Exploration of Five of the Most Used Supervised Learning Algorithms

Adedamola Adesoye
College of Engineering
Northeastern University
Toronto, ON
adesoye.a@northeastern.edu

Abstract

In this report, I explored the performance of supervised learning algorithms on two interesting datasets. The five learning algorithms include Decision Tree (DT), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Neural Networks (NN) and XGBoost. The two datasets were a bank marketing data and adult dataset. KNN and DT were used to handle the adult dataset while the other three were used for the bank marketing data. The learning curve, model complexity and training time of each algorithm on both datasets have been explored and analyzed.

1 Datasets

The datasets used for these projects are, a bank marketing data and an adult data, both of which I got from UCI's Machine Learning Repository. For the bank marketing data, it describes the results of an unnamed Portuguese bank's direct marketing campaign (via phone calls) to its customers to convince them to subscribe to a term deposit. Banks typically make money by borrowing the money in their customers' regular accounts, offering them interests on these funds, lending that same money to borrowers at a higher interest and pocketing the difference. The problem with this business model is that banks don't know when their customers will need their money so they don't know how long they can lend to borrowers. That's where term deposits come in. A term deposit is a type of deposit account held at a bank where money is locked up for an agreed time duration. With this, banks can confidently lend borrowers these funds with no fear of customers wanting that money before they can get it back from the borrowers. If a customer demands to withdraw their money before the agreed time, he'll face a monetary penalty which still profits the bank. Getting as many customers to subscribe to term deposits means a bank can confidently lend a lot more money to a lot more borrowers meaning more money for them. My goal is then to predict which customers are more likely to agree to term deposits which will then allow the bank's marketing team to improve their conversion rate.

The "adult" dataset contains information about individuals from a 1994 U.S.A. Census database. It contains demographic information like age, gender and race of individuals in the U.S.A. The target column contains two columns that tell if an individual earns $50,000 or less per year(<=$50K) or more than $50,000(>$50K). Such information can allow anyone with access to it to properly tailor their marketing or political campaigns to yield a better conversion rate.

Table 1: The basic feature of both datasets.

	Data Set Characteristics	Attribute Characteristics	Associated Tasks	Number of Instances	Number of Attributes
Bank	Multivariate	Real, Categorical	Classification	45,211	17

Marketing					
Adult	Multivariate	Real, Categorical	Classification	32,561	14

1.1 Data characteristic and Preprocessing

The bar charts illustrating the class distribution of both datasets are shown in Figure 1. Class imbalance is evident in both datasets and must be accounted for in calculating the accuracy scoring function.

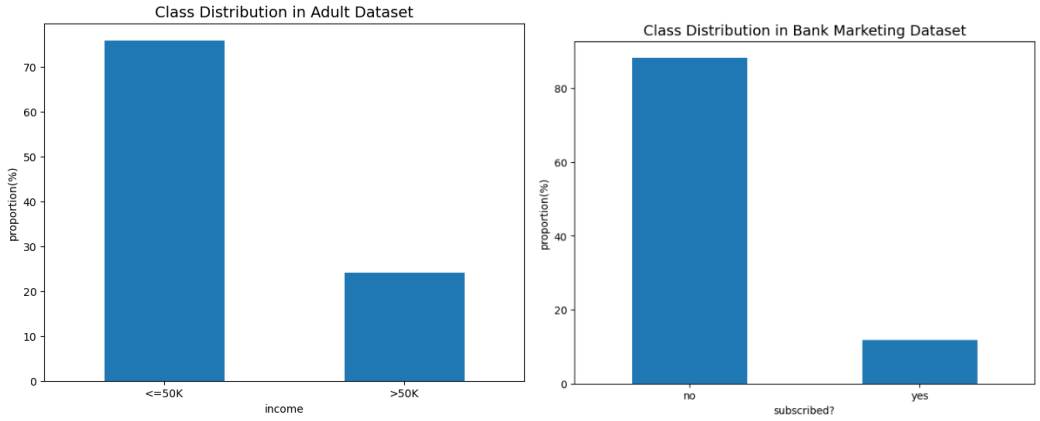


Figure 1: The class frequency of both datasets

The adult dataset had two features ('education' and 'education-num') giving the same information but in different formats. They both indicate an individual's highest level of education but the former does this with categorical variable (e.g: Bachelors, Doctorate) while the latter did this with numerical numbers ranging from (1-16) where "16" represented the highest level of education in the dataset (Doctorate) and "1" represented the lowest level (Preschool). After exploratory analysis, I dropped the 'education' variable to avoid issues with multicollinearity. Before building my model, I scaled the 'education-num' using Python's MinMaxScaler(). I chose this scaler in particular because the 'education-num' feature is an ordinal variable.

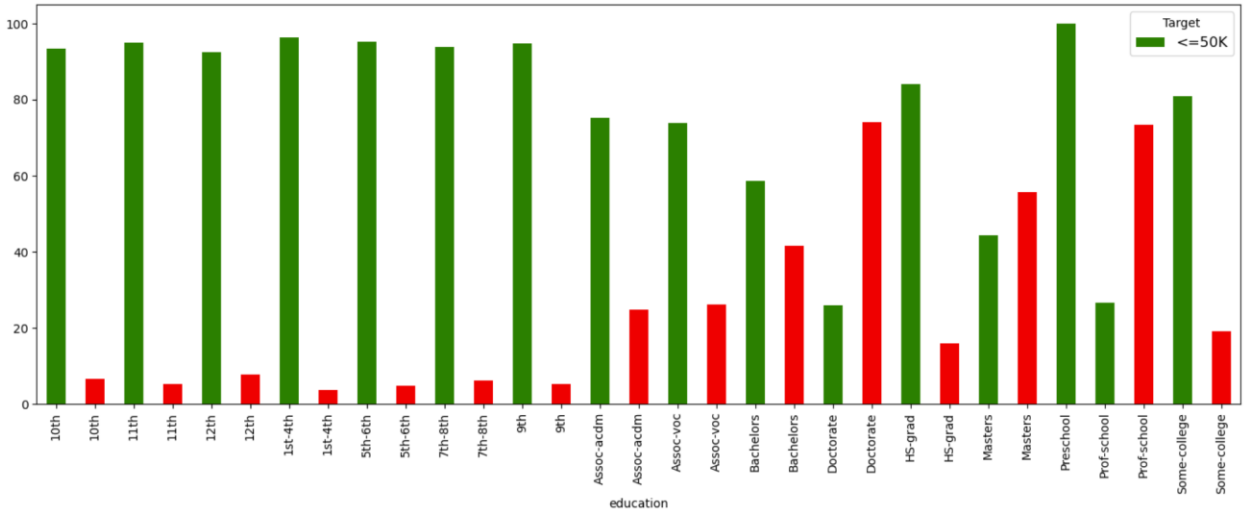
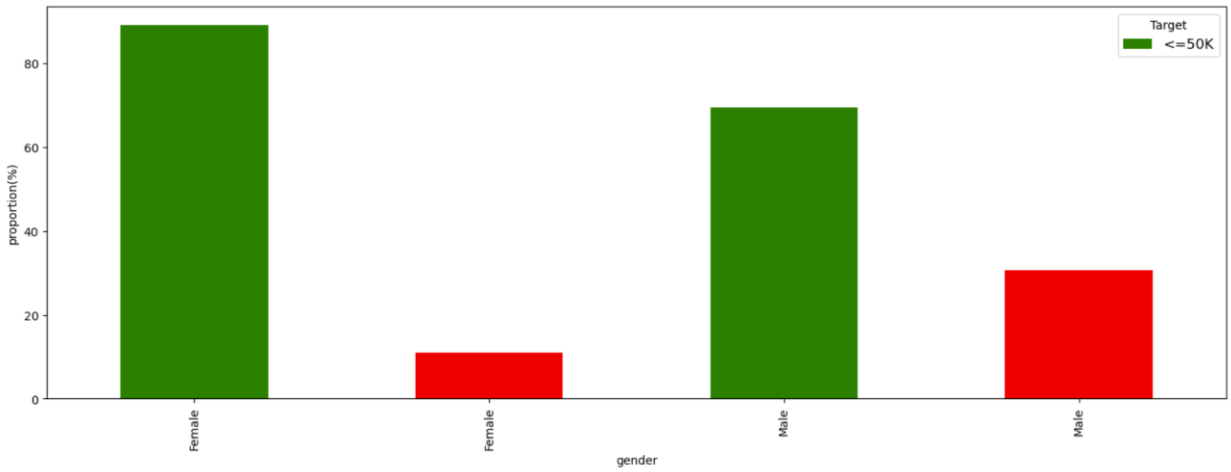
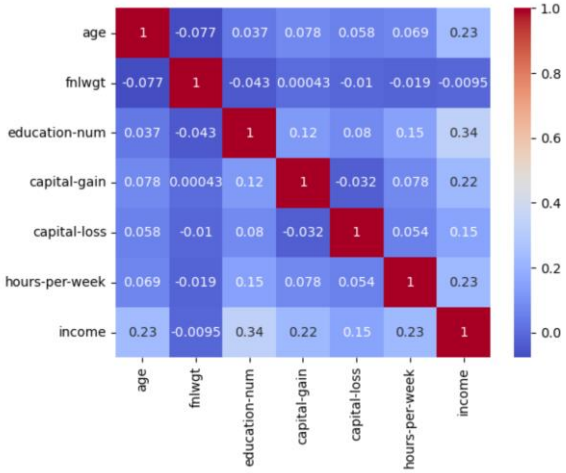


Figure 2 above shows that all 51 people who didn't make it past pre-school don't make more than $50K a year. Also, of the those who have a maximum education level of 12th grade, not more than 10% of them make more than $50k/year. Holding at least a master's degree gives one more than an average chance of earning more than $50k/year.



Males are, at least, twice more likely than females to earn more than $50k/year. This is an interesting piece of information for the gender inequality movement. One should be cautious to assume this proves (or proved in 1994) gender inequality. Other factors like level of education should also be explored before one can boldly say there was a bias against females in 1994.

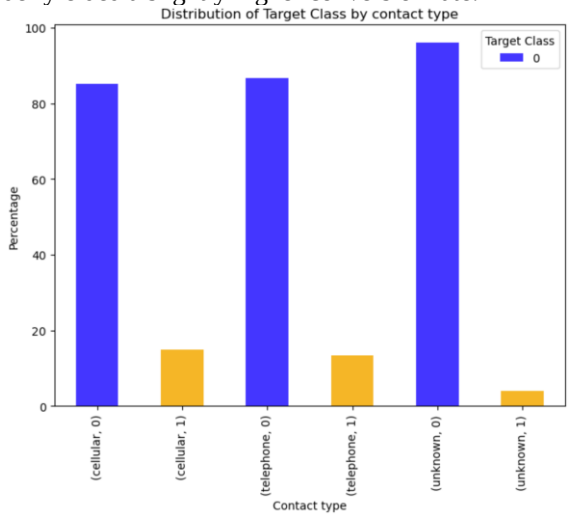
The heatmap below shows some, albeit weak, correlation between the level of education and hours-per-week which intuitively makes sense; the more learned you are, the more high-paying jobs you are employable for and the more hours you work, the more money you can make.



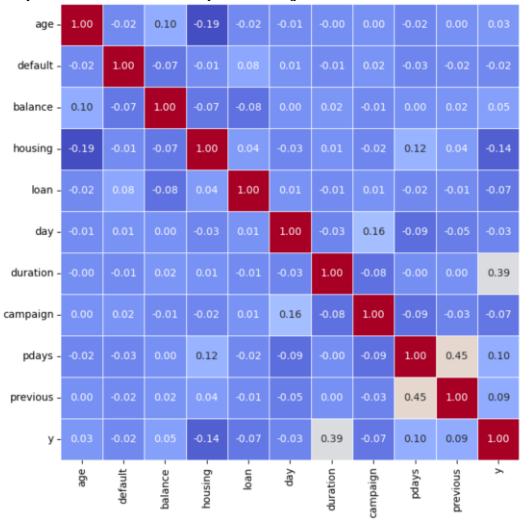
As shown earlier in figure 1, the “adult” dataset contains about 25% of individuals who make more than $50K/year and 75% who don’t. This 32,561-instanced dataset contained 2,399 instances with at least one missing feature value. These values were contained in the ‘occupation’, ‘workclass’ and ‘native-origin’ columns. They represent the occupation of an individual, the employment status of an individual (private, local-government, etc.) and their native origin respectively. All the instances with an unknown ‘workclass’ value also had an unknown ‘occupation’ value. Among other reasons, this may be due to participants being unwilling to disclose their source of income. The other instances (seven in number) with unknown *occupation* values are instance where *workclass* equals a value of *Never-worked.* Unsurprisingly, most of these individuals were teenagers. Surprisingly, all seven instance had values for *hours-per-week* greater than zero. Five even had values as high as 30 recorded. The *native-origin* feature accounted for the remaining 556 instances with at least one missing value. One could fill these values (or those with *race* values equal to *White*) with the most occurring native origin (United States) but I chose not to. With data size available, I chose to drop these and all other instances with at least one missing value. This left me with 30,162 instances and little change in target class proportions (0.8% increase in *>50k*’s proportion).

Along with the *MinMaxScaler* scaling applied on the *education-num* column, I one-hot encoded all categorical variables and scaled all numerical features to a mean of 0 and standard deviation of 1 (assuming normal distribution).

Taking our attention to the banking data, of the 45,211 instances, 36,959 instances had *unknown* values in the *poutcome* column. This column tells if a previous marketing campaign targeted at a customer (the instance being considered) was a *success* or a *failure*. The fourth unique label of this feature is *other*. I chose to drop this feature due to the level of ambiguity it had. For the *contact* feature which represents the type of device (cellular or telephone) a client used in receiving the call, I assumed that cellular meant a mobile phone and telephone meant a landline and with this assumption, I further assumed that calls received on a telephone were more likely to yield a conversion since a mobile phone carrier may be in transit and in a less suitable state to be receptive to a sales pitch. The figure below shows I was wrong. Not much correlation. If anything, calling a cellular holder yielded a slightly higher conversion rate.



The figure below shows the linear correlation relationship among the features. I converted the *default, housing and loan* variables (which had *yes* and *no* labels) to numerical (1 and 0 respectively) labels so they could make to this plot. The represent whether a client had credit in default, housing loan or personal loan respectively.



As shown in the plot below, only *duration* had a significant linear correlation with the outcome of a sales call. This point made itself known in the struggles faced in building the models for prediction. The *duration* feature refers to how long the last converation with a client took in

seconds. This feature was collected only after the result of the contact (subscribed or not) was known and should not be used to build the model for prediction since in reality this data will not be available for prediction.

Before building the model, I dropped the *day* column (representing what day of the month the contact was made). The reasoning behind this is that not only did it not have any significant linear correlation with the target variable, it didn't seem intuitive to remain; with the assumption that the marketers only worked on weekdays, some days, for example the 31st may fall on the weekend 5 times in a year and twice on a weekday making it less likely a conversion day. An alternative is using what day of the week the call was made; midweek may pull more subscribers that both ends of the week. Another way to use this is to make bins to represent the *start*, *middle* or *ends* of months. Maybe customers are more willing to subscribe when their paychecks just come in.
I converted the *education* (containing primary, secondary and tertiary) to (0,1, 2 respectively) and used a MInMaxScaler on the feature. All other numerical variables were scaled with *StandardScaler* and categorical with *OneHotEncoder*. I dropped the *duration* feature before building my models.

2 Model Building and Evaluation

I used a Decision Tree (DT) and K-Nearest Neighbors (KNN) algorithms in building models for predicting the income of U.S.A. residents in 1994. For the predicting whether customers will subscribe or not, I used a Support Vector Machine (SVM), XGBoost Classifier and Neural Networks. Firstly, let's review the work done on the adult data set.

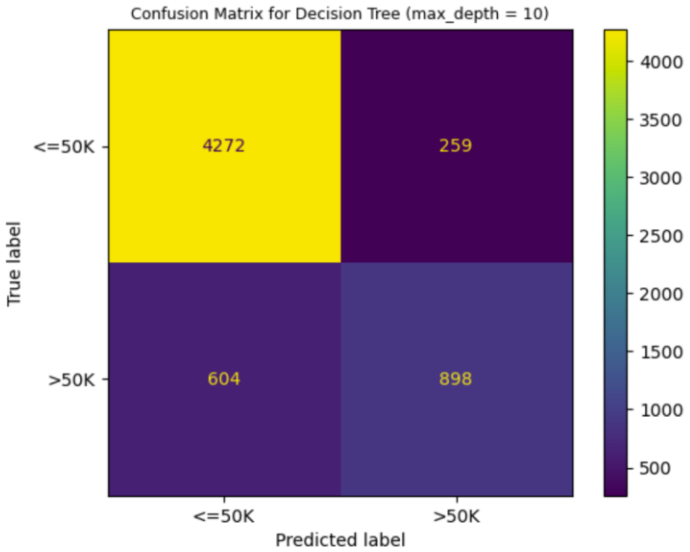
2.1 Decision Trees with some pruning

I treated this problem by regarding income greater than $50k as the positive class. I chose pre-pruning by controlling the maximum depth of the tree as illustrated in Table 2 below. I picked the best estimator based on the highest average f1-score of the stratified 5 folds.

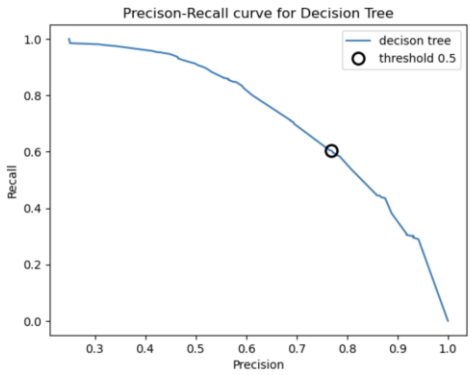
Table 2: The grid search range on splitting criterion, maximum depth, minimum samples per leaf and maximum features to use and the best estimator for both datasets.

Grid Search	Tried Estimators	Best Estimator
Max depth	None, 5, 10	10

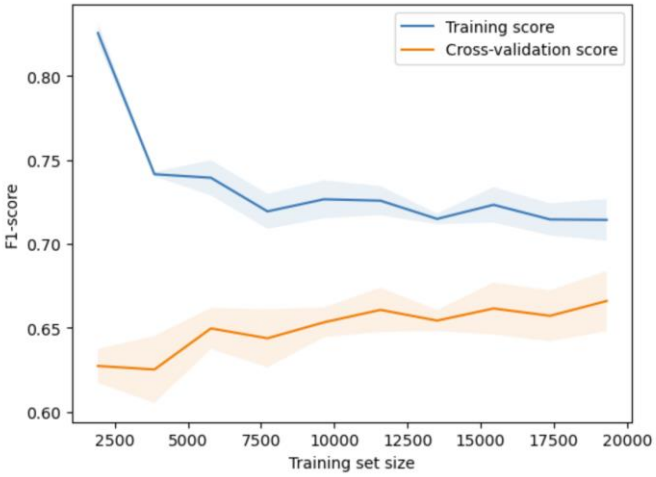
On the test set, the model returned an f1-score of 0.68. The figure below shows the confusion matrix for the decision tree.



A whooping 604 of those who earn more than $50k/year were misclassified as earning equal to or less than that. The figure below show the precision-recall curve for decision tree.



Depending on what the value of correctly classifying a positive class holds, the model's threshold can be reduced to achieve a recall as high as 0.90 with an average precision.



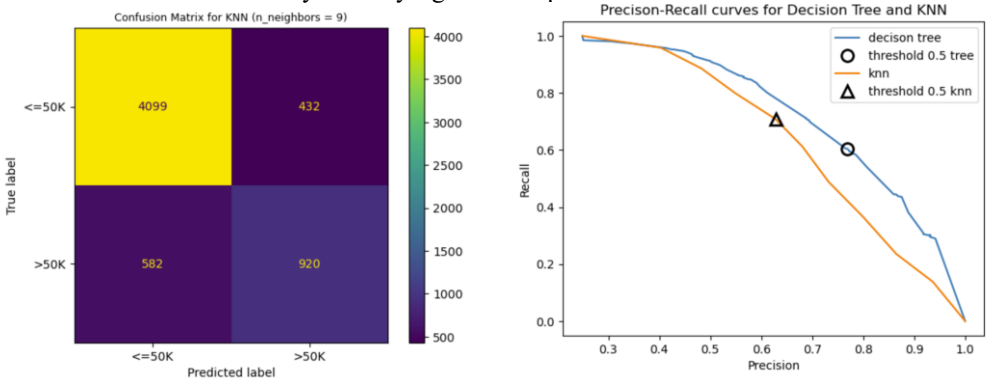
As the training set size increased, the f1-score of the test set increased gradually while that of the training set reduced gradually after a sharp drop in score. This may indicate that the model generalized better as data size increased.

2.2 KNN

The parameter tuned for the KNN was the number of neighbors which yielded the best estimator of 9.

Grid Search	Tried Estimators	Best Estimator
N_neighbors	3,5,7,9	9

On the test data, this produced an f1-score of 0.64. Although yielding a lower f1-score, it performed better in correctly classifying the true positive on the default 0.5 threshold.



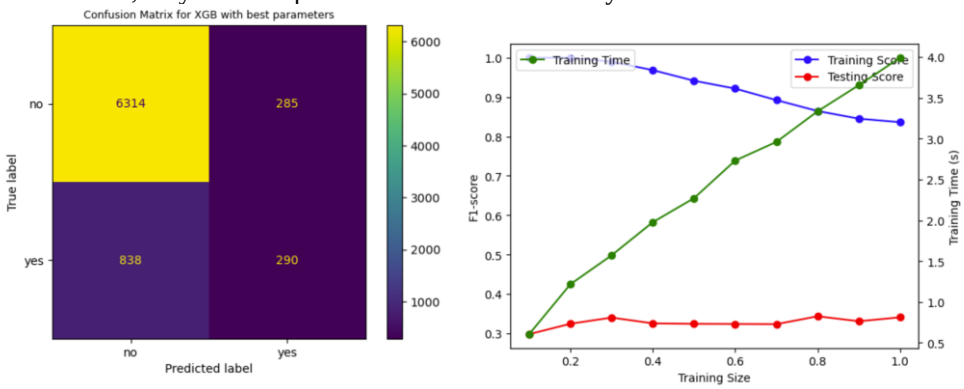
Considering all thresholds, decision tree performed better than kNN when looking at the plot precison-recall graph at the top right.

2.3 XGBoost

The models built using the bank marketing data yielded accuracy scores ranging from 0.80 and 0.90 but the f1-scores were woeful.

Grid Search	Tried Estimators	Best Estimator
Max_depth	3,5,7	7
learning_rate	0.01, 0.1, 0.5	0.5
n_estimators	50, 100, 200	200

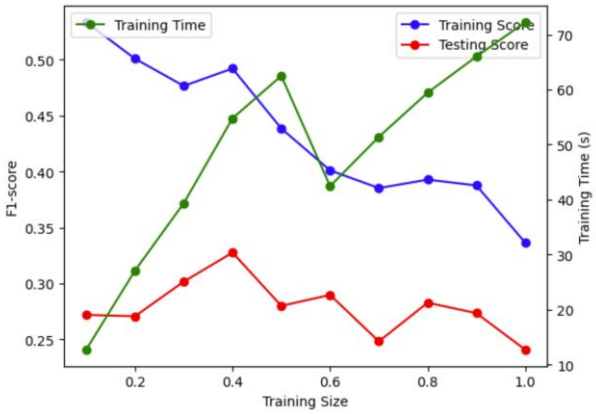
On the test set, the best performing parameters had an f1-score of 0.33. As seen in the confusion matrix below, only 290 true positive classes were correctly classified.



The learning rate to right shows that the model is extremely overfitted with the large gap between the training and test scores on all trining size bins.

2.4 Neural Network

No parameter was tuned for the NN. I used one hidden layer with 50 nodes and maximum iteration = 200. The figure below illustrates the learning rate for the NN.



2.5 SVM

For SVM the kernels kernels were tuned and linear svm yielded the better result with an f1-score = 0.17 on the validation set.

Grid Search	Tried Estimators	Best Estimator
kernel	Linear, rbf	Linear

With the relatively little time spent with the SVM, there is no other significant information I can provide.

3 Conclusions

Of the two datasets, the features in the adult dataset provided the most information for the algorithms to pick up on. As already shown in the heatmap for the bank dataset, there was no

feature showing any linear relationship to split the data. Further feature engineering may be produce better results. The *poutcome* column (the outcome of the previous campaign) maybe review and may give some insight to improve the performance of the model.

There is also room for improvement for the adult dataset. Varying the threshold will yield a better recall for bearable precision trade-off for the minority class. Depending on the goal of the one who possess the data, desiring more of the majority class should prove easier to optimize for.

Acknowledgments

The learning code was produced with multiple discussions with OpenAI's ChatGPT.

References

[1] Chen, J. (2022, March 20). Time Deposit: Definition, How It's Used, Rates, and How to Invest Investopedia. https://www.investopedia.com/terms/t/termdeposit.asp#:~:text=What%20Is%20a%20Term%20Deposit,levels%20of%20required%20minimum%20deposits.