Understanding, predicting and preventing Churn

William RJ Cooper

August 21, 2024

1 Executive Summary

In this report we outline a model that can predict whether a customer will leave a business or not (churn) with around 86% accuracy, using a few pieces of information readily available to most banks. Preventing churn is vital in most businesses, and predicting it is the first step. This report outlines a model that:

- can predict customer churn can be predicted with 86% accuracy
- can output a percentage indicating how likely the customer is to leave or stay
- has a low false negative rate (10%) meaning it is unlikely for the model to predict that a churning customer will stay

This model for predicting customer churn and assigning a confidence value can be used to offer pre-made tiered packages to customers, with scaling value for money meaning that customers most likely to churn can be offered the most lucrative discounts or products to keep them from churning.

2 Introduction

Churn, customer attrition, customer retention, all refer to the loss of customers from a business. Henceforth referred to as simply churn, minimising the loss of customers from any business is beneficial to any business for several reasons:

- It can cost 5 times more to acquire and sign a new customer than retain an existing one
- According to a report by Harvard Business School, 'increasing customer retention rates by 5% increases profits by 25% to 95%'
- 80% of your future profits will come from just 20% of your existing customers (Leading on the Edge of Chaos: The 10 Critical Elements for Success in Volatile Times by Emmett C. Murphy, Mark A. Murphy, 978-0735203129)

Any scheme that seeks to understand, predict and prevent customer churn may be able to increase profits, reduce costs and increase customer satisfaction.

Businesses often keep large amounts of data about their customers, and whether or not a customer has churned would also be easy to calculate. It is for this reason that this problem is perfect for data-driven solution as if there is a link between customer data and churn, these methods can find it.

In this research we choose two datasets containing various information about customers including whether or not they have churned. Even though these datasets are from bank customers and I am not in banking myself, there is no reason why the approach we use here could not be used for other datasets from businesses in other areas. This is why I chose this project, because customer churn affects almost every business.

The ultimate aim of this research is to create a model that, given certain customer attributes will predict whether a customer will churn or not. Our final model also has some benefits over other models that it also outputs a confidence value based on how likely it is to be correct, and can be interpreted as a value showing how likely it is that a customer will churn or not. This has the added benefit that this confidence value could be used to implement a tiered system to offer increasingly valuable products or discounts to customers who are likely to churn. The more likely a customer is to churn, the bigger the discount they can be offered to entice them to stay.

3 Methods

3.1 Datasets

In this section we present an overview of the steps taken from cleaning all the way to applying models to the data. For more detail, please see the source code here.

In our journey to predict customer churn we will focus on two datasets:

- Dataset A
- Dataset B

These datasets both contain attributes and information on their customers, and whether they have churned from the bank. These datasets have 11 attributes in common, namely:

- **Age** - Stored as a single number
- **Credit score** – A number between 0 and 999
- **Country** – Either France, Germany or Spain. Country is stored differently in both sets, one stores it as a string, and one stores as as three separate columns with a '1' indicating that country. We will use the latter strategy, so the former dataset will have to be modified to reflect this. This can be implemented trivially in Python
- **Gender** – Stored as 'Male' or 'Female' in one source, and as two columns in the second. Once again the approach of having two columns will be used.
- **Tenure** – The number of years the customer has been with the bank
- **Account Balance** – Customer bank balance
- **Number of Products** – Number of financial products the customer has purchased from the bank
- **Credit Card** – 0 or 1 depending on if they have a credit card or not
- **Estimated salary** – Number indicating salary
- **Active Member** – 1 if the customer is active and using their account, 0 if not
- **Customer Churn** – 1 indicating that the customer has left the bank, 0 indicating they are still a customer

There are certain attributes that are deemed irrelevant (such as the names of the customers), redundant (attributes that are composites of other attributes, such as the Age/Tenure product) and also attributes that are not common to both datasets. These attributes we omit, sticking to the 11 mentioned above. This forms the first step for merging the datasets, only keep attributes that are common to both and drop all others.

3.2 Cleaning

In this subsection we outline the methods for cleaning and merging then removing outliers and normalising the combined datasets.

3.2.1 Loading

On downloading and reading the csv files into dataframes, pandas gives a warning that columns have mixed types for dataset B. This will be addressed in the next section, for now we drop columns that are not present in both datasets to leave us with the columns we will use.

3.2.2 Invalid Numbers

On further inspection we see that the columns for credit score, age, tenure and number of products are all floating values. This is mostly due to formatting (These values are floating in the csv files) but there are also some invalid numbers in the age and the estimated salary columns. To solve this, we create some helper functions that convert a value into a floating number, and integer or a binary value. If the conversion fails, we replace the number with null. After we apply the appropriate conversion for each column, we then drop all entries containing null values effectively removing entries with invalid formatting. We then convert the columns to the correct types, integers for credit score, age, tenure and number of products. We also take this time to check for NaN values, but none exist so no further action is needed.

3.2.3 One-hot encoding

The Country and gender columns in dataset A store these attributes as string values. Since we are using statistical techniques that will not work with this format, we use one-hot encoding to convert the categorical data into numerical.

3.2.4 Renaming columns

The columns are labelled differently in both datasets, so we create a mapping of old columns names to new column names, and apply this to both datasets so columns representing the same data will have the same name.

3.2.5 Types

At this point we check the dtype of all the columns, which are floating for AccountBalance and EstimatedSalary, and integer for the rest as expected.

3.2.6 Merging

At this point the datasets have been cleaned, null values removed, and column names synced up so the datasets are concatenated into a single dataframe.

3.2.7 Shuffle

To remove any bias that may exist in entering the values, and that certainly exists by concatenating the two different datasets, we shuffle the dataset.

3.2.8 Outliers

Using the IQR test for detecting outliers on the following attributes:

- Age
- Tenure

- Credit Score
- Account balance
- Estimated Salary
- Number of products
- Has credit card?
- Is active member

And this identifies 6690 outliers namely from the 'Credit Score', 'Age' and 'Number of products' attributes, which we remove from the dataset

3.2.9 Normalising

Finally, since most of the techniques require the data to be normalised, we do this for all attributes (Except customer churn which must remain categorical) by subtracting the mean and dividing my the standard deviation for all columns.

3.3 Visualising

Now the data is cleaned and normalised we can apply 2 component and 3 component PCA analysis on the data (we do not use the 'Churn' columns in PCA) and plot the data on 2D and 3D scatter plots, respectively, and colour by churn. This gives us a way to visualise the multi-dimensional data, and hopefully allow us to see a difference between churned and non-churned data. Visualising PCA will give us a clue as to how easy the data will be to classify, as if churned and non-churned data has little overlap, then this is a strong indicator that the data will be easy to classify.

3.4 Models

Now that the data is ready to be analysed, we split the data into training and validation sets and apply a range of models to the data. Our goal in this is of course to predict churn, which comes in two categories, churned and not churned. Because of this, the types of models we use are classifiers and regression models. In this analysis we try 9 different models and compare their effectiveness, namely:

- LightGBM (LGBM) - A fast gradient boosting framework based on decision trees
- Logistic Regression (LR)
- Random Forest classifier (RF)
- Gradient Boosting classifier (GBC)
- MLP classifier (MLP)
- Ada Boost classifier (ADA)
- K-nearest neighbours (KNN)
- Linear Discriminant Analysis (LDA)
- Convolutional Neural Network (CNN) - Keras implementation of a sequential CNN model

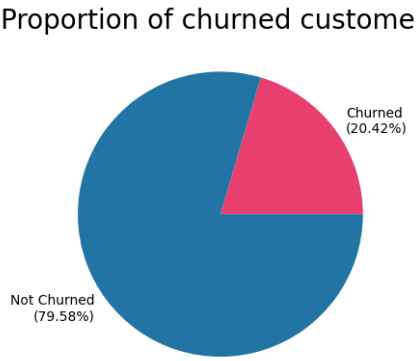


Figure 1: Pie chart showing the proportion of the dataset that is churned (20.42%) against the proportion that is not churned (79.58%)

4 Results

Our first graph simply shows the ratio of churned to not churned customers in the dataset:

As we can see the dataset is heavily biased towards customers who have not churned. This is not surprising as, at least for successful companies, churn should be low. This bias is very important to keep in mind however, since optimising models to predict churn could result in finding local solutions which simply guess 'not churn' and do not predict at all. Because of the bias in the dataset, a model which trivially predicts 'not churned' would have an accuracy of around 80%. Later on we discuss how we can look out for models that do this, and we even see an example of a poor neural network which takes this approach but looks to have a high accuracy.

For the next few charts we compare the churned vs non-churned samples over various attributes including age, activity, number of products purchased, account balance, estimated salary, credit score and tenure. First we see the age attribute, which has the most obvious differences for churn and non churned customers:

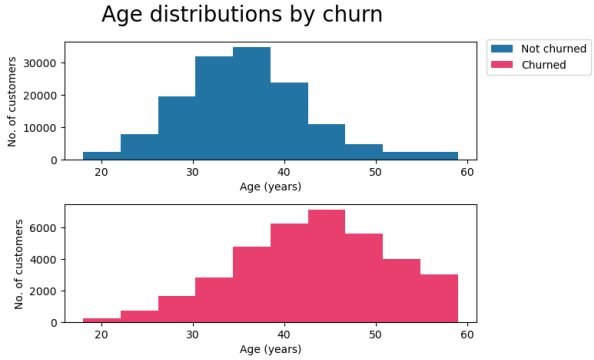


Figure 2: Historgram showing the distribution of age for non churned customers that has a positive skew, and a second histogram showing the distribution for churned customers which skews negatively.

As we will see not all plots show any correlation, but in this case there seems to be a strong link between age and churn. The not churned histogram skews left, showing a younger age, whereas the churned histogram skews right, possibly meaning that churned customers are older than not churned. Next we have customer activity:

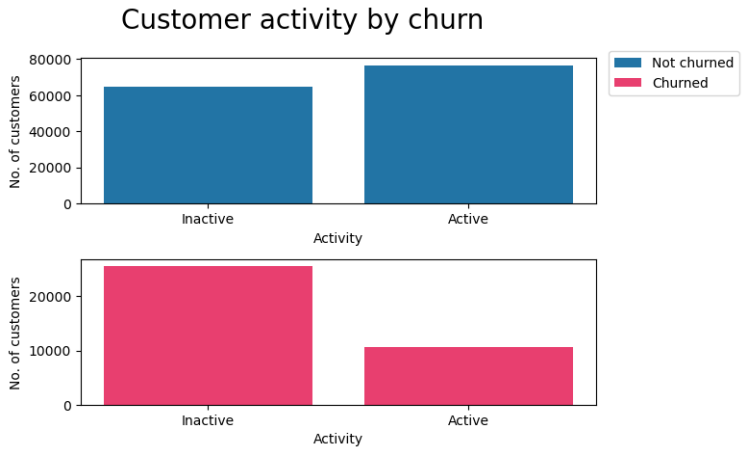
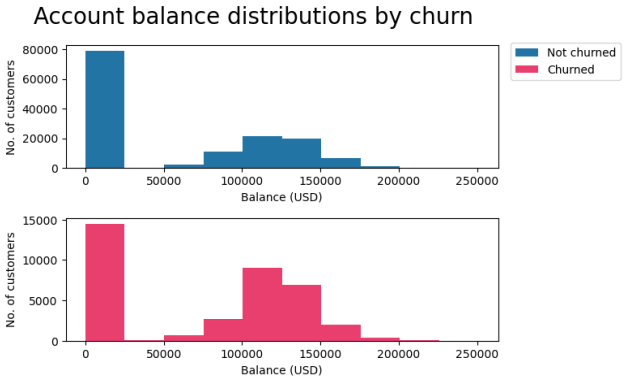


Figure 3: Historgram showing the distribution of age for non churned customers that has a positive skew, and a second histogram showing the distribution for churned customers which skews negatively.

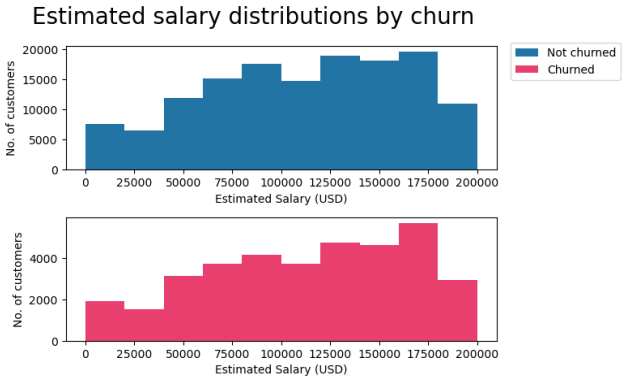
Here we see that vastly more users are inactive in the churned group which could suggest that inactivity leads to churn. However, it is important to note here that we cannot say which way the correlation flows, and indeed in this case it is more likely that the explanation is that customers who leave become inactive. If the causation does flow this way, customer activity may not be an accurate way to predict churn, as churn may come after inactivity. Up next is a plot of the number of products purchased by customers:

Figure 4: Historgram showing the distribution of age for non churned customers that has a positive skew, and a second histogram showing the distribution for churned customers which skews negatively.

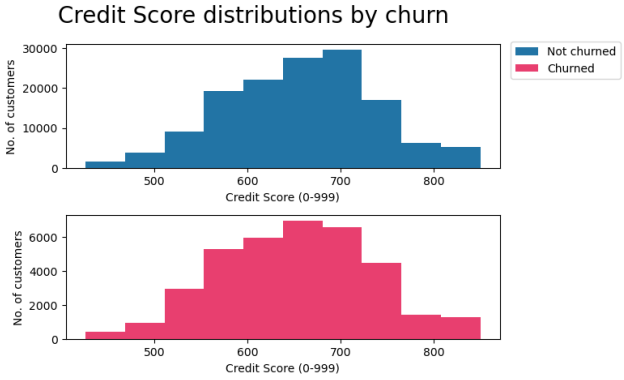
Here again we see big differences between churn and not churned customers, namely that it is much more common for not-churned customers to have purchased more than one product. Purchasing more than one product could be a big indicator that a customer won't churn. Finally we have plots for the other features, however these plots do not seem to show many differences between churn and not-churn so warrant no further discussion:



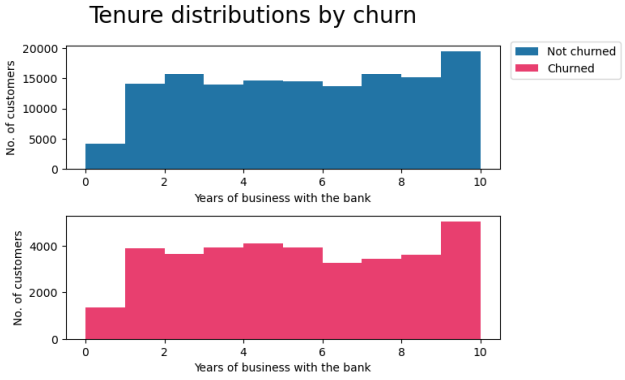
(a) Account balances for churn and non-churned customers



(b) Estimated salaries for churn and non-churned customers



(c) Credit scores for churn and non-churned customers



(d) Tenures for churn and non-churned customers

Figure 5: Frequency distrubutions by churn for account balance (a), estimated salary (b), credit scores (c) and tenure (d) showing no real difference between churn and non-churned customers

The final section of our describtive statistics analysis summarises the plots we have seen so far with a correlation matrix:

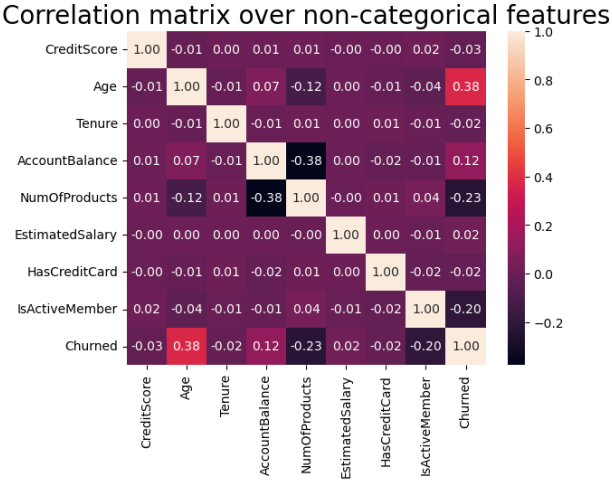
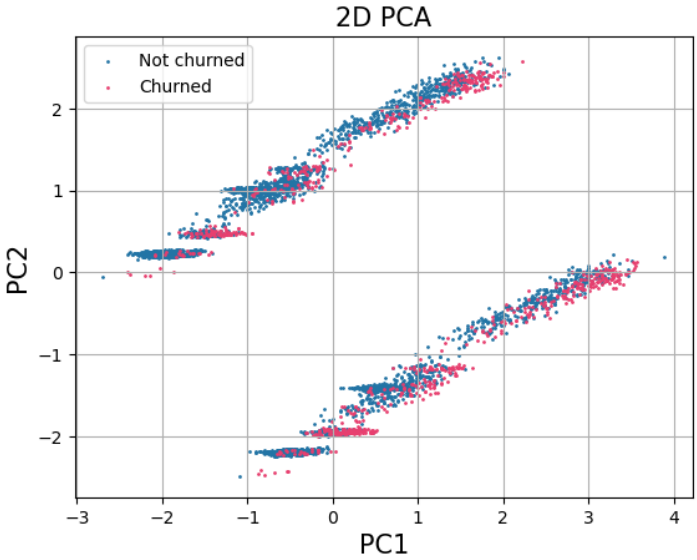


Figure 6: Correlation matrix for all attributes showing a correlation between churn and age, account balance, number of products and activity

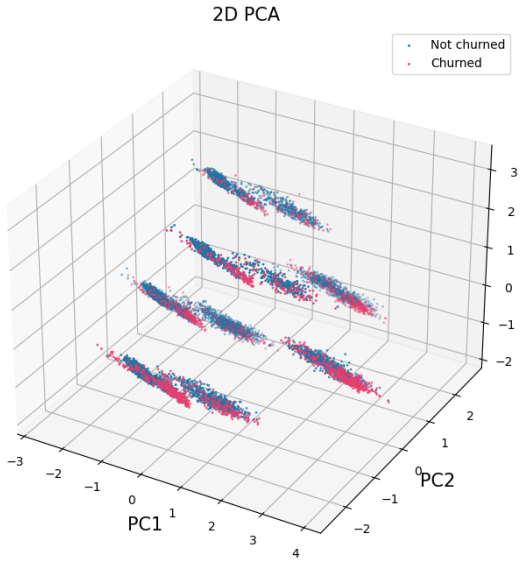
This correlation matrix confirms what we have seen so far, namely a correlation with churn against Age, number of products and activity, it also shows a correlation with tenure too.

These results are extremely promising, as they show that the data is correlated and connected and that optimisation may be used to predict churn.

Before we start optimising, our last step is to use PCA to reduce the dimensionality of the dataset to 2 or 3 dimensions, then plot these points on 2D and 3D scatter graphs, coloring points based on churn. Here are the results:



(a) 2D PCA visualisation with points coloured by churn



(b) 3D PCA visualisation with points coloured by churn

Figure 7: 2D (a) and 3D (b) PCA visualisations showing no visible pattern between churn and non churned data points

Since we removed churn from the PCA analysis, any differences in the churned vs not churned points would entirely be from the features and not churn itself. However, as we see from the plots, there are no discernable differences and there is a great overlap in the data.

As PCA has not given any extra insight into our dataset, we move on to applying our 9 models on the dataset to see if there is any way to extract meaning from the data. We summarise our results with confusion tables from predictions for all 9 models:

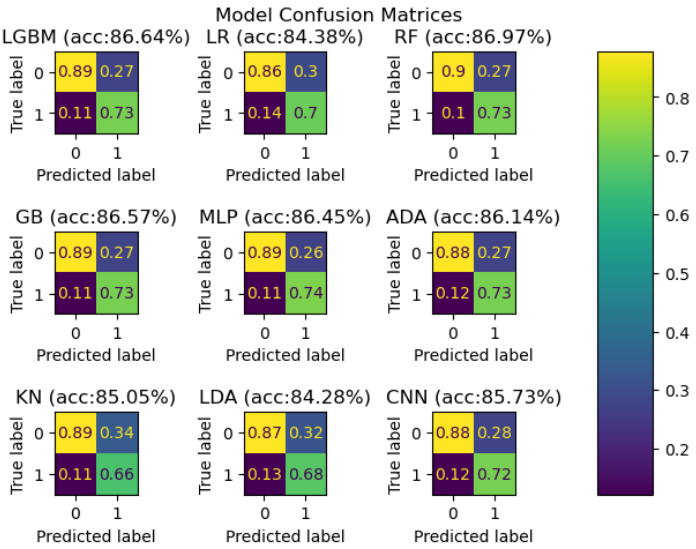


Figure 8: Confusion matrices (by percentage prediction) for all 9 models used, displaying accuracy and model type too

There is a lot going on in this one plot, so lets start with the LGBM model confusion matrix:

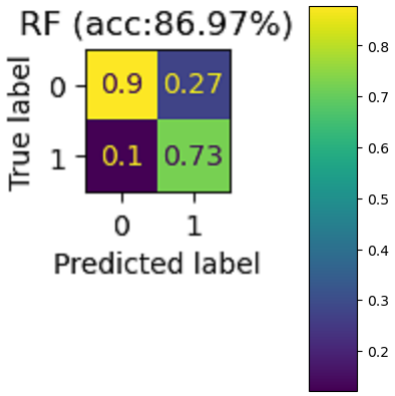


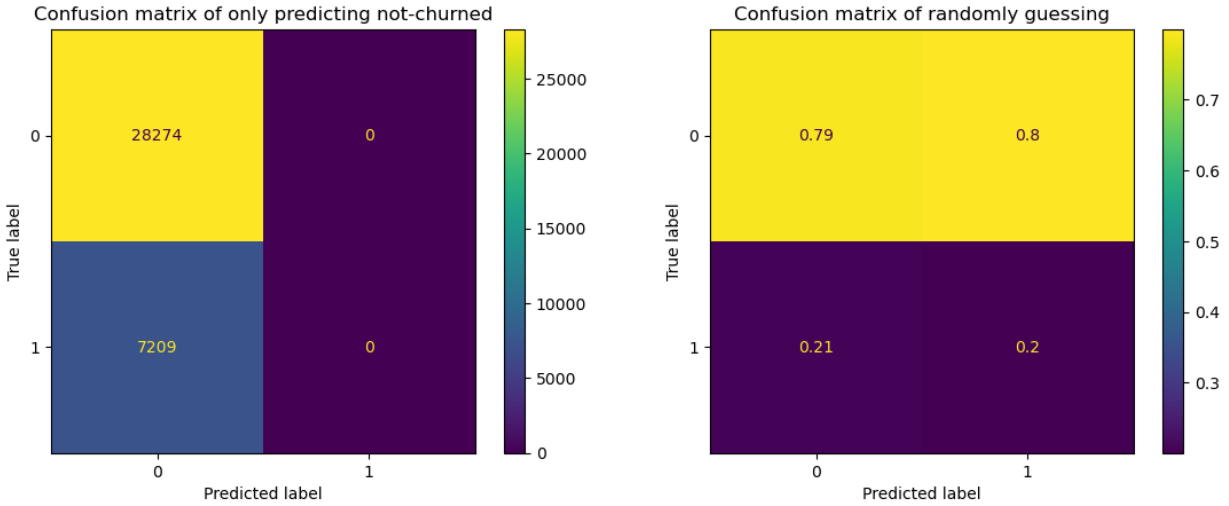
Figure 9: Confusion matrices for Random Forest (RF) which has the highest accuracy (86%) the best false negative rate (10%) and the best true positive accuracy (90%)

First we note that the confusion matrix does not show integer values in the table, but decimals. This is because we show the percentage based on the predicted label. So in the RF example, when the model predicted a '0' is was correct 90% of the time, and when it predicted a '1' it was correct 73% of the time. Presenting the matrix in this was is important as the dataset is very biased towards not churn, so showing the results as percentages removes this bias. It also shows us the percentage of time the model is a true positive. This is important as for models that guess randomly this value will be about 20%, and for models that predict 0 this will be 0%. Getting a high percentage (73% in this case) is enough to convince us that this is not guesswork. We also see the overall accuracy of this model, 86%.

While some models are slightly better than others, they are all around 86% accurate, even the models that are fundamentally different show no higher than 87%. This, I suspect is the upper

bound of the dataset itself, to have multiple models so closely agree on an accuracy must mean that they cannot extract any more meaning from the data.

For comparison, we now show two confusion matrices of trivial models, the first guesses uniformly randomly for churned/not churned, and the second always guesses not churned:



(a) Confusion matrix of a model which guesses 'not churned' every time, showing a true positive rate of 0% but an overall accuracy of around 80%

(b) Confusion matrix of a model which guesses uniformly randomly, showing a true positive rate of 20%

Figure 10: Confusion matrrices of poor trivial models that may have high overall accuracies (80% in both cases) but very poor true positive rates (under 20%)

5 Conclusion

To get an idea of which models might be better than others we train classifiers, regression models, a neural network model and random forest classifiers on a training set then evaluate their effectiveness based on three main factors:

1. Overall accuracy on the validation set
2. True positive accuracy
3. Does the model output a yes/no or a more helpful confidence value?
4. False negative rate

Checking the true positive accuracy is very important because the dataset is about 80% composed of customers who have not churned, and 20% who have. This means that a model that always guesses not churned would be 80% accurate. This model is of course highly undesirable, and it would have a true positive accuracy of 0%. Getting this true positive accuracy as high as possible is what makes an accurate model and proves to be difficult. The third factor is important because it gives more information on the customer (how likely they are to churn or not churn) which can be used to prescribe an outcome more tailored to that customer (Heavy discounts, personalised products, reaching out personally via customer service, etc.). Finally the false negative rate is important because it shows us how many customers the model will predict will stay when in fact they churn. Since this is the worst case outcome, minimising this is important.

From our results we see that the best for accuracy is the RF model, and the best true positive is the RF model. Finally since the CNN is the only model that outputs a confidence value, we see that

RF and CNN models are our best bet for the three questions. To give the best of both, I propose a hybrid model which combines the best of both by using the result of RF with the confidence value from CNN, if it agrees with the RF result, if not, no confidence value is provided.

The best model for false negatives is RF. This represents the cases where the model predicts a customer will not leave when they do. This is certainly a worst case scenario as it will result in customers leaving without the model predicting it. Needless to say this metric should be as low as possible, we are fortunate that the rates are low, at 10% for the RF model.

Finally we propose a list of next steps:

1. Create a python application that applies the model we have chosen and takes customer data and outputs a value indicating a prediction on churn
2. Implement this algorithm in the company to predict churn on all customers and also apply it anytime the data changes for a customer (increased age, another product purchased, etc.)
3. Use the prediction to decide the best course of action for customers who are predicted to churn. If the chosen model outputs a yes/no answer, then choose and apply an action for customers predicted to churn. If the chosen model gives a confidence value, create and apply a tiered solution that maps confidence values to prescribed actions
4. Finally keep track of churn percentages before and after the model is implemented to evaluate the models' success

6 Evaluation

Overall the accuracy of the models are reasonably high, and this shows that it is possible to predict churn from the data. However it would be ideal to get an accuracy above 86%, and here are some steps we could take to achieve this:

1. Include more customer features, such as how many complaints a customer has had, how they rated any customer service or support tickets, and instead of a yes/no for activity give a numeric value (i.e. time since last active)
2. Include more diverse customers, such as customers from other countries, religions, etc.