INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Application of Artificial Intelligence to Optimize Project Management within the PMO area

JOÃO FERNANDO FERREIRA VIEIRA
Setembro de 2024

P.PORTO

Application of Artificial Intelligence to Optimize Project Management within the PMO area

João Fernando Ferreira Vieira

Student number: 1150575

Dissertation to attain the master's degree in Artificial Intelligence Engineering

Advisor: Ana Maria Neves Almeida Baptista Figueiredo

Jury:

President:

Isabel Cecília Correia da Silva Praça Gomes Pereira, Coordinator Professor, Instituto Superior de Engenharia do Porto from Instituto Politécnico do Porto

Members:

António Constantino Lopes Martins, Adjunct Professor, Instituto Superior de Engenharia do Porto from Instituto Politécnico do Porto
Ana Maria Neves Almeida Baptista Figueiredo, Coordinator Professor, Instituto Superior de Engenharia do Porto from Instituto Politécnico do Porto

Porto, September 2024

Abstract

The Project Management Office (PMO) functions as an organizational entity designed to standardize project procedures and leverage efficiencies through project repetition. Beyond standardization, PMOs promote learning from past projects, enabling the adoption of best practices to optimize project delivery in terms of schedule, budget, and quality throughout the project lifecycle.

This study proposes an innovative Machine Learning (ML)-based tool that monitors ongoing projects to predict the likelihood of missing deadlines and estimates the percentage of potential delays. Additionally, the tool recommends the most suitable team members for new projects based on the project's area and category. It also includes a web-based alert system that notifies project managers when a project is at risk of failing to meet its deadline.

To achieve these goals, various Machine Learning techniques and methodologies were employed. Historical project data was collected and analyzed to develop predictive models capable of forecasting potential delays. Supervised learning models were trained on this data to classify projects at risk of missing deadlines and to estimate the delay percentage. The recommendation system for team member assignments was built using data-driven algorithms that consider the expertise and past performance of team members in specific project areas.

Furthermore, feature engineering techniques were applied to enhance the dataset, ensuring that the models could make accurate predictions. Hyperparameter tuning methods such as Grid Search and Random Search were used to optimize the models' performance. A web application was developed to serve as an interface for project managers, providing real-time alerts on projects at risk and displaying visual indicators for easy monitoring.

Finally, this study concludes that the implementation of such a tool in a PMO represents a significant innovation. It is expected to improve the efficiency and effectiveness of project management by enhancing decision-making processes and reducing the likelihood of project delays.

Keywords: Artificial Intelligence, Project Management, Project Management Office

Resumo

O Project Management Office (PMO) funciona como uma entidade organizacional concebida para padronizar procedimentos relativos a projeto e aproveitar as eficiências através da repetição de projetos. Para além da padronização, os PMOs promovem a aprendizagem através de projetos anteriores, permitindo a adoção de melhores práticas para otimizar a entrega de projetos em termos de cronograma, orçamento e qualidade ao longo do ciclo de vida do projeto. Este estudo propõe uma ferramenta inovadora baseada em Machine Learning que monitoriza projetos em curso para prever a probabilidade de incumprimento de prazos e estima a percentagem de atraso desses projetos. Adicionalmente, a ferramenta recomenda os elementos de equipa mais adequados para novos projetos com base na área e categoria do projeto.

Para alcançar estes objetivos, foram empregues várias técnicas e metodologias de Machine Learning. Dados históricos de projetos foram recolhidos e analisados para desenvolver modelos preditivos capazes de prever potenciais atrasos. Modelos de aprendizagem supervisionada foram treinados com estes dados para classificar os projetos em risco de incumprimento dos prazos e para estimar a percentagem de atraso. O sistema de recomendação para a atribuição de membros da equipa foi desenvolvido com algoritmos baseados em dados que têm em consideração a experiência e o desempenho passado dos membros em áreas específicas de projeto.

Além disso, foram aplicadas técnicas de engenharia de características para melhorar o conjunto de dados, garantindo que os modelos possam fazer previsões precisas. Métodos de afinação de hiper parâmetros, como o Grid Search e o Random Search, foram utilizados para otimizar o desempenho dos modelos. Uma aplicação web foi desenvolvida para servir de interface para os gestores de projeto, fornecendo alertas em tempo real sobre projetos em risco e exibindo indicadores visuais para facilitar o acompanhamento.

Por fim, este estudo conclui que a implementação de uma ferramenta deste tipo num PMO representa uma inovação significativa. Espera-se que melhore a eficiência e eficácia da gestão de projetos, potenciando os processos de tomada de decisão e reduzindo a probabilidade de atrasos nos projetos.

Palavras-chave: Inteligência Artificial, Gestão de Projetos, Escritório de Gestão de Projetos

Index

List of Figures

List of Tables

Acronyms

List of Acronyms

PMO	Project Management Office
PMI	Project Management Institute
AI	Artificial Intelligence
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
URL	Uniform Resource Locator
KPI	Key Performance Indicator
NPD	New Product Development
QFD	Quality Function Deployment
MCDM	Multi-Criteria Decision-Making
FGI	Focus Group Interviews
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
LSVM	Lagrangian Support Vector Machine
ANN	Artificial Neural Networks
ML	Machine Learning
PCA	Principal Component Analysis
TOPSIS-F	Fuzzy TOPSIS
OPA	Ordinal Priority Approach
FIIITLSDMU	Flexible, Interactive, Intelligent and Integrative Technique for Large Scale Decision Making under Uncertainty
PNN	Probabilistic Neural Networks
MLFN	Multi-Layer Feed Forward Networks
CEO	Chief Executive Officer

PPM	Project Portfolio Management
IT	Information Technology
AWS	Amazon Web Services
IDE	Integrated Development Environment
REST	Representational State Transfer
API	Application Programming Interface
MAE	Mean Absolute Error
GDPR	General Data Protection Regulation
RNN	Recurrent Neural Network
BI-LSTM	Bidirectional Long Short-Term Memory Network
SMOTE	Synthetic Minority Oversampling Technique
XGBoost	eXtreme Gradient Boosting

1 Introduction

This chapter provides a contextualization of the subject and areas studied in this dissertation, also defining the problem statement, defining an investigation question, goals, and contributions, ending with the explanation of how this dissertation will be structured.

1.1 Contextualization

The Project Management Office (PMO) serves as an organizational entity with the primary goal of standardizing project procedures and capitalizing on efficiencies gained through project repetition. Its role extends beyond mere standardization. PMOs facilitate learning from project experiences, fostering the establishment of best practices that, in turn, optimize the efficiency of project delivery (Philbin, 2016). This optimization is achieved by fostering the development of shared approaches, systems, and methodologies, resulting in improved performance across schedule, budget, and quality parameters throughout the entire project lifecycle. PMOs prove especially valuable in navigating the complexities of technology and engineering projects characterized by technical uncertainties and challenges (Philbin, 2016).

While initially rooted in the IT sector, PMOs have proliferated into diverse industries such as telecommunications, aerospace, and construction. Their significance lies in their ability to align projects with corporate strategy. Unfortunately, data revealed by a Project Management Institute (PMI) survey shown that less than half of the organizations (42%) reported a high level of alignment between projects and organizational strategy (Singh, 2017).

Every year, around $48 trillion is invested in projects, but only 35% are considered successful, leading to significant wasted resources. Outdated technologies, such as spreadsheets and slides, contribute to low project success rates. While project portfolio management applications have seen improvement, planning, team collaboration capabilities, and features provided by Artificial Intelligence (AI) are still behind (Nieto-Rodriguez & Vargas, 2023).

The modernization of project management is advocated, emphasizing the need for advancements in technology. Applying AI and other innovations to project management could potentially improve project's success rates by 25%, translating to trillions of dollars in value (Nieto-Rodriguez & Vargas, 2023).

By combining AI with project management, it would not only be possible to reduce the waste of resources, but also to increase the alignment between the developed projects and the organizational strategy.

1.2 Problem Statement

The usual issue with PMOs is that it lacks updated technologies that can aid in project management, which results in low project success rates (Nieto-Rodriguez & Vargas, 2023). Also, due to the number of repetitive tasks that they have to deal with, the project managers lack time to focus on more important tasks like strategic decision-making or stakeholder engagement (Savio & Ali, 2023).

The value that AI in PMO has to the organizations is not a novelty (Auth et al., 2019; Dam et al., 2019), but not without some difficulties along the way (Savio & Ali, 2023). There are some success cases shown below, but the issue for the organizations are the large amount of time and money they must invest to have some return in the area.

Also, the small number of studies published in applying AI to PMOs can limit the importance of the field and belittle the impact that this technology can have in organizations. It is important that this field of study is brought to the spotlight, so researchers dig deeper in it (Marnewick & Marnewick, 2022).

1.3 Objectives

The main goal of this dissertation is to develop AI models and/or tools that can aid the PMO in improving the planning, execution, and monitoring of projects.

It is also expected that organizations can understand and raise awareness about the value that applying AI to PMOs can bring to their business.

The investigation question this dissertation is aiming to respond to is – Will the application of AI enhance the results of the PMO?

To support the goals of this dissertation, the following secondary goals where defined:

- Understand how the planning, execution, and monitoring of projects is done nowadays.
- Find tasks that can be automated and/or integrated with AI to improve the efficiency of the PMO.
- Query the elements of the PMO to discuss what areas they feel that need optimization.

To achieve the previously referred goals, this dissertation's plans are to find areas that can be perfected with AI. This can be done either by analyzing the PMO processes or by meeting with the PMO team. This step is followed by choosing two use cases where AI can be used, and then followed by the data preparation. After this, the plan is to build, train and compare different machine learning and/or deep learning algorithms to fulfill the previously chosen use cases. Finally, the best models are chosen and integrated with a project management tool, followed by the impact and results analysis.

1.4 Contributions

The main contribution of this work results from the development of models and/or tools that are applied to a real-world PMO, hopefully proving the power and potential value of applying AI to PMOs.

The implementation of AI in project management certainly has an impact for the roles and skillsets of project managers. This shift from old project management practices to new ones could free the project managers from mundane administrative tasks and from repetitive tasks, allowing them to focus on other tasks like strategic decision-making or stakeholder engagement (Savio & Ali, 2023).

Adding to the gain of allowing project manager to focus on other tasks, there are already successful AI implementations in project management that has revolutionized traditional practices and enhancing project outcomes. A success case (Dam et al., 2019) pointed out that AI was able to accurately forecast project timelines, resource requirements, and potential risks. Another success case (Auth et al., 2019) stated that the implementation of AI helps with real-time project monitoring by allowing project managers to track the project status constantly and instantly make data-driven decisions. These cases show how AI can be useful in the PMO, making the project manager's life easier and increasing the project's success rate.

It is also important to continuously check the user's perspective about the application of AI in the PMO, querying about the usability and utility of what was developed, hoping that the continuous feedback from them will allow to improve the models and/or tools developed to meet the organization's needs.

Finally, it is expected that this study will allow the research community to further expand their horizons about the importance and utility of using AI in PMOs, hopefully sparking their interest to invest in this field of research.

1.5 Document Structure

This document is structured into five chapters: Introduction, State-of-the-Art, Developed Solution, Experimentation & Evaluation, and Conclusion.

The present chapter, Chapter 1, serves as an introduction to the dissertation, providing a contextualization of the subject and areas studied, followed by a definition of the problem, then defining the goals of the dissertation, followed by the contributions to the field, finally ending explaining the document structure.

In the second chapter, the State-of-the-Art in the field of Applying AI in PMOs is presented, using a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) statement adaptation. This chapter starts with explaining the research questions, followed by the search query, data sources, inclusion and exclusion criteria, data extraction, explaining what a PMO is, answering the research questions, and ending with a summary of the research results.

In the third chapter it is presented the solution. It starts by explaining the differences between what was initially planned, that can be consulted in Apendix A

Planned Solution, and what was actually implemented in this dissertation's work, followed by a detailed analysis of the data collection, then approaching what was done to understand that data and extract information from it, followed by listing and explaining the methods and tools used to achieve what was planned, then explaining what data protection, security, and ethics concerns were taken into account when developing the project, finally ending with a project architecture diagram and a discussion about what was described in the chapter.

In the fourth chapter, everything that was described in the fourth chapter is combined into trained models that fulfill the PMO's needs for the project, and it is also detailed how each model performed in each use case. The fourth chapter process is comprised of two parts: experimentation & evaluation before the deployment of the models, and experimentation & evaluation after the deployment of the models. Experimentation & evaluation before the deployment details the performance metrics used to evaluate the models, how the models performed, what was done to perfect the model, and what was the selected model. Experimentation & evaluation after the deployment details what was done to evaluate the deployed models, and how to assess the impact that said models will have in the PMO and in the organization.

Finally, in the fifth chapter, the conclusion is presented, starting by summarizing all the chapters, followed by answering the investigation question, giving a critical appraisal, finalizing with stating the limitations had and giving suggestions for future research and development.

2 State-of-the-Art

This chapter starts with explaining the systematic review, defining the PRISMA statement, followed by defining the research questions, explaining the search query, stating the data sources used, defining the inclusion and exclusion criteria, showing the search results, then exploring the existing tools, models, and algorithms, followed by the identification of gaps and challenges in AI for PMO, explaining what is a PMO along the way, ending with a summary of the research results.

2.1 Systematic Review

This section will define the research questions, the research query, the data sources, the search restrictions, the inclusion criteria, and the exclusion criteria. It presents a state-of-the-art in the field of Applying AI in PMO, using a PRISMA statement adaptation for a Systematic Review.

The PRISMA statement is a reporting guideline created with the objective of addressing poor reporting of systematic reviews. This statement comprises a checklist of twenty-seven items along with examples of reporting so the quality of said systematic reviews is improved (Page et al., 2021).

This systematic review's goal is to support the objectives of the dissertation by showing that the implementation of AI in PMOs in not only needed, but also interesting and desired.

Also, this section will explain the most common techniques and areas of the PMO in which AI is used, which will serve as a reference for this project's implementation.

2.1.1 Research Questions

To achieve the previously defined dissertation's main goal, which is to develop AI models and/or tools that can aid the PMO in improving the planning, execution, and monitoring of projects, it is important to understand what the current state of the PMO is and what can be done to enhance it to the fullest.

To do this, three research questions where defined that will be answered by the end of the systematic review.

The goal for RQ1 is to get to know what AI is already being applied and used in the PMOs, because it is expected that wiser choices are made when trying to innovate having a deeper knowledge in this field. The answer to this research question will also show what areas of the PMO are optimized with AI and what areas need optimization with AI.

For the RQ2, the goal is to, out of the AI models and tools that are already applied to the PMOs, get to know what the best solutions and techniques are when developing this kind of software. The answer to this research question will prove what did work and what did not work when developing AI focused on digitalizing the PMOs.

Finally, for RQ3 the goal is to broaden horizons on what areas of the PMO most need the application of AI to further enhance its performance and efficiency. The answer to this research question will hopefully define what area of the PMO makes more sense to develop AI tools for.

Table 1 presents the research questions.

Table 1 – Research Questions

ID	Question
RQ1	What AI models and/or tools exist for the use of AI in PMOs?
RQ2	What are the most used solutions, models, and techniques used for implementing AI in PMOs?
RQ3	What are the existing gaps in the implementation of AI in PMOs?

2.1.2 Search Query

A search query can be created by combining various search terms with Boolean operators, which is used to discover the studies used in this systematic review. For the development of this search query, a combination of the keywords was made along with the Boolean operators. Table 2 presents the research query used.

Table 2 – Search Query

Keyword	Search Term
Artificial Intelligence	("Artificial Intelligence" OR "AI")
	AND
Project Management	("Project Management" OR "Project Planning" OR "Resource Allocation" OR "Project Success Rates ")
	AND
Project Management Office	("Project Management Office" OR "PMO")

2.1.3 Data Sources

The used academic databases used in this dissertation are presented in Table 3.

Table 3 – Data Sources

Academic Database	Uniform Resource Locator (URL)
IEEE Xplore	https://www.sciencedirect.com/
Science Direct	https://www.sciencedirect.com/
b-on	https://www.b-on.pt/

2.1.4 Quality Asessment

The following section describes what was done regarding the quality assessment.

2.1.4.1 Inclusion Criteria

When applying the search query to the academic databases, some inclusion criteria where defined so the given results could have better quality.

For IC1, it was defined that the document should be a journal article or a conference article, because magazines or book chapters are not scientifically validated.

For IC2, a filter was made so the resulting documents are only written in the Computer Science area, because if documents from other areas were retrieved it would be probable that the focus of said documents would not be AI.

Finally, for IC3, the selected documents should have the full text available, because if they did not it would not be usable.

These inclusion criteria can be seen in Table 4.

Table 4 – Inclusion Criteria

ID	Inclusion Criteria
IC1	The document must be a journal article or a conference article
IC2	The document must be in the Computer Science area
IC3	The document's full text must be available

2.1.4.2 Exclusion Criteria

After applying the inclusion criteria mentioned above along with the search query to the academic databases, a total of eighty-two papers were collected. For these eighty-two papers, some exclusion criteria were defined also aiming for greater quality in the results.

For EC1, the documents should not have been published more than six years ago, because in this field of work with the rapidly changing technologies and methodologies, it is expected that the document's content is already not up to date.

For EC2, it was defined that if the abstract is not relevant, it is not worth wasting time reading the rest of the document.

Finally, for EC3, if the document is not written in English, it should not be included.

Table 5 shows the exclusion criteria.

Table 5 – Exclusion Criteria

ID	Exclusion Criteria
EC1	More than 6 years old (published before 2018)
EC2	Abstract is not relevant
EC3	The document is not written in English

If in the end, the paper is not excluded, the paper should be read, a summary is to be made, and the references of the paper are to be analyzed. If the paper's references are interesting, include them in the systematic review.

2.1.5 Data Extraction

After using the search query and applying the inclusion criteria, a total of eighty-two documents were selected for the systematic review.

Table 6 describes the number of documents retrieved from each data source.

Table 6 – Documents selected from each data source

Data Source	Number of documents selected
b-on	Fifteen documents
IEEE Xplore	Three documents
Science Direct	Sixty-four documents
Total	Eighty-two documents

Once the documents were extracted, the exclusion criteria were applied. Table 7 presents the number of records excluded and the reason for which they were excluded.

Table 7 – Number of documents excluded and the respective reason

Reason	Number of documents excluded
More than 6 years old (published before 2018)	Thirty-four documents
Abstract not relevant	Thirty-eight documents
Not written in English	Zero documents
Total	Seventy-two documents

With a total of seventy-two documents excluded, all that was left was a total of ten documents that are useful for this systematic review. Figure 1 is a PRISMA flow diagram that describes the different steps of the search methodology described above.

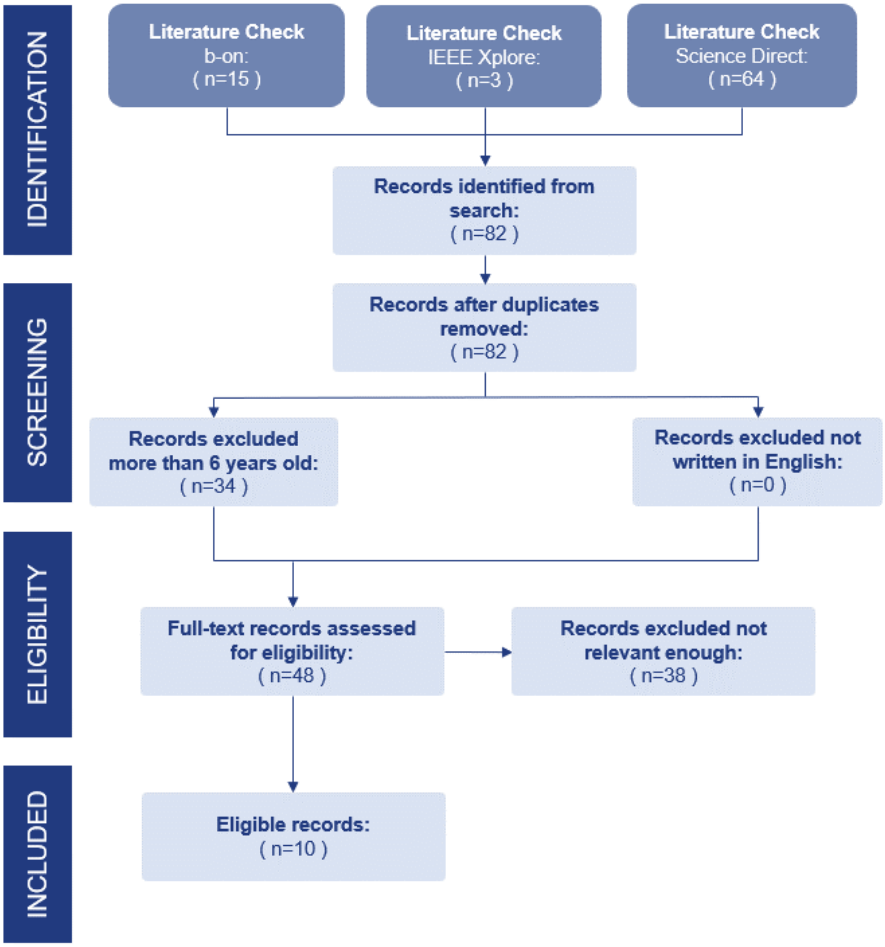

Figure 1 - PRISMA Flow Diagram

2.1.6 Research Results

The following section describes the results attained in the research done by applying the methods detailed in the previous section and shows how these results helped answering the previously defined research questions. Each research question is presented as a subsection, and the articles that helped answer it are identified within the text, along with the explanation of why those articles were useful in answering the question.

Some context is also given in this section so the reader can easily understand what is being shown, one example being what is a PMO.

2.1.6.1 What is a PMO?

Before diving into answering the research questions, it is important that the reader has a notion of what a PMO is and what are the responsibilities of it. The PMO, who can also be called a

center of excellence or center of expertise (Dai & Wells, 2004), is “an organizational body or entity who are assigned various responsibilities related to the centralized and coordinated management of those projects under its domain. The responsibilities of the PMO can range from providing project management support functions, to actually being responsible for the direct management of a project” (*A Guide to the Project Management Body of Knowledge*, 2008). The PMO has a dynamic organizational structure because it is part of and interacts with the organization to solve the organization’s problems (Fernandes et al., 2021).

According to the last statement, the PMO’s structure and responsibilities will vary depending on the organization needs, so before developing any tools one should first identify what the business’s needs are. For this dissertation purpose, the PMO for which the AI algorithms will be implemented focuses on project planning, on supporting project managers by resolving issues which would stop the development of the projects being developed, and on monitoring project development to assure that the project’s deadline is fulfilled as well as the projects being successfully delivered.

2.1.6.2 RQ1: What AI models and/or tools exist for the use of AI in PMOs?

Of the ten articles retrieved, eight focus on developing AI tools to use in the PMO, most of them focusing on project selection and project prioritization.

Article (Beseiso & Kumar, 2021) describes a developed tool for selecting projects with the highest priority from a project portfolio. This tool focuses on considering priority criteria aligned with the organization’s objectives, aiming for correctly selecting projects even in cases where there is incomplete information. This approach addresses challenges such as experts’ uncertainty in project selection, prioritization criteria before the selection process, and evaluation of interdependent projects. The validation of this solution is demonstrated in an example with three experts, three prioritization criteria based on organizational objectives, and four selection criteria.

Paper (Jafarzadeh et al., 2022) details a tool also developed for prioritizing projects, but this time addressing challenges such as uncertainty, decision maker reliability, systematic identification of criteria, and robustness to changes in opinion.

The developed benchmarking system described in (Bilal & Oyedele, 2020) was created for a construction organization with the purpose of evaluating project proposals, aiming to allow

contractors to evaluate the profitability performance of said proposals more effectively. This system was developed by using seventeen Key Performance Indicators (KPIs) extracted from email data, along with eight project attributes to ensure contextualized benchmarking.

Regarding the management of an investment project portfolio management, the described tool in (Wach & Chomiak-Orsa, 2022) make use of predictive analysis to support the processes of decision-making when choosing what projects to invest in from a project portfolio of a mining organization. Some other algorithms were developed to determine what the most effective algorithms were and what the key project attributes that allowed to predict budget deviations were.

In (Bolaños & Barbalho, 2021), the importance of reducing cycle-times in New Product Development (NPD) to stand out competitively is addressed. The proposed tool considers the potential product complexity and the prototyping lead-times when predicting time prediction in NPD projects. The main goal was to aid managers in their NPD development plans, and to guide them into having more assertive launches for their new product.

Similarly to (Beseiso & Kumar, 2021), the tool described in (Mahmoudi et al., 2021) was developed to aid in multiple criteria decision-making even with missing values. This tool was developed for a manufacturing organization which produces refinery equipment, because the organization had limited resources and needed a priority system who could identify critical projects based on the organizational strategy. It considers project duration, the number of items each project has, the client score, the project delay prevention, physical weight, revision number, code number, and the level of difficulty for each project. Despite of the fact that this tool was developed for this specific purpose, it is said that the approach is flexible, interactive, intelligent, and integrative, so it can be applied to any multiple criteria decision-making scenario, like to aid with the selection of the projects with the highest priority in a project portfolio.

The tool described in article (Mossalam & Arafa, 2018) also has the goal to select projects from a project portfolio, but this time is to select them to be monitored, instead of selected to be developed. The study mentions that this type of selection is usually manual, so it presents the intelligent model to make the selection. This intelligent model was developed counting with a database of more than three hundred project, being that from those three-hundred projects, forty-eight were already selected to be in the top management monitoring dashboards. This

tool was evaluated by examining the deviation between the model results and the already selected projects.

The area that the tool described in paper (Liu, 2019) business continuity, more specifically the decision-making behind business continuity. A new Chinese tool developed for implementation in the PMO has the main goal of analyzing the consumer behavior, so the consumer can be assigned an accurate classification. This classification will be of use when the PMO decides to create better marketing strategies and processes aimed at specific combinations of consumers.

2.1.6.3 RQ2: What are the most used solutions, models, and techniques used for implementing AI in PMOs?

Of the ten articles retrieved, eight provide detailed information of what models, techniques, and solutions were used to develop Ai tools to implement in the PMO.

In article (Beseiso & Kumar, 2021), it is proposed and approach that involves the integration of fuzzy logic, Quality Function Deployment (QFD) and a meta-heuristic genetic algorithm. The QFD approach is "considered as one of the most commonly used strategies for managing quality" (Beseiso & Kumar, 2021). The fuzzy logic was introduced in this tool to manage the uncertainty that lies on taking decisions in real-life problems, such as the selection of a project from a project portfolio, and the genetic algorithm to solve optimization problems. Combined, a hybrid approach composed of a hybrid fuzzy-QFD approach along with the use of a meta-heuristic genetic algorithm were used to result in the final tool.

QFD is also used in the tool described in paper (Jafarzadeh et al., 2022), but this time combined with the Z-numbers theory, along with four Multi-Criteria Decision-Making (MCDM) methods, ensemble ranking aggregation, and sensitivity analysis. The Z-numbers theory is an extension to the fuzzy logic, so it is also a theory that aids in taking decisions in real-life problems, and the MCDM is used choose between a set of options based on multiple competing selection criteria. The final tool used this hybrid method, composed of the Z-numbers theory and MCDM along with ensemble ranking and sensitivity analysis.

In paper (Bilal & Oyedele, 2020) text mining is used to analyze emails with project proposals with the objective of determining the KPIs for each project. After these KPIs where identified, they were validated by Focus Group Interviews (FGI) with a total of twenty participants. After this, the tool was integrated with some databases like Oracle Financials, Primavera, Candy, and

other large bodies of unstructured documents with the objective of identifying more KPIs from those data sources. With all the data gathered, a deep ensemble algorithm was developed, composed of a two-staged decomposition and integration approach that, on the first stage, "yields several attribute-specific benchmarks" (Bilal & Oyedele, 2020) and, on the second stage, shrinks them "into a final project-sensitive benchmark to support the tender assessment process" (Bilal & Oyedele, 2020).

Similarly to what was seen on the previous article, the article's (Wach & Chomiak-Orsa, 2022) main goal is also to identify KPIs, but this time to support managing and decision-making in a project portfolio. In this article, predictive analysis is applied to the project portfolio, ranging from the use of logistic regression, Bayesian networks, K-Nearest Neighbor (KNN) algorithm, Support Vector Machine (SVM) algorithm, Lagrangian Support Vector Machine (LSVM) algorithm, Random Trees, Random Forest, and Artificial Neural Networks (ANN).

Regarding the successful and timely delivery of projects, the tool described in (Bolaños & Barbalho, 2021) aims for reducing the cycle-times for NPD. To achieve this goal, the tool is composed of Machine Learning (ML) regression models who considers only "the linear shape of the relation between product complexity and time performance in which a linear regression model could reach good results" (Bolaños & Barbalho, 2021), for a matter of simplicity.

(Mahmoudi et al., 2021) also describes a tool for multiple criteria decision-making which can be applied to project selection from a project portfolio, as mentioned in the previous section, presenting a solution with a different approach from all the articles presented. This tool uses Principal Component Analysis (PCA), K-means algorithm, Fuzzy TOPSIS (TOPSIS-F), and Ordinal Priority Approach (OPA). This approach was called the "Flexible, Interactive, Intelligent and Integrative Technique for Large Scale Decision Making under Uncertainty (FIIITLSDMU)" (Mahmoudi et al., 2021).

Regarding the decision-making in the initial selection of projects to monitor in government organization, the paper (Mossalam & Arafa, 2018) proposes a solution that uses ANNs to achieve this goal. The selected ANN was the Palisade Corporation Neural Add-on because it is a proven well known user friendly leading tool in the domain (Mossalam & Arafa, 2018). This ANN supports two distinct types of networks: Probabilistic Neural Networks (PNN), Multi-Layer Feed Forward Networks (MLFN). Also, this tool has the "Best Fit Method" which assesses a variety of neural network configurations to ensure the best results. However, it is mentioned in the paper

that the Best Fit Method, although ensuring the best results and being more dependable than applying the PNN or the MLFN, it takes longer to give the results, so it is more convenient to not use it when prompt results are needed.

(Liu, 2019) presents a Fuzzy gray situation decision-making algorithm, used for text mining and decision-making for targeting and classifying costumers to allow PMOs to create better marketing strategies and processes aimed at specific combinations of consumers. The study is based on the MCDM to build a "data-mining model based on the characteristics of fuzzy textual information" (Liu, 2019).

2.1.6.4 RQ3: What are the existing gaps in the implementation of AI in PMOs?

Of the ten articles retrieved, two provide information of what gaps exist in the implementation of Ai tools to implement in the PMO.

In article (Riesener et al., 2023) a literature analysis on what the success factors and their corresponding scientific approaches in a multi-project management is made. The goal for this research is to allow companies to remain competitive by using their resources efficiently and effectively with the implementation of appropriate method, processes, and forms of organization. The paper starts by examining case studies and field reports with the companies and the experts from the companies, following with surveys and studies in the industry conducted by associations and consulting firms. Then, with this information in hand, the "critical challenges and success factors for successful multi-project management are derived" (Riesener et al., 2023). Finally, a literature review is conducted to get to know how the identified success factors are addressed in the literature, and what approaches are suggested for this issue. As a result of these actions, the key challenges and success factors presented below were found. Key challenges and success factors: Competency and skill management, identification of bottleneck resources, limit work-in-progress projects, consider economic aspects due to project delays, use of cycles of cadence, flexibility, decision support by digital tools and AI, realistic project planning, and global controlling of multi-project environment.

A bibliometric analysis is made in paper (Marnewick & Marnewick, 2022) on how is the state of project management digitalization. The goal for this research is to perceive how the digitalization is impacting the project management discipline, to highlight the importance of digitalizing the project management processes to be able to enhance and optimize the latter, and to introduce a "Project Management Digitalization Research Agenda Cube" (Marnewick &

Marnewick, 2022) that can be used to draw a path that practitioners and academia can follow to facilitate the digitalization of project management. To achieve these goals, the paper first focuses on the literature review on digitalization of project management, followed by the process of the bibliometric analysis and its results, finalizing with the found results and suggestions of work to be done. The results show that there is a high interest in digitalization within the project management discipline, but it also shows that project managers lack knowledge about the technology trends, and about how technology can be used as a tool as well as used to optimize project management. Also, it is also pointed that the digitalization impacts business models and processes, but the focus found on the literature is not on the digitalization of project management, which shows a clear gap. Another gap mentioned by the paper is that the only permanent entity is the PMO, however the role of the PMO in adopting and increasing digitalization was not mentioned in any article that was part of the paper's literature research. The paper states that the focus of research in the area should be on the role of the PMO in the digitalization project management as well as the digitalization of the PMO itself. The role of the PMO must be researched in the context of the technology to be used, a change in culture (this is one of the most difficult problems to solve as it is necessary for people to adapt and want change) and in the theory of change, as there is a lack of theories to boost understanding of the digitalization of project management. It is also necessary to research the technical skills needed to make digitalization happen and the nature of project management. It is also stated by this paper that the PMO must become the owner of digitalization in project management. Finally, it is concluded by the paper that current literature focuses on the driving points of the digitalization of project management but there is little or no literature on the objects and impacts of digitalization in the PMO.

2.2 Summary

The most common mentioned area for the development of models and tools in the PMO found in the present systematic review is the selection of projects from a project portfolio, with six occurrences, ranging from prioritizing project to be developed first to selecting projects to be monitored, basing the selection process on project's KPIs, the organization's KPIs or the organization's objectives, the most profitable project, or simply the project depending on the client's score.

The least mentioned areas were the profiling of costumers so the PMO could create processes and marketing targeted to certain groups of consumers, with only one occurrence, and the enhancement of project planning to be able to provide more accurate deadlines for projects in NPDs, also with one occurrence.

Regarding the most used models or techniques, the use of the Fuzzy logic or derivations of the fuzzy logic, like the Z-numbers theory, is what occurs the most on this systematic review, followed by the QFD, MCDM, and the ANNs.

Finally, there are a lot of existing gaps in the implementation of AI in PMOs shown by this literature review, giving the fact that most of the papers reviewed aimed for the digitalization of project management itself, not the digitalization of the PMO. In the last article, (Marnewick & Marnewick, 2022), discussed in the previous section, it is concluded that there is little or few articles found in the bibliometric analysis that focus on the subject.

According to this systematic review's results, the PMO area is lacking development of tools to enhance and optimize it. Having into account that the PMO that this dissertation's work will be based on focuses on project planning, supporting project managers by resolving issues which would stop the development of the projects being developed, and monitoring project development to assure that the project's deadline is fulfilled as well as the projects being successfully delivered, it is justified to develop AI tools and models to support this PMO.

The aim of this dissertation is to develop a tool that would be helpful to the PMO in question, which is the PMO from the organization this dissertation is being developed for, and that would be a tool that can monitor all the work-in-progress projects with the goal of predicting deadline deviation, presenting an estimated cost to the organization in case that deviation occurs, and a tool that suggests the best team members to be on a to be started project based on the field of work and the category of said project. To do this, it would be ideal to use the approach used in (Bilal & Oyedele, 2020), using deep learning to assess if a project is in danger of surpassing the deadline, and to assess what the best team members to be on a to be started project. These assessments would be based on historical data. This tool should also be able to alert the project manager when it predicts that a project is going sideways, via an integration with web application that flags the projects at risk.

In the following chapters it is described the technical aspects of the proposed solution.

3 Developed Solution

This chapter describes the actual solution. It starts by explaining the differences between what was initially planned and what was actually implemented in this dissertation's work, followed by a detailed analysis of the data collection, then approaching what was done to understand that data and extract information from it, followed by listing and explaining the methods and tools used to achieve what was planned, then explaining what data protection, security, and ethics concerns were taken into account when developing the project, finally ending with a project architecture diagram and a discussion about what was described in the present chapter.

3.1 Differences between the planned solution and the developed solution

The present section describes the difference between what was initially planned and what was implemented for this dissertation's work and the reasons that led to it.

3.1.1 From Project Deadline Prediction to Project Delay Prediction

The initial goal for this use case was to predict the date of when the project was going to be completed. After some discussion, it was decided that this goal was not logical because there was not the need to determine the end date of a project, but the need to determine whether a project was going to be delayed or not. One can deliver a project before the expected end date and the project can still be delayed, due to the number of hours spent on that project being superior to what was initially estimated for it. So, because it was not logical nor needed to predict the deadline date, this use case was slightly adjusted to better fit the PMO's needs.

3.1.2 From Cost Estimation for Delayed Projects to Percentage of Delay Estimation for Delayed Projects

The initial goal for this use case was to predict the monetary cost that a delay in a project would bring to the organization. To do this, it was planned that a mechanism would predict the amount of hours that a project would be delayed and multiply that amount of hours by the current hourly rate of the organization, resulting in a monetary amount that would be the cost for the

organization. Due to the values that would allow the mechanism to predict that number of hours not being filled, this was not possible. As a workaround, it was brainstormed that it would be possible to, instead of predicting the number of hours that a project would be delayed, predict the percentage of delay based on the estimated hours for said project. This proposed solution would be possible and would offer a similar outcome for the PMO, fulfilling its need to have some sort of estimation of delay.

3.1.3 From three data sources to one data source

It was initially planned that three data sources would be used, being those sources Triskell, Jira, and Data Warehouse. Unfortunately, due to some delays from the organization for which this project is being developed for, it was decided that only the available data, from Triskell, would be used, to avoid any delays in the development of both this dissertation and this project.

3.1.4 From sending warnings to Jira to a web application that shows all the project information

The initial goal was to send a warning to Jira that would warn the PMO that a project is expected to be delayed, but because Jira was not ready to use by the end of the development, it was decided that a web application that shows all the projects would be used instead. This web page is responsible for flagging the projects that are expected to be delayed, show the expected delay percentage, and show the suggested technical contributors for to be started projects.

3.2 Data collection

The present section starts by explaining what data sources were used in the development of this tool, followed by detailing the datasets.

3.2.1 Data sources

The data source used by this tool is Triskell (*Enterprise Portfolio Management - Link Strategy with Execution*). Data regarding spent, allocated, and estimated time in each project was extracted from this data source, as well as the project's actual, and predicted start and end date, and the team members associated to said project. Percentage of completion, project's description, partners, and other project related data were also extracted from this data source.

3.2.2 Triskell Datasets

Triskell (*Enterprise Portfolio Management - Link Strategy with Execution*) is an enterprise Project Portfolio Management (PPM) solution that allows "to capture, align, link objectives, initiatives, programs and portfolios across the entire enterprise to get business challenges under control and ensure results" (*Enterprise Portfolio Management - Link Strategy with Execution*). This solution is implemented in the organization for which this project is developed for, and the next sections will describe what the data content and structure is.

3.2.2.1 Projects file content

This dataset is composed of one thousand, six hundred forty-five (1,645) lines and thirty-one columns. Each line represents a concluded, frozen, or cancelled project and each columns represents an attribute of the project.

The most important columns that compose this dataset are detailed in Table 24.

Table 8 - Triskell's Project dataset columns

Column name	Description of contents
Name	Name of the project composed by "PRJ-" followed by a numeric identification, for example, PRJ-12345
Description	Description of the project
Workflow	State of the project (Complete, Frozen, Cancelled)
Project Technical Manager	Name of the project technical manager
Technical Contributor	Name of the technical contributor
Project Manager	Name of the project manager
Project Category	Project's category (Security, System integration, Extranet, …)
Project Owner	Owner of the project
Customer	Customer for whom the project is being developed to

Partner	Project's partner (in case it is an outsourced project)
Baseline Start	Predicted start date of the project
Actual Start	Actual start date of the project
Start	Date of when the project entry was created
Estimated Hours	Estimated hours for the project completion
Actual Hours	Actual hours used for the project completion
Allocation Hours	Hours allocated to the project
Deviation Hours	Difference between predicted hours and allocated hours

From the thirty-one columns, only seventeen were considered useful to the project, since the value of the useless columns was either empty or had the same value in each row. Having the same value in each row is useless because it changes nothing in the final model.

3.2.2.2 Timesheet file content

The timesheet dataset is composed of three million two-hundred twenty-two thousand, six hundred forty-four lines and eleven (3 222 644) columns. Each line represents time the spent by a certain user in a task of a project in a certain day, and each column represents the attributes of said entry. The most important columns are detailed in Table 25.

Table 9 - Triskell's Timesheet dataset columns

Column name	Description of contents
Root name	Name of the project composed by "PRJ-" followed by a numeric identification, for example, PRJ-12345
Management Status	Status of the approval of the time entry (Approved, Rejected)
User ID	Unique identifier of the technical contributor
User Name	Name of the technical contributor
Hours	Hours spent in the task in this date

Of the eleven columns, only five were considered useful to the project because the other columns were detailing the time entry characteristics, something that is not needed for this project.

3.3 Data Understanding and Extraction

This section describes what was done to understand and extract the data from the dataset. It explains both the detail of the exploratory data analysis and some steps of the data preprocessing executed to understand the kind with which one is working.

3.3.1 Exploratory Data Analysis

One of the steps in the development of this project was the exploratory data analysis. The exploratory data analysis is the process of understanding what kind of data one is working with, the amount of data that is missing, treating outliers in the data, ask questions about data, and define the feature variables and the target variables. Feature variables are the variables that are fed to the model for it to predict the target variables. Correspondingly, the target variables are the variables that are to be predicted based on the feature variables.

A better understanding of the data allows for a smoother data preprocessing.

The next section will describe the exploratory data analysis done for each use case.

3.3.1.1 Project Delay Prediction

Firstly, it is important to get to know the data types of the variables one is working with, because different techniques are applied depending on what types of variables the data is composed by. In Table 10 the types of the variables used by this model are described.

Table 10 - Data Types of the columns used in the “Project Delay Prediction”

Column name	Dataset	Data Type
Project Technical Manager	Project	Categorical
Technical Contributor	Project & Timesheet	Categorical

Customer	Project	Categorical
Baseline Start	Project & Timesheet	Date
Actual Start	Project & Timesheet	Date
Start	Project & Timesheet	Date
Estimated Hours	Project	Numeric
Deviation Hours	Project	Numeric

A new variable, “Project Delayed”, was also created based on whether the “Deviation Hours” value was negative or positive. If the “Deviation Hours” has a negative value, the “Project Delayed” value would be “Delayed”, and if the “Deviation Hours” has a positive or zero value, the “Project Delayed” value would be “Not Delayed”. The “Project Delayed” variable has a categorical data type.

Regarding the feature and target variables and having into account that the goal of this use case is to predict whether a project is going to be delayed or not, it was decided that the target variable is the newly created “Project Delayed” variable and the remaining variables are the features, with the exception of the “Actual Hours” column due to the fact that the this information would not be available for the ongoing projects.

To conclude the exploratory data analysis for this use case, the data was also checked to remove any existing outliers.

3.3.1.2 Percentage of Delay Estimation for Delayed Projects

The same process was applied to the second use case. In Table 11 it is described the types of the variables used by this model.

Table 11 - Data Types of the columns used in the “Percentage of Delay Estimation for Delayed Projects”

Column name	Dataset	Data Type
Project Technical Manager	Project	Categorical

Technical Contributor	Project & Timesheet	Categorical
Customer	Project	Categorical
Baseline Start	Project & Timesheet	Date
Actual Start	Project & Timesheet	Date
Start	Project & Timesheet	Date
Estimated Hours	Project	Numeric
Actual Hours	Project	Numeric

A new variable, “Percentage Deviation”, was also created based on the difference between the “Actual Hours” and the “Estimated Hours” value divided by one hundred. After this relation is made, the row is assigned to one of the possible categories of “Percentage Deviation”, listed on Table 12. Only negative differences were considered because the goal of this use case is to determine the percentage deviation of delayed projects, so it would introduce noise if not delayed projects were considered.

Table 12 - Percentage Deviation possible categories

Percentage Deviation Category	Description
0-20	Rows with a percentage of deviation bigger than or equal to zero and smaller than twenty will have this category
20-40	Rows with a percentage of deviation bigger than or equal to twenty and smaller than forty will have this category
40-60	Rows with a percentage of deviation bigger than or equal to forty and smaller than sixty will have this category
60-80	Rows with a percentage of deviation bigger than or equal to sixty and smaller than eighty will have this category
80-100	Rows with a percentage of deviation bigger than or equal to eighty and smaller than one hundred will have this category
100+	Rows with a percentage of deviation bigger that one hundred will have this category

As it can be observed the “Percentage Deviation” variable has a categorical data type.

Regarding the feature and target variables and having into account that the goal of this use case is to predict the percentage deviation category in which a delayed project fits in, it was decided that the target variable is the newly created "Percentage Deviation" variable, and the remaining variables are the features, with the exception of the "Actual Hours" column due to the fact that this information would not be available for the ongoing projects.

To conclude the exploratory data analysis for this use case, the data was also checked to remove any existing outliers.

3.3.1.3 Team Member Suggestion

The same process was applied to the third use case. In Table 13 is described the types of the variables used by this model.

Table 13 - Data Types of the columns used in the "Team Member Suggestion"

Column name	Dataset	Data Type
Description	Project	String
Project Technical Manager	Project	Categorical
Technical Contributor	Project & Timesheet	Categorical
Project Category	Project	Categorical
Customer	Project	Categorical

Regarding the feature and target variables and having into account that the goal of this use case is to suggest team members for a to be started project, it was decided that the target variable is the "Technical Contributor" variable, and the remaining variables are the features.

3.3.2 Data preprocessing

The data preprocessing stage allows working and better performing models to be developed. This is because without data preprocessing, some models simply would not work or return poor results. This section describes the analysis done to the datasets regarding overall data preprocessing, not specifying by use case, since the same data sources and dataset are used,

and the only difference between use cases is the data used. Only the analysis will be described since what was done to solve the issues identified in the analysis will be detailed in the following sections, hence the missing parts of the data preprocessing.

3.3.2.1 Feature Imputation

Feature imputation consists in learning what values of the dataset are missing. The following table lists the columns and their corresponding number of missing values.

Table 14 - Number of missing values in the dataset

Column name	Number of missing values
Actual Start	991 missing values
Technical Contributor	909 missing values
Project Category	50 missing values
Baseline Start	7 missing values
Customer	6 missing values
Project Technical Manager	No missing values
Start	No missing values
Estimated Hours	No missing values
Actual Hours	No missing values
Description	No missing values

3.4 Methods and Tools

The present section starts by explaining what methods will be used in the development of this tool, followed by detailing what tools will be used to develop the tool.

3.4.1 Data preprocessing

Data preprocessing is composed of feature imputation, encoding, normalization, engineering, and selection, as well as dealing with data imbalances in the dataset. Because data preprocessing is feature oriented, this section describes what methods and tools were used in data preprocessing for each use case, because the used features vary in each of the use cases. Some of the stages are the same for all the use cases, so there is no detail by use case in those cases.

3.4.1.1 Feature Imputation

It was already observed in Table 14 what the number of missing values is for all the features. To be able to have a working model some data manipulation needed to be done, and a meeting with the organization was organized to decide how the data should be manipulated. Table 15 details the decision on how to manipulate each feature. This manipulation of data was applied in all use cases.

Feature imputation is important because most ML models do not work when there are missing values.

Table 15 - Manipulation of data in feature imputation

Column name	What was done
Actual Start	For each project in the project dataset the first time entry on the timesheet dataset was found, retrieving that date and assigning it to the Actual Start feature.
Technical Contributor	For each project in the project dataset all the time entries on the timesheet dataset were found, retrieving the time entry reporter and adding it to the Technical Contributor feature. Only the reporters that were not the Project Technical Manager were added to the Technical Contributor list.
Project Category	For the missing values in the Project Category a new category “Unknown” was created.
Baseline Start	This value should be filled with the Start of the project. In the event of the Start being empty, it should be filled with the Actual Start.
Customer	This value should be filled with "CLIENTE INTERNO - Divisão de Sistemas de Informação", because this value can only be empty when the Costumer is internal.

3.4.1.2 Feature Encoding

Feature Encoding is the process of turning values that are not numbers into numbers, since ML models require all values to be numbers to work.

With this said and based on the data gathered in the Exploratory Data Analysis section, follows how the several types of data were managed.

Table 16 – Feature encoding handling of the several types of data

Type of Data	What was done
Categorical	Every category was One Hot Encoded.
String	Only the "Description" feature is a string. For this feature, the stop words were removed, all the description was transformed into lower case, all the single letter words were removed, all the punctuation and special characters were removed, and then the result was stemmed. After this, TF-IDF vectorizer from sklearn was used to convert the string into a numerical format.

3.4.1.3 Feature Normalization

Feature normalization is the process of converting numerical values that are on different scales into the same scale, due to most ML models not performing well with not normalized data.

Regarding the developed project, feature standardization was used instead of feature normalization, since it simply performed better in the models.

3.4.1.4 Feature Engineering

The process of transforming raw data into features that better represent the problem is called feature engineering.

For both the "Project Delay Prediction" and the "Percentage of Delay Estimation for Delayed Projects" use cases, the "Date" feature was feature engineered. Each date had its day, month, and year extracted and that data was stored into a new feature. As an example, "Actual Start" was replaced by three new features "Actual Start Day", "Actual Start Month", and "Actual Start Year".

Also, for the “Project Delay Prediction” use case, the “Deviation Hours” feature was feature engineered. A new variable, “Project Delayed”, was created based on whether the “Deviation Hours” value was negative or positive. If the “Deviation Hours” has a negative value, the “Project Delayed” value would be “Delayed”, and if the “Deviation Hours” has a positive or zero value, the “Project Delayed” value would be “Not Delayed”.

Lastly, for the “Percentage of Delay Estimation for Delayed Projects” use case, the “Deviation Hours” feature was also feature engineered. A new variable, “Percentage Deviation”, was created based on the difference between the “Actual Hours” and the “Estimated Hours” value divided by one hundred. After this relation is made, the row is assigned to one of the possible categories of “Percentage Deviation”, listed on Table 12.

3.4.1.5 Feature Selection

The most relevant features were determined based on an eXtreme Gradient Boosting (XGBoost) model. By using this tool, one can get an understanding of what features are more valuable than others when predicting the target variable. The features used in each model were presented in Table 10, Table 11, and Table 13, being these the features used in the “Project Delay Prediction” use case, the “Percentage of Delay Estimation for Delayed Projects” use case, and the “Team Member Suggestion” use case correspondingly.

3.4.1.6 Data Imbalances

To finalize the data preprocessing, the data imbalance was resolved. To achieve a balanced dataset, Synthetic Minority Oversampling Technique (SMOTE) from ibmlearn (*Imbalanced-Learn Documentation — Version 0.12.3*) was used for oversampling the minority classes, and random undersampling was used for undersampling the majority classes. Random over sampling was used for the multilabel and multioutput problem of the “Team Member Suggestion” use case.

For the “Project Delay Prediction” use case, oversampling was done to ensure that the same number of “Delayed” and “Not Delayed” projects exist. In this case, there was the need to oversample the “Not Delayed” projects. Table 17 depicts how many final examples the dataset was left with in the “Project Delay Prediction” use case.

Table 17 - Dataset sampling for the “Project Delay Prediction” use case

Category	Number of examples per category before sampling	Number of examples per category after sampling
Not Delayed	485	628
Delayed	628	628

For the “Percentage of Delay Estimation for Delayed Projects” use case, both oversampling and undersampling was done to ensure that the same number of projects with the different percentages of delay exist. Both undersampling and oversampling were used because there were much more examples with delays between zero and twenty percent than the rest, so to prevent overfitting of the model these delays were under sampled, and the rest of the delays were oversampled. Table 18 depicts how many final examples the dataset was left with in the “Percentage of Delay Estimation for Delayed Projects” use case.

Table 18 - Dataset sampling for the “Percentage of Delay Estimation for Delayed Projects” use case

Category	Number of examples per category before sampling	Number of examples per category after sampling
0-20	128	68
20-40	68	68
40-60	38	68
60-80	29	68
80-100	18	68
100+	347	68

Finally, for the “Team Member Suggestion” use case oversampling was also used to ensure that all the technical contributors had contributed to the same number of projects. It is not possible to present all the categories in a table, due to the substantial number of technical contributors in the dataset, but every category was oversampled to have three hundred and forty-four (344)

examples. To do this, each category had to be oversampled independently with random over sampler. This had to be done this way because most, if any, oversampling techniques do not allow sampling for multilabel nor for multioutput problems.

3.4.2 Methods

This subsection details what methods were used to develop each use case. Both deep learning and ML models were assessed for all the use cases, so the best model for each use case could be found. There was no general rule of thumb to select what models should be assessed, the way it was done was by trying a large variety of models and then selecting the model that performs best.

3.4.2.1 Project Delay Prediction

Being the goal of this use case to predict whether a project is going to be delayed or not, this use case is considered a classification problem. The methods used for this classification problem are depicted in Table 19.

Table 19 – Models used in "Project Delay Prediction" use case

Type	Name
Machine Learning	Logistic Regression, XGBoost, SVM, Random Forest, KNN, Naïve Bayes, Decision Tree, Gradient Boost, and AdaBoost
Deep Learning	Recurrent Neural Network (RNN) and Bidirectional Long Short-Term Memory Network (Bi-LSTM)

The ML models always outperformed the deep learning models. In the beginning some traditional models were selected for training, like Logistic Regression and KNN, but those models proved to not perform as well as other more complex algorithms. Next, some ensemble models were trained, like Random Forest, XGBoost, and Gradient Boost, being the first and the last the better performing ones. Regarding the deep learning models, Bi-LSTM and RNN were trained, but both performed similarly and worse than any ML model.

Performance Tuning was applied for every ML model, using GridSearch, but even with this method applied there was no significant difference.

A stack ensemble model combining Logistic Regression, Random Forest, XGBoost, and SVM was also trained, but it did not perform that well.

3.4.2.2 Percentage of Delay Estimation for Delayed Projects

The same logic as the previous use case was applied here, because the same features used by the "Project Delay Prediction" are used to estimate the percentage of delay for delayed projects. The only thing that changed for this use case was the target feature. With this said, the same models presented in Table 19 were used in the present use case.

The ML models always outperformed the deep learning models, but this time the deep learning models did not even return usable results. Overall, all the models performed better in this use case, but like before the traditional models like Logistic Regression and KNN did not perform as well as other more complex algorithms. Some ensemble models were also trained, like Random Forest, XGBoost, and Gradient Boost, being the first and the last the second and third better performing ones, correspondingly. As stated before, the deep learning models, Bi-LSTM and RNN were trained, but both performed so badly that they will not be considered.

This time, the stack ensemble model combining Logistic Regression, Random Forest, XGBoost, and SVM was trained, and it was the best performing model.

Performance Tuning was applied for every ML model, using GridSearch, but even with this method applied there was no significant difference.

3.4.2.3 Team Member Suggestion

The "Team Member Suggestion use" case is considered a multilabel and multioutput classification problem, since the model should be able to suggest one or more technical contributors. The methods used in this classification problem are depicted in Table 20.

Table 20 – Models used in "Team Member Suggestion" use case

Type	Name
Machine Learning	Random Forest, KNN, SVM, XGBoost, DecisionTree, and Naïve Bayes
Deep Learning	RNN and Bi-LSTM

Finally, for the "Team Member Suggestion", the ML models always outperformed the deep learning models as well. Overall, all the models performed similarly in this use case, but this time the traditional models like Logistic Regression performed as well as other more complex algorithms. The best performing model was the ensemble model Random Forest, followed by the traditional model Logistic Regression, and then the Decision Tree model. As stated before, the deep learning models, Bi-LSTM and RNN were trained, but both performed so badly.

Performance Tuning was applied for every ML model, using Grid Search, but even with this method applied there was no significant difference.

3.4.3 Tools

For the successful development of the AI algorithms for the project in hand, some tools, libraries, and frameworks were used. The following section will describe them.

3.4.3.1 Programming Languages

For the development of this project, Python (*Welcome to Python.Org*) was used as the main programming language, due to its extensive libraries, community support, and robustness when dealing with AI. JavaScript was also used for the building of the web application.

3.4.3.2 Libraries and Frameworks

For ML, scikit-learn (*Scikit-Learn: Machine Learning in Python — Scikit-Learn 1.4.0 Documentation*) was used, which is a comprehensive library for machine learning tasks, including regression, classification, and clustering. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable (*XGBoost Documentation — Xgboost 2.1.1 Documentation*), and it was also used.

For Deep Learning, TensorFlow (*TensorFlow*), which is a deep learning framework for building and training neural networks, and keras (*Keras: Deep Learning for Humans*), which is an Application Programming Interface (API) that can also be used for building and training neural networks, were used.

For building and teste the models using Representational State Transfer (RESTful) APIs to inject them with data, Django (*Django*) was used.

For data manipulation and analysis, Pandas (*Pandas - Python Data Analysis Library*) and NumPy (*NumPy*) was used. Pandas is a powerful library for data manipulation and analysis, and NumPy is essential for numerical computations in Python.

Imbalanced Learning (*Imbalanced-Learn Documentation — Version 0.12.3*) was used for sampling, due to the datasets being imbalanced.

Finally, for the visualization of outputs, Matplotlib (*Matplotlib — Visualization with Python*), Seaborn (Waskom, 2021), and Plotly (*Plotly: Low-Code Data App Development*) were used. Matplotlib and Seaborn are libraries for creating static, interactive, and statistical visualizations, and Plotly provides interactive and web-based visualizations.

3.4.3.3 Development Environment

For the development environment, Visual Studio Code (VSCode) (*Visual Studio Code - Code Editing. Redefined*), who is an Integrated Development Environment (IDE) was used for writing, testing, and debugging code.

3.4.3.4 Cloud Services

Amazon Web Services (AWS) (*Cloud Computing Services - Amazon Web Services (AWS)*) was used for deploying the machine learning models.

3.5 Data Protection, Security, and Ethics

This section starts by explaining what the General Data Protection Regulation (GDPR) is and then it explains what data protection, security, and ethics precautions were considered when developing this tool.

3.5.1 GDPR

The GDPR is the “toughest privacy and security law in the world” (*What Is GDPR, the EU’s New Data Protection Law?*, 2018). With it, Europe is ensuring data privacy and security by regulating the collection or target of data related to people in the EU.

3.5.2 Data Protection

Regarding data protection, all personal data, ranging from client's data to team members data, is anonymous. This anonymity is considered because of the GDPR requirements. All data is collected, processed, and stored in a manner compliant with GDPR requirements.

All data processing activities are based on the legitimate interest lawful base specified in GDPR.

All the GDPR principles of data minimization, purpose limitation, and storage limitation were adhered to when handling project data.

3.5.3 Security

All the data is processed in the organization's servers, because using external servers would compromise the data anonymity.

All project data is safeguarded against unauthorized access, disclosure, alteration, or destruction.

The data used in this project is encrypted, has access controls, is target of pseudonymization, and regular security assessments.

3.5.4 Ethics

The AI algorithms were developed and deployed in an ethical manner, respecting the rights and dignity of individuals. Impact assessments shall be conducted to evaluate the potential risks to individuals' rights and freedom posed by the AI algorithms. Transparency and accountability in algorithmic decision-making processes is ensured. The use of AI algorithms in ways that could result in unfair discrimination or infringe upon individuals' rights to privacy and data protection were avoided. Biases and discrimination in AI algorithms' outputs, particularly in "Team Member Suggestion" algorithms, were avoided.

3.6 Project Architecture

In this section the project architecture is presented. **Erro! A origem da referência não foi encontrada.** displays the project architecture in three different layers: Data, Predictive Model, and Web Application. The data layer depicts the datasets extracted from Triskell, which are fed

to the predictive model layer. After being fed to the predictive model layer, the data is analyzed, and it is determined whether the project input is for an ongoing project or for a new project. In case it is a new project, the “Team Member Suggestion” is called to predict the best team members for said project. If is an ongoing project, the “Project Delay Prediction” model is called to predict whether the project is going to be delayed or not. If the project is going to be delayed, the percentage of estimation for delayed projects is called to predict the percentage of delay for said project.

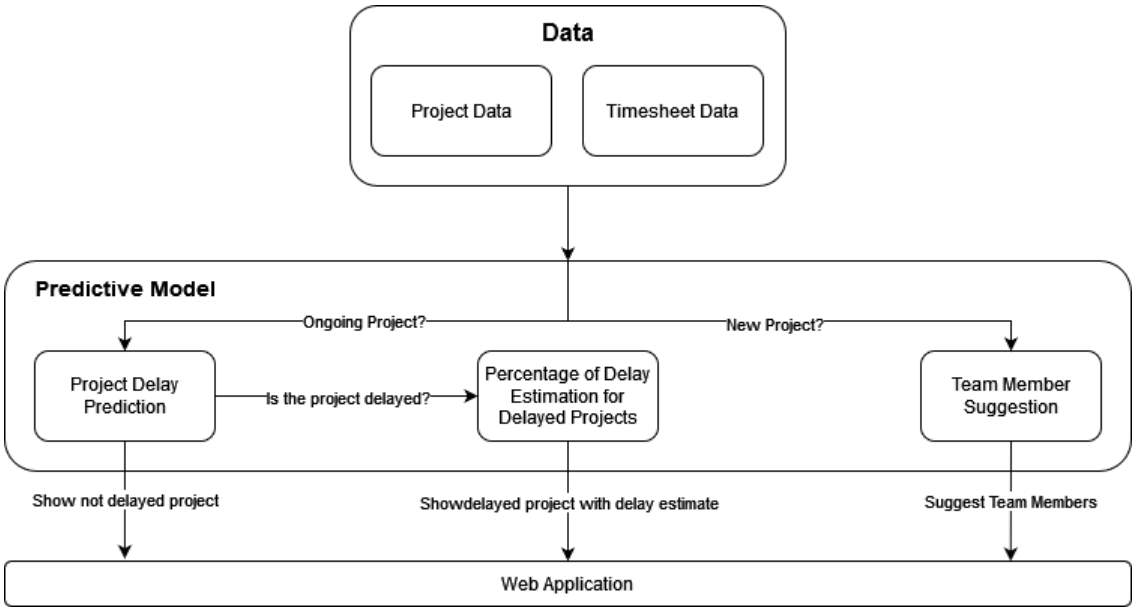

Figure 2 - Project Architecture

All the outputs returned by the three models are then transferred to the web application, which will maintain the list of projects, each one of them with its predictions.

3.7 Discussion

In this section some crucial details for the project development are discussed, as well as weighting the outcome of the developed solution. It started by describing the differences between the planned solution and the developed solution, followed by detailing the data sources and datasets used in the development of this project, next explaining what was done regarding understanding and extracting the data, then listing the methods and tools used for said development, ending with the data protection, security, and ethics concerns.

As explained before, the project had to suffer a switch from what was planned. The use cases to be adjusted due to unrealistic goals, because initial propose goals would only be realistic if there was more data to support them. The issue with the data sources, which went from three data sources to only one data source, because the other two data sources were not ready to be used by the time the development began, and, not to delay the development of this project and dissertation, it was decided that the development should continue without them. It was a fortunate decision, because until the moment of writing this section Jira and the Data Warehouse are still not available for usage.

With the data collection step, one learnt that the Triskell datasets had most of the information needed for a successful project because said datasets had mostly the same information as Jira's dataset, only with a different architecture. Because Triskell has two datasets, the connection between them needed to be found. All datasets have the same value for the project name and the technical contributor's name, to be able to link the data from these datasets. The only common column in all the datasets is the project name, which acts as a unique project identifier.

After the link between the two datasets was established, the data understanding and extraction step started, which was possible due to referred link between the two datasets. The project dataset and the timesheet dataset, both from Triskell data source, were analyzed. The timesheet dataset was used to support the project dataset, because the project dataset had some empty fields, and they were collected from the timesheet dataset. One example is the technical contributor field which has more than nine hundred values empty, according to Table 14. This field was populated with values from the timesheet dataset. To resume, the project dataset had the most valuable information, and the timesheet dataset aided in the feature imputation step.

After data understanding and extraction, data preprocessing ensued. Because there were several key features missing, feature imputation was needed. As noted before, the timesheet dataset aided in this process, since it held the key information needed for the project dataset. For the features that were missing from the timesheet dataset, the PMO provided standard values. Since the target features were not originally part of the dataset, we had to create them using feature engineering. We used feature encoding to ensure the methods worked properly, and feature normalization improved the performance of the models we applied.

Regarding the methods, several models were assessed to ensure that only the best performing ones were selected. For the “Project Delay Prediction” use case, the best performing model was Random Forest. This model was not much better than Gradient Boost, which could be chosen if this model was better time wise, but it was not the case. Regarding the “Percentage of Delay Estimation for Delayed Projects” use case, a Stack Ensemble model was the best performing model, and it could not be compared to other methods performance wise, due to the considerable gap comparing this method to the other tested methods. Finally, for the “Team Member Suggestion” use case, the best performing model was the Random Forest model, not far ahead from the Logistic Regression model. Because the Logistic Regression model is so close to the Random Forest model, and because it predicts a result much faster, one of the two models can be chosen depending on what the business needs are.

Data protection, security, and ethics concerns were considered when developing this project, due to the rising concerns on this matter.

In the next chapter, the experimentation and evaluation process will be detailed.

4 Experimentation & Evaluation

The present section combines everything that was described in Chapter 3 into three trained models that fulfill the PMO's needs for this project. In this section it is detailed how each model performed with all the use cases.

The experimentation & evaluation process is comprised of two parts: experimentation & evaluation before the models' deployment, and experimentation & evaluation after the deployment of the models. Experimentation & evaluation before the deployment details the performance metrics used to evaluate the models, how the models performed, what was done to perfect the model, and what was the selected model. Experimentation & evaluation after the deployment details what was done to evaluate the deployed models, and how to assess the impact that said models will have in the PMO and in the organization.

This division is needed because the models' performance is important both during the training and in a real-world scenario. The next sections will describe both parts of this process.

4.1 Before the deployment

This section describes how the experimentation and evaluation of the developed tool was done, what techniques were used, and what refining was done before it was deployed. The experimentation process before the deployment is done by training the models that were described in Chapter 3, and the evaluation process is done by comparing the models' performance using some performance metrics. Those performance metrics are also described in this section, since they are used to evaluate every model.

4.1.1 Performance metrics

This section describes the performance metrics used to evaluate models.

4.1.1.1 Confusion Matrix

Before describing the performance metrics, there is the need to explain how a model's prediction is classified. According to Figure 3, there are four possibilities when a model makes

a prediction: the prediction is a true positive, the prediction is a false positive, the prediction is a true negative, or the prediction is a false negative. In a binary classification problem, where the model needs to predict whether something happened or not, for a prediction to be a true positive it needs to be both, correctly predicted by the model and be positive, or in this case it needs to have happened. On the same note, for a prediction to be a true negative it needs to be both correctly predicted by the model and not to have happened. Regarding the false negative and false positive prediction, and on the same note as before, the first needs to be both incorrectly predicted by the model and not to have happened, and the latter needs to be both incorrectly predicted by the model and to have happened.

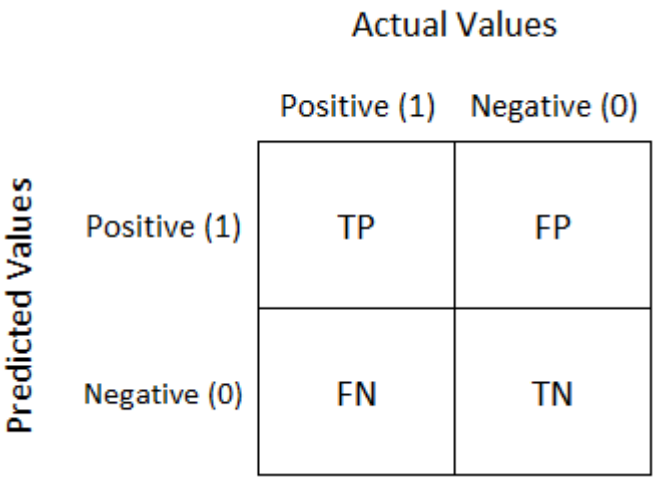

Figure 3 - Confusion Matrix Diagram (Narkhede, 2021)

4.1.1.2 Accuracy

Accuracy measures the proportion of correct predictions out of all predictions made by the model.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (1)$$

, where TP is the True Positive rate, TN is the True Negative rate, FN is the False Negative rate, and FP is the False Positive rate.

One should be careful when using this metric because it can be misleading when the datasets are imbalanced, but it should not be a problem in this dissertation's work because all data was balanced before training.

4.1.1.3 Precision

Precision measures the proportion of positive predictions that are correctly predicted.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

, where TP is the True Positive rate and FP is the False Positive rate.

This metric is useful when the goal is to limit the number of false positives.

4.1.1.4 Recall

Recall measures the proportion of actual positive values that were correctly identified by the model.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

TP is the True Positive rate and FN is the False Negative rate.

This metric is useful when the goal is to limit the number of false negatives.

4.1.1.5 Specificity

Specificity measures the proportion of negative values that were correctly identified by the model.

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

TN is the True Negative rate and FP is the False Positive rate.

This metric is useful when the cost of false positives is high.

4.1.1.6 F1-Score

F1-Score is the harmonic mean of precision and recall.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

This metric is useful when the goal is both, to limit the number of false positives and to limit the number of false negatives.

4.1.2 Project Delay Prediction

This section details what was done for experimenting and evaluating the "Project Delay Prediction" use case's model.

4.1.2.1 Performance

To measure the performance of a model one must resort to the performance metrics mentioned in section 4.1.1. Table 21 presents the performance metrics' values for the trained models of the "Project Delay Prediction" use case's model.

Table 21 - Performance metrics' values for the trained models of the "Project Delay Prediction" use case

Model	Accuracy	Precision	Recall	Specificity	F1-Score
Logistic Regression	0.6786	0.6790	0.6784	0.6786	0.6787
XGBoost	0.6905	0.6911	0.6901	0.6905	0.6906
SVM	0.6984	0.6984	0.6984	0.6984	0.6984
Random Forest	0.7540	0.7539	0.7539	0.7540	0.7539
KNN	0.6508	0.6505	0.6505	0.6508	0.6505
Decision Tree	0.7302	0.7318	0.7286	0.7302	0.7302
Naïve Bayes	0.6786	0.7435	0.5771	0.6786	0.6498
Stack Ensemble Model	0.6349	0.6390	0.6299	0.6349	0.6344

Before the deployment

Gradient Boost	0.7540	0.7603	0.7423	0.7540	0.7512
AdaBoost	0.7183	0.7183	0.7171	0.7183	0.7177

RNN and BI-LSTM did not perform well, so they were not presented here.

4.1.2.2 Model Refining

After finding the best models for the solution, hyperparameter tuning was done to optimize the performance of those models. Grid Search, Random Search, and Bayesian optimization were used to search the hyperparameter space efficiently. Even with these techniques applied, no relevant improvement was detected in the models.

4.1.2.3 Selected model

The goal for the "Project Delay Prediction" use case is to predict whether a project is going to be delayed or not. In a scenario like this, reducing both the number of false negatives and the number of false positives is of most importance because correctly identifying every project correctly is a priority, so safety measures can be applied for the project not to be delayed. The best performance metric to use when reducing the number of false negatives and false positives is key is the *F1-Score*, as mentioned in section 4.1.1.64.1.1.4. As presented in section 4.1.2.1, evaluating each model based on its F1-Score, Random Forest is the best performing model.

4.1.3 Percentage of Delay Estimation for Delayed Projects

This section details what was done for experimenting and evaluating the "Percentage of Delay Estimation for Delayed Project" use case's model.

4.1.3.1 Performance

The performance of the "Percentage of Delay Estimation for Delayed Project" use case's model must also be measured with the performance metrics mentioned in section 4.1.1. Table 22 presents the performance metrics' values for the trained models of the "Percentage of Delay Estimation for Delayed Project" use case's model.

Table 22 - Performance metrics' values for the trained models of the "Percentage of Delay Estimation for Delayed Project" use case

Model	Accuracy	Precision	Recall	Specificity	F1-Score
Logistic Regression	0.8065	0.7894	0.8001	0.8065	0.7947
XGBoost	0.9032	0.9018	0.8980	0.9032	0.8999
SVM	0.7742	0.7668	0.7098	0.7742	0.7372
Random Forest	0.8387	0.8121	0.8218	0.8387	0.8169
KNN	0.7218	0.6973	0.6937	0.7218	0.6955
Decision Tree	0.8911	0.8902	0.8745	0.8911	0.8823
Naïve Bayes	0.4718	0.7013	0.4349	0.4718	0.5369
Stack Ensemble Model	0.9194	0.9165	0.9179	0.9194	0.9172
Gradient Boost	0.8952	0.8903	0.8923	0.8952	0.8913
AdaBoost	0.5806	0.8469	0.5141	0.5806	0.6398

RNN and BI-LSTM did not perform well, so they were not presented here.

4.1.3.2 Model Refining

After finding the best models for the solution, hyperparameter tuning was done to optimize the performance of those models. *Grid Search, Random Search, and Bayesian Optimization* were used to search the hyperparameter space efficiently. As it was the case in the previous use case, even with these techniques applied no relevant improvement was detected in the models.

4.1.3.3 Selected model

The goal for the "Percentage of Delay Estimation for Delayed Project" use case is to predict the percentage of delay a delayed project has in relation with the predicted time. In a scenario like this, reducing both the number of false negatives and the number of false positives is of most importance since correctly identifying the percentage will dictate the safety measures that are

applied for the project not to be delayed. The best performance metric to use when reducing the number of false negatives and false positives is key is the F1-Score, as mentioned in section 4.1.1.64.1.1.4. As presented in section 4.1.3.1, and evaluating each model based on its F1-Score, the Stack Ensemble model is the best performing model.

4.1.4 Team Member Suggestion

This section details what was done for experimenting and evaluating the "Team Member Suggestion" use case's model.

4.1.4.1 Performance

Finally, the performance of the "Team Member Suggestion" use case's model must also be measured with the performance metrics mentioned in section 4.1.1. Table 23 presents the performance metrics' values for the trained models of the "Team Member Suggestion" use case's model.

Table 23 - Performance metrics' values for the trained models of the "Team Member Suggestion" use case

Model	Accuracy	Precision	Recall	Specificity	F1-Score
Logistic Regression	0.5773	0,7434	0.7620	1.0000	0.7526
XGBoost	0.5722	0,7523	0.7088	1.0000	0.7299
SVM	0.5773	0,7510	0.7424	1.0000	0.7467
Random Forest	0.5928	0,7756	0.7429	1.0000	0.7589
KNN	0.5876	0,7435	0.7095	1.0000	0.7261
Decision Tree	0.5825	0,7623	0.7431	1.0000	0.7526
Naïve Bayes	0.5206	0,7157	0.7748	1.0000	0.7441

RNN and BI-LSTM did not perform well, so they were not presented here.

4.1.4.2 Model Refining

After finding the best models for the solution, hyperparameter tuning was done to optimize the performance of those models. Grid Search, Random Search, and Bayesian optimization were used to search the hyperparameter space efficiently. However, like the previous use cases, even with these techniques applied no relevant improvement was detected in the models.

4.1.4.3 Selected model

The goal for the "Team Member Suggestion" use case is to suggest team members for a new project that is about to start. In a scenario like this, reducing both the number of false negatives and the number of false positives is of most importance since correctly identifying the team members that should be assigned the new project is key to the project's success. The best performance metric to use when reducing the number of false negatives and false positives is key is the F1-Score, as mentioned in section 4.1.1.64.1.1.4. As presented in section 4.1.4.1, and evaluating each model based on its F1-Score, Random Forest is the best performing model.

4.2 After the deployment

This section describes how the experimentation and evaluation of the developed tool was done after it was deployed. The experimentation process after the deployment is done by feeding the models that were described in Chapter 3 with some examples from the project's test data and check what their predictions are, and the evaluation process is done by comparing those predictions with the expected predictions.

4.2.1 Experimentation process

Initially the plan was to deploy the models and for them to be consumed by a web application that would show the project list pointing whether a project was delayed or not, and the suggested team members for to be started projects. The task of creating this web application was assigned to an element of the web development team of the organization for which this project is being developed for, that unfortunately was not able to develop it on a timely manner. To allow for the testing of the deployed models, and to somehow overcome this unforeseen issue, mock-up web pages were built with Django so one could see the models performing and returning some predictions. To ensure that the models are performing as they should real

examples from the test dataset were used. This is important because one can feed the models with this data and immediately assess if the result is the same as expected.

4.2.1.1 Project Delay Prediction

As previously explained, the experimentation process is done by feeding project examples from the test data to the deployed models so one can see if they perform as they should. Figure 4 presents the landing page of the mockup web application built only for testing of the deployed “Project Delay Prediction” model, as well as the prediction made by it.

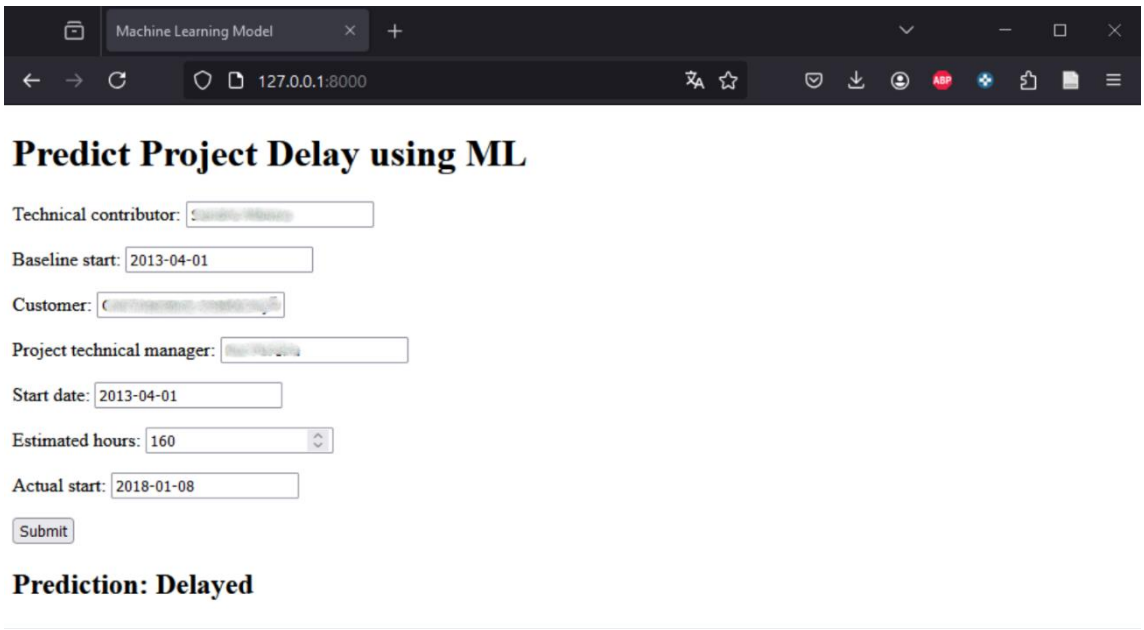

Figure 4 - Prediction made by the deployed "Project Delay Prediction" model

As can be seen in Figure 4, the model predicted that the project is going to be delayed, and in fact this is a delayed project from the test data.

4.2.1.2 Percentage of Delay Estimation for Delayed Projects

The same experimentation process is done on the “Percentage of Delay Estimation for Delayed Projects” model. Figure 5 presents the landing page of the mockup web application built only for testing of the deployed “Percentage of Delay Estimation for Delayed Projects” model, as well as the prediction made by it.

The same project data from the previous test was used since the goal of this use case is to predict the percentage of estimation of a delayed project.

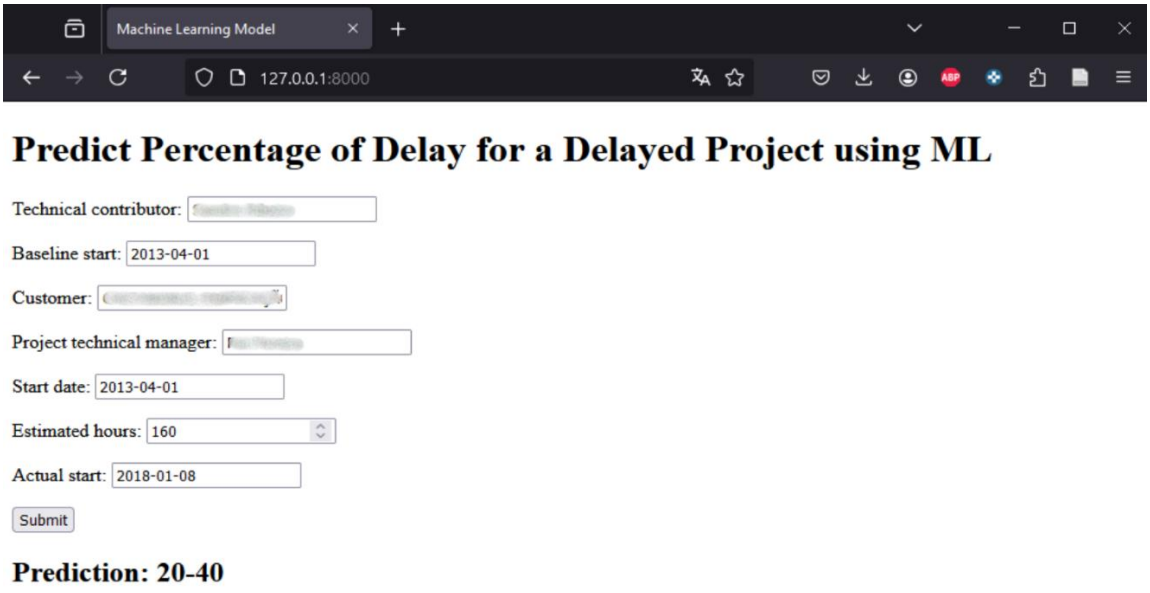

Figure 5 - Prediction made by the deployed "Percentage of Delay Estimation for Delayed Projects" model

In Figure 5one can see that the model predicted that the project is going to be delayed between twenty and forty percent, and in fact this project was delayed around thirty five percent.

4.2.1.3 Team Member Suggestion

Finally, the "Team Member Suggestion" model was also experimented in the same way as the previously presented ones. Figure 6 presents the landing page of the mockup web application built only for testing of the deployed "Team Member Suggestion" model, as well as the prediction made by it.

In this case, a dummy project from the test data was selected so one could be able to compare what technical contributors the model thinks are the best for said project and what technical contributor was responsible for developing the project.

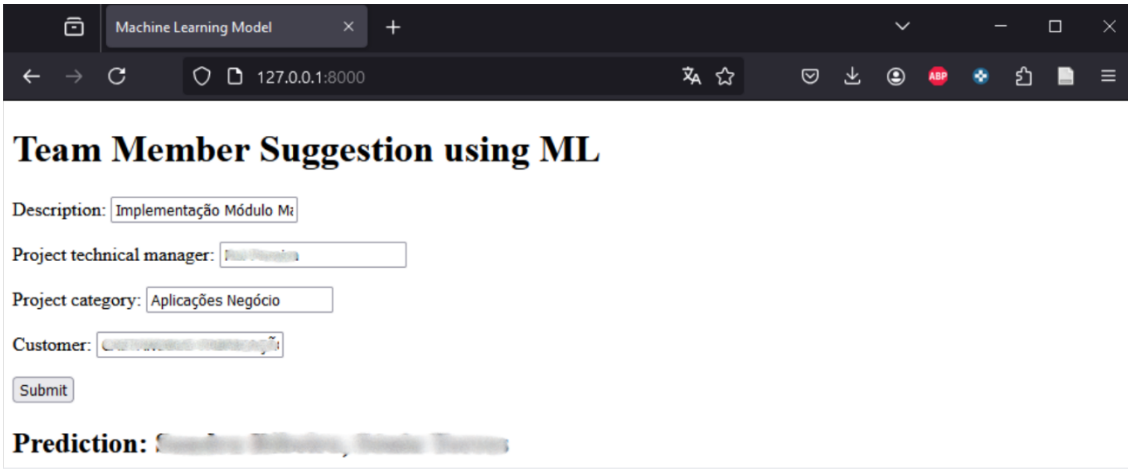

Figure 6 - Prediction made by the deployed "Team Member Suggestion" model

In Figure 6 one can see that the model predicted not only one, but two technical contributors, when the test case only had one technical contributor assigned to it. After checking with the organization, the additional technical contributor suggested by the model is part of the same team as the technical contributor that developed the project.

4.2.2 Evaluation Metrics

This section will describe how to evaluate the developed tool after the deployment. It describes the metrics used to assess the models and the general satisfaction of the organization. Evaluating the solution involves assessing its performance, effectiveness, and impact on the project management process. The evaluation process for the developed tool is in some way a work in progress. One cannot simply have a meeting with the PMO a week after the models were deployed and assess whether they were useful or not, simply because the impact is not felt right away. It is a work in progress due to taking a long time to assess that the models built really had a positive impact on the way the PMO manages its projects. With this said, one can state that the evaluation process will be evolutive.

4.2.2.1 Business Impact Metrics

There is no tool to evaluate the business impact, so only time will tell what the business impact is. There are, however, some objectives that can positively impact the business, like improving cost savings, improving time savings, and improving in the project success rate.

It is possible to improve cost savings by accurately predicting project delays and estimating their monetary impact. If some costs are saved, it is correct to assume that the algorithm had a positive impact in the business.

Time saving is the business impact that suggesting a team member to a project has, and this can be measured by measuring the time saved in manually selecting team members for projects.

By improving the project success rate, the organization's money is not wasted on projects that end up failing. This indicator can be evaluated by determining whether the implementation of AI algorithms leads to an increase in the number of projects completed on time and within budget.

4.2.2.2 User Satisfaction and Feedback

Another metric to evaluate if the project is having a positive impact on the organization is to gather feedback from project managers, team members, and other stakeholders on the usability and effectiveness of the tools. This is done by conducting surveys or interviews to understand user satisfaction, perceived usefulness, and areas for improvement.

5 Conclusion

The present chapter concludes the work developed in this dissertation. It presents a summary of each chapter, along with a critical appraisal of the work developed, and finally presents the answer to the investigation question. Future work is also addressed in this chapter.

5.1 Summary

Chapter 1 provides an overview of the subject and areas covered in this dissertation. It defines the problem statement, introduces the research question addressed in this chapter, outlines the goals and contributions of the dissertation, and concludes with an explanation of the document's structure.

Chapter 2 presents the content of the systematic review. It begins by explaining the PRISMA statement, defines the research questions, describes the search query and data sources, outlines the inclusion and exclusion criteria, and presents the search results. This chapter also explores existing tools, models, and algorithms, identifies gaps and challenges in AI for PMO, explains what a PMO is, and ends with a summary of the research findings.

Chapter 3 presents the actual solution, starting with the differences between the initial plan and what was implemented. It provides a detailed analysis of data collection, explains how the data was understood and processed, lists the methods and tools used, and addresses concerns about data protection, security, and ethics. The chapter concludes with a project architecture diagram and a discussion of the work described.

Chapter 4 focuses on the trained models that meet the PMO's project requirements. It explains how each model performed in various use cases. The process is divided into two phases: experimentation and evaluation before model deployment, which details the performance metrics, model refinement, and model selection; and experimentation and evaluation after deployment, which examines how the models were evaluated and their impact on the PMO and the organization.

Finally, Chapter 5 concludes the dissertation by summarizing each chapter, providing a critical appraisal of the work, answering the research question, and discussing the limitations and future directions for the research.

5.2 Investigation Question: Will the application of AI enhance the results of the PMO?

The answer to the investigation question defined in chapter 1 will allow to validate if the work being developed in this dissertation is relevant, needed, and desired.

To answer this question, it is needed to have into account what was stated in chapter 2.2. where were presented several works that proved helpful and enhanced the PMO, as well as some works in the field that mention that there is a huge gap in this area, urging the scientific community to invest in it. Even with the tool developed in this dissertation, it is possible to observe the benefits to the PMO. As mentioned in 1.1, every year around $48 trillion is invested in projects, but only 35% are considered successful, leading to significant wasted resources, due to outdated technologies, such as spreadsheets and slides, but applying AI and other innovations to project management could potentially improve project's success rates by 25%, translating to trillions of dollars in value (Nieto-Rodriguez & Vargas, 2023). So, the answer to the investigation question is, yes, the application of AI can enhance the results of the PMO. Regarding if the PMO results will be enhanced and considering with what was discussed in **Erro! A origem da referência não foi encontrada.**, only time will tell.

5.3 Achievements

The goal of this project was to enhance the PMO of the organization for which this project is being developed for. With the conclusion of the project one can say that the PMO now has better control not only over all the projects that exist in the organization, but also over what is the expected outcome of each project. This is a powerful tool that allow the project managers to better asses the ongoing projects, because even when they think that a project is going as expected it can be going sideways. With this the PMO expects to ensure fewer delays when delivering projects to the customer, reduce delay costs, and to have a better understanding of what is happening inside the organization regarding the projects.

Regarding the new projects, a tool that can suggest team members relieves the burden of, for instance, finding a candidate for a new project that usually was done by someone who already left the organization. Another useful application for this tool is for those managers that have a large team and many projects. Those managers cannot keep track of every team member that is currently assigned to a specific area or a specific type of projects, it can be useful for them to have the team member suggestion as a memory aid. With this, the PMO expects to achieve a faster start in new projects due to the aid this tool provides.

5.4 Contributions

As explained in section 2.2, the PMO area is lacking development of tools to enhance and optimize it. Most of the developments in this area are more focused on selecting profitable projects from portfolios with the purpose of investing then on monitoring the organization's projects to predict, and consequently try to prevent, their failure. There was no comparable work made in this area, so the contribution for the field is to introduce this area of development and show that the developed work is useful.

Regarding the team member suggestion, the same situation occurs. As far as it is known, no work was done in the area and similar ideas were not found in the investigation described in chapter 2, so the same contribution to the field was made.

5.5 Critical appraisal

The work developed in the present document is somehow interesting. In fact, just imagining the benefits that further development in this field of work could bring to companies, and the impact that it could have in the future of PMOs and in the future of project management itself, makes it is disappointing to see that the interest around the scientific community is not higher. Not only the companies could benefit from this, but every organization that has a lot of projects to manage, like a university, for example.

This research was a fantastic way to broaden horizons, now all that is left to do is to continue the work on the area, hoping to contribute even more.

It was a pity that not everything was concluded due to time constraints and other bespoken causes, but this project can be enhanced, in future work.

5.6 Limitations and Future work

For future work, it would be even more useful to create a chatbot that could answer questions regarding every project that is in the project portfolio, providing various statistics of how the project is performing, if it is on schedule, delayed, or with time to spare, and in case it has time to spare what would be the percentage range ahead of time of said project.

The integration of the developed models with a web application that can list the new or ongoing projects, flagging the ongoing ones that are predicted to delay and predicting the percentage of delay, and suggesting team members for the new projects was not completed, as stated before. So, for future work, this integration needs to be completed.

In a final meeting with the PMO it was stated that Triskell data now is audited, meaning that all fields are now mandatory. With this information, and with the improvement of data available, one can either train better models in the future or improve the ones developed in this project. With this said, another improvement for future work is to introduce online learning of the models so they can learn on the new and improved data. If this process does not improve the model, one must consider introducing batch learning.

Another case that was not possible in the present project was to use both the Jira and Data Warehouse data. The reason for this was explained previously, but another goal for future work is to introduce new models with data from these data sources, or even to replace the current models with data only for these two data sources, depending on the performance.

References

A guide to the project management body of knowledge: (PMBOK Guide) ; an American national Standard ANSI/PMI 99-001-2008 (4. ed). (2008). Project Management Inst. http://digitool.hbz-nrw.de:1801/webclient/DeliveryManager?pid=3423321

Auth, G., JokischPavel, O., & Dürk, C. (2019). Revisiting automated project management in the digital age – a survey of AI approaches. *Online Journal of Applied Knowledge Management (OJAKM)*, *7*(1), 27–39. https://doi.org/10.36965/OJAKM.2019.7(1)27-39

Beseiso, M., & Kumar, G. (2021). A fuzzy computational approach for selecting interdependent projects using prioritized criteria. *Journal of Intelligent & Fuzzy Systems*, *40*(6), 11341–11354. https://doi.org/10.3233/JIFS-202506

Bilal, M., & Oyedele, L. O. (2020). Big Data with deep learning for benchmarking profitability performance in project tendering. *Expert Systems with Applications*, *147*, 113194. https://doi.org/10.1016/j.eswa.2020.113194

Bolaños, R. D. S., & Barbalho, S. C. M. (2021). Exploring product complexity and prototype lead-times to predict new product development cycle-times. *International Journal of Production Economics*, *235*, 108077. https://doi.org/10.1016/j.ijpe.2021.108077

Cloud Computing Services—Amazon Web Services (AWS). (n.d.). Amazon Web Services, Inc. Retrieved September 26, 2024, from https://aws.amazon.com/

Dai, C. X., & Wells, W. G. (2004). An exploration of project management office features and their relationship to project performance. *International Journal of Project Management*, *22*(7), 523–532. https://doi.org/10.1016/j.ijproman.2004.04.001

Dam, H. K., Tran, T., Grundy, J., Ghose, A., & Kamei, Y. (2019). Towards Effective AI-Powered Agile Project Management. *2019 IEEE/ACM 41st International Conference on Software*

Engineering: New Ideas and Emerging Results (ICSE-NIER), 41–44. https://doi.org/10.1109/ICSE-NIER.2019.00019

Django. (n.d.). Django Project. Retrieved February 11, 2024, from https://www.djangoproject.com/

Enterprise Portfolio Management—Link Strategy with Execution. (n.d.). Triskell Software. Retrieved January 21, 2024, from https://triskellsoftware.com/

imbalanced-learn documentation—Version 0.12.3. (n.d.). Retrieved September 26, 2024, from https://imbalanced-learn.org/stable/

Jafarzadeh, H., Heidary-Dahooie, J., Akbari, P., & Qorbani, A. (2022). A project prioritization approach considering uncertainty, reliability, criteria prioritization, and robustness. *Decision Support Systems*, *156*, 113731. https://doi.org/10.1016/j.dss.2022.113731

Jira | Issue & Project Tracking Software | Atlassian. (n.d.). Retrieved January 21, 2024, from https://www.atlassian.com/software/jira

Keras: Deep Learning for humans. (n.d.). Retrieved September 26, 2024, from https://keras.io/

Liu, J.-W. (2019). Using big data database to construct new GFuzzy text mining and decision algorithm for targeting and classifying customers. *Computers & Industrial Engineering*, *128*, 1088–1095. https://doi.org/10.1016/j.cie.2018.04.003

Mahmoudi, A., Deng, X., Javed, S. A., & Yuan, J. (2021). Large-scale multiple criteria decision-making with missing values: Project selection through TOPSIS-OPA. *Journal of Ambient Intelligence & Humanized Computing*, *12*(10), 9341–9362. https://doi.org/10.1007/s12652-020-02649-w

Marnewick, C., & Marnewick, A. L. (2022). Digitalization of project management: Opportunities in research and practice. *Project Leadership and Society*, *3*, 100061. https://doi.org/10.1016/j.plas.2022.100061

Matplotlib—Visualization with Python. (n.d.). Retrieved February 11, 2024, from https://matplotlib.org/

Mossalam, A., & Arafa, M. (2018). Using artificial neural networks (ANN) in projects monitoring dashboards' formulation. *HBRC Journal*, *14*(3), 385–392. https://doi.org/10.1016/j.hbrcj.2017.11.002

Narkhede, S. (2021, June 15). *Understanding Confusion Matrix*. Medium. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Nieto-Rodriguez, A., & Vargas, R. V. (2023, February 2). How AI Will Transform Project Management. *Harvard Business Review*. https://hbr.org/2023/02/how-ai-will-transform-project-management

NumPy. (n.d.). Retrieved February 11, 2024, from https://numpy.org/

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *International Journal of Surgery*, *88*, 105906. https://doi.org/10.1016/j.ijsu.2021.105906

pandas—Python Data Analysis Library. (n.d.). Retrieved February 11, 2024, from https://pandas.pydata.org/

Philbin, S. P. (2016). Exploring the Project Management Office (PMO)–Role, Structure and Processes. *Proceedings of the International Annual Conference of the American Society for Engineering Management.*, 1–11. https://openresearch.lsbu.ac.uk/item/871wz

Plotly: Low-Code Data App Development. (n.d.). Retrieved February 11, 2024, from https://plotly.com/

Riesener, M., Kuhn, M., Keuper, A., & Schuh, G. (2023). A literature analysis on success factors and their corresponding scientific approaches in multi-project management. *Procedia CIRP*, *119*, 1176–1181. https://doi.org/10.1016/j.procir.2023.03.157

Savio, R. D., & Ali, J. M. (2023). Artificial Intelligence in Project Management & Its Future. *Saudi Journal of Engineering and Technology*, *8*(10), 244–248. https://doi.org/10.36348/sjet.2023.v08i10.002

Scikit-learn: Machine learning in Python—Scikit-learn 1.4.0 documentation. (n.d.). Retrieved February 11, 2024, from https://scikit-learn.org/stable/

Serviços de computação em nuvem—Amazon Web Services (AWS). (n.d.). Amazon Web Services, Inc. Retrieved February 11, 2024, from https://aws.amazon.com/pt/

Singh, V. (2017, July 14). *The Project Management Office: Aligning Strategy & Implementation | PMI*. https://www.pmi.org/business-solutions/white-papers/align-strategy-implementation

TensorFlow. (n.d.). TensorFlow. Retrieved February 11, 2024, from https://www.tensorflow.org/?hl=pt

Visual Studio Code—Code Editing. Redefined. (n.d.). Retrieved February 11, 2024, from https://code.visualstudio.com/

Wach, M., & Chomiak-Orsa, I. (2022). Determinants of the use of predictive models in the management of investment portfolios, on the example of KGHM Polska Miedź S.A. *Procedia Computer Science*, *207*, 2374–2383. https://doi.org/10.1016/j.procs.2022.09.296

Waskom, M. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Welcome to Python.org. (n.d.). Retrieved January 21, 2024, from https://www.python.org/

What is GDPR, the EU's new data protection law? (2018, November 7). GDPR.Eu. https://gdpr.eu/what-is-gdpr/

XGBoost Documentation—Xgboost 2.1.1 documentation. (n.d.). Retrieved September 26, 2024, from https://xgboost.readthedocs.io/en/stable/

Apendix A

1 Planned Solution

This chapter presents the planned solution as it was discussed with the organization team throughout the several meetings occurred. It starts by explaining the plan regarding what data sources to be used in the development of this tool, followed by detailing the datasets and explaining some crucial details for the project development, what methods would be used, ending by detailing what tools would be used.

1.1 Data collection

This section describes the data sources and datasets that were initially planned to be used in the development of this tool, their content, and their relevance.

1.1.1 Data sources

As referred before, the aim of this dissertation is to develop something that would be helpful to the PMO in question, which is the PMO from the organization this dissertation is being developed for, and that would be a tool that is able to monitor all of the work-in-progress projects with the goal of predicting deadline deviation, presenting an estimated cost to the organization in case that deviation occurs, and a tool that suggests the best team members to be on a to be started project based on the field of work and the category of said project. The data sources to be used by this tool are Triskell (*Enterprise Portfolio Management - Link Strategy with Execution*), an internal database from the organization called Data Warehouse, and Jira

(*Jira | Issue & Project Tracking Software | Atlassian*). From Triskell, data regarding time spent in each project will be extracted, as well as the project's deadline, the team members associated to said project, and the project's hourly fee. From the organization's internal database, data that maintains the relation between the client and the project will be extracted. Finally, from Jira, information regarding the project's tasks, who is assigned each task, the tasks percentage of completion, and the projects percentage of completion will be extracted.

1.1.2 Datasets

The following sections provide a detailed description of the datasets to be used in this project divided by data source. The data content, lines, and columns for each data source will be specified, and a brief description of the data's relevance will be given.

1.1.2.1 Triskell

Triskell (*Enterprise Portfolio Management - Link Strategy with Execution*) is an enterprise Project Portfolio Management (PPM) solution that allows "to capture, align, link objectives, initiatives, programs and portfolios across the entire enterprise to get business challenges under control and ensure results" (*Enterprise Portfolio Management - Link Strategy with Execution*). This solution is implemented in the organization for which this project is developed for, and the next sections will describe what the data content and structure is.

1.1.2.1.1 Data content

From Triskell, two datasets were extracted. The first dataset is composed of one thousand, six hundred forty-five (1,645) lines and thirty-one columns. Each line represents a concluded, frozen, or cancelled project and each columns represents an attribute of the project.

The most important columns that compose this dataset are detailed in Table 24.

Table 24 - Triskell's first dataset columns

Column name	Description of contents
Name	Name of the project composed by "PRJ-" followed by a numeric identification, for example, PRJ-12345
Description	Description of the project

Workflow	State of the project (Complete, Frozen, Cancelled)
Project Technical Manager	Name of the project technical manager
Technical Contributor	Name of the technical contributor
Project Manager	Name of the project manager
Project Category	Project's category (Security, System integration, Extranet, …)
Project Owner	Owner of the project
Partner	Project's partner (in case it is an outsourced project)
Baseline Start	Predicted start date of the project
Estimated Hours	Estimated hours for the project completion
Actual Hours	Actual hours used for the project completion
Actual Start	Actual start date of the project
Hourly Rate	Hourly rate for the project in euros

Of the thirty-one columns, only fourteen were considered useful to the project, mainly because the other columns were either empty or had the same value in every line.

The second dataset is composed of three million two-hundred twenty-two thousand, six hundred forty-four lines and eleven (3 222 644) columns. Each line represents time the spent by a certain user in a task of a project in a certain day, and each column represents the attributes of said entry. The most important columns are detailed in Table 25.

Table 25 - Triskell's second dataset columns

Column name	Description of contents
Name	Name of the project composed by "PRJ-" followed by a numeric identification, for example, PRJ-12345

Task	Task name
Technical Contributor	Name of the technical contributor
Date	Date of the time entry
Hours	Hours spent in the task in this date
Baseline Start	Predicted start date of the task
Estimated Hours	Estimated hours for the task completion
Actual Hours	Actual hours used for the task completion
Comment	Comment of the technical contributor

1.1.2.1.2 Data relevance

The data presented in the first dataset will be useful to predict what the best technical contributors are for each project, having into account whether the project fulfilled the deadline or not, and if the project's actual hours are according to what was predicted. This dataset will also aid, along with the second dataset, to predict if the project is in risk of not fulfilling the deadline, having into account the baseline start, actual hours, and estimated hours. Also, based on this dataset it will be possible to predict the cost that not fulfilling said deadline will have to the organization.

1.1.2.2 Data Warehouse

Data Warehouse is an internal database from the organization which keeps the relation between the projects maintained both in Triskell and in Jira, with the respective client.

1.1.2.2.1 Data content

From the Data Warehouse, one dataset was extracted. This dataset is composed of one thousand, six hundred forty-five (1,645) lines and two columns. Each line represents a concluded, frozen, or cancelled project and the two columns represent the relation between the client and the project.

Table 26 details the columns that compose the dataset from the Data Warehouse.

Table 26 - Data Warehouse's columns

Column name	Description of contents
Project Name	Name of the project composed by “PRJ-” followed by a numeric identification, for example, PRJ-12345
Client Name	Name of the project’s client

1.1.2.2.2 Data relevance

The data extracted from the Data Warehouse is important to be able to relate the project’s data from Triskell, the data from Jira, and the client for whom the project is being developed for.

1.1.2.3 Jira

Jira (*Jira | Issue & Project Tracking Software | Atlassian*) is a project management tool used to plan, track, release and support software. Jira makes it easier for teams to move work forward and stay aligned with the project goals. This solution is implemented in the organization for which this project is developed for, and the next sections will describe what the data content and structure is.

1.1.2.3.1 Data Content

From Jira, one dataset was extracted. This dataset is composed of one thousand, six hundred forty-five (1 645) lines and fifty-two columns. Each line represents a concluded, frozen, or cancelled project and the two columns represent the relation between the client and the project.

Table 27 details the most important columns that compose the dataset from Jira.

Table 27 - Jira's columns

Column name	Description of contents
Project Name	Name of the project composed by “PRJ-” followed by a numeric identification, for example, PRJ-12345
Project % Completion	Percentage of completion of the project

Task	Task name
Task % Completion	Percentage of completion of the task
Assigned To	Name of the technical contributor

Of the fifty-two columns, only five were considered useful to the project, mainly because the other columns data was already present in Triskell's dataset.

1.1.2.3.2 Data relevance

The data present in this dataset will be used along with the datasets from Triskell to predict if a project is in risk of not fulfilling the deadline by checking the project percentage of completion and the task percentage of completion with the deadline from Triskell's dataset.

1.1.3 Discussion

The present section points out some crucial details for the project development.

All datasets have the same value for the project name, the task name, or the technical contributor's name, to be able to link the data from these datasets. The only common column in all the datasets is the project name, which acts as a project unique identifier.

The dataset from Triskell allow to process project related data, like the project's name, tasks, technical contributors, estimated times and dates, actual times and dates, the number of hours spent by a specific technical contributor in a task, and the hourly rate for the project.

The data from Jira allow to process project and task related data, like the task's name, technical contributors, project's percentage of completion, and tasks' percentage of completion.

Finally, the data from the Data Warehouse allow to link the data from Triskell and Jira to a customer, which can provide the possibility of relating project deviation from the deadline with a specific client.

1.2 Methods and Tools

The present section starts by explaining what methods were planned to be used in the development of this tool, followed by detailing what tools were to be used.

1.2.1 Methods

As described before, the method to be used is deep learning. The use of deep learning will allow the tool to learn if the project is going according to plan based on previous similar projects, predict delays in the project's delivery along with the cost to the organization based on the hourly rate of said project, and suggest what are the best fit team members for a starting project also based on previous similar projects. To integrate the tool with Jira (*Jira | Issue & Project Tracking Software | Atlassian*), a web service will be used so the tool can send the warning in case it predicts that a project is going sideways. This section describes the method to be used in each part of the project.

1.2.1.1 Project Deadline Prediction

To predict project completion dates based on features like planned hours, actual hours, and task completion percentage, a linear regression model will be used. Random Forest regression will also be used because it can capture nonlinear relationships between features and project completion dates.

1.2.1.2 Cost Estimation for Delayed Projects

To estimate the cost impact of potential delays, a regression model and a binary classification models will be used. The regression model will be responsible for predicting the cost overrun based on project features like delay duration and hourly rate. The role of the Binary Classification model will be to classify projects as "at risk" or "not at risk" based on their likelihood of delay, then estimate the cost overrun for "at risk" projects using the regression model.

1.2.1.3 Team Member Suggestion

For suggesting the best team members for a project, KNN will be used. KNN is a collaborative filtering model that will be able to recommend team members based on their past successful projects and their best performing projects, as well as their past successful tasks and best performing tasks. This will be done by comparing what team members have the greatest

performance in a task for a project in a certain area with those who have a lower performance, and then deciding based on that evaluation.

1.2.2 Tools

For the successful development of the AI algorithms for the project in hand, some tools, libraries, and frameworks will be used. The following section will describe them.

1.2.2.1 Programming Language

For the development of this project, Python (*Welcome to Python.Org*) will be used as the main programming language, due to its extensive libraries, community support, and robustness when dealing with AI.

1.2.2.2 Libraries and Frameworks

For ML, scikit-learn (*Scikit-Learn: Machine Learning in Python — Scikit-Learn 1.4.0 Documentation*) will be used, which is a comprehensive library for machine learning tasks, including regression, classification, and clustering. TensorFlow (*TensorFlow*), which is a deep learning framework for building and training neural networks, will also be used.

For building the Representational State Transfer (RESTful) Application Programming Interfaces (APIs) to serve the models, Django (*Django*) will be used.

For data manipulation and analysis, Pandas (*Pandas - Python Data Analysis Library*) and NumPy (*NumPy*) will be used. Pandas is a powerful library for data manipulation and analysis, and NumPy is essential for numerical computations in Python.

Finally, for the visualization of outputs, Matplotlib (*Matplotlib — Visualization with Python*), Seaborn (Waskom, 2021), and Plotly (*Plotly: Low-Code Data App Development*) will be used. Matplotlib and Seaborn are libraries for creating static, interactive, and statistical visualizations, and Plotly provides interactive and web-based visualizations.

1.2.2.3 Development Environment

For the development environment, Visual Studio Code (VSCode) (*Visual Studio Code - Code Editing. Redefined*), that is an Integrated Development Environment (IDE) will be used for writing, testing, and debugging code.

1.2.2.4 Cloud Services

Amazon Web Services (AWS) (*Serviços de computação em nuvem - Amazon Web Services (AWS)*) will be used for deploying the machine learning models.