Telecom Churn Predictor

IE 7275 - Data Mining in Engineering

Objective

Churn prediction is the prediction of which consumers are likely to leave your business or cancel a service subscription based on their behavior with your product.

In this competitive world, where every telecom company is working to get more customers, every company needs to know who its potential customers are and understand what they are really looking for.

Data is used by business leaders to check the trends and improve decision-making.

Every business needs to understand its clientele if it wants to boost sales. With this data, we will try to build the prediction model considering the different features like revenue, recurring charges, calling minutes, etc. to understand the behavior of the telecom industry churners.

Data Description

The objective of the dataset is to be utilized to forecast customer behavior, revenue, productivity, and other factors. Every business needs to understand its clientele if it wants to boost sales.

With this data, we will try to build the prediction model considering the different features like revenue, recurring charges, calling minutes, etc. to understand the behavior of the telecom industry churners.

This dataset was taken from Kaggle. Here is the link to the dataset.
https://www.kaggle.com/code/mahwiz/telecom-customer-churn-deeplearning/data

Variables

MonthlyRevenue	MonthsInService	ReceivedCalls
MonthlyMinutes	UniqueSubs	OutboundCalls
TotalRecurringCharge	ActiveSubs	InboundCalls
DirectorAssistedCalls	Handsets	PeakCallsInOut
OverageMinutes	HandsetModels	OffPeakCallsInOut
RoamingCalls	CurrentEquipmentDays	DroppedBlockedCalls
PercChangeMinutes	AgeHH1	CallForwardingCalls
PercChangeRevenues	AgeHH2	CallWaitingCalls
DroppedCalls	RetentionCalls	AdjustmentsToCreditRating
BlockedCalls	RetentionOffersAccepted	ThreewayCalls
UnansweredCalls	ReferralsMadeBySubscriber	IncomeGroup

Data Pre-Processing

Following are the initial set of actions we carried out on the data frame that was uploaded.

1. We used shape() to find out that we have 51047 rows and 57 attributes.
2. Used describe() to check all the statistical properties of the given dataset i.e mean, median, 25th, 50th, 75th, min, and max values.
3. With the help of info(), we observed the data types of each attribute throughout the dataset along with their corresponding null values.
4. Used isnull() to find the number of null values with each column. There were in total 3515 null values across all columns.
5. Thereafter, we used labelencoder() from sklearn to convert categorial attributes to numerical attributes. We have performed these changes on Homeownership, BuysViaMailOrder, RespondsToMailOffers, OptOutMailings, etc. columns
6. In order to double-check null values, we have omitted all non-important object datatypes and selected only the float64 and int64 datatypes. We have executed this using isnull().sum() through which we have eliminated all null values in our Model Data Frame.
7. Checked the correlation between all the attributes in order to select the most relevant attributes to feed in the model.

Exploratory Data Analysis

a. Categorical Variables

For each categorical feature, the unique values were found out, and it was considered that all the 17 categorical features, including the target variable “churn”, were binary in nature.

Therefore, the hist() function was used to visualize those binary variables graphically. As observed from the below histograms, most of the instances of this dataset do not contain the feature being selected by the label ‘0’ and if they are chosen, it has been indicated using ‘1’.

There is a large number of churning out possible in a case where the Handset is Web Capable. Also, there is a large number of churning out possible if the customer owns a credit card.

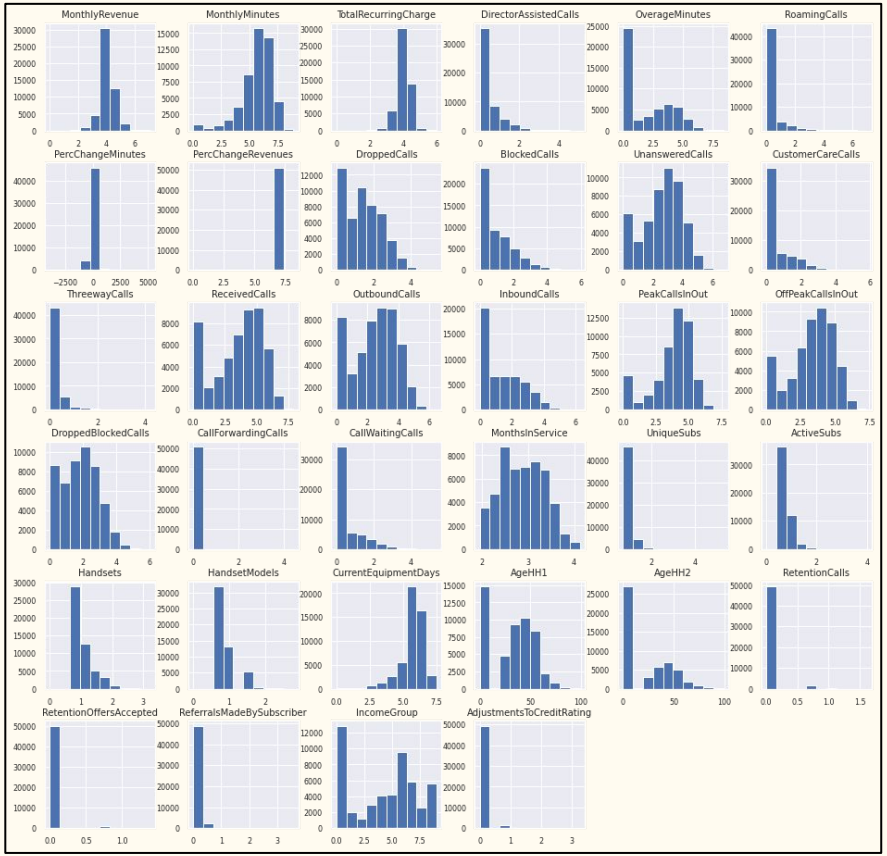
Exploratory Data Analysis

b. Numerical Variables

After performing the data pre-processing steps, there were 51047 instances and 34 numerical variables present in the dataset.

The graph represents the frequency of different numerical features from the total available instances of the dataset.

From the summary statistics, it was evident that there is a scale difference between the different numeric variables, which should be re-scaled before proceeding with the model-building phase.

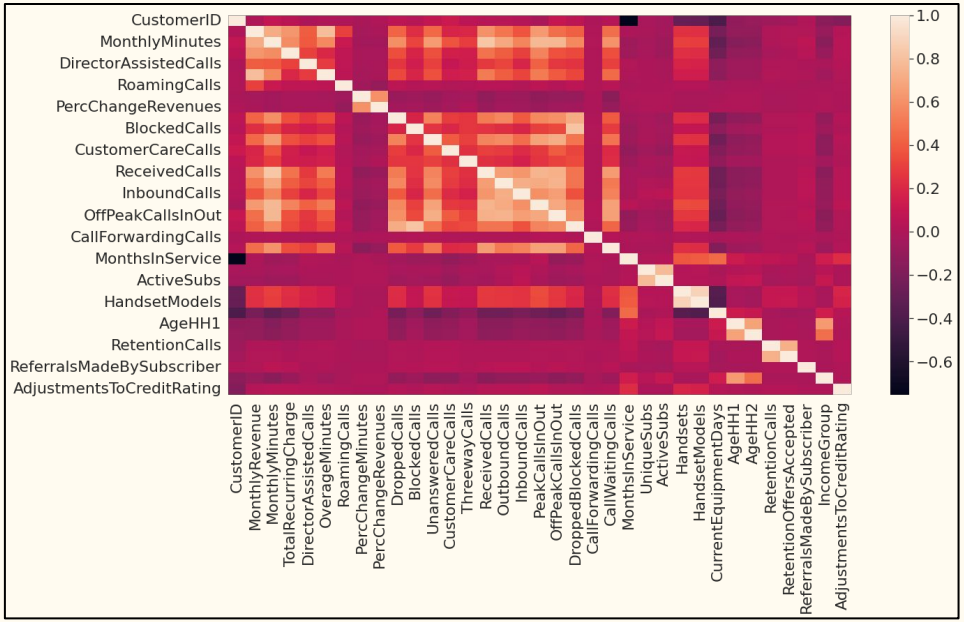


Dimension Reduction and Variable Selection

In the heat map you can see that duration of calls- incoming, outgoing and offpeak calls, and receiving calls are highly correlated.

So thus we removed 3 out of 4 of them also removed customerID ad that variable does not decide churn prediction.

And used PCA for dimensionality reduction.



Data Partitioning

The predictor variables were collectively represented by variable ‘X’ and the target variable ‘Churn’ was represented by variable ‘Y’.

Variables with index 0 to 24 were defined by ‘X’ variable and the target variable with index 25 was represented as variable ‘Y’.

The stratified shuffle split method for data partitioning was used on dataset, such that the train and test datasets were generated with the ratio of 70:30.

The test data was further splitted to get the data for validation with the ratio of 70:30. This means that the 70% of data is used for training of the classification model along with the validation data (30% out of the 70% training data) and other 30% data is used for testing of the classification model and evaluation of the classification performance of those models.

X_train contained 25012 records and 25 variables, whereas the X_validation contained 10720 records and 25 variables. X_test contained 15315 records along with the 25 variables. Y_train contained 125012 records and 1 variable, Y_validation contained 10720 records and 1 variable, while Y_test contained 15315 records and 1 variable.

1. Model Building: Logistic Regression

Logistic Regression is a parametric classification model that relies on a specific model relating the predictor variables with the target variable, such that the outcome variable is categorical in nature. The output is the estimates of the probabilities of belonging to each class, then using a threshold cutoff on the probabilities for classifying into either of the classes. The outcome variable is called logit, which can be modeled as a linear function of the predictors.

Advantages: - Easily understandable and offers an intuitive explanation of predictors. - Computationally fast and cheap to classify large samples of new data.

Disadvantages: - Cannot be used to solve nonlinear problems. - Requires predictor variables to be linearly associated with log odds. - Sensitive to outliers.

Implementation: - The base logistic regression model was executed on the scaled data, resulting in an accuracy of 100%. The ‘liblinear’ solver and penalty = 12 were used for obtaining the optimal parameters for the training model.

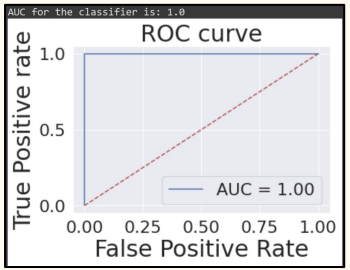
Performance Evaluation: Logistic Regression

A grid-search was used to obtain the best model using penalty = 12 and solver = 'liblinear', with the highest accuracy of 100%.

The sensitivity value of 100% indicates that it was able to classify the True Positives or the non-churning instances correctly.

The specificity value of 100% indicates that it correctly classified the True Negatives or the churning instances. An F1-score of 100.0% indicates the non-churning instances were identified correctly.

The ROC curve was aligned with the top-left corner, with an AUC value of 1.0, indicating perfect discrimination between the non-churning and churning classes. Hence, with 100% of accuracy, this model can be considered the perfect classifier.



Evaluation Matrix

	Overall Accuracy	Validation Error	Sensitivity	Specificity	f1-score
	1.0	0.0	1.0	1.0	1.0

2. Model Building: Neural Networks

Neural networks are models which may be used for a classification or prediction problem. Deep neural networks and feature extraction can both be performed using some sophisticated neural network models. The neurons found in human brains are mimicked by the neural network model because they share characteristics with learning and experience-based memory.

Advantages: - Results in a good predictive performance. - Can tolerate noisy data. - Capable of capturing highly complicated relationships between predictor variables and the target variable.

Disadvantages: - Computationally expensive as they require longer runtime for training, which increases as the number of predictors increases. - The structure of the model lacks interpretability. - Prone to overfitting.

Implementation: - Implemented two different neural networks, one with 25 neurons in input layers, 512 neurons in the hidden layer with RELU as activation function and output layer is single neuron with sigmoid as activation function. With 1st neural network accuracy is 93%. Another network with same input and output layer with hidden layer having 16 neurons init. 2nd network gives accuracy worth 80% which means reducing the neurons in hidden layer will hamper the accuracy

Performance Evaluation: Neural Network

A grid search was used to find the best model using 25 neurons in input layers, 512 neurons in hidden layer with RELU as activation function and output layer is single neuron with sigmoid as activation function with an accuracy of 95.67%.

The sensitivity value of 100% indicates that it correctly classified the True Positives or the non-churning instances.

The specificity value of 84.97% indicates that it correctly classified the True Negatives or the churning instances.

An F1-score of 89.5% indicates low False Positives and low False Negatives, hence correctly identifying non-churning instances.

The ROC curve was closer to the top-left corner, with an AUC value of 0.92, indicating nearly perfect discrimination between the non-churning and churning classes. Another neural network was implemented with the same input and output layer with a hidden layer having 16 neurons. The Accuracy of that model was achieved as 71.08% with a sensitivity of 96.01% and specificity of 9.51%. That model had an F1-score of 100% and an AUC value of 0.52. The second neural network model can be considered a weak classifier with less AUC value.

Performance Evaluation: Neural Network

Evaluation Matrix

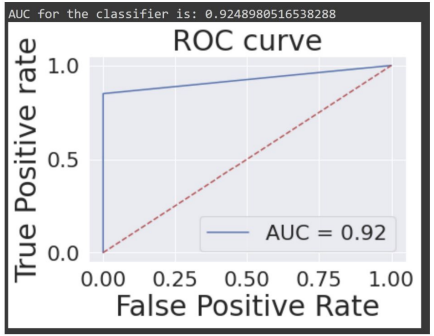
	Overall Accuracy	Validation Error	Sensitivity	Specificity	f1-score
	0.9454129937969311	0.05458700620306889	1.0	0.8106026280018124	0.8953953953953954

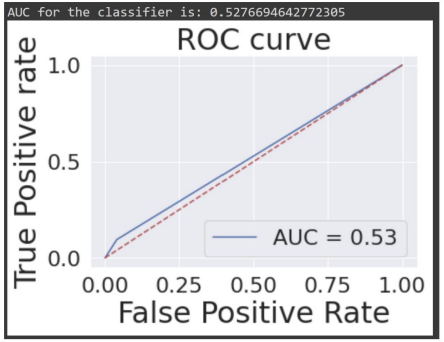
782/782 [==============================] - 2s 2ms/step

Evaluation Matrix

	Overall Accuracy	Validation Error	Sensitivity	Specificity	f1-score
	0.7765589291544238	0.22344107084557618	0.9951380607283735	0.23674671499773448	0.3791727140783745

782/782 [==============================] - 1s 2ms/step





3. Model Building: CNN

Machine learning includes convolutional neural networks, also known as convnets or CNNs. It is a subset of the several artificial neural network models that are employed for diverse purposes and data sets. A CNN is a particular type of network design for deep learning algorithms that is utilized for tasks like image recognition and pixel data processing.

Advantages: - It has very high accuracy in image recognition problems. - It automatically detects important features without any human interventions.

Disadvantages: - CNN does not encode the position and orientation of the object. - The large scaled data is required. - Lack of ability to be spatially invariant to the input data.

Implementation: - The convolutional neural network model was executed, resulting in an accuracy of 81.827%. The implemented network had an input layer with 25 neurons, convoluted 1D, a flattened layer, and a dense output layer with a sigmoid as an activation function.

Performance Evaluation: CNN

The convolutional neural network model was executed, with the input layer with 25 neurons, convoluted 1D, flattened layer, and a dense output layer with sigmoid as an activation function.

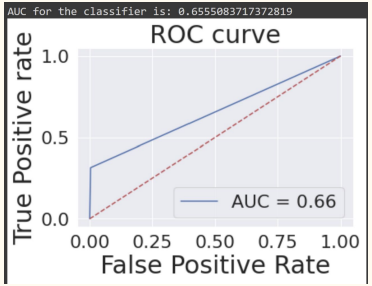
The achieved accuracy for this model to predict the correct classification observation is 80%.

The sensitivity of this model is 99.97% indicating that it was able to classify True Positives and non-churning instances correctly.

The specificity is 31.42% indicating that the model was unable to classify the True Negatives or the churning instances correctly.

A low F1-score of 45.1% represents high False Positives and high False Negatives, hence not correctly identifying non-churning instances.

The ROC curve was away from the top-left corner, with an AUC value of 0.66, indicating poor discrimination between the non-churning and churning classes.



Evaluation Matrix

	Overall Accuracy	Validation Error	Sensitivity	Specificity	f1-score
	0.7890303623898139	0.21096963761018606	0.9866067333272177	0.3010874490258269	0.4513499745287825

782/782 [==============================] - 1s 2ms/step

4. Model Building: Decision Tree Classifier

It is a Supervised Machine Learning technique that can be used to solve both regression and classification problems. It is a tree-like structure that helps to make predictions based on different conditions. Trees split on different predictors and help in the segregation of records.

Advantages: - It helps to perform classification without requiring much computation. - It helps to understand which fields are most important for prediction and classification. - variable subset selection occurs automatically in each split.

Disadvantages: - They are sensitive to changes in data, and any change can result in different splits. - As the splits occur on one predictor at a time, instead of combinations of predictors, they might miss recognizing relationships between predictors. - They are expensive to grow, as multiple sorting operations are carried out in computing all possible splits on every variable.

Implementation: - The base decision tree model was run with min_samples_leaf = 5, resulting in an accuracy 99.836%. For optimizing the decision tree model, max_depth of tree varied up to a length of 30. The arguments impurity criteria ‘gini’ and ‘entropy’ were introduced, and min_samples_leaf = 5 was kept constant as it got the best accuracy.

Performance Evaluation: Decision Tree Classifier

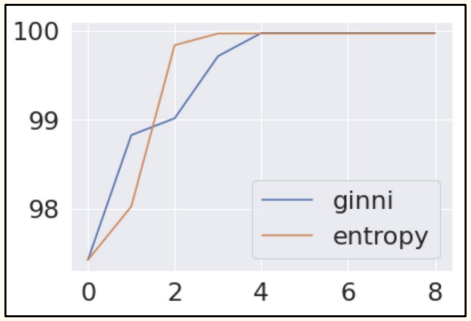
A grid search was done to obtain the best model using min_samples_leaf = 5, impurity = 'entropy', with an accuracy of 99.83%.

The sensitivity value of 99.91% indicates that it correctly classified the True Positives or the non-churning instances.

The specificity value of 99.63% indicates that it correctly classified the True Negatives or the churning instances.

A high F1-score of 99.7% indicates low False Positives and low False Negatives, hence correctly identifying non-churning instances.

The ROC curve was closer to the top-left corner, with an AUC value of 1.0, indicating the nearly perfect discrimination between the non-churning and churning classes.



Evaluation Matrix

	Overall Accuracy	Validation Error	Sensitivity	Specificity	f1-score
	0.9983676134508651	0.00163238654913489	0.9991743876708559	0.9963751699139103	0.9971658542115407

5. Model Building: k-NN Classifier

In the k-NN classifier, the main idea is to use similar records in the training data to classify the new records from test data. The k-nearest neighbors are used used to determine similar records, and then the new record will be classified based on the majority class of k-nearest neighbors.

Advantages: - Simple method with easy implementation, with no parametric assumptions. - As the training is not required, new data can be added seamlessly without impacting the accuracy of the model.

Disadvantages: - It is very sensitive to large-scale dataset and outliers. - It is a little confusing when it comes to finding different distances if the data is large.

Implementation: - k-NN classifier was run over 2(k=2) and 5(k=5) iterations. Distance metrics ‘minkowski’ distance and weight metrics ‘uniform’ has been used in each iteration as an additional argument for obtaining the optimal value of k in k=5.

Performance Evaluation: k-NN Classifier

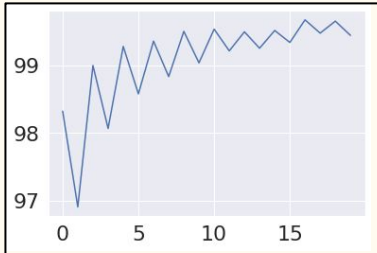
A grid-search was performed to obtain the best metrics k = 5, distance = ‘Minkowski’ and weight = ‘uniform’, with the highest accuracy of 99.24%.

The sensitivity value of 100% indicates that it correctly classified the True Positives.

The Specificity of 97.37% indicates that it correctly classified the True Negatives or the churning instances.

A high F1-score of 94.3% indicates low False Positives and low False Negatives, hence correctly identifying non-churning instances.

For, the value of K=1, the achieved accuracy is 96.89%, with 99.98% sensitivity and 89.28% specificity. The ROC curve was closer to the top-left corner for both K values, with a high AUC value of 0.99 (K=5) and 0.95(K=1), indicating nearly perfect discrimination between the non-churning and churning classes.



Evaluation Matrix

	Overall Accuracy	Validation Error	Sensitivity	Specificity	f1-score
	0.9689193601044728	0.03108063989552723	0.9998165305935235	0.8926144086995922	0.9430349449497366

Final Results

	Model	Overall Accuracy	Validation Error	Sensitivity	Specificity	f1-score
1	Logistic Regression	1.0000	0.0000	1.0000	1.0000	1.0000
2	Neural Network 1	0.9445	0.0546	1.0000	0.8106	0.8954
3	Neural Network 2	0.7777	0.2234	0.9951	0.2367	0.3792
4	CNN	0.7868	0.2110	0.9866	0.3011	0.4513
5	Decision Tree (Gini Index)	0.7777	0.0099	0.9989	0.9685	0.9826
6	Decision Tree (Entropy)	0.9988	0.0016	0.9992	0.9964	0.9972
7	KNN (K=1)	0.9872	0.0311	0.9998	0.8926	0.9430
8	KNN (K=5)	0.9970	0.0072	1.0000	0.9751	0.9874

	Model	Training Error	Test Error
1	Logistic Regression	0.0000	0.0000
2	Neural Network 1	0.0555	0.0546
3	Neural Network 2	0.2223	0.2234
4	CNN	0.2132	0.2110
5	Decision Tree (Gini Index)	0.2223	0.0099
6	Decision Tree (Entropy)	0.0012	0.0016
7	KNN (K=1)	0.0128	0.0311
8	KNN (K=5)	0.0030	0.0072

Thank You!