© University Press 2024
ISSN 2719-6550
ISSN 2719-7417 online

"Journal of Education, Technology and Computer Science"
No. 5(35)/2024
www.eti.ur.edu.pl

Received: 25.09.2024
Accepted for printing: 11.12.2024
Published: 20.12.2024

DOI: 10.15584/jetacomps.2024.5.7

PAWEŁ DYMORA[1], **MIROSŁAW MAZUREK**[2], **ŁUKASZ SMYŁA**[3]

A Comparative Analysis of Selected Data Mining Algorithms and Programming Languages

[1] ORCID ID: 0000-0002-4473-823X, Rzeszow University of Technology, Faculty of Electrical, and Computer Engineering, Department of Complex Systems, Poland
[2] ORCID ID: 0000-0002-4366-1701, Rzeszow University of Technology, Faculty of Electrical, and Computer Engineering, Department of Complex Systems, Poland
[3] ORCID ID: 0009-0006-7788-351X, Rzeszow University of Technology, Faculty of Electrical, and Computer Engineering, Poland

Abstract

This paper evaluates the performance of ten selected data mining algorithms in the context of classification and regression and the effectiveness between two popular programming languages used in data science: Python and R. The algorithms included in the study were Naive Bayes Classifier, K-Nearest Neighbors (k-NN), Support Vector Machine (SVM), Decision Tree, Random Forest, Gradient Boosting Machine (GBM), Logistic Regression, Linear Regression, Ridge Regression, and LASSO Regression. The study aimed to evaluate how the various algorithms perform in classification and regression tasks in the context of a specific problem, in this case fraud detection. The performance of the algorithms was evaluated based on key metrics such as accuracy, execution time, the difference between the best and worst results, and in terms of mean square error (MSE). Moreover, learning tools such as R and Python enable students not only to perform multidimensional data analysis, but also to predict future trends and changes. The ability to work with data, modelling and visualisation are key competences in the context of many areas of modern life and to support the making of accurate business decisions.

Keywords: data mining algorithms, Python, R, accuracy, mean square error

Introduction

Nowadays, our lives are inextricably linked to the vast amount of data generated by various systems and applications. This increase in information creates not only challenges but also incredible opportunities to extract valuable insights that are of great importance both in business and in literally every aspect of our

lives. In today's world, the ability to effectively analyze data has become a key skill for making more accurate business decisions and optimizing the operation of various systems. One of the key tools in this area is data mining, a process that enables the detection of hidden patterns and relationships in data sets. It is worth noting that we also encounter data mining algorithms on various platforms daily. An example is Netflix, which analyzes our preferences and offers personalized recommendations for movies and series based on them. The same is true for online ads, which target our interests based on previous product browsing. These algorithms are also present in social media, providing us with personalized content, suggested contacts, or customized ads. While the concept of data mining may seem abstract and complicated, the reality is that everyone has already encountered its impact on everyday life. Although the term may not be commonly used in the media, the impact that data mining algorithms have is undeniable. Analyzing big data brings many challenges, such as capturing, storing, analyzing, searching, sharing, transmitting, visualizing, querying, and updating, as well as protecting privacy and the source of the data. Analyzing datasets makes it possible to discover new relationships, identify business trends, prevent disease, fight crime, and many other applications.

The article presents the functioning of selected algorithms, showing their importance and inevitable presence in our lives. Considering the potential for future applications, it is safe to say that data mining is a key element of the future, opening up new prospects for both business and society as a whole. Both R and Python are free open-source tools, making them readily available to students worldwide. Their popularity in scientific research, medicine, epidemiology and business data analysis means that knowing how to use them is becoming a valuable competency for students in terms of their competitiveness in the job market.

Data Mining

Data mining is a research process that aims to uncover hidden patterns and information in vast data sets. This interdisciplinary field uses techniques from machine learning, statistics, and database systems to extract valuable insights from data. It is a key step in data analysis, intending to transform raw data into an understandable structure, ready for further analysis and use. The main purpose of data mining is to perform semi-automatic or automatic analysis of large amounts of data to identify previously unknown significant patterns. These patterns can include clustering of data records (cluster analysis), detection of unusual records (anomaly detection), and identification of relationships between data (association rule mining, sequence patterns) (Han, Kamber, Pei, 2011; Fayyad, Piatetsky-Shapiro, Smythm 1996).

To achieve this, database techniques such as spatial indexes are often used. It is worth noting that data collection, data preparation, and interpretation of

results and reporting are not directly related to the data mining process itself, but can be additional steps in the overall process of extracting knowledge from data (KDD). There is a difference between data analysis and data mining. Data analysis typically focuses on testing models and hypotheses on an available dataset, for example, in the context of evaluating the effectiveness of a marketing campaign, regardless of the amount of data. Data mining, on the other hand, uses machine learning techniques and statistical models to uncover hidden patterns in large amounts of data, which exceeds the capabilities of traditional data analysis (Olson, 2007; Deepali, 2013).

In research practice, different methods of model evaluation are often used to select the best model. One popular technique is comparative model evaluation, which involves using different methods for the same data and selecting the best performing model, or building a complex model using several techniques. A key component of this step is model evaluation and combination techniques, which have a significant impact on the effectiveness of the predictive model being developed. Some examples of such techniques are (Hastie, Tibshirani, Friedman, 2009):

1. **Model aggregation (bagging):** This technique involves combining multiple models that are trained on different subsets of the data to produce more stable and accurate predictions. In bagging, the results from different models are often averaged or upvoted to help reduce variance and improve prediction quality.
2. **Reinforcement/Boosting:** Reinforcement is a technique that involves sequentially building weak models, each focusing on improving the errors of the previous model. By focusing on the harder-to-predict cases, amplification leads to an improvement in overall prediction quality. Popular amplification methods include AdaBoost and Gradient Boosting, for example.
3. **Model contamination (stacking, stacked generalizations):** In this technique, the predictions of different models are combined using an additional model called a meta-model. The meta-model is trained on the prediction results of other models, which can lead to even better prediction results by taking advantage of the diversity of predictions.
4. **Meta-learning):** This approach is based on adapting the model to different data sets by analyzing past results and adjusting learning strategies. Meta-learning can help automatically adapt the model to new data, which can lead to better generalization and higher prediction performance.

Selected data mining algorithms

There are many algorithms used in data mining. The article discusses in detail such algorithms as the Naive Bayes Classifier, K-Nearest Neighbors, SVM, Decision Tree, Random Forest, GBM, Logistic Regression, Linear Regression, Ridge Regression, and LASSO Regression (Feng, Pan, Jiafu Daqiang, Athanasios, Xiaohui, 2015; Dymora, Mazurek, Jucha, 2023).

Classification algorithms

In statistics and machine learning, classification is the process of assigning observations to specific categories or classes based on their characteristics or features. This can be something as simple as marking an email as spam or non-spam, to more complex tasks such as diagnosing a disease based on a patient's medical data. In practice, each observation is analyzed for a variety of characteristics, which can be categorical (like blood type), ordinal (like size), numerical (like the number of occurrences of a word in an e-mail), or real (like blood pressure measurement). Classification can be based on a single feature or a combination of multiple features. Classification algorithms vary in their approach but generally work by comparing observations with others that have already been classified, using similarity or distance functions. These algorithms can be based on statistics, such as Logistic Regression, or on machine learning techniques, such as Decision Trees or Neural Networks (Hastie, Tibshirani, Friedman, 2009; Feng, Pan, Jiafu Daqiang, Athanasios, Xiaohui, 2015).

1. Naive Bayes Classifier

The history of machine learning runs deep in time and surprisingly in simplicity. By going back to the roots, we discover fundamental assumptions that are still a key part of today's methods. One of these milestones is Bayes' theorem – named after the 18th-century English mathematician Thomas Bayes. This theorem, seemingly modest in its form, opens the door to deep considerations of probabilistic inference and classification. It is a technique used to solve the problem of sorting decision classes, where the task of the Bayes classifier is to assign a new case to one of the classes, while their set must be finite and defined a priori (Feng-Jen, 2018; Berrar, 2019).

2. K-Nearest Neighbors

As with the history of machine learning, the concepts of K-Nearest Neighbors go deep back in time, with an equally surprising simplicity. So let's go back to the roots to uncover the underlying assumptions that are still the foundation for today's methods. One of those cornerstones is K-Nearest Neighbors, a technique that has revolutionized the way machines learn and process data. The history of K-Nearest Neighbors dates back to the 1950s when it was first introduced by Evelyn Fix and Joseph Hodges in 1951. The algorithm was later extended by Thomas Cover. The basic premise of the K-Nearest Neighbors algorithm is that similar cases should be associated with each other. The method is an example of lazy learning, meaning that there is no training phase for the model – it is constantly busy storing training data. When a new data point arrives, the K-Nearest Neighbors algorithm analyzes the training set to find the k closest points (neighbors) and makes a prediction based on that. The basis of the K-Nearest Neigh-

bors algorithm is the calculation of distances between data points in the feature space (Zhang, 2016; Mucherino, Papajorgji, Pardalos, 2009).

3. Support Vector Machine (SVM)

The history of the SVM algorithm dates back to the early 1960s when Vladimir Vapnik and Alexey Chervonenkis introduced the concept of VC-dimension and the concept of statistical learning theory. However, it was not until the 1990s that the SVM algorithm gained popularity, largely due to the work of Vapnik and his colleagues at AT&T Bell Labs. Their research led to the development of SVM as an effective tool for high-performance classification, especially in cases where the number of features exceeded the number of samples. The introduction of the kernel trick concept allowed SVM to be applied to nonlinear data, greatly expanding its applications (Srivastava, Bhambhu, 2010; Zhang, 2012).

4. Decision Tree

The history of Decision trees dates back to the 1960s when the method was first described by prominent researchers in the field of machine learning. The method has evolved over the years, with advances in computer technology and increasing applications in various fields. The origins of the Decision Tree can be traced to the 1960s when Ross Quinlan developed the ID3 (Iterative Dichotomiser 3) algorithm, which was one of the first Decision Tree algorithms. Quinlan later developed this algorithm into a series of more advanced methods, such as C4.5 and CART (Classification and Regression Trees) (Feng *et al.*, 2015; Lee, Cheang, Moslehpour, 2022).

5. Random Forest

Over the years, the Random Forest algorithm has covered many key stages of development, from its initial concepts in the 1990s to its recognition as one of the most important algorithms in the field of machine learning. Initially, the random decision forest method was proposed by Tin Kam Ho in 1995. He introduced the random subspace method, which in his view was a way of implementing the "stochastic discrimination" approach proposed by Eugene Kleinberg. The method involved using random subspaces of features to build Decision Trees, intending to reduce correlations between trees and improve the overall accuracy of the model. The development of this method was continued by Leo Breiman and Adele Cutler, who registered the trademark "Random Forests" in 2006. Breiman, using Ho's earlier work and his concept of bagging (bootstrap aggregating), introduced a technique that combined random feature selection and bagging to create sets of Decision Trees with controlled variance. Breiman and Cutler's contribution to the development of the Random Forest algorithm was crucial, as they combined various earlier ideas into a coherent methodology that

has gained immense popularity in the world of machine learning (Schonlau, Zou, 2020; Angshuman, Mukherjee, Das, Gangopadhyay, Chintha, Kundu, 2018).

6. Gradient Boosting Machine (GBM)

The Gradient Boosting Machine's origins date back to the 1990s, when Jerome H. Friedman published his paper on the gradient boosting technique. Initially used for regression, the algorithm was later developed and adapted to classification problems by Leora Breiman and Adele Cutler in 1997. Their work, which described the Gradient Boosting Machines algorithm as a technique for combining multiple Decision Trees to improve prediction quality, helped popularize the approach in the world of machine learning. Gradient Boosting Machine (GBM) is a powerful machine learning algorithm applied to both classification and regression problems. Its strength lies in its ability to create complex predictive models by combining multiple weaker models to improve prediction quality. Compared to other algorithms such as Random Forests or Decision Trees, GBM is known for its unique ability to adapt to training data through iterative error correction, making it one of the most widely used tools in the field of machine learning. GBM was quickly recognized as one of the most successful machine learning methods due to its ability to deal with different types of data and its flexibility to adapt to different problems. Through continuous improvement and adaptation, the Gradient Boosting Machine has become an indispensable item in every data scientist's toolbox (Lu, Mazumder, 2020; Natekin, Knoll, 2013).

Regression algorithms

Regression analysis is a comprehensive set of statistical methods that help understand and describe the relationships between different variables. The main purpose of regression analysis is to study how one variable, called the dependent variable, affects other variables, called independent variables or predictors. Regression analysis has two main applications. First, it is used to predict future values of the dependent variable based on known values of the independent variables. This is very similar to what is done in machine learning, where models are built to make predictions based on training data. Second, in some cases, regression analysis can be used to infer possible causal relationships between variables, although this aspect requires caution and additional analysis. It is important to note that regression analysis in itself does not confirm causality between variables but only describes the relationships between them in a given data set (Montgomery, Peck, Vining, 2021).

1. Linear Regression

Linear Regression is one of the simplest and most widely used algorithms in statistics and machine learning. Its primary purpose is to model the relationship

between a dependent variable (y) and one or more independent variables (x). It is widely used for forecasting and in many places gives sufficient results. The task of linear regression is simply to fit a straight line to the data. It is worth noting that linear regression assumes that the relationship between the characteristics and the explanatory variable is more or less linear (Seber, Lee, 2012).

2. Logistic regression

Logistic Regression was introduced as a response to the need to model binary variables. Its origins can be traced back to the 1930s when British statistician Ronald A. Fisher introduced the concept of discriminant analysis as a way to distinguish between two groups. However, the proper formalization of Logistic Regression as a statistical tool is attributed to Joseph Berkson, who in the 1940s introduced the concept of discriminant analysis. Generally, Linear regression is used to estimate the dependent variable in case of a change in independent variables. Whereas logistic regression is used to calculate the probability of an event (Hilbe, 2009; LaValley, 2008).

3. Ridge regression

Another type of regression is ridge regression. The history of Ridge Regression is related to Tikhonov's concept of regularisation, which was invented independently in different contexts. It became widely known through its application to integral equations in the work of Andrei Tikhonov and David L. Phillips. In the statistical literature, thanks to Hoerl, it is known as Ridge Regression. The name is derived from ridge analysis, where 'ridge' refers to the path from the maximum constraint. Ridge Regression is a technique used in statistical analysis and machine learning to model data, especially when the independent variables are highly correlated. It is an extension of Linear Regression that introduces an additional regularisation parameter to reduce the impact of colinear variables on the model. Unlike Linear Regression, which minimizes the sum of squares of the residuals (RSS), Ridge Regression adds a penalty to the size of the regression coefficients, leading to more stable and interpretable models. Compared to Logistic Regression, which is used to predict binary outcomes, Ridge Regression is typically applied to regression problems where the resulting dependent variable is continuous (Saleh, Arashi, Golam Kibria, 2019; Hoerl, 2020).

4. Lasso regression

On the other hand, the lasso regression model was originally developed in 1989. It is an alternative to classical least squares estimation that avoids many of the problems of overfitting when we have a large number of independent variables. Lasso regression (Least Absolute Shrinkage and Selection Operator) is a linear regression technique used to estimate model coefficients that introduce

regularisation. Lasso regression is useful in cases where there are multiple features, some of which may not be significant. It can help to identify significant features, reduce data redundancy, and increase the interpretability of the model (Ranstam, Cook, 2018; Zhang, Wei, Lu, Pan, 2020).

Languages and libraries used for data analysis

Programming languages play a key role in data mining, enabling the creation of algorithms, the automation of data analysis processes, and the visualization of results. They enable analysts to effectively manage, process, and draw valuable insights from data. The choice of the right programming language depends on the specifics of the project, the tools available, and personal preferences. Programming languages such as Python, R, SQL, Java, and SAS are most commonly used in data mining. This article will use two of the most popular ones, Python and R (https://www.taazaa.com/python-tools-for-data-mining/; https://www.rdatamining.com/).

Python

Python is extremely popular due to its simplicity and rich ecosystem of libraries that support various aspects of data analysis. The main libraries used in the survey are (https://www.taazaa.com/python-tools-for-data-mining/):

1. `CSV` is a library that allows data from `csv` files to be easily loaded into Python, which is useful for data analysis, importing and exporting data from different systems, and automating data processing. It also allows data generated in Python to be saved to `csv` files for easy later use or sharing with other users and systems. The `csv` library supports various delimiters and `csv` formats, making it a versatile tool for working with tabular data in Python.
2. `Sckit-learn` is a library in Python that provides tools for machine learning. It is commonly used to build and evaluate predictive models. The library includes a wide range of algorithms, including regression, classification, clustering, and dimensionality reduction. `Scikit-learn`'s main applications include data preprocessing, feature engineering, model selection, and performance evaluation. The library also includes a set of tools for cross-validation and hyper-parameter optimization to create accurate and efficient models. `Scikit-learn` integrates with other tools in the Python ecosystem, such as `NumPy`, `SciPy`, and `matplotlib`, making it a versatile and powerful tool for data analysts and machine learning researchers.
3. `Matplotlib` is a library in Python for creating a variety of graphs and data visualizations. It is useful for data analysis, scientific research, engineering, and machine learning. Matplotlib allows easy customization of the appearance of graphs, supports numerical data from `NumPy`, and supports a variety of user interfaces, making it a versatile tool for data visualization in Python.

4. Seaborn is a high-level visualization library for the Python language that is based on Matplotlib. It is used to create attractive and informative statistical graphs. Seaborn facilitates the generation of graphs such as dot plots, histograms, box plots, and heat maps. With its ease of use and aesthetically pleasing default styles, Seaborn is a popular tool in data analysis and data science.

R language

R is a programming language specifically designed for statistical analysis and data visualization. The research used (https://www.rdatamining.com/):

1. **Readr** is a library in R, part of the tidyverse package, used to quickly and accurately import data from CSV, TSV, and other text formats. It allows easy control of data types, headers, and separators, which is crucial for data analysis and statistical modeling.

2. **Caret** is a tool for building, validating, and comparing machine learning models. It is particularly appreciated for its support in data management, such as partitioning into training and test sets and data standardization. Caret also offers tools for feature selection and dimensionality reduction, which supports model optimization and reduces the risk of overfitting. In addition, caret provides various cross-validation techniques, enabling fair comparison of the performance of different models. Its unified interface makes it easy to experiment with the various machine learning algorithms available in R and to quickly evaluate and select the optimal model. These features make caret an extremely useful tool for data scientists and machine learning professionals.

3. **E1071** is a data analysis and machine learning tool specializing in classification and regression. It provides implementations of algorithms such as support vector machine (SVM), k-nearest neighbors (kNN), Bayesian classifiers, and decision trees. e1071 also provides tools for model cross-validation and data processing, including normalization and standardization. It is a valued tool in the R community for its versatility and usefulness in developing advanced data analysis and predictive models.

4. **Ggplot2** is a visualization library based on graph grammars. It is widely used to create elegant and highly customizable charts. The main tenets of ggplot2 include the use of layers, which are added to the graph, allowing complex data visualizations to be created with ease. The library offers a wide range of geometries such as points, lines, bars, and areas, and also supports different data types such as time series and categorical data. Thanks to its flexibility and aesthetics, ggplot2 is the preferred tool for data analysts and researchers to present data clearly and professionally.

5. **KableExtra** is an extension for the knitr package that allows you to create beautiful and interactive HTML and PDF tables in your reports. The main

features of `kableExtra` include adding formatting to tables, such as coloring rows and columns, adding headers, and footers, and the ability to add labels to tables. This package is particularly useful for generating data reports, presenting analysis results, and visualizing data clearly and professionally. Thanks to its ease of use and flexibility, `kableExtra` is a popular tool in the R environment for those involved in data analysis and results reporting.

Comparative analysis of data mining algorithms

In this work, the *Credit Card Fraud Detection Dataset 2023* dataset was used, which contains information on credit card transactions. This data has been collected for analysis and mining in the field of machine learning, as well as for performing various analyses on financial transactions (https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detectiondataset; Aggarwal, Yu, 2008). The dataset used to perform the study was divided into 31 parts. Each of them is responsible for a different task performed during the data analysis done in the work. The first is the unique ID for each record, so you can find out how many rows are in the database. The first column will not affect the performance of the algorithm. The range of these records is from 0 to 568,629 which means that there are 568,630 records in the database there. The next part belongs to the input data, based on which the algorithms will be able to assign to a class. There are 28 features defined in the database used, and named from V1 to V28.

The purpose of the analysis is to compare the effectiveness and time efficiency of selected data mining algorithms in the Python and R programming languages. An assessment was made of how the different algorithms handle the tested data, how fast they can process the data, and how they perform in different language environments, which is important, especially for large data sets. In this article, we will make a detailed comparison of the results of applying seven different data mining algorithms on a single database, using two popular programming languages: Python and R. The algorithms Naive Bayes Classifier, K-Nearest Neighbors, SVM, Decision Tree, Random Forest, GBM, and Logistic Regression, are widely used in data analysis and pattern detection. The second stage of the comparison will be to examine the MSEs for the Linear Regression, Combs, and LASSO algorithms. In the analysis presented here, each of these algorithms has been tested a hundred times for both the Python language and the R language, on the same database. This duplication of the experiment will allow us to obtain stable and comparable results, and to understand whether the choice of the right algorithm has a significant impact on performance on a given database. The purpose of the analysis is to investigate which algorithm performed best against bank fraud and to understand whether the choice of a particular algorithm can affect the effectiveness of data analysis.

To carry out an evaluation of the effectiveness of the algorithms for Python and R language and to determine whether the choice of one of them can affect

the results achieved, at the beginning of the study we compared the effectiveness of the seven selected algorithms in terms of percentage of matching, then we will focus on the best and worst sample of these algorithms, and finally we will compare the execution time. Next, the mean squared error was examined. Table 1 compares the algorithms in terms of corresponding class assignments. When we look at the first three algorithms, one might think that the programming language does not have much impact on the efficiency of algorithm execution. However, when we come to further analysis we see a difference of almost 8 percentage points in the effectiveness of the Decision Tree algorithm and more than 10 percentage points for the GBM algorithm. Such differences between these two algorithms may tempt us to think about the choice of the programming environment.

Table 1. Efficiency of algorithms by programming language

Algorithm name	Effectiveness Python [%]	Effectiveness R [%]
Naive Bayes Classifier	91.893	91.912
K-Nearest Neighbors	99.918	99.589
Support Vector Machine	99.684	98.677
Decision Tree	99.498	91.979
Random Forest	99.969	99.539
Gradient Boosting Machine	97.875	87.332
Logistic Regression	96.428	96.515

Table 2 shows a comparison of the best and worst results obtained by a given algorithm. During this comparison, it is safe to say that the differences between the best and worst results for both languages were not large. The biggest difference was recorded for the Random Forest algorithm in R language where the difference was almost 2 percentage points. If we look at the previous disparity, it can be considered small. In addition, from this table, one can read that it was the Python language that achieved 100% efficiency in one of the trials. We could not draw such a conclusion from the comparisons of the earlier subsection. Another point one did not see earlier is by far the weakest results for the R language for the GBM algorithm, where the difference could be almost 10 percentage points.

Table 2. Maximum and minimum efficiency of algorithms by programming language

Algorithm name	Max Python [%]	Min Python [%]	Max R [%]	Min R [%]
Naive Bayes Classifier	92.092	91.684	92.112	91.688
K-Nearest Neighbors	99.989	99.846	99.795	99.370
Support Vector Machine	99.848	99.290	98.954	98.336
Decision Tree	99.811	99.231	92.503	91.406
Random Forest	100.000	99.942	99.881	99.144
Gradient Boosting Machine	98.349	97.442	88.200	86.496
Logistic Regression	96.802	95.998	96.740	96.301

Figure 1 shows the speed of the algorithms. The results show the time disparity between the algorithms. It can be seen that 5 out of 7 algorithms were executed faster in Python. The only algorithms executed faster in R were Linear Regression and Decision Tree. However, when we compare the difference in execution speed between them and the GBM algorithm, one can conclude that the values are not large in favor of the R language.

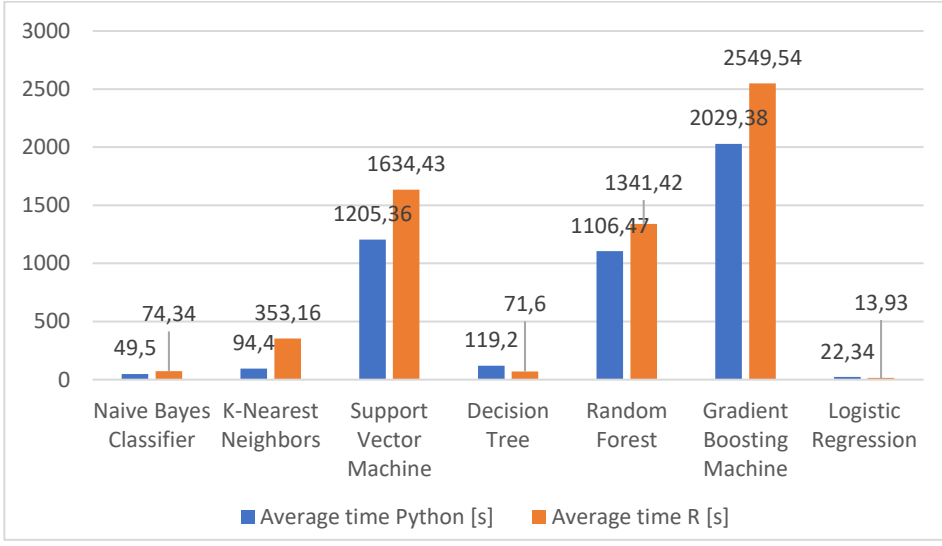

Figure 1. Average execution time of selected algorithms by programming language

The last study was an analysis of the mean squared error of the regression algorithms. Table 3 shows us what the earlier ones did not show us. The poor performance of LASSO Regression is mainly due to the Python language, which completely failed in-class assignments. The error is decisive, and because of this, LASSO Regression performed the weakest, despite the fact that in the R language, this algorithm was not the weakest. It is also worth mentioning that the average MSE turned out to be the best for Linear Regression for both languages.

Table 3. MSE comparison by programming language

Algorithm name	MSE Python	MSE R
Logistic Regression	0.059040064	0.059117396
Ridge Regression	0.059047293	0.059774547
LASSO Regression	0.250000999	0.059125534

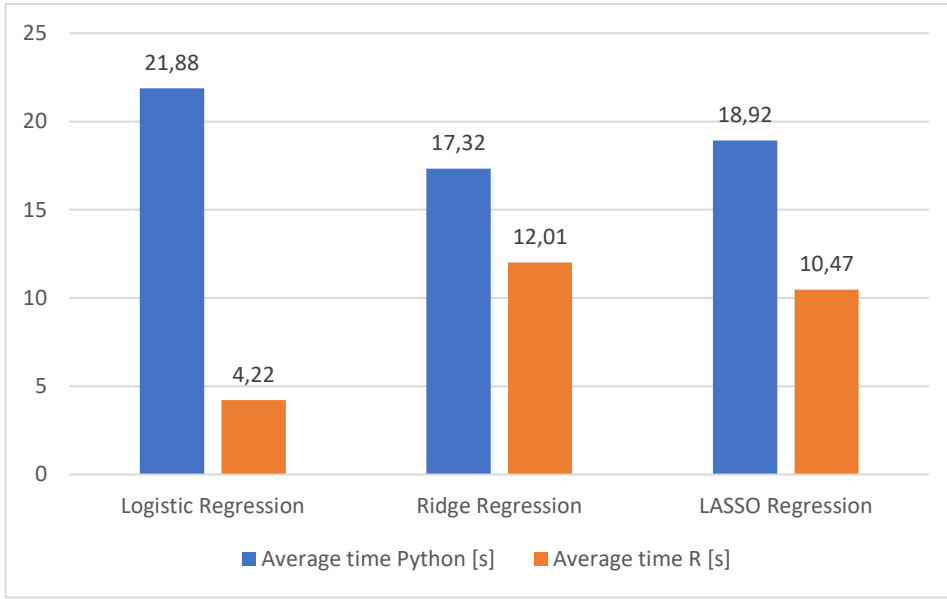

Figure 2. Average execution time of selected algorithms by programming language

Figure 2 shows the execution time. Rather, the results of the algorithms were very close, but when it breaks them down into separate programming languages one can see a much greater disparity, for example, in Linear Regression, where the difference is more than 17 seconds. Having information about the average execution time, which was less than 17 seconds. Algorithm in Python language took more time to get better results.

Conclusion

The article describes and examines 10 different algorithms used in data mining. Each of them is characterized by a different way of execution or different mathematical foundations. It is worth adding that the world of data mining does not end with these algorithms, in fact, it can be said that this is just the beginning. Each of them has its own strengths and weaknesses, so during the process, it's worth checking which algorithm is best suited to the issue at hand.

The research showed that the algorithms used would be able to detect fraud at about 70%. The best algorithms turned out to be the Random Forest and K-Nearest Neighbors algorithms. As for the mean squared error, on the other hand, Linear Regression turned out to be the best. The worst results, on the other hand, were obtained by the Naive Bayes classifier and LASSO Regression algorithms. The GBM and LASSO Regression algorithms achieved the largest differences between their weakest and best results. Another aspect examined was

the algorithm's execution time. Here Logistic and Linear Regression performed the fastest, while GBM and LASSO Regression performed the slowest.

It was possible to correctly classify the entire test set obtained by using Random Forest. In the criterion of time, score difference, and efficiency, the best algorithm turned out to be K-Nearest Neighbors, because it performed much faster than Random Forest. As for linear solutions here there is no problem because in all 3 criteria Linear Regression turned out to be the best. The choice between Python and R is relatively easy, as the efficiency results indicated that 5 out of 7 algorithms obtained better results in Python. Similar results were obtained for the mean squared error, where 2 out of 3 algorithms had a lower error than the R language. The difference between the maximum and minimum results also shows that Python outperformed R by the same ratio. Only the R language proved to be better in terms of execution time for linear algorithms, i.e. those in which the mean squared error was tested, where all 3 algorithms were executed faster. In terms of efficiency, execution speed is again in favor of Python, where 5 algorithms were executed faster than in R.

In addition, learning to program in Python or R and work with ML strengthens students' problem-solving skills, formulating hypotheses, testing models and iteratively improving solutions. This practice-based approach to learning develops the ability to understand complex systems. Also nowadays, many employers expect new employees to be able to work with analytical tools and ML. Sectors such as fintech, biotech, e-commerce, logistics and Industry 4.0 are intensively looking for professionals who can build predictive models, analyse data and automate processes using R, Python and ML tools. Therefore, the ability to use data minig tools and apply appropriate algorithms is important for the educational process of students, and this paper can be a valuable contribution to their education.

References

Aggarwal, Ch.C., Yu, Ph.S. (2008). Privacy-preserving data mining: A survey. In: M. Gertz, S. Jajodia (eds.), *Handbook of Database Security Applications and Trends* (pp 431–460). Springer. Retrieved from: https://www.academia.edu/40139891/Handbook_of_Database_Security_Applications_and_Trends (5.04.2024).

A.R., Kundu, S. (2018). Improved Random Forest for Classification. *IEEE Transactions on Image Processing*, *27*(8), 4012–4024. Retrieved from: https://ieeexplore.ieee.org/document/8357563 (10.05.2024).

Berrar, D. (2019). *Bayes' Theorem and Naive Bayes Classifier*. Retrieved from: https://www.sciencedirect.com/science/article/abs/pii/B9780128096338204731 (1.06.2024).

Deepali, K.J. (2013). Big Data: The New Challenges in Data mining. *Encyclopedia of Bioinformatics and Computational Biology*, *1*, 403–412.

International Journal of Innovative Research in Computer Science & Technology (IJIRCST), *1*(2), 39–42. Retrieved from: https://www.ijircst.org/view_abstract.php?title=Big-Data:-The-New-Challenges-in-Data-Mining&year=2013&vol=1&primary=QVJULTEx (10.05.2024)

Dymora, P., Mazurek, M., Jucha, M. (2023). *Regression Models Evaluation of Short-Term Traffic Flow Prediction*. In: W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, J. Kacprzyk

(eds.), *Dependable Computer Systems and Networks. DepCoS-RELCOMEX 2023* (pp. 51–61). Cham: Springer,

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, *17*(3), 37. https://doi.org/10.1609/aimag.v17i3.1230.

Feng, C., Pan, D., Jiafu W., Daqiang, Zh., Athanasios, V., Xiaohui, R. (2015). *Data Mining for the Internet of Things: Literature Review and Challenges*. https://journals.sagepub.com/doi/10.1155/2015/431047.

Feng-Jen, Y, (2018). An Implementation of Naive Bayes Classifier. *I*n: *International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA* (pp. 301–306). DOI: 10.1109/CSCI46756.2018.00065. Retrieved from: https://ieeexplore.ieee.org/document/8947658/authors#authors (6.05.2024)

Han, J., Kamber, M., Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publ Inc.

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Vergal New York Inc.

Hilbe, J.M. (2009). *Logistic Regression Models*. New York: Chapman & Hall/CRC.

Hoerl, R.W. (2020). *Ridge Regression: A Historical Context*. Retrieved from: https://www.researchgate.net/publication/346041807_Ridge_Regression_A_Historical_Context (25.03.2024)

https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detectiondataset 2023 (7.03.2024).

https://www.rdatamining.com/ (15.05.2024).

https://www.taazaa.com/python-tools-for-data-mining/ (15.05.2024).

LaValley, M.P. (2008). *Logistic Regression*. https://www.researchgate.net/publication/5395164_Logistic_Regression (10.04.2024)

Lee, Ch.S., Cheang, P.Y.Sh., Moslehpour, M. (2022). *Predictive Analytics in Business Analytics: Decision Tree*. Retrieved from: https://www.researchgate.net/publication/357447580_Predictive_Analytics_in_Business_Analytics_Decision_Tree (15.04.2024)

Lu, H., Mazumder, R. (2020). *Randomized Gradient Boosting Machine*. Retrieved from: https://epubs.siam.org/ doi/abs/10.1137/18M1223277?casa_token=aexR9DS_q_IAAAAA:iQmn-VnKJkU2EQt07 iujgg7uOVdiWjUjCss934fO7Z2zrIamAl8RB8_vj_BXouNyrNgRzBn8r2Al3AQ (16.04.2024)

Montgomery, D.C., Peck, E.A., Vining, G.G. (2021). *Introduction to linear regression analysis*. Hoboken, NJ: John Wiley & Sons, Inc.

Mucherino, A., Papajorgji, P.J., Pardalos, P.M. (2009). k-Nearest Neighbor Classification. In: A. Mucherino, P.J. Papajorgji, P.M. Pardalos, *Data Mining in Agriculture* (pp. 83–106). Springer,

Natekin, A., Knoll, A. (2013). Gradient boosting machines, a tutorial. *Front. Neurorobot.*, *7*, 21. https://doi.org/10.3389/fnbot.2013.00021.

Olson, D.L. (2007). Data mining in business services. *Service Business*, *1*, 181–193,

Ranstam, J., Cook, J.A. (2018). LASSO regression. *British Journal of Surgery*, *105*(10), 1348. https://doi.org/10.1002/bjs.10895 (01.09.2024).

Saleh, A.K.M.Eh, Arashi, M., Golam Kibria, B.M. (2019). *Theory of Ridge Regression Estimation with Applications*. John Wiley & Sons, Inc.

Schonlau, M., Zou, R.Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, *20*(1), 3–29. https://doi.org/10.1177/1536867X20909688 https://journals.sagepub.com/doi/10.1177/1536867X20909688 (8.04.2024)

Seber, G.A.F., Lee, A.J. (2012). *Linear Regression Analysis*. John Wiley & Sons, Inc.

Srivastava, D.K., Bhambhu, L. (2010). Data Classification Using Support Vector Machine. *Journal of Theoretical and Applied Information Technology*, *12*(1), 1–7. Retrieved from: https://www.researchgate.net/publication/285663733_Data_classification_using_support_vector_machine (25.04.2024)

Zhang, L., Wei, X., Lu, J., Pan, J. (2020). Lasso regression: From explanation to prediction. *Advances in Psychological Science*, *28*(10), 1777–1788. Retrieved from: https://journal.psych.ac.cn/xlkxjz/EN/10.3724/SP.J.1042.2020.01777 (6.04.2024)

Zhang, Y. (2012). Support Vector Machine Classification Algorithm and Its Application. In: C. Liu, L. Wang, A. Yang (eds.), *Information Computing and Applications.* ICICA 2012. Communications in Computer and Information Science. Vol. 308 (pp 179–186). Berlin, Heidelberg: Springer, https://doi.org/10.1007/978-3-642-34041-3_27.

Zhang, Zh. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicin*, *4*(11), 1–7. Retrieved from: https://atm.amegroups.org/article/view/ 10170/html (10.05.2024).