



Ciencia Latina
Internacional

Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), septiembre-octubre 2024,
Volumen 8, Número 5.

https://doi.org/10.37811/cl_rcm.v8i5

MODERN ML METHODS FOR FAILURE ANALYSIS AUTOMATION ON CLASSICAL ERP SYSTEMS

**MÉTODOS MODERNOS DE APRENDIZAJE AUTOMÁTICO
PARA LA AUTOMATIZACIÓN DEL ANÁLISIS DE FALLAS EN
SISTEMAS ERP CLÁSICOS**

Mario Enrique Vallejo Venegas
University of Guadalajara – México

Ma. Del Rocio Maciel Arellano
University of Guadalajara – México

Victor Manuel Larios Rosillo
University of Guadalajara - México

Jose Antonio Orizaga Trejo
University of Guadalajara - México

Jesus Raul Beltran Ramirez
University of Guadalajara - México

Modern ML methods for failure analysis automation on classical ERP systems

Mario Enrique Vallejo Venegas¹

mario_vallejo@hotmail.com

<https://orcid.org/0000-0002-5607-766X>

Harman International, SAP SCM & Analytics
University of Guadalajara, PhD in IT
Mexico

Ma. Del Rocio Maciel Arellano

ma.maciela@academicos.udg.mx

<https://orcid.org/0000-0002-5548-2073>

University of Guadalajara, PhD in IT.
Smart Cities Innovation Center.
Mexico

Victor Manuel Larios Rosillo

victor.m.lariosrosillo@ieee.org

<https://orcid.org/0000-0002-2899-724X>

University of Guadalajara, PhD in IT.
Smart Cities Innovation Center.
Mexico

Jose Antonio Orizaga Trejo

jose.orizaga@academicos.udg.mx

<https://orcid.org/0000-0001-5649-5514>

University of Guadalajara, PhD in IT.
Smart Cities Innovation Center.
Mexico

Jesus Raul Beltran Ramirez

jrbeltran@academicos.udg.mx

<https://orcid.org/0000-0001-8645-9258>

University of Guadalajara, PhD in IT.
Smart Cities Innovation Center.
Mexico

ABSTRACT

Typically IT teams working with ERP systems have little or no knowledge of artificial intelligence and more specifically Machine Learning (ML) and Natural Language Processing (NLP) models because their working environment is mostly focused in supporting usually commercial ERP systems like SAP, Oracle ERP, Microsoft Dynamics, etc., so they focus on the functional aspects and sometimes in some proprietary development environments like ABAP language for SAP systems, to make very specific customizations. The current research work is aimed to provide detail insights of the state-of-the-art ML and NLP models that could be used in a classical ERP environment and also it pursues the objective of investigating the technical feasibility and the ease or difficulty to provide artificial intelligence to a classic ERP system that does not have it with the intention of automating the analysis of errors and failures in the ERP system that due to its volume is difficult to manage by human operators. The aim is therefore to achieve significant savings in time and IT human resources consumed in failure analysis. Another objective is to share with the reader several lessons learned by the researchers while investigating the available literature and while experimenting and testing with several of the existing models available in Python and C# languages when comparing different technology platforms. This should provide valuable information to IT managers, project managers, developers, and testers who normally work with ERP systems and not with AI, so they are not so familiar with it.

Keywords: artificial intelligence, machine learning, automation, supervised learning, erp

¹ Autor Principal

Correspondencia: mario_vallejo@hotmail.com

Métodos modernos de aprendizaje automático para la automatización del análisis de fallas en sistemas ERP clásicos

RESUMEN

Normalmente, los equipos de TI que trabajan con sistemas ERP tienen poco o casi ningún conocimiento de inteligencia artificial y, más concretamente, de modelos de Aprendizaje Automático (ML) y Procesamiento del Lenguaje Natural (NLP) porque su entorno de trabajo se centra principalmente en dar soporte a sistemas ERP normalmente comerciales como SAP, Oracle ERP, Microsoft Dynamics, etc., por lo que se enfocan en los aspectos funcionales y, a veces, en algunos entornos de desarrollo propietarios como el lenguaje ABAP para sistemas SAP, para realizar personalizaciones muy específicas. El presente trabajo de investigación pretende dar una visión detallada del estado del arte de los modelos ML y NLP que podrían ser utilizados en un entorno ERP clásico y además persigue el objetivo de investigar la viabilidad técnica y la facilidad o dificultad de dotar de inteligencia artificial a un sistema ERP clásico que no disponga de ella con la intención de automatizar el análisis de errores y fallos en el sistema ERP que por su volumen es difícil de gestionar por operadores humanos. Se persigue pues un ahorro significativo de tiempo y recursos humanos de IT consumidos en el análisis de fallos. Otro objetivo es compartir con el lector varias lecciones aprendidas por los investigadores mientras investigaban la literatura disponible y mientras experimentaban y probaban con varios de los modelos existentes disponibles en los lenguajes Python y C# mientras comparaban diferentes plataformas tecnológicas. Esto debería proporcionar información valiosa a los responsables de TI, jefes de proyecto, desarrolladores y probadores que normalmente trabajan con sistemas ERP y no con IA, por lo que no están tan familiarizados con ella.

Palabras clave: artificial intelligence, machine learning, automation, supervised learning, erp

*Artículo recibido 08 septiembre 2024
Aceptado para publicación: 12 octubre 2024*



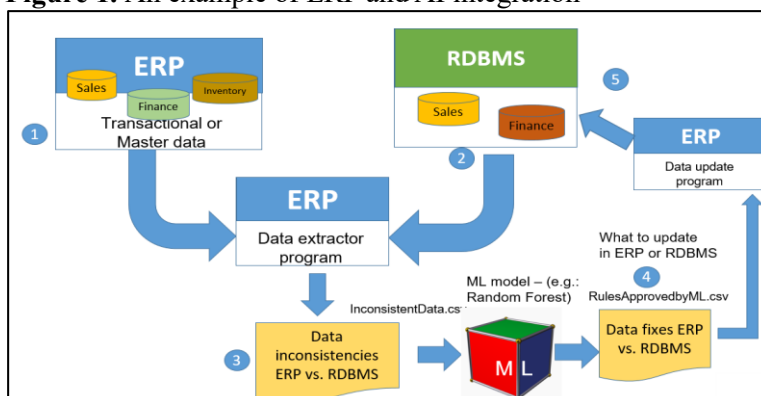
INTRODUCTION

The goal of current research work is to provide details of the state-of-the-art ML and NLP models that could be used in a classical ERP environment. It is also aimed to investigate the technical feasibility (the ease or difficulty) to provide AI to a classic ERP system that does not have it. The objective is to automate failure analysis in the ERP system where high failures number is difficult to manage by human operators (support IT engineers). A second goal is achieving significant savings in time and IT human resources consumed in failure analysis. In the following paragraphs it is provided a detailed discussion of ERP systems, AI, ML, DL, and NLP models to provide the reader with some background and knowledge foundation. Later, in the methodology section, it is explained all the experimentation steps done for this research.

Erp Systems

The industrial revolution brought with it the emergence of large corporations in the United States and Europe. These large scale organizations, many of them dedicated to the manufacture of a diversity of products, had the need to implement inventory and manufacturing controls for which it was necessary to have adequate inventories of raw materials. Thus it appeared the planning of material requirements and the need to have information systems for inventory control, material purchase and production planning. Information systems quickly evolve to MRP (materials requirements planning) and later to ERP (enterprise resource planning) systems around the year of 1960. Also, about the same time, the expert systems appear, being the first attempt to imitate human intelligence using computers for the decision-taking process in large corporations. The integration between ERP and AI could be accomplished as suggested on this work (Figure 1).

Figure 1. An example of ERP and AI integration



Artificial intelligence

Artificial Intelligence (AI) has been a field of research for several decades, but in the twenty-first century it has taken on a special boom largely due to lower hardware costs and increased computational power, in addition to the fact that it has been able to capitalize on the market and has ceased to be a purely academic and research subject and has become a way of doing business and a way of employing thousands of software engineers and many other specialists. AI is intended to simulate human intelligence processes; it's not aimed to duplicate or replace them. Many thought leaders in AI space even think AI's goal should be to augment human capabilities.

What is intelligence? As per John McCarthy (McCarthy, J. (1970, January 1). What is AI? / Basic Questions), *"Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines"*

What is artificial intelligence? John McCarthy (McCarthy, J. (1970, January 1). What is AI? / Applications of AI) defines AI as follows: *'It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable'*.

Brief history of artificial intelligence. Perhaps the generations of people who in this third decade of the 21st century work with information technologies, software development, data science, artificial intelligence, automatic learning and other disciplines similar to data processing with computers, may think that artificial intelligence is something that was born in this 21st century, but it is not so. In fact, it is considered that the discipline and field of knowledge currently known as 'artificial intelligence' was born in 1956 in a workshop held at Dartmouth University (Chow, 2021), in the city of Hanover, in the state of New Hampshire in the United States of America, a workshop that brought together the most brilliant minds of the time, from various disciplines such as cognitive science and computer science. The workshop was held in the summer of 1956 and was called the 'Dartmouth Summer Research Project on Artificial Intelligence'. The organizers of the workshop, such as Assistant Professor John McCarthy, thought that if they could get all the eminent students and professors interested in the subject together to devote time to it and avoid distractions, they could make real progress, for even before the workshop



took place, they were somewhat disappointed with the research papers submitted to the journal *Annals of Mathematical Studies*. It was thought that the contributors to the journal for some reason did not focus on the potential of computers to possess intelligence and this prompted the workshop to be organized by having a group of eminent students to clarify and develop ideas about thinking machines. Professor McCarthy approached the Rockefeller Foundation to request funding for a seminar at Dartmouth for 10 participants and in the summer of 1955, he formalized the project with his friends and colleagues Marvin Minsky of Harvard University, Nathaniel Rochester of IBM Corporation and Claude Shannon of Bell Telephone Laboratories to lay the foundations for artificial intelligence. The key idea of the workshop was that any feature or aspect of both learning and human intelligence could be described in a simple but very precise way as if they were instructions to be followed step by step or as if they were a procedure or an algorithm that a computer could then simulate with a program in some programming language.

In 1956, what many consider to be the first AI computer program was developed, which was intended to mimic humans in their problem-solving ability. This program was called the logic theorist and the code was written by programmers Allen Newell and Herbert A. Simon.

Later in the 1960s to 1970s, "expert systems" were developed, which to this day use a series of rules and knowledge bases to solve specific problems in various fields, but ultimately use deterministic logic with classical programming based on typical concepts such as variables, iterative structures and single and multiple conditionals to translate the logic of business rules to a series of conditionals and result assignment logic in some procedural programming language.

The following decade, from the 1970s to the 1980s, is a stage known as "The AI Winter" (Thorwirth, 2021) in which AI experienced a period of failure and reduced funding and even loss of public interest, as the previous decade created great expectations for AI, possibly exaggerating its capabilities in order to achieve attractive economic profits from the sales of AI software in the market. However, in practice there were really poor advances in trying to use AI in real business applications and this led to a loss of credibility and interest in investing money and resources in AI software.

In 1980, despite "The Winter of AI," a new idea or invention emerged that rekindled interest in artificial neural networks. This idea was called "back-propagation". Neural networks usually have multiple layers, which are called "hidden layers" and work as in mathematical differential calculus and the "chain

rule" to derive, there may be a function that in turn calls another function, i.e. they are nested functions and the innermost function is calculated first and delivers its results to the outermost function that called it so that it works with the result of the first one and generates a new final resultant value.

Then in the 1990s-2000s, support vector machines, decision trees and Bayesian networks became popular. This was due to the availability of large datasets and increased computational power of hardware with the cheapening of microprocessors and memory and the advent of graphics processing units or GPUs for more efficient image processing on computers. It was this same cheapening of electronics and hardware that in the decades from 2000 to 2010 set the tone for the resurgence of deep learning and neural networks of all types (convolutional or CNN, recurrent or RNN, etc.) that made it possible to apply AI to image recognition and natural language processing.

It is from 2010 onwards that AI becomes more general use as it is introduced in industry and in everyday life thanks to social media and mobile devices that are now accessible to everyone. In the year 2024, the time this paper is written, use cases are so common in situations like whatsapp chatbots for natural language processing in all kinds of commercial and government internet sites, computer vision with applications like facial recognition used to unlock cell phones, robotics which is mostly used in industrial environments, and other sectors such as healthcare, finance, transportation and entertainment, the latter where it is very common to see entertainment platforms such as Netflix and Spotify analyzing customer preferences and consumption habits and making suggestions based on these consumption patterns. Large Language Models (LLM) is the most recent invention being used in generative AI with applications like ChatGPT.

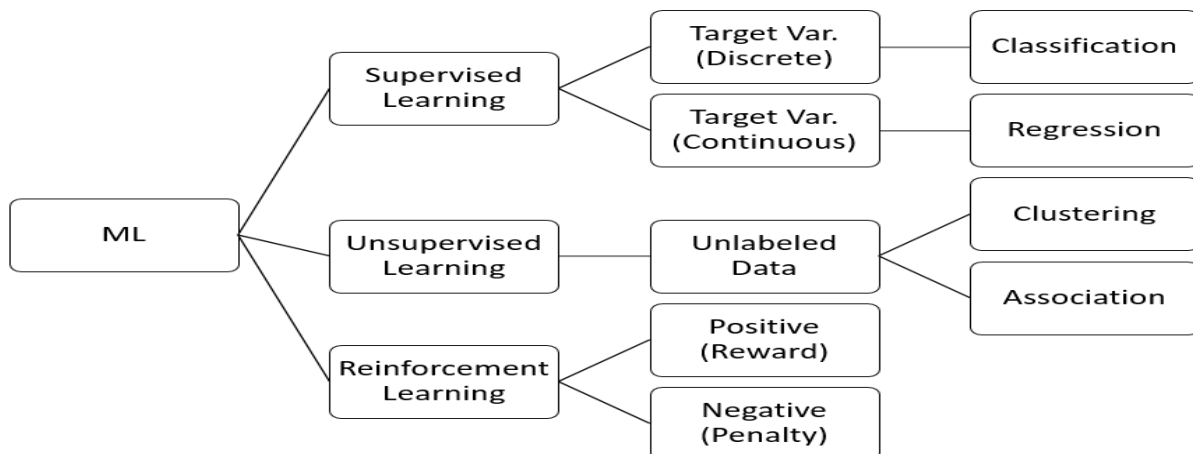
Machine learning techniques

Machine Learning (ML) is a sub-field derived from AI that specializes in applying statistical methods to make classifications or predictions. ML works with data sets and there are several dozen different computer programs that have been developed and tested over the decades that are called models. These models perform statistical calculations by applying various methods that statisticians and mathematicians have developed since at least the middle of the 20th century. The logic or procedure used to perform these calculations is called an algorithm and each model is based on a given set of parameters. When you want to work with a model, you feed it with data sets. Before a dataset can be

used, it is necessary to analyze its columns of information and perform data cleaning or data preparation. For example, if there are columns containing date data, it is necessary to put them in a format that the model can read and interpret as a date. If any column has outliers, which are those values that are too far away from the arithmetic mean, for example 2.5 standard deviations or more, it is necessary to decide what should be done with such data value because they can significantly affect the result of the calculations and the behavior of the model. In many cases it is decided to eliminate such outliers once their nature is understood. In other cases a different strategy is defined where average or mode values could be calculated to replace the outliers. Another scenario that is very often encountered is that there are missing values. This happens frequently, for example, in the field of medicine, where there is no consistency of data because each hospital or medical office, public or private, may have different processes and procedures for capturing and processing data on patients and their diseases, diagnoses, treatments, laboratory studies, follow-up information, etc. For all these reasons, it is necessary to standardize the datasets in order to curate them and have cleaner information to feed the ML models. Once the data is clean, the next step is to feed the model waiting for it to perform either a classification task or a prediction or prognosis task. In ML, the model is said to learn based on its experience with the data without the need to explicitly program it (write specific code) to recognize every possible scenario. Therefore, it is said that the model learns and then it could be stated that there are different types of learning. There are many authors who claim that there are at least three types of learning, so it can be considered that there is a consensus in this conceptualization (**Figure 2**) (Sarker, 2021) (Mahesh, 2018) (Edwards, 2020):

1. Supervised learning
2. Unsupervised learning
3. Reinforced learning

Figure 2. An overview of ML models: Supervised/Unsupervised/Reinforcement learning and general use case



In the following sections, an explanation of the different learning types is provided together with the main models used as of present.

Supervised learning. In this approach, an output is mapped from an input and the function that achieves this is trained with a sample of input-output value pairs. The output is a tag or label that corresponds to a group of input values. The training data is a collection of such input-output pairs. The learning is achieved by the model when prediction accuracy is maximized. Supervised learning is most commonly used for classification tasks, for example: text classification. For example the classification of a series of words to determine if it is a verb, noun, adjective, preposition, adverb, etc. IBM (IBM, 2023, *What Is Supervised Learning?*) Claims that ‘*Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross validation process. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox.*’.

Linear regression is used to identify the relationship between a dependent variable (a number) and one or more independent variables (also numbers), and is normally used to make predictions about future outcomes (the result is also a number).

There are two types of regression functions: simple and multiple linear regression. The first is when there is only one independent variable and one dependent variable. The second, is when there are many

independent variables impacting on the result (dependent variable). Linear regression is used when the dependent variables are continuous.

Logistic regression is used when the dependent variable is categorical, meaning that it has binary outputs, such as *true* and *false* or *yes* and *no*. Logistic regression is used when the dependent variables are discrete. Logistic regression is mainly used to solve binary classification problems, such as email spam identification or violence vs. not-violence classification on thread news on the web, etc.

Naïve Bayes, is a classification method that uses class conditional independence concept from the Bayes Theorem where every feature is totally independent of the others. The probability to produce a given result is the same for every feature (a.k.a. predictor). Hence the term “naïve” which means ingenuity because in the real world characteristics are almost never independent. There is always certain influence between them in most of the real-life problems.

K-nearest neighbor (KNN), is a method that classifies data points based on their proximity and association to other data points. It is a non-parametric algorithm that works by assuming that similar data points can be found near each other by calculating the Euclidean distance between data points, which is a physical distance, and it assigns a category based on the most frequent category or average. It is a very popular model because it is easy to use and it has a low computing processing time, however when the size of the dataset increases, the processing time also grows and performance decreases, and that's why data scientist do not use it classification tasks especially on big datasets. However, KNN is frequently used for image recognition and suggestion engines.

Support vector machine (SVM) is used for classification and regression tasks. It is a model developed by the Russian mathematician Vladimir Vapnik in 1992 at AT&T Bell Laboratories. It can be used for classification and regression tasks. It is used for small dataset as it has long processing times. SVM is based on the idea of finding a hyperplane that best separates the features into different domains. This hyperplane is known as the decision boundary, separating the classes of data points (**Figures 3 and 4**) on either side of the plane. The points closest to the hyperplane are called as the support vector points and the distance of the vectors from the hyperplane are called the margins. The farther SV points are from the hyperplane, more is the probability of correctly classifying the points in their respective region or classes.

Figure 3. Iris flowers (Iris Setosa, Iris Versicolour, Iris Virginica)

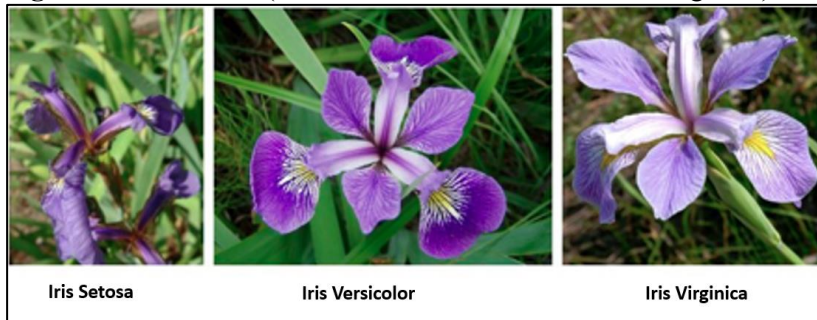
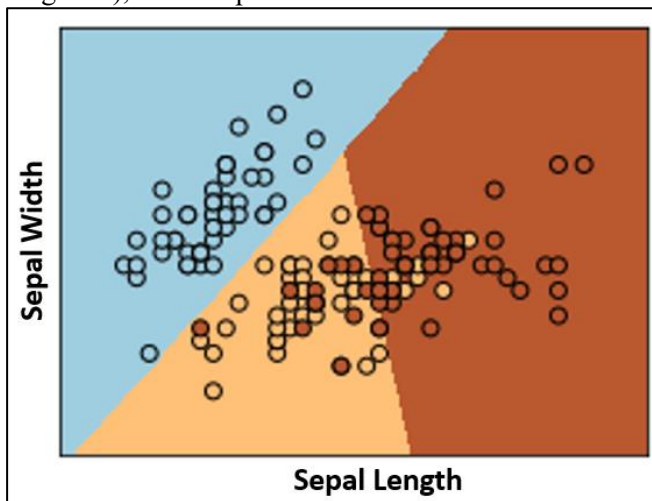


Figure 4. Iris flower classification by length and width of sepal (Iris Setosa vs. Iris Versicolour vs. Iris Virginica), an example of SVM classification



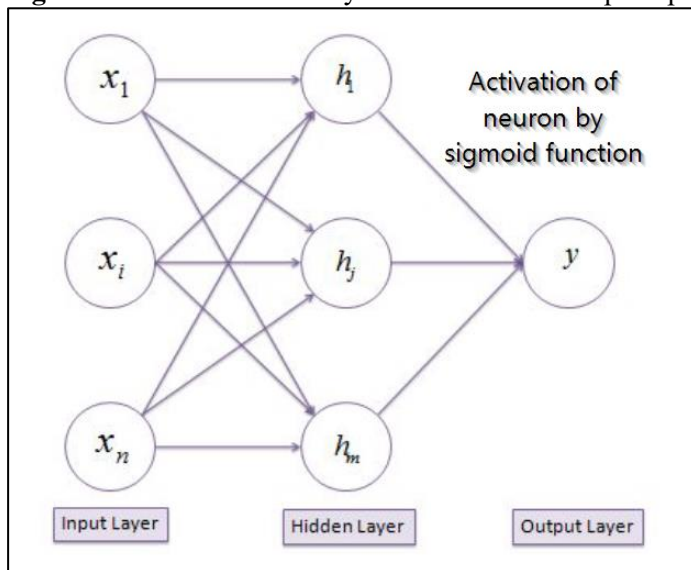
Random forest is a method also used for classification and regression tasks. The forest is made of many decision trees without correlation and are merged to create more accurate predictions by reducing variance. It uses feature bagging and randomness when building individual trees to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

Random Forest is one of the best classification models in the ML algorithms hierarchy.

Neural networks, as prior mentioned, there are several types of neural networks such as "recurrent neural networks", "convolutional neural networks", "modular neural networks", etc., and all of these ultimately are "artificial neural networks" because they all use mathematical models which are not biological in origin. Neural networks process data by imitating the interconnection of the human brain through layers of biological neuron connections (**Figure 5**). Each node is made up of inputs, weights, a bias (or threshold), and an output. If the output value exceeds a given threshold, it activates the node, passing data to the next layer in the network. Neural networks learn this function by adjusting the cost

function in a process known as **gradient descent** which can be thought of as a blindfolded person going down a hill, taking small steps, feeling only the slope and if it is very steep taking a small step forward and continuing until the person feels how the slope ends and managed to reach the bottom of the valley where there is no longer a slope because the floor is even or flat. When the cost or loss function is near or equal zero the steepness of a slope is also zero (or almost zero) meaning the model has completed its descent, it has reach the local minimum of the function and the output of the model is very likely to be accurate yielding the correct answer.

Figure 5. Model of a multilayer neural network or perceptron, with N number of hidden layers



Unsupervised learning. This approach uses machine learning algorithms to analyze and cluster unlabeled data sets. This means, there is no group of humans pre-classifying and labeling the data in order to use it to train a model.

Unsupervised learning models discover hidden patterns or clusters of data by themselves, no human need is needed for this clustering or grouping task because the model has the ability to discover similitudes and differences in data points which makes them the best option to be used in exploratory data analysis (EDA) which is a visual data analysis with the help of predefined libraries or packages available in Python language. These models can also reduce the number of features needed by means of a process called “dimensionality reduction” where each feature is considered a dimension and where the goal is to determine which features can be ignored by the model as they do not have a significant impact on the prediction or classification task. Having fewer features saves memory and computational costs

and generates better predictions by reducing overtraining and improving generalization. It also helps to simplify visualization by focusing on important features during EDA. Principal component analysis (PCA) and singular value decomposition (SVD) are two common methods for dimensionality reduction. IBM (IBM, 2023, *What Is Unsupervised Learning?*) Defines unsupervised learning as ‘*Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.*’.

Unsupervised learning models are used for three main tasks:

1. Association
2. Clustering
3. Dimensionality reduction.

Below a definition each learning method is provided and common algorithms and approaches to carry them out effectively are highlighted.

Association rules. IBM defines this term as it follows: ‘*An association rule is a rule-based procedure for finding relationships between variables in a data set. These procedures are used for shopping cart analysis, a situation that enables companies to better understand the connections between different products. Understanding customer consumption habits allows businesses to develop better cross-selling methods and suggestion engines. For example, the playlist "Customers who bought this item also bought" in Amazon web pages*’ (What Is Unsupervised Learning? | IBM, 2023.).

Techniques like Apriori, Eclat, and FP-Growth are referred as most commonly used for generating association rules. As an example of this, assume “Justin Bieber radio” is set as search string within the application Spotify, then the resulting list may begin with his song "Next to you" and there is a significant probability that the next song shown on this channel is a “Sean Kingston” song, like "Eanie Meanie", based on users’ previous listening habits and those of other customers of the Spotify application.

Clustering IBM defines this approach as follows: ‘*Clustering is a data mining technique which groups unlabeled data based on their similarities or differences. Clustering algorithms are used to process raw,*

unclassified data objects into groups represented by structures or patterns in the information.’ (What Is Unsupervised Learning? | IBM, 2023)

It is also referred that there are several types of clustering: probabilistic with methods like the Gaussian Mixture Model (GMM) and, hierarchical a.k.a. ‘hierarchical cluster analysis’ that can be in turn agglomerative ("bottom-up approach") or divisive. Also, overlapping and exclusive clustering where data points can either overlap or be mutually exclusive on clusters. It is recommended further reading of the original sources for more details.

Dimensionality reduction. IBM defines this concept as ‘*Dimensionality reduction is a technique used when the number of features, or dimensions, in a given dataset is too high. It reduces the number of data inputs to a manageable size while also preserving the integrity of the dataset as much as possible*’. (What Is Unsupervised Learning? | IBM, 2023.)

It is understandable that the more data the more accurate results but also the worse performance of ML models which can lead to overfitting. This last concept refers to a situation when a given model produces correct results when is ran for training data but not when executed for new data. ML models need to be trained first with around twenty percent of the data records in a given dataset. Once fitting is successfully done, the model can be released to production to be used with new real-world data with the expectation that model’s predictions or classifications are still accurate as they were during the fitting phase, unless too many dimensions (features) where used during fitting so the model resulted in a overfitting condition, which is obviously undesired. Two dimensionality reduction methods can be used and an overview is given in the following sections.

Principal component analysis (PCA).

IBM defines this model as follows: ‘*PCA is a type of dimensionality reduction algorithm which is used to reduce redundancies and to compress datasets through feature extraction. This method uses a linear transformation to create a new data representation, yielding a set of "principal components."* The first principal component is the direction which maximizes the variance of the dataset. While the second principal component also finds the maximum variance in the data, it is completely uncorrelated to the first principal component, yielding a direction that is perpendicular, or orthogonal, to the first component.’ (What Is Unsupervised Learning? | IBM, 2023).



It can be inferred that the PCA method makes it possible to "condense" the information provided by multiple variables into just a few components.

Singular value decomposition (SVD). IBM defines this model as follows: *'SVD is another dimensionality reduction approach which factorizes a matrix, A , into three, low-rank matrices. SVD is denoted by the formula, $A = USVT$, where U and V are orthogonal matrices. S is a diagonal matrix, and S values are considered singular values of matrix A '* (What Is Unsupervised Learning? | IBM, 2023). It's also referred that, in a similar way than PCA, SVD is used to reduce noise and compress data, such as image files.

Applications of unsupervised learning. Unsupervised learning allows for faster identification of patterns in large datasets. It is not necessary for engineers or data scientists to label the input data to teach the model what the corresponding output is. The model still has to be trained with about twenty percent of the data available in the dataset to be treated, but before that a dimensionality reduction is performed and all that is fed to the model are numbers, there is no categorical data per se and as they are mathematical models they are optimized to run in the shortest possible time doing only numerical calculations and that allows them to process large datasets in a reasonable time. The following use cases are identified for unsupervised learning.

News sections. Like the ones used by Google News to categorize articles. For example, the results of a school shooting might be categorized under its "USA" news tag.

Computer vision. For example in object recognition, in facial recognition to authenticate users (Face ID in iPhone to unlock the phone).

Medical Imaging. Used in image classification in radiology, oncology, and pathology to quickly and accurately diagnose patients with neoplasia (tumors).

Anomaly detection. Used to discover outlier data points within a data set. These are known as anomalies and can call for attention to a variety of root causes like faulty equipment, human error, or security flaws.

Recommendation Engines. Using past purchasing behavior data, to reveal data trends to develop more effective cross-selling strategies. This is used to make relevant add-on recommendations to customers during checkout for online retailers in sections like "users who bought this article also looked at these others".

Reinforcement learning. IBM (IBM, 2023, what is reinforcement learning?) Defines this approach as follows: *‘Reinforcement learning is a learning paradigm that learns to optimize sequential decisions, which are decisions that are taken recurrently across time steps, for example, daily stock replenishment decisions taken in inventory control.*

At a high level, reinforcement learning mimics how we, as humans, learn. Humans have the ability to learn strategies that help us master complex tasks like swimming, gymnastics, or taking a test. Reinforcement learning broadly seeks inspiration from these human abilities to learn how to act. But more specifically to practical use cases of reinforcement learning, it seeks to acquire the best strategy for taking repeated sequential decisions across time in a dynamic system under uncertainty. It does so by interacting with a simulator of the stochastic dynamic system of interest, also called as an environment, to learn such winning strategies. A strategy to take repeated sequential decisions across time in a dynamic system is also called as a policy. Reinforcement learning tries to learn the winning policy, namely a winning recipe of how to take actions in different states of a dynamic system’.

Reinforcement learning addresses sequential decision-making problems that are often subject to uncertainty, for example: multi-tier, multi-supplier inventory management with lead times under demand uncertainty; control issues such as autonomous manufacturing operations or production plan control; and resource allocation issues in finance or operations.

Reinforcement learning interacts with an environment that is a simulator of the stochastic dynamic system of interest, in order to learn these winning strategies. A strategy for making sequential decisions repeated over time in a dynamic system is also called a policy. Reinforcement learning tries to learn the winning policy, that is, a winning recipe for how to perform actions in different states of a dynamic system.

Deep Learning Techniques

Deep learning (DL) is a subset of machine learning and has demonstrated significantly superior performance to some traditional machine learning approaches. DL uses a combination of multi-layered artificial neural networks, processing- and data-intensive training, inspired by our latest understanding of human brain behavior. This approach has become so effective that it has even begun to surpass human

capabilities in many areas, such as image and speech recognition and natural language processing. DL models process large amounts of data and are generally unsupervised or semi-supervised.

DL uses concepts analogous to those of neural biology, such as the notion of the artificial neuron analogous to the biological neuron. Another example is the artificial neural network analogous to mammalian neural networks. But similarities end here. DL uses several types of mathematical models that are generally called "neural networks" and have different specific names such as "recurrent neural networks" or "convolutional neural networks", etc., but that these networks are also "artificial neural networks" because all the mathematical models they use are artificial since they are not biological or natural. A conclusion that seems obvious, but that is very important to emphasize because within single-layer ML there is the "artificial neural network" or ANN model and in the computational or information technology profession it is ignored the fact that all computational neural networks (recurrent, convolutional, adversarial) used in DL (multilayer) are also artificial even if their name does not include that word.

DL learns features and tasks directly from data which can be images, text or sound. In medicine, for example, deep learning models are used for image interpretation in imaging or radiology, deep learning models learn to classify input images into appropriate categories.

Natural language processing techniques

According to Beysolow (Beysolow II, 2018), '*natural language processing (NLP) is a subfield of computer science that is focused on allowing computers to understand language in a "natural" way, as humans do. Typically, this would refer to tasks such as understanding the sentiment of text, speech recognition, and generating responses to questions*'. NLP is a part of the field of AI and with the techniques of conversion of words to vectors (vectorization) it is also considered a part of ML focused on human-computer communication. NLP addresses the inherent problem that while human communications are often ambiguous and imprecise, computers require unambiguous and precise messages to be able to communicate between machines.

METHODOLOGY

Several NLP models that could be used in a classical ERP system to investigate the technical feasibility and the ease or difficulty to provide artificial intelligence to a classic ERP system were explored.



Two software development platforms, Python and Microsoft ML.Net, were investigated. The corresponding IDEs were installed and various packages or extensions required to work with ML and NLP were configured. Several small programs were written in Python and C# to experiment with NLP principles. The goal was to compare the two referred platforms for easy-of-use and performance. Both were running on Windows 11 for a 64-bit CPU. No detailed instructions on how to download, install, and configure the development platforms for Python and ML.Net are provided in the paper because there is plenty of public documentation about it on the Internet. Hence only a reference to corresponding websites is provided in the bibliography, so the reader can check detailed installation instructions there. Also, in the part of ERP system, several ABAP programs were written to emulate a production environment in SAP ERP where multiple job failures would occur. The job logs of the failures were downloaded to local files and were fed as input to Python and C# programs executing NLP methods to experiment with.

In the following paragraphs the steps of the experimentation are described from selecting the programming language and references to where to download and install integrated development environments (IDE) up to ERP software development, integration and execution of the components. It is provided also a discussion of the software development in Python, and C#.

Mathematical model of bag of words

According to (Ghosh, T., & Kumar, S., 2022), the mathematical model for Bag of Words is given by three statistical calculations that are term frequency, inverse document frequency, and the product of both, the Term frequency–inverse document frequency (TF-IDF). A brief introduction is given in the following lines.

Term frequency.

Term frequency, is the relative frequency of term “t” within document “d”.

$$TF_{(i)} = \frac{\log_2(Freq(i,j)+1)}{\log_2(L_j)}$$

Freq (i, j) = Frequency of term i in document j.

L (j) = Total number of terms in document j.

A different notation can be:

$$TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

Inverse document frequency.

IDF measures the rarity of a term across a collection of documents. It is aimed to penalize words that are common across all documents. It is calculated as follows:

$$IDF(t, D) = \log \left(\frac{\text{Total number of documents in the corpus } N}{\text{Number of documents containing term } t} \right)$$

Term frequency–inverse document frequency (TF-IDF).

The TF-IDF score for a term in a document is obtained by multiplying its TF and IDF scores.

TF-IDF (t, d, D) = TF (t, d) × IDF (t, D) or also:

$$TF-IDF(t, d, D) = tf_w \times \log\left(\frac{1}{1 + df_w}\right)$$

The document frequency (df_w) is the number of documents where a word is seen.

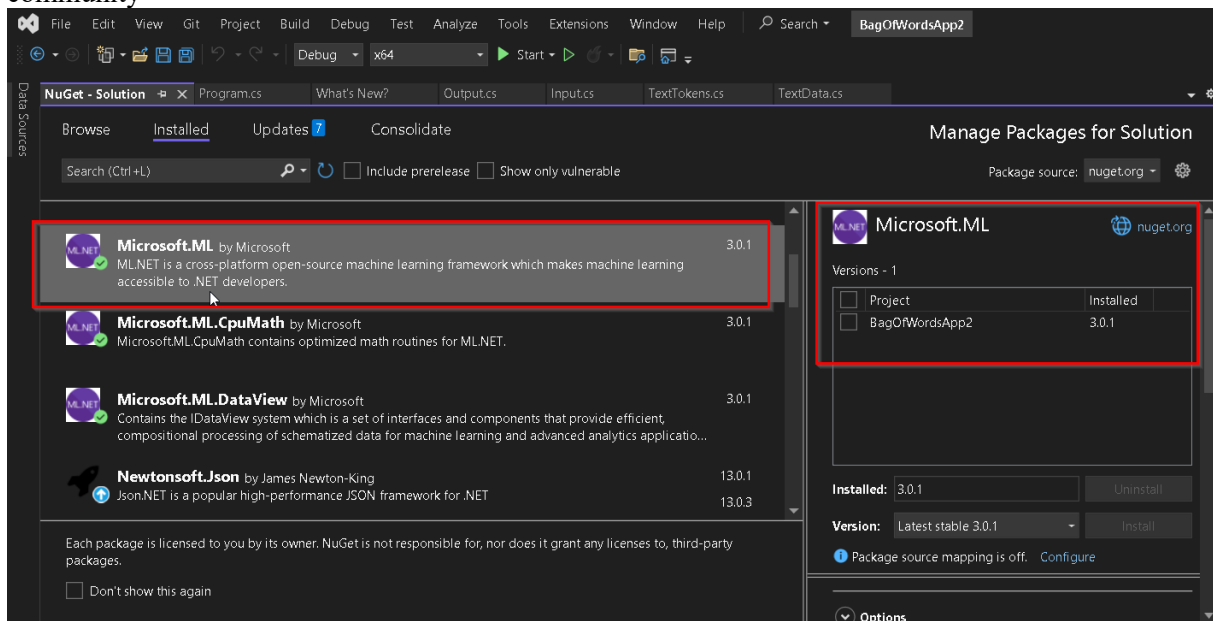
The inverse document frequency denoted by (idf) is computed by dividing the total number of documents in our corpus by the document frequency (df_w) for each term and then applying logarithmic scaling on the result. We can add 1 to the document frequency (df_w) for each term to prevent division by 0 for terms that don't exist in corpus.

Software installation

As for C# (pronounced “C Sharp”), it is worth to mention the researchers selected this language because is a general-purpose, multi-paradigm (object-oriented, structured, functional -supports lambda expressions-) programming language, fast to code, fast to execute, easy to learn language. The researchers also selected Visual Studio 2022 as IDE for C# because it has a comprehensive feature set: code editing, debugging, rich marketplace for extensions and plug-ins, friendly user interface, extensive documentation with code examples directly from Microsoft, and many other features beyond the scope of current research.

Visual Studio 2022 Community edition (free to download) was installed (Audrel, 2020) (Anandmeg, n.d.) and Nuget package Microsoft.ML 3.0.1 was also installed (Natke, n.d.) As shown on (**Figure 6**).

Figure 6. Nuget package Microsoft.ML 3.0.1 was installed after installing Visual Studio 2022 community

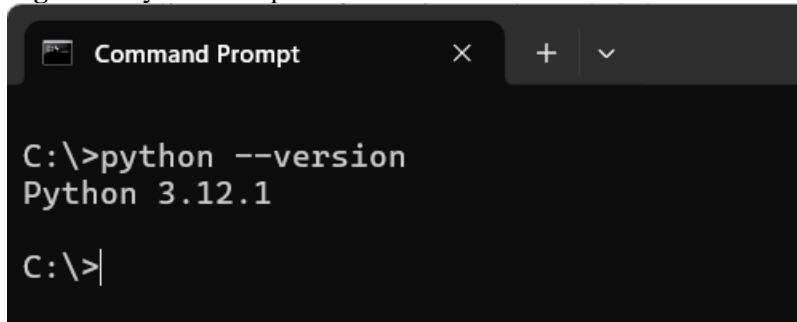


As for Python, the researchers selected this language because it is very popular for academic research, it has many ready to use libraries (packages) like Numpy for mathematical functions, Pandas for managing of datasets, scikit-learn for ML, and spaCy for NLP, Matplotlib and Seaborn for data visualizations in graphs of all kinds. Python is easy to read and write. It has a very large support community.

The reserachers selected Pycharm as IDE because it has an intelligent code editor that uses different colors to differentiate reserved python keywords from variables and function/class names. It has code autocompletion features, it has a comprehesnive debugging tool, and it supports scientific Python libraries like spaCy designed specially for NLP, the subject of current research work applied to ERP systems.

A standard python interpreter was installed first (Python.org, n.d.) because interpreter needs to be configured in PyCharm IDE to execute Python code and then PyCharm Professional Edition was installed as IDE (JetBrains, 2021) on Windows 11 computer (**Figures 7 and 8**).

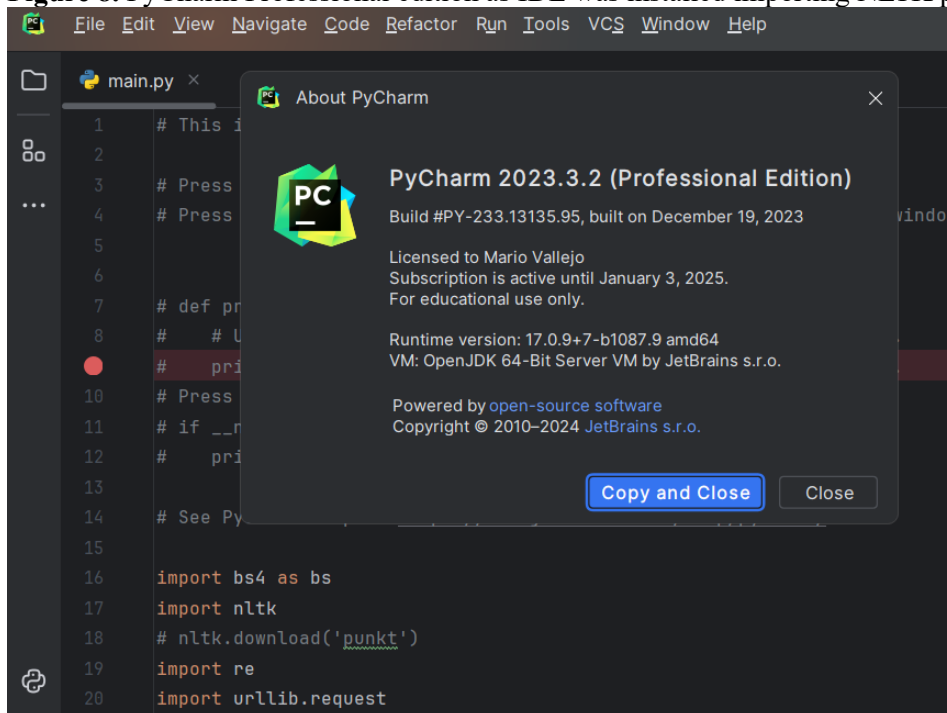
Figure 7. Python interpreter version 3.12.1 was installed



```
C:\>python --version
Python 3.12.1

C:\>|
```

Figure 8. PyCharm Professional edition as IDE was installed importing NLTK package.



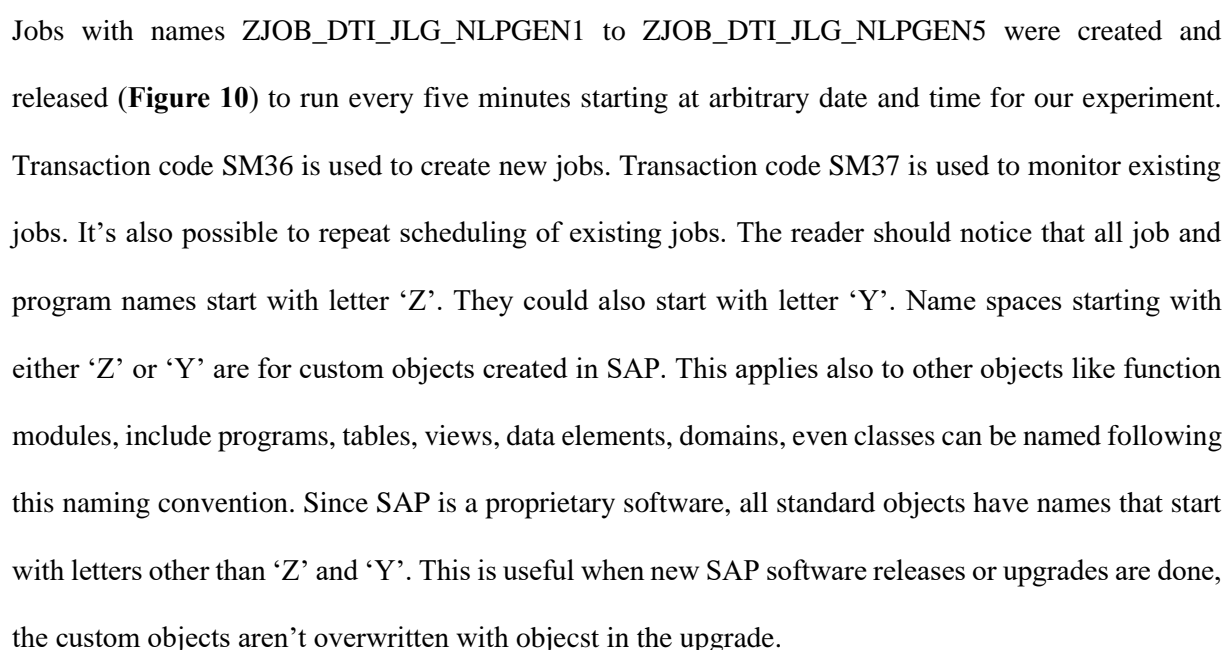
ERP SOFTWARE CONSTRUCTION

A production environment in SAP ERP with multiple job failures was simulated in order to produce a job log that could be downloaded to a local text file to be used as input for NLP packages both on PyCharm and C#. The simulation was done on a private SAP development server. Hence details about system ID, DNS names, IP addresses, programs, job names or custom objects can't be publicly shared. All the objects for this experimentation were created as local objects, meaning they are temporary objects and can't be transported to a QA nor Production SAP server.

For this purpose, several programs were created in SAP's ABAP language.

ABAP programs Z_DTI_JLG_NLPGEN1, 2, ... to Z_DTI_JLG_NLPGEN5 (**Figure 9**) are aimed to simulate jobs for different tasks running on the background of SAP system. These programs create

Figure 9. ABAP programs Z DTI JLG NLPGEN1, 2, ... to Z DTI JLG NLPGEN5.

[illegible]

failed jobs (**Figure 11**). To read job status for any failed jobs in an ABAP program we can use conditions like the following:

```
SELECT list-of-fields FROM TBTCO WHERE
```

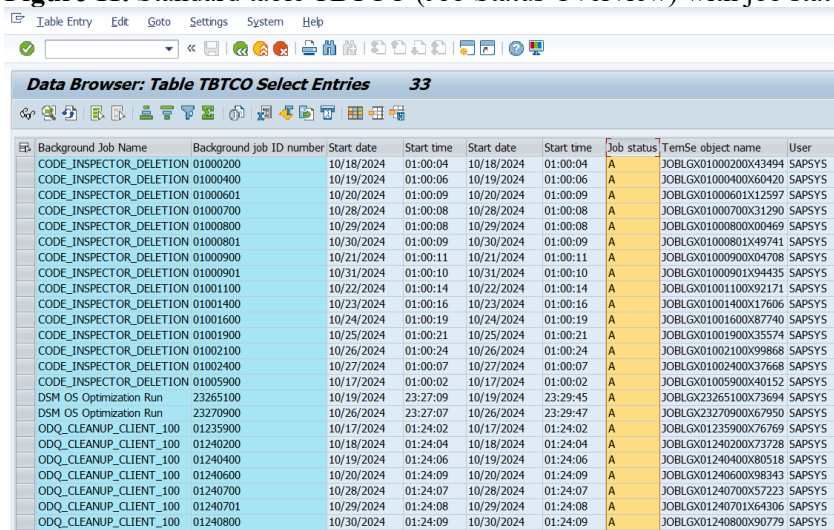
```
JOBNAME = ‘*’
```

```
STRTDATE = ‘10/04/2023’ or any start date
```

```
STATUS = ‘A’
```

```
SDLUNAME = ‘*’.
```

Figure 11. Standard table TBTCO (Job Status Overview) with job status ‘A’ for failed jobs.

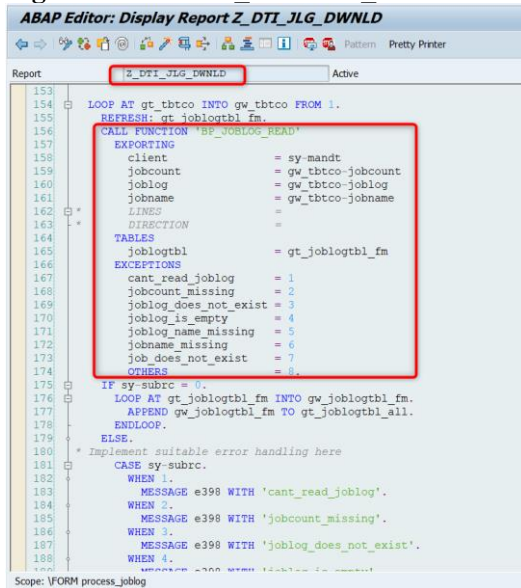


The screenshot shows the SAP Data Browser interface for the TBTCO table. The table is filtered for job status 'A'. The columns include Background Job Name, Background job ID number, Start date, Start time, Job status, TemSe object name, and User. The table contains 33 entries, all with job status 'A'.

Background Job Name	Background job ID number	Start date	Start time	Job status	TemSe object name	User
CODE_INSPECTOR_DELETION 01000200		10/18/2024	01:00:04	A	JOBLGX01000200X43494	SAPSYS
CODE_INSPECTOR_DELETION 01000400		10/19/2024	01:00:06	A	JOBLGX01000400X69420	SAPSYS
CODE_INSPECTOR_DELETION 01000601		10/20/2024	01:00:09	A	JOBLGX01000601X12597	SAPSYS
CODE_INSPECTOR_DELETION 01000700		10/28/2024	01:00:08	A	JOBLGX01000700X31290	SAPSYS
CODE_INSPECTOR_DELETION 01000800		10/29/2024	01:00:08	A	JOBLGX01000800X00469	SAPSYS
CODE_INSPECTOR_DELETION 01000801		10/30/2024	01:00:09	A	JOBLGX01000801X49741	SAPSYS
CODE_INSPECTOR_DELETION 01000900		10/21/2024	01:00:11	A	JOBLGX01000900X04708	SAPSYS
CODE_INSPECTOR_DELETION 01000901		10/31/2024	01:00:10	A	JOBLGX01000901X94435	SAPSYS
CODE_INSPECTOR_DELETION 01001100		10/22/2024	01:00:14	A	JOBLGX01001100X92171	SAPSYS
CODE_INSPECTOR_DELETION 01001400		10/23/2024	01:00:16	A	JOBLGX01001400X17606	SAPSYS
CODE_INSPECTOR_DELETION 01001600		10/24/2024	01:00:19	A	JOBLGX01001600X87740	SAPSYS
CODE_INSPECTOR_DELETION 01001900		10/25/2024	01:00:21	A	JOBLGX01001900X35574	SAPSYS
CODE_INSPECTOR_DELETION 01002100		10/26/2024	01:00:24	A	JOBLGX01002100X99868	SAPSYS
CODE_INSPECTOR_DELETION 01002400		10/27/2024	01:00:07	A	JOBLGX01002400X37668	SAPSYS
CODE_INSPECTOR_DELETION 01005900		10/17/2024	01:00:02	A	JOBLGX01005900X40152	SAPSYS
DSM OS Optimization Run 23265100		10/19/2024	23:27:09	A	JOBLGX23265100X73694	SAPSYS
DSM OS Optimization Run 23270900		10/26/2024	23:27:07	A	JOBLGX23270900X67950	SAPSYS
ODQ_CLEANUP_CLIENT_100 01235900		10/17/2024	01:24:02	A	JOBLGX01235900X76769	SAPSYS
ODQ_CLEANUP_CLIENT_100 01240200		10/18/2024	01:24:04	A	JOBLGX01240200X73728	SAPSYS
ODQ_CLEANUP_CLIENT_100 01240400		10/19/2024	01:24:06	A	JOBLGX01240400X80518	SAPSYS
ODQ_CLEANUP_CLIENT_100 01240600		10/20/2024	01:24:09	A	JOBLGX01240600X98343	SAPSYS
ODQ_CLEANUP_CLIENT_100 01240700		10/28/2024	01:24:07	A	JOBLGX01240700X57223	SAPSYS
ODQ_CLEANUP_CLIENT_100 01240701		10/29/2024	01:24:08	A	JOBLGX01240701X64306	SAPSYS
ODQ_CLEANUP_CLIENT_100 01240800		10/30/2024	01:24:09	A	JOBLGX01240800X90779	SAPSYS

The FM ‘BP_JOBLOG_READ’ is used to retrieve job logs. This FM call was put on program Z_DTI_JLG_DWNLD (**Figure 12**). The table TBTCO is used as driver table to know which are the jobs that failed for a specific date and provides key search fields like failed jobname and joblog (TemSe object name) field which is a search key with values like ‘JOBLGX01000200X43494’ that is also used by the referred FM to search for failed job logs, finally the method ‘FILE_SAVE_DIALOG’ is used to save or download job logs to a local file i.e.: the output file ‘ZDOWNLOAD_JOBLOG.TXT’ to be loaded in Python and C# NLP models (**Figure 13**). The output path in our experiment was arbitrarily defined as "E:\DATAMV\00.PhD\Python-NLP\PycharmProjects\Bag-of-Words2\ZDOWNLOAD_JOBLOG.TXT" but it could be any path available.

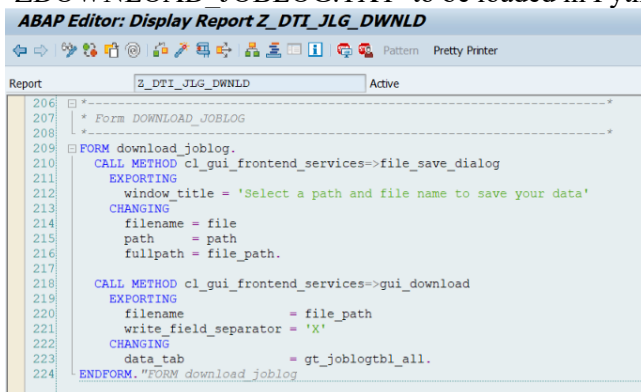
Figure 12. FM 'BP_JOBLOG_READ' is used to retrieve job logs.



```

ABAP Editor: Display Report Z_DTI_JLG_DWNLD
Report: Z_DTI_JLG_DWNLD Active
153
154 LOOP AT gt_tbtco INTO gw_tbtco FROM 1.
155 REFRESH: gt_joblogtbl_fm.
156 CALL FUNCTION 'BP_JOBLOG_READ'
157 EXPORTING
158   client           = sy-mandt
159   jobcount         = gw_tbtco-jobcount
160   joblog           = gw_tbtco-joblog
161   jobname          = gw_tbtco-jobname
162   LINES           =
163   DIRECTION        =
164   TABLES
165     joblogtbl      = gt_joblogtbl_fm
166   EXCEPTIONS
167     cant_read_joblog = 1
168     jobcount_missing = 2
169     joblog_does_not_exist = 3
170     joblog_is_empty = 4
171     joblog_name_missing = 5
172     jobname_missing = 6
173     job_does_not_exist = 7
174     OTHERS          = 8.
175 IF sy-subrc = 0.
176   LOOP AT gt_joblogtbl_fm INTO gw_joblogtbl_fm.
177     APPEND gw_joblogtbl_fm TO gt_joblogtbl_all.
178   ENDLOOP.
179 ELSE.
180   * Implement suitable error handling here
181   CASE sy-subrc.
182     WHEN 1.
183       MESSAGE e398 WITH 'cant_read_joblog'.
184     WHEN 2.
185       MESSAGE e398 WITH 'jobcount_missing'.
186     WHEN 3.
187       MESSAGE e398 WITH 'joblog_does_not_exist'.
188     WHEN 4.
189       MESSAGE e398 WITH 'joblog_is_empty'.
189   ENDCASE.
190 ENDIF.
191
Scope: VFORM process_joblog
  
```

Figure 13. Method 'FILE_SAVE_DIALOG' is used to save output file 'ZDOWNLOAD_JOBLOG.TXT' to be loaded in Python and C# NLP models.

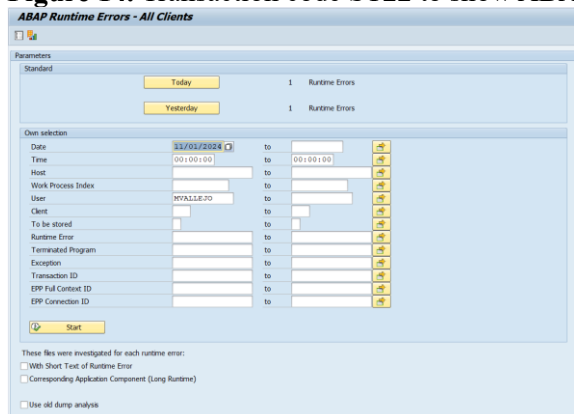


```

ABAP Editor: Display Report Z_DTI_JLG_DWNLD
Report: Z_DTI_JLG_DWNLD Active
206
207 * Form DOWNLOAD_JOBLOG
208
209 FORM download_joblog.
210   CALL METHOD cl_gui_frontend_services=>file_save_dialog
211   EXPORTING
212     window_title = 'Select a path and file name to save your data'
213   CHANGING
214     filename     = file
215     path         = path
216     fullpath     = file_path.
217
218   CALL METHOD cl_gui_frontend_services=>gui_download
219   EXPORTING
220     filename     = file_path
221     write_field_separator = 'X'
222   CHANGING
223     data_tab     = gt_joblogtbl_all.
224 ENDFORM."FORM download_joblog
  
```

Since research is incipient, and experimental, all prototype programs were created in the development environment but this could easily be implemented in production to retrieve and download real job logs for real failed jobs. Normally, when a job fails it produces an ABAP dump and it can be displayed with SAP transaction code ST22 (**Figure 14**). Such information could also be retrieved for NLP and not only the job log.

Figure 14. Transaction code ST22 to show ABAP short dumps as additional source for NLP.



Python Software Construction

An experimental python program was written in Pycharm importing NLTK library which supports the NLP models in Python (**Figure 15**). Then, Python program reads file 'ZDOWNLOAD_JOBLOG.TXT' from local path, in this case "E:\DATAMV\00.PhD\Python-NLP\PycharmProjects\Bag-of-Words2\" (**Figure 16**). Finally the input file is passed to the bag of words algorithm to produce a table of frequencies and get the bag of words (**Figure 17**).

Figure 15. Libraries imported in Python.

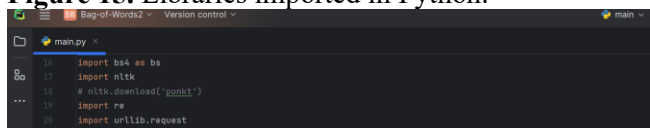


Figure 16. Input file ZDOWNLOAD_JOBLOG.TXT from SAP ERP read by Python.



Figure 17. Algorithm to create the bag of words in Python.



C# Software Construction

An experimental C# program was written in Visual Studio 2022 importing Microsoft.ML library which supports the NLP models in C# (**Figure 18**). Then, C# program reads file 'ZDOWNLOAD_JOBLOG.TXT' from local path, in this case

"E:\DATAMV\00.PhD\MicrosoftML\Projects\MicrosoftML-NLP\BagOfWordsApp2-new\BagOfWordsApp2\BagOfWordsApp2\bin\x64\Debug" (**Figure 19**). Finally the input file is passed to the bag of words algorithm to produce a table of frequencies and get the bag of words (**Figure 20**).

Figure 18. Libraries imported in C# (Micosoft.ML supports NLP methods).

```

1 using Microsoft.ML;
2 using Microsoft.ML.Data;
3 using System;
4 using System.Collections.Generic;
5 using System.IO;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using HtmlAgilityPack;
10 using Newtonsoft.Json.Linq;
11

```

Figure 19. Input file ZDOWNLOAD JOBLOG.TXT from SAP ERP read by C#.

```

26 // SOURCING FROM ERP ERROR LOG FILE
27
28 //Path E:\DATAMV\00.PhD\MicrosoftML\Projects\MicrosoftML-NLP\BagOfWordsApp2-new\BagOfWordsApp2\bin\x64\Debug
29 StreamReader streamReader = new StreamReader("ZDOWNLOAD_JOBLOG.TXT");
30 string TextDataLines = streamReader.ReadToEnd();
31 streamReader.Close();
32 Console.WriteLine("Contents of file:");
33 Console.WriteLine(TextDataLines);
34 Console.WriteLine(Environment.NewLine);
35 Console.WriteLine("Press Enter Key to Continue...");
36 while (Console.ReadKey().Key != ConsoleKey.Enter)
37 {
38 }
39
40 // End of SOURCING INPUT TEXT FILE

```

Figure 20. Algorithm to create the bag of words in C#.

```

113 var dataView = context.Data.LearnFromEnumerable(data);
114 var bagWordsPipeline = context.Transform.Text.ProduceWordBags(
115     "BagOfWords",
116     "Text",
117     ngramLength: 1,
118     useAllLengths: false,
119     weighting: Microsoft.ML.Transforms.Text.NgramExtractingEstimator.WeightingCriteria.Tf
120 );
121 var bagWordTransform = bagWordsPipeline.Fit(dataView);
122 var bagWordsDataView = bagWordTransform.Transform(dataView);
123 var predictionEngine = context.Model.CreatePredictionEngine<Input, Output>(bagWordTransform);
124 var prediction = predictionEngine.Predict(data[0]);
125 VBuffer<ReadOnlyMemory<char>> slotNames = default;
126 bagWordsDataView.Schema["BagOfWords"].GetSlotNames(ref slotNames);
127 var bagWordsColumn = bagWordsDataView.GetColumn<VBuffer<float>>("bagWordsDataView.Schema["BagOfWords"]");
128 var slots = slotNames.GetValue();
129 Console.WriteLine("Ngrams:");
130 foreach (var row in bagWordsColumn)
131 {
132     foreach (var item in row.Items())
133     {
134         Console.WriteLine($"{slots[item.Key]}:");
135     }
136     Console.WriteLine();
137 }
138 Console.WriteLine("Word Counts:");
139 foreach (var item in row.Items())
140 {
141     Console.WriteLine($"{slots[item.Key]}: {item.Value}");
142 }

```

Since research is incipient, only two NLP models were investigated for tokenization and for creating bags of words and the SAP ERP job log text file was used as input for both NLP models in both languages as explained before, Python and C Sharp. Both platforms were tested and the different technical issues were solved as they appeared until the expected results were produced.

RESULTS AND DISCUSSION

As a result, it was possible to simulate an SAP ERP production environment of a so to speak “chaotic nature”, with multiple failed background jobs (**Figure 21**), and an extractor program downloaded the job log of failures into a plain text file (**Figure 22**). The text file with the downloaded job log is shown (**Figure 23**). This work was of a medium scale complexity.

Figure 21. Transaction code SM37 showing failed jobs ZJOB_DTI_JLG_NLPGEN*.

JobName	Spool	Job doc	Job Created	Status	Start date	Start Time	Duration(sec.)	Delay (sec.)
ZJOB_DTI_JLG_NLPGEN1			11/01/2024	Canceled	11/01/2024	04:00:11	0	11
ZJOB_DTI_JLG_NLPGEN1			11/01/2024	Canceled	11/01/2024	04:05:11	0	11
ZJOB_DTI_JLG_NLPGEN1			11/01/2024	Canceled	11/01/2024	04:10:11	0	11
ZJOB_DTI_JLG_NLPGEN1			11/01/2024	Canceled	11/01/2024	04:15:11	0	11
ZJOB_DTI_JLG_NLPGEN1			11/01/2024	Canceled	11/01/2024	04:20:11	0	11
ZJOB_DTI_JLG_NLPGEN2			11/01/2024	Released	11/01/2024	04:00:11	0	11
ZJOB_DTI_JLG_NLPGEN2			11/01/2024	Canceled	11/01/2024	04:05:11	0	11
ZJOB_DTI_JLG_NLPGEN2			11/01/2024	Canceled	11/01/2024	04:10:11	0	11
ZJOB_DTI_JLG_NLPGEN2			11/01/2024	Canceled	11/01/2024	04:15:11	0	11
ZJOB_DTI_JLG_NLPGEN2			11/01/2024	Canceled	11/01/2024	04:20:11	0	11
ZJOB_DTI_JLG_NLPGEN3			11/01/2024	Released	11/01/2024	04:00:11	0	11
ZJOB_DTI_JLG_NLPGEN3			11/01/2024	Canceled	11/01/2024	04:05:11	0	11
ZJOB_DTI_JLG_NLPGEN3			11/01/2024	Canceled	11/01/2024	04:10:11	0	11
ZJOB_DTI_JLG_NLPGEN3			11/01/2024	Canceled	11/01/2024	04:15:11	0	11
ZJOB_DTI_JLG_NLPGEN3			11/01/2024	Canceled	11/01/2024	04:20:11	0	11
ZJOB_DTI_JLG_NLPGEN4			11/01/2024	Released	11/01/2024	04:00:11	0	11
ZJOB_DTI_JLG_NLPGEN4			11/01/2024	Canceled	11/01/2024	04:05:11	0	11
ZJOB_DTI_JLG_NLPGEN4			11/01/2024	Canceled	11/01/2024	04:10:11	0	11
ZJOB_DTI_JLG_NLPGEN4			11/01/2024	Canceled	11/01/2024	04:15:11	0	11
ZJOB_DTI_JLG_NLPGEN4			11/01/2024	Canceled	11/01/2024	04:20:11	0	11
ZJOB_DTI_JLG_NLPGEN5			11/01/2024	Released	11/01/2024	04:00:11	0	11
ZJOB_DTI_JLG_NLPGEN5			11/01/2024	Canceled	11/01/2024	04:05:11	0	11
ZJOB_DTI_JLG_NLPGEN5			11/01/2024	Canceled	11/01/2024	04:10:11	0	11
ZJOB_DTI_JLG_NLPGEN5			11/01/2024	Canceled	11/01/2024	04:15:11	0	11
ZJOB_DTI_JLG_NLPGEN5			11/01/2024	Canceled	11/01/2024	04:20:11	0	11

Figure 22. Running program Z_DTI_JLG_DWNLD to download ‘induced’ failed jobs.

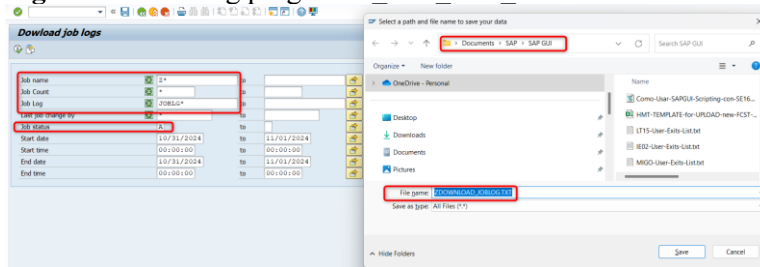


Figure 23. Example of contents of file ZDOWNLOAD_JOBLOG.TXT, the input for NLP.

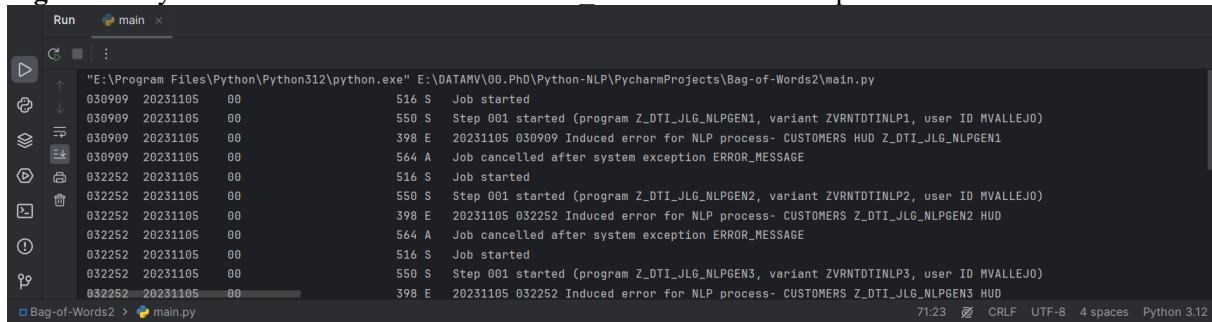
```

JobName      Job Doc      Job Created      Status      Start date      Start Time      Duration(sec.)      Delay (sec.)
-----
ZJOB_DTI_JLG_NLPGEN1      0      11/01/2024      Canceled      11/01/2024      04:00:11      0      11
ZJOB_DTI_JLG_NLPGEN1      0      11/01/2024      Canceled      11/01/2024      04:05:11      0      11
ZJOB_DTI_JLG_NLPGEN1      0      11/01/2024      Canceled      11/01/2024      04:10:11      0      11
ZJOB_DTI_JLG_NLPGEN1      0      11/01/2024      Canceled      11/01/2024      04:15:11      0      11
ZJOB_DTI_JLG_NLPGEN1      0      11/01/2024      Canceled      11/01/2024      04:20:11      0      11
ZJOB_DTI_JLG_NLPGEN2      0      11/01/2024      Released      11/01/2024      04:00:11      0      11
ZJOB_DTI_JLG_NLPGEN2      0      11/01/2024      Canceled      11/01/2024      04:05:11      0      11
ZJOB_DTI_JLG_NLPGEN2      0      11/01/2024      Canceled      11/01/2024      04:10:11      0      11
ZJOB_DTI_JLG_NLPGEN2      0      11/01/2024      Canceled      11/01/2024      04:15:11      0      11
ZJOB_DTI_JLG_NLPGEN2      0      11/01/2024      Canceled      11/01/2024      04:20:11      0      11
ZJOB_DTI_JLG_NLPGEN3      0      11/01/2024      Released      11/01/2024      04:00:11      0      11
ZJOB_DTI_JLG_NLPGEN3      0      11/01/2024      Canceled      11/01/2024      04:05:11      0      11
ZJOB_DTI_JLG_NLPGEN3      0      11/01/2024      Canceled      11/01/2024      04:10:11      0      11
ZJOB_DTI_JLG_NLPGEN3      0      11/01/2024      Canceled      11/01/2024      04:15:11      0      11
ZJOB_DTI_JLG_NLPGEN3      0      11/01/2024      Canceled      11/01/2024      04:20:11      0      11
ZJOB_DTI_JLG_NLPGEN4      0      11/01/2024      Released      11/01/2024      04:00:11      0      11
ZJOB_DTI_JLG_NLPGEN4      0      11/01/2024      Canceled      11/01/2024      04:05:11      0      11
ZJOB_DTI_JLG_NLPGEN4      0      11/01/2024      Canceled      11/01/2024      04:10:11      0      11
ZJOB_DTI_JLG_NLPGEN4      0      11/01/2024      Canceled      11/01/2024      04:15:11      0      11
ZJOB_DTI_JLG_NLPGEN4      0      11/01/2024      Canceled      11/01/2024      04:20:11      0      11
ZJOB_DTI_JLG_NLPGEN5      0      11/01/2024      Released      11/01/2024      04:00:11      0      11
ZJOB_DTI_JLG_NLPGEN5      0      11/01/2024      Canceled      11/01/2024      04:05:11      0      11
ZJOB_DTI_JLG_NLPGEN5      0      11/01/2024      Canceled      11/01/2024      04:10:11      0      11
ZJOB_DTI_JLG_NLPGEN5      0      11/01/2024      Canceled      11/01/2024      04:15:11      0      11
ZJOB_DTI_JLG_NLPGEN5      0      11/01/2024      Canceled      11/01/2024      04:20:11      0      11

```

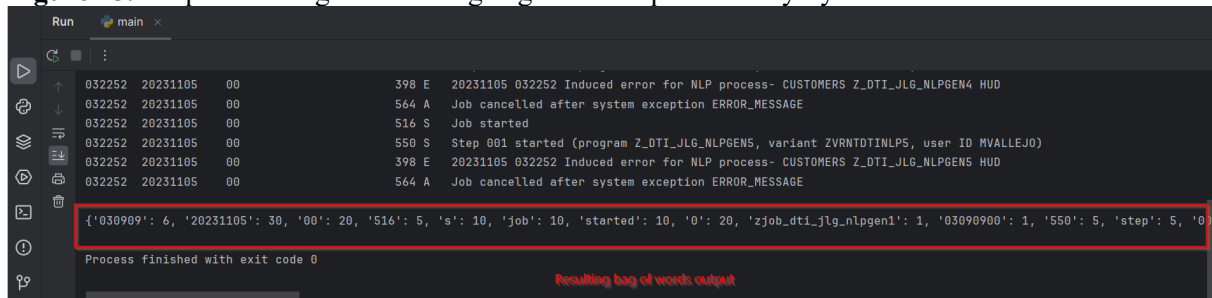
PyCharm, was also successfully installed and configured. The installation and configuration proved to be of low complexity. But, building the model for Bag of Words was a medium complexity job. The file ‘ZDOWNLOAD_JOBLOG.TXT’ was loaded by python program (**Figure 24**). The output of the process showed the resulting bag of words where for example the output string ‘zjob_dti_jlg_nlpgen1’: 1’ denotes that the word ‘zjob_dti_jlg_nlpgen1’ was counted 1 time and the word ‘030909’ was counted 6 times (**Figure 25**). The idea is to compare with the output of NLP model written in C#.

Figure 24. Python read the file ‘ZDOWNLOAD_JOBLOG.TXT’ and printed it out.



```
"E:\Program Files\Python\Python312\python.exe" E:\DATAMV\00.PhD\Python-NLP\PycharmProjects\Bag-of-Words2\main.py
030909 20231105 00 516 S Job started
030909 20231105 00 550 S Step 001 started (program Z_DTI_JLG_NLPGEN1, variant ZVRNTDTINLP1, user ID MVALLEJO)
030909 20231105 00 398 E 20231105 030909 Induced error for NLP process- CUSTOMERS HUD Z_DTI_JLG_NLPGEN1
030909 20231105 00 564 A Job cancelled after system exception ERROR_MESSAGE
032252 20231105 00 516 S Job started
032252 20231105 00 550 S Step 001 started (program Z_DTI_JLG_NLPGEN2, variant ZVRNTDTINLP2, user ID MVALLEJO)
032252 20231105 00 398 E 20231105 032252 Induced error for NLP process- CUSTOMERS Z_DTI_JLG_NLPGEN2 HUD
032252 20231105 00 564 A Job cancelled after system exception ERROR_MESSAGE
032252 20231105 00 516 S Job started
032252 20231105 00 550 S Step 001 started (program Z_DTI_JLG_NLPGEN3, variant ZVRNTDTINLP3, user ID MVALLEJO)
032252 20231105 00 398 E 20231105 032252 Induced error for NLP process- CUSTOMERS Z_DTI_JLG_NLPGEN3 HUD
```

Figure 25. Output showing the resulting bag of words produced by Python.



```
032252 20231105 00 398 E 20231105 032252 Induced error for NLP process- CUSTOMERS Z_DTI_JLG_NLPGEN4 HUD
032252 20231105 00 564 A Job cancelled after system exception ERROR_MESSAGE
032252 20231105 00 516 S Job started
032252 20231105 00 550 S Step 001 started (program Z_DTI_JLG_NLPGEN5, variant ZVRNTDTINLP5, user ID MVALLEJO)
032252 20231105 00 398 E 20231105 032252 Induced error for NLP process- CUSTOMERS Z_DTI_JLG_NLPGEN5 HUD
032252 20231105 00 564 A Job cancelled after system exception ERROR_MESSAGE

{'030909': 6, '20231105': 30, '00': 20, '516': 5, 's': 10, 'job': 10, 'started': 10, '0': 20, 'zjob_dti_jlg_nlpgen1': 1, '03090900': 1, '550': 5, 'step': 5, '0': 10}

Process finished with exit code 0
```

Alternatively, for comparison purposes, the Visual Studio 2022 IDE and Microsoft.ML Nugget packages were installed. Models were also built for Tokenization and for Bag of Words, which was a medium-high complexity job.

Visual Studio 2022 IDE and Microsoft.ML Nugget package, were also successfully installed and configured. The installation and configuration proved to be of low complexity. But, again (as in Pycharm), building the model for Bag of Words was a medium complexity job. The file ‘ZDOWNLOAD_JOBLOG.TXT’ was loaded by C# program (**Figure 26**). The output of the process showed the resulting bag of words, The layout of the output is slightly different from the output produced by Pycharm, but the values are the same. The line with the label ‘Ngrams:’ shows all the words while the line with label ‘Word counts:’ shows the number of times a given word was found. For example the output string ‘zjob_dti_jlg_nlpgen1’ (second red color box on ‘Ngrams’ row) denotes that the word ‘zjob_dti_jlg_nlpgen1’ was counted 1 time (on the line for ‘Word count:’) and the word ‘030909’ (on the row for ‘Ngrams:’) was counted 6 times on the line for ‘Word count:’ (**Figure 27**). It is shown that results were the same in Python than in C#.

Figure 26. C# read the file 'ZDOWNLOAD_JOBLOG.TXT' and printed it out.

```

032252 20231105 00 516 S Job started
      0 0 ZJOB_DTI
      03190200
032252 20231105 00 550 S Step 001 started (program Z_DTI_JLG_NLPGEN, variant ZVR
      0 001
      001
      Z_DTI_JLG_NLPGEN
      MVALLEJO
      ZVRNTDTINLP5
032252 20231105 00 398 E 20231105 032252 Induced error for NLP process- CUSTOMERS
      0 20231105
      Z_DTI_JLG_NLPGEN
      HUD
      Induced error for NLP process- CUSTOMERS
      HUD
032252 20231105 00 564 A Job cancelled after system exception ERROR_MESSAGE
      0
Press Enter Key to Continue...

```

Figure 27. Output showing the resulting bag of words produce by C#.

```

E:\DATAMV\00.PhD\Microsoft x + v
0
Press Enter Key to Continue...
NGrams: 030909-20231105-00-516-s-job-started-0-zjob_dti_jlg_nlpgen1-03090900-550-step-001-program-z_dti_jlg_nlpgen1-varia
nt-zvrntdtinlp1-user-id-mvallejo-398-e-induced-error-for-nlp-process-customers-hud-564-a-cancelled-after-system-exceptio
n-error_message-032252-zjob_dti_jlg_nlpgen2-03182400-z_dti_jlg_nlpgen2-zvrntdtinlp2-zjob_dti_jlg_nlpgen3-03183700-z_dti_
jlg_nlpgen3-zvrntdtinlp3-zjob_dti_jlg_nlpgen4-03184600-z_dti_jlg_nlpgen4-zvrntdtinlp4-zjob_dti_jlg_nlpgen5-03190200-z_dt
i_jlg_nlpgen5-zvrntdtinlp5-
Word Counts: 6-30-20-5-10-10-20-1-1-5-5-10-5-4-5-2-5-5-10-5-5-10-10-10-10-10-10-10-5-5-5-5-5-24-1-1-4-2-1-1-4-2-1-
1-4-2-1-1-4-2-
Press Enter Key to Finish...

```

The interconnection of technological platforms was feasible and of medium difficulty, although we did not

try to automate the download of data from the SAP ERP system or to automate the loading of data into the NLP models because we wanted to give priority to the implementation of the logic of the NLP models. However, the automation of the interface is a must in order to achieve an optimization of the software and to eliminate the need manual data loads into the NLP models.

CONCLUSIONS

The first conclusion reached is that it is definitely feasible and relatively easy to provide artificial intelligence to classic ERP systems, even if they are traditional systems, with client-server architecture and may seem very conventional as they are already very proven technologies and well known by the IT industry. Some persons might even think that ERP systems are a thing of the past and that the new artificial intelligence technology cannot be easily applied to them, which is false.

The second conclusion is that it is more difficult to determine what type of artificial intelligence is most useful to apply to ERP systems. It depends on the problem to be solved. Trying to predict the consumption of products sold by a company, or trying to analyze dozens or hundreds of purchase

contracts, or measuring customer satisfaction, or even automating IT staff tasks is not the same need, and therefore there will not be a single artificial intelligence applicable to all problems.

The third conclusion is that there are definitely programming languages that have specialized in artificial intelligence and are therefore easier to use. For example, it was easier to implement NLP models in Python than in C#. Because more lines of code are written in C# than in Python because Python is higher level programming language. However C# is faster to run, because it is compiled and in particular the Visual Studio 2022 IDE is easier to use because it has code auto-completion functions (predictive text) and a wizard that suggests how to solve syntax errors. It is also very easy to get help on the statements used in the programs.

The fourth conclusion is that the work presented here is incipient and further work is needed on more complex models in both NLP and ML. In the case of NLP, further research on the NER (Named Entity Recognition) model is suggested. Research on the applicability of Large Language Models (LLM) in ERP systems is also suggested. Such models are the basis of more advanced artificial intelligences such as ChatGPT. Also, it is suggested as future research work going deeper in exploring and experimenting with models like Naïve Bayes with Bag of Words for classification of texts from ERP system failure logs. The idea would be to recognize relevant error messages, job or program names, days, months or times when failures are produced, etc., from ERP system failure logs and program system to suggest a corrective action.

BIBLIOGRAPHIC REFERENCES

- Anandmeg. (n.d.). *Install visual studio and choose your preferred features*. and choose your preferred features | Microsoft Learn. <https://learn.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2022>
- Audrel, Carolline & Wijaya, Vellicia & Azwir, Hery. (2020). *Information System Development Using Microsoft Visual Studio to Speed Up Approved Sample Distribution Process*. Journal of Industrial Engineering. 5. 14-24. 10.33021/jie.v5i1.1268.
- Beysolow II, T. (2018). What Is Natural Language Processing? In T. Beysolow II (Ed.), *Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning*



- Algorithms for Natural Language Processing* (pp. 1–12). Apress. https://doi.org/10.1007/978-1-4842-3733-5_1
- Chow, R. (2021, September 30). *Dartmouth Summer Research Project: The Birth of Artificial Intelligence*. History of Data Science. <https://www.historyofdatascience.com/dartmouth-summer-research-project-the-birth-of-artificial-intelligence/>
- Edwards, G. (2020, January 21). *Machine Learning | An Introduction*. Medium. <https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0>
- Ghosh, T., & Kumar, S. (2022). Chapter 11 Natural Language Processing. In *Practical Mathematics for AI and Deep Learning: A Concise yet In-Depth Guide on Fundamentals of Computer Vision, NLP, Complex Deep Neural Networks and Machine Learning* (English Edition) (pp. 456–458). BPB Publications.
- IBM. (2023, June 14). *What is supervised learning?*. IBM. Retrieved June 14, 2023, from <https://www.ibm.com/topics/supervised-learning>
- IBM. (2023, June 14). *What is unsupervised learning?*. IBM. Retrieved June 14, 2023, from <https://www.ibm.com/topics/unsupervised-learning>
- IBM. (2023, Jun 14). *What is reinforcement learning?*. IBM. <https://developer.ibm.com/learningpaths/get-started-automated-ai-for-decision-making-api/what-is-automated-ai-for-decision-making/>
- JetBrains. (2021, June 2). Download PyCharm: The python IDE for data science and web development by jetbrains. <https://www.jetbrains.com/pycharm/download/?section=windows>
- McCarthy, J. (1970, January 1). *What is AI? / Applications of AI*. What is AI / Applications of AI. Retrieved June 11, 2023, from <http://jmc.stanford.edu/artificial-intelligence/what-is-ai/applications-of-ai.html>
- McCarthy, J. (1970, January 1). *What is AI? / Basic Questions*. What is AI / Basic Questions. Retrieved June 11, 2023, from <http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html>
- Mahesh, B. (2018). *Machine Learning Algorithms—A Review*. 9(1).
- Natke. (n.d.). *ML.NET Documentation - tutorials, API reference*. Tutorials, API reference | Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/machine-learning/>

Python.org. (n.d.) *Download python*. <https://www.python.org/downloads/>

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>

Thorwirth, Z. (2021, September 1). *AI Winter: The Highs and Lows of Artificial Intelligence*. History of Data Science. <https://www.historyofdatascience.com/ai-winter-the-highs-and-lows-of-artificial-intelligence/>

