DEVELOPMENT OF A PREDICTIVE MODEL USING MACHINE LEARNING TO DETERMINE CUSTOMER PAYMENT BEHAVIOR WITH DATA FROM AN ECUADORIAN BANK

PABLO SEBASTIÁN GARCÍA GUEVARA
Escuela Politécnica Nacional, Ecuador, marco.molinab@epn.edu.ec

MARCO EDUARDO MOLINA BUSTAMANTE
Escuela Politécnica Nacional, Ecuador, carlos.montenegro@epn.edu.ec

CARLOS ESTALESMIT MONTENEGRO ARMAS
Escuela Politécnica Nacional, Ecuador, pablo.garcia01@epn.edu.ec

Abstract

The objective of this paper is to develop a predictive model using machine learning to determine customers' payment behavior with data from an Ecuadorian Bank. This predictive model can help the bank anticipate the behavior of customers who are likely to default or stop paying a loan so that appropriate measures can be taken, such as selling the portfolio or putting pressure on the collection. The analysis of customer payment behavior was performed using time series, and four machine learning models were used Recurrent Neural Networks Model, Convolutional Temporal Networks Model, TRANSFORMER Model and N-BEATS Model using PYTHON's DARTS library. These models were compared with the classical Markov chains model, and three metrics were used to compare them: i) Error, ii) F-Score, iii) Confusion Matrix and their behavior was analyzed by increasing the number of predictions made. The final part analyzes the reasons for the models not being so robust and proposes how machine learning models could be better applied for this type of prediction.

Keywords: Machine Learning, Markov, Time Series, Banking.

DOI Number: 10.14704/nq.2022.20.13.NQ88275 Neuro Quantology 2022; 20(13):2203-2220

1. Introduction

Banking entities are dedicated to obtaining profits through the movement of money, one of the most profitable ones is lending money to individuals and small and large companies. Banks lend millions of dollars a year in the country since the more they lend, the more money they generate, thanks to the interest payment.

In order to maintain their stability, Ecuadorian banks must have good solvency, which is measured through the relationship between the funds they have to face unforeseen events with the risks they assume. In other words, they must balance the money they lend and the money they need for operating expenses and allow their clients to withdraw their money when necessary.

In order to maintain good solvency, these entities require that people pay their debts to the bank on time so that they have the liquidity to cover the bank's expenses and can obtain more profits by lending the money they receive back to the bank.

However, since not all debtors pay their obligations on time, an additional collection process is required to ensure that those who fall behind are brought up to date as soon as possible. The effectiveness of this process is significant since it represents an essential expense for the Banks and when it is not effective, this expense is in vain.

For this reason, a way of predicting the payment behavior of debtors is required to determine whether or not it is necessary to press for payment, either because they are not going to pay anyway or because, although they are late with their installments, they will eventually pay the same. In addition, if it is possible to determine

which debtors will definitively stop paying their obligations, actions can be taken such as collecting from guarantors, applying mortgage guarantees and selling portfolios at better prices since a bad payer is identified before other institutions classify it as such.

1.1 Research question

Is it possible to develop a predictive model using machine learning to determine customer payment behavior with data from an Ecuadorian Bank that works better than the Markov chains?

2. General Objective

Develop a predictive model with the use of machine learning for the determination of customer payment behavior with data from an Ecuadorian Bank.

2.1 Specific objectives

- Design a predictive model using machine learning to determine customer payment behavior.
- Train the predictive model for the determination of the payment behavior of customers with training data obtained from the customer portfolio of an Ecuadorian Bank.
- To compare the behavior of the predictive model that uses machine learning to determine customers' payment behavior with the model obtained by using Markov chains, using test data obtained from the customer portfolio of an Ecuadorian Bank.

Methodology

As described above, this work seeks to develop a predictive model for determining the payment behavior of customers with data from an Ecuadorian bank. Furthermore, it seeks to design a new solution more effectively than the Markov above chains. For this purpose, it was decided to use the Design Science methodology since, according to Hevner, in the design science paradigm, the knowledge and understanding of a problem domain and its solution are achieved in the construction and application of an artifact.

The development of this artifact is a predictive model that does not intend to solve a specific problem of an organization but to provide a better solution to the problem of predicting the portfolio behavior of banks. This artifact's objective is to find a better solution to this problem. That is, it falls within Hevner's definition of design science objectives: The resulting IT artifacts extend the boundaries of human problem-solving and organizational capabilities by providing intellectual and computational tools. Theories about their application and impact will follow their development and use.

Finally, this methodology was chosen since it proposes a mathematical basis for many types of quantitative evaluations of an IT artifact, including optimization tests, analytical simulation, and quantitative comparisons with alternative designs [21]. In this work, it was decided to perform a comparison between our predictive model with the classical Markov chain solution. Future work could be carried out using the same methodology to evaluate the model in a given organizational context, providing the opportunity to apply empirical and qualitative methods.

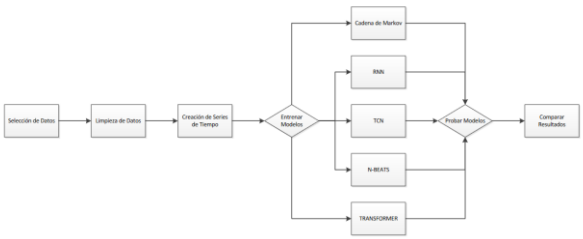

Figure 1. Methodology diagram

Source: Own elaboration

3.1 Data

The data for this project corresponds to the loan repayments of clients of an Ecuadorian Bank whose name will not be mentioned. The data contains information on the dates and amounts of installment payments and data on the types of loans and client information. Information is available on more than 570000 installments paid to correspond to more than 22000 loans delivered. The columns included in the dataset are listed below:

idOperation, idCustomer, numberFee, dateFeePayment, dateFeeMaturity, amountFeeCapitalFee, amountFee, numberFeesStartOperation, typeOperation, percentageInPaymentOperation, amountOperation, amountFeeOperation, rateOperation, termOperation, numberFeesOperation, isVehicle, classVehicle, vehicleMarkVehicle, vehicleValue, portfolioSold, portfolioBad, portfolioDischarged, codeClientSex, codeClientAge, codeClientStudyLevel, codeClientMaritalState, codeClientResidenceType, codeClientProvince, codeClientCity, codeClientIncome, clientExpenses, portfolioOverdueNoPaymentsToBorrow

3.1.1 Data Selection

Among all the data on loans, it is observed that the great majority correspond to loans for the purchase of vehicles; for this reason, only loans that indicate YES in the column esVehiculo are used, and after filtering, 414517 installments that correspond to 12573 loans were obtained.

3.1.2 Data cleansing

To clean the data, within the dataset, there are two types of information, and the first one refers to the characteristics of the loan and the client who requested it, and the second one refers to the installments that have been paid on that loan. To create the time series of loan payments, only the first time series of information is needed, but a DBSCAN clustering algorithm is used to ensure that the loans are similar to each other, which allows determining the clusters based on a minimum number of samples to be considered a cluster. Trying to minimize the number of loans considered outliers, it was determined that the minimum number of samples should be 10 and 2 clusters are obtained, shown below.

Figure 2. DBSCAN Cluster

As seen in Figure 2, there are not many loans in cluster 1, so it will only work with the loans corresponding to cluster 0.

Finally, since it is validated that several credits were paid in a very fast way from which it is not possible to learn the behavior of the payers, only the credits that lasted more than 18 months to be paid are used, and with this, 8843 credits are used to train the Machine Learning models.

3.1.2 Creation of time series

For the prediction of customer payment behavior, time series will be used for which a category of delinquency will be determined for each installment based on the data available. First, the analysis determines when the customer is in arrears when paying the installment by subtracting the columns dateDueFee minus datePayFee. If it is less than 0, it is taken as 0 and the category to which the customer belongs is determined based on Table 1.

Number of days in arrears	Blackberry Category
0	0
1-7	1
8-30	
31-60	
61-90	
91-120	5
121-180	
Greater than 180	

Table 1. Delinquency categories according to days in arrears

Based on the categories indicated, a category is assigned to each installment recorded for each of the credits available and a time series is created based on which prediction models can be generated using Markov chains and Machine Learning Algorithms.

3.2 Markov chains

3.2.1 Creation of transition matrix

As indicated above, it is possible to calculate the transition matrix based on the historical observation of an event; the following matrix is generated:

$$MT = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} & P_{04} & P_{05} & P_{06} & P_{07} \\ P_{10} & P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} \\ P_{20} & P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & P_{26} & P_{27} \\ P_{30} & P_{31} & P_{32} & P_{33} & P_{34} & P_{35} & P_{36} & P_{37} \\ P_{40} & P_{41} & P_{42} & P_{43} & P_{44} & P_{45} & P_{46} & P_{47} \\ P_{50} & P_{51} & P_{52} & P_{53} & P_{54} & P_{55} & P_{56} & P_{57} \\ P_{60} & P_{61} & P_{62} & P_{63} & P_{64} & P_{65} & P_{66} & P_{67} \\ P_{70} & P_{71} & P_{72} & P_{73} & P_{74} & P_{75} & P_{76} & P_{77} \end{bmatrix}$$

To calculate P_{ij} Which is the probability that the next state is j given that the current state is *i* and is calculated according to (1).

$$P_{ij} = \frac{N_{ij}}{N_i} \quad (1)$$

Where

N_{ij} is the number of times the next state is j given that the current state is *i*, N_i is the number of times event *i* occurs. To calculate the transition matrix, 70% of the vehicle credits is used and 30% of the credits will be used to test this model.

By calculating the transition matrix describing the vehicle loan payment event of this Ecuadorian Bank, it is possible to predict how the payment behavior of the clients of the loans separated to test the model. To test the model, it must calculate the last six payments of the loans separated to test the Markov chains model. To calculate the default status of a loan at time t+1 it is done according to (2).

$$e_{t+1} = e_t MT \quad (2)$$

Where

e_{t+1} is an 8-component vector indicating the state of the system at time t+1, e_t is an 8-component vector indicating the system's state at time t and MT is the 8x8 transition matrix. The state at time t is defined by (3).

$$e_t = [c_0 \quad c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7] \quad (3)$$

Where

c_n corresponds to category n, where it is equal to 1 when at time n the event is in category n and 0 for all others, i.e. if at time t the event is in category 4, the status would be given by (4).

$$e_t = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0] \quad (4)$$

That is, to calculate the predictions of the behavior of the time series at time t+k, it is done according to (5).

$$e_{t+k} = [c_0 \quad c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7]MT^k \quad (5)$$

The calculated system states allow creating of a time series and comparing the real one with the calculated one to determine how good the Markov chain model is.

4. Results

The data gathered are compared to measure the results of the predictions made by the five models. Each of the models is already trained on 30% of the available loans, called test data. To make the predictions using the models, the information of the last 6 installments of each loan is removed from the test data and the models are applied to predict those 6 installments.

The results obtained are compared with the real data, and 3 measures were used to determine the effectiveness of the prediction models. The first is the error, which measures the percentage of wrong predictions obtained by the model; the second is the F-score allowing to measure the model's accuracy. Finally, the Confusion Matrix is calculated to see better the error relation between the predicted value and the obtained value.

Each of the measures mentioned above was calculated for the models performing the 3 and 6 months prediction. Below is 1 table for each model with the summary of the Error, Hits and F-score applied for the test and training data predictions.

	Correct Training	Correct Test	Wrong Training	Wrong Test	F-Score Training	F-Score Test
3 month forecast	0.4664	0.4648	0.5336	0.5352	0.6806	0.6829
6 month forecast	0.3601	0.3530	0.6399	0.6470	0.5679	0.5717

Table 2. Summary of Error, Hits and F-score for the Markov Chain Model with Training and Test Data.

	Correct Training	Correct Test	Wrong Training	Wrong Test	F-Score Training	F-Score Test
3 month forecast	0.4465	0.4172	0.5535	0.5828	0.5784	0.5856
6 month forecast	0.3824	0.3635	0.6176	0.6365	0.5473	0.5474

Table 3. Summary of Error, Hits and F-score for the RNN Model with Training and Test Data.

	Correct Training	Correct Test	Wrong Training	Wrong Test	F-Score Training	F-Score Test
3 month forecast	0.3069	0.3136	0.6931	0.6864	0.5316	0.5441
6 month forecast	0.2841	0.2827	0.7159	0.7173	0.5048	0.5091

Table 4. Summary of Error, Hits and F-score for the TCN Model with Training and Test Data**.**

	Correct Training	Correct Test	Wrong Training	Wrong Test	F-Score Training	F-Score Test
3 month forecast	0.3310	0.3086	0.6690	0.6914	0.4988	0.4890
6 month forecast	0.2912	0.2755	0.7088	0.7245	0.4737	0.4707

Table 5 - Summary of Error, Hits and F-score for the NBEATS Model with Training and Test Data.

	Correct Training	Correct Test	Wrong Training	Wrong Test	F-Score Training	F-Score Test
3 month forecast	0.2962	0.2939	0.7038	0.7061	0.5116	0.5033
6 month forecast	0.2698	0.2646	0.7302	0.7354	0.4935	0.4903

Table 6. Summary of Error, Hits and F-score for the TRANSFORMER Model with Training and Test Data**.**

Based on the results of tables 2 to 6, it can be validated that there is no overfitting since there is no considerable difference between the errors obtained from the predictions in the training and test data for any of the Machine Learning or Markov Chains models. Next, it proceeded to compare the results of each of the measures described above.

Error

The following figure shows the error obtained for each of the models for a 3-month prediction, applied to the test data:

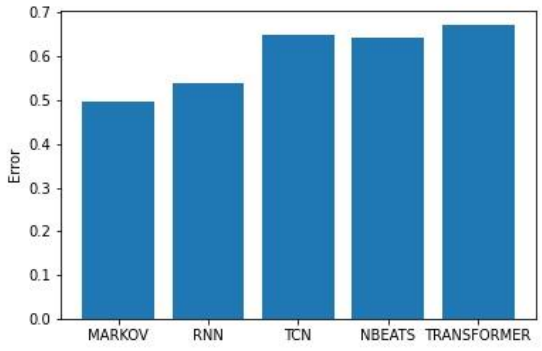

Figure 3. Error per model in 3-month prediction.

As shown in Figure 3, the error is similar between the RNN and Markov Chain models and is the lowest at around 50%, while the errors of the other 3 models are higher and are close to 65%. Figure 8 shows the error obtained for each of the models for a 6-month prediction, applied on the test data.

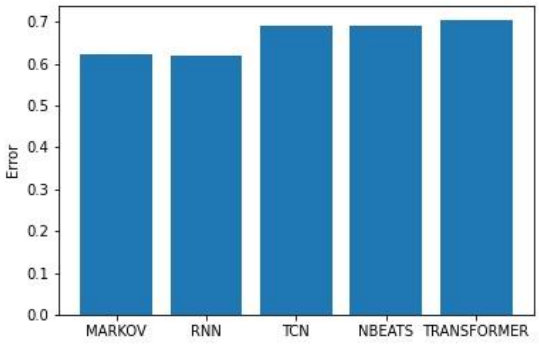

Figure 4. Error per model in 6-month prediction

As shown in Figure 4, the errors are similar between the RNN and Markov Chain models and are the lowest at around 60%, while the errors of the other 3 models are higher and close to 70%.

It is also interesting to compare how the error behaves as the number of months increases; for this, the last 18 installments of each of the loans were eliminated. Additionally, as seen in the parameters with which the Machine Learning models were created, at least 6 months of data are required to make the predictions; for this reason, only loans with at least 24 installments are used.

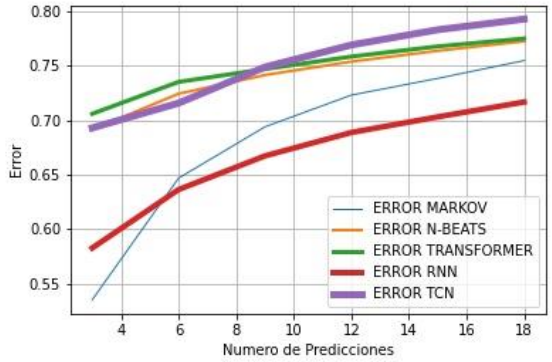

Figure 5. Error of each Model by Number of Predictions

As can be seen in Figure 5, the errors of the NBEATS, TRANSFORMER and TCN models are always greater than the error of the Markov chains, while the RNN model has a more significant error only for predictions of up to 6 months and after that has a lower error and an almost linear behavior with a very low slope.

Based on the results obtained by measuring the error, it is validated that the Markov Chains and the RNN models are the best. Markov presents a better performance with few predictions, while the RNN Model proves to be better for a larger number of predictions. The rest of the models show a much higher error than the other two mentioned above for a few predictions and this behavior is maintained as the number of predictions increases.

F-score

The F-score coefficient is a value between 0 and 1 that allows us to calculate the accuracy when comparing 2 results; in this case, the actual values of the installments of each of the loans are compared, with the predicted values using each of the machine learning models and the Markov chain model. The value of 0 is the worst possible accuracy, while 1 indicates that both values are equal, i.e., the accuracy is total.

The following is a comparison of the F-score results for each of the 5 models in the 3-month predictions applied to the test data:

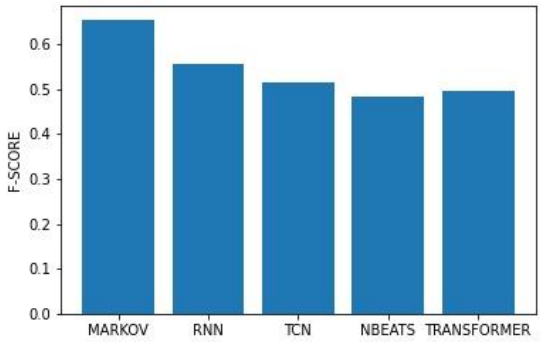

Figure 6. F-score per model in 3-month forecast

As seen in Figure 6, it is validated that the accuracy of the Markov chain model is higher than the machine learning models, around 0.65, while of these models, the RNN Model is the best with an F-score of 0.55; the rest is around 0.5.

Also, each model's F-score is compared when a 6-month prediction is made on the test data.

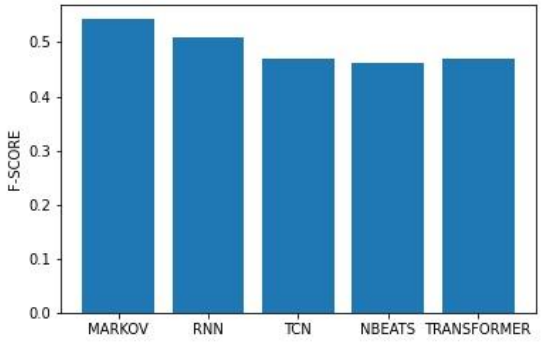

Figure 7. F-score per model in 6-month forecast

As can be seen in Figure 7, the trend is maintained and the Markov chain model has the highest F-score and the RNN model is in second place, although the difference is already reduced, while with a 3-month prediction, there was a difference of around 0.1 in 6 months, this difference is reduced to 0.05. The Markov model has an F-score of around 0.55 and the RNN model around 0.5, while the rest of the models have an F-score of around 0.46.

As was done with the error analysis, it is interesting to compare how the F-score coefficient behaves as the number of months

predicted increases. Similarly, a prediction of up to 18 months will be made and the F-score value for each model will be compared, as well as loans with at least 24 installments.

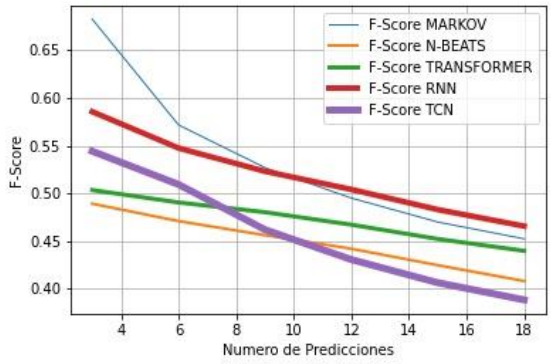

Figure 8- F-score of each Model by Number of Predictions

Figure 8 shows that the accuracy of the models decreases as the number of predictions increases. The Markov chain model has a higher F-score than the machine learning models up to the prediction at 9 months when the RNN model has a higher F-score than the rest of the models and this is maintained up to 18 months. The rest of the models always have a lower F-score than the Markov chain model.

When analyzing the model accuracy using the F-score coefficient, it was observed that the Markov chains model is the best as it presents higher fidelity than the machine learning models. And once again, it can be observed that the RNN model is the only one of the machine learning models that perform better when the number of predictions increases.

Confusion Matrix

The confusion matrix of a model allows for representing more important information about the behavior of the model since it allows to represent of how the model predicts each of the states and with which states are confused with each other at the time of making a prediction; the confusion matrix in our case is given by the following.

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} & P_{04} & P_{05} & P_{06} & P_{07} \\ P_{10} & P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} \\ P_{20} & P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & P_{26} & P_{27} \\ P_{30} & P_{31} & P_{32} & P_{33} & P_{34} & P_{35} & P_{36} & P_{37} \\ P_{40} & P_{41} & P_{42} & P_{43} & P_{44} & P_{45} & P_{46} & P_{47} \\ P_{50} & P_{51} & P_{52} & P_{53} & P_{54} & P_{55} & P_{56} & P_{57} \\ P_{60} & P_{61} & P_{62} & P_{63} & P_{64} & P_{65} & P_{66} & P_{67} \\ P_{70} & P_{71} & P_{72} & P_{73} & P_{74} & P_{75} & P_{76} & P_{77} \end{bmatrix}$$

Where

Pxy is given by the prediction being x given that the actual value is y, meaning that when x=y is the percentage of correct predictions and is on the diagonal of the matrix. The confusion matrices for each model are shown below and then compared.

88,31	6,88	4,81	0.00	0.00	0.00	0.00	0.00
33,59	51,31	15,10	0.00	0.00	0.00	0.00	0.00
10.74	5,86	83,24	0.15	0.00	0.00	0.00	0.00
4,59	1,67	49,95	43,59	0.21	0.00	0.00	0.00
2,96	0.27	37,20	46,63	12,94	0.00	0.00	0.00
1,01	1,01	27,70	11,49	58,45	0.00	0.00	0.34
0.72	0.36	11,15	7,91	29,14	0.00	45,68	5,04
0.00	0.00	13,33	3,33	5,83	0.00	0.00	77,50

Table 7. Confusion matrix for the Markov Chain Model for 3-month prediction.

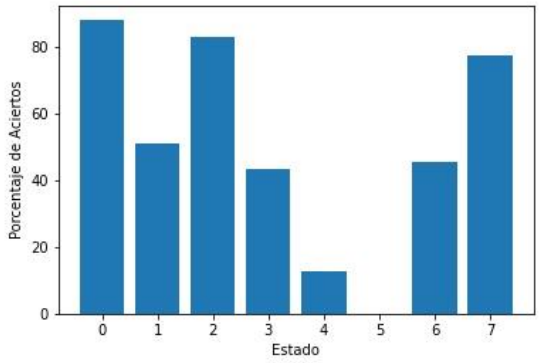

Figure 9. Percentage of the Markov chain model successes by the state for 3-month prediction.

The first thing that can be observed in Figure 9 is that the matrix is not balanced since the values of

the diagonal are very different, since for states 0. 2 and 7 they are close to 80% while states 1, 3 and 6 are close to 45% and for state 4 it is only 12% and finally state 5 has a hit of 0%, i.e., the Markov Chain model is unable to predict state 5.

It is also interesting to make an augmented hit chart in which each state will take into the hits when the predicted state is x but the real state is x-1 and x +1. It is not the same as predicting a state 1 when the real value is 7 than to predict a state 1 when the real value is 2.

86,59	3,40	9,30	0.71	0.00	0.00	0.00	0.00
51,46	24,99	22,65	0.90	0.00	0.00	0.00	0.00
20.70	2,91	73,59	2,74	0.00	0.00	0.00	0.05
9,02	0.81	62,27	27,56	0.10	0.00	0.00	0.25
6,18	0.13	52,56	33,90	6,31	0.00	0.00	0.92
3,41	0.73	40.88	10.46	42,09	0.00	0.00	2,43
2,53	0.25	27,27	10.10	20.45	0.00	32,07	7,32
0.91	0.00	38,60	8,81	2,13	0.00	0.00	49,54

Table 8. Confusion matrix for the Markov Chain Model for 6-month prediction**.**

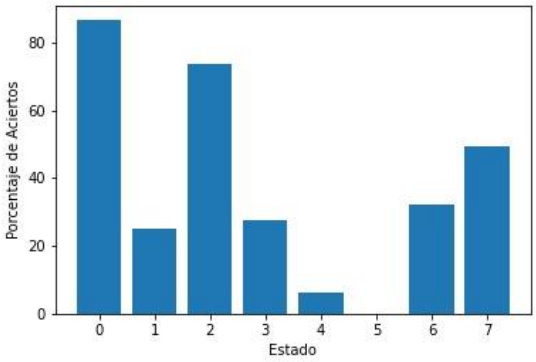

Figure 10. Percentage of the Markov chain model successes by the state for 6-month prediction**.**

It can also be seen in Figure 10 that the matrix is not balanced since the values of the diagonal are very different, since for states 0. 2 and 7 they are greater than 50% while states 1, 3 and 6 are close to 25% and for state 4 it is only 6% and once again state 5 has a hit of 0%, that is, it cannot predict state 5.

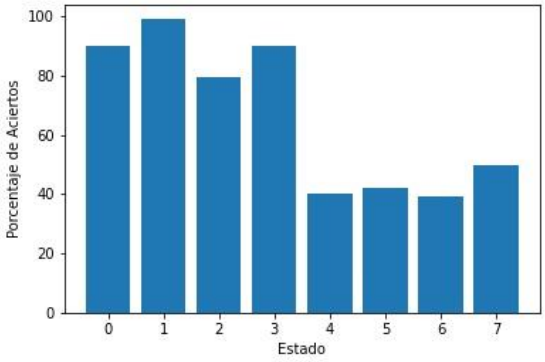

Figure 11. Increased hit percentage of the Markov chain model by the state for 6-month prediction.

When analyzing the increased percentage of hits again, this time from Figure 11, it is validated that the error percentages increase considerably, especially in state 1, which increases from 25% to almost 100%, in state 3, which increases from 27% to 90% and state 5, which increased from 0 to 50%. This results in a difference between the model error of 62.42% and an increased hit error of 21.35% for a 6-month prediction.

64,43	27,83	6,43	1,21	0.09	0.00	0.00	0.00
16,23	64,27	17,60	1,66	0.24	0.00	0.00	0.00
3,65	22,72	59,68	12,62	1,23	0.10	0.00	0.00
0.63	7,72	42,23	40.15	8,65	0.52	0.10	0.00
1,08	2,16	24,53	35,31	29,38	6,74	0.27	0.54
0.34	1,69	15,88	25,00	14,86	23,31	2,36	16,55
0.00	0.36	8,27	14,03	11,15	9,35	18,71	38,13
0.00	2,50	7,50	13,33	3,33	3,33	1,67	68,33

Table 9. Confusion matrix for the RNN Model for 3-month prediction**.**

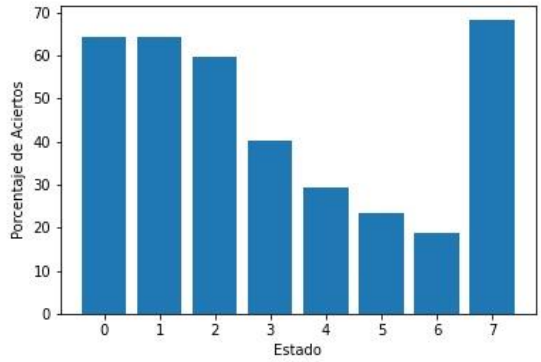

Figure 12. Percentage of hits of the RNN model by state for 3-month prediction.

Analyzing Figure 12 validates that the confusion matrix, despite having lower percentages of correctness than the Markov chain model, is more balanced since there are 4 states, 0.1,2,7 with similar percentages of correctness close to 60%, and there is no state that has correctness of 0 since the lowest is 19% for state 6.

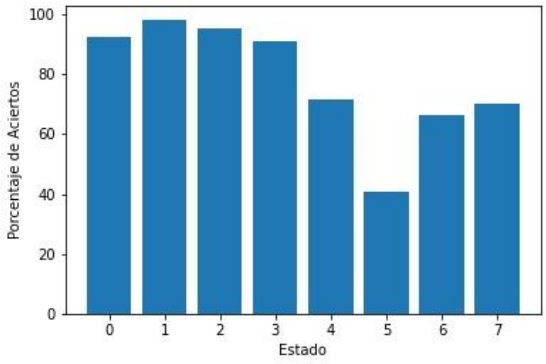

Figure 13. Increased hit percentage of the RNN model by the state for 3-month prediction.

When analyzing Figure 13, it is validated that the highest increase is in state 6, which goes from a 19% accuracy to 65%, and now the lowest is state 5, which is around 40%, in the same way, it is observed that it is more balanced than for the 3-month prediction than the Markov chain model, despite the fact that the error is 25.51% higher.

61,73	27,13	7,92	2,80	0.38	0.02	0.00	0.02
17,78	57,46	20.24	3,76	0.41	0.03	0.00	0.32
4,94	23,74	54,11	14,32	1,76	0.24	0.02	0.86
1,56	9,02	42,37	35,87	7,61	0.96	0.10	2,52
1,71	4,20	27,33	34,95	19,71	4,47	0.26	7,36
0.97	1,70	18,25	30.66	15,09	17,52	1,70	14,11
0.76	1,52	14,14	20.20	12,63	7,32	13,13	30.30
0.00	2,43	9,73	34,95	6,08	2,13	0.61	44,07

Table 10. Confusion matrix for the RNN Model for 6-month prediction**.**

Figure 14 shows that, as for the 3-month prediction, the confusion matrix is validated to be more balanced than in the Markov chain model for a 6-month prediction; although the error is similar for the Markov chain model and the RNN model, the increased percentage of correctness was also analyzed.

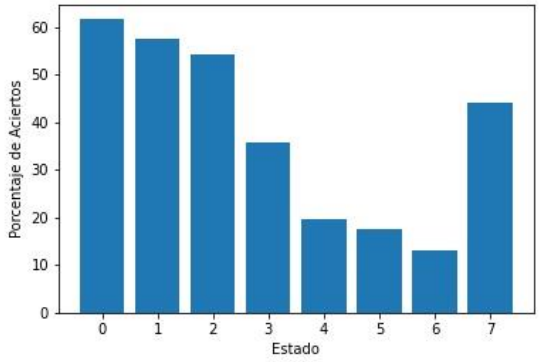

Figure 14. Percentage of hits of the RNN model by the state for 6-month prediction.

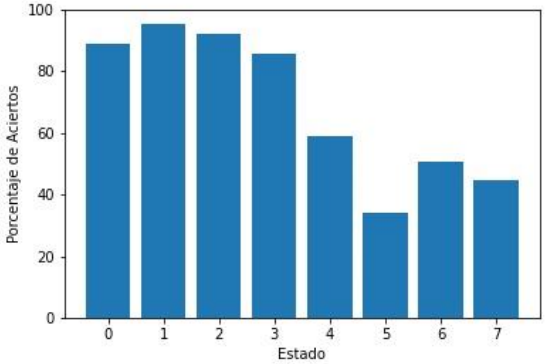

Figure 15. Increased percentage of hits of the RNN model by the state for 6-month prediction

Figure 15 shows that the percentage of hits is more balanced than for the Markov Chains model, although it is also noted that although the error in the case of Markov Chains and the RNN model is almost equal in the case of the increased error of hits, there is a difference since for the RNN model it is 29.07%.

58,14	31,07	8,45	1,98	0.31	0.04	0.00	0.00
14,45	61,06	21,28	2,68	0.42	0.12	0.00	0.00
3,35	20.80	59,04	13,60	2,41	0.79	0.00	0.00
1,15	8,03	40.56	40.15	5,74	4,38	0.00	0.00
1,35	2,96	28,03	40.16	11,32	14,82	1,35	0.00
0.68	0.68	19,59	27,03	15,54	7,77	28,72	0.00
0.36	1,44	13,31	21,22	11,15	10.43	42,09	0.00
0.00	1,67	10.00	12,50	8,33	8,33	59,17	0.00

Table 11. Confusion matrix for the TCN Model for 3-month prediction.

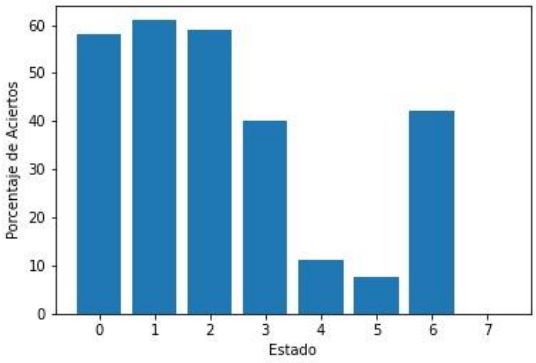

Figure 16. Percentage of hits of the TCN model by the state for 3-month prediction.

It can be seen in Figure 16 that the confusion matrix is unbalanced since, for example, for state 7 the percentage of correctness is 0%; that is to say that this model cannot predict state 7. For states 4 and 5 the percentage of correctness is less than 12%; it can also be seen that for states 0.1,2, the percentage of correctness is close to 60% and for states 3,6, the percentage of correctness is close to 40%.

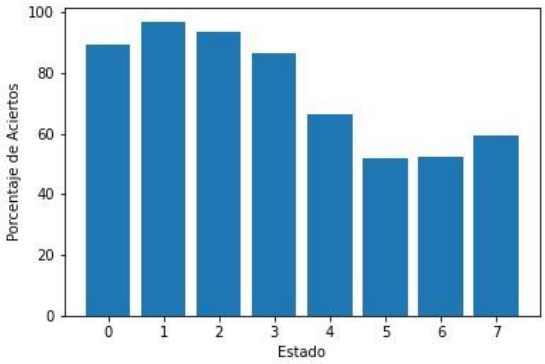

Figure 17. Increased hit percentage of the TCN model by the state for 3-month prediction.

Figure 17 shows a more balanced percentage of hits, and the most drastic increases occur in state 4,5,7, which go from less than 12% to more than 50% of hits when taking the increased hit. The error of the increased hit percentage is still 26.44%, which is higher than that of the Markov chain model.

48,83	37,60	9,76	2,40	0.62	0.78	0.00	0.00
11,18	60.18	23,60	3,56	0.72	0.72	0.03	0.00
3,13	22,52	56,78	12,78	2,47	2,30	0.02	0.00
1,11	9,17	45,14	32,14	5,89	6,55	0.00	0.00
1,18	4,73	31,41	34,95	11,43	12,09	4,20	0.00
0.97	0.73	22,14	33,09	14,60	7,30	21,17	0.00
0.76	2,02	18,18	25,76	12,12	11,11	30.05	0.00
0.00	1,52	13,37	24,62	16,72	18,54	25,23	0.00

Table 12. Confusion matrix for the TCN Model for 6-month prediction**.**

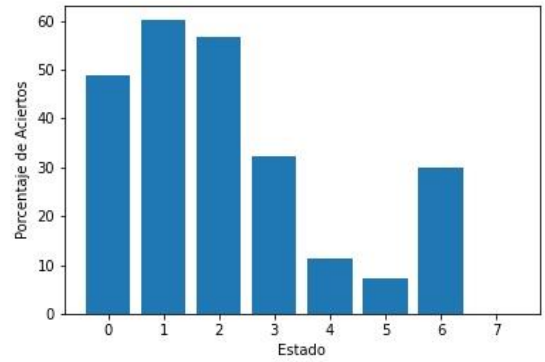

Figure 18. Percentage of TCN model hits by the state for 6-month prediction.

It is validated that for 6 months, as shown in Figure 18, there is still an unbalanced confusion matrix, there is still a very low percentage of error for states 4 and 5 and for state 7 the percentage of correctness is still 0%, that is to say, it cannot predict state 7.

Figure 19 validates considerable increases in the percentage of success in all states, but especially in states 4 and 5. In addition, it is validated that there is a considerable increase in the error compared to that of the Markov Chains model, which is 33.81%.

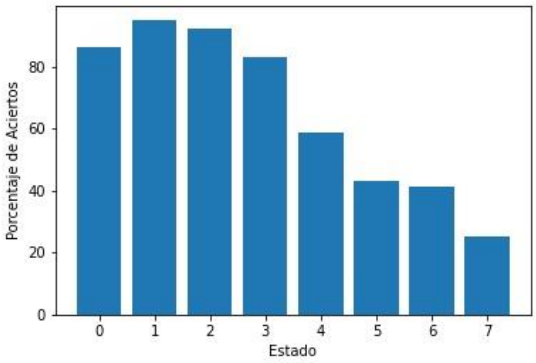

Figure 19. Increased hit percentage of the TCN model by the state for 6-month prediction.

82,64	14,07	3,19	0.04	0.04	0.00	0.00	0.00
47,38	46,25	6,18	0.12	0.06	0.00	0.00	0.00
9,96	44,60	43,91	1,13	0.39	0.00	0.00	0.00
2,19	21,38	63,40	9,59	2,92	0.52	0.00	0.00
2,16	8,89	49,33	14,82	20.75	3,50	0.27	0.27
0.68	7,77	32,43	10.81	12,50	20.95	4,39	10.47
0.00	3,96	17,99	5,04	7,91	24,82	19,42	20.86
0.83	4,17	15,00	2,50	5,83	8,33	20.83	42,50

Table 13. Confusion matrix for the NBEATS Model for 3-month prediction**.**

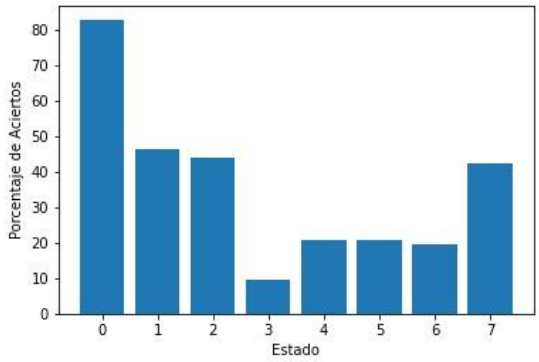

Figure 20. Percentage of hits of the NBEATS model by state for 3-month prediction.

It can be seen in Figure 20 that the confusion matrix is unbalanced since for state 1 it has a high percentage of correctness and for the rest of the states it is less than 50% and even for states 3,4,5 and 6 it has a percentage of correctness of less than 21%, additionally, it can be seen that no state has 0% correctness.

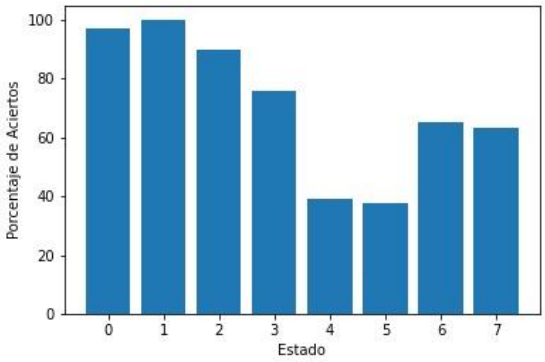

Figure 21. Increased hit percentage of the NBEATS model by state for 3-month prediction.

Figure 21 shows that the percentage of hits is more balanced, and the most drastic increases occur in state 3, which goes from less than 10% to more than 75% of hits when taking the increased hit. The error of the increased hit percentage is still 31.1%, which is higher than that of the Markov chain model.

77,79	16,12	5,63	0.22	0.13	0.04	0.04	0.02
41,67	47,64	9,88	0.38	0.09	0.03	0.12	0.20
10.89	43,44	42,22	1,71	0.59	0.29	0.20	0.66
3,58	22,92	59,55	7,15	3,58	0.81	0.55	1,86
2,76	12,88	51,51	9,86	13,14	3,42	1,18	5,26
1,22	9,49	39,42	10.95	10.46	16,30	3,65	8,52
1,26	6,82	27,78	5,81	8,08	18,18	15,40	16,67
1,22	4,56	39,82	4,26	6,99	4,56	9,73	28,88

Table 14. Confusion matrix for the NBEATS Model for 6-month prediction.

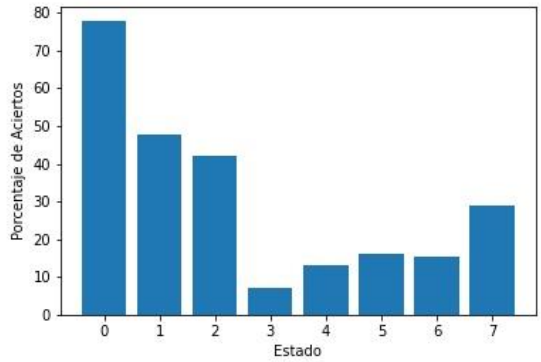

Figure 22. Percentage of NBEATS model hits by state for 6-month prediction.

Based on Figure 22, it is validated that for 6 months the confusion matrix is still unbalanced, there is still a very low percentage of error for states 3, 4, 5 and 6 and the percentage of success for state 1 is still very high, close to 80%.

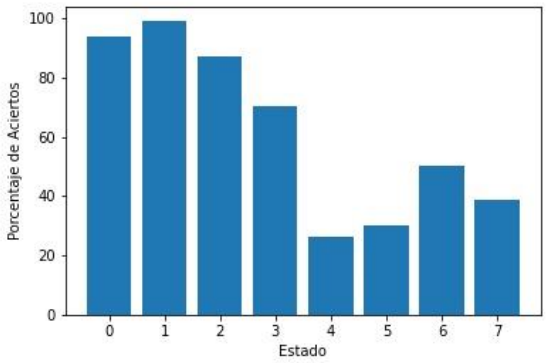

Figure 23. Increased hit percentage of the NBEATS model by state for 6-month prediction.

Figure 23 validates considerable increases in the percentage of correctness of all states, but especially in state 3. Additionally, it is validated that there is a considerable increase in the error compared to that of the Markov Chains model, which is 34.43%.

80.85	14,84	4,23	0.09	0.00	0.00	0.00	0.00
41,50	50.12	8,03	0.30	0.06	0.00	0.00	0.00
9,12	40.81	48,10	1,72	0.25	0.00	0.00	0.00
2,09	16,16	68,20	10.43	2,09	1,04	0.00	0.00
1,62	5,66	51,48	22,37	11,59	6,47	0.81	0.00
0.68	5,74	33,45	11,49	9,80	21,96	16,89	0.00
0.00	2,52	18,71	5,76	8,63	25,54	38,85	0.00
0.83	4,17	15,00	1,67	2,50	7,50	68,33	0.00

Table 15. Confusion matrix for the TRANSFORMER Model for 3-month prediction**.**

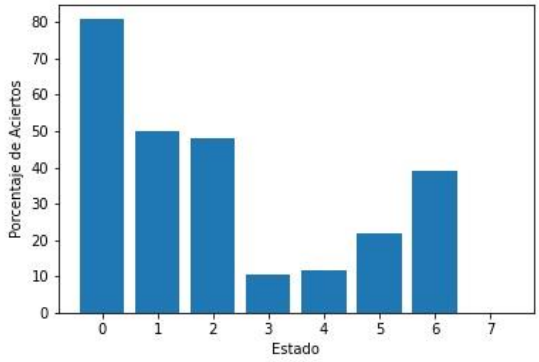

Figure 24. Percentage of hits of the TRANSFORMER model by state for 3-month prediction.

It can be seen in Figure 24 that the confusion matrix is unbalanced since for state 0 it has a high percentage of correctness and for state 7 it has a percentage of correctness of 0%. States 1 and 2 have a hit rate of about 50% and states 3, 4 and 5 have a hit rate of less than 22%.

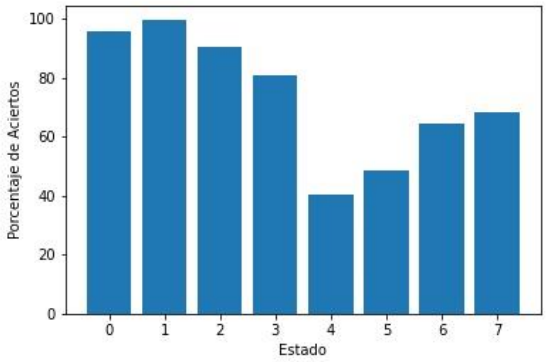

Figure 25. Increased hit percentage of the TRANSFORMER model by state for 3-month prediction.

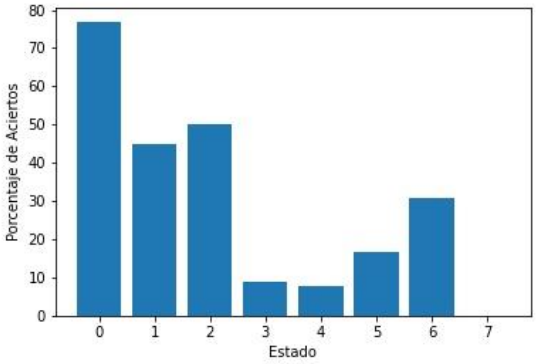

Figure 26. Percentage of hits of the TRANSFORMER model by state for 6-month prediction.

In Figure 25, the percentage of hits is more balanced, and the most drastic increases are in state 1, which goes from less than 50% to close to 100% of hits when taking the increased hit, state 7 goes from 0% to more than 60%. Nevertheless, the increased hit percentage error is 37.95%, which is still higher than that of the Markov chain model.

76,90	15,12	7,23	0.47	0.02	0.22	0.04	0.00
38,98	45,09	14,74	0.49	0.17	0.20	0.32	0.00
10.30	35,61	50.17	1,93	0.44	0.66	0.88	0.00
3,58	16,22	64,84	8,92	2,12	1,66	2,67	0.00
2,63	8,80	52,69	15,77	7,62	4,86	7,62	0.00
1,22	6,33	41,12	11,44	9,00	16,55	14,36	0.00
1,26	3,79	28,54	7,83	7,58	20.20	30.81	0.00
1,22	3,04	38,91	4,26	2,43	6,08	44,07	0.00

Table 16. Confusion matrix for the TRANSFORMER Model for 6-month prediction.

Based on Figure 26, it is validated that for 6 months there is an unbalanced confusion matrix, even though there is a very low percentage of error for states 3, 4 and 5. The percentage of correctness for state 0 is still 0% and the correctness of state 0 is still high, close to 80%.

Figure 27 validates considerable increases in the percentage of correct predictions in all the states, but especially in states 1, 3 and 7. Additionally, it is validated that there is a considerable increase in the error compared to that of the Markov Chains model, which is 35.29%; it is the only model in which it is observed that the error of correct predictions is lower for a 6-month prediction than for a 3-month prediction.

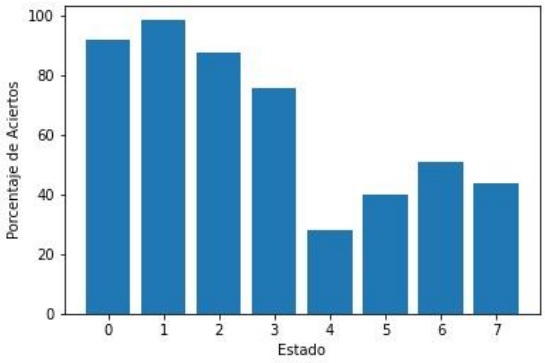

Figure 27. Increased hit percentage of the TRANSFORMER model by state for 6-month prediction.

Figure 27 validates considerable increases in the percentage of correct predictions in all the states, but especially in states 1, 3 and 7. Additionally, it is validated that there is a considerable increase in the error compared to that of the Markov Chains model, which is 35.29%; it is the only model in which it is observed that the error of correct predictions is lower for a 6-month prediction than for a 3-month prediction.

Based on what was observed in the analysis of the confusion matrix of the 5 models applied for predictions of 3 and 6 months, it is observed that the RNN model is the best since it has a more balanced confusion matrix, i.e., it is capable of predicting each of the states similarly, despite having a slightly higher prediction error, it has a better behavior with the data.

Finally, the study analyzed how well each model predicts each possible state and how this accuracy percentage changes as the number of predictions increases.

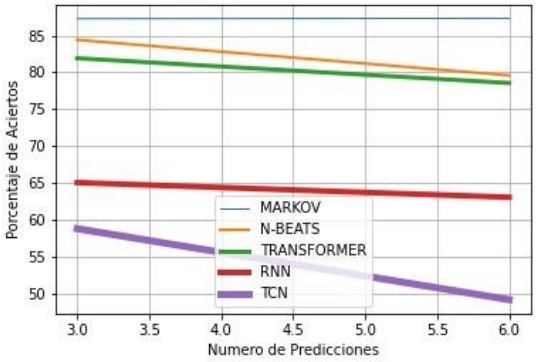

Figure 28. Percentage of hits for each model for state 0 by number of predictions.

Based on Figure 28, it is validated that the Markov chain model is the one that best predicts state 0 and is almost invariant accuracy with the number of predictions, while the machine learning models that best predict this state are TRANSFORMER and NBEATS, the RNN model also has an almost invariant accuracy when increasing the number of predictions, although it is lower than the other 2.

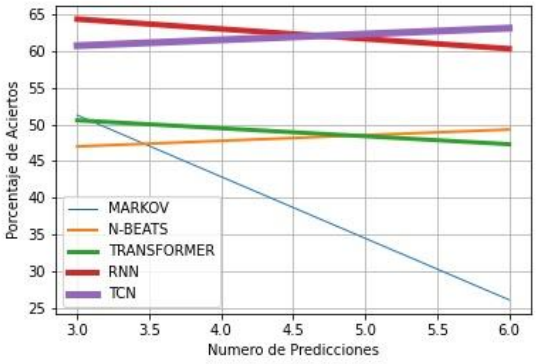

Figure 29. Percentage of hits for each model for state 1 by number of predictions.

Figure 29 validates that the TCN and RNN models are the best predictors of state 1, and do not decrease much with an increasing number of predictions. The NBEATS and TRANSFORMER models have a lower accuracy but are also invariant with an increasing number of predictions. Finally, the Markov chain model has lower accuracy and decreases dramatically with an increasing number of predictions.

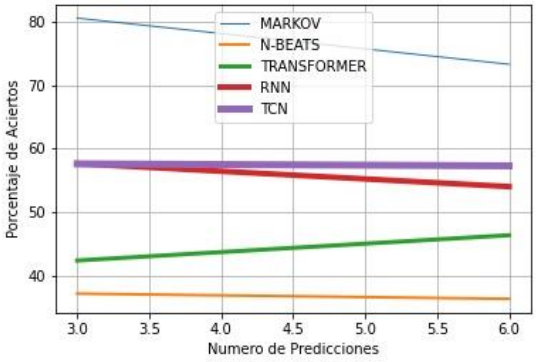

Figure 30. Percentage of hits for each model for state 2 by number of predictions.

For state 2 based on Figure 30, it is validated that Markov is the best predictor, but it decreases with the increase in the number of predictions, while the TCN and RNN models have a lower percentage of correct predictions but not as low as the other machine learning models.

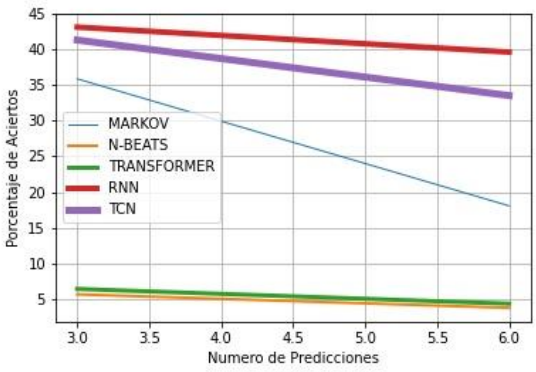

Figure 31. Percentage of hits for each model for state 3 by number of predictions.

For state 3, based on Figure 31, it is validated that the RNN model is the one that best predicts and the accuracy does not vary much as the number of predictions increases. On the other hand, the TCN model has a larger decrease, but the most drastic decrease is in the Markov chain model, but the accuracy is higher than the other machine learning models at all times.

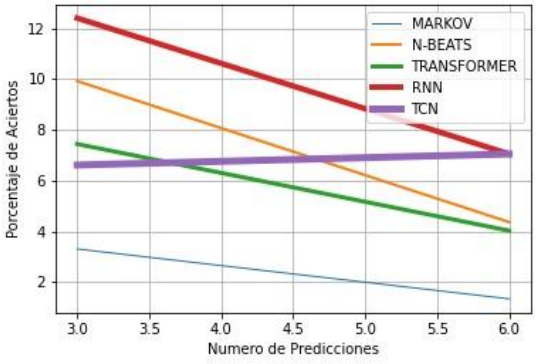

Figure 32. Percentage of hits for each model for state 4 by number of predictions.

For state 4, based on Figure 32, it can be seen that all models have very low accuracy, although the best one is the RNN model, although there is a pronounced decrease with the increase in the number of predictions.

5. Conclusions

The first thing that can be concluded is that with the data available from an Ecuadorian Bank, the Markov Chain model for a prediction of up to 6 months is the one that has a lower error in the predictions and shows a higher accuracy, although it does not describe the behavior of the data as well and has an unbalanced confusion matrix.

It was also observed that of the 4 machine learning models used, the RNN model, which are recurrent neural networks, has the best behavior and is quite similar to that of Markov chains. Although it has a higher error and lower accuracy, it describes the behavior of the data better and has a more balanced confusion matrix.

The other 3 machine learning models, when analyzed with the segmentation of 8 states for the payment of installments, do not show good results. However, when segmenting with only 4 states, the predictions of the NBEATS and TRANSFORMER models improve considerably.

In general, all the models perform better when segmenting the payment status of each installment into 4 states instead of 8. However, as indicated above, the usefulness of predictive models for financial institutions is reduced and information on the behavior of the time series describing the payment of a client's installments is lost.

It is concluded that none of the models, when trained with 8 states, proves to be very robust, although the Markov Chains model does not show considerably better behavior. It is validated that this is partly because the available data does not prove to be a balanced sample; that is, it has a similar number of observations of each of the states that can take the predictions.

It was also observed that there appears to be missing information in the available data, as there are problems with the dates on which payments are expected to be made and the installment number. Additionally, some loans do not show all the payment information but are truncated at a certain point, which generates a

lack of data that does not allow us to have a more balanced data sample.

Finally, based on the results obtained, it can be said that the Markov chain model is the one that gives the best results with the data available from the Ecuadorian Bank from which the data was obtained. However, the machine learning models, especially the RNN model, showed very similar results, so it can be deduced that if the data could be better maintained and completed, the machine learning models could give us a much better result.

The main recommendation that can be made is that the Ecuadorian Bank from which the data was obtained needs to be more rigorous in maintaining the data. This can be achieved with a more rigorous information management system that allows the bank's personnel to enter the information more accurately and without errors, and that is complemented by an information systems department that keeps people aware of the functioning of its infrastructure and the importance of having real and complete information in its databases.

In addition, when the confusion matrices of each model were analyzed, it was observed that certain models gave better results for predicting certain states. For this reason, for future work, a mixed model can be created that allows using a mixture of the 5 models, using the strengths of each one to produce a model that allows obtaining a better result in terms of the predictions obtained.

This can be done using the 4 machine learning models that were developed in this work and based on how well each model predicts each of the states a function can be developed that allows choosing the best prediction of the 4 that were made and thus create a resulting time series that has higher accuracy than any of the 4 models separately. This solution would be similar to that of Figure 33.

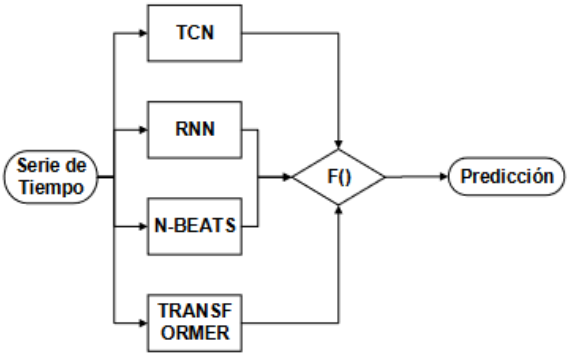

Figure 33. Proposed Diagram for Future Work.

References

[1] R. F. Kosobud and H. H. Stokes, "OPEC short-run market share behavior: Implications, theories and facts," *Energy Econ.* vol. 2, no. 2, pp. 66-80. 1980.

[2] A. A. Markov, "Extension of the law of large numbers to dependent quantities (in Russian)," *Izv Fiz-Matem Obs. Kazan Univ (2nd Ser)*, vol. 15, pp. 135-156, 1906.

[3] C. J. Geyer, "Practical markov chain monte carlo," *Stat. Sci.* , pp. 473-483, 1992.

[4] T.-C. Lee, G. G. G. Judge, and A. Zellner, "Estimating the parameters of the Markov probability model from aggregate time series data," 1970.

[5] G. A. Timofeeva and N. A. Timofeev, "Forecasting credit portfolio components with a Markov chain model," *Autom. Remote Control*, vol. 73, no. 4, pp. 637-651, 2012.

[6] W. L. Winston and J. B. Goldberg, *Operations research: applications and algorithms*, vol. 3. Thomson Brooks/Cole Belmont, 2004.

[7] T. Mitchell, "Machine learning," 1997.

[8] V. Thyagarajan and M. Saiful, "Retail banking loan portfolio equilibrium mix: A Markov Chain model analysis," *Am. J. Appl. Sci.* vol. 2, no. 1, pp. 410-419, 2005.

[9] A. Bellotti, D. Brigo, P. Gambetti, and F. Vrins, "Forecasting recovery rates on non-performing loans with machine learning," *Int. J. Forecast.* Vol. 37, no. 1, pp. 428-444, 2021.

[10] T. Bellotti and J. Crook, "Loss given default models incorporating macroeconomic

variables for credit cards," *Int. J. Forecast.* , vol. 28, no. 1, pp. 171-182, 2012.

[11] G. Petneházi, "Recurrent neural networks for time series forecasting," *arXiv Prepr. arXiv1901.00069*, 2019.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.* vol. 9, no. 8, pp. 1735-1780. 1997.

[13] K. Cho *et al.* , "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv Prepr. arXiv1406.1078*, 2014.

[14] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery \& data mining*, 2019, pp. 2828-2837.

[15] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Trans. neural networks*, vol. 5, no. 2, pp. 240-254, 1994.

[16] Y. He and J. Zhao, "Temporal convolutional networks for anomaly detection in time series," in *Journal of Physics: Conference Series*, 2019, vol. 1213, no. 4, p. 42050.

[17] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *Electronics*, vol. 8, no. 8, p. 876, 2019.

[18] P. Lara-Ben\'\'\itez, M. Carranza-Garc\'\'\ia, J. M. Luna-Romera, and J. C. Riquelme, "Temporal convolutional networks applied to energy-related time series forecasting," *Appl. Sci.* vol. 10. no. 7, p. 2322, 2020.

[19] B. N. Oreshkin, G. Dudek, and E. Pełka Paweł and Turkina, "N-BEATS neural network for mid-term electricity load forecasting," *Appl. Energy*, vol. 293, p. 116918, 2021.

[20] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," *arXiv Prepr. arXiv1905.10437*, 2019.

[21] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q.* , pp. 75-105, 2004.

[22] A. Vaswani *et al.* , "Attention is all you need," *Adv. Neural Inf. Process. Syst.* , vol. 30. 2017.

[23] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv Prepr. arXiv1803.01271*, 2018.