

CBA Policy Rate Path Forecasting

Authors: Zhora Stepanyan
BS in Data Science
American University of Armenia
Yerevan, Armenia
zhora_stepanyan@edu.aua.am

Supervisor: Arsen Grigoryan
Head of Asset, Liability and Capital Management Division
Ameriabank CJSC
Yerevan, Armenia
arsenagrigoryan@gmail.com

Abstract—This study presents a comprehensive analysis and predictive modeling of the Central Bank of Armenia’s (CBA) policy rate path. Utilizing advanced data scraping techniques and sophisticated machine learning algorithms, this research aims to provide market participants, investors, and policymakers with a robust tool for strategic economic planning and risk management. Historical data from various sources, including the official CBA website, and rates from neighboring countries such as Georgia and Russia, were meticulously compiled and analyzed. The study employed several machine learning models from Scikit Learn and some AutoML techniques, with automatic machine learning showing the most promise due to its superior handling of complex datasets and ability to mitigate overfitting. The results indicate that precise and timely predictions of policy rates are crucial for effective economic decision-making and underscore the significant role of advanced analytics in financial policy planning. In this paper you will see my analysis and work for finding the CBA policy rate forecasted path.

I. INTRODUCTION

The Central Bank of Armenia’s policy rate expected path is of much interest to market participants. It is a pivotal economic indicator that significantly influences local financial markets, especially the bond market. For market participants, individual investors, and financial institutions, accurate anticipation of the refinancing rate change is very important for strategic and tactical asset allocations and risk management. Considering all the mentioned reasons for the level of importance of the refinancing rate in our everyday life, it is becoming an invaluable asset to have the predicted/expected path of the CBA policy rate. By integrating historical data and possible significant features that can help the process of prediction, the algorithm helps in preemptively adjusting portfolio strategies to align with the anticipated economic conditions dictated by the policy rate changes.

Moreover, for portfolio managers, this predictive capability means they can better hedge against potential risks associated with interest rate risk and optimize their asset allocations for maximum performance. This is particularly important in a dynamic economic environment where timely and accurate information can significantly impact the financial outcomes of investment decisions as Mishkin and Serletis discuss the foundational elements of financial markets and their sensitivity to policy changes [3]. Thus, the development of a reliable forecasting algorithm for the CBA’s policy rate is not merely a technical achievement but a strategic enhancement that

could lead to more robust and resilient financial planning and execution in Armenia’s bond markets.

II. DATA

The most important, difficult, and time-consuming work was the data preparation part. For this part of the project, I spent a lot of time and used different sources for retrieving every column of my final data. My main column is the column of Refinancing rate or the so-called CBA policy rate which is the main component and target variable of my machine learning model that I will introduce to you later in this paper. Data from the Central Bank of Armenia (CBA) was crucial for the analysis, aligning with documented practices in economic data utilization as described in Bernanke’s exploration of monetary policies [5].

The prediction process of the target variable is quite an important part, but most of the prediction importance relies on the chosen feature set because it is the crucial part of the decision-making process of how much is going to be the rate in the future.

For choosing features I started by studying the scholarly articles demystifying the central bank policy rate and the variables influencing it for the major central banks in the world and later have dived into observing some articles from CBA official website (www.cba.am). After all, I concentrated the “Minutes on Refinancing Rates” articles by CBA press release that summarizes the main points about Central Bank Board meetings. The main idea of the Minutes by CBA press release is the discussion of external sector observations, Armenian economic developments, financial sector, and Inflation Developments. The most important part of the release is monetary policy decisions which include the decision made for changing the refinancing rate path. The release also focuses on future outlooks indicating that the CBA will continue monitoring the economic indicators and adjust policies as necessary to ensure price stability and absorb any risks that might lead to deviations from the inflation target.

Initially, I had an intent of incorporating natural language processing of those minutes, deriving the sentiment of policymakers and further analyzing its impact on policy rate setting. However, taking the limited time I had, I preferred to shift the focus to other variables influencing the policy rate.

So, reading and summarizing a couple of “Minutes” on Refinancing Rates by the press release of CBA helped me

to make the decision on which variable datasets to include as factors in my main data.

A. Introduction to the Main Data

In this section, I am going to introduce you to my data, define column names, and dive deep into each column creation process. So, I have 12 columns in the data, 9 of them are of type float, 2 of them are of type int and one of the columns is of datetime format. The data contains 221 rows x 12 columns at all.

Although my Refinancing rate column data was available from 2003-12-01 until now, I filtered to have it from 2006-01-01 because most of the features were available from that date on the official website of the CBA. So, in this case, I focused on the width of the data rather than on the length of it, paying more attention to having more features that can contribute to the accuracy of my future models. Also, excluding less than 3 years from my data is not going to cause data scarcity, when I have more columns.

Time_Period	Refinancing_Rates	georgia_rate	Russia Rate	USD Dollarization	Core_Inflation_1	Core_Inflation_2	CPI	Year	Month	Reserves		
0	2006-01-01	3.50	3.50	12.0	451.742500	1.511290	-1.405190	-1.256455	-2.973920	2006	1	681.873684
1	2006-02-01	4.00	4.00	12.0	450.110000	1.536412	-0.916822	-0.898108	-2.529050	2006	2	669.368861
2	2006-03-01	4.00	4.00	12.0	450.998261	1.542730	-0.395880	-0.552381	-2.043668	2006	3	674.680634
3	2006-04-01	4.00	4.00	12.0	449.985263	1.548592	1.191824	0.872023	-1.509252	2006	4	680.035795
4	2006-05-01	4.00	4.00	12.0	439.731905	1.561499	2.216325	1.712238	-0.729713	2006	5	680.035795
...
216	2023-12-01	9.25	10.00	16.0	404.474792	2.460454	-0.393539	0.704472	1.980419	2023	12	2251.613260
217	2024-01-01	8.75	9.50	16.0	404.960952	2.448313	-0.440835	0.500455	-0.858544	2024	1	2251.613260
218	2024-02-01	8.75	9.00	16.0	404.268095	2.449819	-0.654158	0.132493	-1.255498	2024	2	2251.613260
219	2024-03-01	8.50	9.00	16.0	400.418000	2.516169	-0.709863	0.132230	-1.237474	2024	3	2251.613260
220	2024-04-01	8.50	8.25	16.0	391.970000	2.516169	-0.709863	0.132230	-1.100000	2024	4	2251.613260

221 rows x 12 columns

Fig. 1. Printed Main Data.

So, my data columns are refinancing rates of Armenia, Georgia, Russia, and USD rates over time with its Dollarization column as well. Also, I have some important economic factors like core inflation, consumer price index(CPI), and international reserves of Armenia in millions USD. In the end, I also separated year and month from the “Time_Period” column to have separate contributions to the prediction process. I will introduce each column’s meaning and its format of numbers in depth in further sections.

B. Data Columns

First of all, let’s start with the first column called “Time_Period” which is just a simple date time format column that serves as a time factor for the prediction process. All columns of data have been brought to the same dimension, the whole data is converted to monthly. The Time_Period of every row is the first day of each month indicating more right-sorted data.

The column next to the “Time_Period” column is the main target column called “Refinancing_Rates”. This column represents the CBA policy rate historical data in percentage form. Below you can see the graph of refinancing rates from the end of 2003 year to 2024 year our days. The path of refinancing rates is quite interesting and in every part, each big rate change is explainable from extraordinary events that

happened during these dates. The reasons for the strict rise and fall parts around 2014-2015 and after 2020 years are easily understandable. If we investigate deeper into these parts we can see that refinancing rates went from 6.75 to 10.5 and from 4.25 to 10.75 respectively. The main reason for the first sharp go-up was mainly due to the USD/AMD exchange rate free fall from 408 to 481. The other strict refinancing rate change happened after the 2020 year, the main causes of which were COVID-19 and the 2020 Artsakh war period that resulted in economic tragedy.

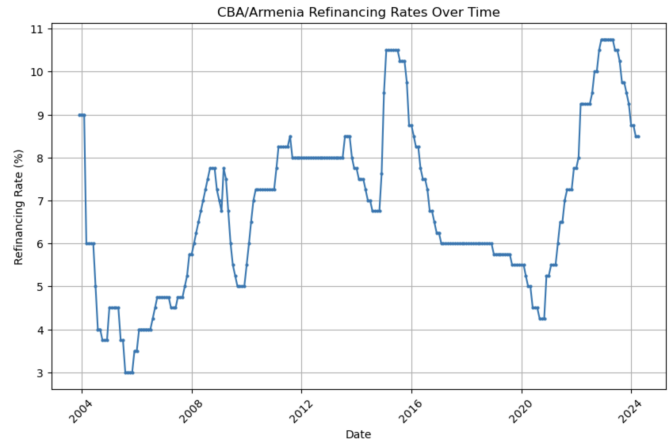


Fig. 2. CBA Policy Rate Historical Path

With the above-mentioned causes, I took into consideration also the USD/AMD rate as well which helped a lot during the prediction process. This rate seems to be of major focus by Armenian policymakers due to its vital impact on inflation and the financial stability of the banking system.

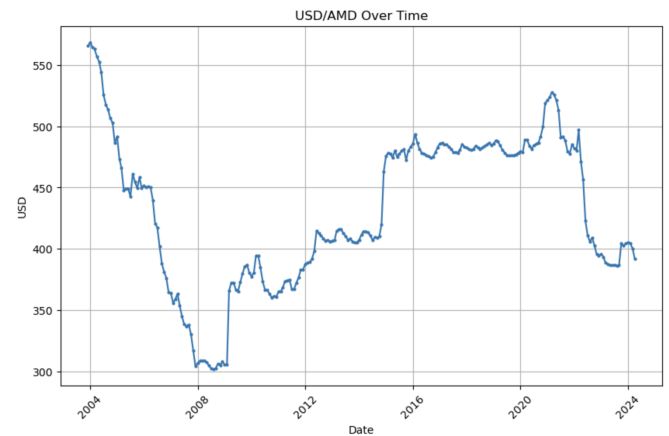


Fig. 3. USD/AMD Rate Historical Path

Besides having the main focus on monetary policy setting in Armenia, I have analyzed whether the policy rate moves in the region have an influence on the decisions by the CBA (i.e. whether earlier policy move in Georgia or Russia somewhat influences further decision-making in the CBA). The two columns after the “Refinancing_Rates” column are the Georgia

and Russia refinancing rate columns correspondingly. After some analysis of the historical paths of the rates of central banks of these countries, we understood that adding these datasets to the main data will play a crucial role because they have the same trend of rate change especially during the previous couple of years. Below you can see the graph containing these rates from 2003 to 2024 with some obvious trends that all of them have in common. If we take specific parts of the graph below we can see that during the 2015 period, all central banks of these countries have made the same decision to raise rates. After that, the part when the rates decrease is almost working with the same principle. Especially, if we take into consideration the part of the graph after 2020 we can see that Georgia and Armenia move with the same path with little time deviation. Russia is also moving with the same trend, with a little difference: it is doing a very strict raise to 20%, and lowering the rate by bringing it to the same number as before. After analyzing the above-mentioned parts of the graph it is becoming obvious that Armenia's central bank decisions are similar to the ones that Georgia and Russia have with a a little different time deviation. So, we decided that adding the rates of these countries to the main data would enhance the weight and value of features and would help in the future of the prediction process. Below you can see the graph of refinancing rates of Armenia, Georgia, and Russia with green, blue, and red colored lines correspondingly:

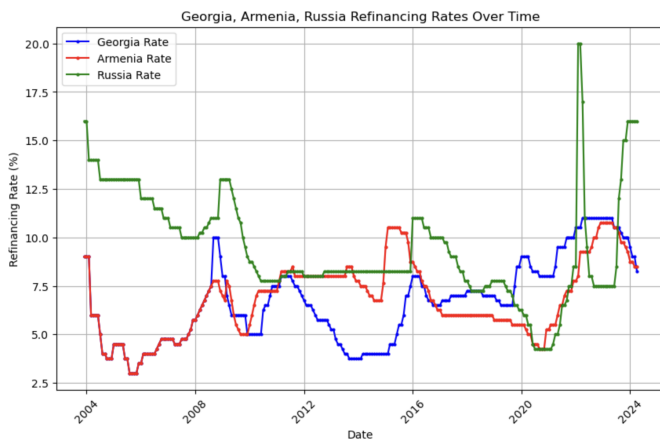


Fig. 4. Armenia, Russia, Georgia Refinancing Rates

Like the reason mentioned above that the USD/AMD rate was added I have also included the Dollarization factor in the data. Dollarization is the term for when the U.S. dollar is used in addition to or instead of the domestic currency of another country. Dollarization tends to be the driving factor for exchange rate depreciation in turbulent times. In terms of the Dollarization of CBA, I have chosen the "D2" type of dollarization from 5 types of dollarization. D2 is the ratio of foreign currency deposits and loans of residents to total deposits and loans of residents. So, here in the main data dollarization column, we've got the D2 factor presented which was the one that was needed in terms of its description.

The other two columns next to the Dollarization column are Core inflations represented in two different formats. In the original data of core inflation, we had the numbers around the 100 range. In my main data, the core inflations are represented as a result of the number in the original data of the core inflation column deduced by 100. The first format of core inflation in the main data is calculated using the seasonal products & administrative regulated services prices exclusion method which is used for the CBA monetary policy-making process. CBA monetary policy decisions also include decisions related to the refinancing rate change. The second core inflation column is calculated using the exclusion and adjustment method which is used for other analytical purposes. So, both "Core_Inflation_1" and "Core_Inflation_2" are numbers representing the percentage inflation change compared to the same month of the previous year. As the data of core inflation was very important for further prediction process of my target variable, it forced me to change my data range from 2003-12-01 to 2006-01-01 because of having the inflation data available from the 2006 year.

Next to the core inflation columns is the CPI column which is the consumer price index in percentage compared to the same month in the previous year. Here we used the same principle of deducing 100 from each month to have the percentage change of CPI compared to the previous year's same month like core inflations. CPI is used to estimate the average variation between two given periods in the prices of products consumed by households. In short, CPI is the instrument to measure inflation which is used for CBA monetary policy-making processes.

Year and Month columns are separated manually to affect the prediction process too. The idea that I have monthly trends in my data and separating year and month from the time_period column can potentially help the machine learning model capture those patterns more effectively. So, separating year and month from time_period was beneficial for the model performance which was proved with statistical information about the model's performance like MSE (Mean Squared Error) and other measures as well. The measures showed that having Year and Month columns separately in my data causes the statistical numbers of the model's performance to be lower than compared to the case when the data is without these important columns. Also, when checking the feature importance feature of the selected model shows quite good numbers for these two separated columns: they appear to be in the top rows of the feature importance data frame.

The last column that I have added to my data is the Reserves column which is the Reserves of Armenia converted into U.S. dollars with exchange rate at the time of conversion. As known to us these numbers of Reserves of Armenia in million USD can also help with the prediction and model performance because decisions related to the refinancing rate change are made based on the number of reserves compared to the previous month that our country has.

C. Data Creation Process

As my solution has been intended for use by professionals in the financial sector, an automatic retrieval option for the data is a must. Hence, I have chosen the live web scraping of the data. My main source used for scraping the data is the official CBA website www.cba.am. Besides the columns on rates of Georgia and Russia, all other columns are from CBA's official website. The process of getting data from the CBA website was not a difficult one because the data is publicly available to everyone and I had full access to it. I have used pandas functions to directly read the .xlsx and .csv files from the website URLs. Each data imported from the CBA website was prepared individually. During my work with datasets from CBA, I noticed an important fact that the datasets of this website are created individually and do not have one standard template for all of them, and almost all of them were messed up. Bringing different columns of different datasets together into one main data was not an easy data preparation task. I tried to solve the preparation issues using Pandas and Numpy simple operations to bring all datasets into one single data with a common Time_Period column. I also used the MonthBegin library from pandas.tseries.offsets for making every row date start from the first day of each month to make it easy for the joining process. Also, used the relativedelta library from dateutil.relativedelta package for date format multiplication issues. In the end, when it came to the part of joining each column separately to the main data, I used pd.merge() function from the pandas library and joined the columns with matching dates.

For the Georgia refinancing rates column, I used the website of the National Bank of Georgia <https://nbg.gov.ge/en>. By performing the same type of operations on the .csv file retrieved from the official Bank of Georgia website, I created the Georgia refinancing rate historical data from the 2008 year. Before the 2008 year, I used the CBA rate historical data to join them together and have the complete data. The reason that I merged the CBA rate data before 2008 and NBG(National Bank of Georgia) rate data after 2008 is to have the same dimensions as my initial refinancing rate column. Initially, both Georgia and Armenia rate paths look the same, that's why I acted in that way to have the Georgia rate included in my data with its full matching length.

The data on the refinancing rate of the Central Bank of Russia is made in a completely different way compared to all other columns because I used a web scraping method for it. Web scraping method means sending HTTP requests to the link and getting back responses. I did the scraping part using famous libraries called BeautifulSoup and requests. With a simple scraping code, I got the data on Russia's refinancing rate from two places. The first one was the official historical record of refinancing rate which was available until 2016. The second part of data after 2016 is called Key Rate which I found from a different place located on the CBR website. After finding both parts from different places I joined them together and had the full Russia refinancing rate data starting from 2003

to 2024.

After taking the above-mentioned steps I was ready to go with the main data through the Machine Learning(ML) part which I will describe in detail in the next sections.

III. METHODOLOGY

In this section, I am going to introduce you to the methods that I used for the prediction process which is mainly about the Machine Learning algorithms. Before moving to the models that I have used in my project I would like to introduce the pre-processing setting of my models. I have separated my features and the target into separate variables after which I have performed train and test split with a test size of 20% of my data. As my data is a time series data, I did the train and test splitting part using a split point.

As already known my target variable is the Refinancing_Rates column and the features are the rest of the data without the 'Time_Period' column because I have already separated the Year and Month columns to have better results.

As I intend to make this app useful for practitioners in the financial sector, the app should have an option to retrain the ML model any time in the future.

Hence, I have opted for going into 2 paths: manually selecting the models that I have assured will perform well and allowing users of the app have the option to make use of AutoML, introducing new models into the scope as data relationships allow for such.

A. Manual Prediction

First of all, after getting the whole ready data in hand I started my testing with the ML models from Scikit-Learn library. The main models I have run my data with are GradientBoosting, RandomForest, and Support Vector Machines(SVM). The use of machine learning techniques such as Gradient Boosting and SVM is well-documented for their effectiveness in complex datasets, as noted by Géron [2].

Gradient Boosting is a powerful ensemble machine learning technique that builds models incrementally in the form of an ensemble of weak prediction models, typically decision trees. Gradient Boosting is particularly useful for datasets with heterogeneous features like numeric and categorical and is generally robust against overfitting, especially in large datasets. For my specific case, this model might help capture complex patterns in data changes over time, adjusting to economic indicators dynamically. With Gradient Boosting I had pretty good results compared to the other models from Scikit Learn Library that you can see below:

TABLE I
GRADIENT BOOSTING MODEL PERFORMANCE METRICS

Metric	Value
MSE	5.355298608561662
RMSE	2.31415181190899
MAE	1.9343055055682665
R-squared	-0.22888844368314398

The results shown above are the main statistical records that I used for model performance definition. MSE stands for Mean Squared Error, RMSE stands for Root Mean Squared Error, and MAE stands for Mean Absolute Error. The performance metrics such as MSE and RMSE are standard for evaluating forecasting accuracy, a methodology supported in both academic and practical contexts [7]. The results of these numbers should be as low as possible.

The next model that I ran on the main data was Random Forest. Random Forest is an ensemble learning method for classification and regression that operates by constructing a multitude of decision trees at training time and outputting a class that is the mode of the classes(classification) or mean prediction(regression) of the individual trees. This model is less likely to overfit the training data, making it a robust choice for many problems. For my data, it could provide robust predictions against the volatility in economic data due to its ensemble nature, reducing the risk of overfitting to more stable trends. You can see the results of the Random Forest model in Table II.

TABLE II
RANDOM FOREST MODEL PERFORMANCE METRICS

Metric	Value
MSE	9.381345138888888
RMSE	3.06289816005836
MAE	2.5745
R-squared	-1.1527514094083315

The results shown in the table to be less effective than the ones I had with Gradient Boosting because the numbers are higher than expected.

Support Vector Machine model was the next one that I ran on my model and got the prediction results. Support Vector Machines(SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane that categorizes new examples. SVM was used for one main reason the relationship between features and policy rates is not straightforward or linear. It could also be effective for using kernel methods to address non-linearities in my data. But the results with SVM were better than one I had with RandomForest.

TABLE III
SUPPORT VECTOR MACHINES PERFORMANCE METRICS

Metric	Value
MSE	6.161995949342507
RMSE	2.482336792085737
MAE	2.201667276059712
R-squared	-0.41400249839722103

So, SVM was automatically deducted from the list of potential models that could work on my data, because its results are not better than the ones last two models have.

In the end, after trying all of these models on my data I decided to create a stacked ensemble with the three estimators Random Forest, Gradient Boosting, and SVM and run on my model with the hope of having the best results over the previous ones. As we know stacked ensemble or stacking is an advanced machine learning technique that combines multiple predictive models to generate a new model. Typically, individual models, known as base learners are trained on the same data, and their predictions are used as inputs to a final model known as meta learner to make the final prediction. So, I used Gradient Boosting, Random Forest, and Support Vector Machines to create the stacked Ensemble. The results are shown below:

TABLE IV
STACKED ENSEMBLE PERFORMANCE METRICS

Metric	Value
MSE	5.07025012453242
RMSE	2.2517215912568807
MAE	2.052115723389251
R-squared	-0.16347793840283842

After comparing all of the results of the models by Scikit Learn and the result of the stacked ensemble I understood that my best performance model is the Stacked Ensemble because it unites all the strengths and weaknesses of these three models that contribute to the final prediction, with especially their strengths.

B. Automatic Prediction

The results of the Gradient Boosting model were not satisfying and I tried to find a better model for my data. After long research, I came up with the idea of implementing automatic machine learning for my prediction. To get the best results I could, I started the implementation of the so-called H2O Auto ML library to my data. H2O AutoML is an automated machine learning framework designed to handle many aspects of the machine learning workflow including data pre-processing, feature engineering, model training, and validation. It simplifies the process of machine learning models by automatically selecting the best model and tuning parameters. H2O AutoML efficiently explores numerous models and their hyperparameters to find the optimal solution within a specified time limit or number of models. This includes a wide range of models from simple linear regressions to sophisticated ensemble models. One of the strengths of H2O AutoML is the ability to automatically combine predictions from a diverse set of models to improve the final prediction accuracy, typically using stacked ensembles. The benefits of using this kind of solution for my data are its efficiency, performance, and scalability. H2O's AutoML is particularly suited for this application, providing efficient model selection and tuning as detailed in the H2O documentation [6]. I have tried this tool with the same setting as the SKLearn models described above. So, I divided my data into train and test separate data with a test size of 20% as done during the manual

prediction part. With the statistical data below you can see how well the models offered by this tool fit the best with my data:

model_id	evjance	rmse	mse	mae	rmste	mean_residual_d
StackedEnsemble_All1Models_1_AutoML_1_20240506_12607_0404698	0.201171	0.0404698	0.138948	0.0265638	0.	0.
StackedEnsemble_BestOfFamily_1_AutoML_1_20240506_12607_0405962	0.201485	0.0405962	0.1365	0.0256996	0.	0.
GBM_5_AutoML_1_20240506_12607_0408205	0.228954	0.0488205	0.125464	0.0272936	0.	0.
GBM_2_AutoML_1_20240506_12607_model_2_060432	0.245829	0.060432	0.162958	0.0331222	0.	0.
GBM_7_AutoML_1_20240506_12607_0608736	0.246726	0.0608736	0.177695	0.034459	0.	0.
DNF_1_AutoML_1_20240506_12607_0718304	0.266515	0.0718304	0.173448	0.0329763	0.	0.
DeepLearning_grid_2_AutoML_1_20240506_12607_model_1_0711824	0.2668	0.0711824	0.193236	0.0355084	0.	0.
GBM_4_AutoML_1_20240506_12607_0768549	0.27578	0.0768549	0.187272	0.0374117	0.	0.
GBM_grid_1_AutoML_1_20240506_12607_model_3_0717149	0.277804	0.0771749	0.202458	0.0373051	0.	0.
XGBoost_2_AutoML_1_20240506_12607_0786314	0.288413	0.0786314	0.196837	0.037522	0.	0.

[27 rows x 6 columns]

stackensemble prediction progress: ██████████ (done) 100%

Fig. 5. AutoML Models Performance Metrics

As you can see the best results with AutoML are almost 10 times better than the one I had with SKLearn models. The chosen model for the final prediction is displayed in the image above.

One more thing I would like to pay attention to is the functions called smoothed_transition and custom_round. The first one called smooth_transition is designed to have a smoother transition or gradual change of numbers from the last element of historical data to the first element of the predicted one with 10 steps. As the CBA policy rate has not changed by more than 1% because of the decisions of the CBA board in its history, it was necessary to have this function for having a more realistic prediction path for the refinancing rate. The other custom round function is made for having numbers rounded to 0.00, 0.25, 0.5, and 0.75 float points because the refinancing rates always have numbers ending with these float parts. So, both of these functions are used after the automatic prediction to have more realistic refinancing rates defined by the CBA.

IV. RESULTS

In this section, I am going to show you my prediction results with each model described in both the manual and automatic prediction sections of the methodology part. I have filtered out to have the predictions for 1 year afterward the last historical point that the data has at that moment. Below you can see the predictions path of refinancing rates by manual predictions:

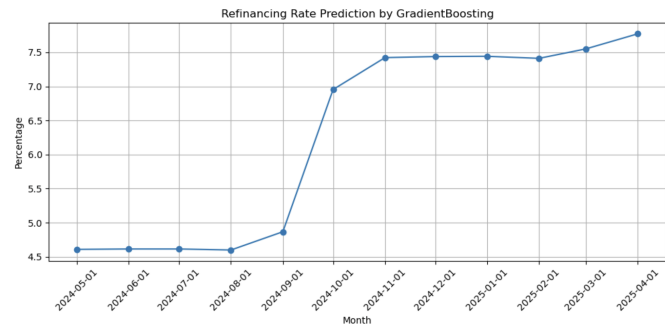


Fig. 6. GradientBoosting Model's Prediction

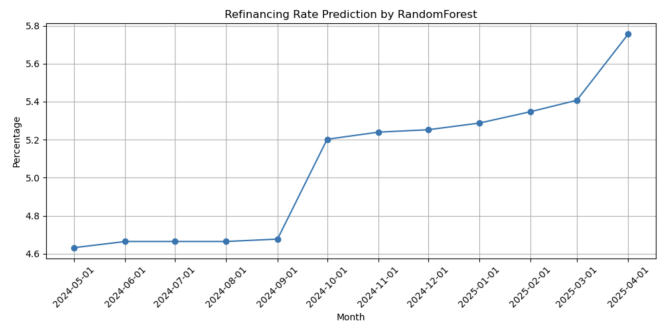


Fig. 7. RandomForest Model's Prediction

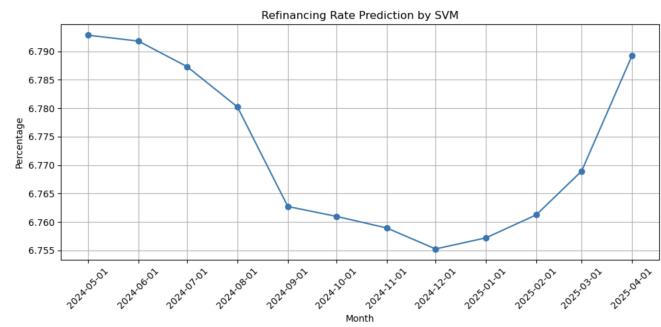


Fig. 8. SVM Model's Prediction

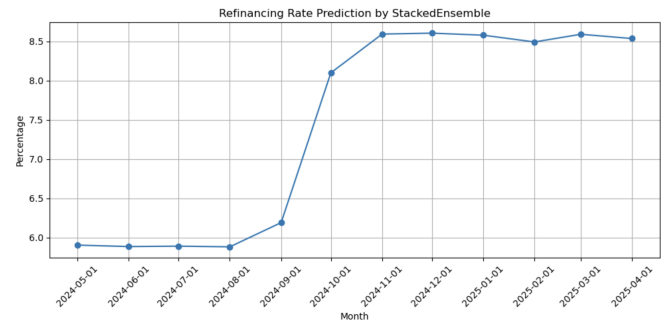


Fig. 9. StackedEnsemble Model's Prediction

As discussed above all of these prediction paths that we have do not have as many good model performance results and for the final prediction, I have chosen the results given by the H2O AutoML. The fan chart of the final prediction that includes uncertainty levels with red lines you can see below:

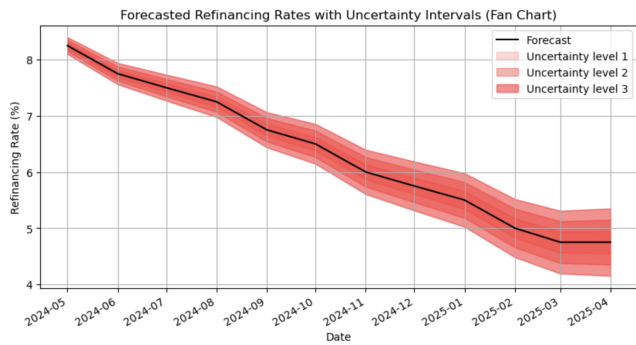


Fig. 10. Final Prediction Fan Chart

As we can see the refinancing rates are going to decrease for the next 11 months and stop decreasing at 4.75

I also created a feature importance chart to see which feature is considered the most important during the prediction process with the feature importance feature that AutoML offers. Below is the list of features sorted in descending order by the first model of the list the picture of which I placed in the upper part of the report that has the features of displaying the importance:

	variable	relative_importance	scaled_importance	percentage
0	Year	146.899841	1.000000	0.392081
1	Core_Inflation_1	54.529362	0.371201	0.145541
2	Russia Rate	42.419460	0.288765	0.113219
3	Dollarization	41.636120	0.283432	0.111128
4	USD	32.146015	0.218829	0.085799
5	Reserves	14.913290	0.101520	0.039804
6	georgia_rate	14.410746	0.098099	0.038463
7	Core_Inflation_2	11.914435	0.081106	0.031800
8	CPI	11.654952	0.079339	0.031108
9	Month	4.142518	0.028200	0.011057

Fig. 11. Feature Importances by Descending Order

V. CONCLUSIONS

In conclusion, I have successfully developed a predictive model for the policy rate path of the Central Bank of Armenia, utilizing a blend of advanced live data scraping techniques and sophisticated machine learning algorithms. The project's core objective was to equip market participants, investors, and policymakers with reliable predictive tools to aid in strategic economic planning and risk management, which has been effectively achieved.

Throughout this research, extensive data has been gathered from multiple sources, emphasizing the importance of comprehensive and accurate data collection in economic forecasting. Key predictors identified—such as the refinancing rates of neighboring countries like Georgia and Russia, and local economic indicators including core inflation and consumer price index—highlight the interconnected nature of global and local economic factors.

The machine learning phase of the project evaluated several models, with Stacked Ensemble method emerging as the most effective in terms of predictive accuracy, as measured by Mean Squared Error, Root Mean Squared Error, and other relevant metrics. The exploration into ensemble techniques comprising Gradient Boosting, RandomForest, and SVM, demonstrated

marginal improvements in prediction accuracy, reinforcing the strength of combining multiple model predictions.

The implementation of H2O AutoML represented a significant enhancement in the modeling process, automating the selection and tuning of models, which not only streamlined the analytical workflow but also improved the outcomes. The superior performance of H2O AutoML underscores the potential of automated machine learning tools in handling complex predictive tasks efficiently and with high accuracy.

For future work, exploring additional machine learning models and expanding the dataset to include more granular temporal resolutions or additional predictive factors could provide deeper insights and potentially more accurate forecasts. Further, the integration of real-time data analysis and the adoption of more dynamic modeling techniques could enhance the responsiveness and relevance of the forecasting tools developed in this project.

In conclusion, I not only achieved my initial goals but also laid down a promising pathway for future advancements in economic forecasting using machine learning. It highlighted the critical role of accurate data, the potential of machine learning in economic analysis, and the power of automation in enhancing predictive modeling. This project stands as a testament to the potential of integrating advanced data science techniques in the field of economics for robust and resilient financial planning and policy-making.

REFERENCES

- [1] E. Dabla-Norris and H. Floerkemeier, "Transmission Mechanisms of Monetary Policy in Armenia: Evidence from VAR Analysis," November 1, 2006. [Online].
- [2] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2019.
- [3] F. S. Mishkin and A. Serletis, *The Economics of Money, Banking, and Financial Markets*, 4th Canadian ed. Toronto, ON: Pearson Canada, 2010.
- [4] C. McConnell, S. Brue, and S. Flynn, *Macroeconomics*, 21st ed. New York, NY: McGraw-Hill Education, 2020.
- [5] B. S. Bernanke, *21st Century Monetary Policy: The Federal Reserve from the Great Inflation to COVID-19*. New York, NY: W.W. Norton & Company, 2022.
- [6] H2O.ai, "H2O Documentation," Accessed on: 2024. [Online]. Available: <https://docs.h2o.ai>
- [7] Scikit-Learn, "Scikit-Learn Documentation," Scikit-Learn 0.21 Documentation, Accessed on: 2024. [Online]. Available: <https://scikit-learn.org/0.21/documentation.html>
- [8] GitHub Link, Available: <https://github.com/zhora-707/CBA-Rate-Path-prediction>