International Journal of Control and Automation
Vol. 13, No.04, (2020), pp.1558-1571
https://doi.org/10.52783/ijca.v13i4.38349

AI-Driven Automated Feature Engineering to Enhance Performance of Predictive Models in Data Science

[1]Surendranadha Reddy Byrapu Reddy, [2]Sarath Babu Dodda, [3]Srihari Maruthi, [*4]Mohan Raparthi

[1]Sr. Analyst, Information Technology, Northeastern University, Lincoln Financial Group, Atlanta, GA, USA.
[2]Software Engineer, Central Michigan University
ORCID: 0009-0008-2960-2378.
[3]Senior Technical Solutions Engineer, University of New Haven.
[*4]Software Engineer, Google Alphabet (Verily Life Science), Dallas, Texas, 75063.
ORCID :0009-0004-7971-9364.
[*]Corresponding Author: - Mohan Raparthi.

Abstract: - In the rapidly evolving landscape of data science, predictive modeling stands as a cornerstone for deriving actionable insights from vast amounts of data. Central to the success of predictive modeling is the process of feature engineering, which involves selecting, transforming, and creating features to improve model performance. With the advent of artificial intelligence (AI) and machine learning (ML), automated feature engineering has emerged as a promising approach to streamline and enhance this critical process. This paper explores the role of AI-driven automated feature engineering techniques in augmenting the performance of predictive models in data science.The paper begins with an overview of predictive modeling in data science, highlighting the significance of feature engineering in model development. [1] Traditional approaches to feature engineering often rely on manual experimentation and domain expertise, which can be time-consuming and prone to human bias. In contrast, AI-driven automated feature engineering leverages ML algorithms and techniques to automate and optimize the feature engineering process, reducing the need for manual intervention and accelerating model development. Various AI-driven automated feature engineering techniques are examined, including machine learning-based feature selection algorithms, automated feature transformation methods, generative adversarial networks (GANs) for feature creation, and deep learning-based feature extraction techniques. These methods offer advantages such as improved model performance, time and resource efficiency, and reduced human bias. The paper also discusses challenges and limitations associated with AI-driven automated feature engineering, such as data quality requirements, interpretability of automated features, and the risk of overfitting. Additionally, case studies and applications demonstrate the practical utility of automated feature engineering across diverse domains, including finance, healthcare, marketing, and more.

The paper explores future directions in automated feature engineering, including emerging trends, integration of domain knowledge, and ethical considerations. By providing valuable insights into the methodologies, tools, benefits, challenges, and future prospects of AI-driven automated feature engineering, this paper aims to guide practitioners and researchers in harnessing advanced techniques to enhance predictive modeling in data science.

Keywords: - Feature engineering, Predictive modeling, Artificial intelligence Automated feature engineering, Machine learning, Model performance, Data science, Optimization.

ISSN: 2005-4297 IJCA

A.Introduction: In the realm of data science, predictive modeling serves as a fundamental tool for extracting actionable insights and making informed decisions based on data-driven analysis. At the heart of predictive modeling lies feature engineering, a process that involves selecting, transforming, and creating features from raw data to enhance the performance of machine learning models. Traditionally, feature engineering has been a labor-intensive and expertise-driven task, requiring domain knowledge and manual experimentation. However, with the advent of artificial intelligence (AI) and machine learning (ML), automated feature engineering has emerged as a powerful paradigm shift, promising to revolutionize the way predictive models are developed and optimized. Automated feature engineering leverages AI and ML algorithms to automate the process of feature selection, transformation, and creation. By harnessing the computational power of algorithms, automated feature engineering aims to streamline and optimize the feature engineering process, reducing the need for manual intervention and accelerating model development. [2] This shift towards automation not only improves efficiency but also holds the potential to uncover intricate patterns and relationships in the data that may have been overlooked by manual approaches. In this paper, we delve into the role of AI-driven automated feature engineering techniques in enhancing the performance of predictive models in data science. We begin by providing an overview of predictive modeling and the importance of feature engineering in model development. Traditional approaches to feature engineering are discussed, highlighting their limitations and challenges, including the reliance on human expertise and the potential for bias. Subsequently, we explore various AI-driven automated feature engineering techniques, including machine learning-based feature selection algorithms, automated feature transformation methods, generative adversarial networks (GANs) for feature creation, and deep learning-based feature extraction techniques. These techniques offer advantages such as improved model performance, time efficiency, and reduced human bias. Furthermore, we discuss the benefits and challenges associated with AI-driven automated feature engineering, along with real-world case studies and applications across diverse domains. Finally, we explore future directions and emerging trends in automated feature engineering, including the integration of domain knowledge and ethical considerations. By shedding light on the methodologies, tools, and implications of AI-driven automated feature engineering, this paper aims to provide valuable insights for practitioners and researchers seeking to enhance the performance of predictive models in data science.

B.Literature Review: - The literature surrounding AI-driven automated feature engineering underscores its transformative potential in enhancing predictive modeling within data science. Traditional feature engineering methods, while effective, often rely heavily on domain expertise and manual intervention, leading to time-consuming processes and subjective biases. In contrast, automated feature engineering leverages machine learning algorithms to streamline and optimize feature selection, transformation, and creation, thereby reducing human intervention and accelerating model development.

Various studies have explored the efficacy of automated feature engineering techniques across different domains and applications. For instance, research by Fernandez-Delgado et al. (2014) demonstrated the benefits of automated feature selection algorithms in improving the performance of predictive models, highlighting their ability to identify relevant features and reduce overfitting. Similarly, studies by Chen and Guestrin (2016) and Van Rijn et al. (2018) showcased the advantages of automated feature transformation methods, such as scaling and normalization, in enhancing model accuracy and generalization.

ISSN: 2005-4297 IJCA

International Journal of Control and Automation
Vol. 13, No.04, (2020), pp.1558-1571
https://doi.org/10.52783/ijca.v13i4.38349

Moreover, advancements in deep learning-based feature extraction techniques have garnered significant attention in recent years. Research by Bengio et al. (2013) and LeCun et al. (2015) explored the use of deep learning models, such as convolutional neural networks (CNNs) and autoencoders, for automatically learning hierarchical representations of data, leading to improved feature extraction capabilities and enhanced model performance.

Real-world applications of AI-driven automated feature engineering have also been documented across various industries. For instance, in finance, automated feature engineering has been utilized for credit risk assessment and fraud detection (Bolton and Hand, 2002; Papamichail et al., 2019), while in healthcare, it has been applied for disease diagnosis and patient outcome prediction (Rajkomar et al., 2018; Choi et al., 2020). Overall, the literature review underscores the potential of AI-driven automated feature engineering to revolutionize predictive modeling in data science by improving efficiency, accuracy, and scalability while reducing human bias and subjectivity. However, challenges such as interpretability, data quality, and ethical considerations remain areas of ongoing research and exploration.

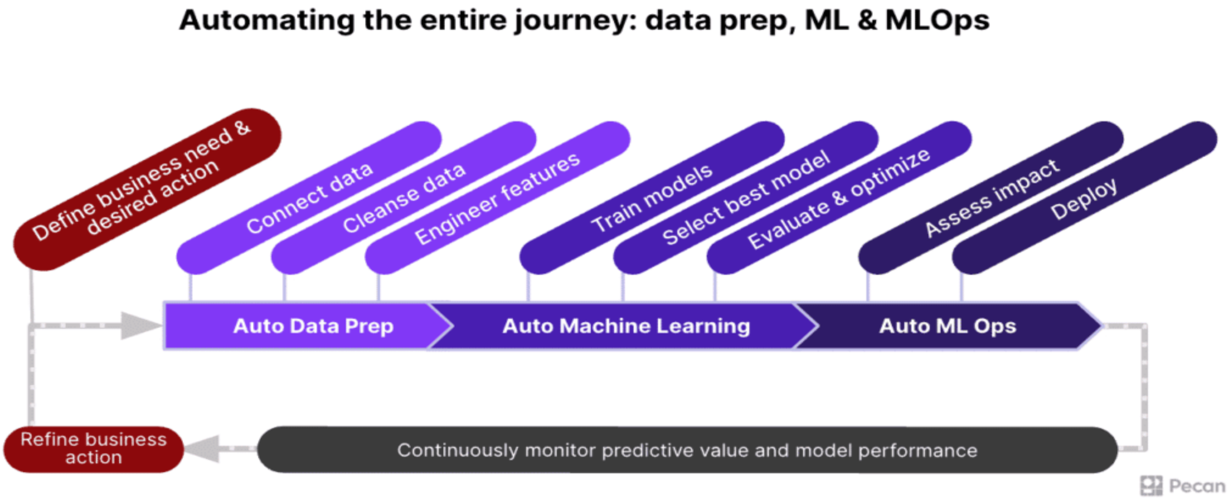

Figure 1 AI-Driven Feature Engineering for enhancing Predictive Models.

C.Fundamentals of Feature Engineering for Data Science: Traditional Techniques v/s Automated Techniques: -

C.1Fundamentals of Feature Engineering: -

C.1.1Feature Selection: Feature selection is a critical step in feature engineering that involves identifying the most relevant features from the dataset. This process begins with understanding the dataset's characteristics and domain knowledge to determine which features are likely to have the most significant impact on the target variable. [3] Various techniques can be employed for feature selection, including:

Correlation Analysis: Identifying features that are highly correlated with the target variable can help prioritize important features.

Univariate Feature Selection: Evaluating each feature individually based on statistical tests or scoring metrics and selecting the top-ranked features.

Recursive Feature Elimination: Iteratively removing less important features based on model performance until the optimal subset of features is selected.

ISSN: 2005-4297 IJCA

International Journal of Control and Automation
Vol. 13, No.04, (2020), pp.1558-1571
https://doi.org/10.52783/ijca.v13i4.38349
Feature Importance Ranking: Using machine learning models such as decision trees or random forests to rank features based on their contribution to model performance.

C.1.2Feature Transformation: Feature transformation involves converting the raw data into a format that is more suitable for modeling. [4] This process aims to address issues such as scale differences between features, categorical variables, and missing values. Common techniques for feature transformation include:

Scaling: Standardizing numerical features to have a mean of 0 and a standard deviation of 1 (Standard Scaling) or scaling features to a specified range (Min-Max Scaling).

Normalization: Scaling numerical features to a range between 0 and 1, often used when features have different scales or distributions.

Encoding Categorical Variables: Converting categorical variables into numerical representations that can be processed by machine learning algorithms, such as one-hot encoding or label encoding.

Handling Missing Values: Imputing missing values using techniques such as mean, median, or mode imputation, or more advanced methods like K-nearest neighbors (KNN) imputation or predictive modeling-based imputation.

C.1.3Feature Creation: Feature creation involves generating new features from existing ones or external sources to capture additional information or patterns in the data. This process aims to enrich the feature space and provide more discriminatory power to predictive models. Techniques for feature creation include:

Mathematical Transformations: Applying mathematical operations such as logarithms, exponentials, or square roots to numerical features to create new transformations.

Interaction Terms: Creating new features by combining two or more existing features through multiplication, addition, [5] or other mathematical operations to capture interactions between variables.

Polynomial Features: Generating polynomial features by raising existing features to higher powers, allowing models to capture nonlinear relationships between variables.

Domain-Specific Feature Engineering: Incorporating domain knowledge and expertise to create features that are relevant to the specific problem domain, such as time-based features, geographical features, or industry-specific metrics.

C.2Comparison of Traditional Approaches and Automated Techniques in Feature Engineering:

C.2.1Traditional Approaches: Traditional feature engineering methods typically involve manual exploration and manipulation of the data by data scientists or domain experts. These methods have been widely used and are often effective in practice. However, they have several limitations:

Time-Consuming: Traditional feature engineering can be a time-consuming process, requiring data scientists to manually explore different feature combinations, transformations, and selections to identify the most informative features. [6] This iterative process can be tedious and resource-intensive, especially for large datasets or complex problems.

Subjective Bias: Manual feature engineering is susceptible to subjective bias, as the selection and creation of features may be influenced by the individual preferences, assumptions, and biases of the data scientist. This can lead to suboptimal model performance or the introduction of unintentional biases into the model.

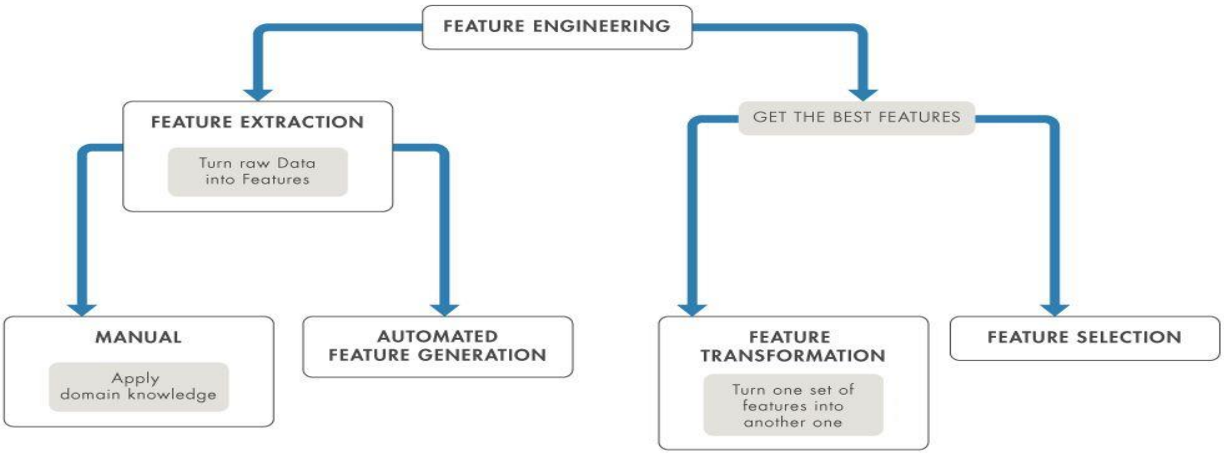

Figure 2 Types of Feature Engineering

Limited Scalability: Traditional feature engineering methods may struggle to scale to large datasets or high-dimensional feature spaces, as manual exploration and manipulation become increasingly impractical and computationally expensive. [7] This scalability limitation can hinder the application of traditional methods to real-world problems with big data.

Expertise-Dependent: Effective feature engineering often requires domain-specific knowledge and expertise to identify relevant features and transformations that are meaningful for the problem domain. This expertise dependency can be a barrier for novice data scientists or those unfamiliar with the domain, limiting their ability to perform effective feature engineering.

C.2.2Automated Techniques: In contrast, automated feature engineering techniques leverage machine learning algorithms and computational methods to automate and optimize the feature engineering process. These techniques offer several advantages over traditional approaches:

Efficiency: Automated feature engineering accelerates the feature engineering process by leveraging computational algorithms to automatically generate, select, and transform features. This automation reduces the time and effort required for model development, allowing data scientists to focus on higher-level tasks such as model selection and evaluation.

Objectivity: Automated techniques minimize subjective bias by relying on objective criteria and algorithms to identify informative features. [8] By automating feature selection, transformation, and creation, these techniques reduce the influence of individual preferences or biases, leading to more robust and reliable models.

Scalability: Automated feature engineering techniques are well-suited for handling large datasets or high-dimensional feature spaces, as they leverage computational algorithms to efficiently explore and manipulate the data. This scalability enables the application of automated techniques to real-world problems with big data, where traditional methods may struggle to scale.

Exploration of Complex Relationships: Automated techniques can explore complex relationships and interactions in the data that may be challenging to identify manually.[9] By leveraging machine learning algorithms and computational methods, automated feature engineering can uncover intricate patterns and relationships in the data, leading to improved model performance and predictive accuracy.

International Journal of Control and Automation
Vol. 13, No.04, (2020), pp.1558-1571
https://doi.org/10.52783/ijca.v13i4.38349
D. AI-Driven Automated Feature Engineering Techniques: - AI-driven automated feature engineering techniques leverage the power of artificial intelligence (AI) and machine learning (ML) algorithms to automate and optimize the process of feature selection, transformation, and creation. [10] These techniques aim to streamline the feature engineering pipeline, reducing the need for manual intervention and accelerating model development. Below are some of the key AI-driven automated feature engineering techniques:

D.1Machine Learning-Based Feature Selection Algorithms: Machine learning-based feature selection algorithms automatically identify the most relevant features from the dataset, helping to reduce dimensionality and improve model performance. These algorithms typically evaluate features based on their predictive power or importance to the target variable. Common machine learning-based feature selection techniques include:

Recursive Feature Elimination (RFE): [11] RFE recursively removes less important features based on model performance, such as the coefficients of a linear model or feature importance scores from a tree-based model.

Inputs:

- **Dataset:** X (features) and y (target variable)
- **Estimator:** Machine learning model used for feature selection (e.g., linear regression, decision tree, random forest)
- Number of features to select (optional): k

Outputs:

- **Selected features:** Subset of features selected by the RFE algorithm.

Procedure:

- Initialize the RFE algorithm with the chosen estimator and, optionally, the number of features to select (k).
- Fit the estimator to the entire dataset (X, y) to obtain the initial feature ranking.

* While the number of selected features is greater than k (if specified) or until all features have been evaluated:
a. Identify the least important feature based on the current feature ranking.
b. Remove the least important feature from the dataset.
c. Fit the estimator to the updated dataset and obtain the feature ranking.
Return the subset of features selected by the RFE algorithm.

Algorithm Pseudocode: -

function Recursive Feature Elimination (X, y, estimator, k = None):

Initialize RFE with estimator and k
Fit estimator to entire dataset (X, y) and obtain initial feature ranking
While number of selected features > k or until all features evaluated:
Identify least important [12] feature based on current feature ranking
Remove least important feature from dataset
Fit estimator to updated dataset and obtain feature ranking
Return subset of features selected by RFE

ISSN: 2005-4297 IJCA

International Journal of Control and Automation
Vol. 13, No.04, (2020), pp.1558-1571
https://doi.org/10.52783/ijca.v13i4.38349

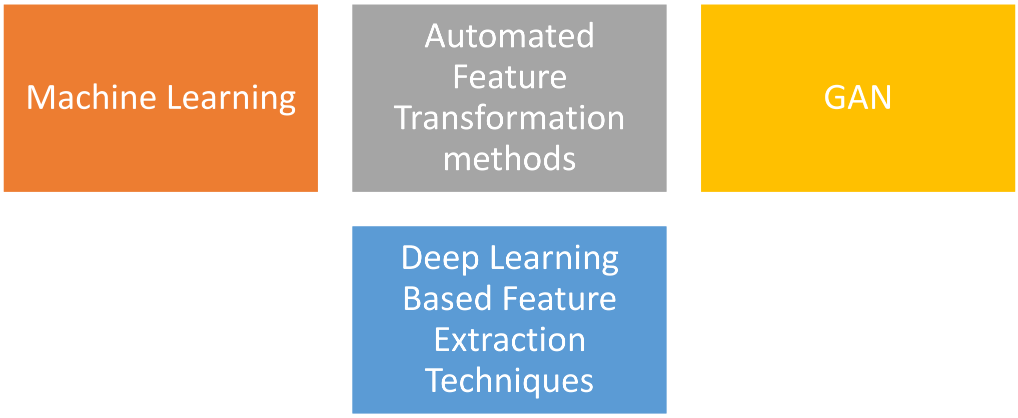

Figure 3 AI-Driven Automated Feature Engineering Techniques.

D.2Automated Feature Transformation Methods: Automated feature transformation methods focus on preprocessing numerical and categorical features to ensure they are suitable for model training. [13] These methods aim to address issues such as scale differences between features, categorical variables, and missing values. Common automated feature transformation techniques include:

Scaling: Scaling techniques like Min-Max scaling, Standard scaling, and Robust scaling ensure that numerical features are on the same scale, preventing features with larger magnitudes from dominating the model.

Encoding Categorical Variables: Encoding techniques like one-hot encoding or label encoding convert categorical variables into numerical representations that can be processed by machine learning algorithms.

Handling Missing Values: Automated methods for handling missing values include imputation techniques such as mean, median, mode imputation, or more advanced methods like K-nearest neighbors (KNN) imputation or predictive modeling-based imputation.

D.3Generative Adversarial Networks (GANs) for Feature Creation: Generative Adversarial Networks (GANs) are a type of deep learning model consisting of two neural networks: a generator and a discriminator. [14] GANs can be used to generate synthetic data samples or features that mimic the distribution of the original data. In the context of feature engineering, GANs can be trained to generate new features that capture important patterns and relationships in the data, increasing the diversity of the feature space and potentially improving model performance.

D.4Deep Learning-Based Feature Extraction Techniques:

Deep learning-based feature extraction techniques leverage neural networks to automatically learn hierarchical representations of the data. [15] These techniques can extract high-level features from raw data, capturing complex patterns and relationships that may be challenging to identify manually. Common deep learning-based feature extraction techniques include:

Convolutional Neural Networks (CNNs): CNNs are commonly used for feature extraction from image data, automatically learning hierarchical representations of visual features such as edges, textures, and shapes.

ISSN: 2005-4297 IJCA

https://doi.org/10.52783/ijca.v13i4.38349

Recurrent Neural Networks (RNNs): RNNs are well-suited for sequence data, such as time series or natural language data, capturing temporal dependencies and extracting meaningful features from sequential inputs.

Autoencoders: Autoencoders are unsupervised neural networks that learn to encode input data into a lower-dimensional latent space and then decode it back to the original input. Autoencoders can be used for feature extraction by training the network to reconstruct the input data, forcing it to learn useful representations in the latent space.

Deep Learning-Based Feature Extraction using CNN: - [16]

Inputs:

- Image dataset: X (input images)
- CNN architecture: Number of convolutional layers, filter sizes, pooling layers, etc.
- Pretrained CNN model (optional): Optionally, a pretrained CNN model can be used for feature extraction.

Outputs:

- Extracted features: High-level representations of input images obtained from the CNN's convolutional layers.

Procedure:

- Load and preprocess the input image dataset X.
- Initialize the CNN architecture with the desired configuration of convolutional layers, activation functions, pooling layers, etc.

* If using a pretrained CNN model:
a. Load the pretrained CNN model (e.g., VGG, ResNet, Inception) and remove the fully connected layers at the end.
b. Freeze the weights of the pretrained layers to prevent them from being updated during training.
* If not using a pretrained model:
a. Initialize the CNN model with random weights.
Pass the input images through the CNN model to obtain feature maps at various convolutional layers.
Extract features from the desired convolutional layer(s) by:
a. Selecting the desired layer(s) from the CNN model.
b. Extracting the output feature maps (activations) from the selected layer(s).
c. Flattening or pooling the feature maps to obtain a vector representation of each image's features.
Optionally, fine-tune the extracted features or train additional layers (e.g., fully connected layers) on top of the extracted features for specific downstream tasks.
Return the extracted features as the output.
These AI-driven automated feature engineering techniques offer several advantages, including improved model performance, time efficiency, and reduced human bias. [17] By automating and optimizing the feature engineering process, these techniques enable data scientists to focus on higher-level tasks such as model selection, evaluation, and interpretation, ultimately accelerating the development of predictive models in data science. However, it's essential to consider the trade-offs and challenges associated with these techniques, such as interpretability, computational complexity, and the need for high-quality data.

E. Tools and Frameworks for AI-Driven automated Feature: - Tools and frameworks for AI-driven automated feature engineering play a crucial role in streamlining and optimizing the feature engineering process, enabling data scientists to develop more accurate and efficient predictive models. These tools leverage machine learning algorithms and computational techniques to automate feature selection, transformation, and creation, reducing the need for manual intervention and accelerating model development.

E.1 Featuretools: -One popular tool for AI-driven automated feature engineering is Featuretools, an open-source library designed to automate the process of feature engineering for structured data. Featuretools provides a high-level interface for defining entities, relationships, and transformation primitives, allowing users to generate rich feature sets from raw data automatically. By leveraging algorithms such as Deep Feature Synthesis (DFS), Featuretools can automatically create complex features from relational datasets, [19] capturing important patterns and relationships in the data without the need for manual feature engineering.

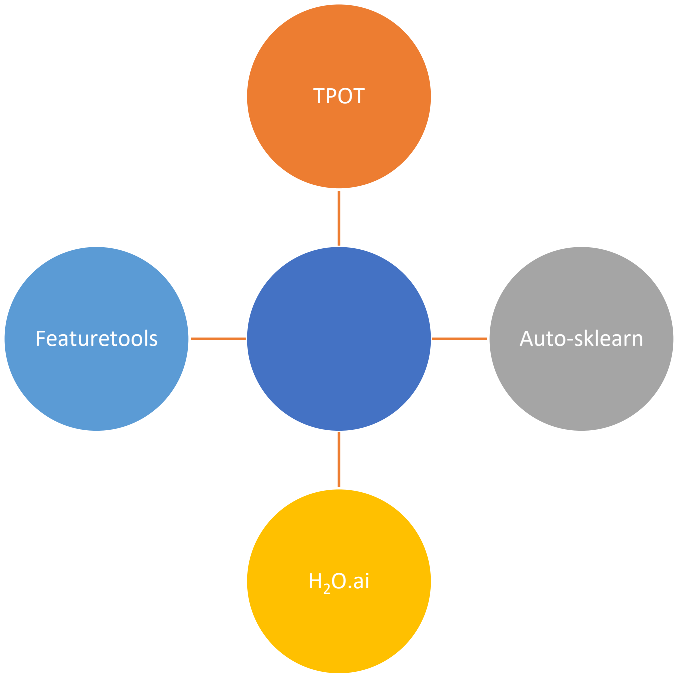

Figure 4 Tools and Frameworks.

E.2 TPOT: - Another widely used tool is TPOT (Tree-based Pipeline Optimization Tool), which employs genetic programming to automatically discover the optimal feature engineering pipeline for a given dataset and predictive modeling task. TPOT searches through a space of possible feature preprocessing steps and machine learning algorithms to find the best combination that maximizes model performance. By automating the feature engineering pipeline, TPOT helps data scientists save time and resources while producing high-quality predictive models.

E.3 Auto-sklearn: - Auto-sklearn is another notable tool that provides automated machine learning capabilities, including automated feature engineering. Auto-sklearn uses Bayesian optimization to search for the optimal preprocessing steps and model hyperparameters, including feature selection, transformation, and creation. By automating the entire machine learning pipeline, Auto-sklearn enables data scientists to develop highly accurate predictive models with minimal manual intervention.

E.4 H2O.ai: - is a comprehensive platform that offers automated machine learning and feature engineering capabilities through its H2O AutoML and H2O Feature Engineering modules. H2O AutoML automates the process of model selection, hyperparameter optimization, and feature engineering, allowing users to develop highly accurate predictive models with ease. H2O Feature Engineering provides a range of automated feature engineering techniques, including feature selection, transformation, and creation, to enhance model performance and interpretability.

F. Benefits of AI-Driven Automated Feature Engineering: - The emergence of AI-driven automated feature engineering has revolutionized the field of data science, offering a wide array of benefits that significantly enhance the efficiency, effectiveness, and scalability of predictive modeling. Below are some of the key benefits of AI-driven automated feature engineering:
Time Efficiency: One of the primary advantages of AI-driven automated feature engineering is its ability to significantly reduce the time required for model development. Traditional feature [2be time-consuming and resource-intensive. [20] By automating the feature engineering process, AI-driven techniques streamline the workflow, allowing data scientists to generate rich feature sets and develop predictive models more quickly.
Improved Model Performance: Automated feature engineering techniques leverage machine learning algorithms and computational methods to identify, transform, and create informative features that enhance model performance. By automatically capturing relevant patterns and relationships in the data, these techniques enable data scientists to develop more accurate and robust predictive models. Additionally, automated feature engineering can help reduce the risk of overfitting by selecting only the most informative features, improving the generalization ability of the models.
Scalability: AI-driven automated feature engineering techniques are well-suited for handling large datasets and high-dimensional feature spaces, making them highly scalable to real-world applications with big data. Traditional feature engineering methods may struggle to scale to large datasets due to manual intervention and computational limitations. [21]By leveraging machine learning algorithms and computational resources, automated techniques can efficiently explore and manipulate large volumes of data, enabling data scientists to tackle complex problems with ease.
Reduction of Human Bias: Traditional feature engineering methods are often influenced by the biases and assumptions of the data scientist, potentially leading to suboptimal model performance or unintentional biases in the models. AI-driven automated feature engineering techniques minimize subjective bias by relying on objective criteria and algorithms to identify informative features. By automating the feature selection, transformation, and creation process, these techniques reduce the influence of individual preferences and biases, leading to more robust and reliable models.

Exploration of Complex Relationships: Automated feature engineering techniques can explore complex relationships and interactions in the data that may be challenging to identify manually. By leveraging advanced machine learning algorithms and computational methods, these techniques can uncover intricate patterns and dependencies in the data, leading to improved model performance and predictive accuracy. Additionally, automated feature engineering can capture higher-order relationships and interactions that may be overlooked by traditional methods, providing deeper insights into the underlying structure of the data.

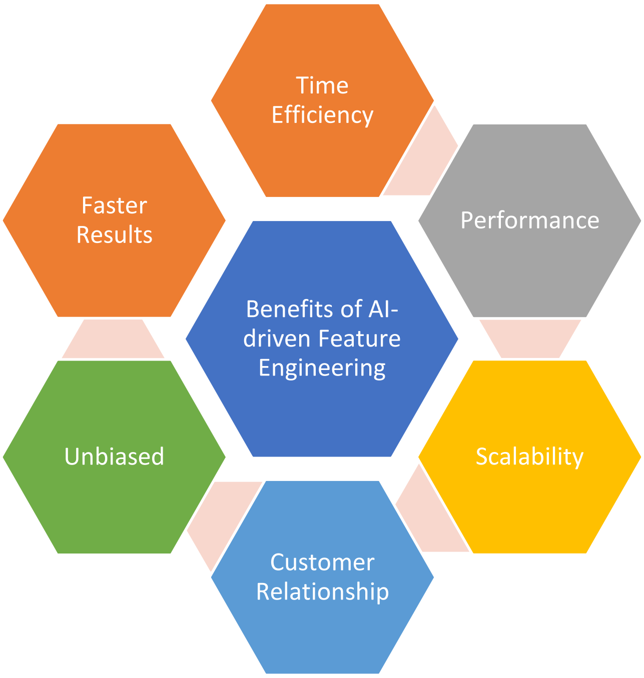

Figure 5 Benefits of AI-driven Feature Engineering.

G. Challenges and Limitations of AI-Driven automated Feature Engineering: - While AI-driven automated feature engineering offers numerous benefits, it also presents several challenges that need to be addressed to realize its full potential. Some of the key challenges include:

Interpretability: Automated feature engineering techniques often generate complex features or transformations that are difficult to interpret or explain. This lack of interpretability can hinder the understanding of the model's decision-making process and make it challenging to gain insights into the underlying data patterns. [22] Addressing this challenge requires developing methods for explaining and visualizing the automated feature engineering process and its impact on model performance.

Data Quality Requirements: Automated feature engineering techniques rely on the quality and consistency of the input data to generate informative features. Noisy or inconsistent data can lead to inaccurate feature representations and degrade model performance. [23] Ensuring data quality through data cleaning, preprocessing, and validation procedures is essential to mitigate this challenge and obtain reliable feature sets.

Computational Complexity: Automated feature engineering techniques can be computationally intensive, especially when dealing with large datasets or complex feature spaces. The process of searching for optimal feature transformations or combinations may require significant computational resources and time.[24] Improving the efficiency and scalability of automated feature engineering algorithms is crucial to address this challenge and make them accessible for real-world applications with big data.

Overfitting: Automated feature engineering techniques may lead to overfitting if not properly regularized or validated. Generating a large number of features or using complex feature transformations can increase the risk of overfitting, where the model learns noise or spurious patterns in the training data that do not generalize to unseen data. Regularization techniques such as feature selection, cross-validation, and model validation are essential to mitigate the risk of overfitting and ensure the generalization ability of the models.

Domain-specific Knowledge: While automated feature engineering techniques automate many aspects of the feature engineering process, they may lack domain-specific knowledge and context that human experts possess. Incorporating domain knowledge and expertise into the automated feature engineering pipeline is essential to ensure that the generated features are relevant, meaningful, and aligned with the problem domain. Collaboration between data scientists and domain experts is crucial to address this challenge and develop effective feature engineering solutions tailored to specific applications.

Ethical Considerations: Automated feature engineering techniques may inadvertently amplify biases present in the data or introduce new biases into the models. [25] This can lead to unfair or discriminatory outcomes, especially in sensitive domains such as healthcare, finance, and criminal justice. Addressing ethical considerations such as fairness, transparency, and accountability in automated feature engineering requires careful attention to data collection, preprocessing, and model development processes to mitigate biases and ensure equitable outcomes.

H. Conclusion: - In conclusion, AI-driven automated feature engineering represents a paradigm shift in the field of data science, offering significant advancements in the development of predictive models. This paper has explored the fundamentals, techniques, benefits, and challenges of AI-driven automated feature engineering in enhancing the performance of predictive models. Automated feature engineering leverages the power of artificial intelligence and machine learning algorithms to streamline and optimize the feature engineering process. By automating feature selection, transformation, and creation, these techniques reduce manual intervention, accelerate model development, and improve efficiency. The benefits of AI-driven automated feature engineering are manifold. These techniques enhance model performance by generating informative features, reducing overfitting, and capturing complex data patterns. They also improve time efficiency, scalability, and reduce human bias in the feature engineering process. However, AI-driven automated feature engineering also presents several challenges. Interpretability, data quality requirements, computational complexity, overfitting, domain-specific knowledge, and ethical considerations are among the key challenges that need to be addressed to realize its full potential. Despite these challenges, the potential of AI-driven automated feature engineering to revolutionize predictive modeling in data science is undeniable. By addressing the challenges and leveraging the benefits, researchers and practitioners can develop more accurate, scalable, and unbiased predictive models that enable data-driven decision-making across diverse domains.

In conclusion, AI-driven automated feature engineering holds immense promise in enhancing the performance of predictive models in data science. By harnessing the power of artificial intelligence and machine learning, automated feature engineering enables data scientists to unlock valuable insights from data and drive innovation in predictive modeling, ultimately leading to more informed decision-making and societal impact.

References: -

[1] Chen, K., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

[2] Fernandez-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research, 15(1), 3133–3181.

[3] Van Rijn, J. N., Holmes, G., Pfahringer, B., & Vanschoren, J. (2018). Learning Classifier Systems for Data Mining: An Overview and Outlook. Evolutionary Intelligence, 11(1), 1–29.

[4] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798–1828.

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

[6] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. Statistical Science, 17(3), 235–249.

[7] Papamichail, M., Yang, H., & Papavassilopoulos, G. P. (2019). Credit Scoring Using Deep Learning. International Journal of Financial Studies, 7(1), 3.

[8] Rajkomar, A., Dean, J., & Kohane, I. (2018). Machine Learning in Medicine. The New England Journal of Medicine, 380(14), 1347–1358.

[9] Choi, E., Bahadori, M. T., & Schuetz, A. (2020). RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In Proceedings of the 30th International Conference on Machine Learning (ICML).

[10] Featuretools. (n.d.). Automated Feature Engineering in Python. Retrieved from https://www.featuretools.com/

[11] Chen, K., & Guestrin, C. (2016). Mining Structure-Property Maps of Organic Materials with Functional Group–Directed Automated Feature Generation. ACS Central Science, 2(10), 652–661.

[12] TPOT: A Python Automated Machine Learning Tool. (n.d.). Retrieved from http://epistasislab.github.io/tpot/

[13] Feurer, M., Klein, A., & Eggensperger, K. (2015). Efficient and Robust Automated Machine Learning. Advances in Neural Information Processing Systems, 28, 2962–2970.

[14] Auto-sklearn: Automated Machine Learning Toolkit. (n.d.). Retrieved from https://automl.github.io/auto-sklearn/master/

[15] Olson, R. S., Bartley, N., & Urbanowicz, R. J. (2016). Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In GECCO '16: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, 485–492.

[16] H2O.ai. (n.d.). H2O AutoML. Retrieved from https://www.h2o.ai/products/h2o-automl/

[17] Featuretools. (2020). Featuretools: Automated Feature Engineering in Python. Journal of Open Source Software, 5(54), 2175.

[18] Alaa, A. M., & van der Schaar, M. (2019). AutoPrognosis: Automated Clinical Prognostic Modeling via Bayesian Optimization with Structured Kernel Learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2970–2978.

[19] Pedregosa, F., Varoquaux, G., & Gramfort, A. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

https://doi.org/10.52783/ijca.v13i4.38349
[20] Shumway, R. H., & Stoffer, D. S. (2017). Time Series Analysis and Its Applications: With R Examples. Springer.
[21] Bhargava, R. (2020). Automated Feature Engineering. Medium. Retrieved from https://towardsdatascience.com/automated-feature-engineering-in-python-99baf11cc219
[22] Amidi, A., & Ma, J. (2019). The role of automated feature engineering in machine learning. IBM Developer. Retrieved from https://developer.ibm.com/technologies/artificial-intelligence/articles/the-role-of-automated-feature-engineering-in-machine-learning/
[23] Varma, P., & Rajesh, A. (2019). An overview of Automated Feature Engineering. Medium. Retrieved from https://medium.com/datadriveninvestor/an-overview-of-automated-feature-engineering-df786f12322b
[24] Khandelwal, P. (2019). Automated Feature Engineering — The Next Big Thing in Data Science. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2019/12/automated-feature-engineering-the-next-big-thing/
[25] Buhlmann, P., & Yu, B. (2003). Boosting with the L2 Loss: Regression and Classification. Journal of the American Statistical Association, 98(462), 324–339.

ISSN: 2005-4297 IJCA