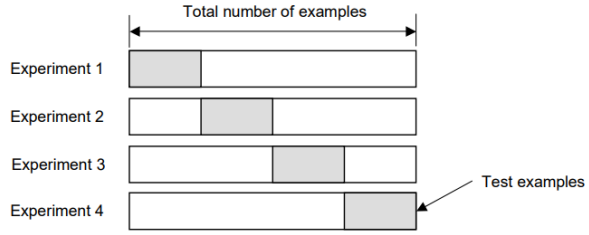
- Split the dataset into two groups.
    1. Training set: used to train the classifier
    2. Test set: used to estimate the error rate of the trained classifier





6.6.2 K Fold Cross Validation

- Create a K-fold partition of the dataset.
    1. For each of the K experiments, use K-1 folds for the training and the remaining one for testing.



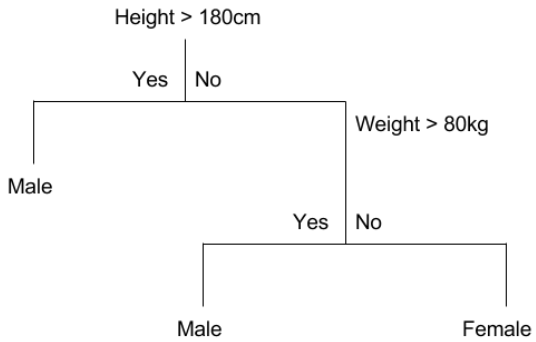
    2. The advantage of using K-Fold Cross Validation is that all the examples in the dataset are used for both training and testing.
    3. Error is estimated using the mean of all k estimators' errors.

7 Modeling

In modeling phase, we will build a model from the data. First we will divide the data into training and testing sets, and then train the model using training set, then test the model performance using the testing set.

7.1 CLASSIFICATION AND REGRESSION TREES

Classification And Regression Tree (CART) are a non-parametric supervised learning method used for both Classification and Regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.



Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items. Algorithms use different metrics for

measuring "best". These generally measure the homogeneity of the target class variable within the subsets.

Gini Impurity:

Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

To compute Gini impurity for a set of items with J classes, suppose $i \in \{1, 2, \ldots, J\}$, and let p_i be the fraction of items labeled with class i in the set.

$$\mathrm{I}_G(p) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = \sum_{i=1}^{J} p_i(1 - p_i) = \sum_{i=1}^{J} (p_i - {p_i}^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} {p_i}^2 = 1 - \sum_{i=1}^{J} {p_i}^2$$

Information Gain:

Entropy is defined as below

$$\mathrm{H}(T) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

where $p_1, p_2, \ldots$ are fractions that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree.[15]

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|a)}^{\text{Weighted Sum of Entropy (Children)}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{a} p(a) \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

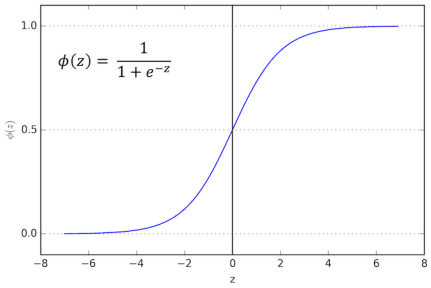
References: Wiki, machinelearningmastery.com

7.2 LOGISTIC REGRESSION

Logistic regression is named for the function used at the core of the method, the logistic function.

The logistic function, also called the sigmoid function is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$



Logistic regression uses an equation as the representation, very much like linear regression.

Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

Logistic regression equation:

$$y = \frac{e^{\beta 0+\sum_{i=1}^{n} \beta i*xi}}{1 + e^{\beta 0+\sum_{i=1}^{n} \beta i*xi}}$$

7.3 NAÏVE BAYES

Bayes' Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.

P(h|d) = (P(d|h) * P(h)) / P(d)

Where

- P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.
- P(d|h) is the probability of data d given that the hypothesis h was true.
- P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
- P(d) I s the probability of the data (regardless of the hypothesis).

- After calculating the posterior probability for a number of different hypotheses, you can select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the maximum a posteriori (MAP) hypothesis.

$$MAP(h) = max(P(h|d))$$

References: machinelearningmastery.com

7.4 SUPPORT VECTOR MACHINES

The numeric input variables (x) in your data (the columns) form an n-dimensional space. For example, if you had two input variables, this would form a two-dimensional space.

A hyper plane is a line that splits the input variable space. In SVM, a hyper plane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions you can visualize this as a line and let's assume that all of our input points can be completely separated by this line. For example:

$$B0 + (B1 * X1) + (B2 * X2) = 0$$

Where the coefficients (B1 and B2) that determine the slope of the line and the intercept (B0) are found by the learning algorithm, and X1 and X2 are the two input variables.

You can make classifications using this line. By plugging in input values into the line equation, you can calculate whether a new point is above or below the line.

- Above the line, the equation returns a value greater than 0 and the point belongs to the first class (class 0).
- Below the line, the equation returns a value less than 0 and the point belongs to the second class (class 1).
- A value close to the line returns a value close to zero and the point may be difficult to classify.
- If the magnitude of the value is large, the model may have more confidence in the prediction.

The distance between the line and the closest data points is referred to as the margin. The best or optimal line that can separate the two classes is the line that as the largest margin. This is called the Maximal-Margin hyper plane.

The margin is calculated as the perpendicular distance from the line to only the closest points. Only these points are relevant in defining the line and in the construction of the classifier. These points are called the support vectors. They support or define the hyper plane.

The hyper plane is learned from training data using an optimization procedure that maximizes the margin.

Kernels can be used that transform the input space into higher dimensions such as a Polynomial Kernel and a Radial Kernel. This is called the Kernel Trick.

The *kernel function* can be any of the following:

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma \langle x, x' \rangle + r)^d$. d is specified by keyword `degree`, r by `coef0`.
- rbf: $\exp(-\gamma \|x - x'\|^2)$. γ is specified by keyword `gamma`, must be greater than 0.
- sigmoid $(\tanh(\gamma \langle x, x' \rangle + r))$, where r is specified by `coef0`.

7.5 K NEAREST NEIGHBORS (KNN)

The model representation for KNN is the entire training dataset. KNN has no model other than storing the entire dataset, so there is no learning required.

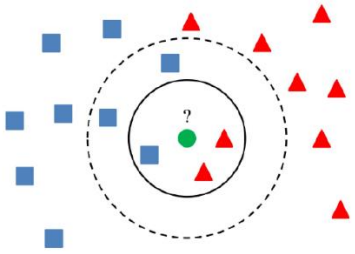
Because the entire training dataset is stored, you may want to think carefully about the consistency of your training data. It might be a good idea to curate it, update it often as new data becomes available and remove erroneous and outlier data.

KNN makes predictions using the training dataset directly.

Predictions are made for a new instance (x) by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances.

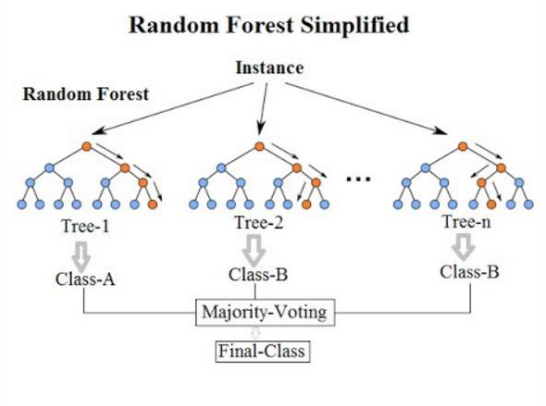
The Target variable can be calculated as the class with the highest frequency from the K-most similar instances. Each instance in essence votes for their class and the class with the most votes is taken as the prediction.

If you are using K and you have an even number of classes (e.g. 2) it is a good idea to choose a K value with an odd number to avoid a tie.



7.6 RAMDOM FOREST

Random forest is like bootstrapping algorithm with Decision tree (CART) model. Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART model with different sample and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation.



7.7 MULTI-LAYER FEED FORWARD NEURAL NETWORK

Multi-layer feed forward neural network work in two phases.

1. First phase calculates the Activation values of the each layer from the previous layer till the last layer.
   - ✓ For activation function we can use Sigmoid, Tan h or relu etc.

2. Second phase propagates the error (prediction in 1st phase and actual value) till the input layer. Here propagating the error means updating the weights with respect to the error generated in the 1st layer using the chain rule.

   - ✓ The weights are updated using the following formula

$$\Delta w_{ji}(t+1) = -\eta \frac{\partial E}{\partial w_{ji}(t)} + \alpha \Delta w_{ji}(t)$$

Where η = momentum, and α is learning rate

8 Evaluation Metrics

The goal of the ML model is to learn patterns that generalize well for unseen data instead of just memorizing the data that it was shown during training. Once you have a model, it is important to check if your model is performing well on unseen examples that you have not used for training the model. To do this, you use the model to predict the labels on the evaluation dataset (held out data) and then compare the predicted target to the actual label (ground truth).

8.1 ACCURACY

Accuracy is the fraction of predictions our model got right.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

8.2 PRECISION

Precision measures what proportion of positive identifications was actually correct.

Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

8.3 RECALL

Recall measures what proportion of actually positives was identified correctly.

$$\text{Recall} = \frac{TP}{TP + FN}$$

8.4 AREA UNDER ROC CURVE (AUC)

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

True Positive Rate

False Positive Rate

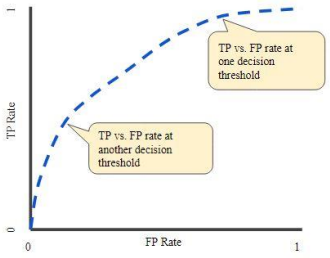
True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$TPR = TP / (TP+FN)$$

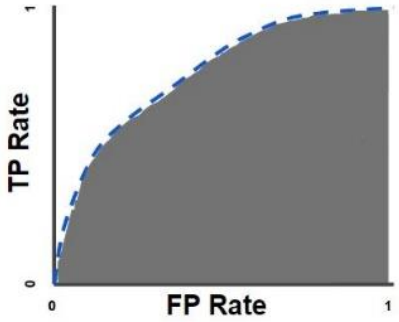
False Positive Rate (FPR) is defined as follows:

$$FPR = FP / (FP+TN)$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.



AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0, 0) to (1, 1).



9 Results

9.1 CLASSIFICATION AND REGRESSION TREES (CART)

9.1.1 CART (Hold Out (70 – 30) + without Feature selection)

Criterion	Max depth	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
Gini	5	78.72	61.93	49.54	83.25
Gini	6	78.38	60.59	50.99	82.73
Gini	7	77.72	59.13	49.54	81.81
Gini	8	77.63	59.00	49.00	80.43
Gini	9	77.15	58.05	47.38	80.21
Gini	10	76.77	56.72	49.36	79.94
Gini	11	76.58	57.01	44.68	79.97
Gini	12	76.68	56.89	46.84	79.98
Gini	13	76.49	56.45	46.48	79.82
Info. gain	5	78.90	62.55	49.36	**83.33**
Info. gain	6	78.72	61.99	49.36	82.06
Info. gain	7	78.24	60.95	48.10	81.73
Info. gain	8	78.24	61.11	47.56	81.01
Info. gain	9	77.72	59.34	48.64	80.29
Info. gain	10	77.20	57.93	48.64	79.16
Info. gain	11	77.10	58.91	42.88	78.96
Info. gain	12	76.68	56.92	46.66	78.26
Info. gain	13	76.72	57.11	46.30	78.08

9.1.2 CART (Hold Out (70 – 30) + Feature selection)

Criterion	Max depth	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
Gini	5	79.91	61.97	61.08	**83.63**
Gini	6	80	63.71	55.68	83.38
Gini	7	80.05	63.56	56.58	82.11
Gini	8	79.62	63.21	53.87	81.74
Gini	9	78.96	61.99	51.71	81.12

Gini	10	78.67	61.96	49.01	80.94
Gini	11	78.63	61.87	48.83	80.91
Gini	12	78.63	61.87	48.83	80.83
Gini	13	78.63	61.87	48.83	80.91
Info. gain	5	79.62	62.58	56.04	**83.24**
Info. gain	6	79.48	62.35	55.5	82.9
Info. gain	7	79.67	62.8	55.68	80.99
Info. gain	8	79.43	62.79	53.51	81.06
Info. gain	9	78.77	61.66	50.99	80.53
Info. gain	10	78.48	61.61	48.29	80.22
Info. gain	11	78.44	61.47	48.29	80.12
Info. gain	12	78.44	61.47	48.29	80.08
Info. gain	13	78.44	61.47	48.29	80.03

9.1.3 CART (Hold Out (70 – 30) + PCA (no components = 5))

Criterion	Max depth	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
Gini	5	79.95	65.07	49.54	**82.74**
Gini	6	79.91	64.85	49.73	82.59
Gini	7	79.19	61.75	52.64	82.2
Gini	8	78.44	59.55	53.37	81.36
Gini	9	78.06	59.03	51.18	80.68
Gini	10	76.97	57.55	43.72	79.86
Gini	11	77.82	58.32	51.73	79.88
Gini	12	77.49	58.37	46.99	79.91
Gini	13	77.3	57.38	49.54	79.44
Gini	14	77.25	57.62	47.54	79.32
Info. gain	5	80.14	64.32	53.19	**83.75**
Info. gain	6	79.86	62.97	54.83	83.07
Info. gain	7	79.91	63.5	53.55	82.81
Info. gain	8	79.86	63.08	54.46	82.13

Info. gain	9	79	60.27	56.65	81.13
Info. gain	10	77.16	57.49	46.81	80
Info. gain	11	77.3	57.14	51	79.87
Info. gain	12	77.16	56.77	51.18	79.13
Info. gain	13	77.3	57.14	51	79.17
Info. gain	14	77.25	57.14	50.27	79.08

9.1.4 CART (Stratified 10-Fold CV + without Feature selection)

Criterion	Max depth	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
Gini	5	79.33	61.92	58.15	**82.87**
Gini	6	79.15	62.62	53.60	82.74
Gini	7	79.10	63.46	50.83	82.43
Gini	8	78.47	61.95	49.11	81.67
Gini	9	78.55	62.13	49.33	81.43
Gini	10	78.48	62.01	48.58	81.46
Gini	11	78.28	61.58	48.63	81.30
Gini	12	78.19	61.44	48.79	81.22
Gini	13	78.20	61.46	48.47	81.27
Gini	14	78.24	61.41	48.26	81.17
Info. gain	5	79.33	62.18	57.30	**82.87**
Info. gain	6	79.10	62.67	52.96	82.76
Info. gain	7	79.23	63.60	51.52	82.36
Info. gain	8	78.86	62.87	49.97	81.65
Info. gain	9	78.83	63.02	49.43	81.56
Info. gain	10	78.62	62.81	48.42	81.35
Info. gain	11	78.55	62.77	47.72	81.12
Info. gain	12	78.48	62.71	47.29	81.16
Info. gain	13	78.34	62.31	47.19	80.84
Info. gain	14	78.32	62.22	46.87	80.76

9.1.5 CART (Stratified 10-Fold CV + Feature selection)

Criterion	Max depth	Accuracy(%)	Precision(%)	Recall(%)	AUC(%)
Gini	5	79.52	62.24	58.59	83.51
Gini	6	79.54	63.23	55.27	**83.70**
Gini	7	79.44	63.73	52.92	83.46
Gini	8	79.05	63.03	51.69	82.98
Gini	9	79.00	63.09	50.73	82.89
Gini	10	79.01	63.12	50.78	82.86
Gini	11	78.94	62.80	50.99	82.84
Gini	12	78.94	62.80	50.99	82.84
Gini	13	78.95	62.82	51.05	82.83
Gini	14	78.95	62.82	51.05	82.83
Info. gain	5	79.55	62.35	58.43	83.50
Info. gain	6	79.56	63.30	55.32	**83.57**
Info. gain	7	79.62	63.85	53.51	83.13
Info. gain	8	79.37	63.81	52.12	82.83
Info. gain	9	79.00	63.38	49.98	82.63
Info. gain	10	79.07	63.62	49.87	82.64
Info. gain	11	79.04	63.46	50.03	82.62
Info. gain	12	79.00	63.29	50.08	82.63
Info. gain	13	79.01	63.31	50.14	82.59
Info. gain	14	79.01	63.31	50.14	82.58

9.1.6 CART (Stratified 10-Fold CV + PCA (no components = 5))

Criterion	Max depth	Accuracy(%)	Precision(%)	Recall(%)	AUC(%)
Gini	2	76.47	55.61	57.57	77.16
Gini	3	77.31	61.91	37.98	79.74
Gini	4	77.09	57.22	54.73	**80.38**
Gini	5	77.17	59.69	44.19	80.37
Gini	6	76.94	58.66	45.26	79.91
Gini	7	77.03	58.44	47.35	79.60

Gini	8	76.82	57.16	49.60	79.42
Gini	9	76.59	57.06	47.56	79.01
Gini	10	76.54	56.72	48.63	78.87
Gini	11	76.32	56.61	47.62	78.30
Info. gain	2	75.09	53.06	58.85	76.64
Info. gain	3	77.50	61.68	41.89	79.75
Info. gain	4	77.27	59.75	46.59	80.29
Info. gain	5	77.40	60.63	43.66	**80.75**
Info. gain	6	77.28	59.43	46.01	79.99
Info. gain	7	77.00	58.32	49.00	79.28
Info. gain	8	77.13	58.81	46.65	79.04
Info. gain	9	76.77	57.58	49.64	78.57
Info. gain	10	76.43	56.56	50.29	77.96
Info. gain	11	76.02	55.97	50.02	77.57

9.2 K NEAREST NEIGHBORS

9.2.1 K Nearest Neighbors (Hold Out (70 – 30) + without feature selection)

# Neighbors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	77.58	58.4	51.35	79.15
10	77.35	60.43	40.18	81.67
15	78.58	61.17	50.81	82.54
20	78.58	62.91	45.23	82.99
25	79.34	62.96	52.07	83.22
30	79.62	64.78	49.37	83.23
35	79.57	63.6	52.25	83.37
40	79.1	63.38	48.65	83.53
45	79.43	63.66	50.81	83.73
50	79.62	63.98	51.53	**83.75**

9.2.2 K Nearest Neighbors (Hold Out (70 - 30) + Feature Selection)

# Neighbors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	77.58	58.4	51.35	79.15
10	77.35	60.43	40.18	81.67
15	78.58	61.17	50.81	82.54
20	78.58	62.91	45.23	82.99
25	79.34	62.96	52.07	83.22
30	79.62	64.78	49.37	83.23
35	79.57	63.6	52.25	83.37
40	79.1	63.38	48.65	83.53
50	79.43	63.66	50.81	**83.73**

9.2.3 K Nearest Neighbors (Hold out (70 – 30) + PCA (no components = 5))

# Neighbors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	78.2	59.53	50.64	78.75
10	79.72	66.95	43.53	81.96
15	80.24	65.71	50.27	83.21
20	80.47	67.52	48.09	83.63
25	80.43	65.81	51.55	83.98
30	80.66	68.03	48.45	84.08
35	81.04	67.78	51.73	84.14
40	80.85	68.17	49.54	84.15
50	81.04	67.45	52.46	**84.22**

9.2.4 K Nearest Neighbors (Stratified 10-Fold CV + without feature selection)

# Neighbors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	75.27	53.74	50.24	76.65
10	77.44	59.59	47.19	79.88
15	77.60	58.18	56.23	81.02
20	77.88	59.38	53.24	81.45
25	78.02	58.85	57.68	81.94
30	78.32	60.02	55.38	**82.10**

35	78.59	59.88	59.02	82.34
40	79.11	61.44	57.57	82.46
50	78.88	60.40	59.71	82.51
55	79.09	61.22	58.16	82.54
60	79.01	60.71	59.60	**82.57**

9.2.5 K Nearest Neighbors (Stratified 10-Fold CV + Feature Selection)

# Neighbors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	75.27	53.75	50.24	76.65
10	77.45	59.59	47.19	79.88
15	77.60	58.19	56.23	81.02
20	77.89	59.39	53.24	81.45
25	78.03	58.86	57.68	81.94
30	78.33	60.02	55.38	**82.10**
35	78.60	59.89	59.02	82.34
40	79.11	61.44	57.57	82.46
50	79.10	61.22	58.16	82.54
55	79.01	60.71	59.60	**82.57**
60	79.10	61.18	58.37	82.58
65	78.85	60.34	59.66	82.54
99	78.64	60.17	58.16	82.65

9.2.6 K Nearest Neighbors (Stratified 10-Fold CV + PCA (no components = 5))

# Neighbors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	78.19	60.63	50.82	78.84
10	78.98	65.43	44.46	81.08
15	79.48	64.51	50.83	82.43
20	79.16	65.12	46.60	83.09
25	79.66	65.03	51.04	83.30
30	79.60	65.61	48.90	83.55
35	79.96	65.50	52.27	83.78
40	80.19	66.79	50.93	**83.93**

50	80.09	65.84	52.38	83.71
55	79.67	65.64	49.86	83.85
60	79.72	65.09	51.41	**83.93**
65	79.86	65.57	51.09	83.89

9.3 LOGISTIC REGRESSION

9.3.1 Logistic Regression: (Hold Out (70 – 30) + without feature selection)

Model	Accuracy	Precision	Recall	AUC
Logistic Regression	80.00%	63.77%	52.97%	83.75%

9.3.2 Logistic Regression: (Hold Out (70 – 30) + Feature Selection)

Model	Accuracy	Precision	Recall	AUC
Logistic Regression	80.00%	63.77%	52.97%	83.77%

9.3.3 Logistic Regression (Hold Out (70 – 30) + PCA):

#Components	Variance(%)	Accuracy(%)	Precision(%)	Recall(%)	AUC(%)
2	52.00	81.00	66.11	51.29	83.50
3	59.00	81.00	65.45	55.86	85.16
4	63.00	82.00	65.34	55.95	**84.83**
5	66.00	81.00	65.67	56.10	84.36
6	70.00	80.00	65.84	52.23	84.23
7	72.00	81.00	69.12	51.05	**85.45**

9.3.4 Logistic Regression: (Stratified 10-Fold CV + without feature selection)

Model	Accuracy	Precision	Recall	AUC
Logistic Regression	80.00%	65.00%	55.00%	84.00%

9.3.5 Logistic Regression: (Stratified 10-Fold CV + Feature Selection)

Model	Accuracy	Precision	Recall	AUC

Logistic Regression	79.00%	64.00%	53.00%	84.00%

9.3.6 Logistic Regression (Stratified 10 fold CV + PCA):

#Components	Variance(%)	Accuracy(%)	Precision(%)	Recall(%)	AUC(%)
2	52.00	79.21	63.91	50.56	81.73
3	59.00	79.58	64.81	50.83	83.72
4	63.00	79.58	64.75	50.99	83.69
5	66.00	80.03	65.55	52.59	**83.92**
6	70.00	79.95	65.17	52.75	83.94
7	72.00	79.91	65.17	52.43	83.92

9.4 NAÏVE BAYES

9 4.1 Naive Bayes (Hold Out (70 - 30) + Without Feature Selection)

Priors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
[0.5, 0.5]	69.14	45.42	85.94	81.70
Using original priors of the dataset	70.05	46.25	85.58	81.70

9.4.2 Naive Bayes (Hold Out (70 - 30) + Feature Selection)

Priors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
[0.5, 0.5]	71.42	47.55	83.78	83.12
Using original priors of the dataset	73.18	49.39	80.18	83.12

9.4.3 Naive Bayes (Hold Out (70 - 30) + PCA (No of components = 5))

Priors	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
[0.5, 0.5]	71.71	47.44	80.87	82.91
Using original priors of the dataset	79.29	59.4	64.48	82.91

9.4.4 Naive Bayes (Stratified 10-Fold CV + Without Feature Selection)

Priors	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
[0.5, 0.5]	68.10	44.77	85.23	81.88
Using original priors of the dataset	69.14	45.69	84.59	81.88

9.4.5 Naive Bayes (Stratified 10-Fold CV + Feature Selection)

Priors	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
[0.5, 0.5]	68	45	86	82
Using original priors of the dataset	69	45	85	82

9.4.6 Naive Bayes (Stratified 10-Fold CV + PCA (No of components = 5))

Priors	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
[0.5, 0.5]	78.45	48.81	80.41	82.57
Using original priors of the dataset	78.45	59.22	60.88	82.57

9.5 SUPPORT VECTOR MACHINES

9.5.1 SVM (Hold Out (70 – 30) + without feature selection)

Kernel	C Parameter	Accuracy(%)	Precision(%)	Recall(%)	AUC(%)
Linear	0.1	78.96	63	49	83.2
Linear	1	79	63	49	83.2
Linear	10	79	63	49	83.2
RBF	0.1	79.81	65	50	**83.36**
RBF	1	79.76	64	53	83.32
RBF	10	79.86	64	53	83.31
RBF	100	78.86	63	46	83.31
Sigmoid	0.1	78.58	61	50	83.15
Sigmoid	1	78.1	60	52	83.07
Sigmoid	10	75.02	53	50	80

Sigmoid	100	72.75	48	43	76.74

9.5.2 SVM (Hold Out (70 – 30) + Feature selection)

Kernel	C parameter	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
Linear	0.1	78.96	63.00	49.00	83.2
Linear	1	79	63.00	49.00	83.2
Linear	10	79	63.00	49.00	83.2
Linear	100	79	63.00	49.00	83.2
RBF	0.1	79.81	65.00	50.00	**83.36**
RBF	1	79.76	64.00	53.00	83.32
RBF	10	79.86	64.00	53.00	83.31
RBF	100	78.86	63.00	46.00	83.31
Sigmoid	1	79.76	64.00	53.00	83.31
Sigmoid	10	79.76	64.00	53.00	83.31
Sigmoid	100	79.76	64.00	53.00	83.3

9.5.3 SVM (Hold Out (70 – 30) + PCA (n_components = 5))

Kernel	C parameter	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
Linear	0.1	78.96	63.00	49.00	83.2
Linear	1	79	63.00	49.00	83.2
Linear	10	79	63.00	49.00	83.2
Linear	100	79	63.00	49.00	83.2
RBF	0.1	80.81	66.00	54.00	84.73
RBF	1	80.85	65.00	56.00	**84.65**
RBF	10	81	66.00	56.00	84.64
RBF	100	80.66	65.00	56.00	84.61
Sigmoid	0.1	79.81	65.00	50.00	83.36
Sigmoid	1	79.76	64.00	53.00	83.32
Sigmoid	10	79.86	64.00	53.00	83.31
Sigmoid	100	78.86	63.00	49.00	83.31

9.5.4 SVM (Stratified 10-Fold CV + without feature selection)

Kernel	C parameter	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
Linear	0.1	80.08	65.14	53.72	**83.52**
Linear	1	80.03	65.02	53.72	83.43
Linear	10	79.99	64.86	53.77	83.37
RBF	0.1	79.45	66.25	46.38	83.05
RBF	1	79.93	66.56	49.27	82.87
RBF	10	79.92	66.22	49.91	80.72
RBF	100	79.86	65.12	52.11	80.01
Sigmoid	0.1	77.23	74.09	22.70	83.26
Sigmoid	1	79.79	63.97	54.73	**83.59**
Sigmoid	10	75.47	54.31	48.42	75.30
Sigmoid	100	73.82	50.79	50.18	74.61

9.5.5 SVM (Stratified 10-Fold CV + Feature selection)

Kernel	C parameter	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
Linear	0.1	76.62	55.84	60.19	81.03
Linear	1	76.45	55.37	61.15	81.01
Linear	10	76.45	55.37	61.15	**81.23**
Linear	100	76.45	55.37	61.15	81.01
RBF	0.1	77.46	61.78	40.02	77.84
RBF	1	77.77	65.17	36.11	75.67
RBF	10	77.61	60.36	46.6	73.02
RBF	100	77.74	60.23	48.20	72.71
Sigmoid	1	69.09	41.14	38.73	74.01
Sigmoid	10	69.08	40.89	36.54	73.44
Sigmoid	100	70.02	43.02	39.11	73.43

9.5.6 SVM (Stratified 10-Fold CV + PCA)

Kernel	C parameter	Accuracy(%)	Precision(%)	Recall(%)	AUC (%)
Linear	0.1	78.80	62.30	51.26	**82.21**
Linear	1	78.80	62.34	51.15	82.20

Linear	10	78.80	62.34	51.15	82.20
Linear	100	78.81	62.38	51.15	82.20
RBF	0.1	78.59	65.16	41.84	77.71
RBF	1	78.65	64.85	42.91	77.22
RBF	10	78.75	63.32	47.56	76.73
RBF	100	78.92	63.98	47.35	76.24
Sigmoid	0.1	76.86	57.28	51.15	79.47
Sigmoid	1	72.0	47.27	45.74	72.89
Sigmoid	10	71.53	46.4	46.33	67.88
Sigmoid	100	71.01	45.44	45.31	67.86

9.6 RAMDOM FOREST

9.6.1 Random Forest (Hold Out (70 – 30) + Without Feature Selection)

# Trees	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	78.53	63.42	43.42	83.88
10	78.81	65.08	41.98	84.23
20	79.71	68.09	43.06	**84.52**
30	79.43	66.04	44.86	84.47
40	79.47	66.05	45.22	84.11
50	79.52	65.41	47.02	**84.65**

9.6.2 Random Forest (Hold Out (70 – 30) + Feature Selection)

# Trees	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	77.25	60.21	39.81	81.72
10	76.96	60.48	35.85	81.71
20	77.20	62.93	32.43	81.91
30	77.20	62.93	32.43	81.98
40	77.20	62.93	32.43	**82.06**
50	77.20	62.93	32.43	82.10

9.6.3 Random Forest (Hold Out (70 – 30) + PCA (n_components = 5))

# Trees	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	79.71	65.87	45.71	82.06
10	79.28	64.00	46.63	82.19
20	79.43	65.41	44.44	**82.42**
30	79.57	65.44	45.53	82.07
40	79.47	64.42	47.17	82.26
50	79.28	63.93	46.81	82.40

9.6.4 Random Forest (Stratified 10-Fold CV + Without Feature Selection)

# Trees	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	79.72	66.27	45.42	84.04
10	79.02	67.60	45.31	84.13
20	79.59	68.37	44.40	**84.28**
30	79.63	67.69	44.46	84.26
40	79.62	68.11	44.03	84.23
50	79.67	68.27	44.14	84.38

9.6.5 Random Forest (Stratified 10-Fold CV + Feature Selection)

# Trees	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	77.87	61.50	42.54	81.93
10	77.74	62.02	46.81	81.89
20	77.55	63.66	45.00	**82.07**
30	77.98	63.09	44.89	82.04
40	77.84	62.06	41.89	81.98
50	77.70	61.50	46.06	82.07

9.6.6 Random Forest (Stratified 10-Fold CV + PCA (n_components = 5))

# Trees	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
5	78.21	62.73	43.17	81.44

10	78.58	62.53	44.67	81.83
20	78.37	62.80	44.30	82.24
30	78.55	63.58	44.83	**82.29**
40	78.55	63.47	45.10	82.28
50	78.52	63.89	44.62	82.23

9.7 MULTI LAYER FEED FORWARD NEURAL NETWORK (MLP)

9.7.1 MLP (Hold out (70 – 30) + without feature selection)

# Hidden nodes	Alpha	Momentum (%)	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
7	1.00E-04	0.3	79.95	65.56	50.09	**84.39**
10	1.00E-04	0.6	79.28	63.11	51.17	84.31
12	1.00E-03	0.5	79.95	64.10	54.05	84.28
2	9.10E-03	0.9	79.62	64.04	51.35	84.28
11	1.20E-02	0.9	79.62	63.04	54.41	84.28
12	1.00E-03	0.2	79.52	63.69	51.53	84.27
4	1.00E-04	0.2	79.66	64	51.89	84.27

9.7.2 MLP (Hold out (70 – 30) + Feature selection)

# Hidden nodes	Alpha	Momentum	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
11	1.00E-04	0.9	79.4	63.37	52.07	**84.38**
2	1.00E-04	0.6	79.81	65.46	49.18	84.34
10	0.0001	0.9	79.38	63.45	50.99	84.26
6	1.00E-06	0.8	79.47	63.55	51.53	84.25
9	1.00E-06	0.4	79.24	63.63	49.18	84.25
6	9.10E-03	0.4	79.43	63.78	50.45	84.25
8	1.10E-03	0.1	79.38	63.63	50.45	84.21

9.7.3 MLP (Hold out (70 – 30) + PCA (n_components = 5))

# Hidden nodes	Alpha	Momentum	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
9	1.00E-04	0.1	80.75	66.06	53.55	**84.97**
9	1.00E-06	0.1	80.85	66.29	53.73	84.96
9	1.00E-06	0.3	80.85	66.29	53.73	84.96
9	1.00E-06	0.4	80.85	66.29	53.73	84.96
9	1.00E-06	0.5	80.85	66.29	53.73	84.96
9	1.00E-06	0.6	80.85	66.29	53.73	84.96
9	0.0098	0.1	80.61	65.55	53.73	84.90

9.7.4 MLP (Stratified 10-fold CV + without feature selection)

# Hidden nodes	Alpha	Momentum (%)	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
2	0.0001	0.1	80.47	67.01	52.59	**84.75**
3	0.0001	0.1	80.60	66.58	54.25	84.25
4	0.0001	0.1	79.95	65.58	52.06	84.07
4	0.01	0.9	79.76	65.17	51.58	84.24
6	0.001	0.5	79.58	64.74	51.20	83.89
7	0.001	0.5	79.59	64.23	52.60	83.6
9	0.0001	0.7	79.61	64.40	52.11	83.43

9.7.5 MLP (Stratified 10-fold CV + Feature selection)

# Hidden nodes	Alpha	Momentum	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
2	0.0001	0.1	79.27	63.50	51.90	83.90
2	0.001	0.5	79.48	63.92	52.65	83.87
3	0.01	0.5	79.46	63.95	52.17	83.70
4	0.01	0.1	79.69	64.72	52.06	84.02
6	0.0001	0.9	79.95	64.84	53.82	**84.05**
7	0.001	0.9	79.62	64.64	51.84	83.98
9	0.0001	0.1	79.76	65.26	51.31	83.75

9.7.6 MLP (Stratified 10-fold CV + PCA (n_components = 5))

# Hidden nodes	Alpha	Momentum	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
2	0.0001	0.5	80.19	65.67	53.50	84.31
3	0.0001	0.1	80.18	65.36	54.20	84.14
4	0.0001	0.1	80.23	65.94	53.18	84.42
6	0.0001	0.1	79.98	65.61	51.95	84.27
6	0.01	0.3	80.06	65.67	52.43	84.20
9	0.001	0.9	80.1	65.61	52.97	**84.51**
9	0.01	0.5	79.86	65.45	51.52	84.44

10 Comparison of models

10.1 HOLD OUT + WITHOUT FEATURE SELECTION

Model	Accuracy	Precision	Recall	AUC
CART	78.90	62.55	49.36	83.33
K Nearest Neighbors	79.62	63.98	51.53	83.75
Naïve Bayes	70.05	46.25	85.58	81.70
Logistic Regression	80.00	63.77	52.97	83.75
Support Vector Machines	79.81	65	50	83.36
Random Forest	79.52	65.41	47.02	84.65
Multi-layer Perceptron	79.95	65.56	50.09	**84.39**

10.2 HOLD OUT + FEATURE SELECTION

Model	Accuracy	Precision	Recall	AUC
CART	79.91	61.97	61.08	83.63
K Nearest Neighbors	79.43	63.66	50.81	**83.73**
Naïve Bayes	73.18	49.39	80.18	83.12
Logistic Regression	80.00	63.77	52.97	83.77

Support Vector Machines	79.81	65.00	50.00	83.36
Random Forest	77.20	62.93	32.43	82.06
Multi-layer Perceptron	79.4	63.37	52.07	**84.38**

10.3 HOLD OUT + PCA (N_COMPONENTS = 5)

Model	Accuracy	Precision	Recall	AUC
CART	79.95	65.07	49.54	82.74
K Nearest Neighbors	81.04	67.45	52.46	84.22
Naïve Bayes	79.29	59.4	64.48	82.91
Logistic Regression	82.00	65.34	55.95	**84.83**
Support Vector Machines	80.85	65.00	56.00	84.65
Random Forest	79.43	65.41	44.44	82.42
Multi-layer Perceptron	80.75	66.06	53.55	**84.97**

10.4 STRATIFIED 10-FOLD CV+ WITHOUT FEATURE SELECTION

Model	Accuracy	Precision	Recall	AUC
CART	79.33	61.92	58.15	82.87
K Nearest Neighbors	79.01	60.71	59.60	82.57
Naïve Bayes	69.14	45.69	84.59	81.88
Logistic Regression	80.00	65.00	55.00	84.00
Support Vector Machines	80.08	65.14	53.72	83.52
Random Forest	79.59	68.37	44.40	84.28
Multi-layer Perceptron	80.47	67.01	52.59	**84.75**

10.5 STRATIFIED 10-FOLD CV + FEATURE SELECTION

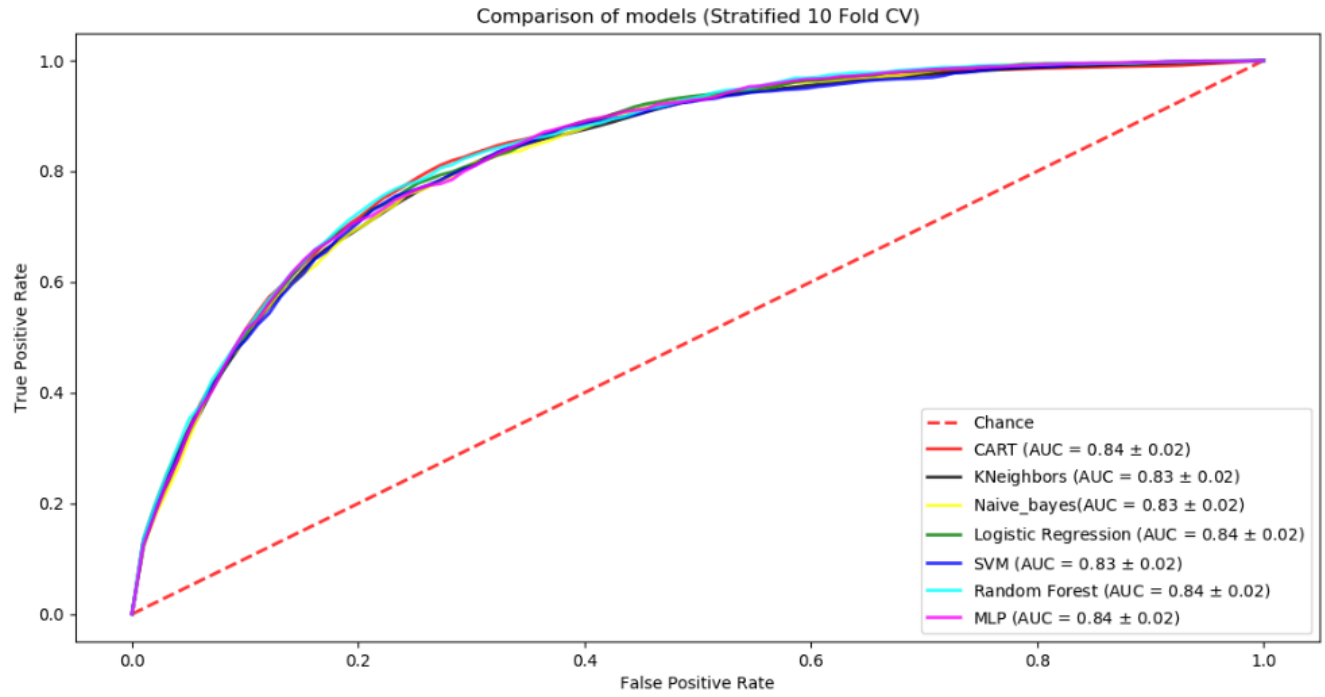
Model	Accuracy	Precision	Recall	AUC
CART	79.54	63.23	55.27	**83.70**
K Nearest Neighbors	78.33	60.02	55.38	82.10
Naïve Bayes	69	45	85	82
Logistic Regression	79.00	64.00	53.00	**84.00**
Support Vector Machines	76.45	55.37	61.15	81.23

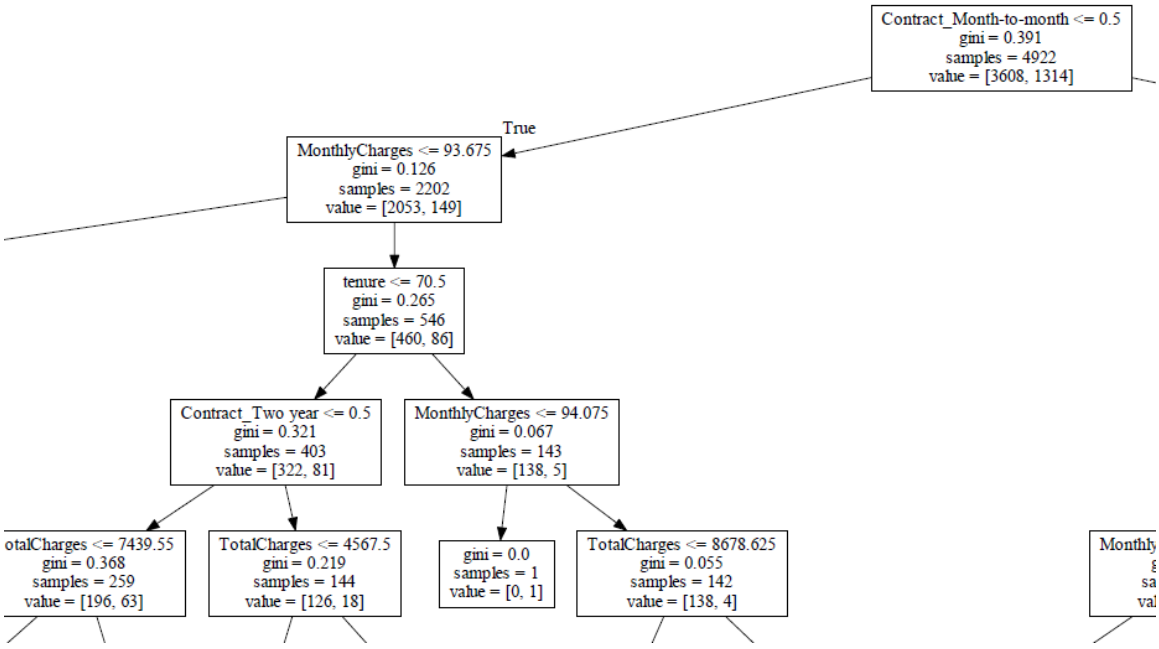
Random Forest	77.55	63.66	45.00	82.07
Multi-layer Perceptron	79.95	64.84	53.82	**84.05**

10.6 STRATIFIED 10-FOLD CV + PCA (N_COMPONENTS = 5)

Model	Accuracy	Precision	Recall	AUC
CART	77.09	57.22	54.73	80.38
K Nearest Neighbors	79.72	65.09	51.41	83.93
Naïve Bayes	78.45	59.22	60.88	82.57
Logistic Regression	80.03	65.55	52.59	**83.92**
Support Vector Machines	78.80	62.30	51.26	82.21
Random Forest	78.55	63.58	44.83	82.29
Multi-layer Perceptron	80.1	65.61	52.97	**84.51**

10.7 AREA UNDER ROC FOR ALL THE MODELS





Above figure is part of the tree generated by CART model

11. Conclusion

There is no significant difference between the performances of the classifiers with/without feature selection. So it's better to go for feature selection and principle component analysis is also not showing any improvement in the Performance (may be as the features are not linearly correlated). Using the Stratified 10-fold cross validation, Logistic Regression and Multi-layer feed forward neural network are giving the best performance (Area under ROC) among all the models but they lack human comprehensibility.

Area under ROC for Classification and Regression Trees (CART) is only 0.30% less than the best estimators and also CART will gives us a Sequence of decisions (or Decision Tree), why a particular instance is classified as churn or not churn.

Based on all performance metrics and human comprehensibility, we have selected the CART to be used as the final model for churn prediction.

*** Thank You ***