Abstract

In today’s age where players aggressively compete for greater customer mindshare, retaining customers and building brand loyalty is a common challenge for most enterprises. Customers tend to abandon brands that do not offer them timely services, products and value. However, sorting through voluminous customer data to understand trends, patterns and issues and develop proactive strategies for customer retention is almost impossible without robust digital support.

This paper analyzes how machine learning can be used to amplify existing CX platforms and drive customer retention. It evaluates the effectiveness of commonly used machine learning algorithms and explains why Support Vector Machine proved to be the best-fit among these. It also examines how Infosys developed, tested and fine-tuned this ML model and integrated it with leading SaaS CRM products including Oracle CX Cloud and SFDC to enable effective churn management.
Customer churn and its implications

According to an article published in Harvard Business Review, “Customer churn rate is a metric that measures the percentage of customers who end their relationship with a company in a particular period.”

Customer churn is a common problem in nearly every industry. Left unaddressed, it results in erosion of brand value, revenue and market share. In US alone, companies lose nearly US $83 billion every year due to customer churn and abandoned purchases. A Forrester report estimates that companies incur two to five times more cost to acquire a new customer rather than retain an existing one.

Many studies have shown that tackling churn can significantly improve a company’s operational KPIs. For instance, a mere 5% reduction in churn can improve profitability by at least 25%. A leading US telecom operator witnessed a 35% year-on-year (YoY) increase in net revenues when maintaining low churn rates. Clearly, the timely addressal of customer churn is of paramount importance to ensure profitability and long-term success.

While many companies rapidly adopt software as a service (SaaS) to improve customer experience (CX) journeys, it is also important for them to buttress these solutions with new use cases that can amplify SaaS capabilities. One such use case for customer churn management is leveraging machine learning to rapidly amplify the capabilities of CX. This can deliver faster returns on CX investments and ensure positive financial growth for the enterprise.

How machine learning can reduce churn

Machine learning is a proven technology that provides accurate, precise and predictive business insights that were previously unavailable. For subscription/usage-based businesses like insurance, telecom or digital content providers, managing customer churn is a looming concern. For example, a US-based media services provider saw a fivefold jump in their subscriber base from nearly 30 million in 2012 to almost 150 million in 2018. Considering such tremendous growth in the subscriber base and the ensuing data, it is no surprise that enterprises struggle to protect their consumers while demonstrating YoY financial growth. Moreover, using traditional methods to identify customer behavioral patterns in order to retain them can be daunting and, in many cases, futile.

Machine learning algorithms can solve the challenge of sorting through massive volumes of historic data and identifying patterns by applying proven mathematical models and providing precise predictions. These help enterprises make the right decisions and draft relevant customer retention strategies.

According to Gartner, “The capability to transform data into actionable insight is the key to a competitive advantage for any organization. But the ability to autonomously learn and evolve as new data is introduced – without explicitly programming it to do so – is the holy grail of business intelligence. That’s what machine learning offers: a capability that accelerates data-driven insights and knowledge acquisition.”
A machine learning framework for churn management

Our machine learning framework helped us select the most optimal ML algorithm to tackle customer churn. The framework leverages data to predict possibility of churn and identify loyal customers. Using modern dashboards on both Oracle CX Cloud and SFDC, insights are made available on intelligent dashboards that amplify the capabilities of the cloud platforms.

**Fig 1: Key features of the framework**
1. Gather data

**Feature re-engineering** – In the real world, data is gathered, cleansed and feature engineered from disparate systems for consumption by ML models. For the Churn Management solution data was sourced from publicly available datasets on the Internet. The data was further divided into training set, cross validation set and testing set. A 60-20-20 ratio is an acceptable industry-standard split between these three datasets.

The first step of feature re-engineering involved the conversion of character data types to numerals. For example, the ‘State’ column consisted of categorical character values like OH, CA, NY, etc. Such categorical values were transformed using one hot encoding. This conversion was necessary since the underlying ML model is built on mathematical equations.

The second step involved feature scaling and mean normalization. Features in the churn dataset had different magnitudes. It was important to balance out the variances for mathematical reasons, particularly since ML algorithms perform better when features are of comparable magnitudes.

Mean normalization is a statistical technique used for feature scaling. It ensures that the mean of any feature is zero.

\[
x_{ij} = \frac{x_{ij} - \mu_j}{s_j}
\]

Here, \(x_{ij}\) is an instance of \(j^{th}\) feature and \(\mu_j\) is the mean of the \(j^{th}\) feature where \(j\) takes a value from 1 to \(n\) and \(i\) takes a value from 1 to \(m\).

\[
\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_{ij}
\]

\(s_j\) is the range (max – min) or standard deviation of all values of the \(j^{th}\) feature in the dataset. This approach was applied to columns based on their existing magnitudes.

Post feature re-engineering – After transforming the character fields and performing feature scaling and mean normalization, the data is ready to be fed into the ML model. Fig 3 shows the revised data file. It is easy to observe the transformation in the data values in Fig 3 compared to Fig 1.

**Table 1: Attributes/features used by the Infosys Churn Management solution**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Field Name</th>
<th>Feature Re-engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>State</td>
<td>☑ (Char to number transformation)</td>
</tr>
<tr>
<td>2</td>
<td>Account length</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Phone number</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>International plan (Yes/No)</td>
<td>☑ (Char to number transformation)</td>
</tr>
<tr>
<td>5</td>
<td>Voice mail plan (Yes/No)</td>
<td>☑ (Char to number transformation)</td>
</tr>
<tr>
<td>6</td>
<td>Voice mail message (Count)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Day minutes</td>
<td>☑ (mean normalization)</td>
</tr>
<tr>
<td>8</td>
<td>Day calls</td>
<td>☑ (mean normalization)</td>
</tr>
<tr>
<td>9</td>
<td>Day charges</td>
<td>☑ (mean normalization)</td>
</tr>
<tr>
<td>10</td>
<td>Evening minutes</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Evening calls</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Evening charges</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Night minutes</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Night calls</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Night charges</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>International minutes</td>
<td>☑ (Feature scaling)</td>
</tr>
<tr>
<td>17</td>
<td>International calls</td>
<td>☑ (Feature scaling)</td>
</tr>
<tr>
<td>18</td>
<td>International charges</td>
<td>☑ (Feature scaling)</td>
</tr>
<tr>
<td>19</td>
<td>Customer service calls</td>
<td></td>
</tr>
</tbody>
</table>

**Fig 2: Raw data file**

**Fig 3: Revised data file**
2. Train the model

In this step, we evaluated various machine learning models and identified the best model for the churn prediction solution. The training set was leveraged across different algorithms to train the model. The models were then adjusted and parameters selected in order to optimize them. The validation set was used to evaluate each optimized model based on a generalized dataset and compare its performance with other models. The best performing model was selected by the end of the validation stage. The test set was used to re-validate the final model based on a generalized dataset before being released for real-world deployment.

Logistic regression

Initially, the churn solution was built using the logistic regression algorithm. This algorithm predicts the group to which the object being observed belongs to. For example, in a churn scenario, the object would be either a churned customer or a continuing customer. This type of classification is known as binary classification.

Two models of logistic regression, linear and polynomial model, were tested. The quadratic polynomial model demonstrated better accuracy and was chosen as the right model for logistic regression.

The computational steps involved in the model development are listed below:

- Initialized vector \( \theta \) with a random value
- Calculated the hypothesis for all rows of the training set, i.e., find \( h_{\theta}(x) \). The hypothesis \( h_{\theta}(x) \) was obtained using the sigmoid function. Sigmoid function is a mathematical function that can take any real value and map it between the range of 0 and 1

\[
S(x) = \frac{1}{1 + e^{-x}}
\]

- Calculated the cost of the model at initial \( \theta \) using the following cost formula.

\[
J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left( y^{(i)} - \frac{1}{1 + e^{-\theta^T x^{(i)}}} \right)^2
\]

Since churn is a classification problem (see Notes: point A), we chose three different classification algorithms to evaluate the best fit. The three algorithms were logistic regression (linear and polynomial), Support Vector Machine (SVMs) and neural networks. Prediction accuracy, precision, recall, and F1 scores were used as KPIs for choosing the best model.

For each model, churn parameters/features were charted in a two-dimensional matrix called \( X \). This contained the data from each column of the input file. The actual values (in this case, churn indicator) containing the labels were represented as vector \( Y \).

A hypothesis \( h_{\theta}(x) \) was formulated using matrix \( X \) and vector \( \theta \) constrained by a cost function \( J(\theta) \). The cost function was minimized over the dataset to obtain an optimum hypothesis.

**Equation 1:**

\[
X = \begin{bmatrix}
    x_{11} & x_{12} & \cdots & x_{1n} \\
    x_{21} & x_{22} & \cdots & x_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{m1} & x_{m2} & \cdots & x_{mn}
\end{bmatrix}, \quad Y = \begin{bmatrix}
    y_1 \\
    y_2 \\
    \vdots \\
    y_m
\end{bmatrix}, \quad \theta = \begin{bmatrix}
    \theta_1 \\
    \theta_2 \\
    \vdots \\
    \theta_n
\end{bmatrix}
\]

\[
h_{\theta}(x) = z(\theta^T x), \text{ where } z \text{ is a function of } X \text{ and } \theta.
\]

\[
\min J(\theta) = Cost(h_{\theta}(x), Y)
\]
The cost function for logistic regression has to be a convex function that converges to a global minimum for the optimum value of $\theta$. Equation 3 for the cost function is derived from two cost functions as shown below:

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

- Optimized the value of $\theta$ over multiple iterations so that the cost of the model was minimal for the training set. This was done using an existing optimization library available in Octave
- Calculated the hypothesis for the validation set using $\theta$ from the earlier step
- Compared the hypothesis values with the labels for the validation set and calculated the precision, recall and F1 scores over the validation set
- Re-validated the model over the test set and calculated the precision, recall and F1 scores

The Receiver Operating Characteristic (ROC) curve was found to be reasonably good as shown in Fig 6.

**Support Vector Machine**

Support Vector Machine (SVMs) is another algorithm that is commonly used for classification problems. It classifies the dataset by finding a hyperplane (a line in two dimensions) that separates the positive labels from the negative labels with the widest distance between them. This hyperplane is called the decision boundary. It is drawn in an imaginary $n$ dimensional space where $n$ is the number of features in the dataset.

For our churn scenario, we used a popular SVM library ‘LIBSVM®’ to build our model. The software solves the following optimization problem for a given training set of instance-label pairs $(x_i, y_i)$ where $i=1$ and $x_i \in R^n$

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^m \xi_i$$

subject to $y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i$, where $\xi_i \geq 0$, $w$ is a real-valued vector of same dimension as feature vector $x_i$ and $C$ is a regularization parameter.
The computational steps involved in the model development are listed below:

- Used the training set to train the model
- Conducted a grid search to find the optimum values of parameters of the model like the value of C (the regularization parameter)
- Calculated the hypothesis for the validation set using the optimized model derived from the training set
- Compared the hypothesis values with the labels for validation set and calculated the precision, recall and F1 scores over the validation set
- Re-validated the model over the test set and calculated the precision, recall and F1 scores

Neural networks

Neural network is another model used for classification problems. The neural network used in our solution comprised 19 input features that corresponded with the 19 different columns listed in Table 1. The steps involved in building the neural network for churn prediction are listed below:

- Selected two hidden/activation layers for the churn prediction solution in addition to one input layer and one output/hypothesis/prediction layer
- Inputs to the neural network at layer 1 formed the matrix X of dimensions [2000 x 19]. In other words, 2000 records from the training set were used as inputs with each record containing 19 columns for the 19 input attributes (see Equation 1 for the matrix representation of X)
- Used the advanced optimization function in Octave and passed the cost function $J(\theta)$ that computes forward/backward propagation
- Derived the outputs from the neural network, which were three theta values ($\theta_1$, $\theta_2$, $\theta_3$) corresponding to each layer. These values were used on the validation set to predict the list of possible customers with the propensity to churn
- Plotted the ROC curve to visualize the efficiency of the model
- The ROC curve of the model is shown below. As noticed, the area under the curve is found to be reasonable.
As seen in Fig 9 below, based on the area under the ROC curve, neural network was not the best model for the given dataset.

![ROC curve for neural network](image)

Fig 9: ROC curve for neural network

<table>
<thead>
<tr>
<th>ML Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression – Linear</td>
<td>0.690</td>
<td>0.180</td>
<td>0.286</td>
</tr>
<tr>
<td>Logistic Regression – Polynomial</td>
<td>0.842</td>
<td>0.577</td>
<td>0.684</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>0.868</td>
<td>0.595</td>
<td>0.706</td>
</tr>
<tr>
<td>Neural Network</td>
<td>0.8</td>
<td>0.28</td>
<td>0.415</td>
</tr>
</tbody>
</table>

Table 2: Scorecard of outputs from the three ML models

From Table 2, it can be inferred that the SVM algorithm was the most accurate in predicting churn. The data file and the predictions provided by the SVM algorithm were further fed into Oracle CX as well as Salesforce Dot Com (SFDC) applications to create churn dashboards.
3. Amplify CX

The ultimate goal of using ML models for CX is to amplify and enhance the capabilities that the platform offers by providing relevant and actionable insights for the business.

In this case, we created churn dashboards on Oracle and SFDC Cloud. These dashboards help users slice and dice insights across various dimensions. For example, users can analyze the country states that have the highest concentration of customers who are likely to churn. They can also view how many of their most loyal customers are on the verge of churning. Such insights enable the management to create and deploy strategies to avert customer attrition wherein each strategy is contextualized for a particular customer segment.

**Fig 10:** Churn dashboards on Oracle CX Engagement Cloud and SFDC
Conclusion

Customized dashboards fed by ML algorithms provide key insights to predict customer behavior. This is a valuable tool that empowers management teams to make the right decisions and design rollout strategies to minimize customer churn. Some strategies include launching customer-specific campaigns or tailored discounts that incentivize customers to stay loyal to the brand. Moreover, the actual execution of such campaigns is fairly simple as such capabilities are pre-built into modern CX applications. The example described in this paper demonstrated how ML models can be built to counter customer churn. Through this capability, companies can unlock greater value from their CX systems. As enterprises rapidly digitize themselves, investing time and money in building AI/ML solutions for digital platforms like CX can greatly amplify their capabilities and enhance operational productivity by freeing up manual effort. This will help them achieve faster ROI and stay ahead of competition.
About the author

**Rajesh TK** is a Lead Consultant with Oracle Customer Experience (CX) practice. He has over 16 years of experience in implementing CRM solutions for clients across different verticals (Banking, Telecom, Public Sector, Airlines and Media). His current interest is in AI/ML technologies and frameworks and leveraging them to solve business problems in CRM domain.

Contact Email: rajesh_tk@infosys.com

**Vikram Mohan** is a Principal Consultant with the Oracle Customer Experience (CX) practice. He has close to 15 years of work experience. CX Consulting, developing go-to-market initiatives, defining CX roadmaps & adoption strategies, market research and pre-sales are some of Vikram’s key focus areas within the practice. He is also adept at applying machine learning concepts on CX. In the past, Vikram has handled enterprise wide customer relationship management transformation programs across domains including telecom, insurance, hi-tech and public sector.

Linkedin - www.linkedin.com/in/vikram-mohan1

Contact Email – Vikram_mohan@infosys.com
A. ML problems are broadly divided into three types – reinforcement, supervised learning and unsupervised learning. Problems that need output in discrete values like predicting the sales for a given region in a year based on historic data are called regression problems. Problems that entail categorizing data like analyzing a given dataset to segregate sales leads by quality are termed as classification problems.

References

Acknowledgements
1) We would like to thank our Infosys colleagues Deepti Kunapareddy, Shahul J, Shraddha Jathar, and Vignesh Sivaram for their effort in building churn dashboards in Oracle CX Cloud and SFDC.
2) Coursera - Machine Learning by Stanford University, Dr. Andrew Ng