STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, September 2024, pp 1397–1408.
Published online in International Academic Press (www.IAPress.org)

Optimization of the K-Nearest Neighbor Algorithm to Predict Bank Churn

Sonia Akakpo, Patrick Dambra, Rachell Paz, Timothy Smyth, Frank Torre, Chunhui Yu*

Department of Mathematics, Farmingdale State College, State University of New York, USA

Abstract Bank churn occurs when customers switch from one bank to another. Although some customer loss is unavoidable, it is important for banks to avoid voluntary churn as it is easier and cheaper to keep an existing customer than to gain a new one. In our paper, we train and optimize a machine learning algorithm, specifically a k-nearest neighbors algorithm, to predict whether or not a customer will leave their bank using existing demographic and financial information. By giving banks a reliable method for predicting whether or not a customer will churn, they can prioritize certain groups in an effort to increase retention rates. We compare the accuracy of our algorithm to other types of machine learning algorithms, such as random forest and logistic regression models, and increase the accuracy of the k-nearest neighbor algorithm by optimizing the k value used in our model, as well as utilizing 10-folds cross-validation. We determine the most important attributes and weight them appropriately. After optimizing this model, we are able to predict with 85.72% accuracy whether or not the customer will churn.

Keywords Bank Churn, K-Nearest Neighbors, Random Forests, Optimization, Logistic Regression, Machine Learning

AMS 2010 subject classifications 68T01, 68U01

DOI: 10.19139/soic-2310-5070-2098

1. Introduction

1.1. Bank Churn

Bank churn is the departure of customers from their bank, usually in favor of a different bank. According to [8], this has become more common recently, as new communication technology such as the internet has increased consumer awareness of their options. Churn can be divided into three categories: expected, which is customer loss due to the passage of time; involuntary, which is customer exit because of misconduct or failure to meet their obligations; and voluntary, when customers leave by choice. Voluntary churn can be further subdivided into incidental churn, such as when a customer moves to a region not served by a bank, and deliberate churn, when customers leave due to dissatisfaction with the institution. Reasons customers may churn voluntarily include poor quality of service, loss of trust in the bank, high prices, and inconvenience. It is important for banks to develop strategies to predict and prevent churn because keeping existing customers is easier and less expensive than gaining new ones [21]. Specifically, the cost of keeping an existing customer can be from five to twenty-five times lower than gaining a new one [4]. This paper analyzes the behavior of 10,000 customers of the ABC multinational bank to predict the likelihood that a particular customer will churn and how likely a group of customers is to churn. Moreover, this paper aims to identify the most important features that will help banks optimize their products to entice high risk customers to stay with them.

*Correspondence to: Chunhui Yu (Email: chunhui.yu@farmingdale.edu). Department of Mathematics, Farmingdale State College, SUNY. 2350 Broadhollow Road, Farmingdale, NY 11735.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

1.2. Literature Review

We began by looking at the number of publications on customer churn in recent years. We searched for bibliographic terms such as "*KNN* bank churn prediction" in google scholar. Between 2003 and 2023, we found about 2,150 references across articles and reviews published in journals, books and conferences. Figure 1 below emphasizes the recent uptick of interest in this topic from 2003 to 2023. The growing attention to customer churn likely stems from banks recognizing the significance of customer retention. As more banks emerge over time, competition intensifies. Additionally, the rapid advancement of machine learning techniques undoubtedly enhances the analysis of customer data.

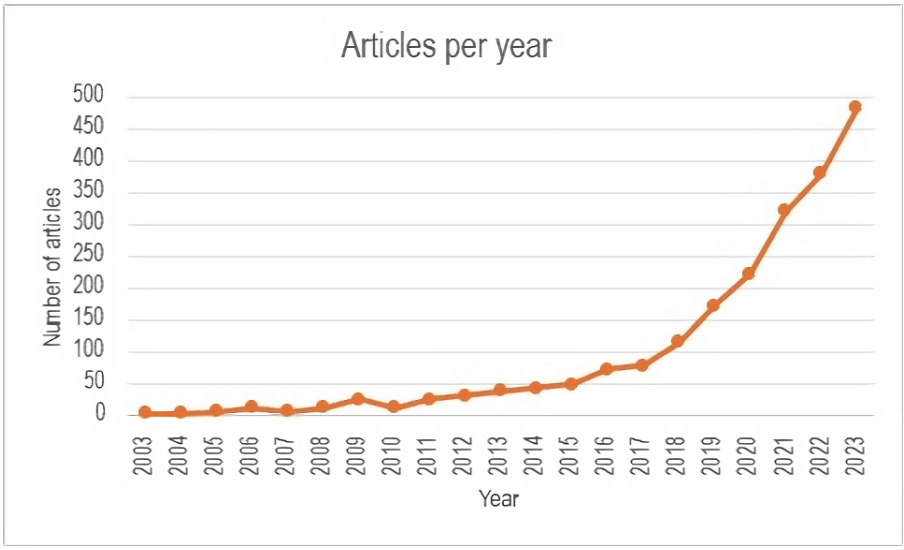

Figure 1. Number of Bank Churn Articles Per Year

Bank churn can be voluntary or involuntary, as stated in the introduction. When customers churn, they usually go to another bank, which creates competition among banks. Machine learning methods aim to make predictions based on data by identifying the most critical features in customer churn and the patterns in banks that are likely to lose customers. Some patterns include poor quality service, lack of technologies, unsatisfactory interest rates, and lack of variety of services. Customer churn is very detrimental for banks. Therefore, it is essential for banks to accurately predict the likelihood that their clients may churn by identifying the most important features and optimizing their products to retain their customers. Generally, a client is unlikely to leave their bank if their account is very active [10].

[7] compared the performance of Exponential smoothing, Prophet, Hybrid Arima-Arch model, *KNN* model, and Long-Short term memory for time series forecasting, which concluded that Exponential smoothing and *KNN* are well-suited for short term forecasting of inflation rate in the U.S. and *KNN* performed the best accuracy. [22] found that a new probability model Markov Chain & Clusters (MC & CL) demonstrates better classification results, comparing with other popular models to classify sequences, including *KNN+DTW*, when there are few training data available, because the new MC & CL model has fewer free parameters than the other popular models. [9] compared the performance of a newly proposed combination model with genetic algorithms, autoencoder, and *KNN* models to predict employee turnover with single *KNN* and DeepAutoencoder-*KNN* models. This study found that the combination model performed significantly better for a low experimental sample size data. [19] compared different machine learning methods, and also evaluated with cross-validations, k-fold validation, or leave-one-out validation. They found that random forest with k-fold validation gives the best accuracy. Both cross-validation methods yield similar results with random forest on an extensive dataset but produce different results when run on a small dataset. However, k-fold validation is preferred due to its computational advantages. They also observed that

the *KNN* model could have performed better if the dataset was not noisy and had missing values. [20] proposed an approach to predicting customer churn using machine learning methods such as *KNN*, logistic regression, k-means clustering to segment customers, decision trees, random forests, and support vector machines. After running their data using the previous methods, they found that the random forest model has a higher accuracy rate. In contrast, logistic regression has the lowest accuracy. [21] also found a similar comparing result for a credit card customer dataset. [2] applied four machine learning methods: Logistic regression, Random forest, Neural Network and SVM on imbalance and balance dataset. They found that the performance of the Random Forest algorithm increased when the hyperparameters were adjusted, which is known as the Improved Random Forest algorithm (ERFA). Meanwhile, [18] proposed a random oversampling (ROS)-voting (random forest [RF]-gradient boosting machines [GBM]) model which has a better classification and prediction success compared to other classic methods. To prevent the issue of imbalanced dataset, the study of [3] focused on the data processing methods. The performance of a model yields better results when data are preprocessed before training the data.They used feature engineering methods to identify the most important features for banking industry to avoid imbalance dataset issues. On the other hand, [17] proposed a strategy to identify and select key indicators to predict the churn of institutional insurees, which can help insurance companies formulate effective marketing strategies for policyholders' churn.

The objective of this present paper is to identify the most important attributes in bank customer churning and to optimize the *KNN* algorithm to achieve a better prediction result in terms of accuracy. We chose to focus on *KNN* because it is simple to implement and computationally cheap to train. These are qualities which we believe make it a useful model for businesses interested in using machine learning techniques.

2. Methodology

This section provides a description of the machine learning algorithms used in the study. While *KNN* is the algorithm we seek to optimize to achieve a higher accuracy score, we use two other algorithms (Logistic Regression and Random Forest Classifier) to compare with our newly optimized accuracy score.

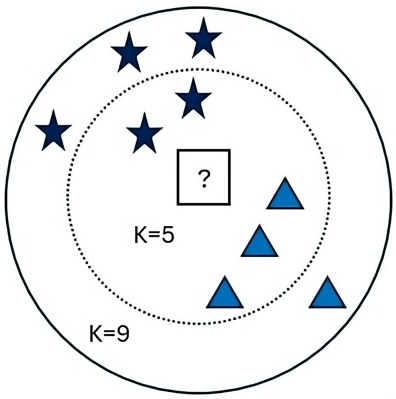

Figure 2. KNN Algorithm Diagram

2.1. K-Nearest Neighbors

K-Nearest Neighbors (*KNN*) is a nonparametric model; that is, a model that cannot be characterized by a bounded set of parameters [13]. By having each hypothesis we generate retain within itself all the training examples, it can then use all of them to predict the next example. This is also referred to as instance-based learning or memory-based learning. *KNN*, when given a query x_q, will find the k examples that are nearest to x_q [13]. This can be denoted as $NN(k, x_q)$. For classification, we first find $NN(k, x_q)$, then take the majority vote of the neighbors (since our

classification is binary). As a way of avoiding ties, an odd number is chosen to be k. When determining a maximum k value, we use the heuristic $k = \lfloor\sqrt{n}\rfloor$, where n is the sample size.

Figure 2 shows how different k values determine the classification of the new, unknown data. As seen, when $k = 5$, the new data has five neighbors, of which the majority are triangles; therefore, the new data point is classified as a triangle. However, changing the value of k from 5 to 9, we see that the new data point is surrounded by a majority of stars, and so it would be classified as a star rather than a triangle.

Paraphrasing the proof written by Devroye et al., [5], we seek to satisfy the condition that $k \to \infty$ as $n \to \infty$ in such a way that $\frac{k}{n} \to 0$. The heuristic used does satisfy this condition because $k = \lfloor\sqrt{n}\rfloor \to \infty$ and $\frac{k}{n} = \frac{\lfloor\sqrt{n}\rfloor}{n} \approx \frac{1}{\sqrt{n}} \to 0$. This theorem can be taken as an asymptotic result, meaning that as more observations are collected $n \to \infty$, the classification error rate of the *KNN* classifier L_n will almost certainly converge to the minimal classification error rate L* one can hope to obtain.

$$L_n \xrightarrow{a.s.} L^* \iff P\left(\lim_{n\to\infty} L_n = L^*\right) = 1$$

The k-nearest neighbors model typically uses the Minkowski distance (or L^P norm) which measures the distance from a query point x_q to an example point x_j defined as

$$L^P(x_j, x_q) = \left(\sum_i |x_{j,i} - x_{q,i}|^P\right)^{\frac{1}{P}}$$

In our study we had $p = 2$, which is equivalent to the well-known Euclidean distance formula:

$$L^2(x_j, x_q) = \sqrt{\sum_i (x_{j,i} - x_{q,i})^2}$$

Python provides a machine learning library known as *Scikit-Learn*. Using this library, we were able to run our computational analysis utilizing the KNeighborsClassifier, which is a supervised neighbors-based learning (or instance-based learning) method similar to that described above [11, 15]. While the basic implementation of KNeighborsClassifier uses uniform weights that assign a value to a query point based on a majority vote, our study determined higher weights for the top three features considered to be most predictive.

2.2. Logistic Regression

Logistic regression is a linear model for classification despite the name. Linear classifiers can either make predictions using a discontinuous function or a continuous, differentiable function. Using a discontinuous function will announce a confident prediction of either 1 or 0, while using a continuous function allows for a gradated prediction [13]. Logistic Regression aims to soften the discontinuous threshold function using the logistic function:

$$Logistic(z) = \frac{1}{1 + e^{-z}}$$

The output of this function will provide the probability of the classification [13].

As stated in *Scikit-Learn* [11, 14], the case of binary class logistic regression; that is, assuming the target value y_i is in the set $\{0, 1\}$ for data point i, the Logistic Regression model will predict the probability of the positive class $\hat{p}(X_i)$ as

$$\hat{p}(X_i) = Logistic(X_i w + w_0) = \frac{1}{1 + e^{-X_i w - w_0}}$$

2.3. Random Forests

Before describing Random Forests, we must explain how decision trees work. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression [11, 16]. DTs represent a function

that will return an output or decision when given a vector of attribute values [13]. These values can be either discrete or continuous. A decision is made after running a sequence of tests with each node in a tree corresponding to a test of an input attribute's value. Each leaf node will then specify a value that will be returned to the function.

The following is mathematical formulation of Decision Trees stated in *Scikit-Learn* [11, 16]:
Given training vectors $x_i \in \mathbb{R}^n, i = 1, \ldots, l$ and a label vector $y \in \mathbb{R}^l$, a decision tree recursively partitions the feature space such that the samples with the same labels or similar target values are grouped together.
Let the data at node m be represented by Q_m with n_m samples. For each candidate split $\theta = (j, t_m)$ consisting of a feature j and threshold t_m, partition the data into $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ subsets

$$\begin{aligned} Q_m^{left}(\theta) &= \{(x, y) | x_j \leq t_m\} \\ Q_m^{right}(\theta) &= Q_m \setminus Q_m^{left}(\theta) \end{aligned} \tag{1}$$

The quality of a candidate split of node m is then computed using an impurity function or loss function $H()$, the choice of which depends on the task being solved (classification or regression).

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H\big(Q_m^{left}(\theta)\big) + \frac{n_m^{right}}{n_m} H\big(Q_m^{right}(\theta)\big)$$

Select the parameters that minimizes the impurity.

$$\theta^* = \mathit{argmin}_\theta G\big(Q_m, \theta\big)$$

Recurse for subsets $Q_m^{left}(\theta^*)$ and $Q_m^{right}(\theta^*)$ until the maximum allowable depth is reached, $n_m < min_{samples}$ or $n_m = 1$. If a target is a classification outcome taking on values $0, 1, \ldots, K-1$ for node m, let

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I\big(y = k\big)$$

be the proportion of class k observations in node m. If m is a terminal node, (the *Scikit-Learn* method call) "predictproba" for this region is set to p_{mk}. In our paper we use the Gini measure of impurity:

$$H\big(Q_m\big) = \sum_k p_{mk}\big(1 - p_{mk}\big)$$

Random Forest (RF) is an amalgamation of several different decisions trees during its training. All the predictive results from the trees are collected to calculate a final decision. Our study is using RF as a classifier; as such, it will use the mode of the classes for its final prediction. This is known as an Ensemble Technique. RF also handles what is known as Feature Importance. Feature Importance is calculated as the decrease in node impurity weighed by the probability of reaching that node. The number of samples that reach the node divided by the total number of samples gives the node probability. A higher value corresponds to a more important feature [12].

In our implementation of RF we use the default version in *Scikit-Learn* which has 100 Decision Trees and uses Gini Impurity.

Algorithm:

- Input dataset with N features and n number of trees.
- A random forest is generated.
- Split the data into training and testing data.
- Fit the data.
- The model then predicts the probability of a customer churning based on majority voting.

2.4. Dataset

Our data, consisting of 10,000 examples of customer data from the ABC Multinational Bank, was sourced from the website Kaggle.com [1]. Each item has 14 variables. The first three, Row Number, Customer Id, and Surname,

were ignored, as they have no predictive value. We use the 12 features shown in the tables to predict customer churn. The numerical feature Number of Products represents how many banking services they were signed up for (i.e., savings account, checking account, credit card, etc.). In training our model we created features in order to show the *KNN* model that there's a relation between the two. We also have the target variable Exited which is 0 if the customer did not churn, and is 1 if the customer did churn.

Numerical Features

Features	Min	Max	Mean	SD
Age	18	92	38.92	10.49
Credit Score	350	850	650.53	96.65
Tenure (Years)	0	10	5.01	2.89
Number of Products	0	4	1.53	0.58
Estimated Salary	$51,002.11	$199,992.48	$100,090.24	$57,510.50
Balance	$0.00	$250,898.10	$76,485.90	$62,397.41
Engineered Features	**Min**	**Max**	**Mean**	**SD**
Balance/Estimated Salary	0.00	10,614.66	3.88	108.34
Tenure/Age	0.06	0.56	0.14	0.9
Credit Score/Age	14.09	46.89	17.87	5.38

Categorical Features

Features	Values	Description
Gender	Male Female	The gender of the customer
Geography	France Spain Germany	Where the customer is located
Is Active Member	Yes No	Whether or not the customer uses the bank often

Figure 3. Feature Description

2.5. Features

Machine learning uses feature variables and a target variable. In our case of predicting bank churn, the target variable is whether or not the customer will leave the bank. We have a binary classification problem where 0 means the customer did not leave, and 1 means the customer left the bank. We use our feature variables to predict if a customer leaves the bank. These variables include age, credit score, and others. We can think of our algorithm as a function where x is a vector of the features and y is the target variable, binary 0 or 1.

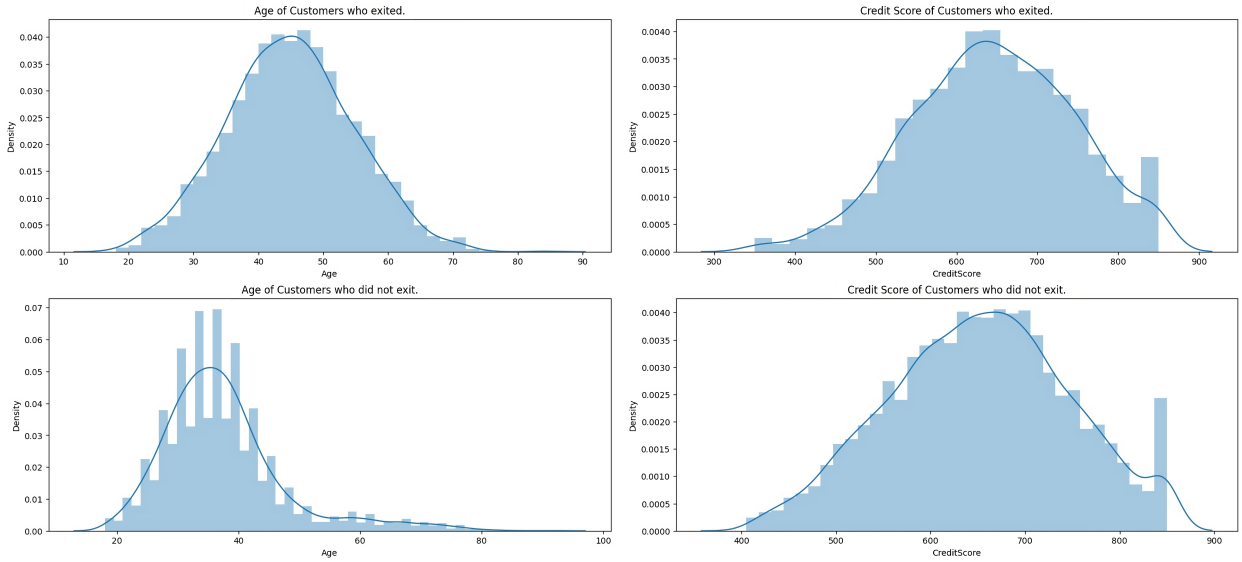

Figure 4. Feature Distribution Examples
Top/Bottom: Customers who Exited/did not Exit
Right/Left: Age/Credit Score

2.6. How is Our Model Trained?

Machine learning models are trained using a large collection of prepared data. We separate the data into a feature vector x, and a label y. We show the model as many examples of feature vectors and labels as possible so the model can see the pattern in the feature vector to accurately predict the label [5]. In practice, to evaluate the quality of our model, we have to test how the model performs on data it hasn't seen before. To do this, we split our data into a training set and a test set. We show the model the training set so it can make predictions and then test its accuracy on the test set which it has not seen before. This way, we're testing how the model would likely perform in a real setting. To score the model, we determine how examples were correctly predicted and take a percentage score. Figure 5 shows how the *KNN* algorithm works. We compare the accuracy score computed to the accuracy score of the RF model.

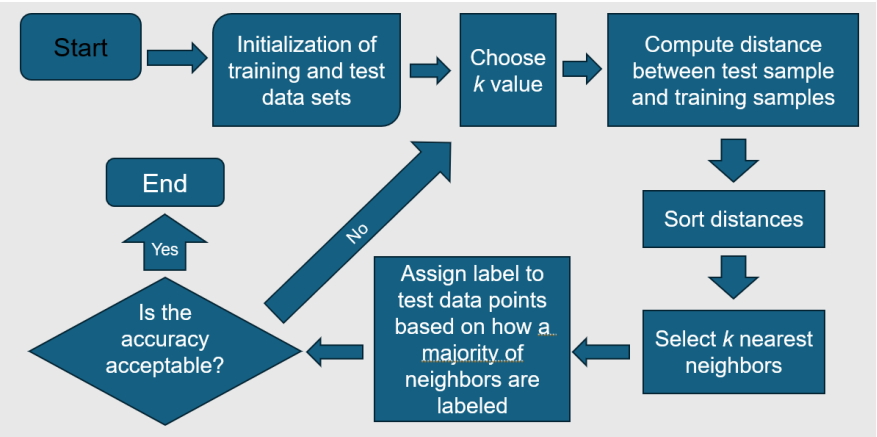

Figure 5. KNN Algorithm Flowchart

We also use k-fold cross validation. K-fold cross validation is a method of training and testing a machine learning algorithm. It separates the dataset into k folds and trains the model k times. Each time it uses one of the folds the dataset is split into as the test set and the rest of the data as training data. The k neighbors used in *KNN* and the k folds used in k-fold cross validation are values which have no relation to each other. We can see in the following picture an example of 10-fold cross validation. In the end it takes the accuracy score of the model on the test set from each iteration and averages the performance, and returns that value as the score of the model's performance.

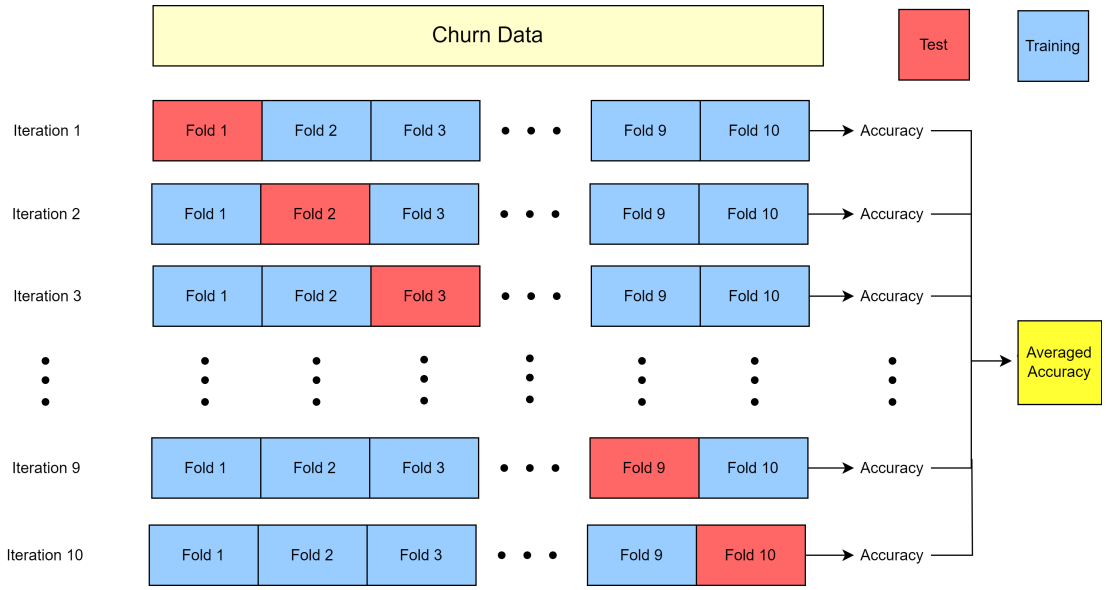

Figure 6. 10-Fold Cross Validation

We trained four algorithms all using *Scikit-Learn*. We trained two versions of *KNN*, one with a version of the dataset with our top three features scaled and one without the feature scaling. We then used the default version of Logistic Regression and Random Forests and trained them on the dataset without feature scaling. To test our model we ran 10-Fold Cross Validation ten times and recorded the resulting average accuracy score for all four models. We then tune the k parameter of *KNN* and the factor we scaled the top three features by 12 to increase the accuracy of the model.

2.7. Problem Solving Method and Analysis

To predict bank churn, we trained a *KNN* algorithm using top 3 feature scaling. Our work was inspired by Enriko et al, [6], who used top 3 and top 2 feature scaling to improve the performance of the *KNN* algorithm to predict Heart Disease. We chose to use top 3 feature scaling because our feature importance results from the RF model in Figure 7 show that the three features, Age, Number of Products, isActiveMember, are more important than the others. We use 10-fold cross validation and the Euclidean distance metric for the *KNN* algorithm. We also train a logistic regression model and a RF model to serve as comparisons to our *KNN* model. The logistic regression model serves as a baseline score for models computationally cheap and similar to *KNN*. The RF model serves as a baseline for comparison to more computationally expensive and complicated models. We use the RF model to determine the top 3 features which we will scale. To score our accuracy we use the default accuracy score in *Scikit-Learn*, which returns a percentage score of how many times the model predicted the label, churned or retained, accurately.

Random Forests are an ensemble model which combines the results of many simple learners to produce an accurate prediction. The RF model we trained consists of 100 Decision Trees which make a prediction, then the prediction is the class the majority voted for. In contrast *KNN* and Logistic Regression are simple models which only implement a single simple algorithm to make predictions.

For our training, we aim to optimize *KNN* by changing the value of k and the factor by which we scale the top 3 features. Thus, for each training iteration, we train a *KNN* model without feature scaling, a *KNN* model with feature scaling, a logistic regression model, and a RF classifier model and record the result of each. Each time we train the model, we train the model 10 times with 10-fold cross validation, and we take the average and record this as the result.

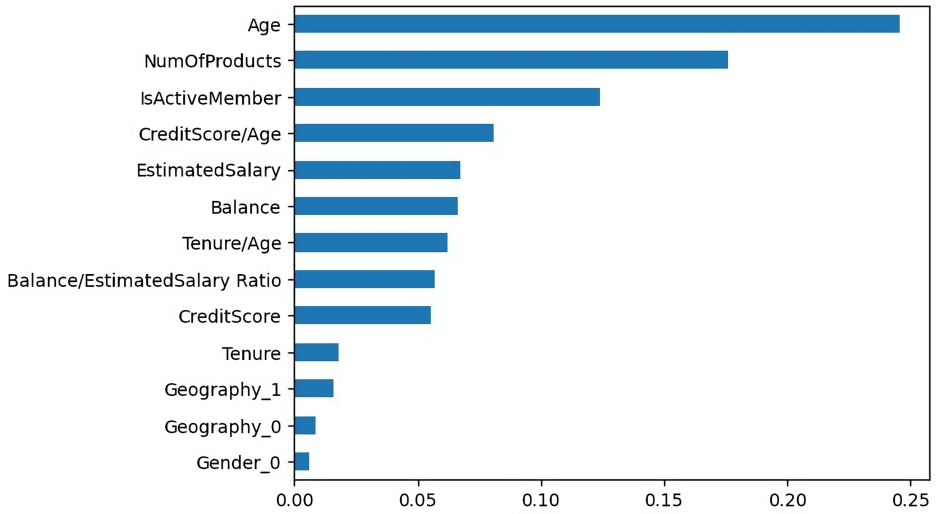

Figure 7. Feature Importance Ranking

To find optimal k and the optimal factor, we used trial and error by changing the values. We also used trends in the accuracy data to make more educated predictions for optimal values. Additionally, we experimented with different distance functions but found that Euclidean distance provided the best results in accuracy. Further, we found that RF and logistic regression models had very little to no variance in their accuracy scores when trained. At most the change was 0.06% differences between scores.

3. Results

Of the variables which were included in our dataset, we used Random Forests to rank feature importance and found that the three most important for predicting churn for a customer were their Age, isActiveMember, and Number of Products as shown in Figure 7. We gave these three attributes the highest weights within the distance formula in order to prioritize the most relevant features.

Table 1 shows how the score of the *KNN* algorithm can differ for small values of k. Here we scaled the top 3 factors by 2. We can see that we have a decent improvement in *KNN* when we scale the top 3 features and with very few changes *KNN* can outperform logistic regression. Our model, when scaled with a factor of 2, reaches its best accuracy when $k = 9$, but still falls short of the RF classifier by a large margin. We also see how the model ends up not improving only by increasing k.

Factor = 2

K	Logistic Regression	KNN No Feature Scaling	**KNN With Feature Scaling**	Random Forest
1	81.48%	78.87%	79.89%	86.08%
3	81.49%	81.42%	82.82%	86.09%
5	81.46%	81.95%	83.57%	86.08%
7	81.48%	82.13%	83.94%	86.08%
9	**81.46%**	**82.17%**	**83.97%**	**86.08%**
11	81.46%	82.19%	83.87%	86.08%
13	81.47%	82.10%	83.85%	86.08%
15	81.46%	81.91%	83.70%	86.08%

Table 1

Table 2 shows how the accuracy score of *KNN* changes with the factor we scale the top 3 features by a factor of 12. We use an optimal $k = 57$ we found through trial and error to illustrate this.

Factor = 12

K	Logistic Regression	KNN No Feature Scaling	**KNN With Feature Scaling**	Random Forest
51	81.44%	80.84%	85.68%	86.06%
53	81.45%	80.80%	85.70%	86.10%
55	81.46%	80.72%	85.70%	86.09%
57	**81.48%**	**80.68%**	**85.72%**	**86.10%**
59	81.47%	80.65%	85.65%	86.08%
61	81.47%	80.61%	85.69%	86.10%
63	81.48%	80.59%	85.65%	86.12%

Table 2

Table 3 shows our final result and optimized model. We found that by scaling the top 3 features by 12 and setting $k = 57$ we got an accuracy score of 85.72% which is very close to RF's accuracy score of 86.10%. We found that optimizing the factor by which the top 3 features are scaled results in the *KNN* model having a much higher accuracy score. We see from the graph that the optimal factor by which to scale the top 3 factors would be 12, after which the accuracy of KNN with feature scaling decreases.

K = 57

Factor	Logistic Regression	KNN No Feature Scaling	**KNN With Feature Scaling**	Random Forest
2	81.48%	80.68%	82.53%	86.10%
4	81.46%	80.67%	84.19%	86.08%
6	81.48%	80.71%	85.32%	86.10%
8	81.47%	80.65%	85.54%	86.11%
10	81.46%	80.69%	85.58%	86.10%
12	**81.48%**	**80.68%**	**85.72%**	**86.10%**
14	81.47%	80.68%	85.66%	86.09%
16	81.48%	80.66%	85.62%	86.08%
18	81.47%	80.64%	85.60%	86.10%
20	81.46%	80.67%	85.46%	86.06%

Table 3

4. Conclusion & Future Works

Our objective in this paper was to optimize a *KNN* model to predict bank churn. We used 10,000 examples of customer data from the ABC Multinational Bank to train our model. We optimized the model by applying weights to the most important variables, applying 10-fold cross validation, and iterating until the optimal k-value ($k = 57$) was found. The novel feature of our research in bank churn was optimizing *KNN* by tuning both k and the factor by which we scale the top three features. Our optimized *KNN* model, when compared to two other machine learning algorithms, gave us an accuracy score of 85.72%. This accuracy score was better than a computationally cheap logistic regression model and similar to a computationally expensive RF classifier.

Our work has some limitations, mainly due to the fact that we only analyzed data from one bank, with customers from just three countries. This means our model may not achieve the same accuracy if used with data from a different company or location; the weights might have to be adjusted to correct for differences between customer bases. Additionally, there is the possibility that customer behavior will change over time, necessitating occasional re-calibration of the algorithm so that its accuracy does not decrease as time passes. This is especially important since we used heuristic methods to optimize it. Nevertheless, we believe that the *KNN* algorithm can be adapted for different datasets using the process outlined in this paper. Having done so, and thus identified which customers may be likely to churn, banks could target those customers with incentives to stay. The precise incentives would depend on why the customers might leave; those dissatisfied with customer service could be the target of extra attention from specialists, while those seeking better financial terms could receive special offers. However, banks would have to do some extra work to learn what might work for specific customers; another limitation of our work is that, while the feature selection process might help identify broad trends causing customers to churn, the *KNN* algorithm cannot provide the reason any specific customer might consider leaving.

There are several avenues for future research work using this algorithm. It could be used to predict customer churn in different industries; in this case, the algorithm would have to be adapted to use different variables, as other businesses may not have access to customer information with the same level of detail as banks. Another avenue for future research could be to apply this approach to binary classification of different types of data. Such data includes medical data to determine whether someone might have a particular disease, financial data to predict whether someone will default on a loan, or voter data to predict election results, among others.

REFERENCES

1. M. Akturk, *Churn for Bank Customers*, Kaggle.com, 2020.
2. S. Al-Sultan and I.Al-Baltah, *An Improved Random Forest Algorithm (ERFA) Utilizing an Unbalanced and Balanced Dataset to Predict Customer Churn in the Banking Sector*, IEEE Access, 2024.
3. J.Brito, B. Bucco, R. Heldt, J. Becker, S. Silveira, B. Luce and J. Anzanello, *A framework to Improve Churn Prediction Performance in Retail Banking*, Brito et AL.Financial Innovation, 2024.
4. A. Coser, A. Aldea, M. Mihaela, and L. Besir, *Propensity to Churn in Banking: What Makes Customers Close the Relationship with a Bank?*, Economic Computation & Economic Cybernetics Studies & Research, vol. 2, pp. 77–94, 2020.
5. L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Vol. 31. Springer Science & Business Media, 2013.
6. I. A. Enriko, M. Suryanegara, and D. Gunawan, *Heart Disease Prediction System using k-Nearest Neighbor Algorithm with Simplified Patient's Health Parameters*, Journal of Telecommunication, Electronic and Computer Engineering, vol. 8, no. 12, pp. 59–65.
7. Y. Jouilil and M. Laousse, *Comparing the Accuracy of Classical, Machine Learning, and Deep Learning Methods in Time Series Forecasting: A Case Study of USA Inflation*, Statistics, Optimization & Information Computing, September 2023.
8. J. Lappeman, M. Franco, V. Warner, and L. Sierra-Rubia, *What social media sentiment tells us about why customers churn*, Journal of Consumer Marketing, vol. 35, no. 5, pp. 385–403, 2022.
9. C. Lim, E. Malik, K.Khaw, A.Alnoor, X.Chew, Z.Chong, and M.Akasheh, *Hybrig Ga- Deepautoencoder, KNN model for Employee Turnover Prediction*, Statistics, Optimization & Information Computing, January 2024.
10. R.A. de Lima Lemos, T.C. Silva, and B.M. Tabak, *Propension to customer churn in a financial institution: a machine learning approach*, Neural Computing and Applications, vol. 34, no. , pp. 11751–11768, 2022.
11. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, E. Duchesnay, M. Perrot, M. Brucher, D. Cournapeau, A. Passos, J. Vanderplas, V. Dubourg, R. Weiss, P. rettenhofer, M. Blondel, O. Grisel, and B. Thirion *Scikit-learn: Machine Learning in Python* Journal of Machine Learning Research, 12, 2825–2830
12. S. Ronaghan *The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark* Medium 2019, November 1
13. S.J. Russell and P. Norvig *Artificial Intelligence* Pearson Education 2009

14. Scikit-Learn Developers *"1.1. Linear Models — scikit-learn 0.22.2 documentation,"* scikit-learn.org, Jul. 29, 2019. https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression, (accessed Jun. 03, 2024).
15. Scikit-Learn Developers *"1.6. Nearest Neighbors — scikit-learn 0.21.3 documentation,"* scikit-learn.org, Jul. 29, 2019. https://scikit-learn.org/stable/modules/neighbors.html, (accessed Jun. 03, 2024).
16. Scikit-Learn Developers *"1.10. Decision Trees — scikit-learn 0.23.1 documentation,"* scikit-learn.org, Jul. 29, 2019. https://scikit-learn.org/stable/modules/tree.html#tree, (accessed Jun. 03, 2024).
17. K. Shahroodi, S.A. Darestani, S. Soltani and A.E. Saravani *Developing strategies to retain organizational insurers using a clustering technique: Evidence from the insurance industry*, Technological Forecasting and Social Change, 201, p.123217, 2024.
18. M. Simsek and I.C. Tas *A classification application for using learning methods in bank costumer's portfolio churn*, Journal of Forecasting, 43(2), pp.391-401, 2024.
19. I. Tandan and E. Goteman, *Bank Customer Churn Prediction: A comparison between classification and evaluation methods*, Uppsala University Department of Statistics, 2020.
20. H. Tran, N. Le, and V. Nguyen, *Customer Churn Prediction in the Banking Sector Using Machine Learning-Based Classification Models*, Interdisciplinary Journal of Information, Knowledge, and Management, vol. 18, pp. 87–105, 2023.
21. H. Wang and X. Miao, *Customer Churn Prediction on Credit Card Services using Random Forest Method*, Proceedings of the 2022 7th International Conference on Financial Innovation and Economic Development, 649–656, 2022.
22. S. Yulin and I. Palamar, *Probability Model Based on Cluster Analysis to Classify Sequence of Observations for Small Training Set*, Statistics, Optimization & Information Computing, March 2020.