Rory Hurley

Customer Churn Prediction in the Telecommunications Industry using Support Vector Machines and Multi-Layer Perceptrons

1. Introduction

Artificial Intelligence (AI) and machine learning (ML) comprise a large set of supervised and unsupervised algorithms used for many tasks including prediction, classification and anomaly detection. One of the most successful implementations of machine learning and AI in recent years has been that of customer relations management (CRM) and customer churn. CRM is a strategy for building, managing and strengthening loyal and long-lasting customer relationships (Vafeidis et al., 2015). This strategy is key in many customer-centric industries, including banking, insurance, retail and telecommunications.

The telecommunication industry in particular is a very competitive market where there is the problem of customers churning out (leaving) and switching to other competitors (Mishra, K and Rani, R., 2017). Given that recent reports suggest the cost of customer acquisition is anywhere from 5 to 25 times more expensive than retaining an existing one (Gallo, A., 2015), it is within the interests of telecommunications companies to invest and research rigorous analytical methods that can keep churn rates low, retaining customers. This study employs two modern AI and ML approaches, a Multi-Layer Perceptron and a Support Vector Machine, to predict customer churn within the telecoms industry and critically compares the algorithms and their respective results.

2. Data and Pre-processing

The data used in this study is the IBM 'Telco Customer Churn' dataset (IMB, 2015). The original data consists of 19 predictor variables related to customer account information, information of services that each customer has signed up for and demographic information. There is also a binary variable, 'Churn', which indicates whether the customer left or not. In total there were 7,043 records. Initial exploratory analysis found that far more customers in the dataset stayed at (5,163) than left (1,869) the company, so data rebalancing was done using the ADASYN algorithm which balanced the data whilst reducing the bias introduced by the class imbalance and adaptively shifting the classification decision boundary toward the difficult examples (Haibo et al., 2008). Columns with "Yes/No" entries were converted to binary 1 or 0. Continuous variables were binned and then one-hot-encoded along with the categorical data, resulting in 43 binary predictor variables.

3. Algorithms

3.1. Support Vector Machine (SVM)

The SVM, first introduced by Cortes and Vapnik (1995), is a popular supervised algorithm for classification, regression and anomaly detection. The aim of the SVM is to create the optimal decision boundary between two classes to allow good classification by ensuring that the hyperplane separating classes is at the maximum distance, or 'margin', possible from either class. By doing this, the upper bound of the expected generalisation error is reduced (Mokhtar, A. and Elsayad, A., 2013). For a SVM classifier, the 'Box Constraint', or 'C' parameter, can be defined. Low C values prioritise simplicity and look for a soft-margin to separate classes, whereas high C values prioritise making few mistakes, at the risk of overfitting the training data. Often, data will not be linearly separable. One of the reasons that the SVM is so popular is due its ability to add a new dimension to the data to transform it to a state where it can be more easily separated by a hyperplane. This is called the 'Kernel Trick'. Different kernels can be use and in these cases, an important parameter, 'Kernel Scale' or 'gamma', is introduced. Small gamma values mean less complexity, and large values result in more complexity but more overfitting. The advantages of using an SVM is that it is good at dealing with high dimensional data, and also that it works well on small datasets, and solutions always converge at global minima. Its limitations are that picking the right kernel and parameters can be computationally expensive, and it does not scale well to large datasets (Auria, L. and Moro, R.A., 2008)

3.2. Multi-Layer Perceptron (MLP)

The MLP is a supervised classifier and consists of an input layer, one or more hidden layers and an output layer. There are modifiable weights and biases that connect each neuron in one layer to those in the next. Each neuron in the hidden layer sums its input attributes after multiplying them by the strengths of the respective connection weights and computes its output using activation function of this sum, which may range from a simple threshold function, or a sigmoidal, hyperbolic tangent, or radial basis function (Mokhtar, A. and Elsayad, A., 2013). Networks with just two layers of weights are capable of approximating any continuous functional mapping (Bishop, 1995) and overcome the limitations of a single Perceptron. The network learns by comparing its output with the true output, and then propagating the error back through the network using the Backpropogation algorithm in order to adjust the weights correctly. This process is repeated until the global error is minimised. Two important parameters are often used in this process; 'Learning Rate' and 'Momentum'. The advantages of Neural Networks are that they have the ability to learn and model non-linear and complex relationships, the ability to work with larger amounts of data in less time than typical ML algorithms, and do not impose any restrictions on the input variables (like how they should be distributed). Disadvantages are that MLPs are too much of a 'black-box', making it difficult to interpret results, despite the accuracies. Further, the MLP is not well-suited to small datasets.

4. Experimental Design and Hypothesis

The purpose of this study will be to train SVM and MLP models using training the Telco data in order to make a model that will generalise well to unseen data. This section will discuss the methodology for model selection and algorithm comparison. The following hypotheses are proposed:

- Hypothesis 1: We expect that the thorough experiments conducted in this study will improve on results from similar projects using this dataset. The null hypothesis is that neither model will be able to predict churn with accuracy >80%
- H2: MLP will be a better predictor of churn than the SVM, the null hypothesis being that MLP will not outperform the SVM.

4.1 Methodology

The data was separated with 80% used for training, and 20% used for testing, avoiding data snooping. The training data would form the basis for hyperparameter tuning and eventually model selection. Of the training data, stratified 10-fold cross validation as, according to Kohavi (1995), this is a better scheme, in terms of bias *and* variance, when compared to regular cross-validation.

4.2. SVM Methodology

As discussed in 3.1., there are various kernel functions that can be used to transform the data to ensure separability. Kernel functions used in this study were 'linear', 'gaussian' and 'polynomial'. Of these kernel functions, each has hyperparameters that define the kernel box constraint and the kernel scale, with the addition of a polynomial order for the 'polynomial' kernel. The aim of the methodology was to find the optimum kernel and hyperparameter combination that would best model the data and then generalise well on unseen data.

Due to the broad range of continuous values that could be used as hyperparameters, Bayesian optimisation (illustrated in Figure 1) was used to approximate initial values for the hyperparameters for each kernel function. A grid search was then carried out using values ±25% of the estimate given by the Bayesian optimisation. This mean that by the end of the ensemble of Bayesian optimisation and grid search, a total of 45 models were run using 10-fold cross validation. The average accuracies of these were recorded, as well as the average wall time to train each model, and then an optimal model was selected for testing on the held-out test data.

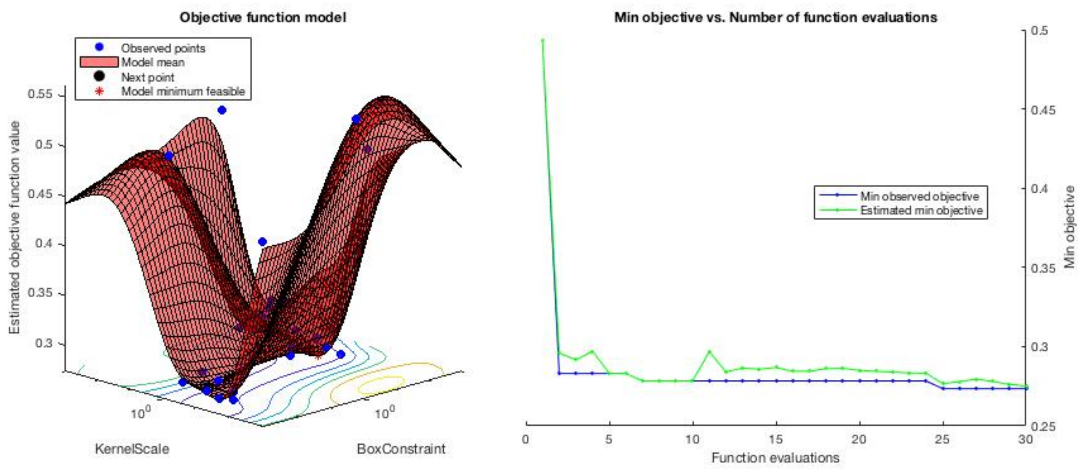

Figure 1 - Visual representation of Bayesian Optimisation process.

4.3. MLP Methodology

The MLP used in this project used the backpropagation algorithm for gradient descent with momentum and adaptive learning rate, known as 'traingdx' in Matlab. This algorithm was selected as it allowed for parameterisation of both momentum and learning rate identified in 3.2. As well as these, the architecture of the network was important. During some trials carried out in the pre-testing phase, it was evident that a network architecture of 3 hidden layers produced better results than 1 or 2 layers, so this would be used with varying layer sizes.

As this was a binary classification task, it was deemed suitable to use a cross-entropy performance function with probabilistic 'softmax' output as it has been found to perform better than the other machine learning methods in similar classification tasks (Kemal et al., 2019). The network would therefore be formed of three hidden layers with varying sizes of [10, 30, 50] neurons, varying momentums [0.01, 0.1 0.5, 0.9] and varying learning rates [0.01, 0.1 0.5, 0.9]. A grid search was conducted using these parameters, with maximum iterations of 300 and with 15 validation checks, which were found to maximise performance without risk of overfitting. There were 48 different configurations.

6. Model Selection and Results

6.1. SMV Model Selection

Table 1 shows the results of the grid search experiments following Bayesian optimisation. It is clear that following 10-fold cross validation, the best performing all used the 'gaussian' kernel. These consistently achieved accuracies over 88%, with the best achieving 89.019% accuracy on the validation sets, using a Box Constraint of 1181.576 and Kernel Scale of 0.318. Interestingly, these parameters were 25% higher and 25% lower, respectively, of the estimates from Bayesian optimisation.

Further, all models using the ‘gaussian’ kernel all surpassed other models with respect to time, all being trained in under 7 seconds on average.

6.2. MLP Model Selection

Table 2 shows the results of the grid search outlined in 4.3. Following 10-fold cross validation, accuracies of training experiments varying from 73% up to 81%. The general trend showed that hidden layers with more neurons (50) produced more accurate results, and 3 hidden layers with 10 neurons each generally performed at the lower end of the spectrum. Within each structure, the best performances were reached when learning rate was highest at 0.9, whereas momentum made least difference although performance appears best when 0.5 or higher. Better performance in the larger networks comes at the expense of average time taken, however these were still mostly under 20 seconds so made little practical difference. The optimal hyperparameters were selected from the parameter combination with the best 10-fold accuracy. This was an architecture of 3 hidden layers with 50 neurons each, a learning rate of 0.5 and a momentum of 0.9.

Rank	Kernel	Box Constraint	Kernel Scale	Polynomial Order	10-fold Accuracy	10-fold Time
1	gaussian	1181.576	0.318	-	89.017	6.304
2	gaussian	708.946	0.530	-	88.834	4.841
3	gaussian	945.261	0.318	-	88.748	6.188
4	gaussian	945.261	0.424	-	88.711	5.407
5	gaussian	1181.576	0.424	-	88.699	5.112
6	gaussian	708.946	0.318	-	88.662	6.368
7	gaussian	1181.576	0.530	-	88.625	4.884
8	gaussian	708.946	0.424	-	88.406	5.197
19	polynomial	0.004	0.261	2	75.284	315.196
20	polynomial	0.006	0.261	2	75.210	306.299
21	polynomial	0.007	0.261	2	75.174	298.073
22	linear	57.138	1.013	-	72.718	305.103
23	linear	71.422	1.013	-	72.645	303.200
24	linear	42.853	1.013	-	71.753	307.035
25	linear	71.422	1.350	-	71.484	307.230
26	linear	57.138	1.350	-	71.289	304.812
27	linear	42.853	1.350	-	70.837	265.461
37	polynomial	0.006	0.348	4	51.423	227.972
38	polynomial	0.007	0.435	4	51.411	222.670
39	polynomial	0.006	0.435	4	50.764	210.770
40	polynomial	0.004	0.348	4	50.629	228.438
41	polynomial	0.007	0.348	4	50.446	232.939
42	polynomial	0.004	0.435	4	50.239	235.738
43	polynomial	0.007	0.261	4	49.896	232.991
44	polynomial	0.004	0.261	4	49.884	238.581
45	polynomial	0.006	0.261	4	49.566	243.933

Rank	Structure	Momentum	Learning Rate	10-fold Accuracy	10-fold time
1	[50,50,50]	0.5	0.9	0.805	17.638
2	[50,50,50]	0.5	0.01	0.800	15.384
3	[50,50,50]	0.5	0.5	0.799	14.395
4	[50,50,50]	0.5	0.1	0.792	14.338
5	[30,30,30]	0.5	0.9	0.788	7.093
6	[50,50,50]	0.9	0.5	0.787	8.359
7	[50,50,50]	0.9	0.01	0.784	8.518
8	[30,30,30]	0.5	0.5	0.780	5.452
9	[50,50,50]	0.9	0.9	0.778	6.955
10	[30,30,30]	0.5	0.1	0.778	6.321
21	[10, 10,10]	0.9	0.5	0.757	1.763
22	[10, 10,10]	0.5	0.9	0.755	2.071
23	[10, 10,10]	0.5	0.1	0.752	2.483
24	[30,30,30]	0.1	0.01	0.745	3.395
25	[10, 10,10]	0.1	0.1	0.742	1.419
26	[50,50,50]	0.1	0.01	0.742	5.205
27	[10, 10,10]	0.1	0.01	0.742	2.247
28	[50,50,50]	0.01	0.01	0.740	5.591
29	[30,30,30]	0.01	0.01	0.740	3.903
30	[10, 10,10]	0.01	0.01	0.738	9.670
31	[10, 10,10]	0.9	0.1	0.736	1.866
32	[10, 10,10]	0.01	0.1	0.734	7.308
40	[50,50,50]	0.01	0.9	0.720	4.215
41	[10, 10,10]	0.1	0.9	0.718	1.079
42	[50,50,50]	0.01	0.1	0.716	3.076
43	[10, 10,10]	0.01	0.5	0.714	1.227
44	[50,50,50]	0.1	0.9	0.713	2.971
45	[50,50,50]	0.1	0.5	0.708	2.978
46	[30,30,30]	0.1	0.9	0.701	2.120
47	[30,30,30]	0.1	0.5	0.699	1.372
48	[30,30,30]	0.01	0.5	0.697	2.378

Table 1 (Left) - SVM Results (Top, Middle, Lower 8) and Table 2 (Right) - MLP Results (Top 10, Middle 10, Lower 8)

6.3 Algorithm Comparison

Following the model selection process outlined in 6.1. and 6. 2. the best performing SVM and MLP models from the experimental training phase were selected with their respective parameters. Previously unseen testing data was used to generate output classification predictions which were then compared to the true labels. From this, key model performance statistics such as accuracy, precision, recall and specificity could be compared as well as confusion matrices and ROC curves.

Figure 2 shows the confusion matrix for the best SVM model. From this we can see that the model performs with an accuracy of 89.15%, very slightly surpassing the performance of the model during training. In addition to this, the SVM performed with high precision with 92.2% of positive predictions correctly classified. The high recall of 86.7% suggests that the model does perform well in terms of predicting the number of customers leaving that were correctly identified, but even better at predicting the number of customers staying with the company, with specificity at 91.9%. The SVM ROC curve illustrates the models good performance with its shape relatively close to the axes, and resulted in a strong AUC score of 0.891.

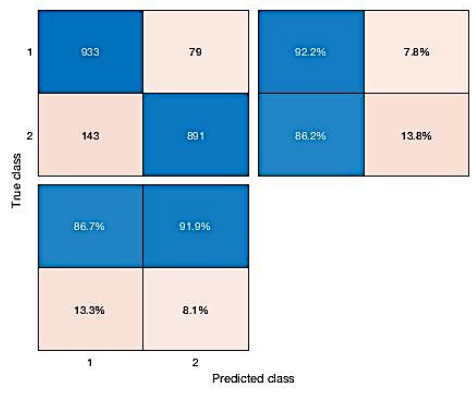

Figure 2 - SVM Confusion Matrix

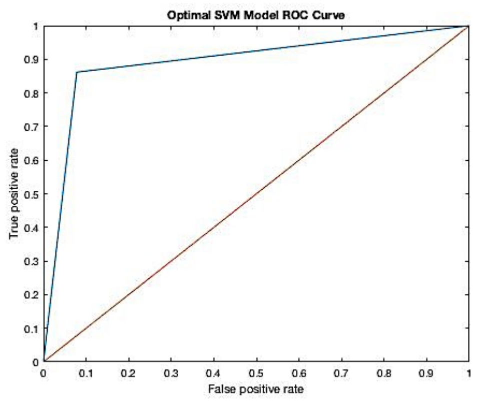

Figure 3 – SVM ROC Curve

In contrast to the SVM, when predictions were made by the MLP on the unseen test data, the MLP's accuracy was in fact lower than that of the model in the training phase at 73.6% which suggests there may have been some degree of overfitting occurring. In addition to the poorer accuracy, the MLP was inferior in performance across the board. The confusion matrix in Figure 4 shows that the precision is lower at 69.1%, meaning that it was worse in terms of correct positive predictions. Further, the MLP didn't correctly identify as many leaving customers than the SVM, with a recall of 75.5% and had its specificity was lower at 72.1%. The ROC curve for the MLP model is also shown in Figure 5. This shows a similar trend to the SVM ROC, but is not as tight to the axes and thus the area under the curve (AUC) is 0.813.

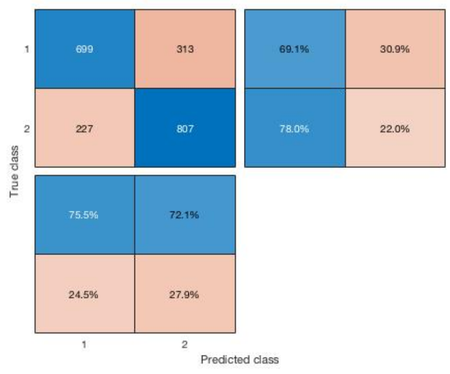

Figure 4 - MLP Confusion Matrix

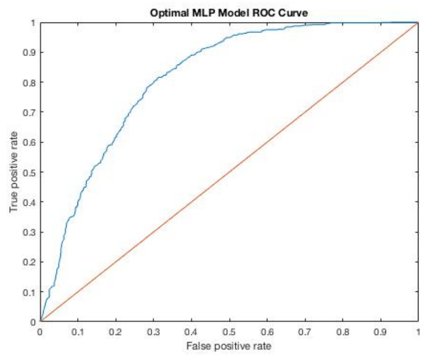

Figure 5 – MLP ROC Curve

7. Discussion & Conclusion

Following analysis of the results, we can accept Hypothesis 1, as the SVM achieved performance >80%, and reject Hypothesis 2 as SVM outperformed the MLP. It is clear that the performance of the SVM model created during the training experiments far surpasses the performance of the MLP. This means we can reject the initial hypothesis and accept the null hypothesis. This is consistent across all performance metrics used in this study. However, that is not to say that this was due to poor parameterisation of the MLP. On top of the fact that a thorough experimental design was carried out during the model training stages, we can see that the MLP performed better than most of the SVM configurations seen in Table 1, with the exception of the SVMs using a Gaussian kernel. Therefore, it is more suitable to conclude that the SVMs with Gaussian kernels performed exceptionally, rather than the MLP performing poorly. Further, the MLP as well as, or better than publications using this dataset on the website *Kaggle* – the only instances of this dataset being used in experiments at the time of completion.

The reason for the exceptional performance of the SVM with Gaussian kernel could be due to the underlying distribution of the data. During the initial exploratory analysis, it was found that many of the predictors were normally distributed, such as age, monthly bills, and total bills. Furthermore, the use of ADASYN for adding synthetic data to rebalance the data could have introduced some bias that would have improved performance using a Gaussian kernel (Garcia et al., 2009). These could be potential reasons for the strong SVM performance with the SVM with Gaussian kernel.

One other possible factor for the superior performance of the SVM could be due to the decision to use an ensemble of Bayesian optimisation and grid search in the methodology. This was a good decision due to the fact that when initialising the SVM grid search, there is an incredibly large set of possible hyperparameter configurations to choose from, so being able to find approximations for these as a basis for the grid search narrows the choice down and eventually led to finding the optimal parameters in the grid search. Had this ensemble not been used, the optimal parameters would likely have been missed completely and thus limiting performance of the SVM.

This study has shown that both MLPs and SVMs can be used for the application of predicting customer churn in the telecommunications industry. Despite the inferior performance of the MLP compared to the SVM, the method still performs well and if implemented by a telecoms company, could lead to substantial cost savings with regards to mitigating customer churn and retaining customers. The SVM was found to have performed exceptionally, surpassing MLP performance as well as other projects using this dataset. This success is likely attributed to the combination of Bayesian optimation and grid search to find the best performing hyperparameters. This study has proved that AI and ML models such as the SVM can be of high commercial value and worthy of implimentation for applications such as classification. Further work could look focus on applying these classification algorithms elsewhere in telecoms data, or indeed applying alternative algorithms to solve similar problems with imbalanced datasets.

References

Auria, L., Moro, R.A. (2008) ‘Support Vector Machines (SVM) as a Technique for Solvency Analysis’, *DIW Berlin German Institute for Economic Research, pp. 7.*
Bishop, C. (1995) ‘Neural Networks for Pattern Recognition’, Clarendon Press, Oxford. Chapter 4, pp. 116.
Cortes, C. and Vapnik, V. (1995) ‘Support Vector Networks’, *Machine Learning*, Kluwer Academic Publishers, Vol 20, pp. 273-297
Gallo, A. (2014) ‘The Value of Keeping the Right Customers’ *Harvard Business Review*, Accessed 27/03/2019, Available at: https://hbr.org/2014/10/the-value-of-keeping-the-right-customers
Haibo, H., Yang, B., Garcia, E.A. and Li, S. (2008) ‘ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning’, *2008 International Joint Conference on Neural Networks (IJCNN 2008),* pp. 1322 – 1328
Kemal, A., Kiliçarslan, S., Cömert, O. (2019) ‘Classification and diagnosis of cervical cancer with stacked autoencoder and softmax classification’, *Expert Systems with Applications*, Volume 115, pp. 557-564.
Kohavi, R. (1995) ‘A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection’, *International Joint Conference on Artificial Intelligence*
Mishra, K. and Rani, R. (2017) ‘Churn prediction in telecommunication using machine learning’ *International Conference on Energy, Communication*, *Data Analytics and Soft Computing* (ICECDS), Chennai, 2017, pp. 2252-2257.
Mokhtar, S. and Elsayad, A. (2013) ‘Predicting the Severity of Breast Masses with Data Mining Methods’, *International Journal of Computer Science Issues (IJCSI)* Volume 10, Issue 2
Stacker IV, M. (2015) ‘Using Customer Behavior Data to Improve Customer Retention’, IBM, Accessed 27/03/2019, Available at: https://www.ibm.com/communities/analytics/watson-analytics-blog/predictive-insights-in-the-telco-customer-churn-data-set/
Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G., Chatzisavvas, K.Ch. (2015) ‘A comparison of machine learning techniques for customer churn prediction’, *Simulation Modelling Practice and Theory*, Volume 55, June 2015, pp. 1-9