Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/50/20241218

Predicting customer subscriptions to fixed-term deposit products based on machine learning approach

Haodong Yi

Bucknell University, Lewisburg, United States

hiddenedgey@gmail.com

Abstract. In the contemporary dynamic financial milieu, financial institutions confront the exigency of comprehending and tailoring services to meet the idiosyncratic demands of individual customers, with a particular emphasis on forecasting fixed-term deposit commitments. The integration of machine learning proffers a robust framework to disentangle the intricacies inherent in customer decision-making processes. This investigation expounds upon a systematic framework encompassing data rectification, validation, and the process of feature curation, underscoring the imperative nature of a scrupulous and methodical approach. The exposition introduces an array of machine learning models, including XGBoost, Logistic Regression, Random Forest, Neural Networks, and Gaussian Naive Bayes, offering elucidation on their respective applications. Noteworthy attention is accorded to the Random Forest and Neural Networks models, with detailed explanations of their operational principles and strengths. The study underscores the criticality of conscientious data preprocessing, featuring a presentation of pertinent Python libraries and methodologies for data refinement, validation, and feature selection. The discourse culminates in a delineation of the potential of neural networks as a potent instrument in the domain of machine learning, affording insight into their intricate architecture and the iterative training process, whilst accentuating their versatility across diverse domains. In summation, this inquiry furnishes a comprehensive and pragmatic compendium on the utilization of machine learning methodologies for the prediction of customer subscriptions within the financial sector.

Keywords: Customer Subscriptions, Fixed-Term Deposit Products, Machine Learning Approach

1. Introduction

In today's rapidly evolving financial landscape, banking institutions face a critical challenge: how to effectively engage potential clients and tailor their services to individualized needs. Central to this challenge is the accurate prediction of customer behavior, particularly their inclination towards fixed-term deposit instruments. As the financial sector increasingly relies on data-driven approaches, the integration of machine learning techniques emerges as a potent solution for unraveling the complexities inherent in customer decision-making processes, thereby enhancing predictive accuracy.

The decision to invest in fixed-term deposit products depends on various factors, including economic conditions, client preferences, and financial objectives. Traditional analytical methods have shown limitations in capturing the nuanced nature of this decision-making paradigm. Machine learning, an

DOI: 10.54254/2755-2721/50/20241218

innovative approach, employs algorithmic capabilities to uncover hidden patterns, relationships, and insights within extensive datasets.

At its core, this machine learning initiative aims to develop predictive models that determine whether a customer will subscribe to a bank's fixed-term deposit products. By leveraging historical data encompassing diverse customer profiles, financial indicators, and market trends, these models can make accurate predictions, ultimately helping banks craft targeted marketing strategies and optimize resource allocation.

Key components of this machine learning approach include feature engineering, algorithm selection, and model evaluation. Feature engineering involves selecting and transforming relevant variables to create a robust feature set. These features serve as inputs for machine learning algorithms, which can range from classic techniques like logistic regression to advanced ensemble methods such as Random Forest and XGBoost. The performance of these models is rigorously assessed through cross-validation and metrics like precision, recall, and the F1-score, ensuring their suitability for real-world application.

One significant advantage of using machine learning to predict customer subscriptions is its adaptability and ability to improve over time. As new data becomes available, the models can be retrained, incorporating the latest information, and enhancing their predictive capabilities. This adaptability is especially valuable in the dynamic financial sector, where market conditions and customer behaviors are subject to constant change.

Moreover, the utilization of machine learning in predicting customer subscriptions offers a notable advantage through its inherent adaptability and capacity for continual improvement. With access to new datasets, these models can undergo retraining, incorporating the most current data to enhance their predictive prowess. This adaptability is particularly pertinent in the dynamic financial sector, characterized by the constant flux of market conditions and evolving customer behaviors.

Furthermore, this predictive approach has the potential to revolutionize customer engagement strategies. Accurately identifying customers with a heightened inclination to subscribe to fixed-term deposit products empowers banks to focus their marketing efforts on these high-potential segments. This strategic alignment facilitates a more efficient allocation of resources, leading to a proportional increase in conversion rates. Additionally, the insights derived from the systematic analysis of model predictions offer invaluable feedback for refining product offerings, tailoring communication strategies, and cultivating a competitive edge within the market landscape.

By exploring the potential of machine learning in predicting customer subscriptions to fixed-term deposit products, we delve into a realm where data-driven insights seamlessly intersect with financial decision-making. By harnessing the power of algorithms and data analysis, our overarching goal is to equip banking institutions with the necessary tools to make informed and judicious decisions. These decisions, we contend, will not only benefit the bottom line but also foster a banking environment characterized by heightened personalization and responsiveness to the needs of valued clientele. In traversing this trajectory, we unveil a glimpse into the prospective landscape of customer-centric banking, where predictive analytics unleashes hitherto untapped opportunities and nurtures sustainable growth.

2. Method

In the pursuit of constructing a robust and highly accurate predictive framework, a meticulous and systematic approach is paramount. This entails a sequence of sophisticated operations meticulously orchestrated to extract meaningful insights and optimize the predictive prowess of the model.

At the onset of this data-driven journey, an intricate codebase is meticulously engineered to execute a sequence of crucial tasks that lay the foundation for informed decision-making. A fundamental facet of this codebase revolves around data cleansing, where intricate algorithms are strategically deployed to ensure the data is devoid of imperfections and inconsistencies. Missing values, often a recurrent challenge, are meticulously imputed through advanced statistical techniques, preserving data integrity and preventing potential distortions in the subsequent analytical processes.

DOI: 10.54254/2755-2721/50/20241218

The heart of this intricate codebase beats with the rhythm of data analysis, an endeavor characterized by a judicious selection of predictive models and their methodical comparison. The data dances through a symphony of algorithms, each model meticulously designed to uncover the underlying patterns and relationships within the data. The metrics of each model's performance are rigorously evaluated, fostering an environment of quantitative discernment that guides the selection of the most optimal predictive model.

In the grand tapestry of predictive analytics, the zenith of this endeavor culminates in the utilization of the paramount predictive model. The culmination of the arduous preparatory steps leads to the momentous decision to employ the most superior model for the prediction exercise. This model, nurtured and refined through the meticulous interplay of data refinement and methodical analysis, is poised to make predictions with a heightened level of precision and accuracy.

2.1. Data Cleaning

To create the model, the first step is to clean the dataset. Cleaning a dataset is a crucial step in the data preprocessing pipeline, as it directly influences the quality and reliability of subsequent analyses and machine learning models. It involves identifying and rectifying inconsistencies, errors, and missing values within the dataset. Properly cleaned data ensures that the insights drawn from analyses are accurate and actionable, leading to more informed decision-making.

In Python, several libraries and techniques can be employed for dataset cleaning. First and foremost, an initial exploration of the dataset is imperative. This involves examining summary statistics, identifying missing values, and visualizing the distribution of variables. Python libraries such as Pandas and NumPy are instrumental in performing these tasks. Pandas provides powerful tools for data manipulation and allows for the seamless identification and handling of missing values. Techniques like imputation or removal of missing data points can be employed, depending on the nature of the dataset and the analysis at hand.

Furthermore, outliers, which can skew analytical results, should be addressed. Techniques like z-score analysis or the interquartile range (IQR) method, facilitated by Python libraries like SciPy, can be used to detect and manage outliers appropriately. Additionally, categorical variables may require encoding or transformation to numerical values for compatibility with machine learning algorithms. Python's Scikit-learn library offers preprocessing modules that facilitate this conversion.

2.2. Data Validation

Data validation constitutes a pivotal procedure aimed at upholding the precision, uniformity, and dependability of data employed in analytical endeavors or machine learning frameworks. It encompasses a methodical scrutiny of data to authenticate its caliber and integrity prior to its deployment for decision-making purposes. This procedural step assumes paramount significance, as data inaccuracies or omissions have the potential to yield erroneous deductions and subpar business determinations. Through the imposition of stringent validation protocols, organizations are poised to engender a heightened trust in the dependability of their insights, thereby culminating in decision-making processes that are both well-informed and efficacious.

In Python, there are several techniques and libraries available for data validation. One fundamental step is to perform exploratory data analysis (EDA) using tools like Pandas and NumPy. This involves summarizing statistics, identifying missing values, and visualizing distributions to gain a comprehensive understanding of the dataset. Additionally, Pandas provides functionalities for data cleaning, allowing for the seamless handling of missing values through imputation or removal. Outliers, which can skew analytical results, can be detected and managed using techniques like z-score analysis or the interquartile range (IQR) method, with support from libraries like SciPy. Categorical variables may require encoding or transformation into numerical values for compatibility with machine learning algorithms, which can be accomplished using Scikit-learn's preprocessing modules. Overall, data validation in Python involves a systematic approach of identifying and rectifying issues related to missing values, outliers, and data

DOI: 10.54254/2755-2721/50/20241218

types, setting the stage for accurate analyses and reliable decision-making. This meticulous process is crucial for ensuring that organizations can trust the data they rely on for critical business insights[1].

2.3. Forward Selection

After data validation, it is important to filter the data in order to boost the accuracy of the result of the models.

In this case, we use forward selection. Forward selection is a valuable technique in the realm of feature selection, which plays a pivotal role in enhancing the efficiency and interpretability of predictive models. This method involves a systematic approach to selecting the most informative attributes from a larger pool of potential features, progressively augmenting the model's complexity. Unlike other techniques that consider the entire feature space at once, forward selection commences with an empty set of variables and iteratively appends the most influential attributes based on predefined criteria, such as their predictive power or contribution to model performance.

In Python, implementing forward selection typically involves a combination of feature ranking and model evaluation. Initially, a base model is trained using a subset of features, and each attribute's relevance is assessed. The chosen metric could be, for instance, feature importance scores in a Random Forest or coefficients in a linear regression model. The most promising feature is then integrated into the model, and the process iterates, with additional variables incrementally incorporated. At each step, the model's performance is evaluated, and the selected features are assessed for their collective contribution to predictive accuracy. This iterative refinement continues until a predefined stopping criterion is met, such as a specified number of features or a threshold improvement in model performance. The final set of selected attributes represents the optimal combination for constructing a parsimonious and accurate predictive model. Forward selection serves as an indispensable data filter by sieving through the feature space to identify the most influential variables, streamlining model complexity and refining predictive precision [2].

2.4. Models Employed

In this study, we employed different models such as XGBoost, Random Forest, Logistic Regression, K-Nearest Neighbors, Neural Networks, and Gaussian Naive Bayes. Each model represents a different approach to solve the problem. This article will make a general introduction to all models and specifically explain random forest and neural networks models.

XGBoost, short for eXtreme Gradient Boosting, is a highly efficient and versatile machine learning algorithm known for its exceptional predictive performance across a wide range of tasks. It belongs to the ensemble learning family, which involves combining multiple models to enhance overall accuracy. XGBoost particularly excels in structured data problems such as regression, classification, and ranking[3].

This algorithm works by sequentially training an ensemble of weak learners, typically decision trees, with each subsequent tree attempting to correct the errors of the previous ones. What sets XGBoost apart is its focus on optimizing both predictive accuracy and computational speed. It achieves this through a process called gradient boosting, which minimizes a loss function by adjusting the weights assigned to each training example. Additionally, XGBoost incorporates regularization techniques to control model complexity, preventing overfitting and ensuring robust generalization to unseen data.

Logistic Regression is a fundamental and widely used statistical model for binary classification tasks in machine learning. Despite its name, it is primarily employed for classification, not regression. It's particularly useful when the dependent variable is categorical and binary, meaning it has only two possible outcomes, such as 0 or 1, Yes or No, True or False. [4,5].

Once trained, the model can be used to make predictions on new, unseen data. The output of the logistic function can be interpreted as the probability of the input belonging to the positive class. By applying a threshold (usually 0.5), the probability can be converted into a binary classification decision.

The Random Forest model is a powerful ensemble learning technique widely employed in predictive modeling tasks. It leverages the strength of multiple decision trees to enhance predictive accuracy and

DOI: 10.54254/2755-2721/50/20241218

robustness. The model operates on the principle of aggregating the outputs of numerous individual decision trees, each of which is constructed through a process of bootstrapped sampling and feature selection. In this paradigm, a diverse set of trees is generated, thereby mitigating the risk of overfitting and fortifying the model's generalization capabilities.

Each decision tree within the Random Forest is cultivated via a process that involves randomly selecting a subset of training data and features. This stochasticity introduces variability, rendering each tree distinct in its structure and predictive patterns. During the prediction phase, the outputs of the constituent trees are amalgamated through a voting or averaging mechanism, resulting in a final prediction that is robust and less susceptible to noise or idiosyncratic patterns in the data. Moreover, the model possesses the ability to quantify feature importance, providing valuable insights into the relative contributions of different variables towards the predictive outcome. This attribute endows the Random Forest with interpretability, a desirable quality in scenarios necessitating a deeper understanding of the underlying mechanisms driving predictions[6,13,14]. Furthermore, the model is proficient in handling diverse data types, accommodating both categorical and numerical attributes, which further broadens its applicability across a spectrum of real-world domains.

Besides random forest models, neural networks are also a notable and valuable model. A neural network model, a fundamental concept in the realms of artificial intelligence and machine learning, replicates the intricate interconnections observed within the human brain. This emulation serves to facilitate sophisticated data processing and the discernment of patterns. The model is characterized by an interconnected array of nodes, colloquially referred to as "neurons," organized into stratified layers. The initial layer receives and processes raw data, which subsequently traverses one or more concealed layers before culminating in an output layer. This final layer is responsible for generating conclusive predictions or classifications. Each connection linking neurons is imbued with a weight, dictating the degree of influence exerted by one neuron upon another. Throughout the training process, these weights undergo iterative adjustments with the aim of minimizing disparities between anticipated and actual outcomes. This intricate calibration is accomplished through a mechanism known as backpropagation. This procedure entails the computation of gradients for the loss function concerning each weight, followed by their iterative refinement using optimization strategies like gradient descent[7,8].

In Python, neural network models can be implemented using dedicated libraries like TensorFlow or PyTorch. These frameworks provide a comprehensive suite of tools for constructing, training, and evaluating neural networks. The process typically begins with data preprocessing, where input features are standardized or normalized to facilitate convergence during training. The network architecture is then defined, specifying the number of layers, the number of neurons in each layer, and the activation functions that govern the flow of information between nodes. The choice of activation functions influences the model's capacity to capture complex relationships within the data. Popular functions include sigmoid, tanh, and rectified linear unit (ReLU) [9,15].

Subsequently, the model is trained on labeled data using an optimization algorithm that minimizes a specified loss function, such as mean squared error for regression tasks or categorical cross-entropy for classification tasks. Throughout training, the model's performance is monitored on a separate validation set to prevent overfitting, a phenomenon where the model learns to excessively fit the training data, compromising its generalization to unseen data. Once training is complete, the model can be deployed for making predictions on new data.

Overall, neural network models in Python encapsulate a sophisticated approach to machine learning, leveraging the interconnected structure of neurons to process complex data and discern intricate patterns. Through libraries like TensorFlow and PyTorch, practitioners have access to powerful tools for constructing, training, and deploying neural networks across a diverse array of applications, from image recognition to natural language processing. The iterative refinement of weights via backpropagation, guided by optimization algorithms, empowers these models to learn and adapt, ultimately culminating in highly accurate predictions and classifications.

DOI: 10.54254/2755-2721/50/20241218

3. Results

Figure 1 encompasses the outcomes arising from the application of eight distinct models, elucidating their diverse methodologies. Of paramount importance is the evaluation of the Area Under the Curve (AUC). Notably, Gradient Boosting exhibits the most favorable AUC within the training dataset. However, it is imperative to mitigate a substantial dissonance between the AUC values of the training and test datasets. In consideration of this dual criterion, the Random Forest model emerges as the most appropriate selection for this particular dataset.

Figure 2 presents the receiver operating characteristic curve (ROC) for all models employed in this study. In response to the preceding observation regarding AUC, it is evident that the Random Forest model attains the highest ROC score, substantiating its designation as the optimal model for this research endeavor[11-13].

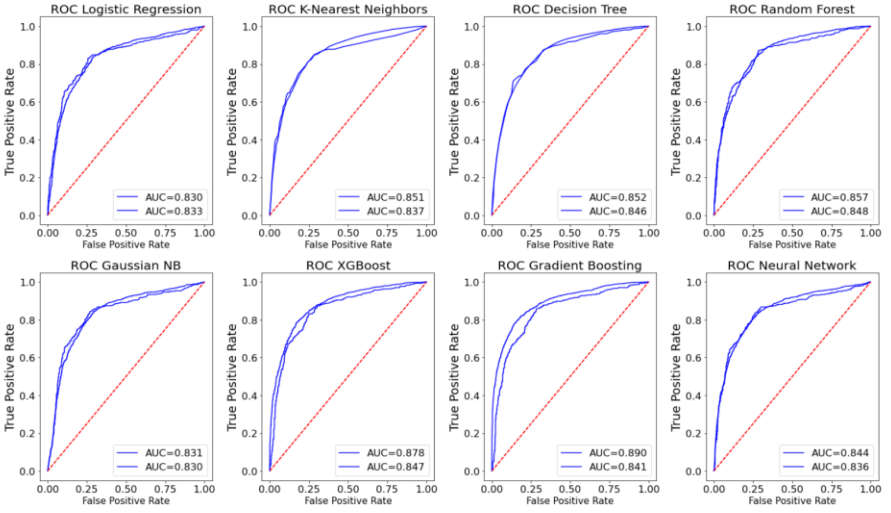

Figure 1. AUC Results of All Models

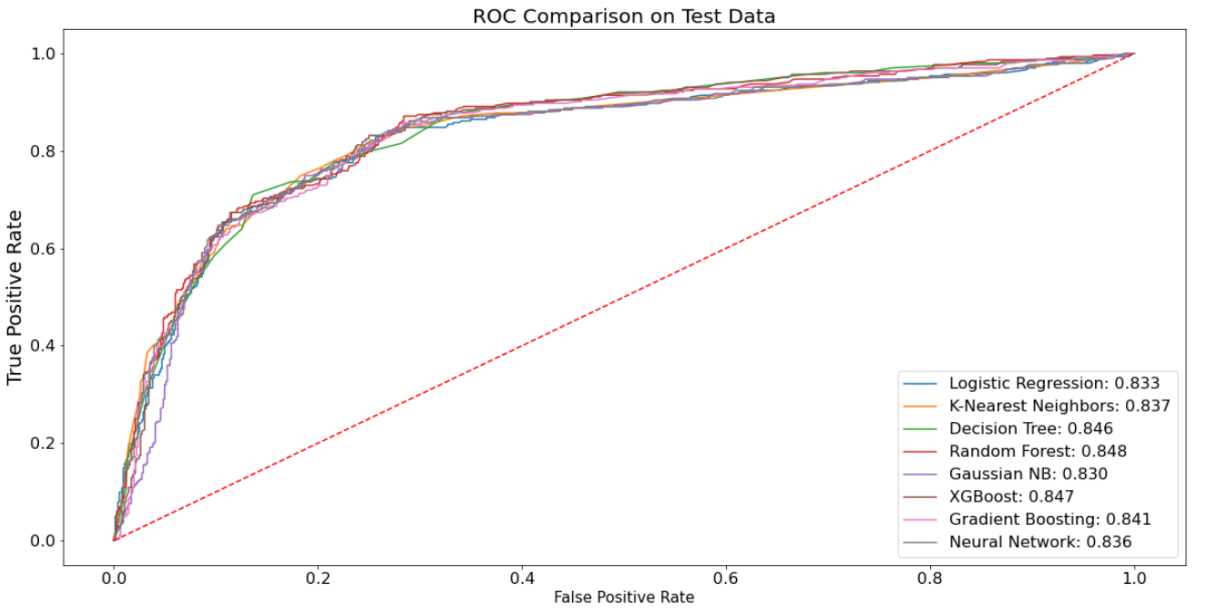

Figure 2. ROC Results of All Models

DOI: 10.54254/2755-2721/50/20241218

Figure 3 provides the anticipated outcome achievable through the adoption of the Random Forest Model within the banking context. Despite inherent stochasticity in the training process, there is a discernible consensus that when the ratio approximates 49, the bank stands to attain its zenith in returns, approximately amounting to $9200, thereby elevating the 50th bin by 1.73 units.

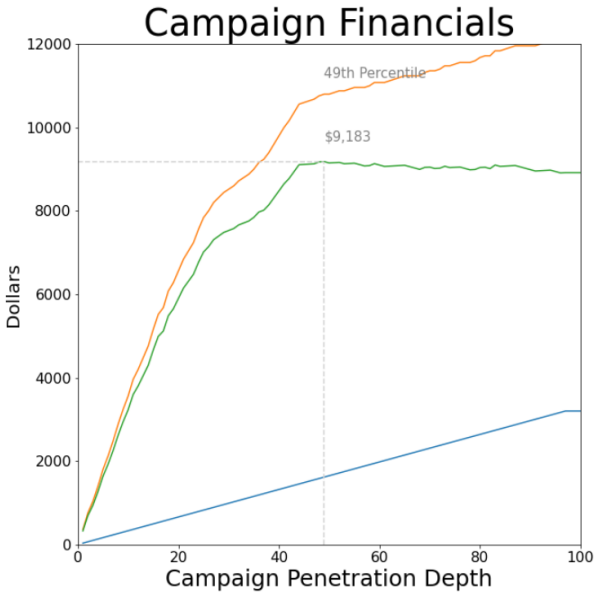

Figure 3. Final Output of Improvement with Random Forest

4. Conclusion

In the dynamic landscape of modern finance, the ability to understand and predict customer behavior is paramount. Machine learning techniques, particularly the powerful XGBoost algorithm, offer a transformative solution to this challenge. Through a meticulous process of data validation, cleaning, and model selection, we empower organizations to make informed, strategic decisions. Techniques like forward selection further refine our predictive models, ensuring that only the most influential variables are considered. With the Random Forest model, we harness the collective wisdom of multiple decision trees to enhance accuracy and robustness. Meanwhile, neural networks emulate the intricate connections of the human brain, providing a sophisticated tool for pattern recognition and data processing. These models, implemented in Python with libraries like TensorFlow and PyTorch, open doors to a new era of predictive analytics, where data-driven insights drive sustainable growth and customer-centric banking experiences. Through meticulous attention to data quality and the strategic application of advanced modeling techniques, we pave the way for informed, efficient, and effective decision-making in the financial sector.

References

[1] Pragati Baheti: Train Test Validation Split: How To & Best Practices https://www.v7labs.com/blog/train-validation-test-set#:~:text=Training%20data%20is%20the%20set,after%20completing%20the%20training%20phase.

[2] Damian Kozbur:Testing-Based Forward Model Selection The American Economic Review, Vol. 107, No. 5, PAPERS AND PROCEEDINGS OF THE One Hundred Twenty-Ninth Annual Meeting OF THE AMERICAN ECONOMIC ASSOCIATION (MAY 2017), pp. 266-269 (4 pages)

DOI: 10.54254/2755-2721/50/20241218

[3] Carla Martins: Gaussian Naive Bayes Explained and Hands-On with Scikit-Learn https://pub.towardsai.net/gaussian-naive-bayes-explained-and-hands-on-with-scikit-learn-4183b8cb0e4c
[4] Jason Brownlee: A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/
[5] "Hybrid Artificial Intelligent Systems" , Springer Science and Business Media LLC, 2023
[6] Ulrike Grömping. Variable Importance Assessment in Regression: Linear Regression versus Random Forest The American Statistician, Vol. 63, No. 4 (NOVEMBER 2009), pp. 308-319 (12 pages)
[7] Richard Briesch, Priyali Rajagopal Neural network applications in consumer behavior Journal of Consumer Psychology, Vol. 20, No. 3 (July 2010), pp. 381-389 (9 pages)
[8] Explained: Neural networks' Ballyhooed artificial-intelligence technique known as “deep learning” revives 70-year-old ideas. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414
[9] Aniket Vatsa, Ananda Shankar Hati, Vadim Bolshev, Alexander Vinogradov, Vladimir Panchenko, Prasun Chakrabarti. "Deep Learning-Based Transformer Moisture Diagnostics Using Long Short-Term Memory Networks" , Energies, 2023
[10] Classification: ROC Curve and AUC https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
[11] Galamo Monkam, Weifeng Xu, Jie Yan. "A GAN-based Approach to Detect AI-Generated Images" , 2023 26th ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPDWinter), 2023
[12] Wenlu Du, Ankan Dash, Jing Li, Hua Wei, Guiling Wang. "Safety in Traffic Management Systems: A Comprehensive Survey" , Designs, 2023
[13] Yi Lin, Yongho Jeon: Random Forests and Adaptive Nearest Neighbors Journal of the American Statistical Association, Vol. 101, No. 474 (Jun., 2006), pp. 578-590 (13 pages)
[14] David Muchlinski, David Siroky, Jingrui He, Matthew Kocher: Comparing Random Forest with Logistic Regression for Predicting Class-Imbalanced Civil War Onset Data Political Analysis, Vol. 24, No. 1 (Winter 2016), pp. 87-103 (17 pages)
[15] Tomaso Poggio, Andrzej Banburski, Qianli Liao: Theoretical issues in deep networks Proceedings of the National Academy of Sciences of the United States of America, Vol. 117, No. 48 (December 1, 2020)