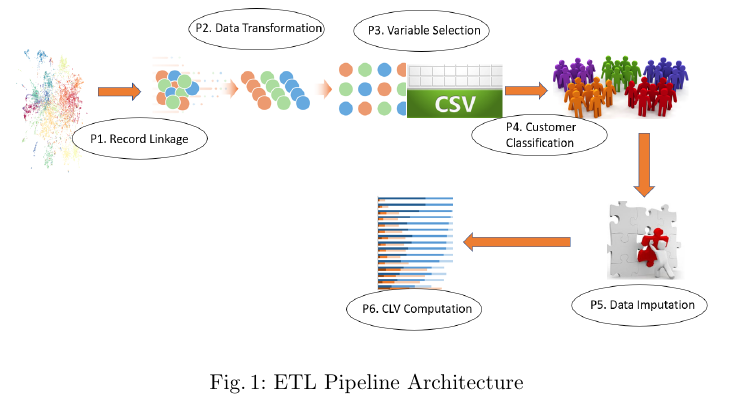
**Predicting Customer Churn for Insurance Data**

Michael Scriney, Dongyun Nie, and Mark Roantree

Dublin City University, Ireland



**3. Data Transformations**

In this section, we provide a brief outline of the Extract Transform Load (ETL) architecture but focus on those components which are novel to our architecture and crucial to imputing retention data. The dataset used in this work originated from our collaborator in the insurance sector. In this sector, transactions focused on selling policies and not on building customer profiles and thus, the Extract component in our architecture acquired approx. 500,000 insurance policies.

**3.1 System Architecture**

An ETL pipeline comprises a series of components extracting data from input sources, transforming the data to match the system’s data model, and loading into a data mart (data cube) for reporting and analysis. Our approach, shown in figure 1, is a specialised form of ETL [14], due to the specific requirements of the task (customer lifetime value) and the nature of the data. In particular, this work began with a dataset that was policy-focused and not customer-focused. In effect, it was not suited to analysis by customer. Thus, the first step involved a process known as record linkage where, upon acquisition, data was pivoted to be customer-focused, where a customer record contained 1 or more policies. This work was presented in [10] and, while it provided a more holistic customer record, the data was not suited to the imputation algorithms necessary to impute the missing CLV variables. In addition, the dataset was still unclassified in terms of customer types (good, bad, average).

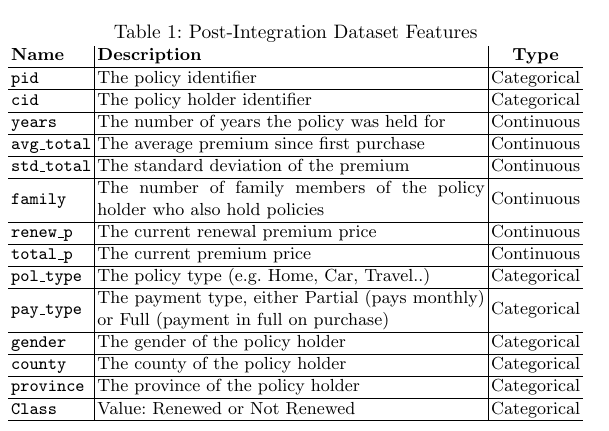
**3.2 Churn Analysis Data Transformation**

In this paper, we focus on components P2 and P5 from figure 1. The data used is initially based on two large imports: detail and aggregate. Detail provides a policy centric view year-on-year, recording the type, current and renewal premium for each policy. Aggregate is an aggregation of a unified customer record, detailing high level information on customers who hold policies generated during earlier work [10]. The large data imports are combined with other data sources within the warehouse to provide a dataset suitable for predicting customer churn. There are three processes involved in the transformation (P2) of a dataset suitable for churn analysis: Aggregation, Augmentation and Preparation. Aggregation constructs the initial per-policy view which provides information on policy renewals. Augmentation adds features to this dataset such as customer information and pricing. These two processes can be equated to the E and T processes within a standard ETL (Extract, Transform, Load) architecture. The final process Preparation provides a final transformation of the dataset so that is it ready for machine learning algorithms.

**Aggregation**. The goal of the first step is to construct a policy centric view containing those policies that may or may not been renewed. This involves a RollUp operation on the detail view to create an aggregated view containing the policy identifier (policy id), the number of years for which the policy is held (years held) and whether or not the policy was renewed (renewed). In total, the dataset used for our work contains 443,893 unique policies, of which 300,646 were not renewed with the remaining 143,247 renewed by the customer.

**Augmentation**. The next step is augmentation where views within the warehouse are integrated with the policy-centric aggregation. In total, seven additional views are integrated: policy prices, family policy holders, latest renewal premium, insurance type, location, payment method and gender. Policy Prices include the average premium and the standard deviation for premium, which can indicate the amount of variation in year-on-year premium prices. Family Policy Holders is the number of family members per customer who also hold policies with the company. Latest Renewal Premium is the latest premium for a given policy. Insurance Type is the type of insurance, which has four possible values: Private Motor, Commercial Motor, Home and Travel. Location is the county the customer resides in. Payment Method indicates if the premium is paid either in full or monthly. Finally, Gender relates to the gender of the policy holder. The result is a dataset with fourteen dimensions including a class label of Renewed or Not Renewed as seen in table 1 where: Name is the name of the feature; Description briefly describes the feature and Type indicates if Categorical or Continuous. For our evaluation, the dimensions representing unique identifiers (pid and cid) were not used.

**Preparation**. There are four steps in the preparation phase: cleaning, sam- pling, encoding and splitting. In this dataset, just 27 records were removed leaving a dataset with 443,866 rows. Determining whether or not a policy is renewed is, in effect, a classification problem. The class labels for each policy are Renewed or Not Renewed. As is common with real world data, our dataset has a class imbalance where 300,621 records are labelled "Renewed" and the re- maining 143,245 records are labelled "Not Renewed". This class imbalance can greatly affect classification results and three methods are generally employed to resolve this: undersampling, oversampling and synthetic sampling. As we have a large number of records for the minority Renewed class, *undersampling* was the method selected to address this issue. Using this method, 143,245 records with the class label Not Renewed were randomly chosen so that both classes had the cardinality. The downside to this approach is that some of the Not Renewed data could have increased the effectiveness our analysis. This is addressed in our conclusions. After undersampling, the dataset comprised 286,490 records, with an equal distribution of the classes renewed and not renewed (143,245 records each). The encoding step transforms categorical dimensions so they are ready for machine learning algorithms. The dimensions encoded were insurance type, payment method and county. The final step splits the data into training and testing sets using the 80/20 configuration.



**4. Algorithm Selection and Validation**

Due to the characteristics of insurance data and the fact that research into customer lifetime value is quite theoretical in nature, it was decided to use a range of statistical methods and try to determine what works best. In this section, we begin by presenting the set of algorithms used to impute the retention value (churn), then proceed to discuss the evaluation strategy and results and finally, we present a discussion on the results.

**Algorithm Selection.** The process of determining customer churn in any domain is generally a classification problem with two classes: Renewed and Not Renewed. The classification methods we employed were: Bernoulli Naive Bayes; Multinomial Naive Bayes; two types of support vector machines; two decision trees; and a series of artificial neural network (ANN) configurations. Support Vector Machines are used regularly in classification. For our experiments we used one Linear Support Vector machine to provide a baseline to our other methods. Two experimental configurations using Naive Bayes were employed, one using a Bernoulli model and another using a multinomial model which has been shown to have increased performance on binary data in some instances [9]. For both models, 100 different alpha values were used on each, ranging from 0.0 to 1 in degrees of 0.01. Two decision trees using the CART (Classification and Regression) algorithm were employed, the first using entropy as the splitting measure and the second using Gini impurity. For both approaches, a decision tree was created for each level of depth until the maximum depth was reached and at each depth, test data was used to obtain the accuracy of the tree. Artificial Neural Networks (ANNs) have seen extensive use in predicting customer churn due to their ability to model interactions between features that may otherwise go unnoticed. For our experiments, 20 different configurations of ANNs were constructed with various configurations of hyperparameters.

**Evaluation Metrics.** We now describe the evaluation metrics used to compare the different prediction models. The measures TP, TN, FP and FN are true positive, true negative, false positive and false negative respectively. The metrics are: Accuracy, Precision, Recall, Specificity and F1 score. Standard accuracy (percentage of correct classifications) is insufficient in evaluating a classifier but provides a useful baseline. The precision, recall and specificity metrics provide more information as to the actual performance of a classifier within classes. All model configurations will be validated using all 5 metrics.

**Results and Discussion.** We begin this section with an overview of the 4 different algorithms in isolation, reporting on their relative performances. We then take a comparative view across all algorithms, using different configurations for the more complex models. Unsurprisingly the Linear SVM show a weak performance with an accuracy of 0.754 and an F1 score of 0.76. However, this model was always intended as a baseline for our evaluation of other models. For both Naive model types, 100 different alpha values were used, ranging from 0.0 to 1 in degrees of 0.01. Interestingly, these changes had no effect on the accuracy score across model configurations. For the algorithm which incorporated entropy, the best performing tree had a depth of 11 with an accuracy of 88.82%. For the Gini-tree, the best performing depth was also 11 and with a very similar accuracy of 88.72%. The results of the top 5 performing configurations can be seen in Table 2, where id is the experimental id; epoch is the number of epochs; hlayer is the number of hidden layers; hnode is the number of hidden nodes,; tr ac is the accuracy of the training data,; tr l is the loss of the training data; te ac is the accuracy on the test data; and finally, te l is the loss on the test data. For all 5 configurations, a dropout rate of 0.02 was used. From Table 2, experiment 5 is the best performing with an accuracy of 0.888 on the test dataset. This model consisted of one hidden layer with 31 hidden nodes with a dropout rate of 0.2%. There were other models with increased training accuracy tr ac but are not shown as they have a lower te ac than 88.6% which is generally an indication of over fitting.

