Proceedings of the 8th North American International Conference on Industrial Engineering and Operations Management, Houston, Texas, USA, June 12-15, 2023

Improvement of the Internal Audit Planning System in an IT Company through Predictive Analytics

Patricio Alejandro Zamora Navarro, Rafael Alejandro Ruiz Miller, Roberto Serna Zuazua and José D. Morcillo, Ph.D.
Escuela de Ingeniería y Tecnologías, Universidad de Monterrey, México
patricio.zamora@udem.edu, rafael.ruizm@udem.edu , roberto.serna@udem.edu,
jose.morcillo@udem.edu.mx

Abstract

Internal auditing is a fundamental process for quality assurance and compliance of processes within a company. For this reason, it is very important that companies can adequately plan its execution. One of the most important aspects to consider for effective planning is the calculation of the estimated time it takes to execute the audits. This project proposes the use of machine learning to develop a model that can predict the effort time (referred to as Actual Effort) that is required to carry out a certain number of audits in a period of two weeks (referred to as Sprint) in an information technology services company. In the early stages of the project, it was intended to develop an algorithm that predicts the Actual Effort for a single audit, but due to inefficient results, the approach of predicting the Actual Effort per Sprint was preferred. Many types of models were implemented along the project and various evaluation metrics, such as coefficient of determination (R^2), symmetric mean absolute percentage error (SMAPE), and others, were evaluated. The best results were obtained by an ETR model where an R^2 of 96% and a SMAPE of 8.47% were obtained.

Keywords
Audit Planning, Effort, Prediction Model, Machine Learning and Regression.

1. Introduction
According to Auren México (2021), an audit is "the process of reviewing and verifying the operation of a company, either comprehensively or in a specific aspect, which is carried out with a precise and detailed methodology to identify incidents, eventualities, diagnoses, panoramas and points of opportunity". Today, the importance of internal auditing in companies is more significant than ever because quality in all aspects of the company (also known as Total Quality Management) is now a necessity to maintain competitiveness in a market. Internal auditing is precisely responsible for making this a reality and verifying that the proper functioning of a company's internal processes are being met (Grimaldo, 2014). Additionally, among the many benefits of applying internal auditing, a company can improve risk management and its business operation (Sampattikorn et al. 2012).

Proper planning and organization in the execution of internal audits are necessary for their best use and effectiveness. Audit effectiveness contributes to proper procedure realization and operations of the company and, therefore, contributes to the whole organization's effectiveness (Dittenhofer 2016).

In this context, the following paper evaluates the current internal audit planning system of an information technology services company. Currently, the company has a very limited capacity on the part of internal auditors, therefore, internal audits need to be adequately planned. If a planned audit is not carried out, the repercussions could include financial penalties by the client, since in certain projects the number of audits to be carried out is contractually agreed. Likewise, it can generate dissatisfaction on the part of the client in the presence of errors and non-compliance with internal company policies.

Unfortunately, the current method used for planning the audits to be carried out by estimating the hours of effort available is deficient and imprecise. Based on a comparison between planned and actual hours for conducting audits in various 2022 sprints, on average, there is a 25% margin of error. The margin of error that exists in the hours of effort directly affects the performance and results of the auditing team in terms of the number of audits that are planned and those that are actually carried out and, as a consequence, the previously mentioned repercussions take place.

1.1 Objectives
Develop a predictive model that can estimate the hours of effort required to perform a certain number of audits in a sprint, in order to support the internal audit planning system of the company, and thus reduce the margin of error that exists between the planned Effort hours and actual effort hours by at least 10%.

To achieve the objective, the following particular objectives must be met:

- Determine the variables that result in the best effectiveness and accuracy of the predictive model.
- Implement multivariate predictive models that make possible to explain and predict Actual Effort in such a way as to reduce the current margin of error by at least 10%.
- Develop an user friendly Actual Effort prediction tool to support the company's internal audit planning system.

2. Literature Review
Estimating the effort required to carry out an audit is essential for the successful planning and execution of the internal audit system. The main purpose of literature research was to learn based on the experience and practice of other researchers and professionals, as well as to know the algorithms and metrics obtained from other studies like the one being carried out. This project focuses on supporting the planning system with the prediction of the hours of effort required to perform a certain number of audits in a sprint. Some of the challenges we encountered was the limited information available in professional papers or articles that covered the same topic of predicting hours of effort in audits. Some of the most relevant issues in some works were task execution times, estimation of hours of effort in projects, among others.

There are different evaluation methods and algorithms used to make these estimates, each with their advantages and disadvantages. Four methods are presented below: the expert-based method, the neural network-based method, and the decision tree model and support vector machine (SVM).

The expert-based estimation method uses the experience and knowledge of expert professionals in the field to make estimates, so some of its virtues are specialized knowledge, especially those who have direct experience with similar projects (Faria and Miranda 2012). Expert-based estimates may be biased by personal opinions and individual preferences. However, the estimates may be biased by personal opinions and individual preferences, impairing the objectivity of these.

Several studies have used machine learning models to estimate the duration of different tasks or projects based on different contexts. These models include neural networks, decision trees and other approaches (White and Hassan 2019; Lishner and Shtub 2022; Chen et al. 2010; Hamada et al. 2014).

In addition to many proposed techniques, Artificial Neural Networks and decision trees are methods widely found in the literature for predicting time-based projects (Chen et al. 2010; Attarzadeh et al. 2012: Kuenkaikaew et al. 2013).

The neural network-based estimation method uses machine learning algorithms to predict the effort required based on historical data, so the model can learn from patterns and trends in historical data and adapts to different contexts (Aggarwal 2018). Neural networks have the virtue of capturing non-linear and complex relationships between variables. In addition, several development environments for implementing the algorithms are used, such as Jupyter Notebook, RStudio, Google Colab, among others. However, the downside of this machine learning method is a reliance on historical data. In addition, some advanced models of neural networks require technical knowledge for their implementation and training. Goodfellow et al. (2016) describes a neural network as a model inspired by the human brain to learn and perform specific tasks. It consists of a set of interconnected units called neurons, organized in layers and connected to each other to process information in parallel. In (White and Hassan 2019), logistic regression models and recurrent neural networks (RNN) with GRU cells and attention mechanisms are used to predict task duration. The models of neural networks achieve an approximate accuracy of 80%. In (Lishner and Shtub 2022), a tool based on artificial neural networks (ANN) is proposed to calculate the duration of projects. This tool automatically adapted to different forecasting methods and data sets and showed high accuracy. In (Chen et al. 2010), a model based on neural networks is proposed to improve the accuracy in project duration prediction. The model stands out for obtaining a MAPE of 8.38%. Nevertheless, in (Hamada et al. 2014), deep neural network algorithms were used, where an average error rate of 38% was obtained. The model can predict the estimated value of the project time, which optimizes the scheduling process. However, the MAPE is very high and far from the metrics that are sought.

In (Neskorodieva et al. 2020) it is implemented different neural network models and techniques, such as Jordan neural networks, Elman neural networks, LSTM (Long-Range Short-Term Memory), among others, to forecast indicators in the audit task. The techniques with the lowest root mean square error were Modified Liquid State Machine (MLSM), Echo State Network (ESN) and Liquid State Machine (LSM).

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It works primarily by finding an optimal hyperplane in a high-dimensional feature space that better separates data points of different classes. SVM aims to maximize the spread between the support vectors, which are the data points closest to the decision limit (MathWorks, 2022). In (Kuenkaikaew 2013) decision trees (J48), logistic regression and support vector machines (SVM) are used to predict audit results. However, the SVM obtains the highest accuracy with 79.36% approximately.

A Decision Tree is a supervised learning algorithm used for both classification and regression problems. It is based on the hierarchical structure of a tree, where each internal node represents a feature, each branch represents a possible decision based on that attribute, and each leaf represents a result or output (IBM n.d). In (Kuenkaikaew 2013) a set of variables is used to predict the state of sales transactions. Decision trees, logistic regression and SVM are used, obtaining a precision that varies from 61.53% to 79.41%.

3. Methods

The methodology applied for this project is CRISP DM. This methodology is particularly made for data analytics related projects. The methodology is explained step-by-step below (IBM 2021):

1. **Business Understanding:** This phase of the methodology focuses on understanding the context of the business or research and the specific problem that is being attacked with the project. Additionally, certain aspects of the project are established, such as objectives, restrictions, and others.
2. **Data Understanding:** During this phase of the methodology, data collection and a preliminary analysis of the same are carried out to generate initial observations and evaluate data effectiveness to make the model.
3. **Data Preparation:** This phase of the methodology involves a series of processes and actions that serve to separate the important or relevant data from what is collected. Data is usually modified or reworked in different ways, to enrich or make the information more practical for modeling. This phase is usually very laborious compared to the others.
4. **Modeling:** This phase, as its name would imply, consists of using filtered and clean information to build a model that can explain the behavior of the data.
5. **Evaluation:** This phase consists of applying different metrics and tools to verify the effectiveness and quality of the model or models obtained in the previous phase. Based on the evaluation results, decisions are made on how to proceed with the project.
6. **Deployment:** This is the last phase of the methodology; in essence, it consists of using the model in business or research to see its contribution and effectiveness in real cases.

In case of obtaining unsatisfactory results in any of the phases, it is possible to return to the previous phases to try to complement or modify the data, hoping to obtain better results. Some phases like Modeling and Evaluation can be performed in parallel, since it is often the case that as models are generated, they need to be evaluated to improve their performance.

4. Data Collection

The initial data provided by the company consisted of a record of completed audits from early 2020 to mid-2022. The records contained various variables that were used for a descriptive analysis and initial preliminary models. Feature and Permutation Importance tests were implemented to evaluate the significance of the variables and how much they could explain the dependent variable (Actual Effort). Both tests and preliminary models showed unsatisfactory results, so it was determined that more variables needed to be added to the database to obtain better results.

To complement the initial database, two other databases were merged to it (provided by the company's auditors). While the number of variables to generate models increased, a significant portion of records were lost (from around 3000 records, more than half of them were lost) due to inconsistencies. This new merged database gave overall slightly better results, but not close enough to the objective that was established. A few more changes were applied along the way, generating more different versions of the database; these changes mainly included adding and/or removing variables and reprocessing them, adding and/or removing records. Still, the auditors that were collaborating in this project disapproved of the processed data since many records were eliminated.

Due to the previously mentioned reasons, it was determined that, to avoid losing records, the original or initial database would be reworked by the auditors. The initial database, besides not having enough variables, had many spelling mistakes and missing information. The auditors reworked the database by correcting any existing mistakes or missing values and complemented it with more variables. The newly reworked database was tested and used to generate many models, but still, even after applying several techniques (for instance, Principal Component Analysis or PCA) the results were still unsatisfactory.

Every model generated was based on predicting the Actual Effort per audit. As previously mentioned, generally, the results with this approach were non-compliant with the objectives that were established. A last technique that was applied to try to get better results was grouping the data. With the data available and at the request of the company's auditors, it was concluded that grouping the records by Sprint (period of two weeks) was the best approach to carry out. Before, each of the records represented a project or audit, but after grouping them, each record represents a Sprint. The database variables were also reworked because of the same grouping; the nature of the records changed, and the original variables were no longer suitable for these records. The new generated variables are the following:

- Number of projects or audits carried out in the Sprint (referred to as Project Count).
- Sprint (number).
- Semester (number).
- Count of projects carried out by each type of WBS Category (WBS Category is an internal categorization the company has for its audits. For each category, in total 11, there is a different column or variable).
- Count of projects carried out by each Audit Type (Audit Type is an internal categorization the company has for its audits. For each category, in total 3, there is a different column or variable).
- Count of projects carried out by each Engagement Type (Engagement Type is an internal categorization the company has for its audits. For each category, in total 2, there is a different column or variable).
- Number of hours of effort (Actual Effort) it took to perform the audits in the Sprint (new dependent variable).

The only significant downside with this new database is that, due to the grouping, the number of records has considerably decreased; from around 3000 records, it was reduced to only 48 (in other words, there were only 48 sprints registered in the database).

5. Results and Discussion

5.1 Initial Results

As previously mentioned in the Data Collection Section, many changes were applied to the database to get better results. Each change that was applied generated different versions of the same database which were all modeled using different types of models, such as: Multiple Linear Regression, Random Forest Regressor, K-Nearest Neighbors Regressor, Support Vector Regression, Gradient Boosting Regressor, Extra Tree Regressor, XGBoost Regressor, Artificial Neural Network Regressor (ANN Regressor) and even automatic modeling tools such as Auto ML and Lazy Regressor. Additionally, for each change or version of the database, Feature and Permutation Importance tests were applied to evaluate the importance or weight of each variable for modeling. It is worth mentioning that these tests also calculate a metric called model score, which is a metric that explains how much (in percentage) the variables of a database explain the behavior of the dependent variable (ranges between 0% and 100%; the higher, the better). Each of these models were also evaluated. The various metrics that were applied for evaluation included: coefficient of determination ($\boldsymbol{R^2}$), root mean square error (RMSE), mean square error (MSE), mean absolute percent error (MAPE), symmetric mean absolute percent error (SMAPE), and mean arctangent percent error (MAAPE).

There were 5 main versions of the database before the last version was obtained. From version 1 through 5 the database and models worked around predicting the Actual Effort for a single audit. All these versions, generally, showed poor results compared to the established objectives. From these 5 versions, the 5th version (referred as V5) showed the best results of them all. Firstly, the results for the Feature Importance test are shown below:

Table 1. Feature Importance Values for the V5 Database's Top 5 Variables.

Feature	Importance
Sprint_num	0.2072
Audit Type_num	0.1769
Type of Engagement_num	0.1421
Expertise_num	0.1271
WbsSubcategory_num	0.1047

According to Table 1, the most significant variable is the Audit Type, which in previous versions, always manages to position itself among the most important ones. Sprint is placed first and the Engagement Type variable (a new variable that was added in this version) is placed third. It is worth mentioning that the differences between the importance of these variables are small (between a variable and the one that precedes or proceeds it), therefore, it is important to consider all these variables when modeling (including the Expertise variable onwards; there are more variables in the test that are not shown). The model score obtained with this test is 0.8043, which is not bad, but not as high as desired. The Permutation Importance test showed similar results in the positioning of variables but differed more in importance values.

Table 2. Permutation Importance Values for the V5 Database's Top 5 Variables.

Feature	Importance
Audit Type_num	0.5599
Type of Engagement_num	0.3219
Expertise_num	0.1882
WbsCategory_num	0.1294
WbsSubcategory_num	0.1029

According to Table 2, the Audit Type and Type of Engagement variables have a significantly higher importance compared to the rest of the variables. The Sprint variable (which was the most significant in the Feature Importance test) is not even on the top 5 for this test. Regarding having these differences, it is good to see that most of the variables coincide between both tests. The model score for this test is also very close to the previous test, having a value of 0.8138.

In this version of the database (V5), several modeling techniques were used, such as XGBoost Regressor, Auto ML, Lazy Regressor, and ANN Regressor. In the V5 database, several records were found with values of the Y variable, Audit Effort, ranging from 0 to 1, which caused the MAPE to increase considerably due to its relative nature. To address this situation, the metrics SMAPE and MAAPE were used.

This V5 database has a more realistic behavior of the dependent variable with nearly 3,000 records. As the volatility of the data increased, efforts were made to clean the database, removing the top three outliers, and more advanced techniques were employed to improve the models.

5.2 Improved Results

Then, the approach of grouping the data was applied to try to get better results. As already specified in the Data Collection Section, the database was grouped by Sprints, generating the last version (version 6) of the same. Variables had to be reworked, so applying Feature and Permutation Importance tests was necessary to evaluate the effectiveness of the new variables. Several tests and models were implemented, trying different combinations of the variables. The best results were obtained with a model that used the following variables: Project Count, Sprint, Semester, ENG Managed Services, and ENG Specialized Services (the last two of these are variables related to the count of projects according to the Type of Engagement). Table 3 shows the results of the Feature Importance test.

Table 3. Feature Importance Values for the V6 Database.

Feature	Importance
Project Count	0.8513
ENG Managed Services	0.0733
ENG Specialized Services	0.0437
Sprint	0.0276
Semester	0.0038

As seen in the previous table, the most important variable for the model (and with a very considerable weight) is Project Counts. This variable alone explains around 85% of the behavior of the data. This variable is followed by the variables related to the Type of Engagement (Table 4). Although most of the variables have little weight, the combination of these is optimal, because with few variables (5 variables) the behavior of the data is effectively explained; the combination of these variables resulted in a model score of 0.9845, in other words, around 98% of the behavior of the dependent variable is explained by these variables. The results of the Permutation Importance test are very similar to those of the previous and the same can be inferred.

Table 4. Permutation Importance Values for the V6 Database.

Feature	Importance
Project Count	0.8508
ENG Managed Services	0.0505
ENG Specialized Services	0.0106
Sprint	0.0018
Semester	-0.0107

Since the results of the Permutation Importance are very similar to those of the Feature Importance test, the observations previously made are supported. The model score for this test is 0.969, which is also very close to the score of the Feature Importance test.

In this version of the database (V6), the Lazy Regressor and Auto ML modeling techniques were used with the 5 variables mentioned above. Some of the advantages of the Lazy Regressor is that it performs a quick evaluation of various regression algorithms on a data set, which simplifies the modeling phase.

According to the evaluation of the V6 database with the Lazy Regressor algorithm, the top three suggested techniques, ordered from best to worst, are ExtraTrees Regressor, followed by Huber Regressor and Orthogonal Matching Pursuit CV. The results obtained are shown in Figure 1.

Extra Trees Regressor

PARAMETER	RESULT
R^2 Train	100.00%
R^2 Test	96.00%
Difference:	3.69%
MAPE Test:	8.27%
RMSE Test:	4.90

Extra Trees Regressor Optimized

PARAMETER	RESULT
R^2 Train	98.00%
R^2 Test	93.70%
Difference:	3.86%
MAPE Test:	8.82%
RMSE Test:	5.68

Huber Regressor

PARAMETER	RESULT
R^2 Train	90.00%
R^2 Test	95.00%
Difference:	5.13%
MAPE Test:	9.02%
RMSE Test:	5.23

Huber Regressor Optimized

PARAMETER	RESULT
R^2 Train	90.0%
R^2 Test	95.60%
Difference:	5.24%
MAPE Test:	8.52%
RMSE Test:	5.15

Orthogonal Matching Pursuit CV

PARAMETER	RESULT
R^2 Train	90.00%
R^2 Test	95.00%
Difference:	5.05%
MAPE Test:	8.24%
RMSE Test:	5.023

Figure 1: Results of the models suggested by Lazy Regressor for V6 database.

Figure 1 shows multiple tables containing the name of the model implemented, the parameters used as performance metrics and the result for each parameter. The best model suggested by Lazy Regressor is the Extra Trees Regressor with its default parameters, which resulted in an R2 test of 96.0%, a MAPE test of 8.27%, and an RMSE test of 4.90 hours.

The results of the models in this version of the database are more satisfactory than the previous ones, meeting the particular objective established at the beginning of the project, in which the margin of error (MAPE) must be less than or equal to 10%. And although the model with the Octagonal Matching Pursuit CV technique has a slightly lower MAPE, a difference of 0.03%, its RMSE is higher, as well as the difference between the R2 in the training set and the R2 in the test set is 5.05%, which is 1.36% higher than the Extra Trees Regressor model.

On the other hand, the Auto ML technique was implemented by using the same 5 variables that were used with the Lazy Regressor. Auto ML is a tool whose purpose is to simplify the development of machine learning models, including the process of building, training, and model fitting.

Based on the evaluation of the V6 database with the Auto ML algorithm, the main suggested techniques, ordered from best to worst, are ExtraTrees Regressor, followed by Gaussian Process Regressor, Hist Gradient Boosting Regressor, Random Forest Regressor, and SGDRegressor. The results obtained with the parameters suggested by Auto ML are shown in Figure 2.

Extra Trees Regressor Optimized

PARAMETER	RESULT
R^2 Train	100.00%
R^2 Test	96.00%
Difference:	4.15%
MAPE Test:	8.52%
RMSE Test:	4.97

Random Forest Regressor Optimized

PARAMETER	RESULT
R^2 Train	100.00%
R^2 Test	93.80%
Difference:	6.16%
MAPE Test:	10.33%
RMSE Test:	5.72

Hist Gradient Boosting Regressor Optimized

PARAMETER	RESULT
R^2 Train	100.00%
R^2 Test	89.60%
Difference:	10.36%
MAPE Test:	11.26%
RMSE Test:	6.03

Gaussian Process Regressor Optimized

PARAMETER	RESULT
R^2 Train	98.00%
R^2 Test	91.00%
Difference:	7.21%
MAPE Test:	13.91%
RMSE Test:	6.37

Figure 2: Results of the models suggested by Auto ML for V6 database.

The best model suggested by Auto ML is the same shown by Lazy Regressor function, Extra Trees Regressor; however, the main difference is that in this case the model is run with the parameters that Auto ML suggested. This resulted in an R2 test of 96.0%, a MAPE test of 8.52% and an RMSE test of 4.97 hours.

In summary, the selected model was the Extra Trees Regressor with its default parameters (suggested by the Lazy Regressor function), which resulted in an R2 test of 96.0%, a MAAPE of 8.19%, a SMAPE of 8.43%, a MAPE of 8.27% and an RMSE test of 4.80 hours.

5.3 Validation

Once the best model was obtained and its evaluation was carried out, it was decided to carry out a validation test to check its effectiveness with completely new and unknown real data for the model. For this validation test, the data from the first 6 Sprints of the year 2023 were taken. With this data, forecasts were made with the model to obtain the (estimated) Actual Effort for each of these Sprints. Table 5 compares real effort hours and those predicted by the model, including differences and margins of error.

Table 5. Validation Test

Sprint	Effort	Predicted Effort	Difference	MAPE
2023-1-Sprint 1	196.36	155.4546	-40.9054	20.83%
2023-1-Sprint 2	196.5	167.7919	-28.7081	14.61%
2023-1-Sprint 3	334.72	290.9789	-43.7411	13.07%
2023-1-Sprint 4	163.05	129.971	-33.079	20.29%
2023-1-Sprint 5	261.3	276.0063	14.7063	5.63%
2023-1-Sprint 6	155.55	145.2431	-10.3069	6.63%
			-23.67236667	**13.51%**

As can be seen in Table 5, the model forecasts are effectively close to the actual data (Effort column). The largest differences (between real and predicted) in the test show values around 40 hours, which is a bit high; while the lowest values show values around 12 hours, which is in an acceptable range. Most of the differences show negative numbers, meaning that the model, apparently, tends to forecast fewer hours than the real values. Regarding margins of error, the highest values are around 20% (in two Sprints), while the lowest are around 6% (in two Sprints). On average, there is a difference of 23 hours less than the real value, which is equivalent to an average margin of error of 13.51%, which, although it shows a value greater than that obtained in the evaluation of the model (SMAPE and MAAPE), is excellent since the margin of error remains within the established objective (maximum of 15% margin of error). With these results, the effectiveness of the forecasts that the model is capable of obtaining is strengthened.

5.4 Deployment

The second part of the deployment consists of developing an App. The software was coded in Python, using the Streamlit function as the main framework and other libraries such as Plotly for interactive graphics, Pandas and Sklearn for data analytics. The application design was incorporated with a minimalistic and intuitive architecture and the testing was carried out to ensure the application's functionality in all scenarios. Design and functionality tests were conducted locally on different devices to verify the tool's adaptability and performance. After successful testing, the tool was deployed on an official server from the company, enabling access for any auditor with the necessary permissions. Post-deployment, detailed documentation was created for the tool's code. The aim was to provide future effectiveness with an understanding of the program and to ensure the tool's continuous usability. Figure 3 shows a screenshot of the App developed. As can be seen, the users will be able to predict the Actual Effort easily and fast. The App not only shows the forecasts and their confidence limits, but also a picture of the Audit Effort dynamics.

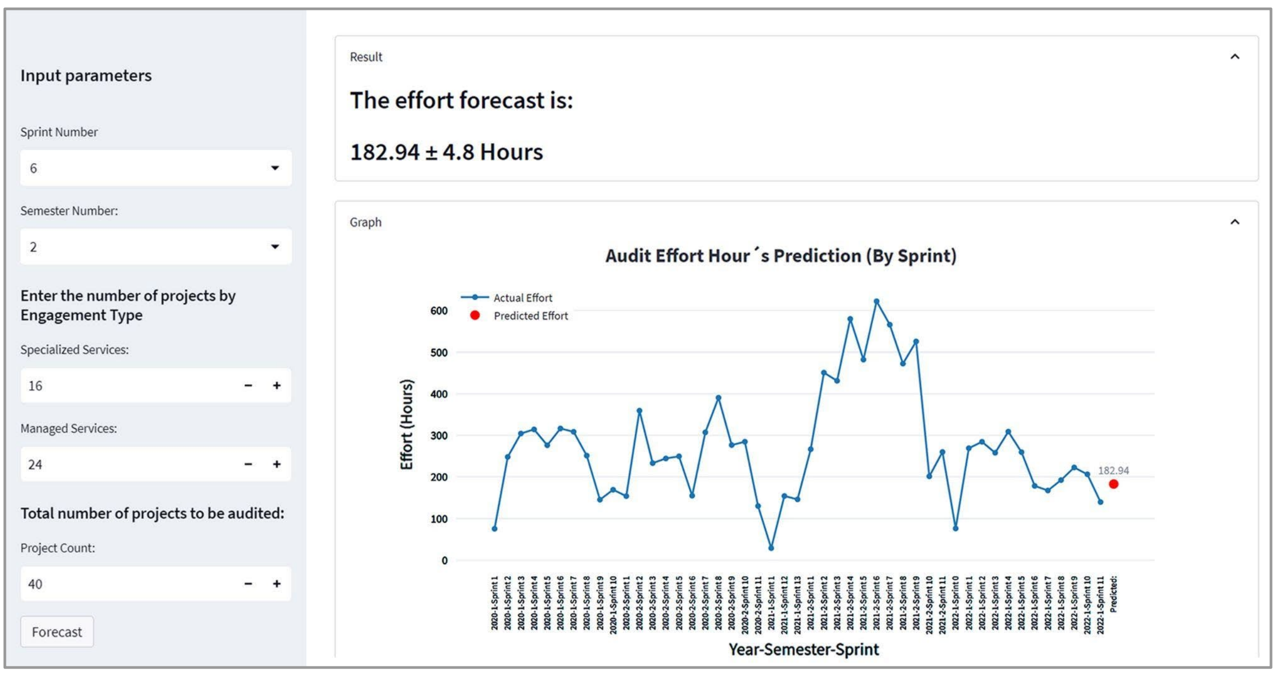

Figure 3. Audit Effort´s Planning Tool.

Figure 3 showcases the web application’s design. On the left side of the page, you can find the sidebar, which includes all input parameters for the model: our independent variables. At the end of the sidebar lies a “Forecasting” button, which when pressed reveals two boxes, one containing the model´s result, and the other containing a time series graph. The graph uses our Sprint variable as horizontal axis and our Actual Effort variable as the vertical axis.

6. Conclusion

Throughout this project, different models were run to forecast the hours of effort required for audit planning in a sprint. As part of the recapitalization of the actions carried out, there were several versions of databases, from V1 to V6.

Techniques such as Random Forest Regression, Support Vector Regression, kNN and Multiple Linear Regression were implemented for each of the databases. As the modeling stage progressed, more advanced techniques were used, such as the use of ANN, the use of the functions Lazy Regressor and Auto ML, as well as Gradient Boosting Regressor, XGBoost Regressor, Extra Tree Regressor, Linear and Kernel PCA, among others. Finally, relevant changes were applied to the V6 database, which consisted of grouping the database. These groupings were made based on the sprint number, semester, and year. This helped reduce the high volatility of the data from previous versions.

After evaluating the metrics, it was observed that the Random Forest Model (Extra Trees Regressor) performed better, with lower error values in the SMAPE, MAAPE and RMSE. The Extra Trees Regressor model with its default parameters achieved the following results: an R2 test of 96.0%, a MAAPE of 8.19%, a SMAPE of 8.4%, a MAPE of 8.27%, and an RMSE test of 4,795 hours.

In view of this and the proposed objectives, the use of this model was recommended for the automated forecasting tool that was developed using Streamlit in order to contribute to the planning of internal audits. This tool constitutes an improvement in the process because in a matter of minutes, even seconds, a forecast can be made with high accuracy. The company's actual process for forecasting the hours of effort required to carry out the audits during a sprint is done in a very "artisanal" way in Excel where the information is entered manually, and simple formulas are used; this resulted in a time consuming and tedious process. The deliverables are the best model implemented with its metrics and characteristics, the automated forecasting tool already published in the local company network, as well as the documentation on how to use the tool following the company guidelines. It is important to highlight that the results, metrics, and conclusions of this study are specific to version 6 of the database.

The only downside to this forecast tool is that the V6 database contains very little data. As an improvement opportunity, it is desirable that, in the future, the model could be retrained with more and fresh data. This will help identify any pattern or behavior that the model didn't catch with the current data.

References

Aggarwal, C. An Introduction to Neural Networks. In: Neural Networks and Deep Learning. Available https://doi.org/10.1007/978-3-319-94463-0_1, May 25, 2023.

Attarzadeh, I.. Mehranzadeh, and Barati, A., Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation, *2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 167-172, Phuket, Thailand, July 24-26, 2012.

Auren México. Auditorías en las empresas. Available: https://auren.com/mx/blog/importancia-de-las-auditorias-en-las-empresas/, Accessed on December 14, 2022.

Chen, X., Liu, L. and Li, Y., An Improved Method for Project Duration Forecasting, *2010 International Conference on E-Business and E-Government,* pp. 2644-2647, Guangzhou, China, May 07-09, 2010.

Dittenhofer, M. (2016), "Reengineering the internal auditing organization", *Managerial Auditing Journal*, vol. 16, no. 8, pp. 458-468.

Faria, P. and Miranda, E., Expert Judgment in Software Estimation During the Bid Phase of a Project -- An Exploratory Survey, *2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement,* pp. 126-131, Assisi, Italy, March 7, 2012.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning.* 1st Edition, The MIT Press, Massachusetts, 2016.

Hamada, M., Abdallah, A., Kasem, M., and Abokhalil, M., Neural Network Estimation Model to Optimize Timing and Schedule of Software Projects, *2021 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, pp. 1-7 , Nur-Sultan, Kazakhstan, April 28, 2021.

IBM. Conceptos básicos de ayuda de CRISP-DM. Available https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview, August, 17, 2021.

IBM. Decision Tree. Available https://www.ibm.com/topics/decision-trees, May 19, 2023.

Kuenkaikaew, S. Predictive audit analytics. Available https://doi.org/doi:10.7282/T3S46PZQ, January, 26, 2023

Kuenkaikaew, S. The Predictive Audit Framework. *The International Journal of Digital Accounting Research*, vol. 13, no. 19, pp. 37–71, 2013.

Lishner, I., and Shtub, A. Using an Artificial Neural Network for Improving the Prediction of Project Duration, http://dx.doi.org/10.3390/math10224189/. Accessed December 21, 2022.

MathWorks. Support Vector Machine (SVM). Available https://www.mathworks.com/discovery/support-vector-machine.html, May 25, 2023.

Neskorodieva, T., Fedorov, E., and Izonin, I. Forecast Method for Audit Data Analysis by Modified Liquid State Machine. *International Workshop on Intelligent Information Technologies and Systems of Information Security*, pp 25-35, Khmelnytskyi, Ukraine, Dec 31 2019.

Sampattikorn, S., Ussahawanitchakit, P., and Boonlua, S., (2012), Best Internal Audit Practices and Goal Achievement Sustainability: An Empirical Examination of Thai Listed Firms, *Journal of International Business and Economics*, 12(5), 40-66.

White, R. and Awadallah, A., Task Duration Estimation, *The Twelfth International Conference on Web Search and Data Mining (WSDM)*, pp. 1-9, Melbourne, Australia, February 11-15, 2019.

Biographies

Patricio Alejandro Zamora Navarro is a current student of Business Management Engineering at Universidad de Monterrey. His current passion and development of skills and knowledge involve topics related to predictive analysis, programming languages, web development and machine learning. He is currently working as an intern at an information technology services company, where he is getting professional experience and development with programming skills; he is also getting programming knowledge through the online learning platform: DataCamp.

Rafael Alejandro Ruiz Miller is an 8th semester student of Business Management Engineering at Universidad de Monterrey. Currently he is working as an analyst intern at a global information technology services company, where he is developing skills such as: data engineering, data analysis, web applications development, programming with Python, R, and more. He continues to expand his knowledge through the learning platform of DataCamp. In addition to his academic achievements, he has shown entrepreneurial interest by co-founding and serving as the director of a digital marketing company for two years. Presently, he is also pursuing a diploma in business ethics to further enhance his understanding of responsible corporate practices.

Roberto Serna Zuazua is a Business Administration Engineering student at the Universidad de Monterrey. He currently works as a data analyst in a global information technology services company, where he develops knowledge related to predictive and descriptive analysis, programming languages such as Python, R and ACL, web development with Sharepoint and Javascript, among others. In addition, he graduated with honorable mention of excellence from high school, as well as completing the International Baccalaureate Diploma Programme. Roberto has been a member of the Líderes Plus group from the Universidad de Monterrey with the purpose of achieving excellence through leadership. On the other hand, he obtained second place in the Cisco-UDEM Hackathon with the *Twinky* project focused on Smart Cities and pollution. And he served as Secretary in the Student Chapter of the American Society for Quality (ASQ) at the Universidad de Monterrey.

José D. Morcillo received the Electronic Engineer degree, and the M.Sc. degree in industrial automation from the National University of Colombia, Manizales, in 2010 and 2012, respectively. He got his Ph.D degree in Computer Science from National University of Colombia, Medellín, in 2018. He also worked as a postdoctoral researcher at the National University of Colombia, Manizales, in 2020. In 2010, he joined the Perception and Intelligent Control Group, and in 2014, joined the Systems and Informatics Group, working in different I + D projects. Currently, he is working as a full-time professor at University of Monterrey, Mexico, in the Engineering and Technologies Department. His research interests include modeling, simulation and control of power electronics and electricity markets, applying the theory of dynamical systems, system dynamics and data analytics/machine learning.