EFFICIENCY AND PREDICTION IN HUMAN RESOURCE MANAGEMENT USING PYTHON MODULES

Mihai ANDRONICEANU
Ministry of Transport and Infrastructure, Romanian Railway Authority, Bucharest, Romania
androniceanu.mihai@afer.ro

Abstract
Python has become an increasingly popular platform due to its ability to automate processes, analyze data, and improve the efficiency of human resources management (HR) operations. Python can assist in enhancing recruitment processes, employee performance evaluation, employee satisfaction analysis, and more. Python is a highly popular programming language, known for its simple syntax and versatility, which is why it was chosen for this research. The research aimed to provide HR managers with an analysis model for organizational employee mobility. The specific objectives were: (1) identifying the correlations between research variables and their impact on employee mobility; (2) identifying the main causes of employee mobility; (3) ranking the variables with the most impact on human resources fluctuation; (4) developing predictions regarding the stability of human resources in positions and functions within an organization. The research is a pilot study conducted within an organization using Python modules based on data from the year 2024. The research results show the significant influence on employee mobility exerted by factors such as the level of education, seniority, and age of employees within the sample. Another useful outcome for HR managers is the predictive model obtained with the help of Python modules, which allows them to both analyze and predict the profile of employees with a higher degree of stability within the organization. The research demonstrates how various artificial intelligence applications can be integrated into Python-specific modules to enhance human resource management and organizational efficiency.
Keywords: human resources management; Python modules; stability of the human resources; mobility, fluctuation, artificial intelligence.

1. INTRODUCTION

Python is an essential programming language in today's technological landscape due to its accessibility, active community, a vast ecosystem of modules and libraries, and its versatility across various fields. Its utility in applying Python modules makes this language a powerful tool for data analysis and the development of solutions based on artificial intelligence, machine learning, and process automation (Harrison & Chen, 2024). In the field of knowledge, Python allows researchers and professionals to extract valuable insights from large datasets and develop algorithms capable of solving complex problems, making it indispensable in data science, software development, and many other fields of research and application (Oliphant, 2007). Python is a high-level, interpreted, and easy-to-learn programming language that has become essential in various domains, from software development to data science, artificial intelligence, and process automation. In an ever-expanding digital age, where the volume and complexity of data are growing exponentially (Androniceanu, M, 2024a, 2024b), Python provides a set of extremely powerful and flexible tools that facilitate the analysis and processing of this data. In the current context of

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

technology and research, Python has evolved from a general-purpose programming language to a dominant platform in the field of data science and artificial intelligence application development (Miller & Jones, 2018). This is largely due to its accessibility, the extensive community of users and developers, and its vast ecosystem of open-source libraries and modules. Python is considered the reference language for data analysis, thanks to powerful libraries such as Pandas (for data manipulation), NumPy (for numerical calculations), and Matplotlib or Seaborn (for data visualization). Python is widely used in developing machine learning algorithms and deep learning models, thanks to libraries like sci-kit-learn, TensorFlow, and Keras. These libraries are used to build predictive models, classification, regression, and other machine-learning techniques. In automation and scripting (Bishop, 2020). Python is frequently used to automate repetitive tasks, such as file processing, data manipulation, and integration with external APIs. Due to its easy-to-learn syntax and support for various libraries, Python is often preferred by IT professionals for workflow automation (Pedregosa & Varoquaux, 2023). One of Python's main advantages is its vast ecosystem of modules and libraries, which significantly extend the capabilities of the language, allowing it to be applied in numerous fields. Python modules are collections of functions, classes, and methods that are already optimized to solve specific problems. These modules are essential for improving productivity and reducing the complexity of development processes. The flexibility and accessibility of Python modules allow developers to apply them to a wide range of applications, from statistical analysis to web application development or artificial intelligence systems (Androniceanu, A, 2024a; 2024b; Lazaroiu et al., 2022). Thus, Python modules provide a standardized set of solutions for common problems, saving time and resources. Another major advantage of Python is its active community. There are numerous open-source modules available, which are continuously developed and maintained by programmers from around the world. This means that users can quickly find solutions to their problems, contribute to improving existing libraries, and utilize the latest innovations in the technology field (Androniceanu et al., 2020). The relevance of Python Modules for the Knowledge Domain and Data Programming is proven by their widespread use for a variety of diverse analyses. In the field of knowledge and scientific research, Python and its modules are essential for managing and analyzing data. Today, in many scientific disciplines, from computational biology and astronomy to social sciences and neuroscience, data plays a central role. Python enables researchers to perform complex data analyses and develop algorithms that can extract valuable insights from these large and complex datasets. Important modules include: (1) Pandas: Used for data manipulation, Pandas provides an extremely efficient data structure for working with tabular data, such as CSV files or SQL databases. It allows researchers to perform complex data manipulations in a highly intuitive manner. (2) SciPy: A powerful module for performing advanced mathematical calculations, including integrals, optimization, and statistics. It is frequently used in research for mathematical modeling and data simulation. (3) TensorFlow and Keras: These modules are essential for researchers working in deep learning and artificial intelligence, enabling them to build and train complex neural networks. In the field of data programming, Python is indispensable due to its ability to efficiently and quickly manage and analyze large amounts of data. In an era of Big Data, where the volume of available information is enormous, Python provides the necessary tools for processing, cleaning, analyzing, and visualizing data.

2. PYTHON MODULES IN THE LITERATURE

In the specialized literature, Python modules are a fundamental concept for organizing and reusing code in a Python application (Buitinck & Louppe, 2021). A module in Python is a file that contains a set of functions, classes, and variables that can be used in other Python files via import. Modules allow developers to structure programs more efficiently by breaking down the code into smaller, manageable, and reusable parts. In international literature, many key authors have studied and used Python modules, especially in fields such as software development, data analysis, computer science, and machine learning (Kelleher, 2020). Although there are no authors solely focused on the study of Python modules, we have selected the most important and relevant authors from the literature who have had a significant impact on the use and promotion of Python, including its modules.

The first and most important contributor through his work is Guido van Rossum, the true creator of Python. He is the chief architect of the language and its core libraries. In addition to creating Python, van Rossum was the principal architect of its evolution, as well as of the associated core libraries. Although van Rossum has not written many theoretical papers or books like other authors, he had a significant impact through official documentation and his contributions to the development of Python and its ecosystem. Guido van Rossum began developing Python in the 1980s, and the first version of the language was released in 1991 (van Rossum & Drake, 2009). The reason he created Python was the desire to develop a language that was simple to use and allowed for fast writing of clear and easy-to-maintain code. Python was inspired by the ABC programming language, an experimental language, but Guido wanted to create a more powerful, versatile, and accessible language. Guido van Rossum actively contributed to writing many PEPs (Python Enhancement Proposals), many of which relate to Python's core modules and libraries. One important example is PEP 8, which is the official style guide for writing Python code. It includes rules for indentation, variable naming, and code structure, all contributing to the creation of a consistent and clear language. Guido van Rossum has written about Python as a language for the future, emphasizing its flexibility and versatility, as well as its importance in fields such as data science, web development, process automation, and machine learning. Another researcher with remarkable contributions to data analysis is Wes McKinney (2010), who focuses on the use of modules such as Pandas, NumPy, Matplotlib, and scikit-learn for data analysis. Wes McKinney is an extremely important name in the field of data science and Python programming, being particularly recognized for the development of essential libraries like Pandas. Pandas revolutionized the way data is manipulated and

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

analyzed in Python, and McKinney's work had a significant impact on the data science community. Pandas is an open-source Python library used for data manipulation and analysis. Wes McKinney began developing Pandas in 2008 while working on a financial data analysis project at AQR Capital Management. Pandas was created to address the limitations of other data processing libraries available at the time, particularly for handling tabular data (such as CSV files, Excel, and SQL databases). Pandas was built on top of libraries like NumPy and introduced two essential data structures: DataFrame and Series, which are widely used in data analysis. These structures allow for efficient data manipulation, complex aggregations, and the fast processing of large datasets. Although McKinney is better known for his software contributions, his work has also been supported by research in the field of data science, particularly related to the efficient processing and analysis of large data sets. Throughout his career, McKinney (2021) has participated in the development of solutions for fast data manipulation, especially exploring how rapid and efficient operations can be performed on large datasets with minimal resource consumption. Jake VanderPlas is a renowned data science expert, professor, author, and active contributor to the Python community. VanderPlas is especially known for his contributions to data analysis, and machine learning, and for promoting the use of Python in data science. He is an active advocate of open-source tools and a leading educator in the field of data science. VanderPlas authored a reference book that had a significant impact on learning and applying data science with Python. VanderPlas has been a consistent advocate for Python as the preferred language for data science, thanks to its strong community and vast ecosystem of available libraries. He also championed tools like Jupyter, which transformed the way researchers and developers interact with data and code.

David Beazley also made remarkable contributions to the field of Python modules knowledge with his books Python Essential Reference and Python Cookbook, which include information on Python's standard modules and their advanced usage. David Beazley is a key figure in the world of programming, particularly in the Python community. He is known for his significant contributions to the development of Python, as well as for his research and work in areas such as distributed systems, concurrent programming, performance analysis, and programming language design. Beazley has had a profound impact on the way Python is used, especially in the context of advanced software applications, and his work has been extremely influential both in education and in industry. He contributed to the development of efficient synchronization techniques and the improvement of performance in concurrent programming contexts, aiming to make Python more efficient in handling multiple tasks simultaneously (Beazley & Jones, 2013). He explored various ways to improve the efficiency of Python applications, including using profilers and debugging tools. Beazley also contributed to documenting and developing some of the most useful techniques for performance analysis, including the use of modules like timeit, cProfile, and line_profiler. In addition to his contributions to Python, Beazley is known for his research in the area of compilers and programming languages. He has conducted studies and work on the development and optimization of programming languages, including creating interpreters and compilers for various applications. In the specialized literature, Python modules are a fundamental concept for organizing and reusing code in a Python application. Modules allow developers to structure programs more efficiently by splitting the code into smaller, more manageable, and reusable parts. In international literature, many key authors have studied and utilized Python modules, especially in fields such as software development, data analysis, computer science, and machine learning. Although there are no authors who are exclusively focused on studying Python modules, we have selected the most important and relevant authors who have had a significant influence on the use and promotion of Python, including its modules (Pylab, 2014).

Mark Lutz has also made significant contributions to the field of Python knowledge with his books, such as Learning Python and Programming Python, which are extremely valuable from a scientific perspective. Lutz has had a significant influence on how Python is learned and applied by developers, and his works are considered essential for anyone wanting to deeply understand the Python language. His books are considered a reference material in the literature. He created a comprehensive guide that helps readers build a solid foundation in Python. Lutz explores more complex topics, such as object-oriented programming, using advanced modules, working with databases and networks, and developing graphical user interfaces (Lutz, 2010). Lutz has been an advocate for the use of Python in the software industry and has highlighted its applicability for rapid prototyping, automation, database systems, and even enterprise applications (Lutz, 2020).

Co-creator of IPython Fernando Pérez (Pérez & Granger, 2022) made significant contributions to the interactive package widely used in scientific research and education. He has made remarkable contributions to the development of some of the most widely used Python modules. Fernando Pérez is an important researcher and developer in the Python community, especially known for his contributions to the development of IPython (which later evolved into Jupyter), an essential tool for Python programmers and researchers in data science. His work has had a significant impact on education and scientific research because IPython and Jupyter have become reference platforms for interactive data analysis, data visualization, and presenting results in an easily shareable format. In 2014, Fernando Pérez and his team took an important step by splitting the IPython project into two components: IPython (which remained an interactive shell for Python) and Jupyter, an interactive tool independent of the Python language that can support multiple programming languages (such as R, Julia, Scala, and others). Jupyter Notebooks quickly became a standard in education, research, and data science due to their ability to combine code, graphics, and explanatory text in a single document that is easy to share. Jupyter allows researchers and developers to create and share interactive notebooks, where they can include documentation, experiments, and data analysis all in one place. In the specialized literature, Hillary Parker stands out for exploring the use of Python and its modules in data science. Hillary Parker is a renowned expert in the field of data science, with a strong focus on data analysis and the use of the Python language for this

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

purpose. She is known both for her work in data science and for her educational and community contributions. Hillary Parker has played an essential role in promoting best practices in data analysis and the use of Python in the data industry. She has been an active member of the data science community and a strong advocate for using Python to solve complex business and scientific problems. Hillary Parker is an active promoter of best practices in data science, including how to write clean, efficient code and build reproducible data analysis models. She emphasizes the importance of understanding code clarity and collaborating effectively in multidisciplinary teams. At DataCamp, a renowned educational site for interactive data science courses, Hillary Parker has led several courses designed for those who want to learn how to apply Python in data analysis, addressing both fundamental concepts and advanced aspects of the analysis process.

According to the literature, Allen B. Downey introduced programming concepts and the use of Python modules in an accessible way (Downey, 2015). Allen B. Downey is a professor, researcher, and renowned author in the field of computer science, with a special focus on programming education and the use of Python for data science and analysis. Another experienced researcher with outstanding contributions is Sebastian Raschka, author of the book Python Machine Learning, which details essential Python modules for machine learning, such as scikit-learn, TensorFlow, Keras, and NumPy. Sebastian Raschka (2017) is an expert in data science and machine learning, known for his contributions to popularizing the use of Python in these fields. He is an author, researcher, and educator, and is appreciated for the educational resources he has created to help students and professionals learn advanced machine learning and data science techniques using Python. Sebastian Raschka also addresses popular Python libraries such as scikit-learn, TensorFlow, and Keras, making him a valuable resource for those starting in machine learning, as well as for those who want to dive deeper into advanced topics. Sebastian Raschka has published the results of several research papers and academic articles in the field of machine learning, being an active contributor to the scientific literature in this domain (Raschka & Mirjalili, 2020). His research has covered topics such as probabilistic modeling, optimization of machine learning algorithms, and model performance evaluation. Additionally, he has been involved in developing techniques and methodologies for improving predictions and the accuracy of machine learning models. He has also worked on applications of machine learning in various fields, including computational biology and natural language processing (NLP).

These authors are just a few of those who have made significant contributions to the Python literature, including the Python module. Although they are not specifically focused solely on modules, their work has profoundly influenced the use of modules in various fields.

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

3. RESEARCH FRAME BASED ON PYTHON MODULES

According to the literature, Python modules require a sequential approach. Some of the most important sequences followed for conducting this research with applicability in the field of human resource management are presented and explained below.

3.1. Setting the purpose, objectives, variables, and hypotheses of the research

The purpose of the research is to create an analysis and prediction program using Python modules for optimizing personnel placement in positions and functions and for reducing human resource mobility within an organization.

The main objectives of the research are: (1) identifying the correlations and the intensity (importance) of the variables impacting both job placement and increasing the stability of human resources in an organization; (2) identifying the factors and causes that influence human resource mobility; (3) ranking the variables with the greatest importance regarding human resource mobility; (4) developing a prediction model for the essential requirements of the employee profile that an organization needs to enhance stability and reduce human resource mobility.

The main research hypotheses are:

Hypothesis 1: Employee mobility is higher among young employees with a high level of education.

Hypothesis 2: Income, age, and years of work negatively influence mobility within the organization.

Hypothesis 3: Based on predictions regarding the necessary employee profile within an organization, employee mobility can be reduced.

The main variables included in the research underlying this work are: age, years in organization, education, gender, professional rank, field of activity/department, adjusted salary, and marital status.

3.2. The database used in the research

The database contains data and information specific to the human resources field over a time span of one year. For this purpose, the Python libraries Matplotlib and Seaborn (Table 1) were used for data related to employee mobility, demographic information, salaries, etc., which helped identify the factors contributing to employee mobility and implement strategies to improve retention.

The component within the organizational structure, coded as follows: ADTIV - Administrative; CONTA - Accounting; INSTA - Installations; LAB - Laboratory; REG - Registers; RSH - Research; SIT - Inspection; HR - Human Resources; TEH - Technical. Column 3: Professional rank (1-12), representing the role or function of the employee. Column 4: Adjusted monthly salary (in rons). Column 5: Entry coded as 1 (Yes) or exit of personnel coded as 0 (No). Column 6: Education level from 1 (basic education) to 6 (higher education, including doctorate). Column 7: Gender of the person (M - Male; F - Female). Column 8: Marital status: Married - C, coded as 1/Unmarried - N, coded as 0. Column 9: Years of work.

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

TABLE 1 - THE FRAME OF THE DATABASE

No.	Age	Organizational component	Professional degree	Revenue (adjusted in rons)	Entry/ Exist	Training level	Gender	Marital status	Years of work
0	1	2	3	4	5	6	7	8	9
1	34	SIT	11	5000	1	4	M	C	9
2	29	CONTA	10	4500	0	3	M	N	3
3	41	RSH	4	4000	1	1	M	N	16
4	36	HR	7	5000	0	4	M	N	5
5	28	ADTIV	9	5500	1	3	M	C	3
6	28	CONTA	8	5000	1	3	F	N	3
7	56	TEH	11	5500	1	4	M	C	31
8	62	TEH	11	5600	0	4	M	C	5
9	65	TEH	4	4500	0	2	M	C	23
10	54	ADTIV	8	6000	0	3	M	C	5
11	28	ADTIV	2	3700	0	2	M	N	3
12	34	ADTIV	1	3900	0	2	F	C	5
13	32	TEH	9	6000	1	4	M	C	7
14	40	TEH	11	7500	1	4	M	N	15
15	45	SIT	10	7000	1	4	M	C	20
16	65	SIT	11	7500	0	5	M	C	25
17	47	INSTA	10	7000	1	4	M	C	22
18	35	INSTA	9	6700	1	4	M	C	10
19	28	LAB	7	5000	1	4	M	C	3
20	30	LAB	8	5500	1	4	M	C	5
21	62	SIT	10	6750	0	4	M	C	20
22	34	LAB	10	5500	1	4	M	C	9
23	28	LAB	7	4500	1	4	M	N	3
24	45	INSTA	10	6500	1	4	M	C	20
25	63	INSTA	11	7500	1	4	M	C	26
26	26	LAB	7	7000	1	4	M	N	1
27	50	INSTA	10	7500	1	4	M	C	25
28	62	REG	11	7000	0	4	M	C	20
29	59	REG	10	6500	1	4	M	C	29
30	45	REG	9	6000	1	4	M	N	30
31	28	ADTIV	2	3500	1	1	M	N	3
32	30	ADTIV	6	4000	1	3	M	C	5
33	28	ADTIV	4	3700	1	2	F	C	3

Legend: Column 1: Age in years of the person who entered/left an organizational component. Column 2: *Elaboration of UML Modules (Unified Modeling Language)*

Among the most popular software tools are the following: (1) Microsoft Visio, which allows the creation of flowcharts, network diagrams, and more, and is used for representing model architecture, data flows, and processes. (2) Lucidchart, an online diagramming tool that enables real-time collaboration and is useful for creating architecture diagrams, flowcharts, and UML diagrams. (3) Draw.io, an online diagramming tool that offers a wide range of templates and shapes and is used to create architecture diagrams, flowcharts, and more. (4) TensorBoard, a visualization tool for TensorFlow that allows monitoring and visualizing training graphs and is used for visualizing the architecture of neural networks and model performance during training. (5) Keras Visualization Tools, for visualizing model architecture, which allows the visualization and interpretation of neural network architectures created with Keras.

In this research, the Lucidchart software was used, which allowed the creation of the following types of UML diagrams: (1) Structure diagrams; (2) Behavior diagrams; (3) Interaction diagrams; (4) Architecture diagrams: These provide an overview of the system's architecture, and (5) Package diagrams, which show how classes and components are grouped into packages and the relationships between these packages.

3.3. Importing Python modules

In the specialized literature, modules are described as a central element in Python programming, being essential for creating scalable and modular applications. They are crucial for organizing code and supporting efficient collaborative development. The example in the next section shows how they were applied and what the main results obtained were. Python includes a set of standard modules that cover a wide range of functionalities, such as file manipulation, working with date/time data, networking, and more. In addition to the standard modules, there are many external modules that can be installed via the pip package manager. For this research, the modules that offer the best functionalities for the stated research goals and objectives were identified. These were used to develop solutions that can grow and evolve with the needs of the users and the system. Python modules are robust and allow data analysis and visualization, predictive model development, and process automation, thus contributing to informed and efficient decision-making. In this research, they helped in causal correlation analysis and predictions regarding the employee profile that an organization needs to reduce human resource mobility.

3.4. The development of the program using UML modules for conducting the research

The development of a program using UML involved clearly defining the system's structure and behavior through diagrams. This helped in its design and development. UML facilitated the design of the software system before implementation, making it easier to understand and allowing for changes during the development process.

3.5. Conducting research in the field of human resources by applying Python modules

Exploring and visualizing data are essential for understanding the distribution and trends in the dataset. After exploring the data, the next step was to apply statistical methods to draw more precise conclusions or to test hypotheses. Hypothesis testing was done using modules like scipy.stats, and statistical tests were applied to verify the hypotheses. Regression and correlations were also determined using the statsmodels and scikit-learn modules, which were used to calculate correlations and apply linear regression techniques or other statistical methods. Machine learning models were trained using modules such as scikit-learn, tensorflow, keras, or pytorch. Using modules like matplotlib, seaborn, plotly, or bokeh charts were created to observe relationships between variables and identify patterns.

3.6. Analysis of results

The analysis of the results obtained using Python modules involved processing and interpreting the data, applying statistical methods, evaluating and interpreting the performance of the models, and communicating the results through reports and clear visualizations.

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

Theoretical and Empirical Researches in Urban Management

Volume 20 Issue 1 / February 2025

3.7. Formulation of the main conclusions

In this final part of the research process, the main findings were summarized, the results were interpreted, and recommendations or suggestions were formulated, based on the data and models analyzed during the research.

In the following section of the paper, the main results of the research are presented and analyzed, focusing on the theme of employee mobility within an organization.

4. RESEARCH RESULTS AND DISCUSSIONS

The following presents and analyzes some of the most important results obtained from the pilot study conducted within an organization, based on data collected from it in 2024. Using the data from Table 1, the relationship between employee mobility and the main variables was analyzed using the Matplotlib and Seaborn modules. The main results are presented in Figure 1.

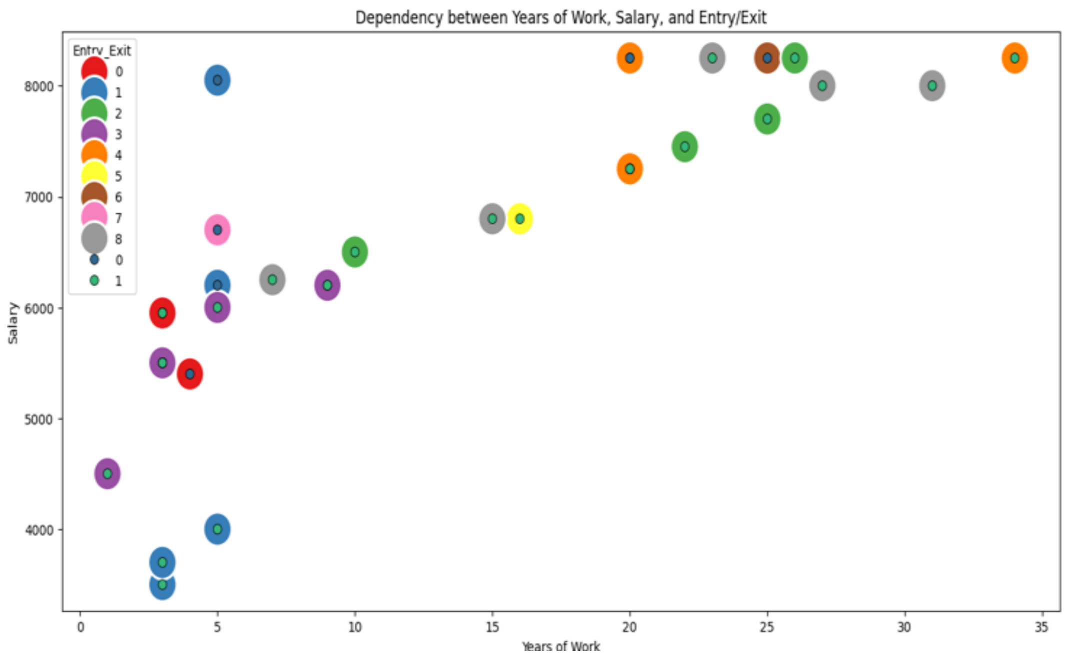

FIGURE 1 - THE RELATIONSHIP BETWEEN SALARY, SENIORITY, AND EMPLOYEE MOBILITY

Entries/Exits are highlighted by the inner point (the small diameter circle within the larger diameter circle), with exits marked in blue and entries in green. The correspondence between colors and the organizational structure is as follows: ADTIV – Red(0), CONTA – Blue(1), INSTA – Green(2), LAB – Purple(3), REG – Orange(4), RCH – Yellow(5), SIT – Brown(6), HR – Light Purple(7), TEH – Gray(8).

Figure 2 contains an analysis of employee mobility based on age and professional qualifications. As can be observed, there is a high mobility among employees aged between 25-30 years. These results confirm the first research hypothesis.

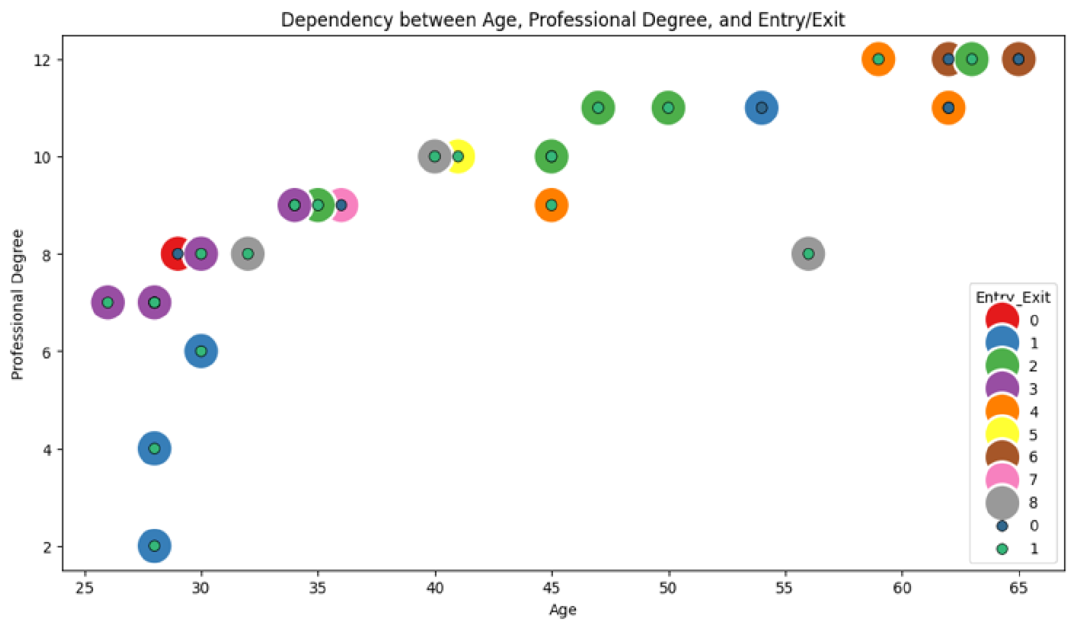

FIGURE 2 - CORRELATIONS BETWEEN, AGE AND PROFESSIONAL DEGREE

The results shown in Figure 3 indicate that the main cause of employee mobility was the level of income earned. Using Python modules, a correlation matrix (heatmap) was created. The intensity of the correlations between the variables represented on the two axes is highlighted by the two colors: blue and its shades, which signify a weak correlation, and red and its shades, which signify strong correlations.

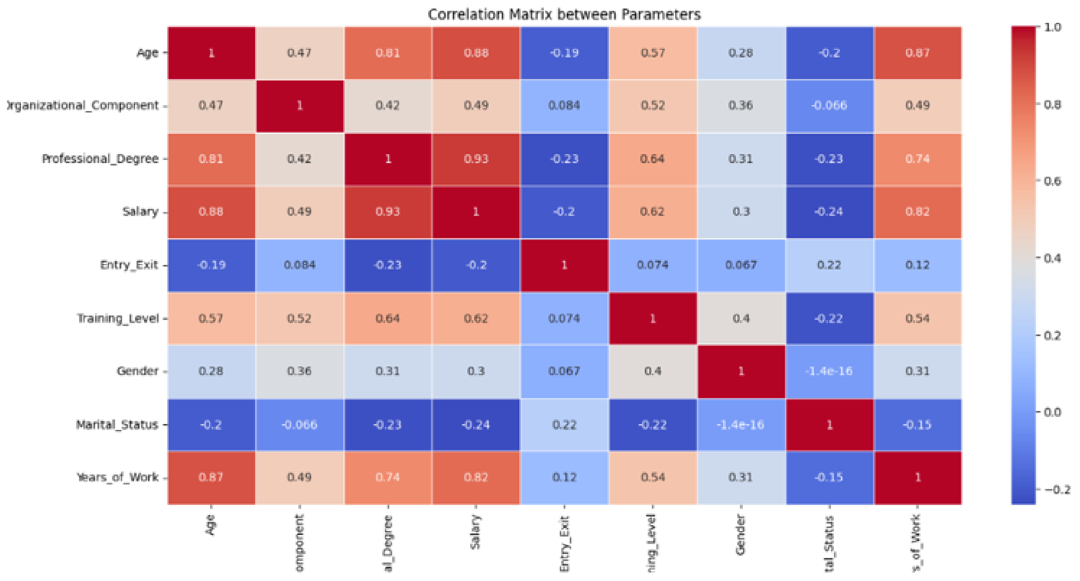

FIGURE 3 – CORRELATION MATRIX BETWEEN PARAMETERS

As shown in Figure 3, there are several types of correlations between the variables included in the research. The most relevant are the following: (1) the positive correlation between age and seniority, meaning that as age increases, work seniority tends to increase as well; (2) the positive correlation between monthly income and education level, meaning that individuals with higher education levels tend to have higher income, as they possess more advanced qualifications and skills; (3) the positive

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

correlation between age and monthly income, as older individuals tend to have higher salaries due to accumulated experience and higher positions they occupy; (4) the positive correlation between seniority and monthly income, as individuals with longer work period tend to have higher salaries due to their experience and loyalty to the organization; (5) the correlation between education level and seniority may vary. The results show that employees with higher income, age, and seniority within the organization have lower mobility, which confirms hypothesis number 2. Correlations show only that there is a relationship between two variables, but they do not indicate whether one causes the change in the other. To determine causality, regression analysis and path analysis were used. Regression is a statistical technique that helped identify causal relationships, while path analysis is an extension of regression analysis that allows the simultaneous examination of causal relationships between multiple variables. Linear regression showed how an independent variable influences a dependent variable. To determine the causes of employee mobility, two regression analysis models were used. The modeling was done using the scikit-learn Python module. The model accuracy is 0.86, meaning that the model correctly predicted 86% of cases. This is a good accuracy, but it shows that the model can be improved either by adjusting the parameters or by using a larger and more diverse dataset. Thus, the classification of the parameters (Table 2) that influence mobility was obtained.

TABLE 2 –CLASSIFICATION OF VARIABLES THAT INFLUENCE FLUCTUATION

No	Professional degree	Gender	Organizational component	Marital status	Monthly salary	Age	Years of work	Training level
	0.719472	0.501909	0.354049	0.328505	-0.000387	-0.032526	-0.090928	-0.601872

According to the results in the table, variables with positive values have a greater influence on mobility, while variables with negative values have a smaller influence.

Another result of the research relates to the importance and impact of the considered variables on human resource mobility. Figure 4 shows the importance of each parameter in predicting entries and exits from the organization. Parameters with higher values are more important in the prediction model. The data can be used in the decision-making process to reduce the causes that lead to employee mobility.

Next, in the path analysis, the statsmodels module was used to track the relationships between independent variables and a dependent variable. The regression coefficients were determined based on the data included in the research. Regression coefficients represent the size and direction of the relationship between the independent variables and the dependent variable. In a linear regression model, the regression coefficients indicate the expected change in the dependent variable for a one-unit change in the independent variable, holding other variables constant. In the statsmodels program, p-values are used to test the statistical significance of the regression coefficients.The analysis of the results shows that (1) age is a significant predictor of monthly income and professional rank, while other variables are not; (2) age and seniority are not significant predictors of education level.To improve the model, the database was completed with new data. Next, as part of the research process, a program was created to generate predictions based on the initial input and output data.Thus, a prediction of mobility was made based on the education level (professional rank) and seniority of the human resources, as shown in Figure 5.

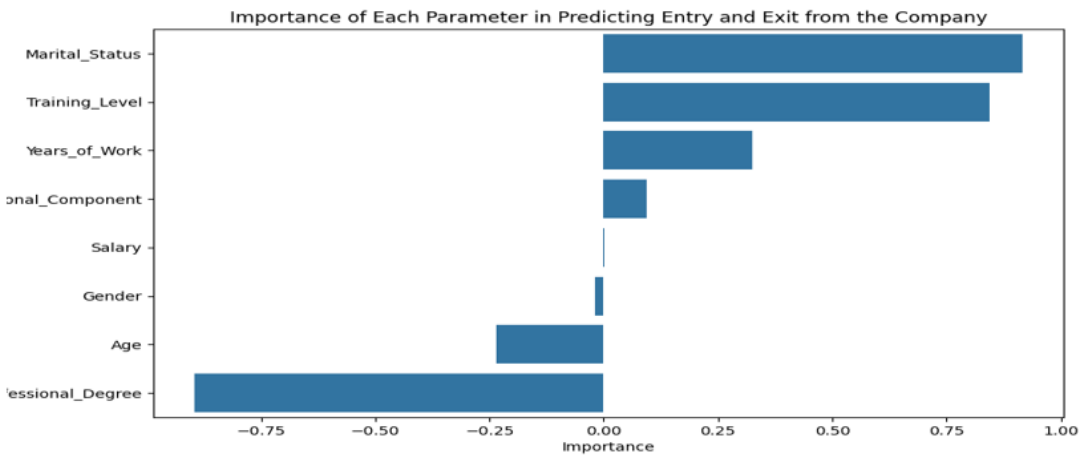

FIGURE 4 - RANKING OF FACTORS INFLUENCING HUMAN RESOURCES MOBILITY

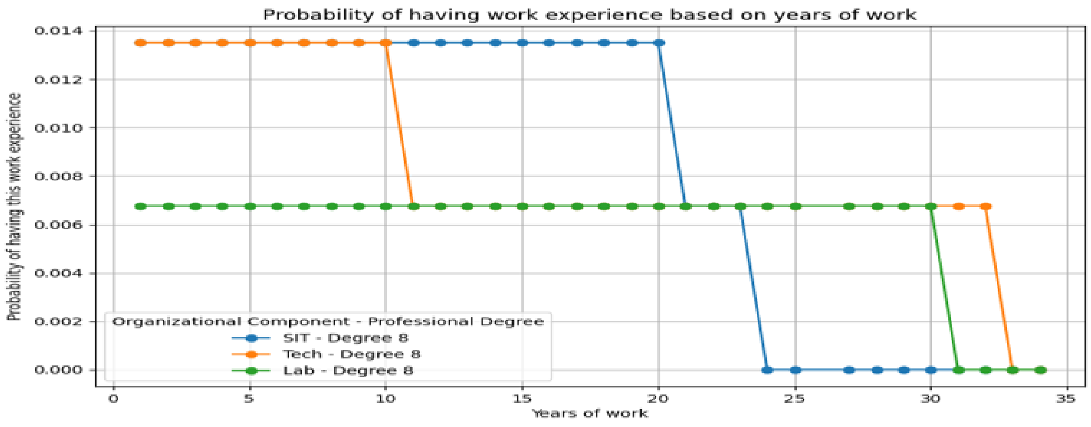

FIGURE 5 - PREDICTION OF EMPLOYEE STABILITY BASED ON PROFESSIONAL LEVEL

The probability distribution varies for different combinations of organizational components and professional ranks. This result suggests that certain structures and ranks tend to have employees with a certain level of seniority. Different organizational components and professional ranks show distinct trends in terms of the average duration of employee seniority. For example, some structures may have a higher probability of having employees with fewer years of seniority, while others may have a higher probability of having employees with more years of seniority. The distribution of seniority years also reflects the fact that certain combinations of organizational components and professional ranks may be more or less attractive to employees. Different organizational components and professional ranks are represented by different lines. This result allows for the comparison of trends between them. Structures and professional ranks that have high probabilities of having employees with greater seniority may reflect greater stability among employees in these positions. Components with lower probabilities but greater seniority could benefit from interventions or retention strategies to reduce employee mobility. These results confirm

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

hypothesis 3. The chart provides a clear visualization of how seniority years vary depending on the combination of organizational components and professional rank. This can help identify trends and areas that require improvements to enhance employee stability and retention within the organization.

The main limitations of the research are the limited database, the one-year period over which the research was conducted; and the small number of Python modules used in the pilot study. These aspects will be taken into account in future research.

5. CONCLUSIONS

As the results of this research conducted using Python modules on employee mobility in an organization show, the level of training, seniority, and salary of employees in the analyzed organization significantly influence the phenomenon studied. In the research on employee mobility analysis, the use of Python modules proved to be essential to create an efficient program that correlates the parameters that influence employee mobility and ranks the main factors. Python modules, such as pandas, numpy and scipy, allowed for complex statistical analyses to identify correlations between various parameters (e.g. salary, working conditions, advancement opportunities) and employee mobility. The use of these modules facilitated the processing and manipulation of large data sets, contributing to obtaining accurate and relevant results.

Libraries such as matplotlib and seaborn facilitated the creation of graphs and diagrams that illustrate the relationships between the analyzed variables. These provided a clear and intuitive view of the data, helping to identify trends and patterns relevant to organizational decisions. The use of Python modules allowed the development of algorithms for classifying and ranking factors that influence staff mobility. For example, scikit-learn was used to implement machine learning models that helped prioritize factors according to their impact on human resources mobility. Python is recognized for its flexibility and adaptability. This aspect was reflected in the ability to quickly and efficiently adjust programs according to new requirements or available data. Python modules allowed for continuous updating and improvement of the implemented solutions.

The results obtained through this research are in line with those obtained by other researchers. For example, a study conducted by the University of California, Berkeley, demonstrated how Python can be used for big data analysis in HR, identifying trends and patterns that influence human resources mobility. This study emphasizes the importance of data analysis for making informed decisions in HR. Another study used Python to develop a machine learning model that identifies factors that contribute to employee mobility and suggests preventive measures.

The results of the study showed how Python can be used to create predictive solutions in HR. Case studies on the use of Python in HR are well-known in the literature: A case study published in the Harvard

Business Review explored how different organizations implemented Python-based solutions to manage employee mobility. This study highlighted the success of these approaches in reducing fluctuation rates and increasing employee satisfaction. Unlike these studies, our research combined both big data analysis and ranking factors that influence employee mobility. While other studies focused on identifying specific factors or using machine learning models, our research integrated both aspects to obtain a more complete and detailed picture of employee mobility.

The added value of this research lies in its ability to provide organizations with a holistic and integrated solution that not only identifies the factors contributing to mobility but also prioritizes them based on their impact. This allows for more informed and effective decision-making, helping to maintain organizational stability and performance.

REFERENCES

Androniceanu, A. (2024a). Generative artificial intelligence, present and perspectives in public administration. *Administratie si Management Public*, 43, 105-119. https://doi.org/10.24818/amp/2024.43-06

Androniceanu, A. (2024b). Artificial intelligence in administration and public management. *Administratie si Management Public*, 42, 99-114. https://doi.org/10.24818/amp/2024.42-06

Androniceanu, A., Sabie, O.M., and Pegulescu, A., (2020). An Integrated Approach of the Human Resources Motivation and the Quality of Health Services, *Theoretical and Empirical Research in Urban Management*, *15*(1), 42–53.

Androniceanu, M., (2024a). The Alfresco platform, a viable and sustainable strategic option for document management. *Management Research and Practice*, *16*(1), March, 46-54.

Androniceanu M., (2024b). Integrated document management system using the Alfresco platform for contracting communication and mobile phone services. *Management Research and Practice*, *16*(2), June, 36-47.

Beazley, D. M., & Jones, B. K. (2013). Python Cookbook: Recipes for mastering Python 3. O'Reilly Media.

Bishop, C. M. (2020). Pattern recognition and machine learning with Python. Journal of Machine Learning Research, 21(1), 1-25. https://doi.org/10.1016/j.jmlr.2020.01.004

Buitinck, L., & Louppe, G. (2021). Reusable machine learning workflows with Python. *Journal of Open Source Software,* 6(59), 2542. https://doi.org/10.21105/joss.02542

Downey, A. B. (2015). Think Python: How to think like a computer scientist (2nd ed.). Green Tea Press.

Harrison, J. S., & Chen, L. (2024). Python for data-driven decision-making in business analytics. *International Journal of Data Science and Analytics, 18*(1), 45-63. https://doi.org/10.1007/s41060-024-00286-3

Kelleher, J. D. (2020). Introduction to machine learning with Python: Practical guides and solutions. *AI Open,* 3(2), 127-136. https://doi.org/10.1002/aiopen.2020.30

Lazaroiu, G., Androniceanu, A., Grecu, I., Grecu, G., and Negurita, O. (2022). Artificial intelligence-based decision-making algorithms, Internet of Things sensing networks, and sustainable cyber-physical management systems in big data-driven cognitive manufacturing. *Oeconomia Copernicana*, 13(4), 1047-1080. https://doi.org/10.24136/oc.2022.030.

Theoretical and Empirical Researches in Urban Management
Volume 20 Issue 1 / February 2025

Lutz, M. (2020). Learning Python: A comprehensive guide to mastering Python for data science and machine learning. *Computers in Industry*, 125, 103339. https://doi.org/10.1016/j.compind.2020.103339

Lutz, M. (2010). Python programming: An introduction to computer science. *Journal of Computing Sciences in Colleges*, 25(3), 8-17. https://dl.acm.org/doi/10.5555/2047692.2047697

McKinney, W. (2021). Python for data analysis: Practical use of Python in modern data science. *Journal of Data Science,* 19(4), 227-235. https://doi.org/10.6339/jdsci.2021.19.4.001

McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56. https://doi.org/10.25080/majora-92bf1922-007

Miller, T., & Jones, R. (2018). Python in artificial intelligence: A survey of its usage and applications. *Artificial Intelligence Review,* 43(2), 234-245. https://doi.org/10.1007/s10462-017-9642-8

Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9(3), 10–20. https://doi.org/10.1109/MCSE.2007.58

Pedregosa, F., & Varoquaux, G. (2023). Practical machine learning with Python: A tutorial on using scikit-learn. *Journal of Artificial Intelligence Research*, 75, 49-71. https://doi.org/10.1613/jair.1.12167

Pérez, F., & Granger, B. E. (2022). Advancements in data analysis with Python: Pandas and NumPy. *Journal of Computational and Graphical Statistics,* 31(3), 585-600. https://doi.org/10.1080/10618600.2021.1873129

Pylab, P. (2014). Data science with Python: How to perform data analysis with Python. *Journal of Data Science*, 12(2), 1-17. https://doi.org/10.6339/JDSCI.2014.12.2.001

Raschka, S., & Mirjalili, V. (2020). Python machine learning: Fundamentals and techniques. *Neural Computing and Applications,* 32(7), 1357-1369. https://doi.org/10.1007/s00542-019-05064-5

Raschka, S. (2017). Machine learning with Python: A guide for beginners. *IEEE Access*, 5, 9057-9068. https://doi.org/10.1109/ACCESS.2017.2738838

van Rossum, G., & Drake, F. L. (2009). Python 3.0 language reference. *ACM Sigplan Notices*, 44(8), 36-37. https://doi.org/10.1145/1594188.1594217