A Subscription-Likelihood Prediction Model for Rich Communication Services

Capstone Project Report

for

Udacity Machine Learning Engineer Nanodegree

By

Oluwasegun Francis Sogunle

March 3, 2019

Chapter 1

Introduction

1.1 Overview

Internet-based communication services and the proliferation of smart phones continuously drive the ubiquity and evolution of modern communication technology. These services now shape the way and manner in which people communicate and collaborate. Example of Internet-based services include Facetime, Skype, Whatsapp and Snapchat. Consequently, customers now rely on these free services much more than the traditional communication services that the Mobile Network Operators (MNOs) provide. Traditional communication services include Short Messaging Service (SMS) and voice call service.

Although MNOs provide their customers (i.e., smartphone users) with internet access infrastructure, their customers often choose to use the free Internet communication services. Furthermore, Internet communication service providers do not have contracts with MNOS. In an attempt to retain their customers, the Global System for Mobile Communications Association (GSMA)[1] introduced the Rich Communication Services [5] framework as a unified communications solution. RCS aims to provide services ranging from traditional voice call and Short Messaging Service (SMS) to instant messaging, voice and video over IP (VVoIP),[2] geolocation exchange, and presence, among others [7].

[1] A consortium of MNOs, Original Equipment Manufacturers (OEMs), and telecommunication-inclined stakeholders.

[2] IP - Internet Protocol.

1.2 Problem Statement

Although GSMA positions RCS at the core of MNO service offerings for future generation telecommunication networks, there has been relatively low adoption and pervasiveness of this suite of services [11]. Challenging issues such as the initial support from equipment manufacturers, creation of attractive billing models and well-tailored service offerings, and targeting the right and adequate set of customers contribute to this state.

MNOs would benefit from models that could (as a first step), *predict the most likely set of customers* based on their Call Detail Records (CDRs).[3] The likely set of customers in this context are those who are active users of the telecommunication network services at a specific location. In other words, those who extensively use the traditional voice call and SMS services that MNOs offer. As a rule, these class of customers are likely to adopt the new RCS service offerings. Moreover, the model will be built on pre-clustered training data, which is further trained with an Artificial Neural Network (ANN). The model's performance will be evaluated based on its predictive effectiveness. Section 1.3 discusses the evaluation metrics in more detail.

1.3 Metrics

As discussed in Section 1.2, the model's ultimate goal is to predict which set of customers are likely to adopt RCS. Hence, this is a binary classification problem which can sufficiently be evaluated with the aid of binary testing techniques. One of such techniques is confusion matrix-based analysis. The confusion matrix comprises the following calculations [13]:

- True Positive (TP): Number of correctly predicted positive cases.
- True Negative (TN): Number of correctly predicted negative cases.
- False Positive (FP): Number of incorrectly predicted positive cases.
- False Negative (FN): Number of incorrectly predicted negative cases.

[3]CDR is a log of a customer's activities while using an MNO's communication infrastructure. For example, the network generates a CDR when a customer places a voice call.

Figure 1.1 shows an example of a confusion matrix for a model that tries to predict people with a disease. The model's diagnoses can be broken down as follows: TP is 12, FP is 3, TN is 77, and FN is 8.

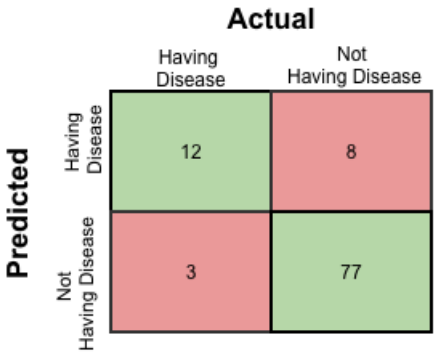

Figure 1.1: Showing a confusion mastrix for disease prediction. Source: [13].

Furthermore, analysis to be performed on the confusion matrix parameters will go beyond mere accuracy prediction. This is due to the fact that the model will most-likely predict least-likely RCS subscribers accurately since they represent a relatively large percentage of the dataset. As a result, the model will have a high accuracy. This makes comparative performance evaluation based on accuracy unreliable. Formula for calculating accuracy is as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

The challenge here is to ensure that the model can be relied upon to identify the much needed customers.[4] In other words, the model precision is a strong requirement since it can help understand the correctly predicted likely customers out of the actual segment of likely customers. Therefore, metrics like precision, recall, F1-score, and area under the Receiver Operating Characteristic (ROC) scores[5] will be calculated for comparisons with the benchmark model. They are defined as follows [3, 4]:

[4]Since predicting the wrong set of customers is not desired.
[5]NOTE: This will be used for free-form visualization in Section 5.1.

- **Precision**: Is the proportion of positive identifications that were actually correct.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**: Is the proportion actual positives that were identified correctly.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-Score**: Is the weighted average of precision and recall.

$$F1 - Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

- **ROC Curve**: Is a graph that plots the True Positive Rate (TPR) against False Positive Rate (FPR). TPR is another name for recall in this case. FPR can be calculated as described below:

$$FPR = \frac{FP}{FP + TN}$$

  Hence, the area under ROC (AUC) measures the area underneath the ROC curve from (0,0) to (1,1).

However, before computing the confusion matrix and performing analysis using the aforementioned metrics, it is important to note that the dataset discussed in Section 2.1 is unlabelled. The first step would be to define and extract ground truth (labels) for each CDR. Thereafter, the adopted clustering technique's effectiveness will be validated by an expert who provides truth values for a sample dataset. Thus, final model evaluation will be conducted on the model that is trained on labelled data.

Chapter 2

Analysis

2.1 Data Exploration

The mobile phone activity dataset [6] extracted from Kaggle provides the necessary CDRs for the model. The proposed approach was to use dataset records obtained within twenty-four (24) hours. That is, using records found in **sms-call-internet-mi-2013-11-01.csv**. The dataset comprises subscriber CDRs that were gathered in Milan. However, after handling NaN[1] values and looking at the number of remaining records, it would be advantageous to expose the model to more training and testing data beyond 24 hours for evaluation purposes. Consequently, the primary dataset is a consolidation of records obtained within five days.[2]

Figure 2.1 shows the proportion of retained records to the removed records from each dataset. The figure clearly shows that simply computing the mean across each column to replace respective NaN values may not be adequate, since significantly large part of the data set contains records with NaN values. Table 2.1 describes the consolidated dataset features. In addition, Table 2.2 presents the mapping of each of the five datasets shown in Figure 2.1 to their actual file names.

Table 2.1: Dataset Feature Definition.

Feature	**Description**

[1] Not-A-Number.

[2] Five out of seven days in total. This is to keep the dataset size below 500MB.

CellID	Identification string of a given square of Milan grid.
countrycode	A nation's country code for the subscriber's number.
smsin	Amount of SMSs received a given square id and during a given time interval.
smsout	Amount of SMSs sent inside a given square id during a given time interval.
callin	Amount of calls received inside the square id during a given time interval.
callout	Amount of issued calls inside a given square id during a given time interval.
internet	Number of CDRs generated inside a given Square id during a given time interval.

Notably, the square grid identifies the location of a specific cell which is identified by a CellID. The dataset comprises 1891928 records and 8 features. However, the **datetime** feature will not be used as it retains a constant value across records.

Table 2.2: Dataset Filename Mapping.

Dataset ID	**File Name**
Dataset 1	**sms-call-internet-mi-2013-11-01.csv**
Dataset 2	**sms-call-internet-mi-2013-11-02.csv**
Dataset 3	**sms-call-internet-mi-2013-11-03.csv**
Dataset 4	**sms-call-internet-mi-2013-11-04.csv**
Dataset 5	**sms-call-internet-mi-2013-11-05.csv**

Dataset 1 had 233466 out of 1891928 records remaining resulting in 87.8% decrease in total record numbers. Dataset 2 had 233466 out of 1891928 records remaining resulting in 87.8% decrease. Dataset 3 had 225540 out of 1828063 records remaining resulting in 87.66% decrease. Dataset 4 had 227781 out of 2299544 records remaining resulting in 90.09% decrease. Dataset 5 had 231063 out of 2397759 records remaining resulting in 90.36% decrease. As a result, the new record length for the consolidated dataset is 1142231 records.

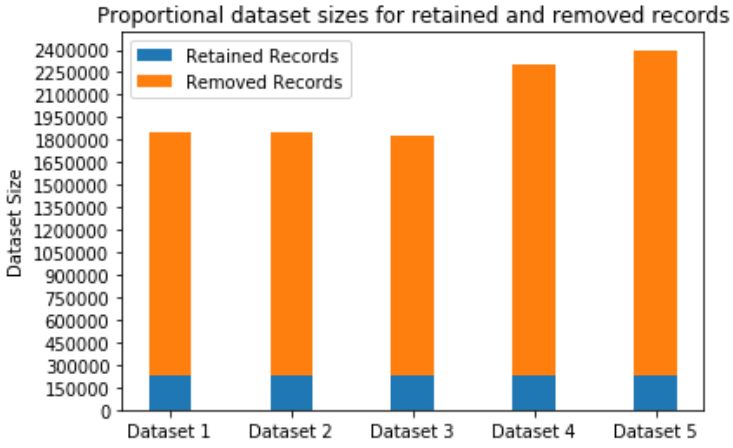

Figure 2.1: Showing the proportion of removed and retained records.

2.2 Exploratory Visualization

In order to distinguish what features can be easily predicted from those that are not, each feature is considered as target feature. In other words, each feature was considered a target feature in order to ascertain how easy it is to predict the feature. This requires the use of a supervised learning algorithm. Therefore, the target feature is considered the dependent variable (y) while the remainder of the features were temporarily considered independent variables (X). For the purpose of this project, the decision tree learning algorithm was applied on the independent and dependent variable(s). After such an exploration, CellID and countrycode features were the most difficult to predict judging by their regression scores — 0.25 and 0.06 respectively.[3] Figure 2.2 shows a bar chart of the respective regression scores for each feature while Figure 2.3 displays the correlation between these features.[4]

[3]The calculations can be found under 'Exploratory Visualization' heading in the Analysis (Jupyter) Notebook.

[4]Value of one (1) indicates absolute correlation.

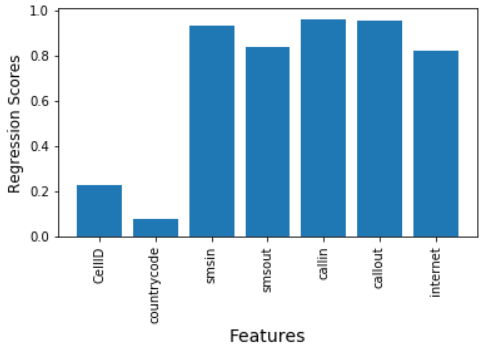

Figure 2.2: Regression Score Bar Chart.

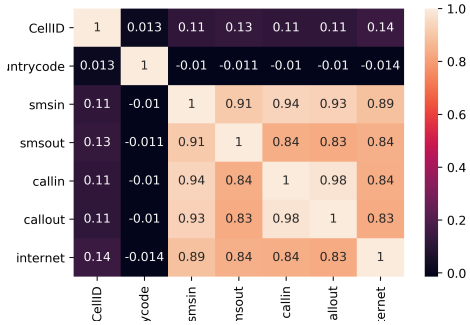

Figure 2.3: Dataset Correlation.

2.3 Algorithms and Techniques

As mentioned in Section 1.3, the initial dataset is unlabelled. In consequence, a human (domain) expert[5] makes a distinction of customers that are most-likely to adopt RCS on the basis of their usage level.[6] Hence, an unsupervised learning is adopted for this first task.[7] Section 2.3.1 discusses this in more detail. Thereafter, Section 2.3.2 discusses the supervised learning approach that is used to build the prediction model.[8]

2.3.1 Unsupervised Learning

A possible approach would be the use of K-means [1] algorithm. K-means can be used on this dataset since k[9] is small and more often than not faster than other techniques such as hierarchical clustering. However, K-means considers the means and not the covariance of that describes the ellipsoidal shapes of clusters. In other words, it may not perform effectively on clusters with different shapes. Furthermore, the dataset will be scaled and therefore would require a technique that is less sensitive to scaling when trying to normalize the distribution. This lead to the adoption of the Gaussian Mixture Model (GMM) [8, 9]. Moreover, GMM considers the degree of association of a datapoint when clustering as opposed to binary outcomes adopted by other

[5]In this case, the author of the text.
[6]That is, usage of communication services that MNOs currently offer.
[7]Task of defining the ground truth for unlabelled data.
[8]That is, the classifier that makes prediction on unseen data with adequate precision.
[9]The number of clusters.

techniques such as K-means. This is important for the chosen dataset since a significant number of data points overlap.

2.3.2 Supervised Learning

To create the prediction model, a simple Artificial Neural Network (ANN) will be constructed.[10] The ANN model will train using the labelled data derived from Section 2.3.1. The combination of the supervised and unsupervised machine learning techniques results in the hybridized predictor. Given the flexibility of ANNs, which is due to their depth of configurable parameters, other supervised learning approaches[11] were considered but not adopted. Moreover, the ease with which a multi-layer feed-forward network can be set-up with adequately tunable parameters also influenced this decision. However, an alternative supervised learning technique is used for the model discussed in Section 2.4.

2.4 Benchmark Model

Considering the fact that this is a novel approach, the benchmark model would be a simple logistic regression model that is built and trained with the labelled data. Logistic regression was chosen simply because it is easier to implement and results computation is relatively quicker than building, training and optimizing a neural network. Moreover, logistic regression requires very few parameters for tuning. This model will form the basis for the proposed hybridized[12] model's performance evaluation.

However, it is noteworthy that the GMM clustering model's performance will be determined by the comparison of its predictions to that of the domain expert. In other words, the expert validates the clustering results. In any case, the target prediction performance of the final model should be above 90% accuracy and the model should outperform its benchmark. Final model's performance will be judged using the other metrics presented in Section 1.3. This is the guiding principle for the evaluation discussed in Section 4.1.

[10] Simply put, a Feed-Forward Neural Network.
[11] Such as Support Vector Machines (SVMs) [10, 14] and Ensemble Methods [2, 12].
[12] Combination of GMM clustering and ANN classification.

Chapter 3

Methodology

3.1 Data Preprocessing

This section presents the techniques used to prepare the dataset for the machine learning algorithms discussed in Section 2.3. The discussion is organized as follows: First, removal techniques for NaN values and unused features were implemented. Following this, samples were extracted from the main dataset. Next is a brief discussion on the normalization and dimension reduction techniques applied on the dataset. Finally, the dimension of the data was reduced as a way for the clustering algorithm to efficiently use the training data.

3.1.1 Removing NaN values and unused features

Section 2.1 previously discussed the rationale behind NaN value removal which is to mitigate its effect on learning by increasing the size of the main sample data. The **datetime** feature contains a string variable across its records. As a result, it contributes the least to the mathematical computations required to find active target telecommunication service users — the likely RCS customers. Hence, training, testing and validation datasets do not comprise this feature.

3.1.2 Sample Extraction

A sample of 10 records were chosen to evaluate the correctness of the GMM clustering algorithm. The logic for this extraction can be found under the

'Sample Extraction' block in the Methodology notebook. This will be serve as the validation basis for the algorithm. The sample is a concatenation of two distinct samples, namely: likely and unlikely groups. Likely groups are for the target RCS customers while unlikely represents the converse set. Figures 3.1 and 3.2 provide a snapshot of these groups respectively.

	CellID	countrycode	smsin	smsout	callin	callout	internet
4308	4352.0	39.0	81.9838	64.3447	73.8145	65.9516	1806.9842
76199	4989.0	39.0	40.3979	37.0648	45.1203	49.5017	1087.1416
74959	3751.0	39.0	27.9552	20.6433	24.7905	31.5422	701.5989
84038	2733.0	39.0	19.4405	17.4271	19.6668	24.8393	498.8634
75486	4278.0	39.0	15.7487	14.9308	13.5218	15.8724	446.0121

Figure 3.1: Likely RCS Customers.

	CellID	countrycode	smsin	smsout	callin	callout	internet
7762	7848.0	39.0	2.9440	2.2726	1.9023	1.8718	104.5199
3978	4022.0	39.0	2.0188	2.0544	1.1605	1.3302	81.5893
69755	8581.0	39.0	1.3320	1.5861	0.8501	1.1218	66.0231
33	34.0	39.0	0.9476	0.9502	0.7321	0.6431	46.7024
131	135.0	39.0	0.6597	0.6859	0.3853	0.4431	35.5341

Figure 3.2: Unlikely RCS Customers.

One could argue that since RCS would be an Internet-based offering to MNO customers, then simply looking for those who use the internet for longer times might provide adequate suggestions for likely RCS customers. This position simply disregards those customers who truly use traditional services for their day-to-day activities and also does not consider that the fact that heavy internet users might not be willing to pay for any MNO service offerings. This is due to the fact that they already have free-subscription-cost-free access to services like Whatsapp and Skype. A model that is based

on this naive assumption will not correctly identify these target customers.[1] Hence, it is important to consider the broad spectrum of these features as a combined usability predictor for the model to be developed.

3.1.3 Normalization and Dimension Reduction

Normalization was achieved by scaling the entire dataset with natural logarithm computation. The resultant dataset with log values was used as an input for the Principal Component Analysis (PCA) dimensionality reduction technique. Figures 3.3 and 3.4 show the dataset before and after normalization. Figure 3.5 shows the explained variance computation for the dataset dimensions. This influenced the decision on the final PCA components to extract for the dataset. In other words, since three (3) components explain greater than 95% of the data,[2] the chosen dimension was 3. Notably, the sample data set[3] was also normalized and transformed using these techniques.

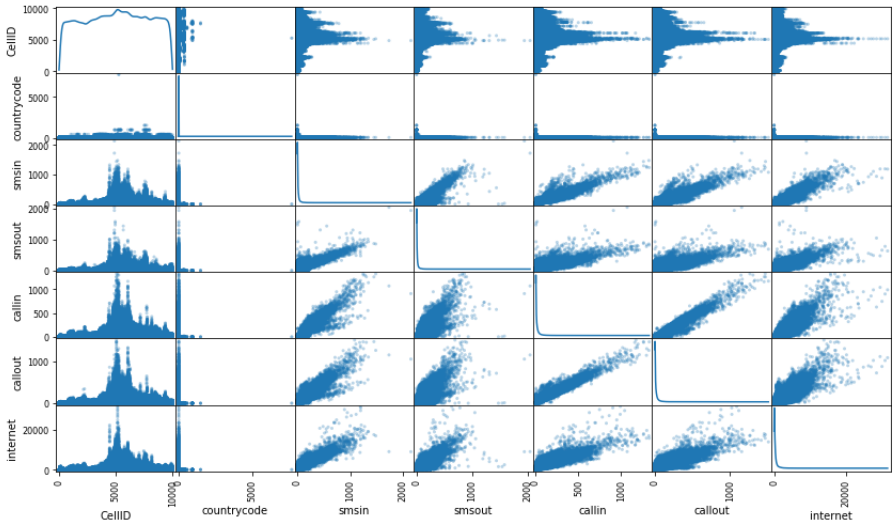

Figure 3.3: Scatter matrix of dataset features before normalization.

[1] This could also be a benchmark model for the proposed technique but it does not consider all the features which the proposed model will consider. Thus making it difficult to adequately make comparisons.

[2] [Component 1+Component 2+Component 3] = [0.8024 + 0.0918 + 0.0654] = **0.9596**

[3] Discussed in Section 3.1.2.

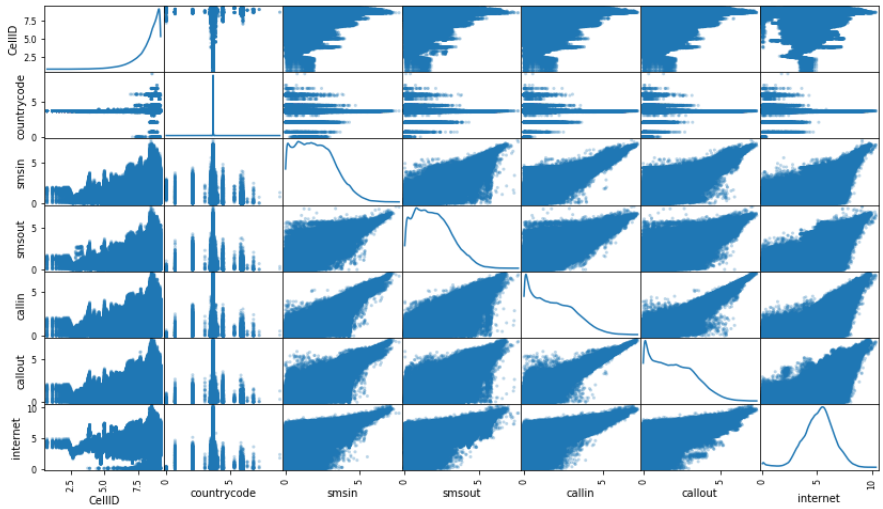

Figure 3.4: Scatter matrix of dataset features after normalization.

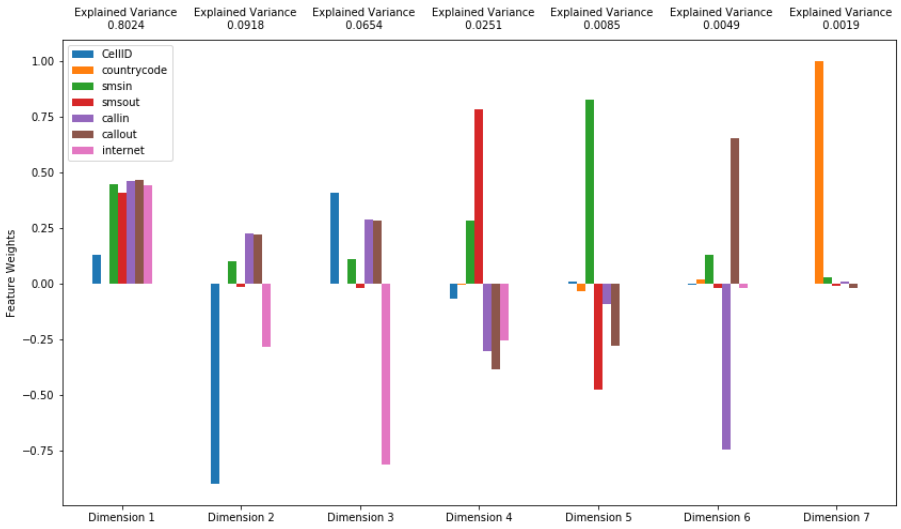

Figure 3.5: Showing the explained variance for each dataset dimension.

3.2 Implementation

This section presents the execution of some the earlier mentioned algorithms and techniques. The discussion starts with the implementation

of the clustering model. Thereafter, the implementation of the benchmark model is presented. Finally, the section discusses the hybridized model implementation.

3.2.1 Implementing the Clustering Algorithm

As mentioned earlier the GMM clustering algorithm was adopted. However, before proceeding to generate clusters, there was an investigation of the optimal number of clusters within the dataset. This exploration used the silhouette score metric on a sample of the dataset[4] as a basis for cluster number (k) performance comparisons. Consequently, the dataset also supports the initial idea that there are two (2) distinct categories of subscribers — those that are likely to subscribe to RCS and those that are not. In other words, the optimal number of clusters for labelling the data supported the earlier claim that there is only one group of users that are likely to adopt RCS. Hence, chosen number of clusters is 2, which is also the optimal number of clusters for the dataset. Figure 3.6 shows the result after applying GMM clustering algorithm on the dataset.

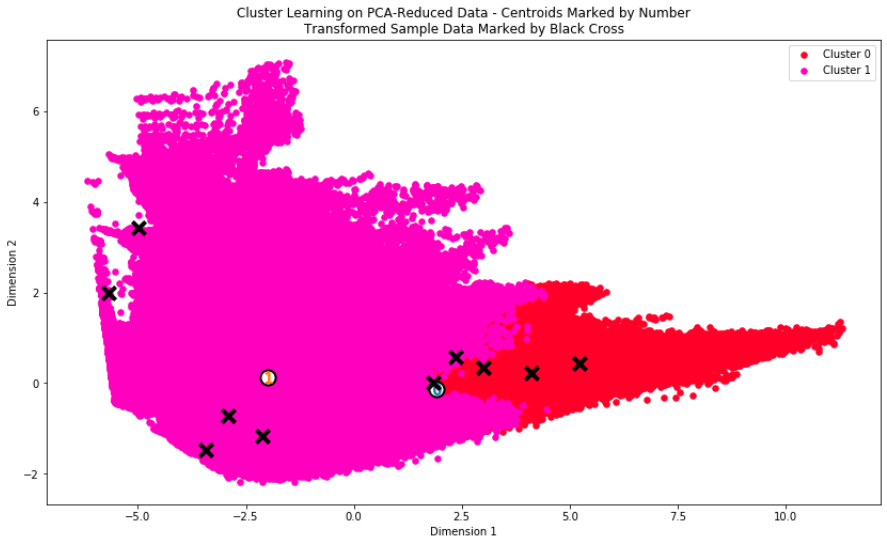

Figure 3.6: Showing the GMM clustering result.

From the clustering results, the points marked with black crosses

[4]About 20000 records.

represent the sample predictions. **Cluster 0** denotes customer segment that are likely to adopt RCS while **Cluster 1** denotes customers that are not likely to adopt RCS.

In order for the domain expert to validate the clustering result, cluster centroids were computed and the entire dataset was re-transformed from logarithmic values. This is possible by simply finding the exponent of the corresponding values in the dataset. Thereafter, a quick run on the sample dataset proved that the cluster results and the sample predictions in Figure 3.6 were 100% accurate. The final step in this implementation is to extract the cluster labels and attach it to the main dataset. Recall that the **Cluster 0** represents likely customers while **Cluster 1** represents the opposite. In order to make this as close to the ground truth as possible, the Cluster IDs are interchanged in the dataset. Hence, Cluster 0 predictions have label '1' while Cluster '1' predictions have label '0'.

3.2.2 Implementing the Benchmark Model

A logistic regression model was implemented as the benchmark model. The implementation logic can be found in the *Experiments and Results* notebook. However, the dataset used was split into training and testing set with the aid of the Scikit-Learn model selection `train_test_split` module. This training portion of the dataset is further split into testing and validation sets for the model in Section 3.2.3.

3.2.3 Implementing the Hybridized Model

This is the proposed (prediction) model for this project. It is built with an ANN that is defined in Keras. It comprises three fully-connected layers; the input, hidden and output layers. The implementation logic can be found the *Experiments and Results* notebook. Table 3.1 presents the training parameters for the keras-built ANN model.

Table 3.1: ANN Tuning Parameters.

Parameter	**Value**
Number of Classes (i.e., Output layer size)	2

(Final)[5] Dropout Value	0.2 for input and hidden layers.
Activation Function	Softmax
Loss Function	Categorical Cross-Entropy
Optimizer	Adaptive Moment Estimation (Adam)
Metrics	Accuracy

3.3 Refinement

The untuned ANN model has an accuracy less than 50%. The dropout value was the only parameter hindering the model's progress. Upon tinkering and tuning this value from 0.5 until final value of 0.2, the model's learning accuracy rose dramatically beyond 95%.

This could mean that other configurations mentioned in Table 3.1 were the most adequate for the model. Nevertheless, as it will be discussed in Chapter 4, the hybridized model outperformed the benchmark model. Notably, the benchmark model also performed beyond the default 90% accuracy measure that was specified in Section 2.4.

[5]Discussed in Section 3.3.

Chapter 4

Results

4.1 Model Evaluation and Validation

For evaluation purposes, confusion matrices are constructed for both benchmark and final models. Figures 4.1 and 4.2 illustrate these constructions. Upon the realization of the confusion matrices, the precision, recall, F1-score and ROC curve areas for each model were computed. Table 4.1 presents a summary of these values for a total number of 228447 CDRs.[1]

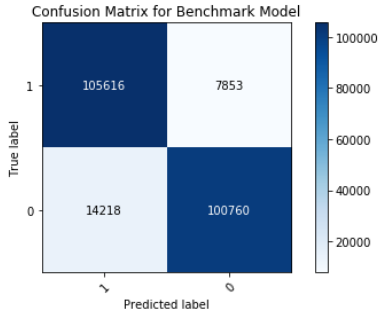

Figure 4.1: Benchmark model's confusion matrix.

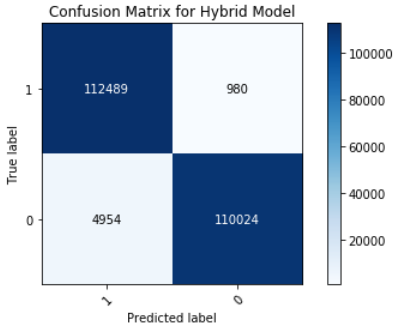

Figure 4.2: Hybridized model's confusion matrix.

[1]Recall, it's an acronym for Call Detail Records.

Table 4.1: Results Summary.

	Benchmark Model	**Hybridized Model**
True Positives	105616	112489
False Positives	14218	4954
True Negatives	100760	110024
False Negatives	7853	980
Accuracy	0.9034	0.9740
Precision	0.8814	0.9578
Recall	0.9308	0.9914
F1-Score	0.9054	0.9743

4.2 Justification

From Table 4.1, it is evident that the hybridized model outperformed the benchmark model, which also gave a good performance on the clustered data. In other words, true positive and true negative values, accuracy, precision, and F1-scores were higher for the hybridized model. Additionally, false positives and false negatives were significantly lower in the hybridized model. Since the precision of the model is of utmost importance, it is safe to say that the hybridized model performed better. It is worth noting that a better model can have an higher precision that is close to 1 but this is reserved for future improvements on this work. Computations for area under the ROC curve are also available. The results also show that the hybridized model made better predictions on the testing data when compared to the benchmark model. However, the free-form visualization section in Chapter 5 will discuss the ROC curve areas in more detail.

Chapter 5

Conclusion

5.1 Free-Form Visualization

The ROC curve is the only performance metric of the benchmark and hybridized models that has not been presented nor discussed till this point. Figures 5.1 and 5.2 provide complete plots and a magnified sectional view of the ROC curve respectively.

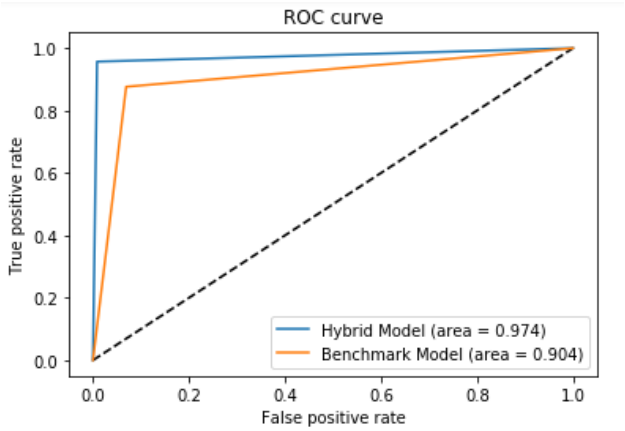

Figure 5.1: ROC curve for benchmark and final model.

As shown in both figures and the area calculations, the hybridized model significantly performs better than the benchmark model. The compared values are 0.974 and 0.904 respectively. This implies a 7% increase in predictive performance from the benchmark model.

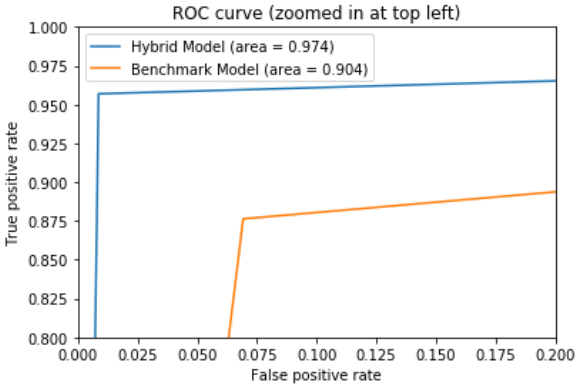

Figure 5.2: Magnified area of the ROC curve.

5.2 Reflection

This project proposes a novel approach for identifying likely RCS subscribers by adopting supervised and unsupervised machine learning algorithms. The most challenging parts of this work were the data preprocessing, implementation and evaluation of the hybridized model.

For data preprocessing, as presented in Section 2.1, there is a significant number of records containing NaN values. The ratio is strongly imbalanced such that sophisticated methods[1] could not be used. Defaulting NaN-affected records to explicit removal significantly reduced the available training dataset. As a result, more datasets were concatenated in order to build the main dataset.

Implementing the hybridized model required the construction and training of an ANN. At earlier runs, training was difficult and the validation loss and training loss stayed relatively at same levels across epochs. It took continuous tinkering to get the ANN model to improve its performance.

All the implementation logic would not have been very significant if effective evaluation techniques were not carefully designed. This is done in order to allow an unbiased comparison between the benchmark model and the final (hybridized) model. One could use existing frameworks that readily make these computations easier but runs the risk of not properly engaging the models. This is the reason that the confusion matrix was considered as the basis of all evaluations in Chapter 4 and beyond.

[1]Such as average computation of values across affected columns.

Moreover, the final model did perform much better than anticipated and could only get better from its current level. However, the GMM-based clustering technique that was used for defining the ground truth predictions is a major contributor to benchmark and hybridized model performances. Simply put, if the clustering was unreliable, the benchmark and final models would not have attained their respective performance levels.

5.3 Improvement

The final model could benefit from more datasets with more features that can possibly increase the number of customer segments to three. In this case, there would be predictions for most likely, likely and unlikely RCS customers. The unlikely segment would create a profile for customers who may be difficult to convince to use any value added service such as RCS. However, most likely and likely RCS customer segments can present insights into service usage patterns that can allow an MNO to create further segments. The further segmentation will allow an MNO to create RCS service profiles that are tailored to customers within their respective segments. In essence, growing beyond the subscription likelihood model to creating specific offerings through RCS service profiles.

Another possible way to improve on this work may be to create a finely-tuned ANN model on the clustered dataset. Since the ANN implementation is a simple ANN feed-forward classifier, it may be beneficial to investigate how other ANN designs will perform on this dataset. More importantly, other supervised learning models such as Ensemble methods can be implemented on the clustered dataset. This will be done in order to investigate the model which gives best performance on previously unseen data.

References

[1] Chris Piech. *Stanford CS221: K Means*. Available at: http://stanford.edu/~cpiech/cs221/handouts/kmeans.html. [Accessed 1 March, 2019].

[2] T. G. Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[3] Google Developers. *Classification: Precision and Recall — Machine Learning Crash Course*. Available at: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall. [Accessed 26 February, 2019].

[4] Google Developers. *Classification: ROC Curve and AUC — Machine Learning Crash Course*. Available at: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc. [Accessed 26 February, 2019].

[5] GSMA. *The RCS Ecosystem - Future Networks*. Available at: https://www.gsma.com/futurenetworks/rcs/the-rcs-ecosystem/. [Accessed 16 February, 2019].

[6] Kaggle. *Mobile phone activity in a city*. Available at: https://www.kaggle.com/marcodena/mobile-phone-activity. [Accessed 16 February, 2019].

[7] *RCS Universal Profile Service Definition Document*. RCC.71. Technical Specification. Version 2.2. GSMA. May 2018.

[8] D. Reynolds. "Gaussian mixture models". In: *Encyclopedia of biometrics* (2015), pp. 827–832.

[9] Scikit-learn. *Gaussian mixture models — scikit-learn 0.20.3 documentation.* Available at: https://scikit-learn.org/stable/modules/mixture.html. [Accessed 1 March, 2019].

[10] A. J. Smola and B. Schlkopf. "A tutorial on support vector regression". In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[11] O. F. Sogunle. *A Unified Data Repository for Rich Communication Services*. Master's Thesis. Rhodes University, South Africa, April 2017.

[12] Towards Data Science. *Ensemble Methods in Machine Learning: What are They and Why Use Them?* Available at: https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f. [Accessed 1 March, 2019].

[13] Towards Data Science. *Model Evaluation Techniques for Classification models.* Available at: https://towardsdatascience.com/model-evaluation-techniques-for-classification-models-eac30092c38b. [Accessed 26 February, 2019].

[14] Towards Data Science. *Support Vector Machine — Introduction to Machine Learning Algorithms.* Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Accessed 1 March, 2019].