Automated Feature Engineering in Software Development

AYUSHI[1], SAHIL GARG[1]

[1]CSE Department, Chandigarh College of Engineering and Technology, Chandigarh, India.

ABSTRACT Software development is an important field of research and evolution. The article explores the various features and applications of automated feature engineering in software development. The article focuses on a certain type of framework called the FeatureGen Frameworks which transforms the base features into ones with more information to better describe any tabular dataset and enhance the learning performance. This article features two of these FeatureGen Frameworks called AutoGluon-Transform and OpenFE comparing both of them based on the approach they take and how they bring improvements into Feature Generation to enhance the prediction of tabular data.

KEYWORDS Automated Feature Engineering, Machine Learning, Software Development, FeatureGen Frameworks

I. INTRODUCTION

We live in an era where technology is constantly changing and evolving. New theories and algorithms are invented every day. Scientists and researchers are constantly working on making the present technology better and increasing its efficiency so that it caters to the needs of the present as well as the needs of the future. Sustainable development[1] is one of the trending topics in the world right now. It is the need of the hour since the resources of the earth are limited. Engineering plays a vital role in various fields of sustainable development to promote it and find better solutions to implement it in practical life[2]. When seen from the perspective of software engineering, the principles and the practical elements of software engineering can be combined with various other elements to make models to promote and find solutions to problems hindering sustainable development[3] of human mankind. Thus, advances in the field of software development can help us all achieve sustainability.

Defining software development[4], it refers to the process of creating, designing, coding, testing, and maintaining software or computer programs. It involves a systematic set of activities that lead to the creation of a software product, which could be an application, system, website, or any other piece of software designed to perform specific tasks or functions. Software development is a multidisciplinary field that requires collaboration between different professionals, including programmers, designers, testers, and project managers. One such field in software development that is trending nowadays is machine learning, thus, when it is combined with other aspects of software development, it can help achieve various goals that were previously very difficult and even sometimes not feasible.

Machine learning is a subset of artificial intelligence (AI) that focuses on the development of algorithms and statistical models. These models help the computers to perform tasks without any programming. The main goal of machine learning is to make the machine capable of performing predictions based on the data that was earlier given to it. It can be used in various fields like stock market prediction[5] and various other types of algorithms[6] like classification of web pages[7] and security of banking mobile applications[8]. One such field in which machine learning is used nowadays is, automated feature engineering utilizes these machine learning algorithms to provide us with desirable results.

Diving into automated feature engineering, refers to the process of using computational techniques, often leveraging machine learning algorithms, to automatically generate new features or representations from existing data. In the context of machine learning and data science, features are the variables or attributes that are used to train a model and make predictions. It is used in various places such as HTTP tunnel detection[9] and classification problems.

Automated feature engineering has various advantages due to which it is used widely and constantly various research and developments are happening in its field. Some of its advantages are time efficiency, scalability, reduction of manual errors, exploration of complex relationships, model agnosticism, adaptability to changes in data, handling high dimensional data, and enhanced model performance among others. Automated feature engineering significantly reduces the time required for the feature engineering process. Instead of manually experimenting with various feature combinations, algorithms can efficiently explore and generate features. Also, automated processes reduce the likelihood of human errors associated with manual feature engineering. Algorithms follow consistent and predefined rules, minimizing the chance of oversight or miscalculation. Along with deep learning models[10, 11] and other techniques such as robotics, automated feature engineering has applications in various fields. Some examples of this include predictive modeling, healthcare analytics[12], natural language processing(NLP), image[13] and computer vision[14], recommendation systems, anomaly detection[15], financial modeling, industrial IoT[16] ,and sensor data, etc.

As more and more data is accumulated, it is accumulated into tabular format. When queried from a database, it also yields results in a tabular format. Hence, predictive analytics on that tabular data is a very important feature that software engineers need to focus on as trends in data and prediction from it can help make informed decisions. This was previously done manually which was really cumbersome. But with the advancements of machine learning, as mentioned earlier, it is being shifted to machine learning algorithms and thus computers to do that labor with high accuracy.

Feature generation is a very computationally heavy process as mentioned afterwards. If not done efficiently it can even become infeasible to do the process using machine learning with the kind of resources in common use today. Thus, to make the process efficient enough, specially designed and well thought out frameworks like AutoGluon-Transform[17] and OpenFE[18] are important.

II. FEATUREGEN FRAMEWORKS

Feature generation is the process of generating features of tabular data. The goal of feature generation is to transform the base features into ones with more information to better describe the data of the tables and enhance the learning performance. The quality of features generated has a serious impact on the learning process for that tabular data. For this two methods are possible, manual and automated. The manual method is very time-consuming and requires case-by-case knowledge of the domain. Hence, an automated way is taken. In this, we use machine learning models to predict the features of the tabular data. Machine learning models vary depending on the quality of features they present as a large dataset may provide millions of features that are computationally very expensive to calculate and are not even important to predict the tabular data. To filter out the features that have an incremental increase in the quality of the predictions of the tabular data, we need to do computations. Based on the variation of how the features are selected, here are two of the related models presented in research [17-18]:

- AutoGluon-Tabular: This is an AutoML framework, extremely accurate machine learning models can be trained on raw tabular datasets like CSV files with just one line of Python code. As seen in Figure 1, it assembles several models and stacks them in several layers. This is based on the expand and reduce technique, in which the basic features are expanded to generate a pool of candidate features, and the ineffective candidate features are then removed. This architecture offers fault tolerance, predictability, robustness, and simplicity. It is really simple to implement—just one line of Python code—and users don't need to understand the specifics of machine learning models to train the model on raw data. Its robustness is demonstrated by the wide range of datasets it can handle and the fact that training continues even if some of the individual ML models fail. The fault tolerance of the model is demonstrated by its ability to be stopped and resumed at any moment. Additionally, we can regulate how long it takes by setting a time_limits variable in its fit() function.
  This framework begins with robust, performant models, such as random forest, for time control and progresses to more costly, less dependable models, like k-nearest neighbors. But there are some drawbacks of this approach as listed in [19] that there may be millions of candidate features with hundreds of features each. Even with an efficient evaluation algorithm, it is very difficult to compute all candidate features on the dataset as it is very computationally expensive and often impractical.
- OpenFE: This model is also based on the expansion and reduction approach like the AutoGluon-Tabular method. But this model differentiates from the former one as the former one uses the following approach to evaluate the ineffectiveness of features: including the new features with the feature set and then re-training the complete model and then testing for the increase in efficiency. But this takes up a lot of time and this OpenFE model introduces a new FeatureBoost algorithm which trains a model with only the new features and not the complete feature set. This algorithm is implemented in the pruning algorithm which coarsely reduces the number of candidate features. This is called Successive Featurewise Pruning. Then the reduced candidate features are fed to the FeatureBoost algorithm along with the base features to further reduce the candidate featureset.

A. APPLICATIONS OF FEATUREGEN FRAMEWORKS

There are a lot of practical applications of feature generation through machine learning models. In table 1 we list some of those applications and a brief on how they are used.

III. CONCLUSION

In conclusion, automated feature engineering is a pivotal advancement in the field of machine learning which offers a transformative solution to the various challenges in today's world. The emergence of automated approaches and frameworks such as the FeatureGen frameworks as discussed in the article has significantly contributed to the streamlining of the process of creating, selecting, and optimizing features, marking a difference in the way we approach data-driven model development and machine learning. The FeatureGen framework, as a representative example of automated feature engineering tools, brings with it a suite of powerful tools and algorithms designed for software development. Its ability to identify relevant patterns, generate new features, and adapt

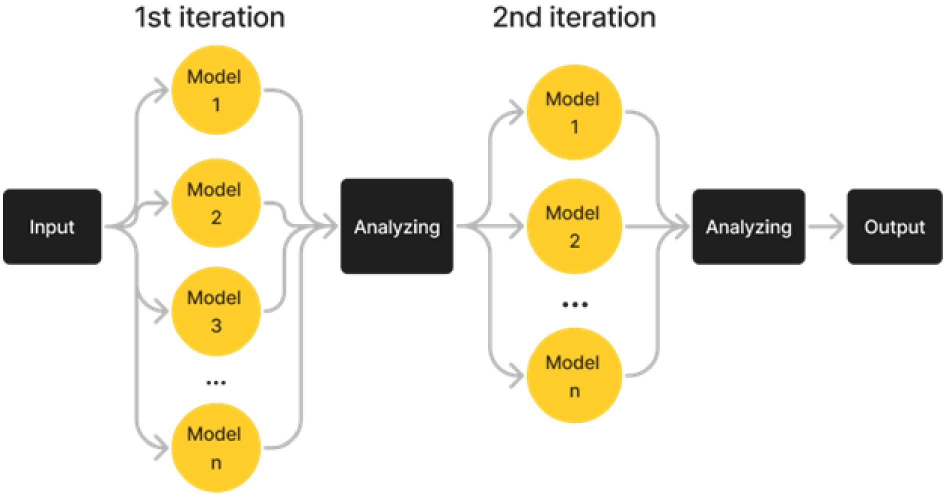

FIGURE 1: Using two stacking layers and n models, the AutoGluon multi-layer stacking approach is represented.

Practical Application	Description
Natural Language Processing[20]	Techniques such as word embeddings, like Word2Vec or GloVe, leverage algebraic operations on word vectors to capture semantic relationships, enriching the feature space with contextually relevant information. This helps transform raw textual data into meaningful numerical representations.
Image Processing	Principal Component Analysis (PCA) or Convolutional Neural Networks (CNNs) are employed to transform high-dimensional image data into a lower-dimensional, more informative representation, preserving essential features for subsequent analysis.
Genomic Data Analysis	Feature generation in genomics involves translating biological information into numerical features for tasks like gene expression analysis or DNA sequence classification.
Finance and Time Series Analysis	Fourier transforms, autoregressive models, or statistical measures such as rolling averages are employed to extract temporal patterns and create informative features that encapsulate underlying market dynamics. Implementing feature generation on the data using machine learning further enhances the feature set and predictions.
Healthcare Analytics	Translating diverse patient data into comprehensive features for predictive modeling or disease diagnosis. This can be combined with mathematical techniques such as wavelet transforms or texture analysis to create features capturing nuanced patterns.
Social Network Analysis	Feature generation aids in characterizing nodes and edges, facilitating community detection or influence analysis. Graph-based metrics like node centrality, clustering coefficients, or spectral methods leverage advanced linear algebra concepts to generate features that encapsulate the network structure and dynamics further enhancing the feature generation.

to changing data dynamics underscores its efficiency and adaptability. By automating the traditionally labor-intensive feature engineering process, FeatureGen not only saves time but also enhances the overall effectiveness of machine learning models. The advantages of automated feature engineering discussed in this article, such as time efficiency, scalability, and adaptability, highlight its relevance across diverse applications. Whether applied in predictive modeling, natural language processing, computer vision, or healthcare analytics, the impact of automated feature engineering on model performance is unmistakable.

REFERENCES

[1] Kumar, S., Singh, S. K., & Aggarwal, N. (2023). Sustainable Data Dependency Resolution Architectural Framework to achieve energy efficiency using speculative parallelization. 2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT). https://doi.org/10.1109/cisct57197.2023.10351343

[2] Rastogi, A., Sharma, A., Singh, S., & Kumar, S. (2017). Capacity and Inclination of High Performance Computing in Next Generation Computing. Proceedings of the 11th INDIACom. IEEE.

[3] Peñalvo, F. J., Sharma, A., Chhabra, A., Singh, S. K., Kumar, S., Arya, V., & Gaurav, A. (2022). Mobile Cloud Computing and Sustainable Development: Opportunities, Challenges, and Future Directions. International Journal of Cloud Applications and Computing (IJCAC), 12(1), 1-20. http://doi.org/10.4018/IJCAC.312583

[4] Basili, V. (n.d.). Software development: a paradigm for the future. [1989] Proceedings of the Thirteenth Annual International Computer Software & Applications Conference. https://doi.org/10.1109/cmpsac.1989.65127

[5] Peñalvo, F. J., Maan, T., Singh, S. K., Kumar, S., Arya, V., Chui, K. T., & Singh, G. P. (2022). Sustainable Stock Market Prediction Framework Using Machine Learning Models. International Journal of Software Science and Computational Intelligence (IJSSCI), 14(1), 1-15. http://doi.org/10.4018/IJSSCI.313593

[6] Singh, I., Singh, S. K., Singh, R., & Kumar, S. (2022). Efficient loop unrolling factor prediction algorithm using machine learning models. 2022 3rd International Conference for Emerging Technology (INCET). https://doi.org/10.1109/incet54531.2022.9825092

[7] Khade, G., Kumar, S., & Bhattacharya, S. (2012). Classification of web pages on attractiveness: A supervised learning approach. 2012 4th Inter-

national Conference on Intelligent Human Computer Interaction (IHCI). https://doi.org/10.1109/ihci.2012.6481867

[8] Sharma, A., Singh, S. K., Kumar, S., Chhabra, A., & Gupta, S. (2023). Security of android banking mobile apps: Challenges and opportunities. Lecture Notes in Networks and Systems, 406–416. https://doi.org/10.1007/978-3-031-22018-0_39

[9] Davis, J. J., & Foo, E. (2016). Automated feature engineering for HTTP tunnel detection. Computers &amp; Security, 59, 166–185. https://doi.org/10.1016/j.cose.2016.01.006

[10] Aggarwal, K., Singh, S.K., Chopra, M., Kumar, S., Colace, F. (2022). Deep Learning in Robotics for Strengthening Industry 4.0.: Opportunities, Challenges and Future Directions. In: Nedjah, N., Abd El-Latif, A.A., Gupta, B.B., Mourelle, L.M. (eds) Robotics and AI for Cybersecurity and Critical Infrastructure in Smart Cities. Studies in Computational Intelligence, vol 1030. Springer, Cham. https://doi.org/10.1007/978-3-030-96737-6_1

[11] Sharma, A., Singh, S. K., Chhabra, A., Kumar, S., Arya, V., & Moslehpour, M. (2023). A novel deep federated learning-based model to enhance privacy in Critical Infrastructure Systems. International Journal of Software Science and Computational Intelligence, 15(1), 1–23. https://doi.org/10.4018/ijssci.334711

[12] Vats, T., Singh, S. K., Kumar, S., Gupta, B. B., Gill, S. S., Arya, V., & Alhalabi, W. (2023). Explainable context-aware IOT framework using human digital twin for Healthcare. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-16922-5

[13] Kaur, P., Singh, S. K., Singh, I., & Kumar, S. (2021, December). Exploring Convolutional Neural Network in Computer Vision-based Image Classification. In International Conference on Smart Systems and Advanced Computing (Syscom-2021).

[14] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A brief review. Computational Intelligence and Neuroscience, 2018, 1–13. https://doi.org/10.1155/2018/7068349

[15] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection. ACM Computing Surveys, 41(3), 1–58. https://doi.org/10.1145/1541880.1541882

[16] Singh, R., Singh, S. K., Kumar, S., & Gill, S. S. (2022). SDN-Aided Edge Computing-Enabled AI for IoT and Smart Cities. SDN-Supported Edge-Cloud Interplay for Next Generation Internet of Things, 41-70.

[17] Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. Autogluon-tabular: Robust and accurate automl for structured data. arXiv preprint arXiv:2003.06505, 2020.

[18] Zhang, T., Zhang, Z.A., Fan, Z., Luo, H., Liu, F., Liu, Q., Cao, W. &amp; Jian, L.. (2023). OpenFE: Automated Feature Generation with Expert-level Performance. Proceedings of the 40th International Conference on Machine Learning, in Proceedings of Machine Learning Research 202:41880-41901 Available from https://proceedings.mlr.press/v202/zhang23ay.html.

[19] Wang, K., Wang, P., Chen, X., Huang, Q., Mao, Z., & Zhang, Y. (2020). A feature generalization framework for social media popularity prediction. Proceedings of the 28th ACM International Conference on Multimedia. https://doi.org/10.1145/3394171.3416294

[20] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: An introduction. Journal of the American Medical Informatics Association, 18(5), 544–551. https://doi.org/10.1136/amiajnl-2011-000464

[21] Poonia, V., Goyal, M. K., Gupta, B. B., Gupta, A. K., Jha, S., & Das, J. (2021). Drought occurrence in different river basins of India and blockchain technology based framework for disaster management. Journal of Cleaner Production, 312, 127737.

[22] Gupta, B. B., & Sheng, Q. Z. (Eds.). (2019). Machine learning for computer and cyber security: principle, algorithms, and practices. CRC Press.

[23] Singh, A., et al. (2022). Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: issues, challenges, and future research directions. International Journal on Semantic Web and Information Systems (IJSWIS), 18(1), 1-43.

[24] Almomani, A., et al. (2022). Phishing website detection with semantic features based on machine learning classifiers: a comparative study. International Journal on Semantic Web and Information Systems (IJSWIS), 18(1), 1-24.

[25] Wang, L., et al. (2018). Compressive sensing of medical images with confidentially homomorphic aggregations. IEEE Internet of Things Journal, 6(2), 1402-1409.

[26] Stergiou, C. L., et al. (2021). InFeMo: flexible big data management through a federated cloud system. ACM Transactions on Internet Technology (TOIT), 22(2), 1-22.