UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Prediction of customer satisfaction level based on best practices followed in the software development process

Geovanne Borges Bertonha

Monograph - MBA in Artificial Intelligence and Big Data

ICMC USP

SÃO CARLOS

Geovanne Borges Bertonha

Prediction of customer satisfaction level based on best practices followed in the software development process

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Ricardo Cerri

Original version

São Carlos
2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B547p

Bertonha, Geovanne Borges
Prediction of customer satisfaction level based on best practices followed in the software development process / Geovanne Borges Bertonha; orientador Ricardo Cerri. -- São Carlos, 2024.
66 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Net Promoter Score. 2. Client Satisfaction. 3. Software Development Best Practices. 4. Artificial Intelligence. 5. XGBoost. I. Cerri, Ricardo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Geovanne Borges Bertonha

Prediction of customer satisfaction level based on best practices followed in the software development process

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Ricardo Cerri

São Carlos
2024

This study is dedicated to all managers and leaders
in software development companies who advocate for and support
data-driven and Lean philosophies.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to MTI Ltd. for providing the data used in this study and for their consistent support of my professional development.

Thank you to the faculty at USP ICMC, all the professors who lectured in the MBA in AI and Big Data course, and my advisor, Prof. Dr. Ricardo Cerri.

My family has been crucial in supporting me throughout this MBA program, especially my wife, Karen; my daughter, Hana; and my stepdaughter, Maria Geovana. Completing this MBA would not have been possible without their strong support.

This monograph is dedicated to my father, Wilsiney Bertonha, and my mother, Elisabeth B. F. Bertonha, for providing me with a solid education and everything I needed to become who I am today.

"In God we trust; all others bring data."
W. Edwards Deming

ABSTRACT

BERTONHA, G. **Prediction of customer satisfaction level based on best practices followed in the software development process**. 2024. 66 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

This study explores the prediction of client satisfaction in software development teams using Artificial Intelligence (AI) algorithms. The objectives are to determine whether client satisfaction, measured by the Net Promoter Score (NPS) (Reichheld, 2003), can be predicted based on best practices followed by software development teams, and to identify which practices most significantly impact client satisfaction. Data from ten agile software development teams at MTI Ltd. (MTI, 1996), collected through a proprietary CMMI5-like framework, were analyzed using Random Forest, XGBoost, Lasso Regression, ARIMA, and LSTM models. The study followed a structured methodology, including data preprocessing, exploratory data analysis, and model implementation. The results indicate that the XGBoost, Random Forest, and LSTM models successfully predicted NPS with satisfactory accuracy, measured by the Symmetric Mean Absolute Percentage Error (sMAPE) and Mean Absolute Error (MAE) (Shcherbakov *et al.*, 2013). The Lasso and AutoARIMA models indicated less favorable performance. Additionally, a Pearson correlation analysis identified key best practices that correlate with NPS. Some limitations were observed, including a small data set, inconsistencies in the data, and limited continuity of the time series. Future research could benefit from larger datasets, alternative prediction approaches, and improved feature selection methods. The study's findings offer valuable insights into optimizing software development practices to enhance client satisfaction.

Keywords: Client Satisfaction. Net Promoter Score (NPS). Software Development Best Practices. Artificial Intelligence (AI) Algorithms. Exploratory Data Analysis (EDA). Random Forest. XGBoost. Lasso Regression. ARIMA. Long Short-Term Memory (LSTM). Time Series Forecasting. Agile Teams. Predictive Modeling. CMMI5 Framework. Data-Driven Decision Making.

RESUMO

BERTONHA, G. **Prediction of customer satisfaction level based on best practices followed in the software development process**. 2024. 66 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Este estudo explora a previsão da satisfação do cliente em equipes de desenvolvimento de software utilizando algoritmos de Inteligência Artificial (IA). O objetivo principal é determinar se a satisfação do cliente, medida pelo Net Promoter Score (NPS) (Reichheld, 2003), pode ser prevista com base em práticas seguidas no processo de desenvolvimento de software e identificar quais delas impactam mais significativamente a satisfação. Dados de dez equipes ágeis de desenvolvimento de software da MTI Ltd. (MTI, 1996), coletados por meio de um framework proprietário semelhante ao CMMI5, foram analisados utilizando os modelos Random Forest, XGBoost, Lasso Regression, ARIMA e LSTM. O estudo seguiu uma metodologia estruturada, incluindo pré-processamento de dados, análise exploratória de dados e implementação de modelos. Os resultados indicam que os modelos XGBoost, Random Forest e LSTM previram com sucesso o NPS com precisão satisfatória, medida pelo Erro Médio Absoluto Percentual Simétrico (sMAPE) e pelo Erro Médio Absoluto (MAE) (Shcherbakov *et al.*, 2013). Os modelos Lasso e AutoARIMA indicaram um desempenho menos favorável. Além disso, uma análise de correlação de Pearson identificou as principais melhores práticas que se correlacionam com o NPS. Algumas limitações foram observadas, incluindo um pequeno conjunto de dados, inconsistências nos dados e continuidade limitada da série temporal. Pesquisas futuras poderiam se beneficiar de conjuntos de dados maiores, abordagens alternativas de previsão e métodos aprimorados de seleção de características. As descobertas do estudo oferecem informações úteis para otimizar práticas de desenvolvimento de software para melhorar a satisfação do cliente.

Palavras-chave: Satisfação do Cliente. Net Promoter Score (NPS). Práticas de Desenvolvimento de Software. Algoritmos de Inteligência Artificial (IA). Análise Exploratória de Dados (EDA). Random Forest. XGBoost. Regressão Lasso. ARIMA. Memória de Longo Prazo e Curto Prazo (LSTM). Previsão de Séries Temporais. Equipes Ágeis. Modelagem Preditiva. Framework CMMI5. Tomada de Decisão Baseada em Dados.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

USP	Universidade de São Paulo
USPSC	Campus USP de São Carlos
ICMC	Instituto de Ciências Matemáticas e de Computação
AI	Artificial Intelligence
ISO	International Organization for Standardization
CMMI5	Capability Maturity Model Integration 5
ITIL	Information Technology Infrastructure Library
EDA	Exploratory Data Analysis
NPS	Net Promoter Score
SVM	Support Vector Machine
tNPS	Transactional Net Promoter Score
KPI	Key Performance Indicator
COVID-19	Coronavirus Disease of 2019
VIF	Variance Inflation Factor
CFS	Correlation-based Feature Selection
RF	Random Forest
XGB	eXtreme Gradient Boosting
XGBoost	eXtreme Gradient Boosting
LASSO	Least Absolute Shrinkage and Selection Operator
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
ARIMA	Autoregressive Integrated Moving Average
CSV	Comma-separated Value
SQL	Structured Query Language

NaN	Not a Number
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
sMAPE	Symmetric Mean Absolute Percentage Error

CONTENTS

1 INTRODUCTION

This introductory chapter includes the contextualization of the topic of study, outlining its motivations and objectives. Additionally, it summarizes the chosen research methodology and provides an overview of this paper's structure.

1.1 Contextualization and problem statement

The software development process has experienced significant transformations since 1948 when Tom Kilburn, a computer scientist, wrote the first piece of software (Shelburne; Burton, 1998). This evolution continued with the creation of FORTRAN in 1957 (Backus, 1978) and reached many other important milestones. However, the concept of commercial software earned more traction due to notable advances in semiconductor technology, enabling the adoption of personal computers in the early 1970s (Abbate, 1999). During this period, companies began leveraging software to optimize their business processes. Software evolved into a commercial good, leading to the founding of multiple software development companies (Ceruzzi, 2012). Consequently, a significant need emerged for software development companies to implement robust processes and best practices. These measures aimed to develop commercial software and improve predictability in company operations, delivering high-quality, stable software that met their client's expectations, ultimately fostering increased client satisfaction.

Over the past few decades, numerous process maturity frameworks have been introduced, including well-known examples such as ISO, CMMI5, and ITIL (Pernet; Cano, 2014). Some software development companies have adopted these frameworks, while others have chosen to formulate their own set of best practices. Regardless of the approach selected by these companies, some pertinent questions arise for software development managers:

- **Q1** "Is it possible to predict the level of client satisfaction based on the best practices followed by the teams?".
- **Q2** "Which best practices have a more direct impact on achieving a high level of satisfaction?".

1.2 Justification

We are currently in the digital transformation era, where software plays an important role in delivering innovation to customers. The increase in the commercialization of software has also contributed to the rise in the number of software development companies. In this competitive landscape, a software development company must find alternatives to stand out continually (Kraus1 *et al.*, 2021). Customizing processes by selecting a set of best practices that align with the primary goal of achieving client satisfaction represents a competitive advantage and is highly valued in this context.

Considering the large number of frameworks, processes, and best practices available in the software development industry, and given the need to create lean and cost-effective teams, it is pertinent for software development team managers to understand which best practices have the most significant influence on customer satisfaction. This understanding is crucial for organizing team activities to prioritize the most relevant tasks, ensuring that they align with the ultimate goal of achieving the highest customer satisfaction levels.

With the continuous evolution of computing power and, consequently, the advance of cloud computing, companies across diverse sectors can now store and process extensive datasets related to their business operations (Hashem *et al.*, 2015). This ability empowers them to conduct essential analyses, extracting useful insights that can impact their decision-making processes. This is particularly relevant for software development companies that are developing commercial software. By gathering comprehensive information about their software development teams and embracing a data-driven methodology, the next step is to explore the integration of Artificial Intelligence (AI). It can help companies improve their predictive capabilities when developing commercial software, enabling them to anticipate important information that can effectively drive their strategies.

1.3 Objectives

In the context outlined above and in response to the questions arising from this landscape, the objectives of this study are as follows:

- **O1** Conduct an exploratory data analysis on a dataset that includes information collected from various software development teams.
- **O2** Utilize and implement Artificial Intelligence algorithms to create models capable of predicting client satisfaction levels (NPS) based on the best practices followed by software development teams.

Throughout this study, the expectation is to gather insightful information from exploratory data analysis and discover a model that demonstrates high performance in predicting client satisfaction levels. Additionally, the expectation is for this model to showcase a strong generalization capability, even when applied to data from diverse software development teams operating in different contexts.

This study utilizes a four-step approach consisting of (1) data preprocessing and transformation, (2) exploratory data analysis, (3) implementation of different types of AI algorithms, and (4) performance analysis and comparison. The details are explained in the Chapter 3.

1.4 Paper structure

This document begins with a "Theoretical Foundation" (Chapter 2), including a review of the literature, and an overview of the algorithms selected to address the problem described in this study. Following this, the "Methodology" (Chapter 3) outlines the methods and objectives. Lastly, in the "Experimental Evaluation" (Chapter 4), the evaluation criteria are highlighted, followed by the "Conclusion" (Chapter 5), which wraps up the paper by outlining the conclusions and key takeaways.

2 THEORETICAL FOUNDATION

This chapter provides a review of previous studies done in this field. It also includes some background information and explains the AI algorithms that will be used in the study experiment.

2.1 Literature review

Using AI algorithms to train models for predicting client satisfaction isn't new. After some research, articles by authors who have done this can be found. For instance, there was a study published in Elsevier titled "*Churn and net promoter score forecasting for business decision-making through a new stepwise regression methodology.*" (Vélez *et al.*, 2020), which focuses on the NPS. In this article, the authors suggested forecasting the likelihood of current clients ending their relationship with a company — known as "churn" — and predicting the NPS. They divided the NPS into three classes: "Detractors", "Neutral", and "Promoters". The study tested various algorithms, including Classification Tree, Nominal Logistic Regression, Binary Logistic Regression, Ordinal Logistic Regression, Neural Network, Binary SVM, and Gradient Boosting model. The Gradient Boosting algorithm with two-level-deep trees produced the best results among predictive methods, achieving 54% accuracy. While there are similarities in predicting the NPS, unlike our study, this mentioned article doesn't relate to best practices in processes.

In a second study titled "*Prediction of Customer Transactional Net Promoter Score (tNPS) Using Machine Learning*" (Kannan *et al.*, 2022), the author utilized multiple attributes originating from complaints of customers of a telecommunication company and applied five different machine learning algorithms to predict their tNPS: Decision Tree, Random Forest, Gradient Boosted Trees, Logistic Regression, and Multilayer Perceptron Neural Network. In this study, the Multilayer Perceptron Neural Network performed the best compared to the rest. This also differs from what is being proposed in our study, as the data has no relation with a software development team applying best practices.

Another study titled "*Relationship between the Net Promoter Score and the Key Performance Indicators using machine learning techniques*" (Huygevoort, 2021) aimed to explore the relationship between the NPS and the Key Performance Indicators (KPIs) of a particular company. The research focused on KPIs relevant to their business, such as service level, on-time delivery, transport punctuality, and quality assurance, to assess their impact on NPS. Machine learning algorithms, including Linear Regression, Support Vector Regression, Decision Trees, and Random Forest, were utilized in the analysis. The findings revealed no relevant relationship between the NPS and the KPIs used by the company. Although this study shares some similarities with the research presented in this paper, the features examined are distinct. The mentioned study focused on KPIs, whereas this study investigates the influence of best practices in software development on NPS.

In the book "*Accelerate: Building and scaling high performing technology organizations*" (Forsgren; Humble; Kim, 2018), Nicole Forsgren, Jez Humble, and Gene Kim used snowball sampling to gather data from responders at various companies in the software industry. The survey included questions about software development practices, such as *DevOps*. The authors then conducted correlation tests and classification methods. The study found that some of these practices have a measurable impact on organizational performance. Unlike the study reported in the book, in this paper's study the data from the best practices survey is verified with evidence provided by the responders. This approach is believed to result in more accurate information compared to relying solely on respondents' answers. On the other hand, the amount of information evaluated in this paper is relatively smaller than the data collected by the authors of the book. Apart from that, the fundamental difference is that while the book focuses on classifying companies into performance groups, this study aims to predict customer satisfaction levels using the NPS.

2.2 AI algorithms used in this study

Time series applies to a wide range of use cases where the main interest is forecasting the future. Although traditional time series algorithms have been applied to problems such as stock data and weather patterns, literally any data that varies over time can be analyzed using time series methods (Nielsen, 2020). The Chapter 3 explains that data is collected regularly at fixed intervals, suggesting that time series forecasting algorithms could be a good fit for this problem.

For this study, five artificial intelligence algorithms were chosen: two tree-based algorithms, Random Forest (RF) (Biau; Scornet, 2016) and Extreme Gradient Boosting (XGB) (Wang; Guo, 2020); a modification of linear regression, Least Absolute Shrinkage and Selection Operator (LASSO) (Ranstam; Cook, 2018); a widely used algorithm for time series problems, Autoregressive Integrated Moving Average (ARIMA) (Siami-Namini; Tavakoli; Namin, 2018); and a recurrent neural network, Long Short-Term Memory (LSTM) (Siami-Namini; Tavakoli; Namin, 2018).

3 METHODOLOGY

This chapter describes the methodology employed in this study.

3.1 Understanding the data sources

This research will use data from ten agile teams at the Japanese company MTI Ltd (MTI, 1996). The data is sourced from a questionnaire covering tens of best practices that the teams have been required to respond to regularly since mid-2021. For each best practice, team members answer "yes" (1) if it is followed, or "no" (0) if it is not. The best practices in this survey are part of MTI Ltd.'s proprietary CMMI5-like framework.

The target variable is the variable that an AI study aims to predict. In this study, the target variable is the Net Promoter Score (NPS). First introduced by Frederick F. Reichheld, NPS has since become widely used across various industries worldwide. The methodology involves asking the question:

"How likely is it that you would recommend our company to a friend or colleague?"

The responses are then grouped into three categories according to a 0 to 10 rating scale:

- "Promoters" (rating of 9–10, extremely likely to recommend)
- "Passively satisfied" (rating of 7–8)
- "Detractors" (rating of 0–6, extremely unlikely to recommend)

Next, by subtracting the percentage of detractors from the percentage of promoters, a number ranging from -100 to +100 is obtained. Companies with exceptional customer loyalty typically achieve NPS of 75% or higher. (Reichheld, 2003).

The NPS scores have also been collected from all agile teams' customers at MTI Ltd. since mid-2020. In this study, the best practices survey data and NPS data will be combined into a single data source for further application of AI algorithms.

3.2 Approach utilized in this study

The process summarized below consists of a four-step approach utilized in this study. It will enable the preparation of the data and a deep understanding of the features and the target variable, ensuring that the AI algorithms can be implemented based on a high-quality dataset:

1. **Data preprocessing and transformation:** Firstly, a series of preprocessing and transformation techniques will be applied to prepare the dataset.

2. **Exploratory data analysis (EDA):** An exploratory analysis of the dataset will be conducted to uncover potential insights into the dataset. During this analysis, the features that will make up the training dataset for the further step will be selected.

3. **Implementation of different types of AI algorithms:** Different types of supervised AI algorithms will be used to predict client satisfaction (NPS) based on the best practices followed by software development teams as exogenous variables.

4. **Performance analysis and comparison:** The efficacy of different AI algorithms will be compared to determine the most suitable approach for the proposed problem.

A diagram illustrating this methodology can be found in Figure 1.

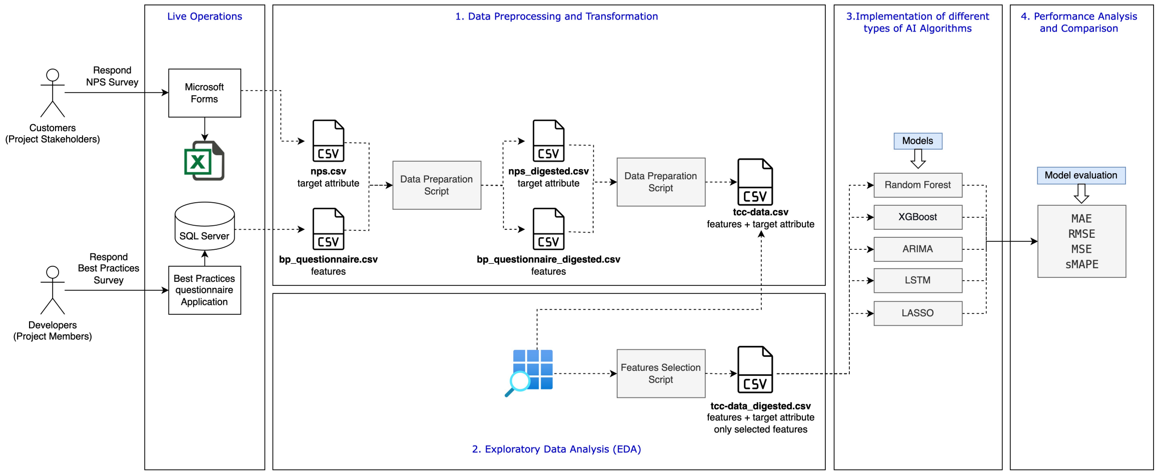

Live Operations
1. Data Preprocessing and Transformation
3.Implementation of different types of AI Algorithms
4. Performance Analysis and Comparison
Respond NPS Survey
Customers (Project Stakeholders)
Microsoft Forms
SQL Server
Respond Best Practices Survey
Developers (Project Members)
Best Practices questionnaire Application
CSV
nps.csv
target attribute
bp_questionnaire.csv
features
Data Preparation Script
nps_digested.csv
target attribute
bp_questionnaire_digested.csv
features
Data Preparation Script
tcc-data.csv
features + target attribute
Features Selection Script
tcc-data_digested.csv
features + target attribute
only selected features
2. Exploratory Data Analysis (EDA)
Models
Random Forest
XGBoost
ARIMA
LSTM
LASSO
Model evaluation
MAE
RMSE
MSE
sMAPE

Figure 1 – Research methodology followed in this study.

In the following sections, the details of the practical implementation of AI algorithms are explained and demonstrated using a Python Jupyter Notebook (Jupyter, 2011). The complete source code is available on the author's GitHub page (Bertonha, 2024).

3.3 Data preprocessing and transformation

The data for this study comes from two sources that need to be combined: best practices survey data (subsection 3.3.1) and NPS data (subsection 3.3.2). Both sources contain sensitive information that must be anonymized. In this study, a third data source will be created by combining both datasets (subsection 3.3.3).

This section explains the initial steps for data processing and transformation. It describes how the data was prepared to be ready for the next stages of the study.

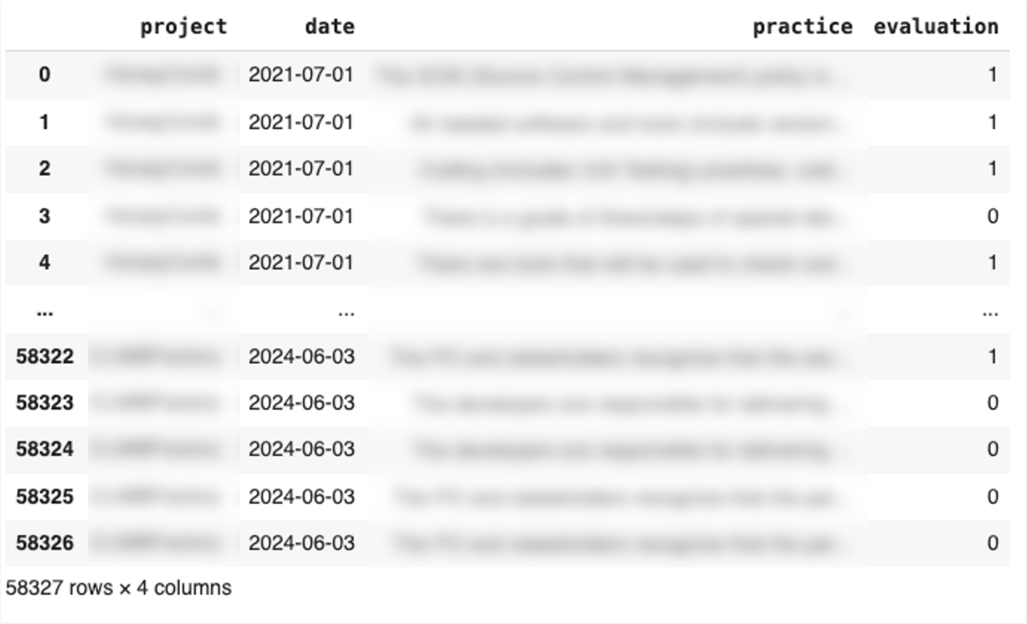
3.3.1 Best practices survey dataset preparation.

The best practices survey data is available in a CSV file exported from an SQL Server database (Microsoft, 2024b). The data structure is as follows:

project | date | practice | evaluation

- **project:** The project name (sensitive information, to be anonymized).
- **date:** The date the survey was responded to by project members.
- **practice:** Contain the statement of one best practice. These statements are considered sensitive information, so the exact content of the statements will not be disclosed in this study.
- **evaluation:** Contain the answer recorded as "1" (yes) or "0" (no).

The dataframe is shown in Figure 2. Please note that in the following figures, the project names and practice statements have been blurred due to their private and sensitive nature.



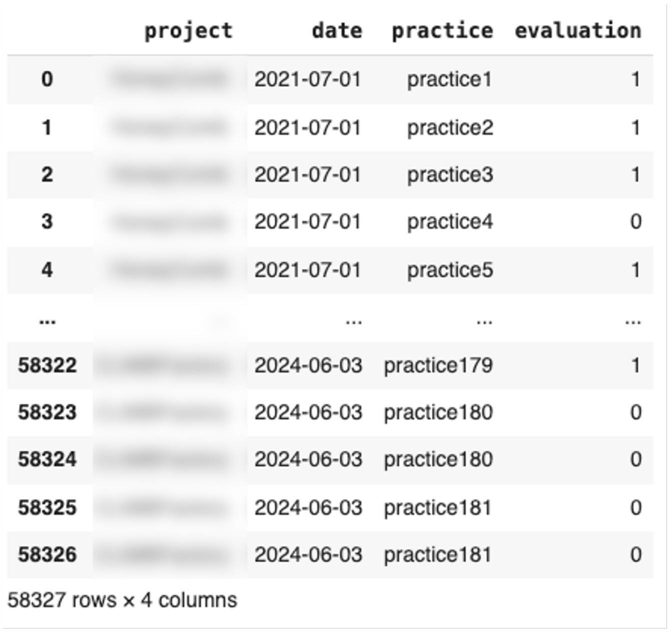
	project	date	practice	evaluation
0	[redacted]	2021-07-01	[redacted]	1
1	[redacted]	2021-07-01	[redacted]	1
2	[redacted]	2021-07-01	[redacted]	1
3	[redacted]	2021-07-01	[redacted]	0
4	[redacted]	2021-07-01	[redacted]	1
...	...	...	...	...
58322	[redacted]	2024-06-03	[redacted]	1
58323	[redacted]	2024-06-03	[redacted]	0
58324	[redacted]	2024-06-03	[redacted]	0
58325	[redacted]	2024-06-03	[redacted]	0
58326	[redacted]	2024-06-03	[redacted]	0

58327 rows × 4 columns

Figure 2 – Best practices survey dataframe, constructed from a CSV file exported from an SQL Server database.

The practice statements are replaced by generic identifiers: "practice1", "practice2", ..., "practiceN". The resulting dataframe is shown in Figure 3.

The data source also contains some duplicated information because some teams may have responded multiple times within the same month. To keep the data consistent,



	project	date	practice	evaluation
0	[redacted]	2021-07-01	practice1	1
1	[redacted]	2021-07-01	practice2	1
2	[redacted]	2021-07-01	practice3	1
3	[redacted]	2021-07-01	practice4	0
4	[redacted]	2021-07-01	practice5	1
...	...	...	...	...
58322	[redacted]	2024-06-03	practice179	1
58323	[redacted]	2024-06-03	practice180	0
58324	[redacted]	2024-06-03	practice180	0
58325	[redacted]	2024-06-03	practice181	0
58326	[redacted]	2024-06-03	practice181	0

58327 rows × 4 columns

Figure 3 – Best practices survey dataframe with anonymized practices.

only one response per month will be considered. The total row count is reduced after the deletion:

Before deletion: 58327

After deletion: 49058

As the next step, it is necessary to pivot the table so that the answers indicating whether the teams follow a certain practice become columns. This is a preparation step for the dataset to ensure it has the ideal shape for working with time series problems.

Current data format:

project | date | practice | evaluation

Target data format:

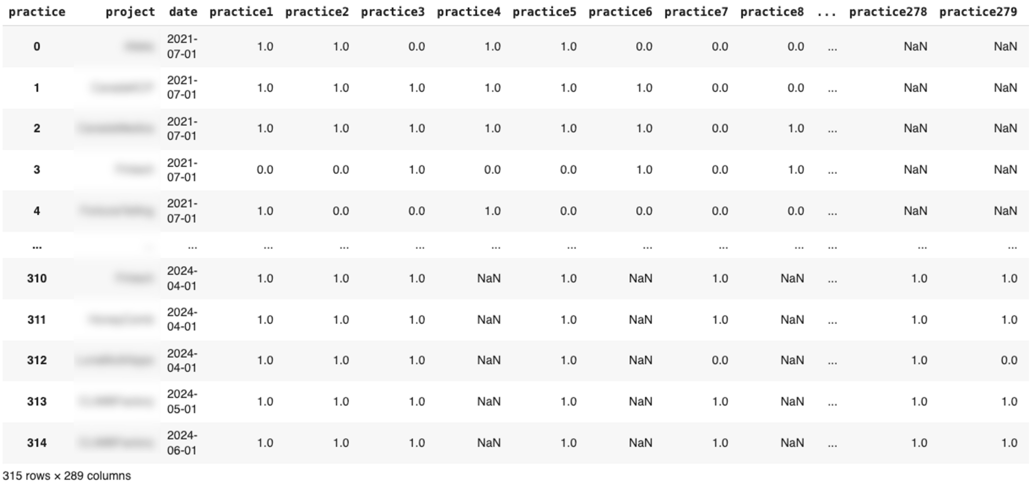
project | date | practice1 | practice2 | ... | practiceN

The resulting dataframe after the pivot operation is shown in Figure 4.

The dataset is now compatible with the expected shape for time series problems. However, there are still some `NaN` values because not all teams answered all the questions. In this study, all unanswered survey questions will be treated as "No" (binary, 0).

3.3.2 NPS survey dataset preparation

The NPS information was collected using Microsoft Forms (Microsoft, 2024a). Microsoft Forms has a feature that allows exporting all the responses as an Excel file, which can then be saved as a CSV file.



practice	project	date	practice1	practice2	practice3	practice4	practice5	practice6	practice7	practice8	...	practice278	practice279
0	[redacted]	2021-07-01	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	...	NaN	NaN
1	[redacted]	2021-07-01	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	...	NaN	NaN
2	[redacted]	2021-07-01	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	...	NaN	NaN
3	[redacted]	2021-07-01	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	...	NaN	NaN
4	[redacted]	2021-07-01	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	NaN	NaN
...	[redacted]	...	...	...	...	...	...	...	...	...	...	...	...
310	[redacted]	2024-04-01	1.0	1.0	1.0	NaN	1.0	NaN	1.0	NaN	...	1.0	1.0
311	[redacted]	2024-04-01	1.0	1.0	1.0	NaN	1.0	NaN	1.0	NaN	...	1.0	1.0
312	[redacted]	2024-04-01	1.0	1.0	1.0	NaN	1.0	NaN	0.0	NaN	...	1.0	0.0
313	[redacted]	2024-05-01	1.0	1.0	1.0	NaN	1.0	NaN	1.0	NaN	...	1.0	1.0
314	[redacted]	2024-06-01	1.0	1.0	1.0	NaN	1.0	NaN	1.0	NaN	...	1.0	1.0

315 rows × 289 columns

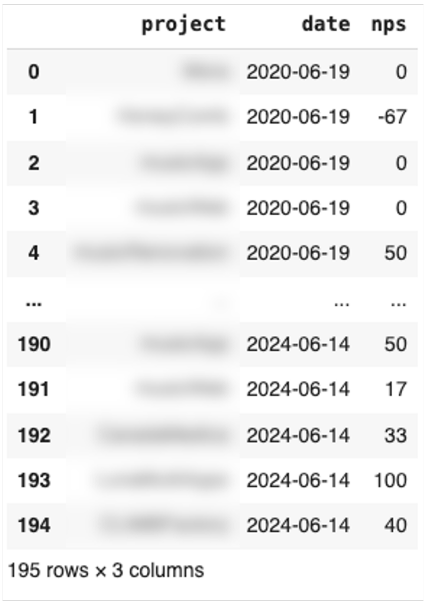
Figure 4 – Best practices survey dataframe with pivoted table.

The data is structured as follows:

project | date | nps

- **project:** The project name (sensitive information, to be anonymized).
- **date:** The date the NPS survey was responded to.
- **nps:** The Net Promoter Score for the project on the given date, ranging from -100 to +100.

The dataframe with the data structure described above is shown in Figure 5.



	project	date	nps
0	[redacted]	2020-06-19	0
1	[redacted]	2020-06-19	-67
2	[redacted]	2020-06-19	0
3	[redacted]	2020-06-19	0
4	[redacted]	2020-06-19	50
...	[redacted]	...	...
190	[redacted]	2024-06-14	50
191	[redacted]	2024-06-14	17
192	[redacted]	2024-06-14	33
193	[redacted]	2024-06-14	100
194	[redacted]	2024-06-14	40

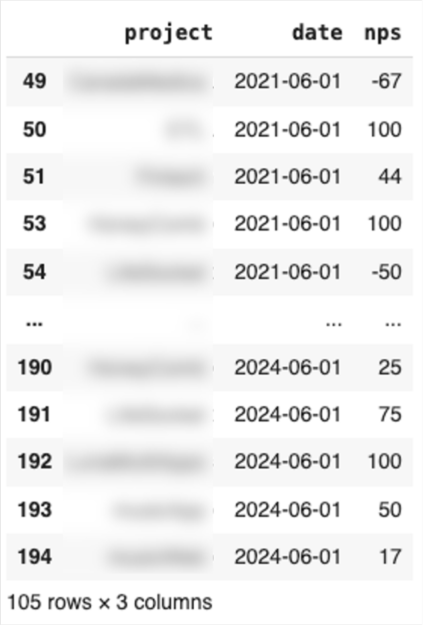
195 rows × 3 columns

Figure 5 – NPS dataframe, constructed from a CSV file obtained from Microsoft Forms.

The dataframe contains duplicate entries because some teams may have submitted the NPS survey multiple times within the same month. In this study, only one response per month will be considered as the time series frequency. The dataframe is cleaned to ensure that only one entry exists for each month.

Another point to be considered is the start date and end date of the time series to be used in this study. Although the NPS survey has been conducted since early 2020, the best practices survey only started in mid-2021. Based on that, this study will only consider NPS data from mid-2021 onwards. The NPS measurements prior to June 2021 will be deleted.

Additionally, some projects started and ended in a very short period, having only a few entries in the survey. In this study, the data for these short-term projects will not be considered and will be removed from the dataframe. The resulting dataframe after these cleaning operations is shown in Figure 6.



	project	date	nps
49	[redacted]	2021-06-01	-67
50	[redacted]	2021-06-01	100
51	[redacted]	2021-06-01	44
53	[redacted]	2021-06-01	100
54	[redacted]	2021-06-01	-50
...	[redacted]	...	...
190	[redacted]	2024-06-01	25
191	[redacted]	2024-06-01	75
192	[redacted]	2024-06-01	100
193	[redacted]	2024-06-01	50
194	[redacted]	2024-06-01	17

105 rows × 3 columns

Figure 6 – Net promoter score dataframe cleaned up.

When plotting the NPS time series on a chart, it is possible to visualize how it changes over time in Figure 7. It is also apparent that some time series are incomplete.

The NPS measurements are taken every 3 months, while the best practices survey is conducted at least once a month. To align the timing between these two data sources, the time series needs to be adjusted to a monthly frequency. This adjustment will create gaps in the NPS data, leading to `null` values. To fill these gaps, in this study the `DataFrame.interpolate` method is used, which estimates the missing values by creating a linear progression between the known data points before and after the gaps. This process is illustrated in Figure 8.

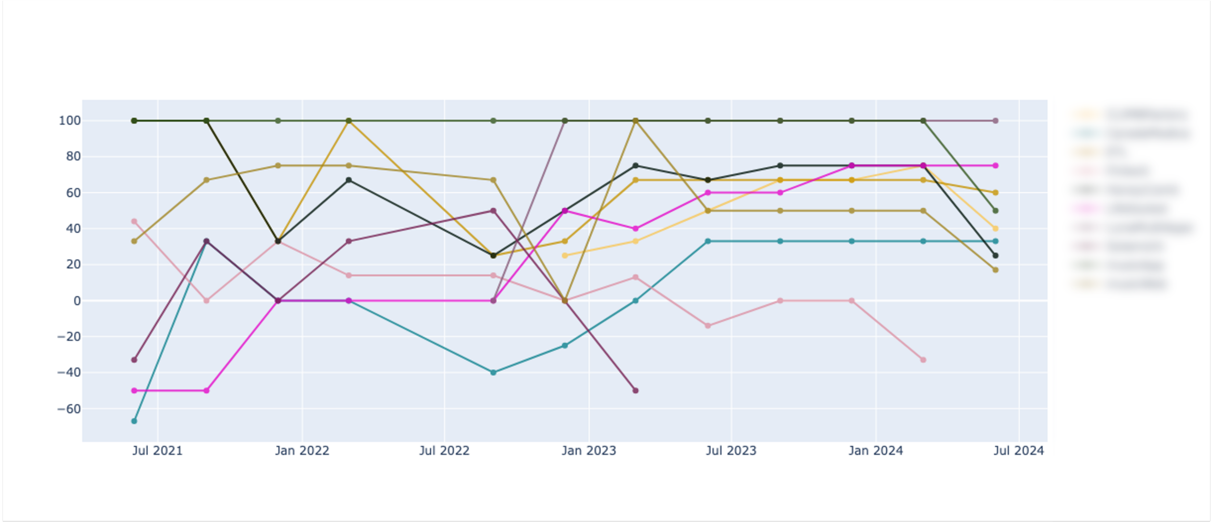


Figure 7 – Net promoter score time series. Some are incomplete.

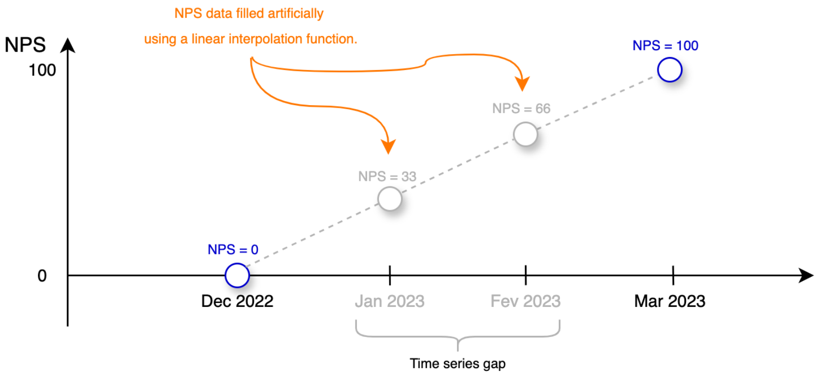


Figure 8 – Example: approach used to fill the gaps in the time series.

By plotting the NPS time series on a chart again, it is possible to visualize in Figure 9 how it changes over time. The incomplete time series are no longer present.

3.3.3 Combination of best practices survey and NPS data into a single dataset

The best practices survey data will be used as exogenous features, while the NPS data is the target field to be predicted. For that reason, the NPS and best practices survey data need to be combined into a single CSV file. This combined file will be imported in a dataframe and used in the next stage of the study. The combined dataset is obtained by merging the NPS and best practices digested datasets. Even after this operation, there will still be some project evaluations missing for some months in the time series. The missing data was filled using the `bfill` and `ffill` methods. The `bfill` method fills missing values with the first value found that comes after the missing value in the time series, while the `ffill` method fills missing values with the last value found before the missing value in the time series.

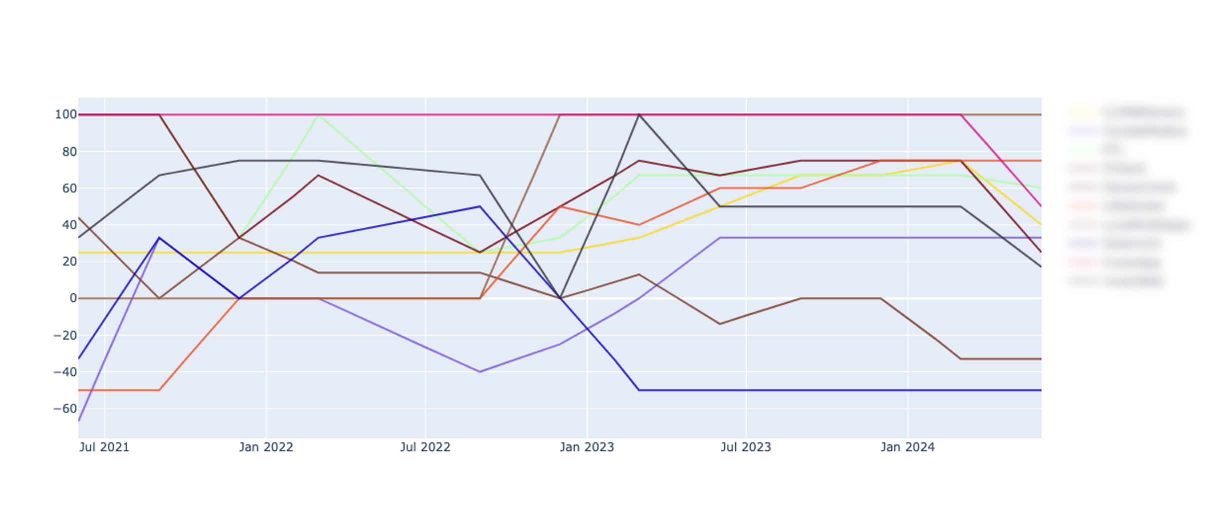
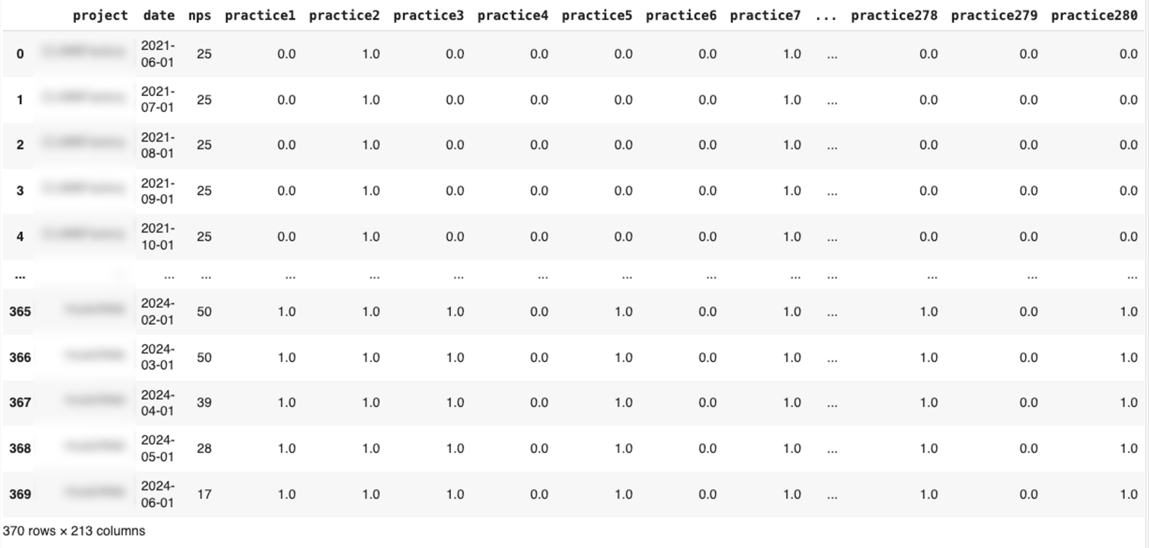


Figure 9 – NPS time series. The incomplete time series are no longer present.

In the next steps, the columns that contain only "0" are deleted, as they are not useful as features for training purposes. The dataframe containing both the NPS and best practices survey data is now structured as follows:

project | date | nps | practice1 | practice2 | practice3 | ... | practiceN

The resulting dataframe is shown in Figure 10.

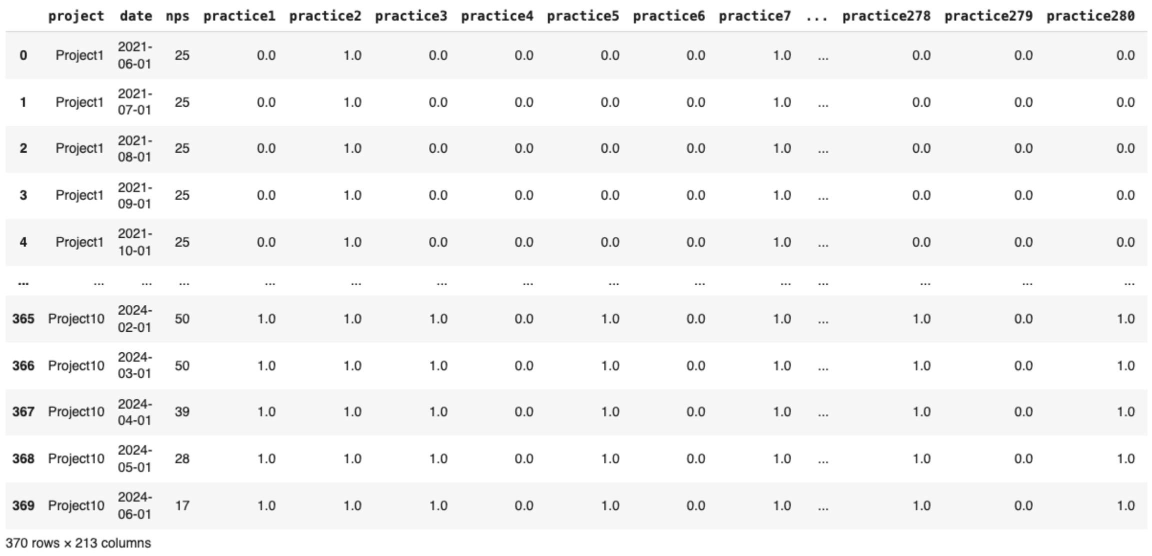


	project	date	nps	practice1	practice2	practice3	practice4	practice5	practice6	practice7	...	practice278	practice279	practice280
0	[redacted]	2021-06-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
1	[redacted]	2021-07-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
2	[redacted]	2021-08-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
3	[redacted]	2021-09-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
4	[redacted]	2021-10-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
...	[redacted]	...	...	...	...	...	...	...	...	...	...	...	...	...
365	[redacted]	2024-02-01	50	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
366	[redacted]	2024-03-01	50	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
367	[redacted]	2024-04-01	39	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
368	[redacted]	2024-05-01	28	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
369	[redacted]	2024-06-01	17	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0

370 rows × 213 columns

Figure 10 – Dataframe containing NPS and best practices survey data.

The project names are also considered sensitive information and need to be anonymized. The anonymization process followed the same approach used for the practices in subsection 3.3.1. The real project names are replaced by identifiers following the pattern "Project1", "Project2", ..., "ProjectN". The resulting dataframe after the anonymization process can be observed in Figure 11.



	project	date	nps	practice1	practice2	practice3	practice4	practice5	practice6	practice7	...	practice278	practice279	practice280
0	Project1	2021-06-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
1	Project1	2021-07-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
2	Project1	2021-08-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
3	Project1	2021-09-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
4	Project1	2021-10-01	25	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
365	Project10	2024-02-01	50	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
366	Project10	2024-03-01	50	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
367	Project10	2024-04-01	39	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
368	Project10	2024-05-01	28	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0
369	Project10	2024-06-01	17	1.0	1.0	1.0	0.0	1.0	0.0	1.0	...	1.0	0.0	1.0

370 rows × 213 columns

Figure 11 – Dataframe containing NPS and best practices survey data, with anonymized project names.

3.4 Exploratory data analysis (EDA)

During the exploratory data analysis, various charts are used to examine the dataframe. These visualizations are important for understanding the composition of the NPS and features. By analyzing the data, key insights can be gained, guiding the selection of the most relevant features for this study based on the patterns and relationships observed.

3.4.1 NPS time series analysis

A chart showing the NPS scores in chronological order is plotted. This helps to visualize how the NPS changes over time for each project. The results are shown in the Figure 12.

To analyze the distribution of NPS values, the histogram chart can be used. This type of chart allows for a visual representation of how NPS scores are spread across different ranges, helping to identify trends in the data. The histogram chart can be visualized in Figure 13

Box plot charts are also used to understand the distribution of NPS for each project by month. The boxplot chart in Figure 14 reveals that the medians show an increasing trend until around March 2024, after which it starts to decline. Additionally, the 25th percentile is concentrated in November and December 2022. From May 2023 onwards, there is an outlier with a negative NPS value.

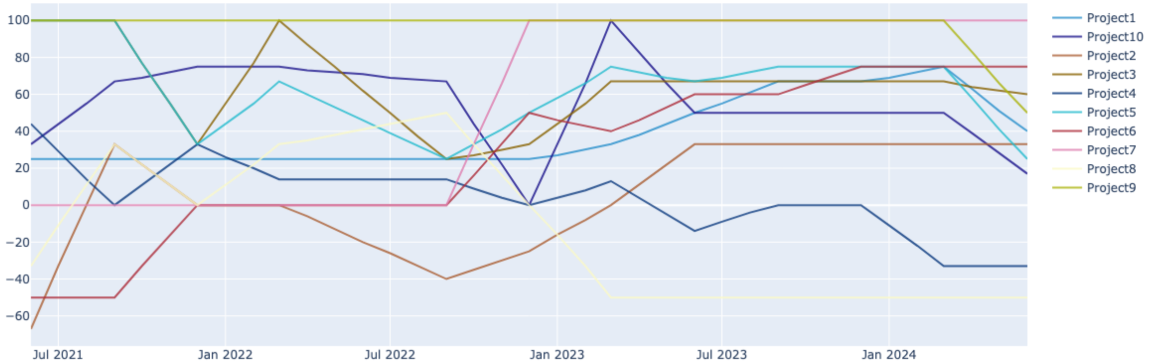


Figure 12 – Time series showing how the NPS score evolves over time.

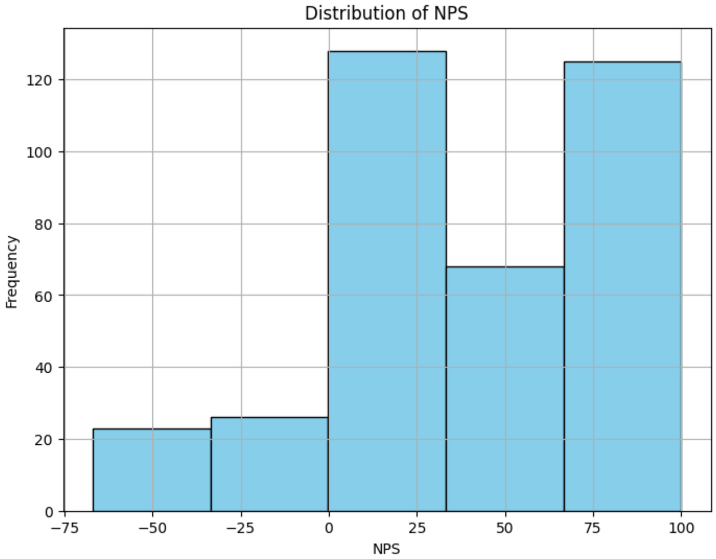


Figure 13 – Histogram chart showing the distribution of NPS.

3.4.2 Exogenous features analysis and selection

Another analysis was performed to examine the correlation between the features (best practices survey responses) and the NPS score. This step will guide the next part of the study, ensuring that the features that appear to have the strongest correlation with the NPS score are selected for use in the implementation of AI algorithms. The result of this analysis is shown in the picture Figure 15

Pearson Correlation is a statistical method that quantifies the relationship between two data sets by comparing their attributes and producing a score ranging from -1 to +1 (Yan, 2022). Pearson Correlation formula is given by:

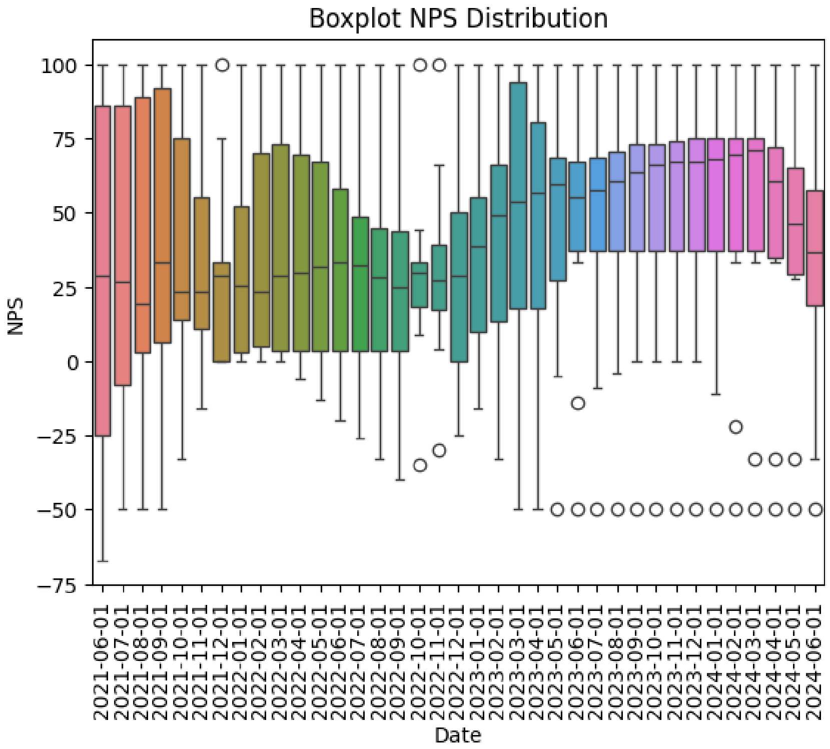


Figure 14 – Boxplot NPS distribution over months.

$$\mathrm{R}(y,\hat{y}) = \frac{\sum_{i=0}^{N-1}((y_i - mean(y)) * (\hat{y}_i - mean(\hat{y})))}{\sqrt{\sum_{i=0}^{N-1}(y_i - mean(y))^2} * \sqrt{\sum_{i=0}^{N-1}(\hat{y}_i - mean(\hat{y}))^2}}$$

By analysing the correlation heatmap matrix using Pearson Correlation, it is possible to see that some best practices are highly correlated between eachother.

The most important information is the correlation between the target attribute to be predicted (NPS) and the other exogenous features (practices), which can be observed in the first column and in the first row of the matrix.

The dataset contains many features, and to enhance its relevance, it's important to select those that contribute most effectively to the outcome. Correlation-based Feature Selection (CFS) is a technique designed to identify subsets of features that show high correlation with the target variable while maintaining low inter-correlation among themselves. The purpose of this approach is to select a feature subset that offers the most valuable information about the target variable while reducing redundancy (Gopika; M.E., 2018).

The features that are most correlated with the NPS are ranked in the chart in Figure 16. A new heatmap matrix containing only the highly correlated features is displayed in a chart in Figure 17.

The next step is to select among those features the ones that are less correlated with each other. Only the features with a correlation lower than 0.6 are selected. The

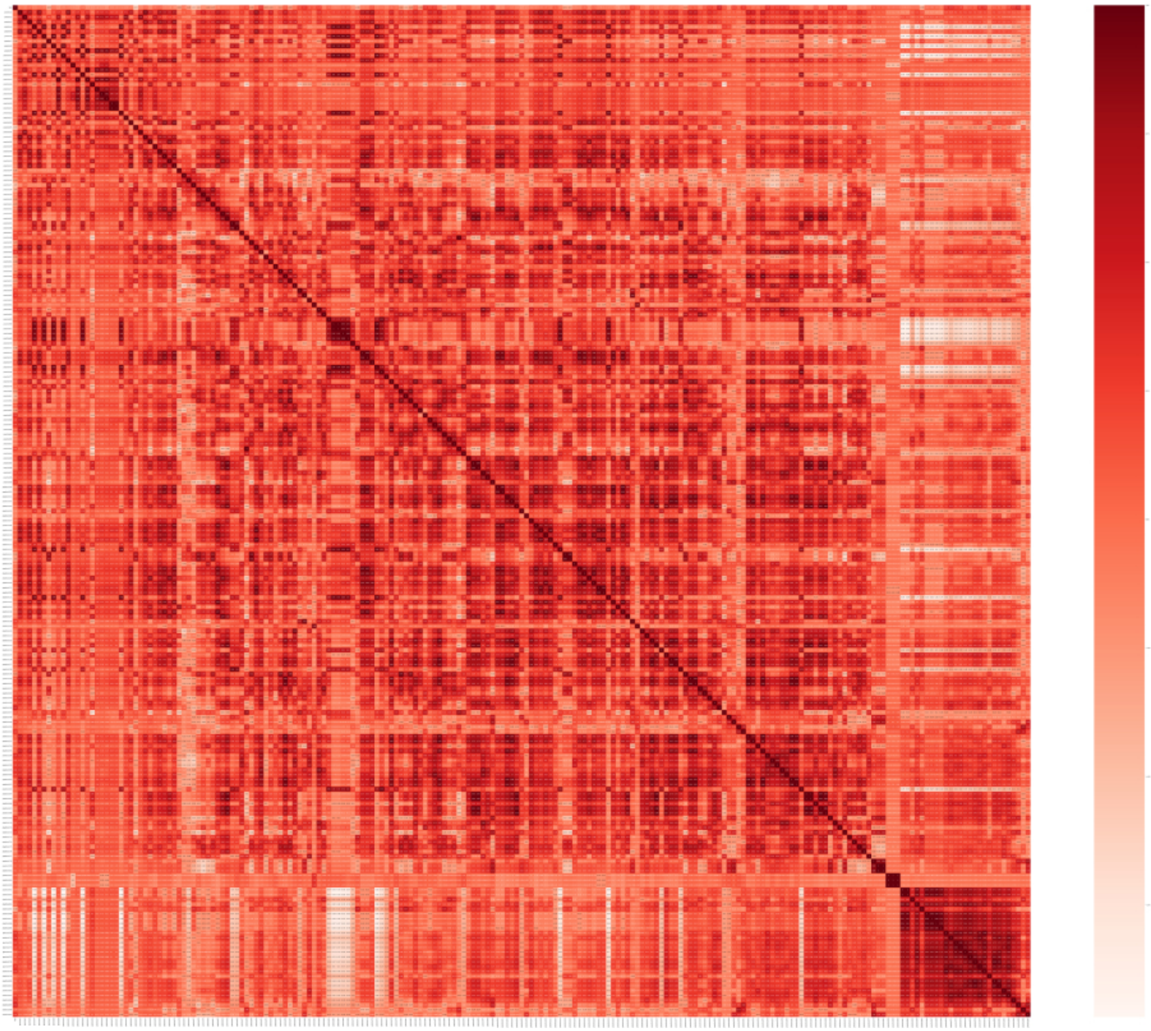


Figure 15 – Heatmap matrix using Pearson correlation to calculate the correlation between each feature.

resulting dataframe after the initial feature selection is shown in Figure 18.

Calculating the Variance Inflation Factor (VIF) helps to address multicollinearity among features. VIF measures the severity of multicollinearity in regression analysis (Akinwande; Dikko; Samson, 2015). The VIF is computed for the dataframe to identify and remove rank-deficient features. As shown in Figure 19, "practice37" has a VIF greater than 5 and will therefore be removed from the dataframe.

Another exploratory analysis involves visualizing the total count of 'Yes' (1) and 'No' (0) responses in the best practices survey. The chart is shown in Figure 20.

To ensure better feature variability and avoid rank-deficient features, this study will remove features that have fewer than 50 samples with a value of 1. The final dataframe used for model training is shown in Figure 21.

The final practices selected as exogenous features for implementing the AI algorithms in the next section are: "practice25", "practice146", "practice81", and "practice271".

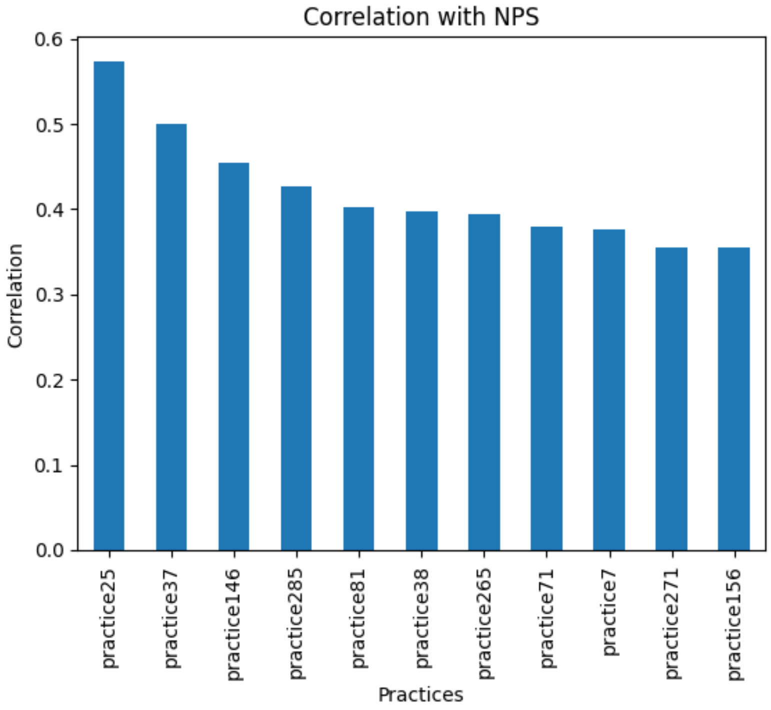
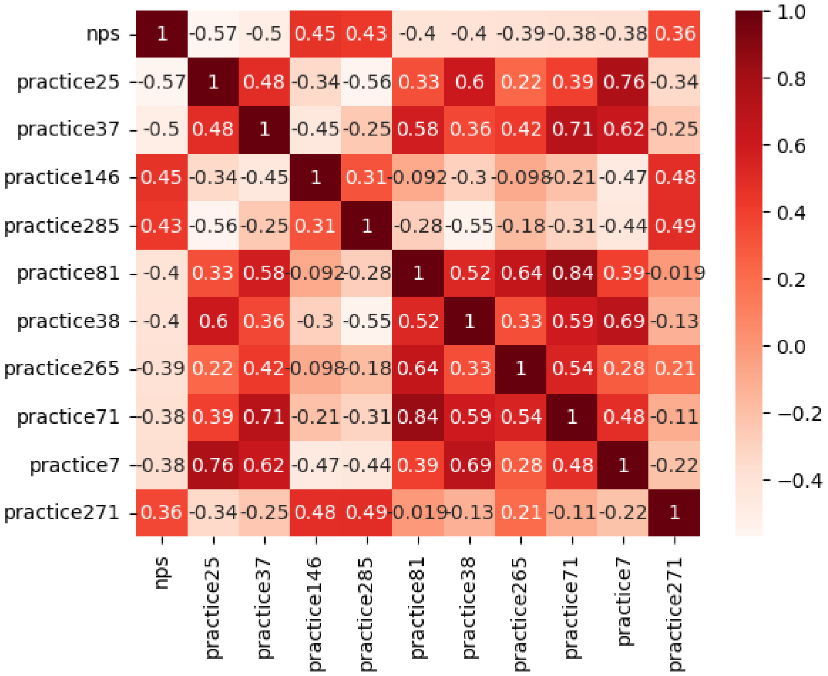


Figure 16 – A chart displaying the features that are highly correlated with the NPS.

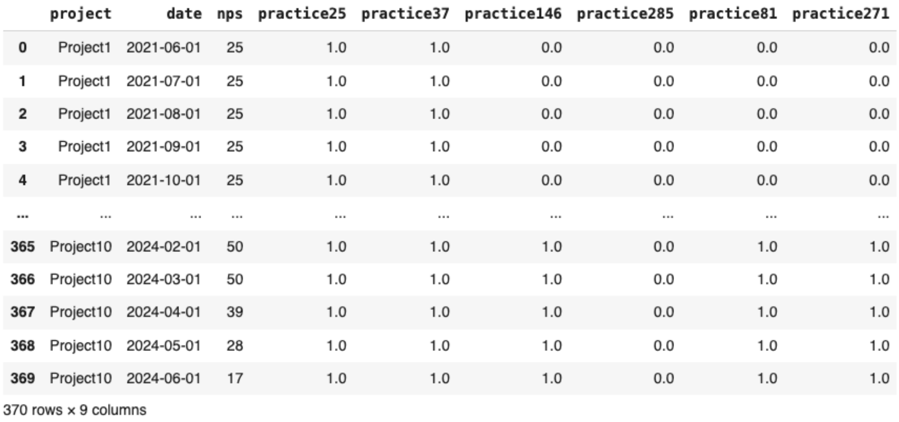


	nps	practice25	practice37	practice146	practice285	practice81	practice38	practice265	practice71	practice7	practice271
nps	1	-0.57	-0.5	0.45	0.43	-0.4	-0.4	-0.39	-0.38	-0.38	0.36
practice25	-0.57	1	0.48	-0.34	-0.56	0.33	0.6	0.22	0.39	0.76	-0.34
practice37	-0.5	0.48	1	-0.45	-0.25	0.58	0.36	0.42	0.71	0.62	-0.25
practice146	0.45	-0.34	-0.45	1	0.31	-0.092	-0.3	-0.098	-0.21	-0.47	0.48
practice285	0.43	-0.56	-0.25	0.31	1	-0.28	-0.55	-0.18	-0.31	-0.44	0.49
practice81	-0.4	0.33	0.58	-0.092	-0.28	1	0.52	0.64	0.84	0.39	-0.019
practice38	-0.4	0.6	0.36	-0.3	-0.55	0.52	1	0.33	0.59	0.69	-0.13
practice265	-0.39	0.22	0.42	-0.098	-0.18	0.64	0.33	1	0.54	0.28	0.21
practice71	-0.38	0.39	0.71	-0.21	-0.31	0.84	0.59	0.54	1	0.48	-0.11
practice7	-0.38	0.76	0.62	-0.47	-0.44	0.39	0.69	0.28	0.48	1	-0.22
practice271	0.36	-0.34	-0.25	0.48	0.49	-0.019	-0.13	0.21	-0.11	-0.22	1

Figure 17 – Correlation heatmap matrix considering the relevant features.

3.5 Implementation of AI algorithms

In the Chapter 2, the AI algorithms chosen for this study were introduced. In the section 3.3 and section 3.4, the dataset was prepared and as a result, a combined dataset including best practices survey responses and net promoter scores is ready for use with AI



	project	date	nps	practice25	practice37	practice146	practice285	practice81	practice271
0	Project1	2021-06-01	25	1.0	1.0	0.0	0.0	0.0	0.0
1	Project1	2021-07-01	25	1.0	1.0	0.0	0.0	0.0	0.0
2	Project1	2021-08-01	25	1.0	1.0	0.0	0.0	0.0	0.0
3	Project1	2021-09-01	25	1.0	1.0	0.0	0.0	0.0	0.0
4	Project1	2021-10-01	25	1.0	1.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
365	Project10	2024-02-01	50	1.0	1.0	1.0	0.0	1.0	1.0
366	Project10	2024-03-01	50	1.0	1.0	1.0	0.0	1.0	1.0
367	Project10	2024-04-01	39	1.0	1.0	1.0	0.0	1.0	1.0
368	Project10	2024-05-01	28	1.0	1.0	1.0	0.0	1.0	1.0
369	Project10	2024-06-01	17	1.0	1.0	1.0	0.0	1.0	1.0

370 rows × 9 columns

Figure 18 – Resulting dataframe containing only the selected features with a correlation of 0.6 or lower.

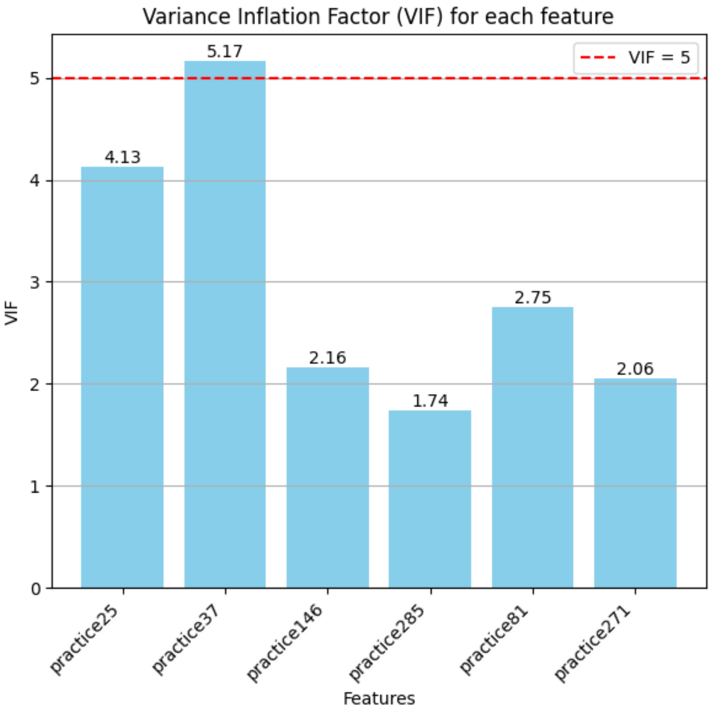


Figure 19 – Variance Inflation Factor for each feature. The feature "practice37" will be removed because it has a VIF greater than 5.

algorithms. This section explains how the AI algorithms were implemented in practice based on the processed dataset.

The framework used in this study requires the target attribute (nps), date (date), and identifiers (project) to be renamed as `y`, `ds`, and `unique_id`, respectively. The Figure 22 shows the resulting dataframe with the renamed columns.

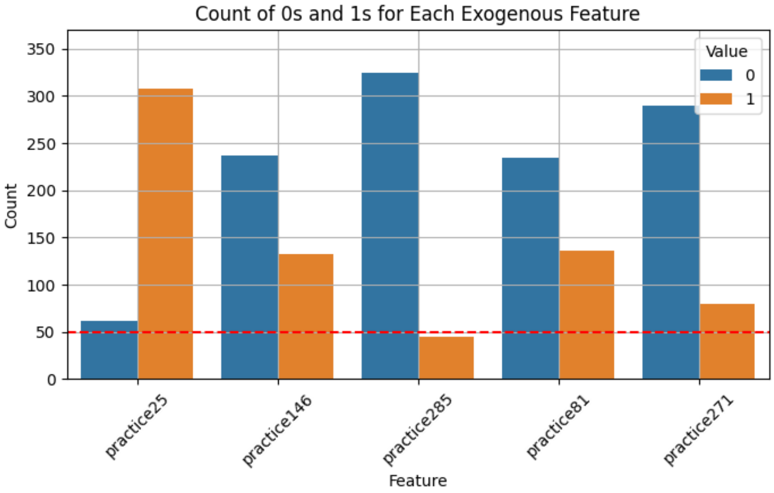
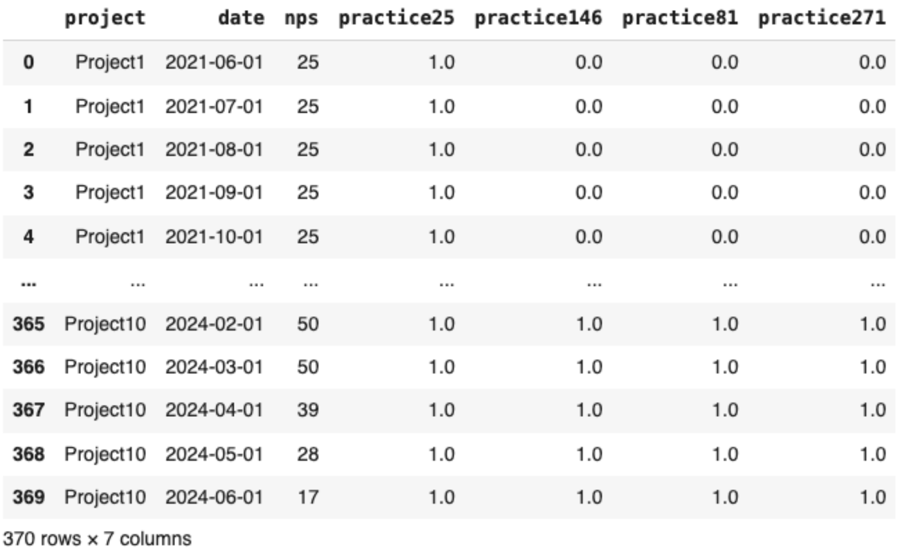


Figure 20 – Count of 0s and 1s for each exogenous feature.



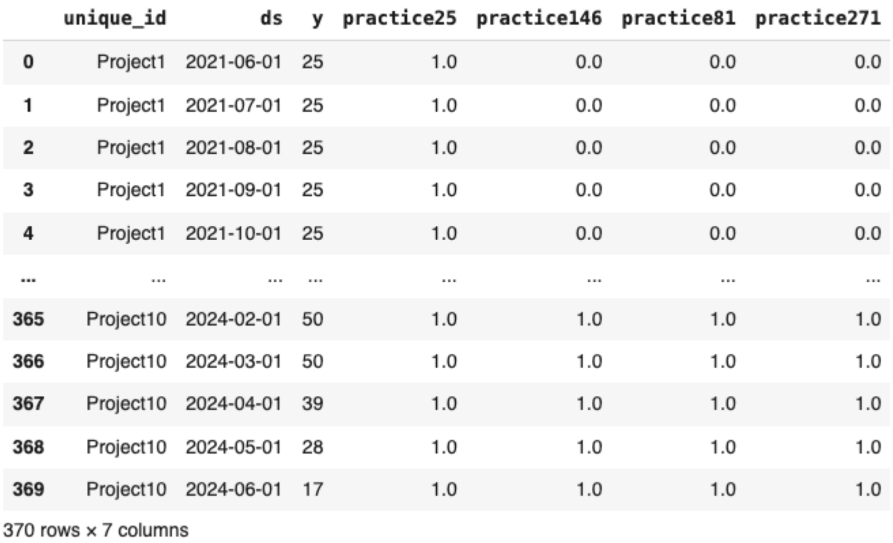
	project	date	nps	practice25	practice146	practice81	practice271
0	Project1	2021-06-01	25	1.0	0.0	0.0	0.0
1	Project1	2021-07-01	25	1.0	0.0	0.0	0.0
2	Project1	2021-08-01	25	1.0	0.0	0.0	0.0
3	Project1	2021-09-01	25	1.0	0.0	0.0	0.0
4	Project1	2021-10-01	25	1.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...
365	Project10	2024-02-01	50	1.0	1.0	1.0	1.0
366	Project10	2024-03-01	50	1.0	1.0	1.0	1.0
367	Project10	2024-04-01	39	1.0	1.0	1.0	1.0
368	Project10	2024-05-01	28	1.0	1.0	1.0	1.0
369	Project10	2024-06-01	17	1.0	1.0	1.0	1.0

370 rows × 7 columns

Figure 21 – Final resulting dataframe with selected features.

3.5.1 Splitting the dataframe into training and testing sets

The time series contains data from June 2021 to June 2024, with a total of 370 rows. In a time series problem, the dataset is typically split based on a cutoff date. The portion of the time series before the cutoff date forms the training set, while the portion after the cutoff date forms the testing set. In this study, November 2023 will be used as the cutoff date for the split. The larger part will be used for training, while the smaller part will be used to test the performance of the models. The resulting training and testing datasets contain 290 and 80 rows, respectively, representing approximately 78% and 22% of the data. The complete time series and its division into training and test sets can be visualized in the Figure 23



	unique_id	ds	y	practice25	practice146	practice81	practice271
0	Project1	2021-06-01	25	1.0	0.0	0.0	0.0
1	Project1	2021-07-01	25	1.0	0.0	0.0	0.0
2	Project1	2021-08-01	25	1.0	0.0	0.0	0.0
3	Project1	2021-09-01	25	1.0	0.0	0.0	0.0
4	Project1	2021-10-01	25	1.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...
365	Project10	2024-02-01	50	1.0	1.0	1.0	1.0
366	Project10	2024-03-01	50	1.0	1.0	1.0	1.0
367	Project10	2024-04-01	39	1.0	1.0	1.0	1.0
368	Project10	2024-05-01	28	1.0	1.0	1.0	1.0
369	Project10	2024-06-01	17	1.0	1.0	1.0	1.0

370 rows × 7 columns

Figure 22 – Final resulting dataframe with the renamed columns.

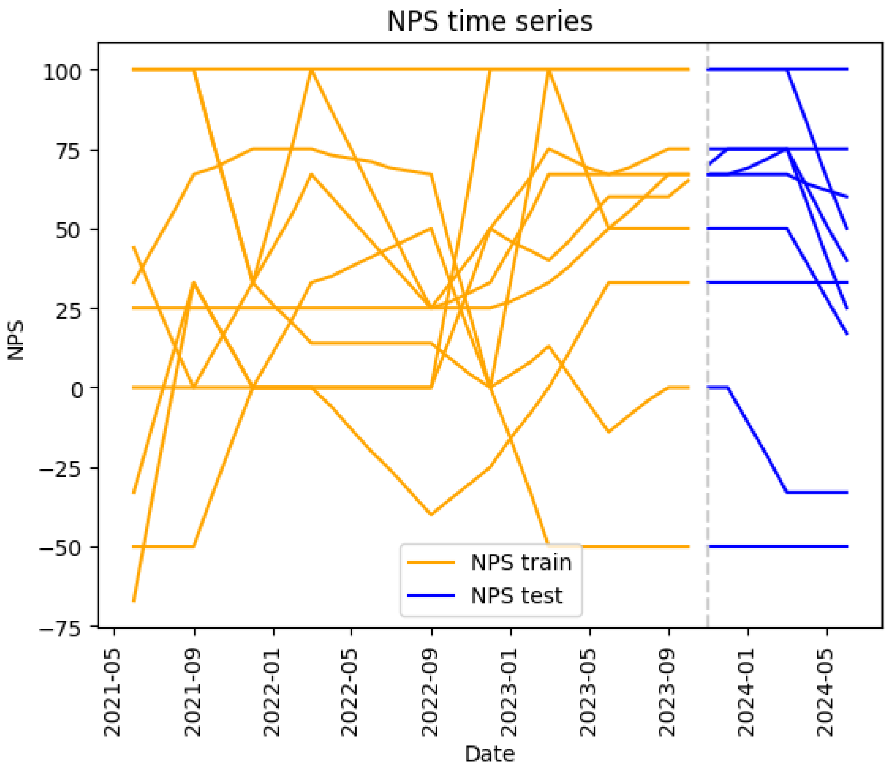


Figure 23 – Complete time series and its division into training and test.

3.5.2 Implementation

In this section, the AI algorithms introduced in section 2.2 are implemented using Python libraries.

Regressor	Parameter	Description	Value
Random Forest	random_state	Random number seed.	0
	n_estimators	Number of trees in the forest.	100
Extreme Gradient Boosting	random_state	Random number seed.	0
	n_estimators	Number of gradient boosting trees in the forest.	100
LASSO	alpha	Constant that multiplies the L1 term, controlling regularization strength.	0.1
	max_iter	The maximum number of iterations.	10000

Table 1 – Regressors parameters used in this study.

3.5.2.1 Random Forest (RF), Extreme Gradient Boosting (XGBoost) and Least Absolute Shrinkage and Selection Operator (LASSO):

In this experiment, a framework called `mlforecast` will be used.

`mlforecast` is a framework designed for time series forecasting with machine learning models, allowing scalability to handle large datasets (mlforecast, 2024). The `mlforecast` framework supports regressors that follow the scikit-learn API (learn, 2024). In the next part, three AI algorithms based on the scikit-learn API will be implemented: Random Forest, XGBoost, and LASSO. The relevant parameters for each model, along with the values that demonstrated the best results in this study, are described in the Table 1.

In addition to the pipeline of models, the `mlforecast` library offers several parameters that simplify working with time series. The library automatically considers the `unique_id` column as the identifier field to distinguish the multiple time series, the `ds` column as the date of the measurement, and the `y` column as the target value to be predicted. The details about the parameters are described in Table 2.

The results are saved in a dataframe, which includes columns with predictions from RF, XGB, and LASSO, along with the actual values (`"y"` column).

3.5.2.2 Autoregressive Integrated Moving Average (ARIMA)

In this study, a framework called `StatsForecast` was used. `StatsForecast` offers a collection of widely used univariate time series forecasting models, including AutoARIMA (NIXTLA, 2024b). In the AutoARIMA implementation, the best ARIMA model is automatically selected using an information criterion.

In this experiment, the default parameters of AutoARIMA were used without any customizations.

Parameter	**Type**	**Default**	**Details**	**Value used in this study**
models	Union		Models that will be trained and used to compute the forecasts.	Pipeline of RF, XGB, Lasso regressors.
freq	Union		Pandas offset, pandas offset alias, e.g. 'D', 'W-THU' or integer denoting the frequency of the series.	'MS' (Monthly data)
lags	Optional	None	Lags of the target to use as features.	[1,2] (Use last 2 measurements as features)
lag_transforms	Optional	None	Mapping of target lags to their transformations.	
date_features	Optional	None	Features computed from the dates. Can be pandas date attributes or functions that will take the dates as input.	[]
num_threads	int	1	Number of threads to use when computing the features.	
target_transforms	Optional	None	Transformations that will be applied to the target before computing the features and restored after the forecasting step.	
lag_transforms_namer	Optional	None	Function that takes a transformation (either function or class), a lag and extra arguments and produces a name.	

Table 2 – Parameters from the mlforescast framework.

The results are saved in a dataframe, which includes columns with predictions from RF, XGB, LASSO, and AutoARIMA, along with the actual values ("y" column).

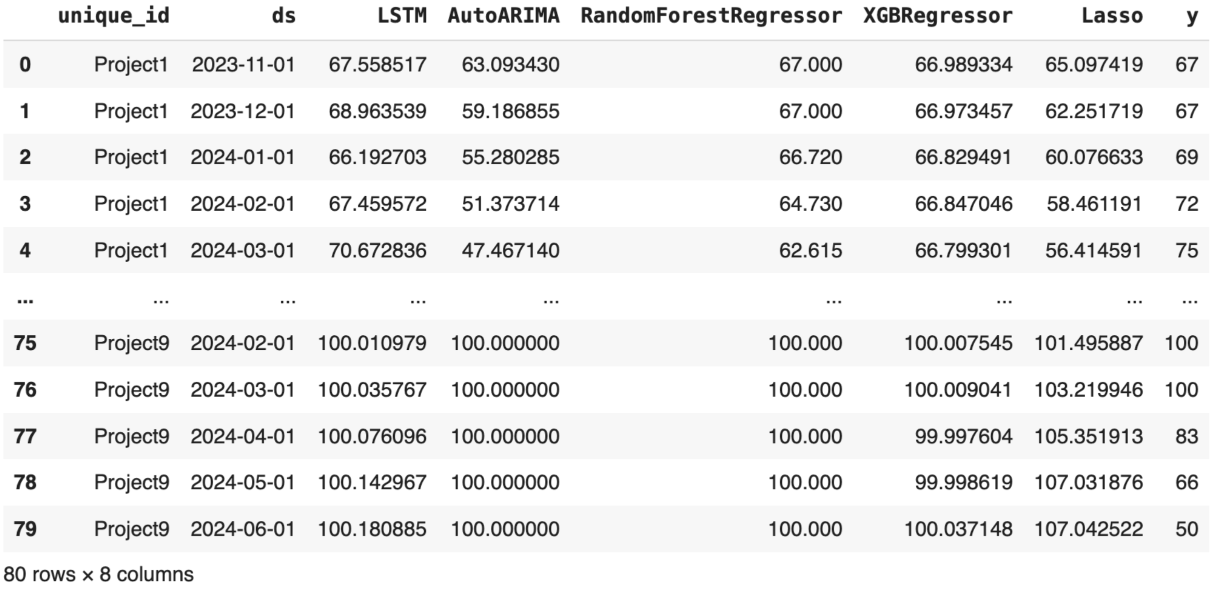
3.5.2.3 Long Short-Term Memory (LSTM)

The final model implemented in this study is LSTM. A framework called `NeuralForecast` was used for this purpose. `NeuralForecast` offers a wide range of neural forecasting models, focusing on performance and usability. The models include classic networks like RNNs, as well as LSTM, which is used in this study (NIXTLA, 2024a). The parameters utilized in the LSTM model are described in Table 3. The parameters were chosen through a series of experiments in which each parameter was individually adjusted, either increased or decreased. The combination of parameters that produced the best results were then selected for the final model.

Parameter	**Type**	**Default**	**Details**	**Value used in this study**
h	int		Forecast horizon.	8 (length of the list used for test)
loss	PyTorch module		Instantiated train loss class from losses collection.	DistributionLoss(distribution= ’Normal’, level=[90])
max_steps	int	1000	Maximum number of training steps.	14500
encoder_-n_layers	int	2	Number of layers for the LSTM.	3
encoder_-hidden_-size	int	200	Units for the LSTM’s hidden state size.	100
context_-size	int	10	Size of context vector for each timestamp on the forecasting window.	1
decoder_-hidden_-size	int	200	Size of hidden layer for the MLP decoder.	100
decoder_-layers	int	2	Number of layers for the MLP decoder.	3
learning_-rate	float	1e-3	Learning rate between (0, 1).	1e-5
scaler_-type	string	‘robust’	Type of scaler for temporal inputs normalization see temporal scalers.	’standard’
futr_-exog_list	string list		Future exogenous columns.	relevant_columns (List of best practices)

Table 3 – Parameters utilized in the LSTM.

The dataframe containing the results of all the experiments is displayed in Figure 24. It includes columns with predictions from RF, XGB, LASSO, AutoARIMA, and LSTM, as well as the actual values ("y" column) from the test dataframe.



	unique_id	ds	LSTM	AutoARIMA	RandomForestRegressor	XGBRegressor	Lasso	y
0	Project1	2023-11-01	67.558517	63.093430	67.000	66.989334	65.097419	67
1	Project1	2023-12-01	68.963539	59.186855	67.000	66.973457	62.251719	67
2	Project1	2024-01-01	66.192703	55.280285	66.720	66.829491	60.076633	69
3	Project1	2024-02-01	67.459572	51.373714	64.730	66.847046	58.461191	72
4	Project1	2024-03-01	70.672836	47.467140	62.615	66.799301	56.414591	75
...	...	...	...	...	...	...	...	...
75	Project9	2024-02-01	100.010979	100.000000	100.000	100.007545	101.495887	100
76	Project9	2024-03-01	100.035767	100.000000	100.000	100.009041	103.219946	100
77	Project9	2024-04-01	100.076096	100.000000	100.000	99.997604	105.351913	83
78	Project9	2024-05-01	100.142967	100.000000	100.000	99.998619	107.031876	66
79	Project9	2024-06-01	100.180885	100.000000	100.000	100.037148	107.042522	50

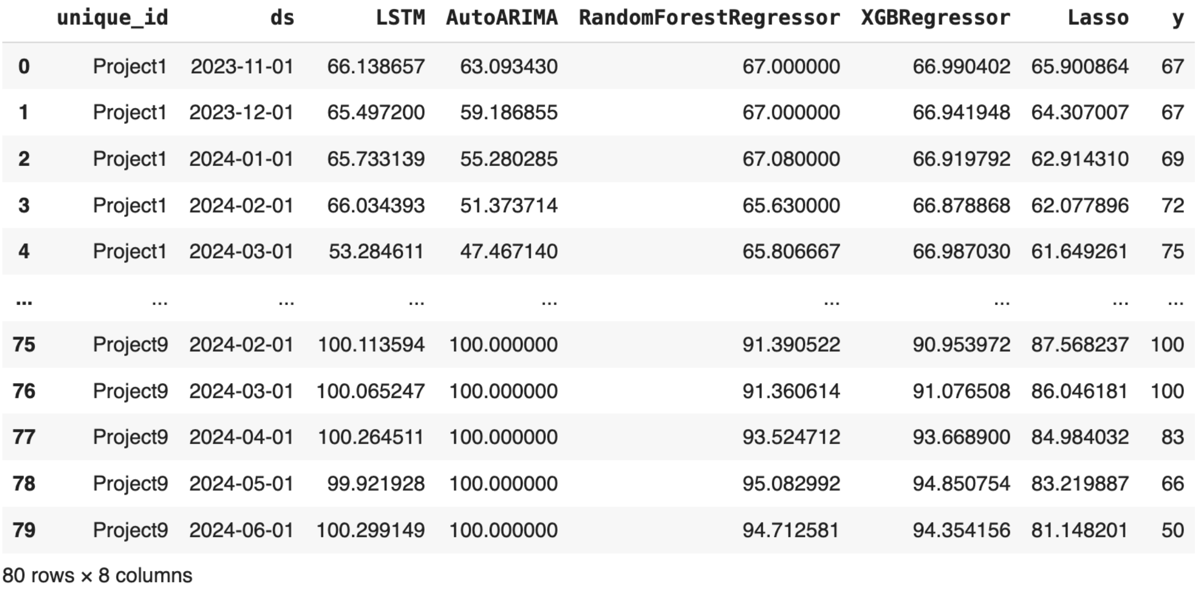
80 rows × 8 columns

Figure 24 – Dataframe containing predictions from all implemented models.

3.6 Repeating the experiment after removing the exogenous features

The same AI algorithms are retrained, but this time excluding the four practices from the dataframe that were used as exogenous features. The objective is to determine whether these exogenous features positively influence the performance of the AI models.

The resulting dataframe, shown in Figure 25, includes columns with predictions from RF, XGB, LASSO, AutoARIMA, and LSTM, as well as the actual values ("y" column) from the test dataframe.



	unique_id	ds	LSTM	AutoARIMA	RandomForestRegressor	XGBRegressor	Lasso	y
0	Project1	2023-11-01	66.138657	63.093430	67.000000	66.990402	65.900864	67
1	Project1	2023-12-01	65.497200	59.186855	67.000000	66.941948	64.307007	67
2	Project1	2024-01-01	65.733139	55.280285	67.080000	66.919792	62.914310	69
3	Project1	2024-02-01	66.034393	51.373714	65.630000	66.878868	62.077896	72
4	Project1	2024-03-01	53.284611	47.467140	65.806667	66.987030	61.649261	75
...	...	...	...	...	...	...	...	...
75	Project9	2024-02-01	100.113594	100.000000	91.390522	90.953972	87.568237	100
76	Project9	2024-03-01	100.065247	100.000000	91.360614	91.076508	86.046181	100
77	Project9	2024-04-01	100.264511	100.000000	93.524712	93.668900	84.984032	83
78	Project9	2024-05-01	99.921928	100.000000	95.082992	94.850754	83.219887	66
79	Project9	2024-06-01	100.299149	100.000000	94.712581	94.354156	81.148201	50

80 rows × 8 columns

Figure 25 – Dataframe containing the predicted values for all the implemented models without the exogenous features.

4 EXPERIMENTAL EVALUATION

In this chapter, the approach for evaluating the accuracy of the forecasting models implemented in the Chapter 3 will be introduced. This will be followed by a comparison of all the implemented models, enabling analysis and further conclusions.

4.1 Evaluating forecasting model accuracy

There are many metrics available to evaluate the accuracy of AI models, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (sMAPE). Each metric's calculation approach was assessed to determine the most suitable option for the context of this study.

4.1.1 Mean absolute error (MAE)

MAE measures the average absolute difference between predicted values and actual target values (Shcherbakov *et al.*, 2013). This methodology gives equal weight to all errors, regardless of their magnitude. These characteristics make MAE suitable for our case study, as from the NPS perspective, it does not matter whether the values were underestimated or overestimated.

The formula for MAE is:

$$\mathrm{MAE}(y, \hat{y}) = \frac{\sum_{i=0}^{N-1} |y_i - \hat{y}_i|}{N}$$

4.1.2 Mean squared error (MSE)

MSE also measures the average difference between predicted values and actual target values, but it amplifies the error by using the squared difference instead of the absolute difference (Shcherbakov *et al.*, 2013).

The formula for MSE is:

$$\mathrm{MSE}(y, \hat{y}) = \frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{N}$$

It is known that if the dataset contains outliers, MSE may not provide accurate insights. From Figure 14, we observed some outliers; therefore, MSE will not be considered to evaluate the accuracy of our models in this study.

4.1.3 Root mean squared error (RMSE)

RMSE measures the square root of the average squared difference between predicted values and actual target values (Shcherbakov *et al.*, 2013). Like MSE, RMSE gives increased weight to larger errors due to the squaring operation, making it more sensitive to outliers (Hutapea, 2016).

The formula for RMSE is:

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{N}}$$

It is known that if the dataset contains outliers, RMSE may not provide accurate insights. However, since the unit of RMSE is the same as the target value (NPS), it remains highly interpretable. Therefore, this metric will be considered to evaluate the models in this study.

4.1.4 Mean absolute percentage error (MAPE)

MAPE is a metric used to evaluate the accuracy of forecasting methods by calculating the average absolute percentage difference between each predicted value and the actual target value (Shcherbakov *et al.*, 2013). While MAPE is generally effective, it cannot be used with datasets that contain zero values. When the actual value is zero, the denominator in the MAPE formula becomes zero, leading to a division by zero and resulting in undefined values.

The formula for MAPE is:

$$\text{MAPE}(y, \hat{y}) = \frac{100}{N} \sum_{i=0}^{N-1} \frac{y_i - \hat{y}_i}{y_i}.$$

It is known that if the dataset contains NPS scores with zeros, it will lead to the division-by-zero problem described above (NPS ranges from -100 to +100). Therefore, this metric will not be calculated in this study.

4.1.5 Symmetric mean absolute percentage error (sMAPE)

sMAPE is another popular method in time series analysis. sMAPE is calculated as the average of the absolute percentage error between the predicted and actual values, with each error weighted by the sum of the absolute values of both the actual and predicted values (Shcherbakov *et al.*, 2013). The result of sMAPE is a percentage, where values close to 0 indicate better accuracy. This makes sMAPE highly interpretable and does not require deep knowledge of the problem domain. Additionally, sMAPE addresses some of the limitations of MAPE, particularly by using a symmetric formula that helps mitigate issues related to division by zero.

	Calculated in this study	**Used as evaluation criteria**
MAE	✓	✓
MSE	✓	
RMSE	✓	✓
MAPE		
sMAPE	✓	✓

Table 4 – Summary of the metrics considered in this study.

The formula for sMAPE is:

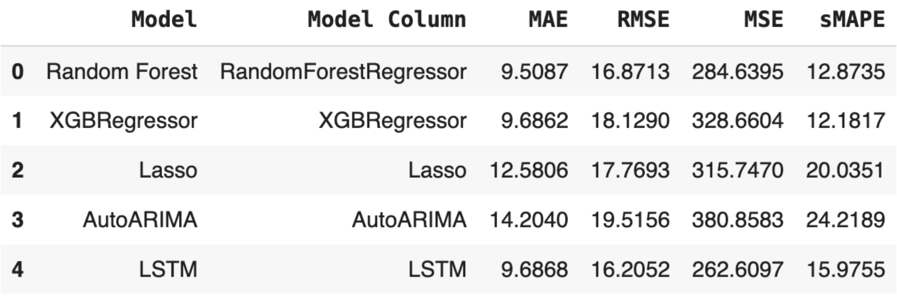
$$\text{sMAPE}(y, \hat{y}) = \frac{100}{N} \sum_{i=0}^{N-1} \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

Given the characteristics of sMAPE, it will be the primary method used in this study to evaluate the accuracy of the implemented models.

4.2 Calculating evaluation metrics

The Table 4 summarizes the metrics selected to evaluate the accuracy of the models implemented in this study.

The selected metrics are then calculated for each model implemented in this study, and the results are saved in a dataframe. The resulting dataframe is displayed in the Figure 26.



	Model	Model Column	MAE	RMSE	MSE	sMAPE
0	Random Forest	RandomForestRegressor	9.5087	16.8713	284.6395	12.8735
1	XGBRegressor	XGBRegressor	9.6862	18.1290	328.6604	12.1817
2	Lasso	Lasso	12.5806	17.7693	315.7470	20.0351
3	AutoARIMA	AutoARIMA	14.2040	19.5156	380.8583	24.2189
4	LSTM	LSTM	9.6868	16.2052	262.6097	15.9755

Figure 26 – Dataframe containing the calculated metrics for each model.

The metrics are displayed in a bar chart in Figure 27 for better visualization and interpretation of the results. The actual NPS scores from the test dataframe are then compared with the predicted values from each model for each project in Figure 28. A scatter plot is created in Figure 29 to compare actual values with predicted values for each model, helping to understand the dispersion of the predictions.

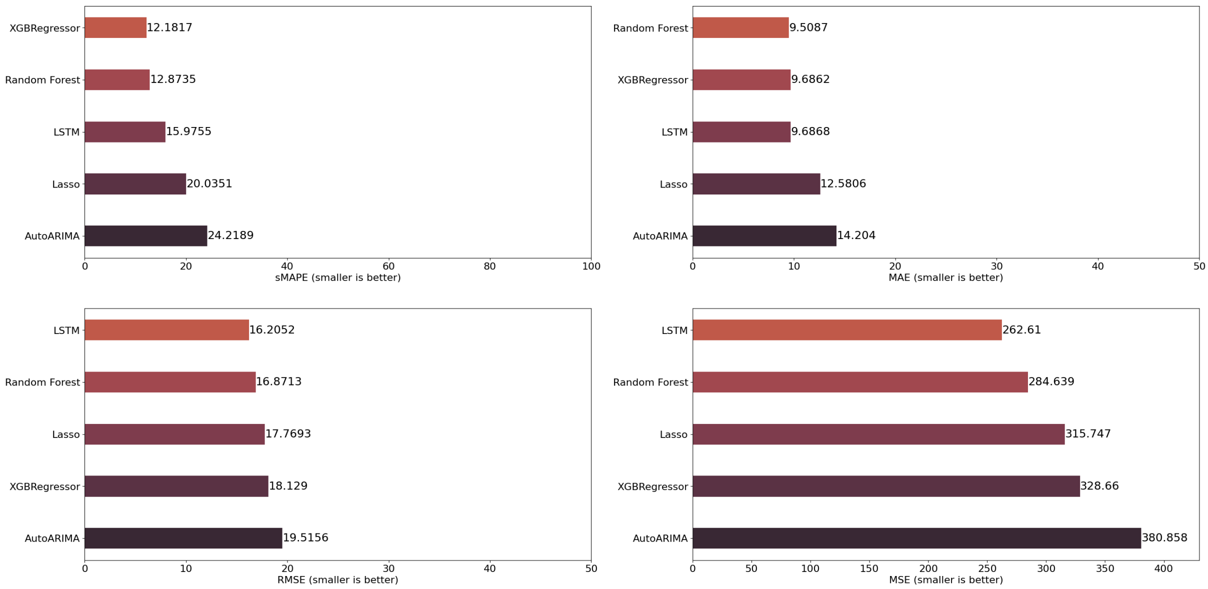


Figure 27 – Bar chart containing all the metrics for comparison.

A second scatter plot is created to show residuals, calculated by the differences between predicted and actual values. This plot helps identify patterns in the residuals that might suggest model bias or changing variability. The chart can be visualized in Figure 30.

4.3 Comparing accuracy of models with and without exogenous features

The accuracy results for the retrained models without the exogenous features are displayed in Figure 31. A chart is plotted to compare the results of the metrics for both approaches (considering the exogenous features and excluding them) in Figure 32.

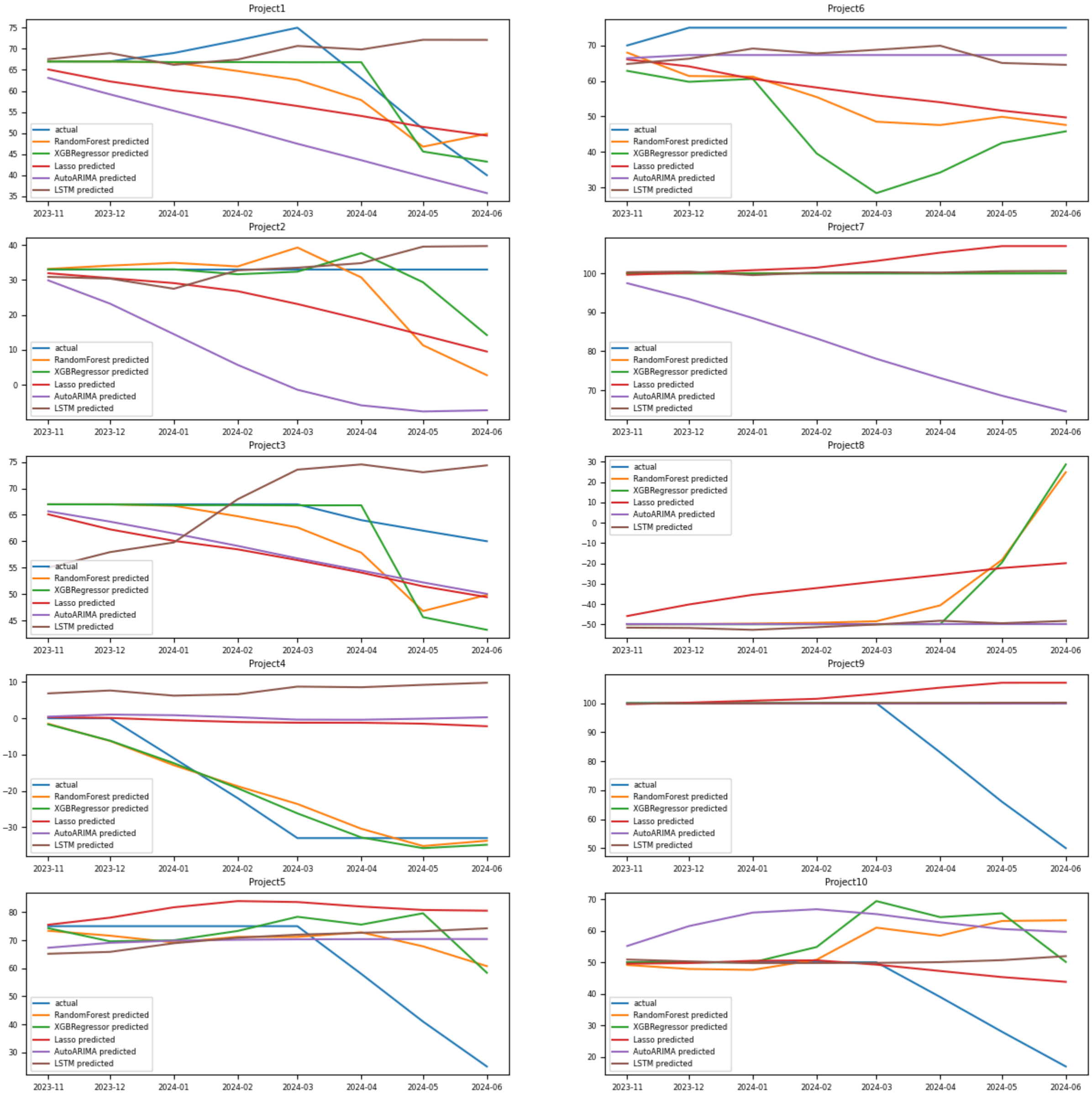


Figure 28 – Timeseries of predicted vs actual.

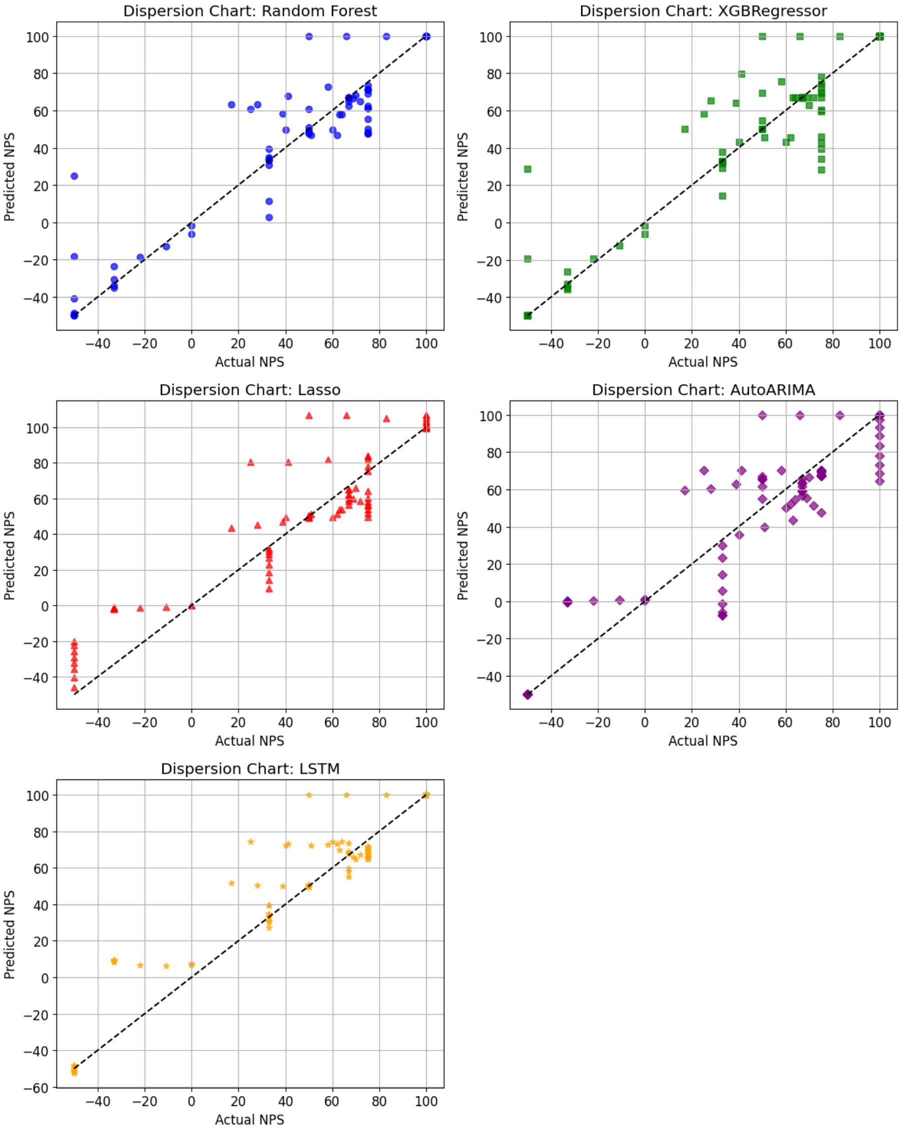


Figure 29 – Scatter plot comparing predicted vs actual values.

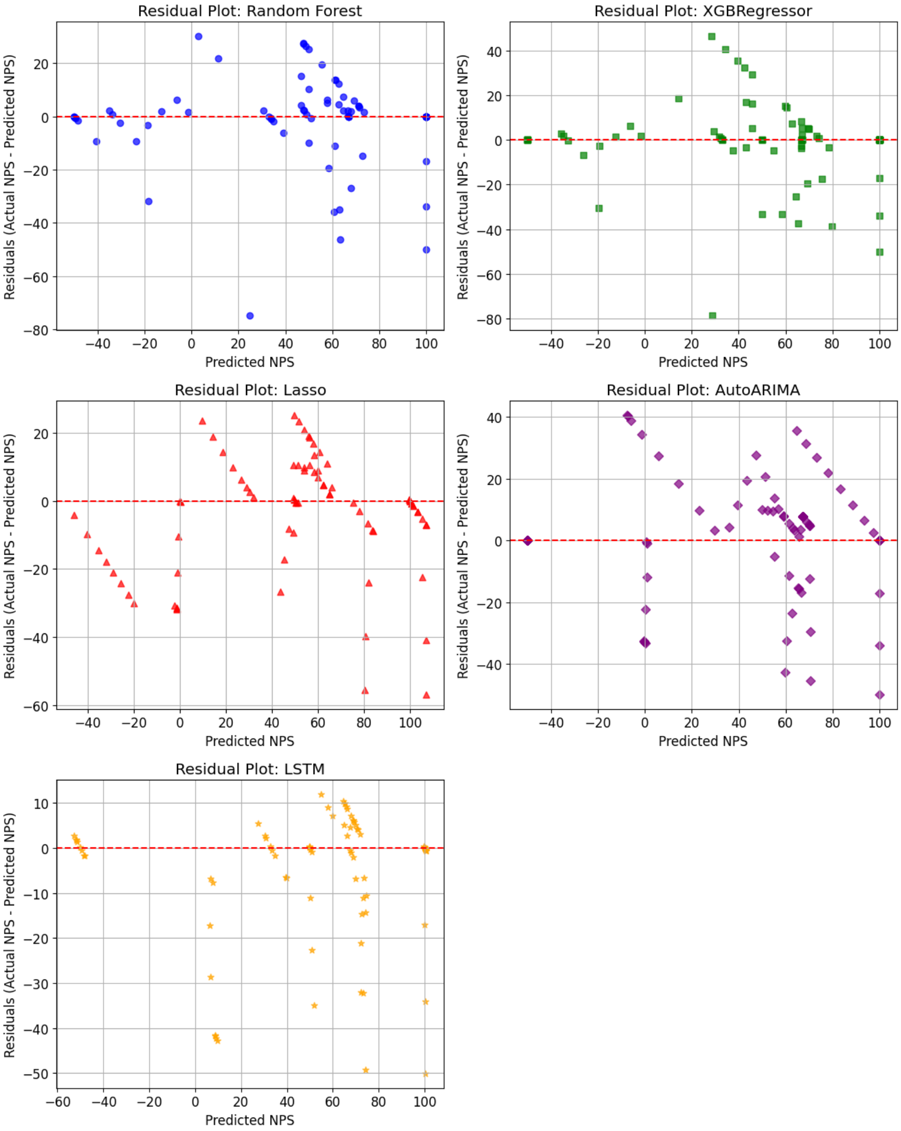
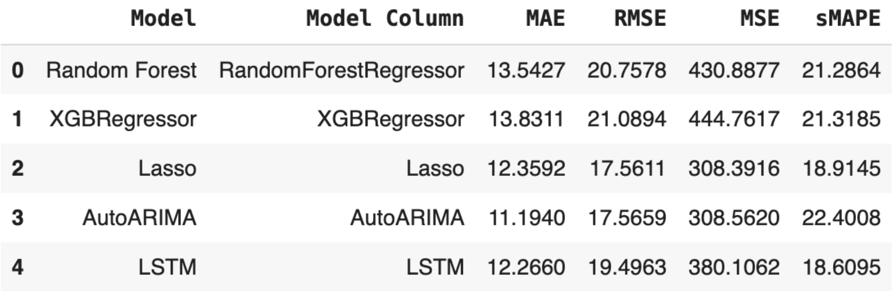


Figure 30 – Residuals charts.



	Model	Model Column	MAE	RMSE	MSE	sMAPE
0	Random Forest	RandomForestRegressor	13.5427	20.7578	430.8877	21.2864
1	XGBRegressor	XGBRegressor	13.8311	21.0894	444.7617	21.3185
2	Lasso	Lasso	12.3592	17.5611	308.3916	18.9145
3	AutoARIMA	AutoARIMA	11.1940	17.5659	308.5620	22.4008
4	LSTM	LSTM	12.2660	19.4963	380.1062	18.6095

Figure 31 – Calculated metrics considering the retrained models without the exogenous features.

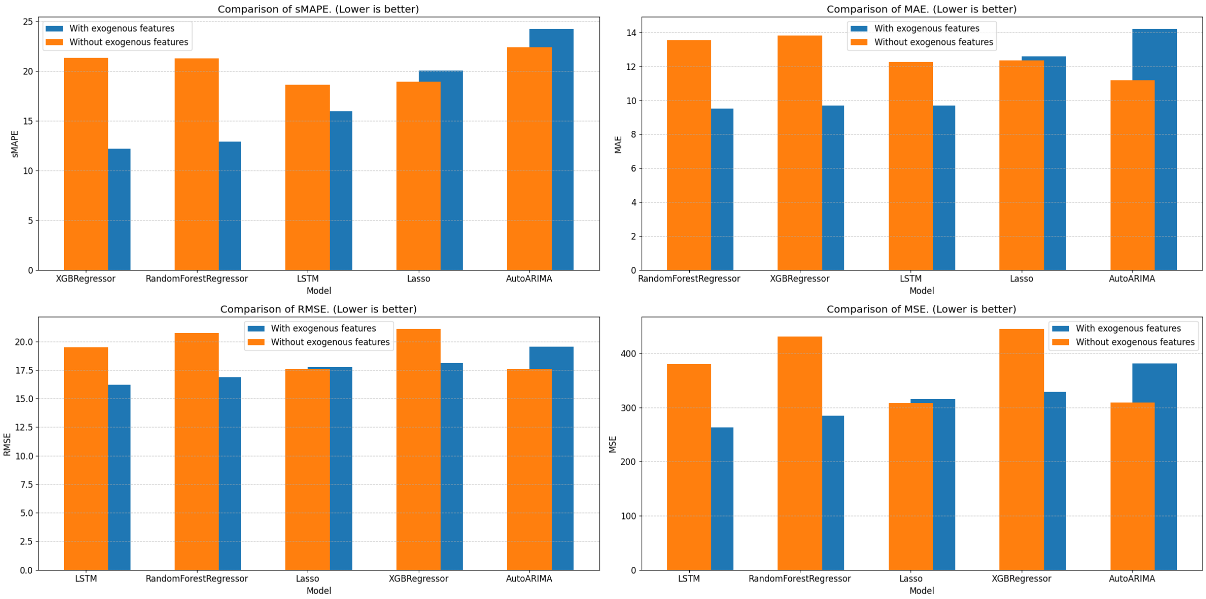


Figure 32 – Calculated metrics considering the retrained models without the exogenous features.

5 CONCLUSIONS

This chapter contains the conclusion of this study by analyzing the results obtained from the experimental evaluation. It also highlights the study's limitations, suggesting possible directions for further research in this field.

5.1 Answering the key questions

In the Chapter 1, this study presented two key questions that technical managers of software development teams may face. The study aimed to address these questions through the proposed methodology and experimental evaluation:

- **Q1** "Is it possible to predict the level of client satisfaction based on the best practices followed by the teams?".
- **Q2** "Which best practices have a more direct impact on achieving a high level of satisfaction?".

To answer **Q1**, it's necessary to evaluate the metrics to see if they meet satisfactory standards. So, what makes an accuracy metric satisfactory? The sMAPE values range from 0% to 100%. A value of 0% means the predictions exactly match the actual values, while 100% means they are a very unsatisfactory fit.

In this study, the XGB, RF, and LSTM models performed well, with sMAPE values below 20%, which is considered satisfactory. On the other hand, the LASSO and AutoARIMA models had sMAPE values above 20%, indicating less favorable performance.

The MAE is measured in the same units as the predicted target. For XGB, RF, and LSTM, the MAE was around 9 NPS points, which is a relatively small deviation given that the NPS scale ranges from -100 to 100.

RMSE is also in the same units as the predicted variable (NPS). It resulted in around 16 units for LSTM and RF and 17 units for LASSO.

The experiment in section 3.6 and the results shown in section 4.3 demonstrated that the RF, XGB, and LSTM models performed relatively better when trained with the exogenous features compared to when these features were excluded.

In conclusion, XGB, RF, and LSTM effectively predicted NPS based on best practices, providing a positive answer to **Q1**.

Regarding **Q2**, this has been addressed during the subsection 3.4.2. Pearson correlations were analyzed to identify the features most strongly correlated with the NPS. The top features, ranked in order of correlation, are: "practice25", "practice37", "practice146", "practice285", "practice81", "practice38", "practice265", "practice71", "practice7", "practice271", and "practice156". Note that detailed descriptions of these software development practices are not provided in this study due to confidentiality of this information.

5.2 Reviewing the initial objectives and results

In addition to addressing the two key questions, the Chapter 1 also established two main objectives:

- **O1** Conduct an exploratory data analysis on a dataset that includes information collected from various software development teams.
- **O2** Utilize and implement Artificial Intelligence algorithms to create models capable of predicting client satisfaction levels (NPS) based on the best practices followed by software development teams.

In section 3.4, exploratory data analysis was performed, providing valuable insights into the dataset. This analysis helped in understanding the data structure and guided key decisions, such as the selection of exogenous features. This work accomplished **O1**.

Additionally, in subsection 3.5.2 of the same chapter, five artificial intelligence algorithms were applied to train models for predicting client satisfaction levels. Some best practices were used as exogenous features in these models. Out of the five models, three achieved satisfactory results, as detailed in this section. This satisfies **O2**.

Based on these results, both objectives **O1** and **O2** can be considered successfully achieved.

5.3 Final remarks, limitations, and future improvements

This study successfully addressed the two key questions and achieved the two primary objectives outlined in Chapter 1. However, it is essential to acknowledge limitations, which can drive future work and improvements. The limitations identified are outlined below:

- **Dataset size:** The dataset utilized in this study is relatively small. Even though it includes data from three years, the final dataset contained only 290 rows for training and 80 rows for validation, a larger dataset could potentially produce models with better performance and generalization capability.
- **Data gaps and inconsistencies:** The dataset had significant parts of missing data, and the frequency of NPS collection did not align with the timing of best practices

surveys. This mismatch resulted in time gaps that had to be filled artificially, which may have affected the model's accuracy.

- **Limited project continuity:** The number of projects that consistently applied best practices over the three years was low, which restricted the scope of the time series analysis to only ten projects.
- **Prediction approach:** The models were trained to predict the final NPS score, which ranges from -100 to +100. An alternative approach could be predicting the raw numbers of promoters, neutrals, and detractors, which might produce more fine insights.
- **Feature selection:** Features were selected based on Pearson correlation. Future work could explore other correlation algorithms to see if they could deliver better results.
- **Correlation does not imply causation:** Further investigations, such as conducting a controlled study, are necessary to determine causality between the practices and NPS.

Despite these limitations, the results of this study are promising. The insights gained here could be further refined and validated with additional data, potentially leading to more robust models. In conclusion, this study has provided valuable insights into the practices that are most relevant for achieving high customer satisfaction, and it represents a good foundation for future research in this area.

REFERENCES

ABBATE, J. The electrical century: Getting small: A short history of the personal computer. **Proceedings of the IEEE**, IEEE, v. 87, 1999.

AKINWANDE, M. O.; DIKKO, H. G.; SAMSON, A. Variance inflation factor: As a condition for the inclusion of suppressor variable(s) in regression analysis. **Open Journal of Statistics**, v. 05, n. 07, p. 754–767, 2015.

BACKUS, J. The history of fortran i, ii, and iii. **ACM SlGPLAN Notices**, IBM Research Laboratory, v. 13, 1978.

BERTONHA, G. **Github of Jupyter Notebook**. 2024. https://github.com/geovanneb/usp_mba-ai-bigdata/ [Accessed: 2024/08/13].

BIAU, G.; SCORNET, E. A random forest guided tour. TEST, v. 25, 2016.

CERUZZI, P. E. A history of modern computing. The MIT Press, 2012.

FORSGREN, N.; HUMBLE, J.; KIM, G. **Accelerate: Building and scaling high performing technology organizations**. [*S.l.: s.n.*]: IT Revolution Press, 2018.

GOPIKA, N.; M.E., A. M. K. Correlation based feature selection algorithm for machine learning. *In*: **2018 3rd International Conference on Communication and Electronics Systems (ICCES)**. [*S.l.: s.n.*], 2018. p. 692–695.

HASHEM, I. A. T. *et al.* The rise of “big data” on cloud computing: Review and open research issues. **Information Systems**, Elsevier, v. 47, 2015.

HUTAPEA, C. **Computational Methods for Flood Forecasting**. 2016. Tese (Doutorado) — University of Houston, 2016.

HUYGEVOORT, R. van de. **Relationship between the Net Promoter Score and the Key Performance Indicators using machine learning techniques**. 2021. Tese (Doutorado) — Tilburg University, 2021.

JUPYTER. **Jupyter**. 2011. https://jupyter.org/ [Accessed: 2024/08/16].

KANNAN, R. *et al.* Prediction of customer transactional net promoter score (tnps) using machine learning. *In*: **Proceedings of the International Conference on Technology and Innovation Management (ICTIM 2022)**. Atlantis Press, 2022. p. 166–179. ISBN 78-94-6463-080-0. ISSN 2352-5428. Available at: https://doi.org/10.2991/978-94-6463-080-0_14.

KRAUS1, S. *et al.* Digital transformation: An overview of the current state of the art of research. **SAGE journals**, SAGE Publications, v. 11, 2021.

LEARN scikit. **scikit-learn**. 2024. https://scikit-learn.org/stable/ [Accessed: 2024/08/13].

MICROSOFT. **Microsoft Forms**. 2024. https://forms.office.com/ [Accessed: 2024/08/11].

MICROSOFT. **SQL Server**. 2024. https://www.microsoft.com/en/sql-server/ [Accessed: 2024/08/11].

MLFORECAST. **mlforecast**. 2024. https://github.com/Nixtla/mlforecast [Accessed: 2024/08/08].

MTI. **MTI**. 1996. https://www.mti.co.jp/eng/ [Accessed: 2024/08/06].

NIELSEN, A. **Practical time series analysis: Prediction with statistics and machine learning**. [*S.l.: s.n.*]: O'Reilly, 2020.

NIXTLA. **NeuralForecast**. 2024. https://nixtlaverse.nixtla.io/neuralforecast/docs/getting-started/introduction.html [Accessed: 2024/08/13].

NIXTLA. **StatsForecast**. 2024. https://nixtlaverse.nixtla.io/statsforecast/docs/getting-started/getting_started_complete.html [Accessed: 2024/08/13].

PERNET, E. H.; CANO, J. J. A systemic maturity model. **International Journal of Computer and Information Engineering**, World Academy of Science, Engineering and Technology, v. 8, 2014.

RANSTAM, J.; COOK, J. A. LASSO regression. **British Journal of Surgery**, v. 105, n. 10, p. 1348–1348, 08 2018. ISSN 0007-1323. Available at: https://doi.org/10.1002/bjs.10895.

REICHHELD, F. F. The one number you need to grow. **Harvard business review**, v. 81, n. 12, p. 46–55, 2003.

SHCHERBAKOV, M. V. *et al.* A survey of forecast error measures. **World applied sciences journal**, v. 24, n. 24, p. 171–176, 2013.

SHELBURNE, B. J.; BURTON, C. P. Early programs on the manchester mark i prototype. **IEEE Annals of the History of Computing**, v. 20, 1998.

SIAMI-NAMINI, S.; TAVAKOLI, N.; NAMIN, A. S. A comparison of arima and lstm in forecasting time series. *In*: **2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)**. [*S.l.: s.n.*], 2018. p. 1394–1401.

VéLEZ, D. *et al.* Churn and net promoter score forecasting for business decision-making through a new stepwise regression methodology. **Knowledge-Based Systems**, v. 196, p. 105762, 2020. ISSN 0950-7051. Available at: https://www.sciencedirect.com/science/article/pii/S0950705120301684.

WANG, Y.; GUO, Y. Forecasting method of stock market volatility in time series data based on mixed model of arima and xgboost. **China Communications**, v. 17, n. 3, p. 205–221, 2020.

YAN, J. Chapter 3 - multidimensional metrics for complementarity. *In*: JURASZ, J.; BELUCO, A. (ed.). **Complementarity of Variable Renewable Energy Sources**. Academic Press, 2022. p. 55–80. ISBN 978-0-323-85527-3. Available at: https://www.sciencedirect.com/science/article/pii/B9780323855273000017.