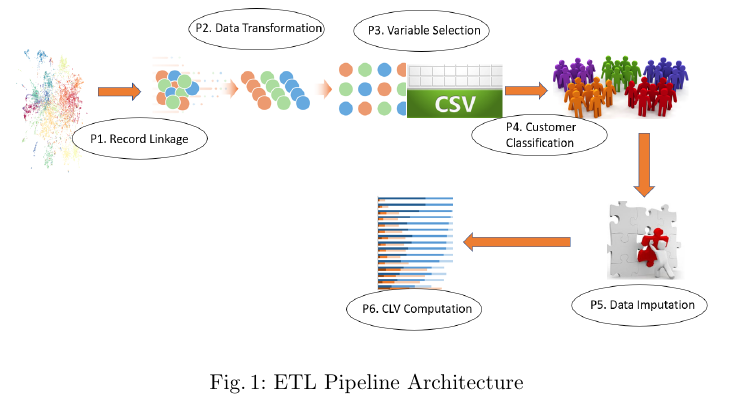
**Predicting Customer Churn for Insurance Data**

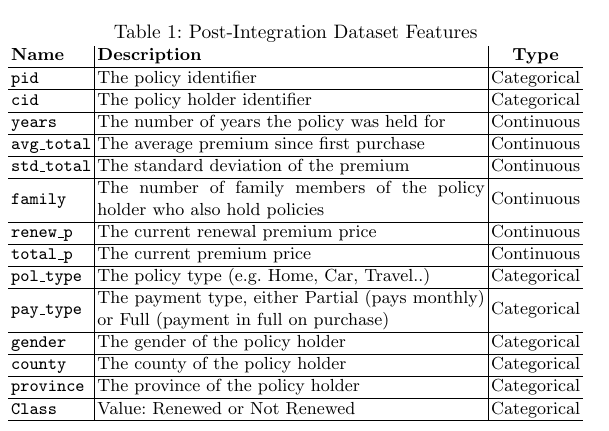
Michael Scriney, Dongyun Nie, and Mark Roantree

Dublin City University, Ireland

**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.



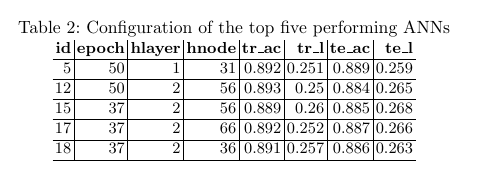
**4. Algorithm Selection and Validation**

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.



**Discussion.** Table 3 provides a comparison across all experimental model and parameter configurations. Method is the classification algorithm and configuration used; acc is the overall accuracy; err is the overall error; pre is the precision; rec is recall; spe is specificity; and F1 is the F1 score. Overall, most models performed well with 7 of the experiments achieving an accuracy > 0.88, with 6 of those having an F1 score > 0.89. Interestingly, the difference between the two decision tree methods (Entropy & Gini split) was so small (> 5 decimal places) that they effectively performed the same. The worst performing method was the multinomial Naive Bayes with an accuracy of 0.69 and an F 1 score of 0.678. In terms of accuracy, the best performing model was ANN-5 with an accuracy of 0.889. This configuration also achieved the highest F1 score with 0.893. On the other hand, both decision tree methods have higher precision (0.938 vs 0.920) and specificity (0.931 vs 0.914) scores. However, ANN-5 had a higher recall rate (0.867 vs 0.851). Between these three high performing configurations, ANN-5 had the highest number of true negatives (24487) while both decision tree methods had a higher number of true positives (26766 and 26767). By examining the NB-Bernoulli model results, there is a clear requirement for more in-depth statistics than accuracy alone. This method has an overall accuracy of 0.776 but there is a difference between the measures for recall and specificity (0.717 and 0.879 respectively), indicating that this method is better at predicting negative classifications over positive ones. If we examine the ANN configurations, while ANN-5 has the highest overall accuracy and F1 score, other methods show a higher value for specificity (the highest being ANN-17 with 0.929). However, ANN-5 shows the highest value for recall out of all neural network configurations indicating that it performs best when predicting positives classes. A high recall value necessitates a low false negative rate. Out of all methods employed ANN-5 has the lowest rate of fn, with 4, 070 records being classified incorrectly. The question of which model and configuration to use depends ultimately on the classification most important to businesses.

