Forecasting Churner in Telecommunication Organization

Master of Engineering (Industrial Engineering)
Chulalongkorn University

Course
Research Problems in IE I (Machine Learning)
(2104691)

Instructor
Asst.Prof. Nantachai Kantanantha, Ph.D.

By
Pongpisut Kongdan (6370175121)

Table of Contents

Table of Figures

Table of Tables

1. Problem Statement

In today telecommunication industry, there is a huge competition, one of the challenges in the telecommunication industry is the retention. Industry is struggling on maximizing profit from acquiring a new customer, which cost company a lot more when compare with retaining the existing customers; Therefore, for the company to increase its profitability,

Customer churn (or customer attrition) is a tendency of customers to leave a brand and stop being client of certain business, which churn rate is the percentage of leaved customer divided by certain period of time. Churn rate is a health indicator for businesses, and in this study, we will apply similar prediction to churn rate, by using advanced analytics to predict consumer behavior that will associate with customers choices whether they will cancel service with the company, which called customer churn prediction.

In general, it's the overall customer experience that defines brand perception and influences how customers recognize value for money of products or services they use, the reality is that even loyal customers won't withstand a brand if they've had several issues with it, therefore if we predict that particular customer are having tendency to leave our service, so we can reach them in time by offering reasonable pricing as a second chance to prove our service value, to lower chance of churn.

2. Data Preparation

The dataset contains customer level information for a telecom company. Various attributes related to the services used are recorded for each customer.

Variables (Attributes)

Features	Description	Type
Account Weeks	Number of weeks of activated account	Quantitative - Discrete
Contract Renewal	1 if customer recently renewed contract, 0 if not	Qualitative - Binary
Data Plan	1 if customer has data plan, 0 if not	Qualitative - Binary
Data Usage	Gigabytes of monthly data usage	Quantitative - Continuous
Service Calls	Number of calls into customer service	Quantitative - Discrete
Day Mins	Average daytime minutes per month	Quantitative - Continuous
Day Calls	Average number of daytime calls	Quantitative - Discrete
Monthly Charge	Average monthly bill	Quantitative - Continuous
Overage Fee	Largest overage fee in last 12 months	Quantitative - Continuous
Roam Mins	Average roaming minutes per month	Quantitative - Continuous

Table (2.1): Attributes and Description

Predict variable (desired target)

Churn - 1 if customer cancelled service, 0 if not (Quantitative – Binary)

The dataset is randomly

- Training data (2,000 records – 60%)
- Cross validation data (667 recodes – 20%)
- Test data (666 records – 20%).

Exploration on dataset

Before moving any further, let us check for descriptive mate of our dataset

	Training Data set										
	Account Weeks	Contract Renewal	Data Plan	Data Usage	Service Calls	Day Mins	Day Calls	Monthly Charge	Overage Fee	Roam Mins	Churn
Count if 0 in %	-	-	-	-	-	-	-	-	-	-	86.50%
Count if 1 in %	-	-	-	-	-	-	-	-	-	-	13.50%
Average if 0	100.46	0.94	0.30	0.88	1.45	175.25	100.14	56.05	9.96	10.15	-
Average if 1	103.69	0.71	0.17	0.56	2.25	208.05	101.69	59.56	10.63	10.58	-
Max if 0	243.00	1.00	1.00	4.75	7.00	309.90	158.00	108.60	17.58	18.40	
Max if 1	225.00	1.00	1.00	4.59	9.00	346.80	165.00	104.90	18.19	17.90	
Min if 0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	16.00	0.00	0.00	
Min if 1	12.00	0.00	0.00	0.00	0.00	46.50	44.00	19.00	3.77	3.90	
Sd	39.34	0.29	0.45	1.28	1.28	54.06	19.68	16.33	2.50	2.81	

	Cross Validation Dataset										
	Account Weeks	Contract Renewal	Data Plan	Data Usage	Service Calls	Day Mins	Day Calls	Monthly Charge	Overage Fee	Roam Mins	Churn
Count if 0 in %	-	-	-	-	-	-	-	-	-	-	83.06%
Count if 1 in %	-	-	-	-	-	-	-	-	-	-	16.94%
Average if 0	99.87	0.93	0.24	0.73	1.61	181.45	99.88	55.87	10.14	10.31	
Average if 1	98.43	0.88	0.27	0.84	1.35	182.38	102.58	56.36	9.69	10.93	
Max if 0	232.00	1.00	1.00	4.64	7.00	307.10	163.00	110.00	16.86	18.90	
Max if 1	224.00	1.00	1.00	5.40	7.00	322.30	148.00	110.00	17.00	20.00	
Min if 0	3.00	0.00	0.00	0.00	0.00	17.60	36.00	16.00	1.56	0.00	
Min if 1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	14.00	3.55	2.00	
Sd	40.20	0.27	0.43	1.26	1.34	53.25	20.70	16.37	2.58	2.79	

	Test Dataset										
	Account Weeks	Contract Renewal	Data Plan	Data Usage	Service Calls	Day Mins	Day Calls	Monthly Charge	Overage Fee	Roam Mins	Churn
Count if 0 in %	-	-	-	-	-	-	-	-	-	-	84.98%
Count if 1 in %	-	-	-	-	-	-	-	-	-	-	15.02%
Average if 0	103.18	0.88	0.27	0.79	1.55	177.87	101.05	55.55	9.96	10.16	
Average if 1	103.24	0.84	0.35	0.97	1.67	179.51	98.77	58.68	10.58	10.05	
Max if 0	215.00	1.00	1.00	4.73	8.00	315.60	158.00	111.30	18.09	18.20	
Max if 1	208.00	1.00	1.00	3.89	9.00	350.80	156.00	101.80	17.55	17.50	
Min if 0	1.00	0.00	0.00	0.00	0.00	7.90	47.00	15.70	2.80	0.00	
Min if 1	2.00	0.00	0.00	0.00	0.00	62.60	42.00	26.00	4.69	5.70	
Sd	40.85	0.33	0.45	1.25	1.38	56.86	20.59	16.77	2.61	2.73	

Table (2.2): Average Values of Attributes and Percentage of Accounted Churn for Training Data Set

From the Figure above we can see our dataset have separated into three set of data with 13.50%, 16.94% and 15.02% of class 1 (cancelled service) in training data, cross validation data and test data, respectively; and predicts 0 have accounted for of 86.50%, 83.06% and 84.98% for training data, cross validation data and test data, respectively. This is particularly important information for evaluate the model we are going to construct, because this means that all the model will predict 1 for at least 13.50%, 16.94% and 15.02%, and 0 for at least 86.50%, 83.06% and 84.98%.

The average values of each attributes of churn and non-churn are not much differed from each other, this can translate to skewed dataset, however we have some degree of difference on service calls and data usage, which could be positive for our construction of prediction model.

For minimum values and maximum is found to be useless, since I have done some investigation and found out that we have some outlier in the dataset which consider it effect to be truly little because of large example data compare to the outlier.

From the conclude from the exploration that this dataset is skewed dataset.

Data Exploration by Visualization

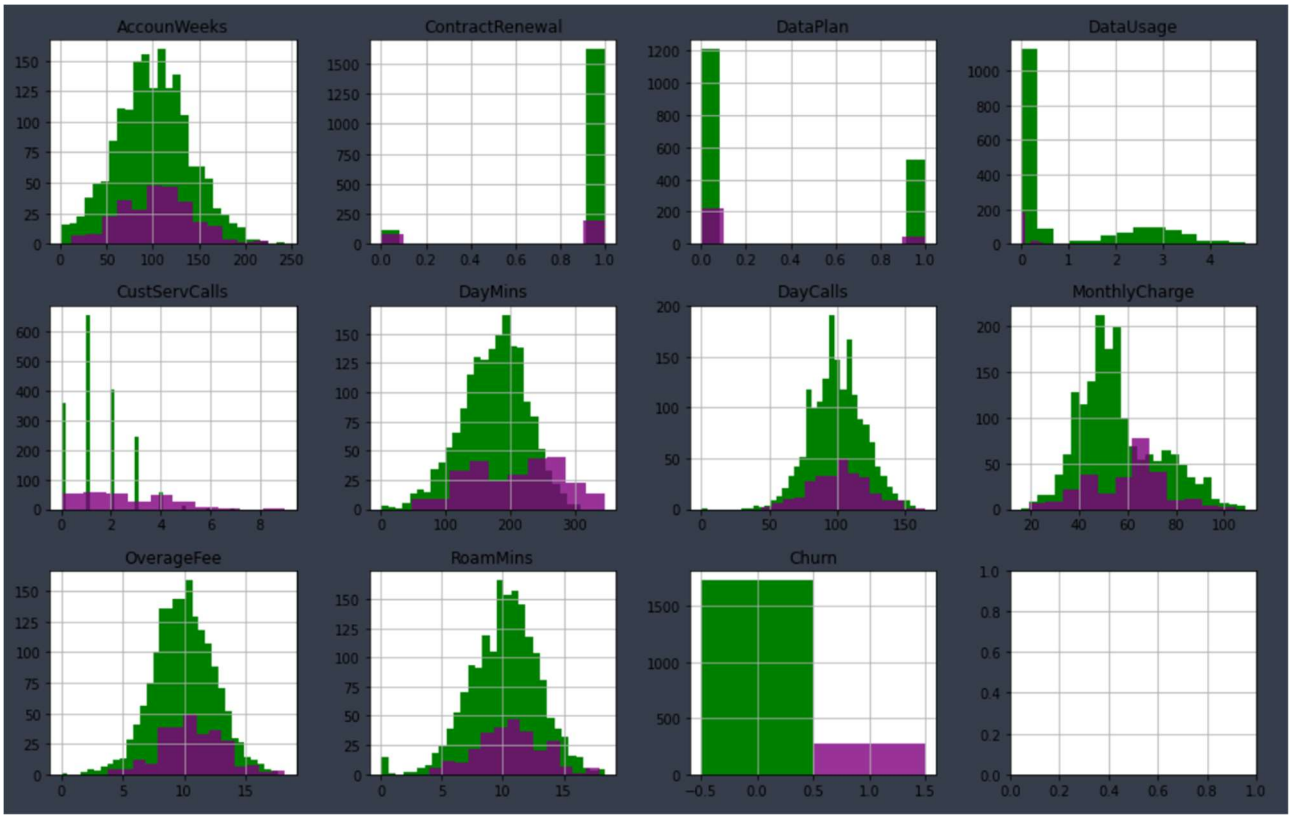

Figure (2.1): Visualization of data distribution on Training Dataset

The histogram above shows the distribution of each attributes in our data, the green color is the non-churn customers (1) and the purple color is the churners (0). The graph has showed that most of the distribution of our data is normal distributed for both churn and non-churn customers. And the green one is more distinguish because 86.50 of our customers are not yet to leave company services or in the others term again skewed dataset.

3. Methodology

In the machine learning world, we already have many persistent algorithms that are commonly used to predict customer attrition, such as logistic regression, random forest, decision tree, neural network, and more.

Logistic regression is an algorithm that usually uses binary classification problems (which ours is binary) by measuring each feature's relationship. However, we will have to be careful with the accuracy, when constructing the model with many features as an input, which can have many relationships that are too nuanced for logistic regression ho handle.

To evaluate the effectiveness of all our models, we will use precision, recall, and F1 score. Precision will determine the proportion of positive that were correctly identified while recall will give us the proportion of real positives identifications. Considering the problem, we have in this study, which is skewed, the precision, recall, and F1 score as its objective should be the best match to evaluate our models.

Develop the forecasting models to predict whether the customer will leave the company (i.e., cancel service) using the training data. After that, we will compare the models using the cross-validation data to select the best model. Finally, we will determine the generalization error of the selected model using the test data.

3.1. Logistic Regression Algorithm

```
data = np.loadtxt(os.path.join('Data', 'Training delima.csv'), delimiter=',')
data_C = np.loadtxt(os.path.join('Data', 'Cross delima.csv'), delimiter=',')
X, y = data[:, [0,3,4,5,7,9]], data[:, 10]
y = y.astype(int)
m = y.size
X_C, y_C = data_C[:, [0,3,4,5,7,9]], data_C[:, 10]
y_C = y_C.astype(int)
```

Figure (3.1.1): Line of code for importing Logistic dataset

Some of the features are linearly dependent on each other; therefore, those features got deleted and used the remaining features to construct our logistic regression are Account Weeks, Data Usage, Cust Service Calls, Day Mins, Monthly Charge, and Roam Mins.

In this construction of the model, we will use logistic regression with its sigmoid function and the chosen features will be normalized to obtained it close range of value to construct the logistic regression model. And for training the model, we will use scipy.optimize.

Regularization terms will be implemented on the model to reduce the value of each theta, this for avoiding overfitting by penalizing each feature, and finally, we will compare the non-regularized model with a regularized model. We will use iterations of 200 for both non-regularization and regularization and lambda = 1 for regularizing terms.

At the end of the logistic regression part, we will use the plotting of the learning curve to diagnose the model's high bias and high variance condition, and other diagnose methods we will use precision, recall, and F1 score to test with the cross-validation set the, since dataset we have is skewed.

3.2. Support Vector Machine

```
data = np.loadtxt(os.path.join('Data', 'Training delima.csv'), delimiter=',')
X, y = data[:, [0,3,4,5,7,9]], data[:, 10]
y = y.astype(int)
m = y.size

data_C = np.loadtxt(os.path.join('Data', 'Cross delima.csv'), delimiter=',')
X_C, y_C = data_C[:, [0,3,4,5,7,9]], data_C[:, 10]
y_C = y_C.astype(int)
```

Figure (3.2.1): Line of code for importing SVM dataset

The features we are going to use in support vector machine will be the same as the logistic regression model since the support vector machine is another alternative algorithm to use along with logistic regression model; therefore, we can use this model to compare with previous of our logistic regression model we have computed.

The kernel we are going to use in our support vector machine is Gaussian Kernel to make our decision boundary since we have the numbers of training data more than 1,000; therefore, Gaussian kernel will be more appropriate than the linear kernel,

The value of C and Sigma are 1 and 0.1 respectively, which have been obtained from several trials and error method, in additional Large C can influent the model to have a lower bias, higher on variance and vice versa, Large Sigma will influent the model to have higher bias and lower on variance and vice versa.

In the last step, we will use precision, recall and F1 score using cross-validation set, same as linear regression due to the skewness of the dataset.

3.3. Neural Network

For the neural network, we are going to use all the feature we have since the neural network has some degree of reputations of handling skewed data and a big set of input-data.

We will try using some hidden layers here which is 5, 10, 15, 20, 25, 50,70,100,200. Lambda to use in this model will be 0.1, 0.2, 0.3, 0,4, 0.5, 0.7, 1, 1.5, 2, 2.5, and 3. The iterations for the model is 50000, and for the activation function, we will use sigmoid. For training the data, we will use scipy.optimize.

All the features will be normalized to obtain it optimal value to construct the neural network model and increase computation speed.

Since we do not have the initial theta, we must randomly initialize the theta, but all of the initial theta must have different values. Moreover, the make sure that our cost function did not decrease for each iteration of backward propagation we will use "utils.checkNNGradients (nnCostFunction, lambda_)" for it application(which is function obtained from the class), also to confirm that implementation of forwarding propagation and backward propagation is correct.

We should obtain the best value of the hidden layers and lambda for the prediction model with several trial and error.

Finally, since the dataset has a default of 86.50 percent of churner, which made this data is skewed Classes, we will always obtain high accuracy and low error, we will use precision, recall and F1 score to diagnose the error and usefulness of the model by validation of the F1 on the cross-validation set.

4. Result and discussion

4.1. Logistic Regression

First, we have normalized X to obtain the value with less difference, since the gradient descent will perform faster with a close-range of numbers, therefore ensuring a better chance of converging to the global minimum point.

Secondly, we initialize the first theta parameter to compute with our regression model, along with obtaining initial gradient descent after computation of cost function. Here is the value we obtain after running the code.

```
Cost at initial zero theta parameter:
 0.6931471805599451
Gradient at initial zero theta parameter:
 [ 0.365      -0.009572    0.02889497 -0.07276916 -0.07084674 -0.02514601
 -0.01759494]
```

Figure (4.1.1): Initial Theta for Logistic Regression Model

Next step is to compute the value for both with regularization terms and within out regularization terms, and the results are showed as below.

```
Cost after gradient descent:
 0.3388584368120874
Theta after gradient descent:
 [-2.225658    0.1151636  -1.35179208  0.62372785 -0.011583    1.27379919
  0.24223529]
```

Figure (4.1.2): Cost Function and Theta for Non-regularization Model

```
Training Accuracy: 87.05 %
Cross Validation Accuracy: 83.81 %
```

Figure (4.1.3): Prediction Accuracy for Non-regularization Model

```
Cost after gradient descent:
 0.33973460469502464
Theta after gradient descent:
 [-2.21277783  0.11266251 -1.21162633  0.6176173   0.07790175  1.10970901
  0.23728492]
```

Figure (4.1.4): Cost Function and Theta for Regularization Model

```
Training Accuracy: 86.90 %
Cross Validation Accuracy: 83.21 %
```

Figure (4.1.5): Prediction Accuracy for Regularization Model

From the figure above we can conclude that both the model with and without regularization terms results pretty much the same in terms of cost function values(0.338 versus 0.339), final theta, Training accuracy(87.05% versus 86.50%), and Cross-validation accuracy(83.81% versus 83.21%), however, both of the accuracies was not the good number since the non-churn customers already accounted for 86.50, therefore only 0.55% and 0.40% improvement in the training set for regularization model and non-regularization model respectively. In addition, changing lambda does not help increase prediction accuracy (found out by several trial and error).

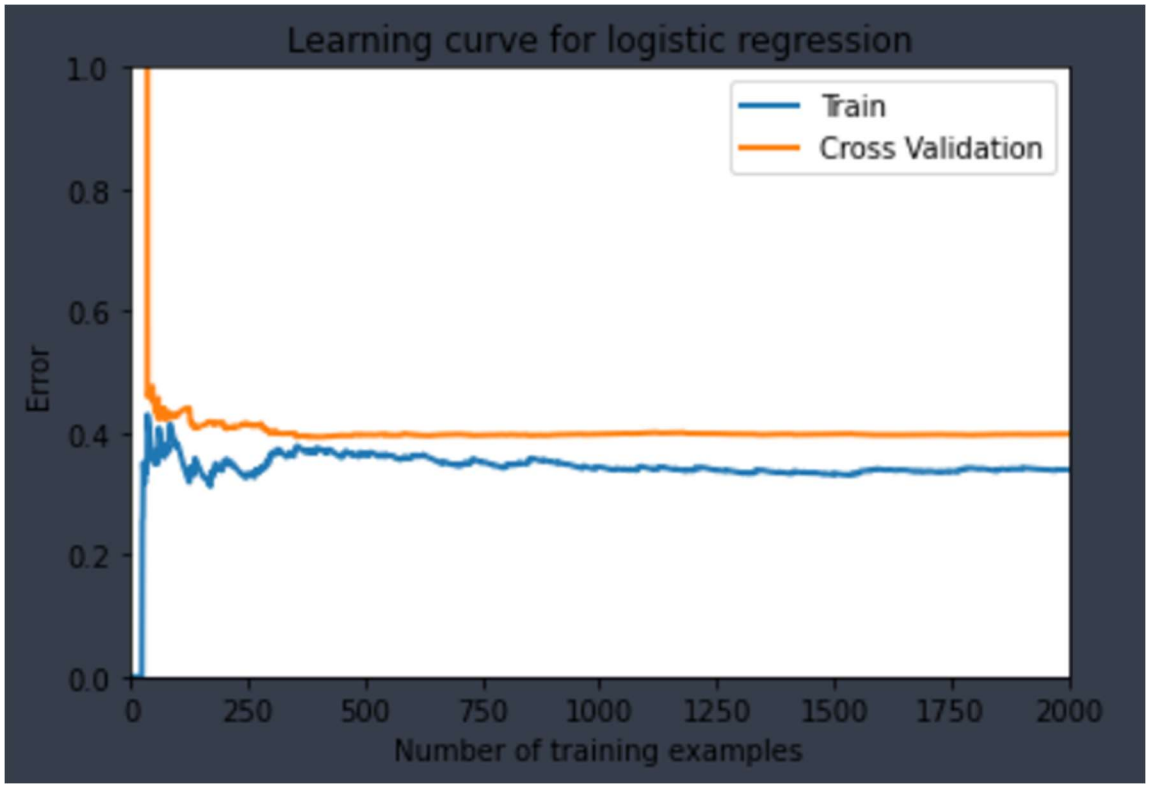

Figure (4.1.6): Learning curve for non-regularization model

From the learning curve, the errors were low for both small training examples, and as it increased, this means we can rely on this prediction accuracy without suffering on high bias or high variance problem, but due to low improvement in prediction accuracy; therefore, this prediction model is not recommended.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.838081	0.692308	0.079646	0.142857

Figure (4.1.7): The F1 score for Logistic Regression

The Recall also incredibly low because the model is prone to predict 0 rather than 1, this also result in low F1 score, therefore the model also not recommended according to this diagnostic method.

4.2. Support Vector Machine

```
Training Accuracy: 100.00
Cross Validation Accuracy: 83.06
                   Model  Accuracy  Precision  Recall  F1 Score
0  Support Vector Machine  0.830585        0.0     0.0       0.0
```

Figure (4.2.1): Result for SVM algorithm

The result happens to be overfitting (high variance) with prediction accuracy of 100% and no sight of improvement for cross-validation set (from 83.06%), after some investigation, it appears that the overfitting problem is huge, in the degree that the model always predicts 0 or non-churn for other datasets, this also made true positive equal to zero, therefore precision, recall, and F1 score equal to zero.

Trying difference C and Sigma also does not help, lowing did solve the overfitting problem but turn out that the support vector machine always predicts 0 even with training dataset. It seems that the support vector machine cannot hold this degree of skewness of this dataset, despite the reputation of handling complex non-linear function. In concluded the support vector machine does not recommend for this prediction.

4.3. Neural Network

The result of Neural Network Model computed accordingly to hour preset parameters are showed as below:

```
Initializing Neural Network Parameters ...
0.351428082185596
Training Set Accuracy: 94.900000
Cross Validation Set Accuracy: 91.454273
            Model  Accuracy  Precision    Recall  F1 Score
0  Neural Network  0.914543   0.841463  0.610619  0.707692
```

Figure (4.3.1): Result of Neural Network Model

After several trial and error with the different preset parameters, we finally got the best value of hidden layers and lambda for regularization terms, which is 10 and 0.3, respectively. During the trial and error, some output computation indicates that if the hidden layer more than 25 prone to cause overfitting (high variance) condition to the model and the lambda beyond 1 gave the lower value of prediction accuracy(underfitting) because of an excessive penalized multiplier.

From final computation using hidden layers = 10 and lambda = 0.3, the cost function we obtain is 0.351 which a little higher than logistic regression, however the prediction accuracy is tremendously improved comparing with logistic regression. The accuracy of this neural network is 8.4% (94.90%-86.50%) and 8.39% (91.45%-83.06) for the training set and the cross-validation set, respectively.

For the F1 score is the best of all model we have which is 0.70 with the close range of precision and recall (0.841 and 0.610) when test with cross validation set, which mean that this model can predict both 0 and 1 accurately and able to handle skewed datasets.

4.4. **Final model**

It is appearing the best model is the Neural Network with sigmoid function as it activation function, as we presume, and here is the result of the final test which is the prediction accuracy on the test dataset with 91.14% accuracy, which is acceptable.

Figure (4.4.1): Final Test Result

5. Conclusion

We have done computing every model of machine learning algorithm to verify that which one is the most suitable algorithm to predict this dataset, but and confirmed that logistic regression and support vector machine is not capable of predicting high skew dataset with many numbers of features contained, however in this project we have constructed the model according to preset parameters which we have defined at the beginning, by adjusting some parameter also could improve model efficiency. For example, in support vector machine algorithm we have huge overfitting value, by trying smaller set of features or getting more training example could help solve this problem, and for logistic regression algorithm trying to use pair of features which have the most influent churn could also increase it accuracy, maybe use difference pair of features in our dataset or acquiring new pair features that not yet to be collected

The neural network model we obtained can help organizations identify potential churners in advance, allowing the organization to craft strategies to prevent customers from leaving the service. With this prediction, the organizations can offer incentives, like discounts or loyalty programs, or provide additional services to reduce the churn rate. Without this tool, the prediction would be acting based on assumptions alone, not a data-driven method reflecting real customer behaviors.

However, the algorithm we implement in this project is not the only algorithm that persists in the machine learning world; therefore, it is most likely for others to perform better than the neural network for supervised learning algorithms and an unsupervised learning algorithm. Further study must be conducted for a better applicable prediction model for churner, and it can benefit both academically and real-life applications.

Reference

- *Teaching Material Provided by Asst.Prof. Nantachai Kantanantha, Ph.D. during Research Problems in IE I (Machine Learning) course, Industrial Engineering, Chulalongkorn University.*
- *https://www.relataly.com/anyone-about-to-leave-predicting-the-customer-churn-of-a-telecommunications-provider/2378/*
- *https://randerson112358.medium.com/predict-customer-churn-using-python-machine-learning-b92f39685f4c*
- *https://www.kdnuggets.com/2019/05/churn-prediction-machine-learning.html#:~:text=One%20of%20the%20ways%20to,churn%20rate%20is%202%20percent.*
- *https://towardsdatascience.com/churn-prediction-with-machine-learning-c9124d932174*
- *https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0191-6*
- *https://docs.scipy.org/doc/scipy/reference/optimize.html*
- *https://www.relataly.com/anyone-about-to-leave-predicting-the-customer-churn-of-a-telecommunications-provider/2378/*
- *https://pypi.org/project/scikit-learn/*

Appendix

Example of Training Dataset

AccounWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls	MonthlyCharge	OverageFee	RoamMins	Churn
66	1	0	0	4	154	133	44	9.95	9.5	1
43	0	0	0	0	251.5	105	61	10.64	9.3	0
37	1	0	0	3	191.4	116	47	8.37	14	0
148	1	1	3.32	1	252.9	129	101.2	14.22	12.3	0
87	1	1	2.57	3	82.6	113	59.7	11.22	9.5	0
75	1	0	0.26	1	209.4	133	56.6	10.58	7.2	0
100	1	0	0	1	166	102	49	11.81	10.9	0
78	1	0	0	2	155	106	42	8.77	11.8	0
98	1	0	0	0	216.8	86	54	9.54	11	0
72	1	0	0.28	1	147	79	41.8	8.12	10.5	0
145	1	0	0	2	129.4	97	38	9.27	1.1	0
76	1	0	0	0	203.6	61	49	8.09	8.4	0
66	1	1	2.16	2	229.4	104	82.6	12.87	8	0
158	1	0	0	2	222.8	101	56	10.15	6.9	0
75	0	0	0	1	222.4	78	66	16.35	8.7	1
149	1	1	3.05	2	264.4	102	94.5	10.98	11.3	0
12	1	0	0	1	249.6	118	64	12.62	11.8	1
121	1	1	2.73	4	237.1	63	85.3	10.28	10.1	0
51	1	0	0	1	197.8	60	53	11.05	8.9	0
54	0	0	0	1	214.1	77	57	12.03	10.1	0
121	1	1	2.03	3	218.2	88	87.3	17.43	7.5	0
150	1	1	1.94	0	209.8	112	68.4	7.75	7.2	0
163	1	0	0	5	202.9	100	50	8.93	12.8	0
157	1	0	0	2	240.2	67	54	7.65	10.2	0
64	0	0	0.26	3	146.7	83	40.6	7.42	12.5	0
127	0	0	0	1	176.9	110	45	8.4	7.7	1
74	1	0	0	0	162.7	102	53	14.6	8.7	0
178	0	0	0	2	124.5	134	34	7.06	11.4	1
65	1	0	0.28	2	192	89	47.8	6.98	5.5	0
96	1	1	2.65	1	172.7	86	67.5	6.67	9.8	0
36	1	1	3.92	0	146.3	128	78.2	8.13	14.5	0
96	1	0	0	1	276.9	105	69	12.35	10.3	1
119	1	1	2.05	3	117.8	66	62.5	12.84	7.6	0
56	1	0	0	4	222.7	133	62	13.85	13.6	0
89	1	1	2.35	1	209.9	113	80.5	12.49	8.7	0
162	1	0	0	2	135.2	98	44	12.1	10.2	0
61	1	1	1.73	4	187.5	124	62.3	7.33	6.4	1
61	0	0	0	2	78.2	103	30	9.8	10.1	0
106	1	0	0	1	158.7	74	33	3.22	10.2	0
117	1	0	0	0	239.9	84	56	8.74	9.8	0
128	1	1	2.92	0	222.9	136	90.2	13.1	10.8	0
64	1	0	0	2	174.5	98	45	9.01	10.7	0
11	1	1	2.7	2	131.5	98	69	11.51	10	0
81	1	0	0	3	183.6	116	45	7.63	12.2	0
43	1	0	0	1	84.2	134	22	4.04	10.8	0
155	1	0	0	2	163	93	46	10.2	15.1	0
78	1	0	0	1	191.7	122	54	12.07	9.1	0
90	1	0	0	3	113.2	108	36	9.47	14.1	0
65	1	0	0	1	213.4	111	57	11.73	2.7	0
168	1	0	0.37	2	183.2	131	50.7	8.96	9.9	0
111	1	0	0	3	99.3	112	40	13.53	9	0
176	1	0	0.32	1	201.9	101	51.2	7.74	9	0
136	1	0	0	1	250.2	121	66	13.36	13	1
9	1	1	3.59	1	193.8	130	86.9	10.13	13.3	0

Example of Cross-Validation Dataset

AccounWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls	MonthlyCharge	OverageFee	RoamMins	Churn
103	1	0	0	0	160.2	104	40	6.95	9.7	0
101	1	1	2.57	2	219.7	137	79.7	9.34	9.5	0
119	1	0	0.23	2	154.5	129	45.3	9.68	13.4	0
154	0	0	0.36	0	145.9	69	46.6	10.41	14.4	1
96	1	0	0	3	150	122	45	10.93	9.8	0
129	1	1	3.08	1	211	99	80.8	7.76	11.4	0
64	1	0	0	1	224.8	111	55	9.5	9.2	0
133	0	1	4.16	2	221.1	137	102.6	13.25	15.4	1
67	1	0	0	2	109.1	134	31	7.12	10.9	0
72	1	0	0.2	2	141.3	133	38	6.75	11.2	0
97	1	1	2.24	3	121.1	105	65.4	13.01	8.3	0
90	1	0	0.22	2	222	93	56.2	9.35	12.4	0
13	1	1	3.05	2	176.6	65	75.5	8.64	11.3	0
73	1	0	0	1	217.8	91	56	11.03	10.3	1
182	1	0	0.36	2	104.9	111	38.6	9.93	8	0
113	1	0	0	3	92.6	85	31	8.88	14.4	0
28	1	0	0	3	236.8	102	55	8.36	9.7	0
50	1	0	0	5	131.1	129	36	8.03	5.6	1
86	1	0	0.34	1	166.2	112	53.4	12.77	5.4	0
101	1	0	0	0	124.8	66	44	12.86	13.4	0
104	1	0	0	1	156.2	93	43	9.65	13.1	0
40	1	1	2.03	1	224.7	69	70.3	6.73	7.5	1
168	1	0	0	2	163.4	134	49	12.01	11.6	1
21	1	0	0	0	146	78	35	5.49	6.8	0
170	1	0	0	2	184.1	106	49	10.25	9.8	0
212	1	0	0.23	2	226	127	67.3	15.23	12.6	1
169	1	0	0	0	100.8	112	37	11.5	9.5	0
232	1	0	0.14	1	165.6	104	46.4	9.8	11.8	0
63	1	0	0.27	0	164.5	75	43.7	7.4	11.2	0
124	1	0	0	4	143.3	120	44	11.54	7.8	1
72	1	0	0.28	0	196.5	88	49.8	7.93	6.8	0
159	0	0	0.44	1	257.1	53	75.4	15.61	8.8	1
85	1	0	0	3	255.3	114	60	9.73	3.7	0
72	1	0	0.27	0	272.4	88	58.7	5.4	12.7	0
86	1	1	3.48	1	225.4	79	89.8	9.36	12.9	0
120	1	0	0	1	149.2	98	42	9.68	11.1	0
132	1	0	0	1	121.5	88	43	12.65	10.7	0
120	1	0	0.3	1	150.6	85	39	5.95	6.4	0
106	1	0	0	2	119.2	142	40	11.42	8.4	0
37	1	0	0	0	191.1	69	44	6.46	12.9	0
44	1	1	3.78	0	221.8	105	89.8	8.09	14	0
12	1	0	0	2	204.6	98	53	10.63	9.8	0
113	1	0	0	2	128.7	100	42	11.36	9.2	1
101	1	0	0	1	174.9	105	52	13.1	8.5	0
72	1	0	0.42	3	118.2	106	39.2	8.36	12.2	0

Example of Testing Dataset

AccounWeeks	ContractRenewal	DataPlan	DataUsage	CustServCalls	DayMins	DayCalls	MonthlyCharge	OverageFee	RoamMins	Churn
40	1	0	0	4	169.7	115	41	7.07	10.5	1
43	1	0	0	1	241.9	101	53	6.47	5.9	0
120	1	1	3.13	0	299.5	83	96.3	8.17	11.6	0
122	1	1	2.32	2	204.5	92	70.2	6.98	8.6	0
113	1	0	0	2	159.8	143	46	10.51	13.1	0
88	1	0	0	3	138.3	116	44	11.8	9.6	0
124	1	0	0	2	151	98	36	6.03	9.2	0
130	1	1	3.29	1	174.5	120	81.9	10.88	12.2	0
134	1	1	2.92	1	142.9	105	61.2	4.43	10.8	0
91	1	0	0	1	133.8	61	37	7.94	10.5	0
114	1	1	3.08	1	206.2	79	88.8	13	11.4	0
76	1	0	0.38	1	90.5	142	37.8	10.59	9.3	0
98	1	0	0	1	169.9	77	41	6.92	8.5	0
162	1	0	0	1	184.5	118	51	11.2	11.6	0
67	1	0	0	2	152.5	131	48	12.62	4.9	0
36	1	0	0	1	175.1	144	49	10.85	9.9	0
36	1	1	2.32	2	29.9	123	40.2	6.46	8.6	0
32	1	0	0	2	164.8	98	48	11.5	14.8	0
83	1	0	0	1	132.4	120	33	6.08	8.6	0
127	0	0	0.32	2	247.5	99	55.2	5.43	10.6	0
91	1	0	0	1	153	123	38	7.06	10.3	1
47	1	0	0	3	155.3	116	43	9.41	12.3	0
81	1	1	3.16	3	115.9	120	71.6	11.83	11.7	0
119	1	0	0	0	294.2	100	70	11.63	9	1
105	1	0	0	2	166.1	93	44	8.8	16.2	0
97	1	0	0	1	151.6	107	39	7.77	14.7	0
72	1	1	1.57	1	92.8	98	54.7	13.56	5.8	0
142	0	1	2.65	0	191.1	109	72.5	7.48	9.8	0
143	1	0	0	3	160.4	120	52	14.3	6.9	0
115	1	1	1.51	2	133.3	110	54.1	9.29	5.6	0
87	0	0	0	1	167.3	119	46	9.93	11	0
103	1	0	0	6	174.7	151	43	7.4	15.8	1
30	1	1	1.78	0	217.4	74	73.8	10.69	6.6	0
163	1	1	3.32	2	231.9	56	91.2	10.59	12.3	0
90	1	0	0	1	203.4	146	54	11.34	7.3	0
37	1	0	0.3	4	221	126	58	10.23	6.8	0
40	1	0	0.25	0	109.4	107	42.5	12.24	7.1	0
103	1	0	0	1	166.6	84	45	9.62	7.7	0
92	1	0	0	0	249.4	118	61	10.58	9.1	0
147	0	1	2.19	3	219.9	118	77.9	10.43	8.1	0
103	1	0	0.24	5	167.8	121	49.4	10.65	13	0
127	0	0	0.3	1	256.5	87	66	11.11	13	0
99	1	0	0	1	254.4	120	57	7.97	6	0
76	1	0	0	2	224.4	121	51	7.4	6.7	0
23	1	0	0	1	113.1	74	34	8.44	6.9	1