Automating Feature Selection and Engineering for Predictive Modeling

Bar Katash

March 5, 2025

Abstract

This project addresses the challenge of improving feature selection and engineering in the predictive modeling pipeline. In part 2 of the project, feature selection and engineering were performed manually, which proved to be time-consuming and prone to human bias. This work presents an automated solution to dynamically select and engineer relevant features, thereby improving model accuracy and reducing manual intervention. By implementing techniques such as correlation-based feature elimination and transformation strategies, the solution aims to enhance model performance while maintaining scalability. Experimental results show improvements in accuracy when applying the automated process compared to traditional manual approaches.

1 Problem Description

Feature selection and engineering play a crucial role in the data science pipeline, directly influencing model performance, interpretability, and computational efficiency. However, traditional methods for feature selection and transformation often suffer from several issues, which limit their effectiveness:

- **Time-consuming and manual effort:** In previous phases of this project, feature selection and engineering were performed manually, requiring significant domain knowledge and iterative experimentation.

- **Lack of a systematic evaluation framework:** Features were selected based on intuition rather than data-driven statistical and machine learning techniques. This led to suboptimal feature sets, where relevant information might be ignored, or irrelevant features might be retained.

- **Redundancy:** High correlation between independent variables can inflate variance in model coefficients, making interpretation difficult and potentially reducing predictive performance.

- **Static transformations with poor generalization:** Handcrafted feature transformations, such as log or square root transformations, were applied uniformly across datasets without assessing their impact, leading to models that failed to generalize well across different data distributions.

- **Computational inefficiency:** Large datasets with high-dimensional feature spaces lead to increased training times and overfitting risks if irrelevant features are not eliminated effectively.

2 Solution Overview

The proposed solution involves automating the feature selection and engineering process in the predictive modeling pipeline. The methodology integrates statistical principles such as **correlation measures, independence tests, variance analysis, and distributional properties** to systematically refine input features. The key components include:

2.1 Feature Selection

- **Correlation-Based Filtering**: Compute pairwise correlations between features and remove those with high correlations (above a set threshold), leveraging **correlation**

coefficients to detect redundant features.

- **Variance Thresholding**: Eliminate features with low variance, as they contribute little to prediction, applying **biased/unbiased variance estimation** to determine optimal thresholds.
- **Chi-Square Test for Categorical Features**: Apply **independence tests (Chi-square)** to evaluate associations between categorical features and the target variable, ensuring statistical relevance.

2.2 Feature Engineering

- **Transformation of Skewed Features**: Apply **moment-based skewness and kurtosis analysis** to detect asymmetry and apply transformations (logarithmic, square root, power) to normalize distributions.
- **Time-Based Feature Extraction**: Extract components such as year, month, and day from time-based features, and compute lag or difference values to capture temporal relationships.

The solution will dynamically evaluate the impact of each transformation or feature and retain only those that improve model performance based on statistical significance.

3 Experimental Evaluation

To evaluate the effectiveness of the proposed automated feature selection and engineering approach, we conducted experiments on four datasets: Video Games, Property Sales, Car Prices, and Flight Prices. The evaluation follows these steps:

- Train a baseline model using the original dataset without automated feature selection and engineering.

- Apply the proposed automated feature selection and engineering process to the dataset.
- Train an enhanced model using the transformed dataset.
- Compare the performance of the baseline and enhanced models using the following metrics:
  - Mean Absolute Error (MAE)
  - Mean Absolute Percentage Error (MAPE)
  - Root Mean Squared Error (RMSE)
  - Coefficient of Determination (R^2)
- Measure and compare the training times of both models.

3.1 Results and Analysis

The results indicate that the enhanced model generally improves performance, particularly in terms of R^2, demonstrating better model fit. However, the overall improvement in performance metrics is relatively small across all datasets.

Additionally, the training time is significantly reduced for the Video Games dataset, indicating computational efficiency. However, for some datasets, such as Flight Prices, the improvement in performance is marginal.

The results suggest that while automated feature selection and engineering can enhance model accuracy and efficiency, the extent of improvement varies depending on the dataset characteristics. Further analysis can explore more advanced feature selection techniques to refine the approach.

Comparison Graphs

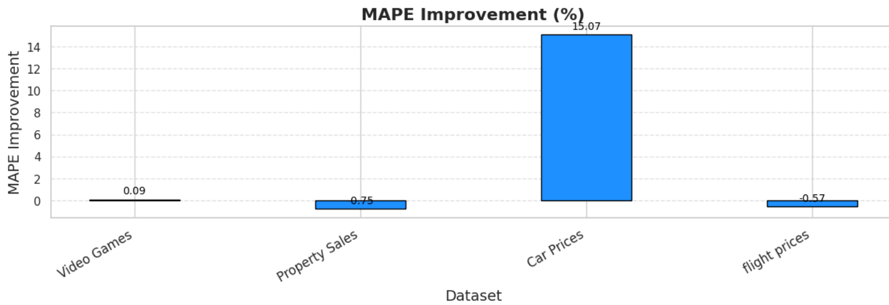

Figure 1: Graphical Comparison of MAPE

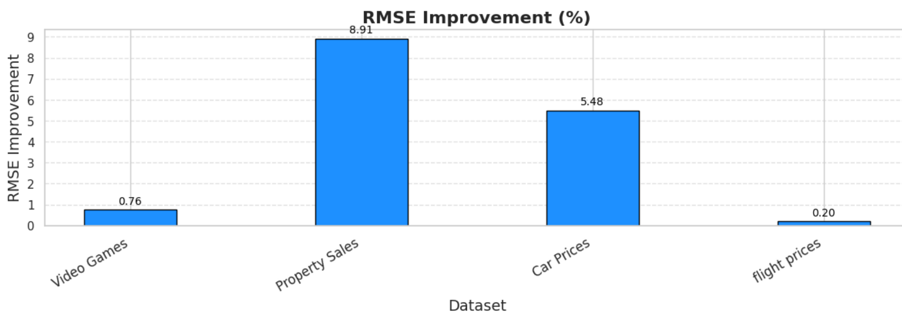

Figure 2: Graphical Comparison of RMSE

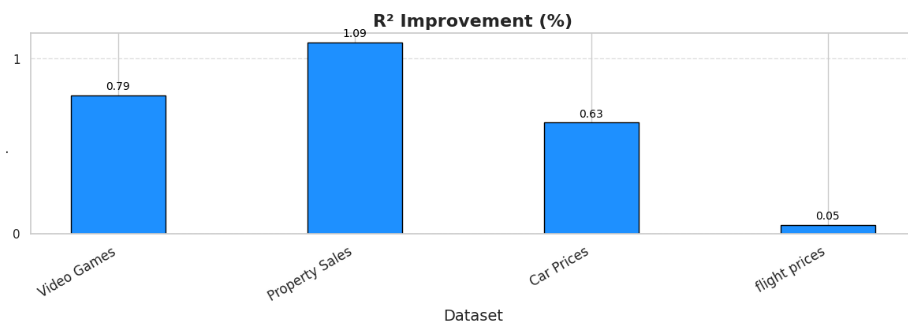

Figure 3: Graphical Comparison of R^2

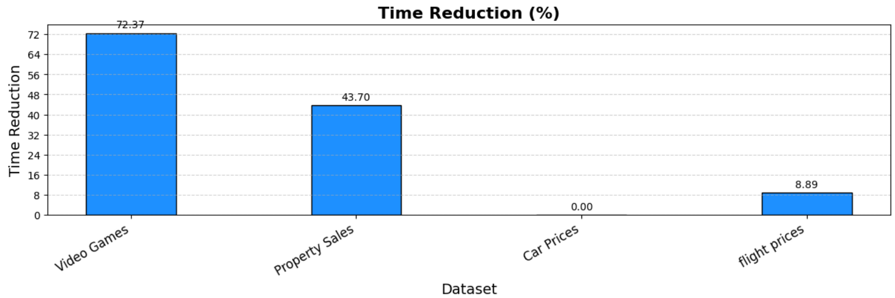

Figure 4: Graphical Comparison of Run Time

4 Related Work

Feature selection and engineering are essential steps in the machine learning pipeline. Numerous studies and tools have been proposed to automate these processes, and several of them align with the objectives of my research proposal.

One key resource, *Feature Engineering for Machine Learning* by Zhang and Su (2021), explores various statistical and machine learning techniques for feature creation and pruning, including automated approaches [1]. The paper discusses the importance of systematically selecting features based on their relevance to the predictive model and highlights the role of automation in reducing human error and bias. This paper directly aligns with my goal of automating feature selection and engineering, and its techniques inspired the idea of using correlation-based feature selection and statistical tests like the Chi-square test for categorical features.

Another relevant work, *Feature Selection Methods and Applications* by Sun et al. (2021), discusses filter-based feature selection methods, such as correlation-based methods and variance thresholds [2]. This paper is particularly useful as it emphasizes the need for removing redundant features and selecting only those with high predictive power. The insights from this article shaped my approach to selecting features with a high degree of correlation with the target variable, and also inspired my use of statistical tests to identify feature relevance.

On the tool front, *FeatureTools* is a widely used Python library for automating feature engineering. It employs a method called Deep Feature Synthesis (DFS) to create new features from raw data by automatically generating aggregations and hierarchical relationships between entities. FeatureTools is particularly useful for relational datasets, and its principles can be adapted to my research project, where the goal is to automate feature engineering, including creating interaction features and applying transformations like logarithmic or power transformations.

In terms of comparing existing techniques, these tools and methods often focus on different aspects of feature selection and engineering, such as using statistical measures, correlation, or more sophisticated machine learning algorithms for feature importance. The difference in my solution lies in its automation and dynamic feature generation, which not only selects relevant features but also generates new ones by applying transformations and interactions. While existing tools like FeatureTools automate feature creation, my approach distinguishes itself by systematically evaluating the impact of these features on model performance, ensuring that only the most relevant and effective features are retained.

References

[1] Zhang, J., & Su, M. (2021). Feature Engineering for Machine Learning. *Towards Data Science*. Retrieved from https://towardsdatascience.com/feature-engineering-for-machine-learning-eb2e0cff7a30/

[2] Sun, L., Wang, H., & Liu, Y. (2021). Feature Selection Methods and Applications. *MDPI*. Retrieved from https://www.mdpi.com/2078-2489/14/3/191

5 Conclusion

In this project, I learned that automated feature selection methods, such as correlation-based techniques and statistical tests, can enhance model accuracy. By dynamically identifying relevant features and applying necessary transformations, the models can generalize better across various datasets and avoid errors caused by irrelevant or redundant features. However, I also found that the effectiveness of the solution can vary depending on the dataset. Different datasets may yield different results, and what works well for one dataset may not generalize to another, making it challenging to apply a one-size-fits-all approach.

I also realized the importance of applying proper transformations, such as logarithmic or power transformations, to handle skewed features, which can improve the performance of machine learning models. This not only enhances accuracy but also increases the robustness of the model when applied to unseen data.

The results demonstrated that this approach reduces computational overhead while still providing a high level of predictive accuracy. However, the improvements were not always consistent across all datasets, which highlights the complexity of generalizing the solution.

In conclusion, this project helped me understand how crucial it is to automate repetitive tasks like feature selection and transformation in the machine learning pipeline. It provided insights into how automation can not only save time but also improve the overall quality of machine learning models, especially when dealing with large and complex datasets. Moving forward, I plan to further explore additional methods for automating other steps in the machine learning pipeline to continue improving efficiency and accuracy, while keeping in mind the variability of results depending on the dataset.