Predicting if a Bank Client will Subscribe to a Term Deposit using Classification Algorithms

Rikesh Patel
Urvi Chawada

Abstract- **To improve bank efficiency, it is really important to predict the long term deposits a client will make. Banks are businesses and term deposits provide them the money they need to loan to their customers, which helps them increase their profits and reduce their costs. This paper uses decision trees and K nearest neighbors to predict whether bank customers subscribe to term deposits or not. It constructs a decision tree and a KNN model of factors of bank customers that mainly influence their term deposits. A series of experiments were conducted using these algorithms, both on unbalanced and balanced data. The results of decision trees show duration, outcome of previous campaign (poutcome) and month as the most influencing factors that push bank clients to subscribe to a term deposit. The results for KNN show age, day and balance as the most influencing factors.**

Keywords- Term deposit; Decision Tree; K Nearest Neighbors; Classification Model

1. INTRODUCTION

Marketing is a crucial part of any business and since banks are businesses, bank marketing is required to make profits. Banks potentially profit from term deposits and therefore, marketing term deposits is essential for banks. Term deposit is a term used to describe the deposits we make in banks for a specific amount of time, and in return we receive a standard interest rate. Banks benefit from these funds help banks to lend money to other individuals or companies, and earn a profit.

This paper takes customer data provided by Portuguese banking institution as a research to predict if a particular bank client will make a term deposit. This helps bank successfully market term deposits to the clients who are predicted to not make a deposit, thus increasing the efficiency of banks. The dependent variable is a binary variable and hence, this paper solves is a classification problem, which is to determine if a client will subscribe to a term deposit. The data represents binary, categorical and continuous variables, altogether. The Decision Tree method was adopted because it is a binary variable, and Decision Trees also help us find the relevance of variables. The KNN classifier was adopted to compare and contrast the results between the two algorithms, and to understand the data.

2. LITERATURE REVIEW

We reviewed two research papers related to our problem prior to conducting our own research.

The first paper we reviewed was published in 2019, titled *Term Deposit Subscription Prediction Using Spark MLlib and ML Packages.* This paper was also trying to predict whether or not clients will subscribe a term deposit using the same dataset as us. This made the paper incredibly relevant to our own research, as we could compare our results directly to their results.The methods used in this paper were Decision Tree, Random Forest, and Gradient Boosted Trees using Spark MLlib.

The second paper we reviewed was also published in 2019, titled *Statistical Decision Research of Long Term Deposit Subscription in Banks Based on Decision Tree.* This paper tries to discover the factors that cause company employees to subscribe a term deposit. Since our problem is pretty similar, this paper is again very relevant as we can compare our relevant attributes to the relevant factors found in this paper.

3. DATA TRANSFORMATION

3.1 Data Introduction

This dataset is made up of 45,211 bank clients with 17 attributes to describe each client. This data is taken from direct marketing campaigns (phone calls) of a Portuguese bank. The goal of mining this data is to predict if the client will subscribe to the term deposit. The 16 attribute variables recorded are assumed to be the influencing factors of whether customers subscribe to the term deposit or not.
The following 17 baseline parameters comprised the input set for the model:

- ***4 Binary***: Default, housing, loan, y (dependent)
- ***6 Categorical***: job, marital, education, contact, month, poutcome
- ***7 continuous***: age, balance, day, duration, campaign, pdays , previous

The output i.e "Does this client subscribe to a term deposit" was a two class binary variable 'y' (yes/no), where 'yes' is client subscribed to the term deposit and 'no' is client did not subscribe to the term deposit.

3.2 Balancing the data

The original data is highly unbalanced with Class 1 (yes) including 5289 instances and Class 2 (no) including 39922 instances. Therefore, the results of the decision trees are biased towards the 'no' class since it has the majority of instances. The data is very large and if all the data is used, it is found that too much data makes the computational efficiency relatively low. A simple solution is to balance the data and randomly extract part of the data by random sampling so that the data can provide accurate analytical results.

The data is balanced based on the minority class. Since the minority class (yes) only contains 5289 instances, 5289 instances were randomly picked from the majority class (no) which is approximately 11% of the total data size. Therefore, the new balanced data set includes 5289 instances of class 'yes' and 'no' each. This paper shows the analysis for both unbalanced and balanced data.

3.3 Data Transformation and Conversion

The data consisted of 7 numeric variables and 10 string variables. Correlation can only be performed on numeric variables and the class variable is a binary string variable. To understand the correlation between the dependent variable and other independent variables, variable y was transformed into numeric ([yes/no] changed to [1/0]). Other binary string variables are transformed into 0s and 1s. Categorical string variables are transformed using numbers 0,1,2,3,... The transformations were performed using SPSS.

4. METHODS

4.1 Decision Tree (DT)

A DT is a tree-like structure, where leaves represent outcome labels, i.e. Yes(1) or No(0), and branches represent conjunctions of input features that resulted in those outcomes.

A binary DT separates the data into two subsets (child nodes) by calculating the best feature split determined by a chosen split criterion. The resulting nodes become the parent nodes and are further split by the most relevant feature. The binary split is performed until all class are classified and it reaches the terminal node. The feature relevance in this data set is performed using the Gini Index since the Decision trees use the Gini Index as the default algorithm for selecting relevant features.

4.1.1 Decision Tree design in this study

The Decision Tree design in the present study was based on the standard CART (Classification and Regression Trees) algorithm implemented using SPSS. The DTs were performed on both the unbalanced and balanced datasets. Hold out partitioning technique was used to split the data into training and testing. 66% of data was used in training and 34% of data was used in testing. Hold out partitioning technique was used as the data size is relatively large. The relevance of the features was determined by the Gini index as it was the default method for DT. The experiment with the Decision Trees was repeated 11 times for the unbalanced data and 10 times for the balanced data. The experiments include additional constraints such as depth for the tree, minimum child node, and minimum parent node. The depth range used was between 10 to 20. The minimum parent node range used was between 10 to 200. The experiment was started with parent node equal to 200 (200 is the approximate square root of the data set size of 45000). The minimum child node range used was between 5 to 100. The experiment was started with a child node equal to 100 (100 is half of the parent node).

4.2 K Nearest Neighbors

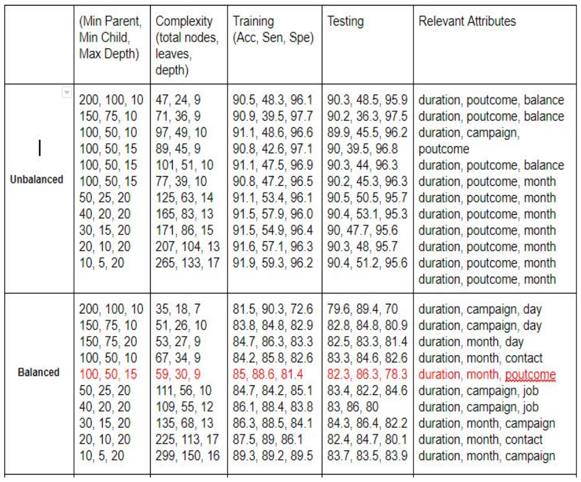
K Nearest Neighbors (KNN) uses a sample's geographic neighborhood to predict its classification. The classifier predicts a new sample using k-closest instances from the training set. The distances are measured by distance matrix such as Euclidean, Minskowski and Mahalanobis' distance. Euclidean distance is used as a distance matrix for this study.

The KNN classifier in the present study is based on the Euclidean distance matrix. It uses the Euclidean distance to find distances between its neighbors. The KNN was performed on both unbalanced and balanced dataset. Originally, the data was highly imbalanced and therefore, the sensitivity of the data was very low. The data was then balanced, and KNN was performed on the balanced data. The experiment with KNN was repeated for 7 times for unbalanced data and 6 times for balanced data. The k's used were 1, 3, 6, 9, 12, 15, 18 for unbalanced and 1, 3, 6, 9, 12, 15 for balanced. We stopped at 18 and 15 for unbalanced and balanced as the accuracy of the classifier wasn't changing much.

5. RESULTS

5.1 Decision Trees

We tested 10 different decision tree models. Performance for each model was measured by their accuracy, sensitivity, and specificity.We were also interested in seeing what the most relevant attributes were for classifying instances. What we discovered is



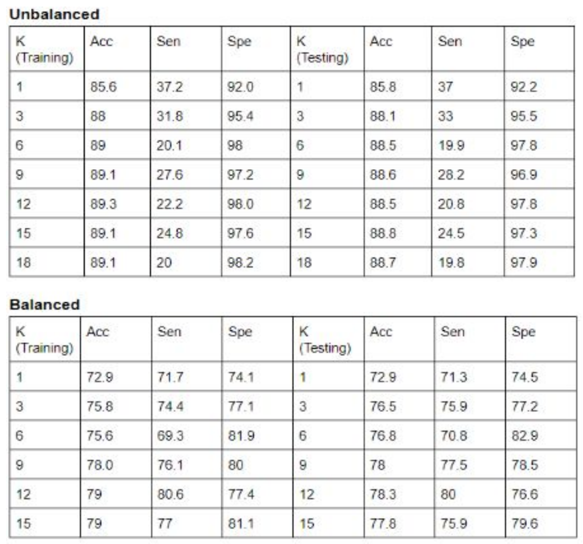
	(Min Parent, Min Child, Max Depth)	Complexity (total nodes, leaves, depth)	Training (Acc, Sen, Spe)	Testing	Relevant Attributes
Unbalanced	200, 100, 10	47, 24, 9	90.5, 48.3, 96.1	90.3, 48.5, 95.9	duration, poutcome, balance
	150, 75, 10	71, 36, 9	90.9, 39.5, 97.7	90.2, 36.3, 97.5	duration, poutcome, balance
	100, 50, 10	97, 49, 10	91.1, 48.6, 96.6	89.9, 45.5, 96.2	duration, campaign, poutcome
	100, 50, 15	89, 45, 9	90.8, 42.6, 97.1	90, 39.5, 96.8	duration, poutcome, balance
	100, 50, 15	101, 51, 10	91.1, 47.5, 96.9	90.3, 44, 96.3	duration, poutcome, month
	100, 50, 15	77, 39, 10	90.8, 47.2, 96.5	90.2, 45.3, 96.3	duration, poutcome, month
	50, 25, 20	125, 63, 14	91.1, 53.4, 96.1	90.5, 50.5, 95.7	duration, poutcome, month
	40, 20, 20	165, 83, 13	91.5, 57.9, 96.0	90.4, 53.1, 95.3	duration, poutcome, month
	30, 15, 20	171, 86, 15	91.5, 54.9, 96.4	90, 47.7, 95.6	duration, poutcome, month
	20, 10, 20	207, 104, 13	91.6, 57.1, 96.3	90.3, 48, 95.7	duration, poutcome, month
	10, 5, 20	265, 133, 17	91.9, 59.3, 96.2	90.4, 51.2, 95.6	duration, poutcome, month
					duration, poutcome, month
Balanced	200, 100, 10	35, 18, 7	81.5, 90.3, 72.6	79.6, 89.4, 70	duration, campaign, day
	150, 75, 10	51, 26, 10	83.8, 84.8, 82.9	82.8, 84.8, 80.9	duration, campaign, day
	150, 75, 20	53, 27, 9	84.7, 86.3, 83.3	82.5, 83.3, 81.4	duration, month, day
	100, 50, 10	67, 34, 9	84.2, 85.8, 82.6	83.3, 84.6, 82.6	duration, month, contact
	100, 50, 15	59, 30, 9	85, 88.6, 81.4	82.3, 86.3, 78.3	duration, month, poutcome
	50, 25, 20	111, 56, 10	84.7, 84.2, 85.1	83.4, 82.2, 84.6	duration, campaign, job
	40, 20, 20	109, 55, 12	86.1, 88.4, 83.8	83, 86, 80	duration, campaign, job
	30, 15, 20	135, 68, 13	86.3, 88.5, 84.1	84.3, 86.4, 82.2	duration, month, campaign
	20, 10, 20	225, 113, 17	87.5, 89, 86.1	82.4, 84.7, 80.1	duration, month, contact
	10, 5, 20	299, 150, 16	89.3, 89.2, 89.5	83.7, 83.5, 83.9	duration, month, campaign

Caption: this chart shows all the models tested. The most optimal model is highlighted in red.

that overall, accuracy was much higher for the models trained and tested on unbalanced data than with balanced data. The most optimal model trained and tested on unbalanced data (Min Parent Nodes = 40, Min Child Nodes = 20, Max Depth = 20) was 90.4% accurate in classifying the testing data. On the other hand, the most optimal model trained and tested on balanced data (Min Parent Nodes = 100, Min Child Nodes = 50, Max Depth = 15) performed slightly worse with 82.3% accuracy in classifying the testing dataset. This disparity can be explained by the bias in the unbalanced dataset towards the negative class. The unbalanced data had 39,922 instances in the negative class and only 5,289 instanced in the positive class. Therefore, because 88.3% of the data was in the negative class, the model was very accurate in classifying the negative class. On the other hand, this model 53.1% sensitive because so little of the data was in the positive class. After the data was balanced, with 5,289 instances taken from both classes, the sensitivity greatly improved. The most optimal model that was trained on balanced data was 86.3% sensitive in classifying the testing dataset. So, although the overall accuracy was not as great, this model is most optimal as its sensitivity is much higher. Furthermore, according to this model, the most relevant attributes for classifying instances are duration of call, month that call took place in, and the outcome of the previous campaign.

5.2 K Nearest Neighbors

We tested six different k-nearest neighbors models. Again, performance for each model was measured by their accuracy, sensitivity, and specificity. Relevant attributes were again gathered. Due to biased towards the negative class, we again noticed that the accuracy was much higher with the models tested on the unbalanced data while the sensitivity was much lower. The most optimal model trained and tested with the unbalanced data (k=1) was 85.8% accurate and 37% sensitive in classifying the testing data. On the other hand, the most optimal model trained and tested with the balanced data (k=12) was 78.3% accurate and 80% sensitive in classifying the testing data. Although the accuracy went down slightly, the sensitivity greatly increased again with the balanced data. This is because the positive class is equally represented as the negative class, and is no longer underrepresented. And so, our best performing model overall was "k=1" tested on balanced data. The relevant attributes gathered by this model were age of client, day that call was made, and account balance of client.


Unbalanced

K (Training)	Acc	Sen	Spe	K (Testing)	Acc	Sen	Spe
1	85.6	37.2	92.0	1	85.8	37	92.2
3	88	31.8	95.4	3	88.1	33	95.5
6	89	20.1	98	6	88.5	19.9	97.8
9	89.1	27.6	97.2	9	88.6	28.2	96.9
12	89.3	22.2	98.0	12	88.5	20.8	97.8
15	89.1	24.8	97.6	15	88.8	24.5	97.3
18	89.1	20	98.2	18	88.7	19.8	97.9

Balanced

K (Training)	Acc	Sen	Spe	K (Testing)	Acc	Sen	Spe
1	72.9	71.7	74.1	1	72.9	71.3	74.5
3	75.8	74.4	77.1	3	76.5	75.9	77.2
6	75.6	69.3	81.9	6	76.8	70.8	82.9
9	78.0	76.1	80	9	78	77.5	78.5
12	79	80.6	77.4	12	78.3	80	76.6
15	79	77	81.1	15	77.8	75.9	79.6

Caption: this chart shows all the models tested.

6. CONCLUSION

Overall, our models were successful in classifying instances according to the target variable 'y'. Our most optimal decision tree model (min parent nodes = 100, min child nodes = 50, max depth = 15) had an accuracy of 85% on the balanced training dataset and 82.3% on the balanced testing dataset. Our most optimal k-nearest neighbors model (k = 12) had an accuracy of 79% on the balanced training dataset and 78.3% on the balanced testing dataset. It is clear from these results that our decision tree model was more accurate in classifying instances than our k-nearest neighbor model. Furthermore, the decision tree model was more sensitive than the k-nearest neighbor model as well, with 88% sensitivity on both training and testing balanced datasets versus 80% sensitivity of the k-nearest neighbor model. It is clear to see that our decision tree model was more accurate and more sensitive in classifying instances according to the target variable when compared to our k-nearest neighbor model.

A significant point of contention between the two models is the relevant attributes that they outlined for classifying instances. According to our decision tree model, the most relevant attributes in determining whether or not a client would subscribe a term deposit were: (1)duration of the call, (2)outcome of the previous marketing campaign, and (3)the month the client was called. On the contrary, according to our k-nearest neighbor model, the most relevant attributes were: (1)client's age, (2)client's current account balance, and (3)the day the client was called. Both models outlined three different attributes from each other as being most relevant. The reason for this is that our k-nearest neighbors is a Euclidean Distance based classifier. Therefore, it can only take continuous attributes into consideration for being relevant in classifying instances, as it is impossible to find the distance according to categorical attributes. That is why every attribute that our k-nearest neighbors model outlined in continuous. The k-nearest neighbors model could not take into consideration certain attributes that our decision tree model could, such as poutcome (outcome of the previous marketing campaign). And so, because the decision tree model was able to take both categorical and continuous attributes into consideration, and the k-nearest neighbors model could only take continuous attributes, the results for relevant attributes are drastically different between the two models.

ACKNOWLEDGEMENTS

Further information on the data used in this paper is available at the University of California Irvine Machine Learning Repository - https://archive.ics.uci.edu/ml/datasets/Bank+Marketing.

REFERENCES

1. Duy Hung, Phan & Duc Hanh, Tran & Duc Tung, Ta. (2019). Term Deposit Subscription Prediction Using Spark MLlib and ML Packages. 88-93. 10.1145/3317614.3317618.
2. D. V. Patil and R. S. Bichkar, "A Hybrid Evolutionary Approach To Construct Optimal Decision Trees With Large Data Sets," *2006 IEEE International Conference on Industrial Technology*, Mumbai, 2006, pp. 429-433.

3. Guo, Junfeng & Hou, Handan. (2019). Statistical Decision Research of Long-Term Deposit Subscription in Banks Based on Decision Tree. 614-617. 10.1109/ICITBS.2019.00153.
4. Rosalind L. Bennett, Daniel A. Nuxoll, Robert A. Jarrow, Michael C. Fu, and Huiju Zhang. 2005. A loss default simulation model of the federal bank deposit insurance funds. In Proceedings of the 37th conference on Winter simulation (WSC '05). Winter Simulation Conference 1835-1843.
5. Stéphane Cédric Koumetio Tekouabou, Walid Cherif, and Hassan Silkan. 2019. A data modeling approach for classification problems: application to bank telemarketing prediction. In Proceedings of the 2nd International Conference on Networking, Information Systems & Security (NISS19). ACM, New York, NY, USA, Article 56, 7 pages.
6. T Shaikhina, et al., Decision Tree and random forest models for outcome prediction in antibody incompatible kidney transplantation. Biomed. Signal Process. Control (2017).
7. Wenliang Du and Zhijun Zhan. 2002. Building decision tree classifier on private data. In Proceedings of the IEEE international conference on Privacy, security and data mining - Volume 14 (CRPIT '14), Chris Clifton and Vladimir Estivill-Castro (Eds.), Vol. 14. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 1-8.
8. Xiao-Yong Lu, Xiao-Qiang Chu, Meng-Hui Chen, and Pei-Chann Chang. 2015. Data Analytics for Bank Term Deposit by Combining Artificial Immune Network and Collaborative Filtering. In Proceedings of the ASE BigData & SocialInformatics 2015 (ASE BD&SI '15). ACM, New York, NY, USA, Article 19, 6 pages.