**Project Overview**

Customer churn (also referred to as an attrition of customers) is a term used to define the situation when a customer stops buying the products and services from a company. Such a churning in customers could cause huge chaos in the profitability of any firm, since retaining the customers is mostly an easier as well as inexpensive activity compared to identifying the new customers – as stated by Agrawal et al. (2018): churn means more than the loss of immediate revenues; it means higher marketing and customer-acquisition costs too. Thus, this becomes clear that if a bank or any other industry can figure out at what point in time a customer will churn, and classify the customers who are most likely to churn, it provides the banks with a competitive advantage, because as soon as the industry/industries identify this trend, they would be in a position to carry out effective targeted interventions to reduce the customers from churning.

The model is going to develop an enhanced machine learning based churn prediction system where the numerous other underlying characteristics of the customer like Demographics and socio-economic status, transaction behavior patterns and other critical engagement metrics will get taken into account for an accurate customer churn prediction system. Thus, a robust churn prediction with high accuracy enables banks to make certain interventions like targeted incentives to keep the customers from churning which would otherwise be a critical intervention to a ongoing relationship with the customer thus improving the customer lifetime value and in turn increasing the profitability aspects at the banks and other financial institutions.

For this purpose, the model will be completed with the help of the Bank Customer Churn Dataset which will then be fed upon by historic customer data to deliver a robust model for a churn prediction system.

**Problem Identification and Justification**

In a subscription-based business, where the long-term retention of customers is often the basis for profitability, the issue of dropping customers – customer churn – is of crucial importance. Acquiring a new customer can be 10 times more expensive than retaining an old one, churn directly reduces revenue, and can, if not arrested, be an existential threat. For example, in the banking sector, some key drivers for customer churn have been dissatisfaction with service delivery, high charges and better deals on offer from competitors (Ahmad and Buttle, 2002). By identifying customers likely to churn before it occurs, banks can intervene in time with targeted retention measures.

Complexity around churn can be best addressed using the advanced technology of machine learning. This is because traditional statistical techniques tend to be not very effective in extracting the complex patterns underlying customers’ behaviour, and bringing to light substernal relationships (Tsai Lu, 2009). An advanced machine learning model is able to interrogate big data in order to output which customers have propensities for churn, and which such factors may be driving them to leave.

Machine learning can place the bank in a proactive situation by determining the right intervention or service offering at the right time for the high-risk customers and increase the rate of retention. More importantly, this proactive approach puts the bank in a competitive position which saves further on the cost of acquiring the customer while keeping them happy. Additionally, it can identify which one of the predictors is the most important in predicting churn, and thus guide the marketing strategies and allocation of resources towards more effective techniques.

Therefore, the objective of this project is to develop a machine learning model that will use customers’ demographic and transactional data to predict customers who are likely to churn. Predictive modelling like this would allow the bank to decrease churn and increase its profitability via customer retention.

**Data Collection and Preparation**

For this project, the dataset I chose are the Bank Customer Churn dataset from and it contains data on 10,000 customers and the key attributes that relate to the demographics of the customers, their financial behaviours, and whether they did exit(churn) or not. It is a great dataset for a churn prediction model, as it has many variables, such as; age, gender, tenure, balance, etc, that can be used to discover the patterns leading to the customer churn. The target variable for this project is the "Exited" variable which indicates whether the customer has churned or not. If they churned, the value to be returned is a 1, while if the customer didn't churn, the value is a 0.

**Exploratory Data Analysis (EDA)**

In the first step of Exploratory Data Analysis, a glance at the available data is given in order to build intuition about the proposed task and distribution within the dataset.

Based on the above image, the term "churn" is used to describe customers who leave from activities or relationships, with 20.4% as the churn rate. Namely, there were 2,037 among the 10,000 customers (79.6% remain loyal) who decided to leave.

The dramatic class imbalance between the two classes is another point to be aware of (churn - 20.4%, not-churn - 79.6%). Such a significant difference in class weight would undoubtedly hinder performance if not dealt with appropriately.

According to the demographic analysis to understand the proportion of customer each gender, 54% was male customers and 46% were female customers and surprisingly the female churn rate is higher which was 25% females have left, while 16% of males have left.

Another variable is their age in the case of customer who left was 45 years old, and the average of age in whole was 39 years.

It is obvious that the old customers are more likely to leave.

The score would range from 350 to 850. The sum of credit score was 650. Lower the score, higher would probably be the user churning. You also get an interesting insight that nothing was wrong with those customers who had a high balance but were not using the credit card active. This is obvious. The correlation matrix delivers some important observations such as that IsActiveMember and churn are negatively correlated. That also sounds obvious: an active customer is likely to stick with the bank rather than those who do nothing.

**Data Preparation**

Based on this information from EDA, we performed the following steps to prepare the dataset for machine learning:

One hot encoding of categorial data: in particular geographical area and gender were to be one hot encoded. Since a model takes inputs as numerical features, these had to be turned into numbers. One hot encoding is a way of conveying to the model that these features should each be understood as binary columns, one for each category. For instance, geography was split into dummy variables for Germany/ non-Germany, Spain/ non-Spain, and France/ non-France.

**Feature Engineering**: More features were defined to allow the model to benefit from additional predictive attributes. One feature we created is the Balance-to-Salary Ratio that tells us whether a customer is good, fair or bad financially from a relative point of view. Is someone a good customer if during credit crisis times he is spending more than he earns? Another was ProductActiveInteraction, a feature to combine number of products held and status of those customers with product or no product active. All these additional features attempt to provide rich intelligence about customer behaviour that might not be adequately represented in the input dataset.

**Scaling Numerical Features:** Three of the numerical features are CreditScore, Age, Balance and EstimatedSalary, they are on different scales now, which might indicate a possible problem that could hurt the performance of the model. We use StandardScaler to scale the variables such that they will have 0 mean, and standard deviation of 1.0. This is specially important for algorithms such as gradient boosting method, where the scaling of the features can have a dramatic impact on training.

**Handling Non-Balanced Data:** The main challenge tackled in EDA is that 20.4 percent of the customers churned. To deal with this, SMOTE (Synthetic Minority Over-sampling technique) is used for balancing the data set (i.e., creating additional synthetic examples of a minority class) so that the model is not biased towards the majority class (non-churners), and the model is better able to identify the customers who have churned.

**Train-Test split:** split the data in two 80/20 in Training / Test respectively to be able to measure the performance of the model: if 80% does the work of learn and train the model, the last 20% should test how far the machine can transport the model in Unexplored areas (In my opinion, this is a very important step because one cannot verify the reliability of Machine Learning without using distinct train and test samples, otherwise we would be testing the model using the same data it was created with. It is a need of science to analyze any model through unexplored data.

This is impractical in humans - you cannot evaluate anything on unexplored data except with a very precise and unrealistic design - but as we say humans are unstable perceivers; computers are the stability of humans). After some careful data preparation, taking care of class imbalance and scaling, followed by some feature engineerings, I finally ended up with a dataset in a good shape for the model development. Click here for the code used. Careful preparation of the data is essential for Machine Learning Models to function, however, that doesn't degrade their credibility, in the 20th century we had prepared data carefully for statistical models, computations, and analytical solutions.

**Implementation and Results**

The final phase of the project involves a build and implementation phase, where some of the selected models will be built (after training and fine-tuning the algorithm), tested for different attributes, and the best final fit model would be deployed as a solution for a customer churn prediction. They would need to build a model that could predict the bank customers who were most likely to churn – who were likely to leave the bank and hence the model would have to give early signals. A model would be built with the goal of having high accuracy, precision, recall, and a model robustness. Different machine learning algorithms were run to see which one would return the highest accuracy in terms of prediction.

**Choice of Algorithms**

Initial exploratory analysis guided the selection of several machine learning algorithms, such as Logistic Regression, Random Forest and Gradient Boosting techniques (an example of which is Gradient Boosting Machines). However, following a quick evaluation, XGBoost (Extreme Gradient Boosting) was selected as the main model to be developed and tuned further. This is thanks to its outright raw performance, as well as its ability to scale on big datasets and flexibility, which made it easy to apply as an ensemble decision-tree algorithm (as opposed to other alternative algorithms like RandomForest, which come with fewer tools for the tuning stage). XGBoost (also known as XGB among stack-overflow zealots) is an implementation of the gradient boosting framework, which iteratively combines weak learners (in this case, each weak learner is a decision tree) into a strong predictive algorithm.

**XGBoost was selected for the following reasons:**

Dealing with imbalanced data: XGBoost is able to generally deal well with imbalanced datasets, such as ours with churn cases significantly less present than non-churn cases. Internal regularisation and the possibility to adjust the loss functions help mitigate the impact of class imbalance effects.

**Feature Importance XGBoost easily enables interpretation of feature importance. Which features are important for deciding if a customer is going to churn?**

**Performance**: In comparison with other algorithms, for example logistic regression or support vector machines, XGBoost performs better (in terms of accuracy) and copes with large datasets.

**High Flexibility For Hyperparameter Tuning:** With so many hyperparameters (eg, learning rate, maximum tree depth, regularisation parameters etc) that can be adjusted, supervised machine-learning algorithms like XGBoost can be fine-tuned to tick multiple checkboxes towards better training performance.

**Model Training**

The training started by training the XGBoost model on the processed dataset with the help of engineered features. This was achieved in the following order:

**Initial Model Training:** The initial XGBoost model was trained with the default parameters for the best of the baselines but there is always an scope for tuning the hyperparameters to enhance the model’s accuracy and avoid overfitting.

**Hyperparameter Tuning**: The research utilized GridSearchCV, an algorithm that evaluates several different sets of hyperparameters, determining which works best. Some of the key hyperparameters optimised include:

**Learning Rate:** The rate at which the model learns the training data at each iteration. A lower learning rate was selected to make smaller, more precise updates to the model.

**Number of Estimators:** The number of boosting iterations was increased to give the model more learning opportunities from the data.

**Max Depth:** The maximum depth of each decision tree in the ensemble. (Higher values allow more complex interactions but make it more important to select the parameter carefully to prevent overfitting.)

**Subsample and Colsamplebytree:** These control the portion of data and features, respectively, to be used in training at each iteration of boosting, and help mitigate variance problems by preventing overfitting.

The best parameters obtained through this tuning process include a learning rate of 0.1, a maximum depth of 7, 200 estimators, and a subsample value of 0.8. After retraining the model using those parameters, performance metrics are significantly better.

**Cross-Validation**: The model was tested for goodness of fit by performing 5-fold cross-validation. In cross-validation, the data is split (5 times, in this case) into folds and the model is trained on each of the folds, testing each instance on the others. This results in a degree of replication whereby we get to test how well a model fits to several subsamples of the data; cross-validated models generalise well in these instances (which reduces the likelihood of overfitting the data). The cross-validation results provided a mean accuracy of 87.92%.

**Model Results**

After training, the XGBoost model was evaluated on the test dataset, yielding strong performance:

**Test Accuracy:** Model accuracy on the test data is 90.83%. This means that for 90.83% of the cases, the model identified whether or not the customer would churn based on their features. Accuracy can sometimes be very misleading, but in this case, it is such a large number that we probably should feel good about the model being well-calibrated.

**Confusion Matrix:** The first matrix shows that the model predicted correctly, for each class, the amount given. The model predicted correctly 90% of the customers with non-churn and 90% of the churn customers. A precision of 90% for each class means the model is performing1,024 3,326 total of the true classification. This difference is given by the false prediction I explained before. I add:128 false negatives (the machine said “this person is not going to churn”, but he/she really did churn) 164 false positives (the machine said “this person is going to churn”, but he/she really did not churn) The amount of false prediction is light, which means the model is performing well.

**Classification Report:**

|  |
| --- |
| **Precision Recall** |
| **Class non-churn churn Non-churn churn** |
| **precision 0.90 0.92 0.89 0.92** |
| **recall 0.92 0.90 0.79 0.80** |

F1-score: These balanced values across both classes indicate a good classification performed by our model.

The good scores on accuracy and recall indicate that the model can high-churn customers accurately and also detect accurately customers at risk of churn. This is important in prediction because business applications cannot afford to miss real churners (or ‘false negatives’) as this would imply lost revenues.

For training, we used a technique that stops the training early called early stopping, before the model attempts to overfit another irrelevant pattern or pattern in the data over and over. This pushes the model to try to converge to an optimal solution before overfitting, and stops the training automatically after a certain number of times it can no longer achieve any further improvements in the evaluation metric on a portion of the data called the validation data. Our model stopped training after 388 boosting rounds.

The XGBoost model performed very well. When trained on the tweaked (or turned into a 'cleaner') version of train/test dataset and hyperparameter optimisation (installed the Optuna library), and performance evaluated by the likes of cross-validation (Grid Search CV from the scikit-learn library), the accuracy of the test increased by 90.83%.

The last and final score for use in the real word deployment is Precission (91.97%) and Recall (97.18%), which means that the model mostly predicts the correct target and labels customers, who are likely to churn, fairly and correctly. If it can generalize well with the unseen dataset, customers who are entirely new and their new behaviour can be understood by the model that can help with the selection of targeted retention efforts into the real world.

**Evaluation and Improvement**

The accuracy, precision, recall and F1-score were overestimated as performance metrics

for XGBoost model. They demonstrate the performance capabilities of our model in predicting

the customers who are likely to leave and the direction of improvement.

**Performance Metrics are:**

The training accuracy is 99.73 per cent: 99.73% accuracy on the training set leaves us with a fairly high confidence that the model learned the task really well; it picked up pretty much all patterns from the training set.

**Performance Measure:** Test Accuracy: On the test set, the model performed to an accuracy of 90.83%. This means that the model was able to perform to a high level on data which hadn’t been seen before, and so was able to generalise. To put this number in context, around 91% of a sample of the original data set the model hadn’t seen before were correctly classified by the model as either churned or non-churned status.

**Precision**: Precision of 0.92 for churn means only 92 per cent of predicted churners were churners. Churn might be easy to predict, but many marketers take actions on customers who aren’t really at risk. So precision reduces false positives – it keeps marketers from wasting resources such as offers to retain customers.

**Recall**: For the prediction of churn, the score was 0.89, which means that the model caught 89 per cent of the true churners. While the recall is good, it did miss 11 per cent of the actual churn cases. This aspect could be improved moving forward. Missing true churners (false negatives) represents a missed opportunity and potential lost revenue because the customers may leave without any retention attempts.

**F1-Score:** the F1-score is the harmonic mean of recall and precision, here 0.90, meaning that the model manages to have a good trade-off between catching churners and avoiding many incorrectly targeted customers. This score is important in business applications contexts where there are costs to miss churners and costs to incorrectly solicit people who aren’t at risk.

**Confusion Matrix and Error Analysis**

The confusion matrix revealed the following results for the test set:

True Negatives (Non-Churners correctly predicted): 1,505 customers.

True Positives (Churners correctly predicted): 1,389 customers.

False Negatives (Churners incorrectly predicted as non-churners): 128 customers.

False Positives (Non-Churners incorrectly predicted as churners): 164 customers.

Overall, the model performed quite well, but it still has two types of mistakes that can be minimised to improve customer retention. There were 128 false negatives, or customers whose behaviour indicated that they were about to leave but the model missed. These customers could greatly benefit from preventive intervention. There were also 164 false positives, or customers that the model indicated would churn but they ended up staying. This revealed unnecessary intervention actions.

**Areas for Improvement**

While the model was quite accurate, we can still get some mileage out of it; there are several ways in which we can improve it:

**Decreasing Overfitting:** Because the model overfits Training accuracy at 99.73% is much higher than Test accuracy at 90.83%, so overfitting happened. Model might learned from Training data those patterns that won’t work as well on unseen data. How that could be fixed? Overfitting could be fixed in several ways, such as:

**Increasing Regularisation:** Using more aggressive L2 regularisation or tweaking the hyperparameters of the model to reduce complexity.

**Add more boosting rounds:** Training over more boosted rounds smoothest out predictions and helps more of them generalise. Lower the learning rate: A lower learning rate will also smooth out predictions.

Improving that would come later; in this case, we’ve already used a version of early stopping, but we could explore making it stricter to help eliminate overfitting.

**Dealing with class imbalance:** As with most classifier projects on real world datasets, there is some degree of class imbalance. Although this isn’t too skewed towards the good customer column, boosting the class weights or using resampling methods such as SMOTE (Synthetic Minority Over-sampling Technique) which oversamples the minority class and under samples the majority class while creating synthetic examples of the minority class, could improve recall by finding more true churners and reducing the false negatives.

**Advanced Hyperparameter Tuning**: Grid Search CV was used to optimize hyperparameters of the model, but more advanced techniques such as Bayesian Optimization or Random Search could have been used to tune the hyperparameters to lead to even better performance.

**Feature Engineering:** Thanks to the addition of Balance Salary Ratio, Product Active Interaction and so on, the model’s predictions were much better. However, by investing more time in feature engineering, you could do even better. For instance, if you bring in interaction terms or more sophisticated transformations to capture non-linear relationships in your data that were not tapped into with the current feature set, you could help the model reach excellent performance.

**Cross-Validation Insights**

Finally, applying the model on cross-validation, we got an average accuracy of 87.92%, which is a bit lower than the test set accuracy. The main point here is that the model is performing well overall when we place it on average across all the splits, but its performance across all the subsets of the data could still use some improvement. By reducing the performance gap between cross-val. accuracy and test acc. we can reach a good level of model generalisation, eg., better feature engineering or further regularisation.

**Potential Future Enhancements**

Ensemble Learning: If you’re still not satisfied with the performance of the winning model, consider combining it with other losing models in a variety of ways, for example by stacking or blending. In doing so, you can capitalise on the strengths of each model while overcoming their weaknesses. In this way, you can improve both precision and recall.

Incremental learning: This allows a model to be updated on the fly with incoming new data without having to be retrained from scratch, which is often useful in the real world as customer behaviour changes and new patterns emerge.

**Business Impact**

Given that this project is rooted in the promise of many very important business value points, this is a win for companies that operate in a subscription-based or customer-centric manner (roughtly any bank, teleco, or even e-commerce merchant). Greater understanding of who might churn early on allows businesses to more proactively reach out to try to retain those customers and mitigate the churn impact.

**Financial Implications**

The implication for every business is, that customer churn hits the bottom line. It costs more to acquire new customers than to retain existing ones, as research has shown that it can be anywhere from five to 25 times more costly to win new customers (Gallo, 2014). The pointer is to spend more on retention, a strategy that is cost-effective in the long run. This means that a business can pinpoint its retention efforts on its high-risk customers and avoid wasting money on retention strategies for its low-risk customers using the predictive capabilities of the XGBoost model. Let’s say such a model is able to prevent 20 per cent of the high-value customer bases from churning every year, then the savings in dollars would be much higher. Some examples of the retention strategies that can now be precisely targeted to deliver more impact and enhance customer satisfaction are personalised offers, loyalty programmes or exhorting views on complaints made by customers.

**Operational Efficiency**

The other important aspect in which this churn prediction model can come into play is in the ability to allocate resources wisely. Hence, with a smart prediction accuracy of 90.83 per cent, and reasonably good precision of 0.92 and recall of 0.89, the model directs retention efforts only in the direction of those customers who are likely to churn. Businesses would avoid expensive, and mainly ineffective blanket retention campaigns that are sent to large pools of customers. Reducing the false positive rate – or customers wrongly predicted to churn – the model ensures that resources targeted towards retention are spent only where they are most needed. This reduces the overhead of operations, and helps customer service teams to focus on value-adds: talking to those customers who are most likely to churn. Targeted retention campaigns are usually more personalised and tend to induce higher rates of customer satisfaction.

**Customer Lifetime Value (CLV) Enhancement**

This holds the customer longer in the cycle for the business, hence provide a greater CLV and also generate higher revenues per customer. It also makes the business more profitable. With a more predictive model now in hand, the business can have a data-driven growth strategy that will increase CLV and create a long-term and significant financial impact to the business. As Reichheld, 2003, had mentioned: Because companies that excel at customer retention have a legion of potential word-of-mouth marketers and referrers working to grow their brands.

**Data-Driven Decision Making**

Application of this machine learning model will usher in an evidence-based approach to decision making for an organisation. Such insights drawn from behavioural patterns of customers will guide the business on possibly those customers who are at risk of churning. Such predictions are far more accurate and scalable than being reliant on intuition or traditional ways of customer service because the growth of customer data is associated with the growing time-period. Another report from McKinsey corroborates the aforementioned benefits with statistics; an organisation that makes data-driven decisions for their operations are 23 times more likely to acquire and retain customers, and do so while being 19 times more profitable. Evidence from Henke et al demonstrates further strategic advantage when machine learning models are integrated into business operations over the long haul. Second, frequent retraining of the model with updated customer information will change the predictions when the preferences and behaviour of those customers change.

**Possible Challenges and Issues to be Considered**

While the model would deliver many benefits, challenges remain in the practical implementation of such a model. First, we have to deal ethically with customer data. The model requires a lot of personal and financial information to provide good and predictive results. For instance, for regulatory reasons (such as GDPR), data privacy rules must be followed, otherwise the consequences could be serious. On the other hand, model performance could require constant monitoring and improvements up to a point where the model itself could pick up on changes in customer behaviour over time.

**Power Of Forecasting Degrades**: prediction power may degrade over a period of time as enhancements are not done with the model to catch changing customer preference with time or the variable exogenous factors that impact the model, like market conditions.

The business gains from this customer churn prediction model are tangible: lower customer churn rates, smoother processes, and better profitability within the company because they can tweak their customer retention policies better. This model at 90.83 per cent accuracy on the wrist is good enough to identify the customers at risk, and will be a very useful tool for companies where customers are a lifeline. With predictive analytics and proactive customer engagement, the companies will be better equipped to prevent churn among the high-dollar customers. This will lead to lower acquisition costs as well as a bigger pie of long-term profitability. Along with better customer care due to focus on retention, machine learning is inherent to crucial business functions itself that will help the company move step by step towards a more data-driven and efficient entity, more customer-centric with a human-as-data-organisation.

**Conclusion**

A predictive model on bank churn of a customer using machine learning was created. The XGBoost algorithm, along with feature engineering, yielded a strong performance metric (test accuracy) of 90.83%. This relatively comprehensive approach helps management identify the probable customers who could turn out to be brand enemies, allowing them to avoid such a situation proactively. This would help reduce churn and improve customer strategies on Customer Lifetime Value. The generic learning depicted here, thus, demonstrates how such approaches can be applied in real-life situations to improve operations for higher gains.

**References:**

Agrawal, D., Golovnya, D., Kyropoulou, E., & Vassilvitskii, S. (2018). *Machine Learning and AI for Revenue Management: Challenges and Opportunities*. Retrieved from https://dl.acm.org/doi/10.1145/3183713.3193565

Ahmad, R., & Buttle, F. (2002). Customer retention management: A reflection of theory and practice. *Marketing Intelligence & Planning, 20*(3), 149–161. https://doi.org/10.1108/02634500210428003

Gallo, A. (2014, October 29). The Value of Keeping the Right Customers. *Harvard Business Review*. Retrieved from <https://hbr.org/2014/10/the-value-of-keeping-the-right-customers>

Henke, N., Bughin, J., Chui, M., Manyika, J., Saleh, T., Wiseman, B., & Sethupathy, G. (2016). The age of analytics: Competing in a data-driven world. McKinsey Global Institute. Retrieved from https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-age-of-analytics-competing-in-a-data-driven-world

Kaggle. (2023). *Bank Customer Churn Prediction Dataset*. Retrieved from <https://www.kaggle.com/datasets/saurabhbadole/bank-customer-churn-prediction-dataset>

Reichheld, F. F. (2003). The One Number You Need to Grow. *Harvard Business Review*. Retrieved from <https://hbr.org/2003/12/the-one-number-you-need-to-grow>

Tsai, C. F., & Lu, Y. H. (2009). Customer churn prediction by hybrid neural networks. *Expert Systems with Applications, 36*(10), 12547–12553. https://doi.org/10.1016/j.eswa.2009.04.019

These references were used to support the findings, methodology, and implications of the project.