MSIS 5633-Predictive Analytics Technologies

Customer Churn Prediction

Predicting and Explaining Customer Churn

By
Naveen Varma Patsamatla

Table of content

Executive Summary:

This report is dedicated to the thorough analysis of customer churn and attrition behavior. Our primary goal is to leverage a dataset consisting of 1,000 rows and 39 columns to develop advanced machine learning models for predictive insights. The objective is to scrutinize customer churn by considering socio-demographic and Behavioral attributes.

In our pursuit of these objectives, we have adhered to the CRISP-DM methodology. It has guided our approach and ensured a robust and systematic analysis.

To attain goals, we explored a diverse array of machine learning models, including Decision Tree, Random Forest, Artificial Neural Network, Logistic Regression, k-Nearest Neighbor, and Naïve Bayes. Rigorous evaluation was conducted, considering all relevant factors, to identify the model that excels in pattern recognition. Our selection process was driven by a comprehensive assessment of balancing power of prediction and ROC curve values.

Through this report, we aim to provide actionable insights and recommendations that empower the business to make informed decisions regarding customer churn, ultimately enhancing overall performance and customer retention.

Phase 1: Business Understanding

It is imperative to grasp the business objectives and goals comprehensively. In this context, our primary aim is to effectively predict customer churn or attrition behavior. And we see business as successful when one can proactively identify customers likely to churn and implement retention strategies. From a data mining perspective, success hinges on the ability to develop a predictive model that not only delivers high accuracy but also strikes a balance between sensitivity and specificity. This equilibrium is essential to meet the specific needs and objectives of the business. Finally, we use Knime platform to solve this issue.

Phase 2: Data Understanding

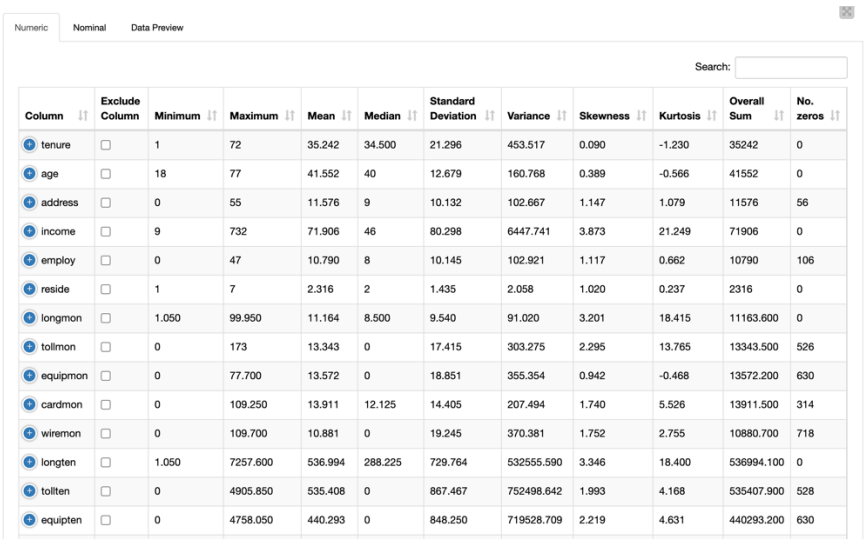
In essence, Data Understanding is the crucial initial phase of any data analysis process, where you collect, describe, explore, and verify the data to prepare for more in-depth analysis and modeling.

Collecting Data:

Data is gathered from the German-telecommunications provider. This is the starting point for our analysis.

Describing the Data:

In the second task, we provide a snapshot of the dataset. It comprises 1000 records and contains 39 features or attributes. These attributes can be divided into two categories: 18 are numeric features, and 21 are nominal, Binomial features. The dataset contains information related to socio-demographics (such as age, marital status, and gender) and Behavioral attributes(Hours of usage, Selected services).



Numeric | Nominal | Data Preview

Search:

Column	Exclude Column	Minimum	Maximum	Mean	Median	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum	No. zeros
tenure	☐	1	72	35.242	34.500	21.296	453.517	0.090	-1.230	35242	0
age	☐	18	77	41.552	40	12.679	160.768	0.389	-0.566	41552	0
address	☐	0	55	11.576	9	10.132	102.667	1.147	1.079	11576	56
income	☐	9	732	71.906	46	80.298	6447.741	3.873	21.249	71906	0
employ	☐	0	47	10.790	8	10.145	102.921	1.117	0.662	10790	106
reside	☐	1	7	2.316	2	1.435	2.058	1.020	0.237	2316	0
longmon	☐	1.050	99.950	11.164	8.500	9.540	91.020	3.201	18.415	11163.600	0
tollmon	☐	0	173	13.343	0	17.415	303.275	2.295	13.765	13343.500	526
equipmon	☐	0	77.700	13.572	0	18.851	355.354	0.942	-0.468	13572.200	630
cardmon	☐	0	109.250	13.911	12.125	14.405	207.494	1.740	5.526	13911.500	314
wiremon	☐	0	109.700	10.881	0	19.245	370.381	1.752	2.755	10880.700	718
longten	☐	1.050	7257.600	536.994	288.225	729.764	532555.590	3.346	18.400	536994.100	0
tollten	☐	0	4905.850	535.408	0	867.467	752498.642	1.993	4.168	535407.900	528
equipten	☐	0	4758.050	440.293	0	848.250	719528.709	2.219	4.631	440293.200	630

Fig 1: snapshot of the data provided

Exploring the Data:

Here, we delve into the dataset by performing exploratory data analysis. This involves applying statistical concepts like mean, median, and mode, as well as using graphical or visual analysis tools. We also utilize measures such as standard deviation, variance, and skewness to gain a better understanding of the distribution of each feature. we did this using data explorer in knime analytics tool.

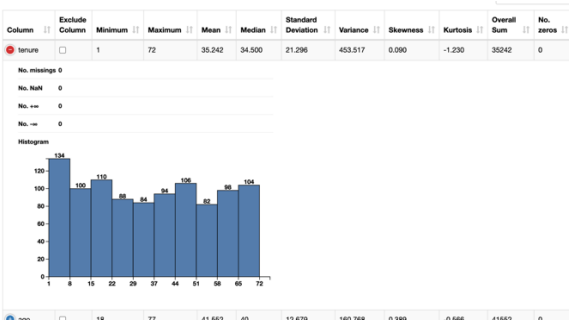


Fig 2: Image showing descriptive Statistics and Histogram of a feature

Verifying Data Quality:
In the fourth task, we assess the quality of the data. We discover that there are numerous missing and null values within the dataset. Additionally, we identify a few outliers that could potentially impact the integrity of our analysis. Furthermore, we observe an imbalance in the data concerning the target variable. To address this imbalance and ensure that our model isn't biased toward the majority class, we need to implement data balancing techniques. We don't have any missing values , nulls, NAN's. By this we can assure data provided is good quality data.

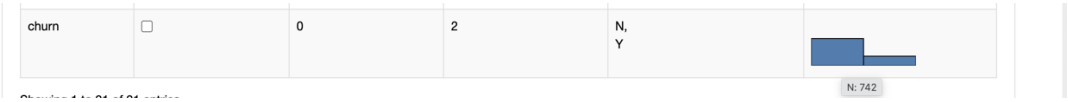


Fig 3: Image describing the data imbalance

Phase 3: Data Preparation

In this phase, we focus on getting the data ready for training our model. We go through several essential steps, such as cleaning the data, transforming it, and reducing it. This phase is crucial because the quality and quantity of the data have a significant impact on the accuracy of our predictions. The better the data we provide to our model, the better the predictions it can make.

Selecting Data :

Choosing the right data (features or variables) is vital. Adding unnecessary or irrelevant data to train our model can lead to overfitting, which means it becomes too sensitive to noise and may produce inaccurate results. In our case, we've determined that the "cust_id" is not an important feature, as it doesn't affect the outcome of our prediction model or help us uncover the underlying parameter values. So, we used Column filter to remove the column in Knime.

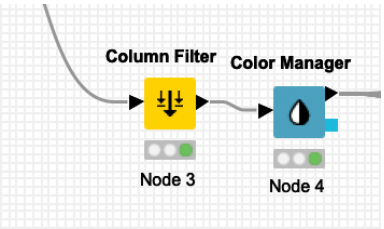


Fig 4 : Using column filter to eliminate a column

Cleaning Data:

This step involves handling missing data, dealing with outliers (either by imputing or removing them), addressing anomalies, and encoding the data in a way suitable for the model. We've confirmed that we don't have any missing values, null values, or Nan's in our dataset. Additionally, we've performed data normalization, specifically using min-max normalization for numerical predictive models. We also used equal-sampling technique to ensure that there are equal amount of yes and no' s to train our model so that our model is not biased.

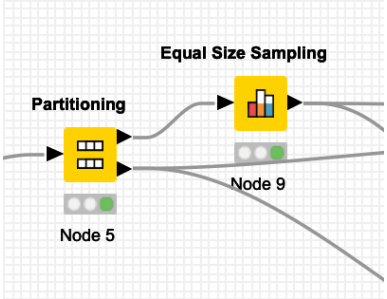


Fig 5 : Using equal-size sampling to balance the data

Formatting Data:

We've performed data encoding, which involves converting necessary columns from nominal to numeric format to make them compatible with number-based models like Artificial Neural Networks (ANN), Logistic Regression, and K-Nearest Neighbors (KNN), among others.

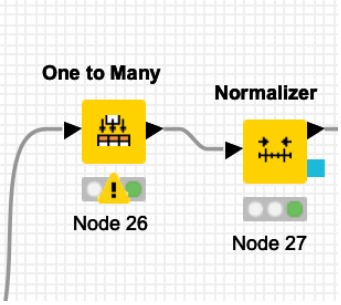


Fig 6: using One to Many Node to perform Binary Encoding

Phase 4: Modeling

Modeling is important aspect as choosing appropriate model to the dataset is often hard and yields in better results.

Selecting Model techniques:

We tried multiple models of both the Set-based models and Number-based models. below here is the list of the models used.
Set-based models: Decision tree (DT), Random Forest(RF)
Number-based models: Artificial neural networks(ANN), Logistic Regression(LR), K-nearest neighbors(KNN).

Generate Test Design:

We generated two test designs to see how models performs and give the better results.
First method: single split method(testing = 70%,traning = 30%)
Second method: K-fold cross validation(K=10).

Building Model:

As part of our analysis, we've constructed several models, and in this discussion, we'll provide an overview of how they were developed.

Decision Tree (DT)

The Decision Tree is a type of non-parametric model used for tasks like classification and regression. It's designed to help us gain a deeper understanding of our data. This model takes the form of a tree, with internal nodes, branches, and leaves. The leaves represent predictions or outcomes, while internal nodes employ various metrics to split into other nodes. These metrics can be Gini impurity or Gain ratio, which measure the quality of the split. In our case, we used Gini impurity as it suits many of our datasets. We didn't employ any pruning methods, and we chose to have binary nominal splits. Based on the decision tree splits longten ,employ variables has more influence on making final predictions of the model as they are close to the root node and highly important in decision making.

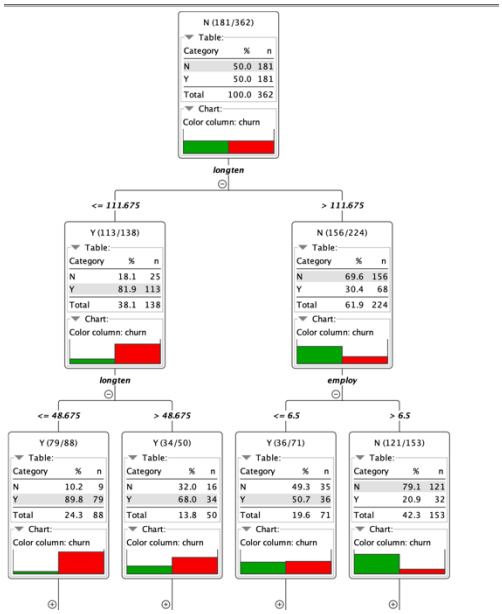


Fig 7: Top3 levels of DT-learner

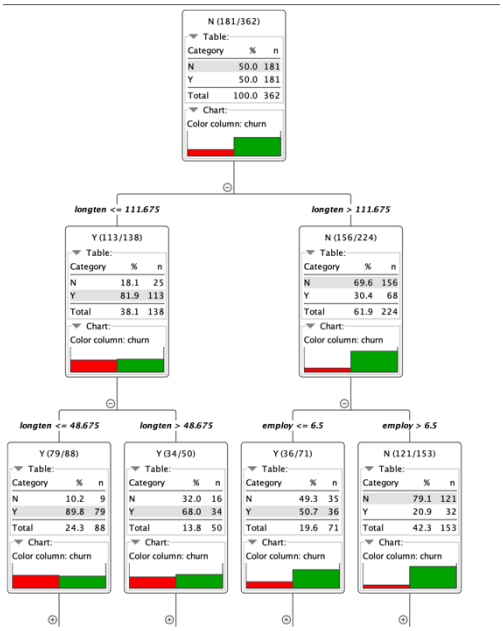


Fig 8: Top3 levels of DT-predictor

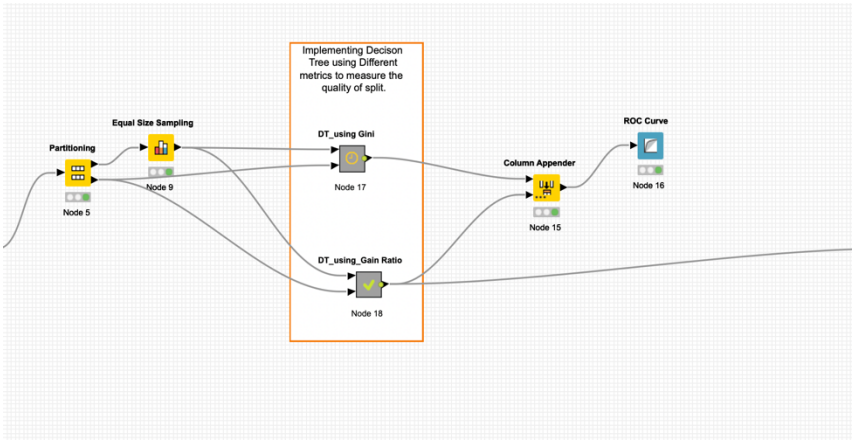
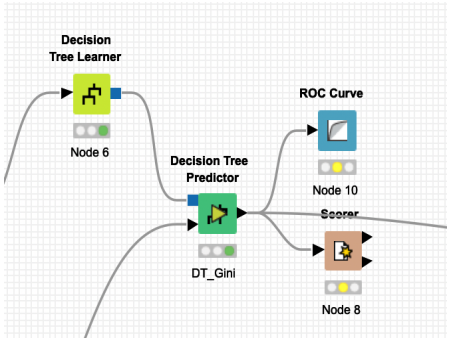


Fig 9: Workflow of Decision Tree models using different metrics for quality of split



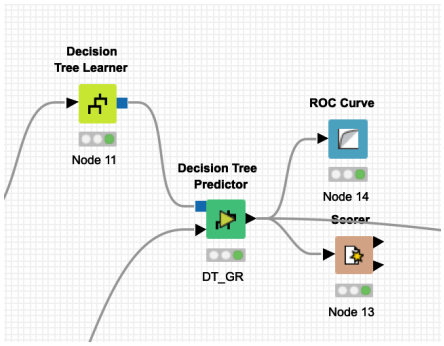


Fig 10,11 : How DT models are built using different metrics

Random Forest (RF)

Random Forest is a supervised ensemble model that combines multiple independent trees to predict the target variable. It makes decisions based on the majority vote of the trees. We built two models using different quality measures for the split, namely Gini impurity and Information Gain ratio, and compared the results for both. We also experimented with different tree quantities, using 100 and 1000 trees to potentially use all variables for splitting. Additionally, we conducted Variable Importance analysis using 1000 trees to understand the significance of each variable.

Based on the Variable Importance Graph is clear that longten, cardten, longmon and few other variables are most used and must be included dataset used for training the model.As we calculated using the weight of each variable occurrence in all the levels.

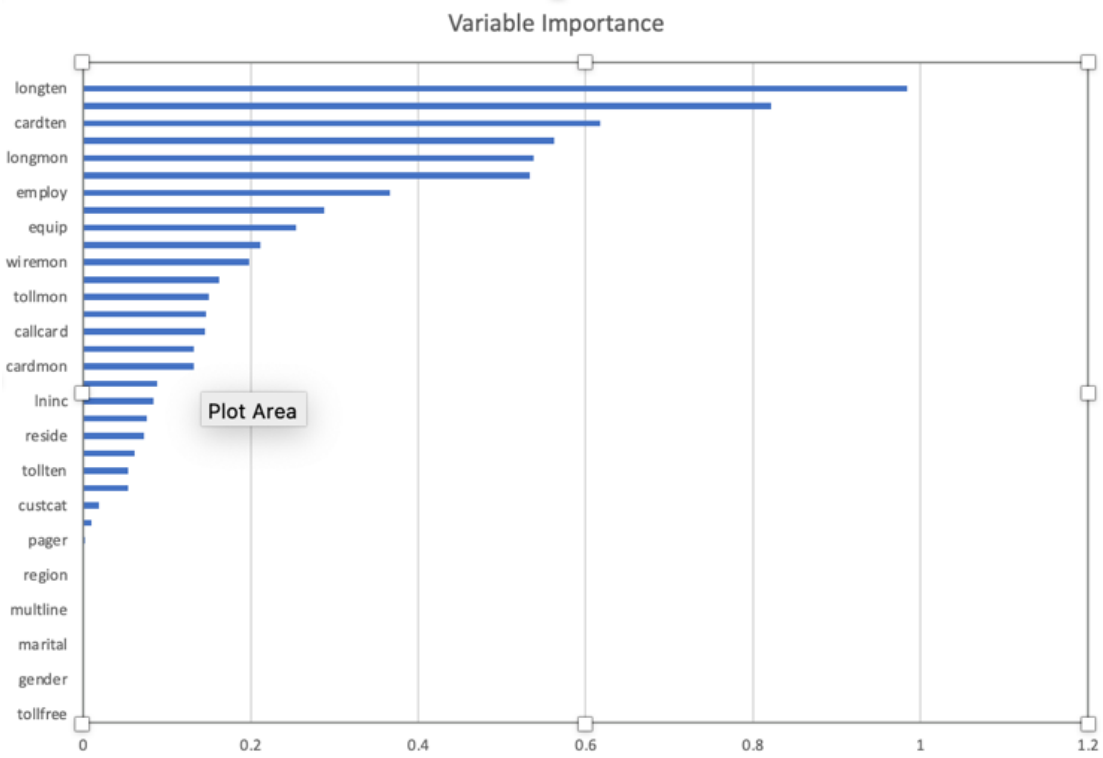


Fig 12: Variable Importance using Excel

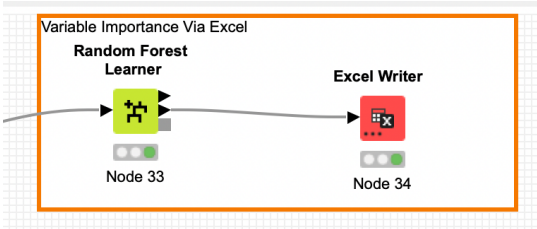


Fig 13 : Importing Values to excel in knime

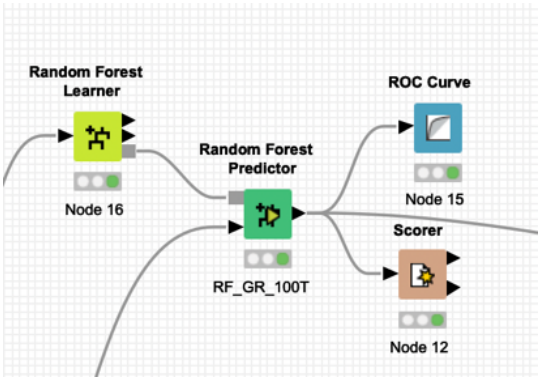


Fig 14: Building RF models using Knime

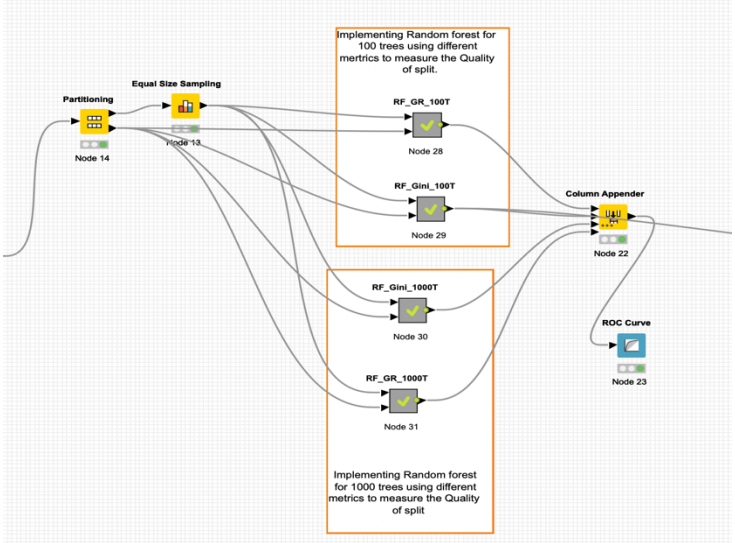


Fig 15: Complete workflow of RF models

Artificial Neural Network (ANN)

ANN is designed to mimic the biological neural networks in the human brain. Neurons in ANN calculate their outputs based on weights from various inputs or the previous layer, employing an activation function. During training, the model adjusts the weights between connections to improve performance. We created two basic multi-layer perception (MLP) ANN models, one with a single hidden layer containing 10 neurons and another with 20 neurons, to compare their results.

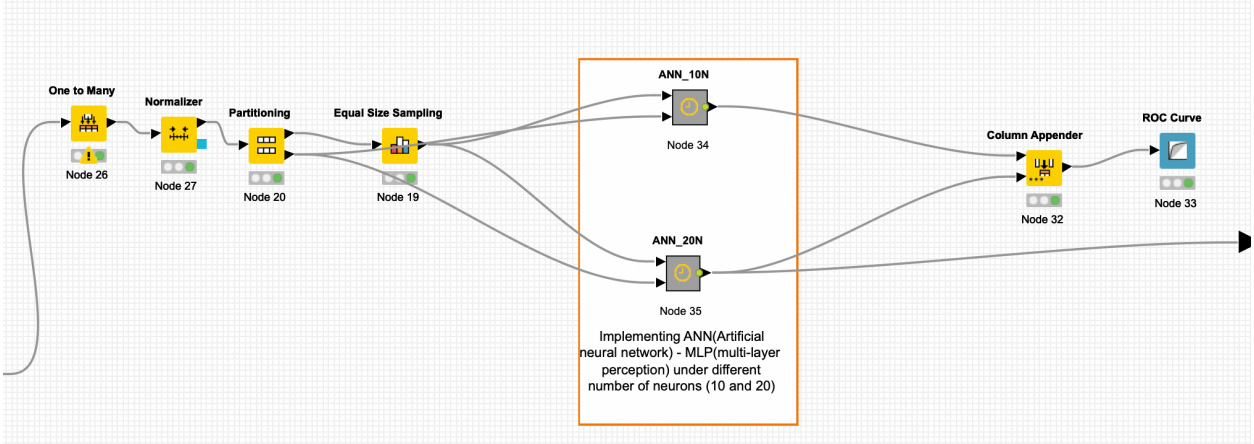


Fig 16: Complete workflow of ANN models

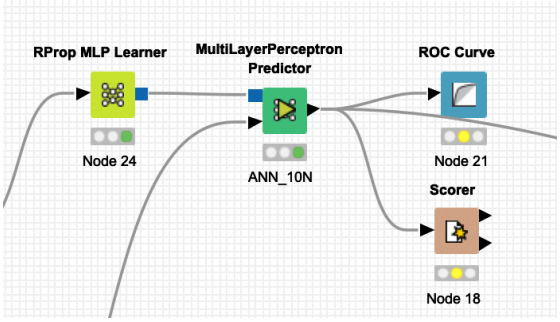


Fig 17: Building ANN models using Knime

K-Nearest Neighbor (KNN)

KNN is a simple non-parametric algorithm used for both regression and classification tasks. It operates under the assumption that similar data points tend to be close to each other in feature space. Importantly, it doesn't rely on any underlying data distribution assumptions and stores the entire dataset for making predictions. In our initial model, we assumed a hyperparameter 'k' value of 5.

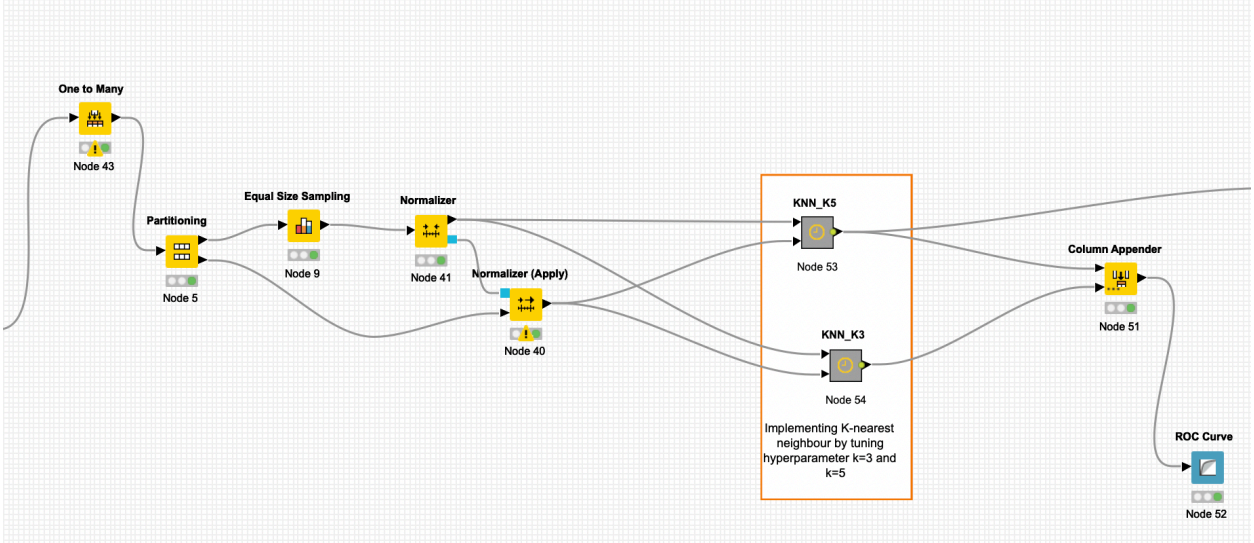


Fig 18: Complete workflow of KNN models

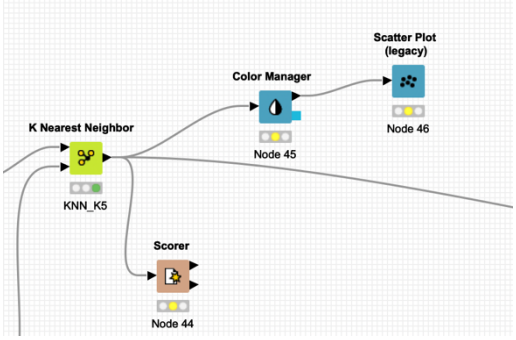


Fig 19: Building of KNN models

Naïve Bayes (NB)

Naïve Bayes is a supervised machine learning algorithm is highly used for classification based on Bayes theorem.

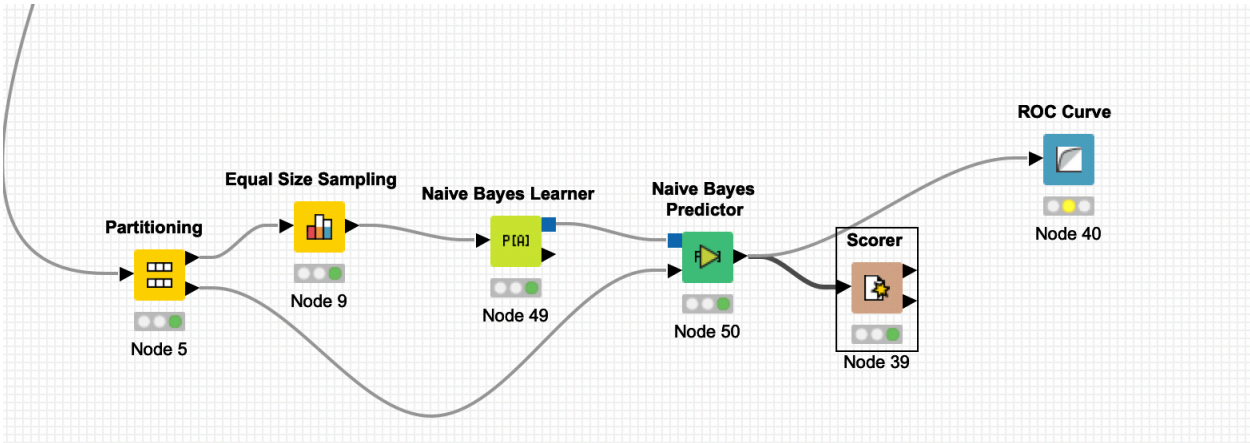


Fig 20: Complete workflow of NB model

Logistic Regression (LR)

Logistic Regression is a commonly used model for classification tasks. It employs logistic or sigmoid functions to map values and make predictions.

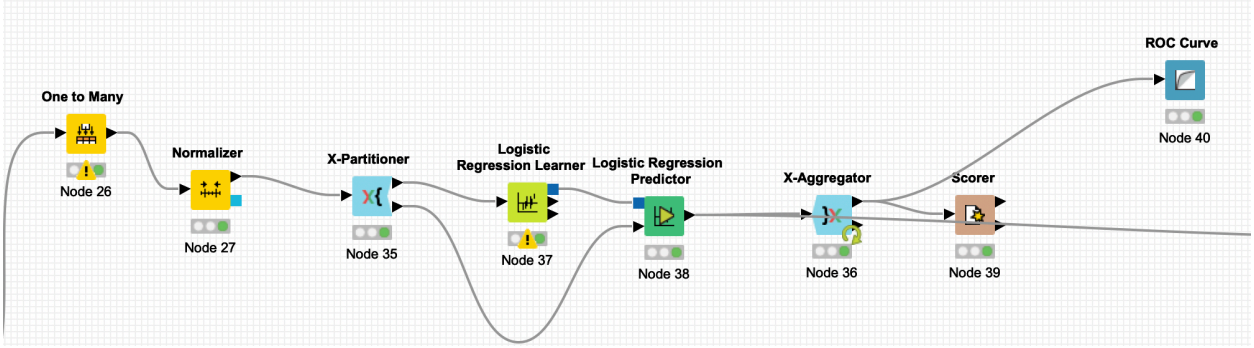


Fig 21: Complete workflow of LR model

Phase 5: Evaluation

Evaluation is very important it is a measure of verifying the model whether the predictions are close to the actual values. Tabulated evaluations of each model for different test designs have been listed below :

Single split Method(70%-train,30%-test):

Decision Tree(DT):

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
DT using GINI	72	0.688	0.731	0.759
DT using Gain ratio	75.3	0.727	0.762	0.787

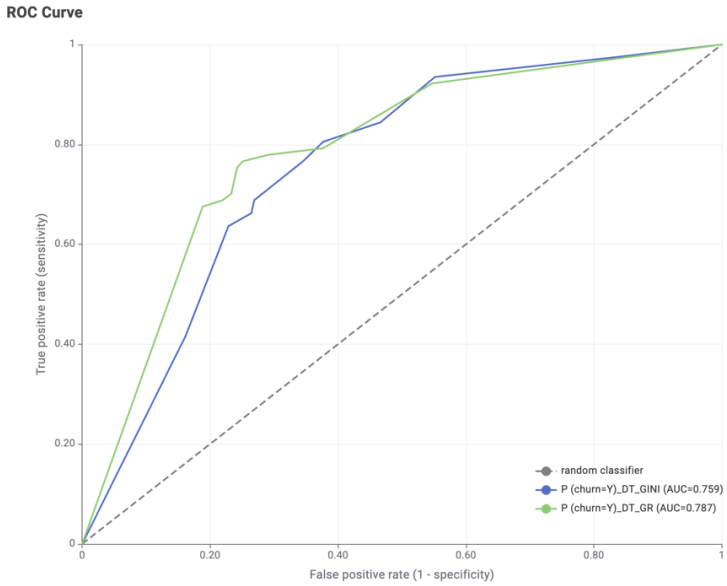


Fig 22: ROC curve for different DT models

The Lines Highlighted with orange color are best models among the observed or tested group of models.

Random Forest (RF):

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
RF using GINI 100T	82.3	0.87	0.807	0.898
RF using Gain Ratio 100T	81.3	0.792	0.821	0.889
RF using GINI 1000T	81.7	0.844	0.807	0.892
RF using Gain Ratio 1000T	82.3	0.818	0.825	0.892

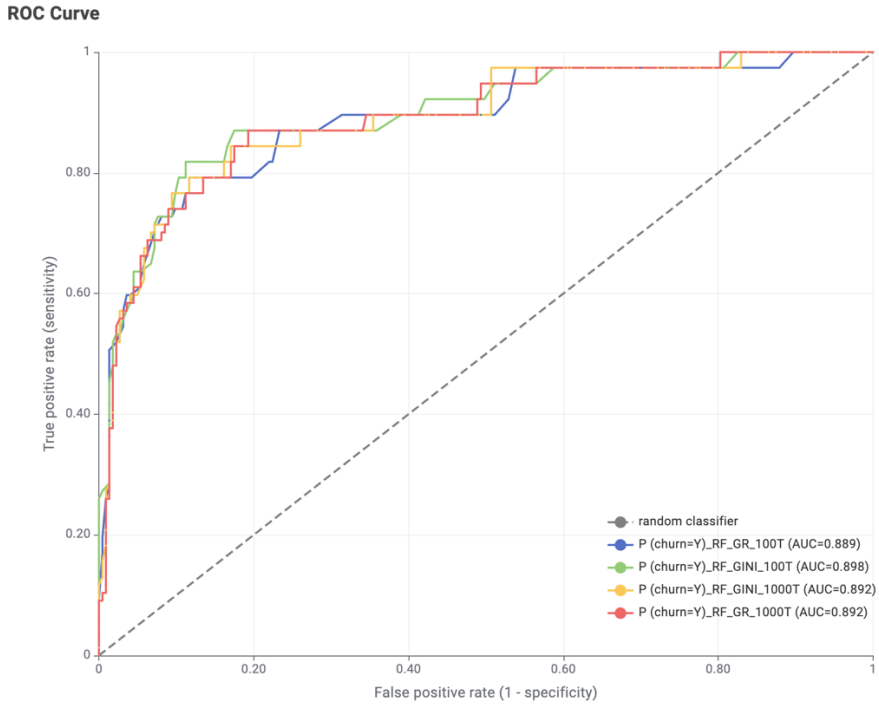


Fig 23: ROC curve for different RF models

ANN-MLP:

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
ANN_10N	76.7	0.87	0.731	0.838
ANN_20N	82.3	0.922	0.789	0.900

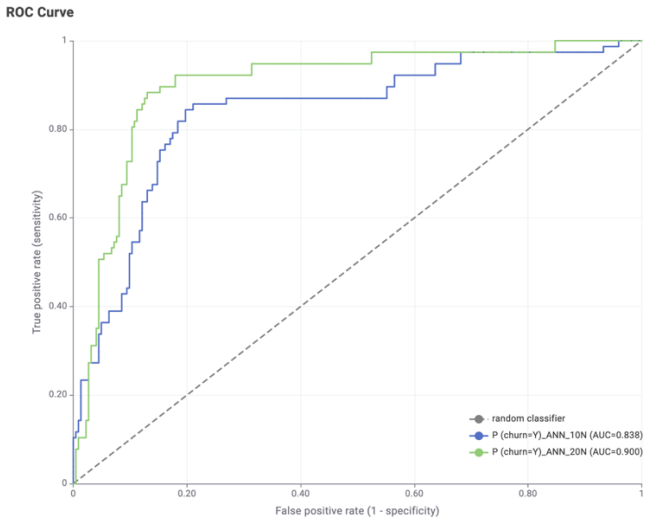


Fig 24: ROC curve for different ANN models

KNN:

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
KNN_K3	0.69	0.714	0.682	0.784
KNN_K5	0.70	0.714	0.695	0.738

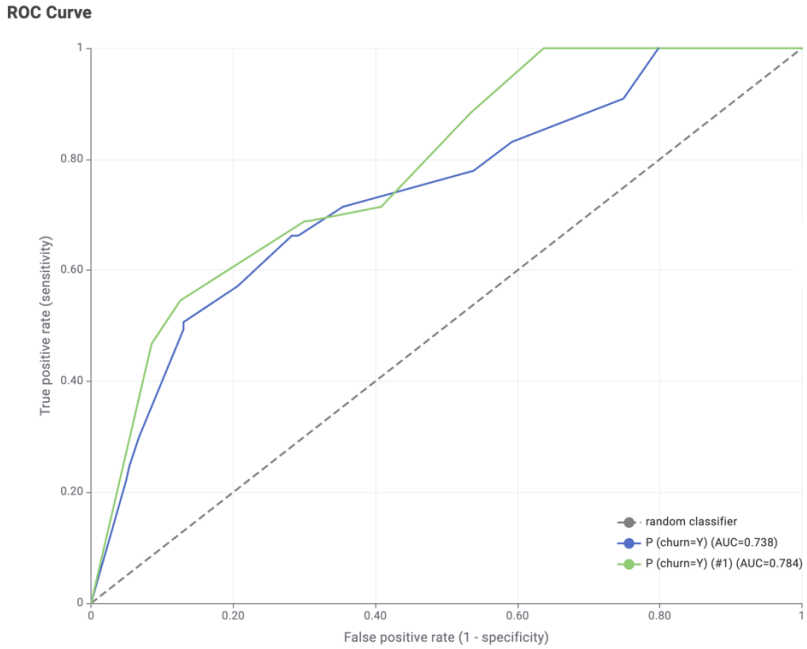


Fig 25: ROC curve for different KNN models

Naïve Bayes

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
NB	56	0.831	0.466	0.773

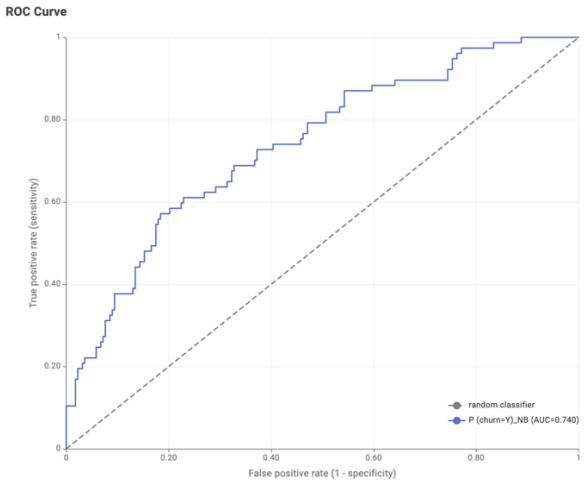


Fig 26: ROC curve for NB model

Logistic Regression

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
LR_balance	0.67	0.662	0.673	0.775
LR_imbalance	0.79	0.455	0.906	0.787

In this study, our objective was to underscore the significance of data set balancing. We introduced two models for comparison: LR_balance, which employed equal-sampling techniques, and LR_imbalance, which did not address data balance. Notably, LR_balance outperformed LR_imbalance in the final results. It was evident that LR_imbalance exhibited a bias toward the majority cluster, which in this case is 'N,' thus excelling in predicting true negatives.

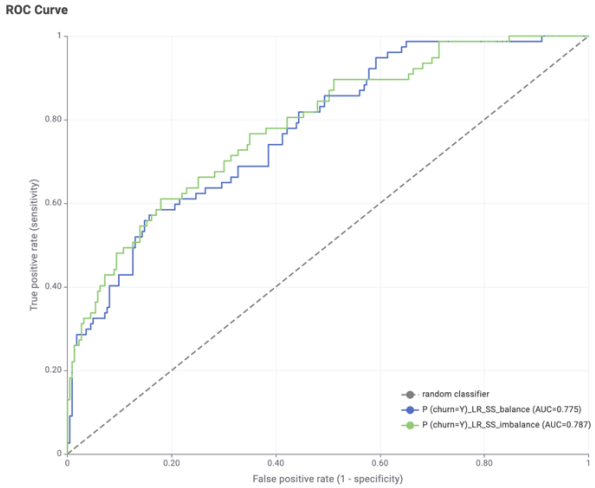


Fig 27: ROC curves for balance and imbalance data of LR

K-fold cross validation(K=10):

Logistic Regression

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
LR	77.1	0.74	0.802	0.831

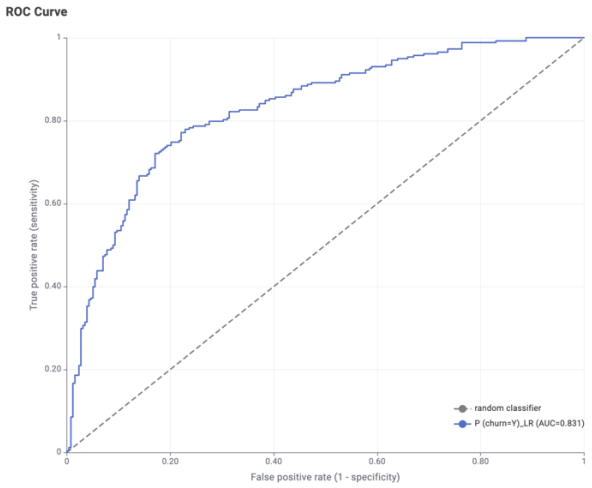


Fig 28: ROC curve for LR model

Model	Accuracy(%)	Sensitivity (Y)	Specificity (Y)	AUC
LR_balance(single split)	0.67	0.662	0.673	0.775

After comparing the results for Logistic Regression using 10-fold cross validation and single split it clear that 10-fold cross validation has better results.

Comparative analysis of all Models:

Now we have multiple models built we need to compare the results to choose the best model. Here I used best models of each category(RF,DT,KNN,ANN,NB,LR) and compared all of them using ROC graph.

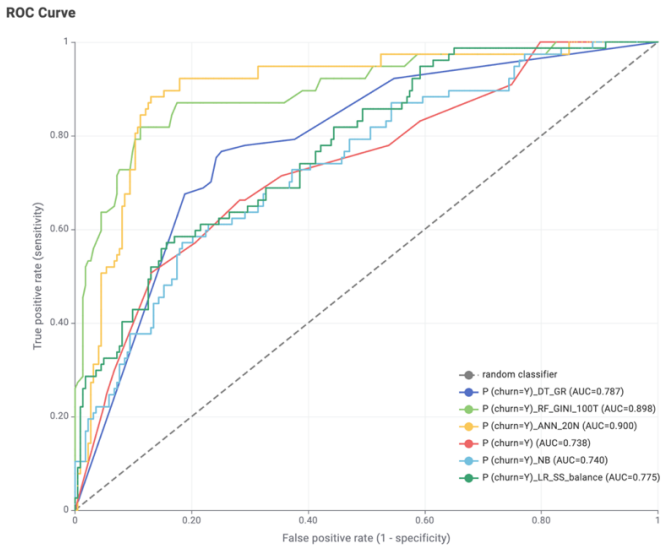


Fig 29: ROC curve for best models from different types

Overall final Workflow

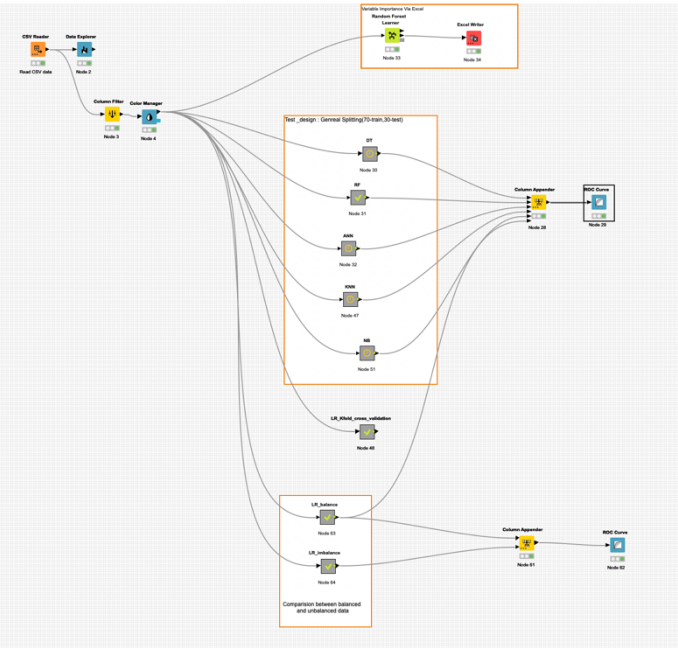


Fig 30: Overall final workflow

Phase 6: Deployment

Model remains undeployed at this stage, as we are still in the process of model development. But would recommend deploying ANN model or RF model to see the better results.

Conclusion

In conclusion, after a comprehensive analysis and evaluation of multiple models for predicting customer churn/attrition behavior, the Artificial Neural Network - Multi-layer Perceptron with 20 neurons in each layer (ANN_20N) emerged as the most promising model, boasting an impressive performance with an AUC of 0.90, Sensitivity of 0.922, Specificity of 0.789, and an Accuracy of 82.3%. This model demonstrates a high capability to accurately predict both true positives and true negatives.

Additionally, the Random Forest model, comprising 100 trees and utilizing the GINI index, also exhibited strong results, with an AUC of 0.898, Sensitivity of 0.87, Specificity of 0.807, and an Accuracy of 82.3%. It, too, demonstrates reliability in predicting true positives and true negatives.

Given the closely competitive performance of these two models, it is recommended to deploy both the Artificial Neural Network and the Random Forest model for predicting customer churn/attrition. This approach ensures a robust and comprehensive predictive framework, maximizing the accuracy and reliability of churn predictions.