Shell.ai Hackathon for Sustainable and Affordable Hackathon

Fuel Blend Properties Prediction Challenge 2025

Level 2: Prototype Submission Phase

Eagle-Team

Destiny Otto

Alexander Ifenaike

Godswill Otto

William Alabi

1. Introduction

The design of sustainable aviation fuels (SAFs) and other renewable fuel blends presents significant challenges due to the complexity of property interactions and stringent quality requirements. Traditional trial-and-error approaches to blend development are increasingly being replaced by integrated design methods that simultaneously optimize both composition and production pathways (König et al., 2020). This shift underscores the need for accurate prediction techniques, as blending rules based solely on component data often struggle to capture the nonlinear effects of multiple blend target properties (Boehm et al., 2024).

Green fuel blends, including bio-alcohols and biodiesel, have been shown to reduce greenhouse gas emissions and harmful pollutants in the transport sector (Chakraborty & Mukhopadhyay, 2019). However, the literature highlights persistent limitations in conventional analytical procedures, especially regarding the precision and accuracy needed for regulatory compliance (Pahl & McNally, 1990). These constraints complicate the certification of new blends and emphasize the importance of developing predictive methods that can account for variability across blend properties.

Historically, blending strategies relied on experimental programs, such as the Auto/Oil Air Quality Improvement Research Program, which exposed challenges in reproducibility, particularly for oxygenated fuels (Pahl & McNally, 1990). Research into combustion characteristics further demonstrates the importance of parameters like laminar burning velocity and ignition delay time in understanding blend behavior (Mumby, 2016). These studies collectively reveal the gaps that predictive modeling must address.

In recent years, machine learning (ML) and deep learning (DL) approaches have emerged as promising tools to model such complexity. For instance, Support Vector Machines and Random Forests have shown success in predicting ignition-related properties, such as Research Octane Number (RON), with high accuracy (Correa Gonzalez et al., 2021).

Building on this momentum, the present project applies ML/DL methods with four key objectives:

1. **Rapidly evaluate** thousands of potential blend combinations.
2. **Identify optimal recipes** that maximize sustainability while meeting technical specifications.
3. **Reduce development time** for new sustainable fuel formulations.
4. **Enable real-time blend optimization** in production facilities.

The remainder of this work outlines the methodology used to design and evaluate the predictive models, the assumptions and limitations underpinning this approach, the data sources employed, and an analysis of model performance on both validation and external test sets.

2. Methodology

2.1 Overview of Approach

The prediction of blended fuel properties was framed as a regression task, since all ten target properties of interest were continuous variables. This framing is important because regression models are well suited to estimate quantitative property values directly from numerical features. The workflow adopted in this project followed a structured pipeline: **data preparation, preprocessing, feature engineering, model training, and evaluation**. The dataset was supplied in a scaled form, with distributions resembling normalized values when compared to their means and ranges, which reduced the need for extensive preprocessing. To improve stability for certain blend target properties, an ensemble strategy was applied by training models across different folds of the training set and averaging their outputs. This approach supported more consistent predictions across properties with varying levels of complexity.

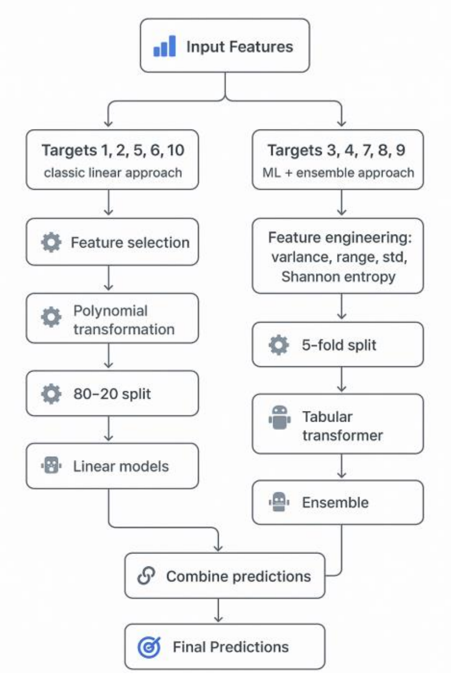

Figure 1 Flow Chart showing the development of machine learning models

2.2 Data Preparation

The dataset contained approx

imately 2,000 blend samples derived from five component fuels, each sample associated with ten blend target properties. A key advantage of the dataset was its cleanliness: **no missing values** were observed, and all features were already scaled, which simplified preprocessing. This ensured that component-level data and blend-level outputs could be directly applied within ML/DL models without extensive normalization or imputation procedures. To evaluate performance, the dataset was partitioned into **80% training and 20% validation sets**, a standard ratio that balances model fitting with reliable assessment. This split allowed for both robust learning and a fair check of the model's generalization capacity.

2.3 Feature Engineering

Feature engineering was critical in enabling the models to learn both straightforward and complex dependencies between component fuels and their blend targets. A closer inspection of the dataset revealed that not all blend target properties behaved in the same way.

For certain blend targets, the dependency structure was **highly linear and property-specific**. For example, some blended targets could be almost entirely determined by combining the five weight fractions with their corresponding component property values (e.g., a given blended property was primarily dependent on the five weight fractions and the five component values of the same property). In such cases, **linear blending rules** acted as effective predictors, where the contribution of each component scaled directly with its proportion in the blend. To capture subtle deviations from perfect linearity, additional transformations such as **polynomial features** and **interaction terms** were introduced on this restricted set of ten inputs (five weights and five component values). This targeted approach reduced feature noise and allowed the models to focus on the most relevant inputs.

In contrast, other blend targets did not exhibit clear linearity and required a broader set of engineered features. For these properties, derived variables such as the **variance, range, and weighted aggregates** of component values were created. These features captured distributional aspects of the blend composition, highlighting not just the mean contribution of each fuel but also how dispersed or unevenly distributed properties were across the components. Furthermore, **cross-interactions** between weights and component properties were introduced to represent nonlinear blending effects, and in some cases, property–property interactions were added to approximate synergistic behavior between fuels.

By tailoring the feature engineering process to the dependency structure of each blend target, the models were provided with a richer and more flexible input space. This ensured that simpler targets were modeled efficiently without unnecessary complexity, while more nonlinear targets benefited from additional expressive power through engineered features. The result was a balance between interpretability, parsimony, and predictive accuracy across the entire range of blend target properties.

2.4 Model Selection, Training and Validation

Selecting appropriate models was a critical step in predicting the ten blend target properties, as each property displayed different levels of complexity. To account for this, multiple algorithms were considered, including **linear regression, gradient boosting (XGBoost), transformer-based architectures**. In practice, the final solution employed a mix of models: linear regression for targets with strong linear trends, XGBoost for properties with moderate nonlinearity, and two transformer-based models for more complex relationships.

Among the transformer models, **TabPFN (Tabular Prior-data Fitted Network)** played a unique role. TabPFN is a pre-trained transformer capable of handling tabular data without extensive hyperparameter tuning. Its design allows for rapid evaluation of small datasets, delivering high accuracy on unseen data within seconds. This property aligned well with the project's need to quickly assess blend combinations while minimizing computational overhead.

Model performance across all approaches was assessed using **Root Mean Squared Error (RMSE)** and **Mean Absolute Percentage Error (MAPE)**. These metrics were chosen to capture both absolute error magnitudes and relative proportional deviations. To ensure robust evaluation, **five-fold cross-validation** was applied across candidate models. For non-transformer algorithms, **grid search with five-fold validation** was used to fine-tune hyperparameters. Together, this process ensured that the final selected models were accurate, consistent, and well-calibrated for prediction.

2.5 Optimization

Beyond predictive accuracy, the project aimed to enable **optimization of blend recipes** for specific performance goals. A key feature was introduced to adjust component fractions dynamically to achieve one, several, or all target properties simultaneously. This formulation naturally led to a **multi-objective optimization problem**, where trade-offs between different properties had to be balanced.

To address this, the **Non-dominated Sorting Genetic Algorithm II (NSGA-II)** was adopted. NSGA-II is an evolutionary optimization algorithm designed to efficiently explore high-dimensional search spaces while maintaining a diverse set of Pareto-optimal solutions. The algorithm works by generating a population of candidate solutions (blend weight combinations), evaluating their performance using the predictive models, and iteratively evolving them through selection, crossover, and mutation. The **cost function** was defined using RMSE, penalizing large deviations between predicted and desired property values.

Practically, the optimization process generated batches of candidate blends by systematically varying component weights. These were fed into the predictive models in **vectorized form**, enabling parallel evaluation across CPU and GPU resources. This approach minimized repetitive function calls, thereby reducing computational time while exploring a broad solution space. As a result, the optimization framework not only identified blends meeting specifications but also provided insight into the trade-offs between competing property targets.

3. Assumptions and Limitations

In this work, several assumptions were necessary to frame the predictive modeling task. First, it was assumed that the component property data provided in the hackathon dataset was accurate and consistent, serving as a reliable foundation for machine learning. Second, the features supplied were assumed to capture the essential chemical and physical interactions between fuel components, sufficient for modeling blend outcomes.

The study is not without limitations. The dataset size was modest compared to industrial-scale datasets, restricting model generalization. Results are bound to the scope of the provided data and may not transfer directly to fuel families beyond those represented. Finally, while deep learning models offer expressive power, their computational cost and training time limited their practicality compared to more lightweight machine learning models.

4. Data Source

Shell Global provided the dataset used in this project **as part of the Shell.ai hackathon** and formed the basis for both model development and evaluation. It comprised two main files: a training dataset (train.csv) and a test dataset (test.csv).

The training dataset contained 2,000 unique fuel blend records, each represented by 65 columns grouped into three categories. The first five columns described blend composition as the volume fractions of five base components. The next 50 columns captured component properties, structured as {component_number}_{property_number} for each of the five fuels and their ten anonymized properties. These values simulated a Certificate of Analysis, offering a detailed characterization of component-level attributes. The final ten columns contained the blend target properties, which served as prediction targets for the models.

The test dataset included 500 additional blends, with the same 55 input features but without target properties. Predictions for this set were submitted on the HackerRank platform, where a leaderboard score quantified model performance.

5. Results and Evaluation

Predictive model performance was rigorously evaluated using a 5-fold cross-validation methodology. This approach ensures that the model's effectiveness is assessed across different subsets of the data, providing a robust measure of its ability to generalize to unseen data. The evaluation was based on two key metrics: Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE), which together offer a comprehensive view of the model's predictive accuracy.

5.1 Evaluation Metrics

The primary metrics used for evaluating the model's performance were Mean Squared Error and Mean Absolute Percentage Error.

- **Mean Squared Error (MSE)**: The mean squared error (MSE) is a perfect performance metric for models that predict continuous variables because of its connection to the information theory concept of cross-entropy. The degree of resemblance between two probability distributions is measured by cross-entropy. The "best" model reduces the cross-entropy between the model predictions and the training data if the objective of modelling is to find the model that most closely replicates the actual data-generating distribution (Hodson, 2021). MSE is a widely used metric in regression tasks that measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. By squaring the errors, MSE gives higher weight to larger errors, which is particularly useful when significant deviations are undesirable (Hodson, 2021). A lower MSE value indicates a better fit of the model to the data. Across the five folds, the TabPFN regressor demonstrated consistent performance with an **average MSE of 0.0070**.

- **Mean Absolute Percentage Error (MAPE)**: MAPE measures the average absolute percent error between predicted and actual values. Due to its highly intuitive interpretation in terms of relative error, the MAPE is frequently employed in practice. For example, as gains and losses are frequently expressed in relative terms, the application of the MAPE is pertinent in the financial industry. Since consumers can occasionally be more sensitive to relative changes than to absolute ones, it is also helpful to calibrate product prices (Myttenaere, 2016). For instance, a MAPE of 10% signifies that, on average, the model's predictions are 10% away from the actual values.

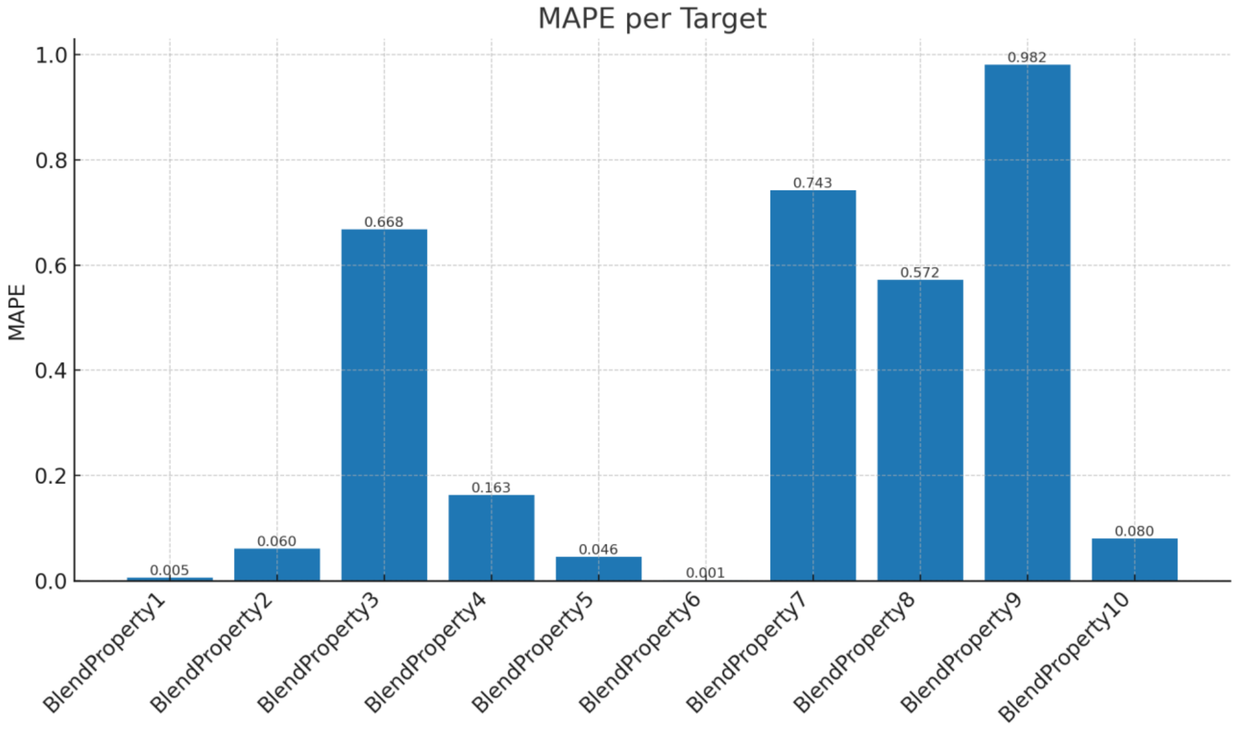

Figure 2 Bar Chart showing the MAPE for each blend property

5.2 Model Comparison and Performance Insights

The model employed in this analysis is the AutoTabPFNRegressor, which is based on a Transformer architecture, a type of deep learning (DL) model. This marks a departure from traditional machine learning (ML) models like Gradient Boosted Decision Trees (e.g., XGBoost, LightGBM) that have historically dominated tabular data tasks.

TabPFN performs better than an ensemble of the best baselines tweaked for four hours. This foundation model, which is built on generative transformers, also enables learning reusable embeddings, density estimates, data production, and fine-tuning. The effectiveness of this method for algorithm creation is demonstrated by the learning algorithm TabPFN, which learns itself from millions of synthetic datasets. Even after 4 hours of tweaking, TabPFN achieves a speedup of

5,140× (classification) and 3,000× (regression) in a single forward pass, outperforming state-of-the-art baselines on our benchmarks, such as gradient-boosted decision trees (Hollmann, 2025).

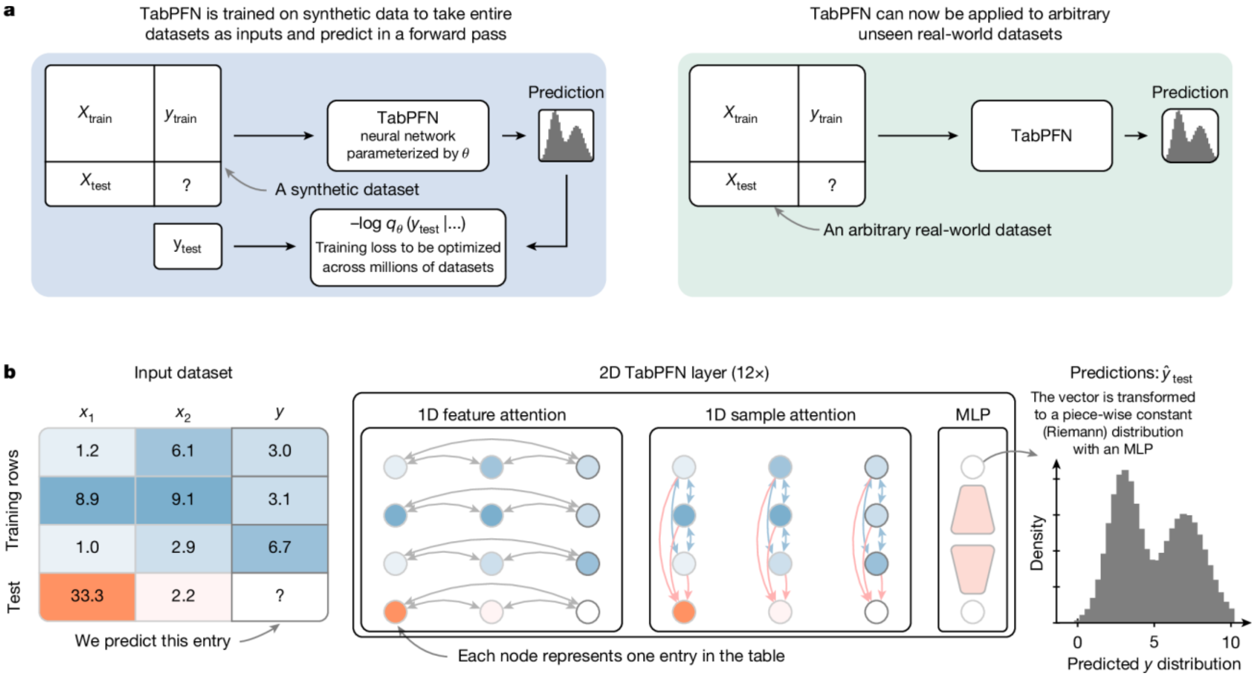
a
TabPFN is trained on synthetic data to take entire datasets as inputs and predict in a forward pass
X_{train}
y_{train}
X_{test}
?
TabPFN neural network parameterized by θ
Prediction
A synthetic dataset
y_{test}
$-\log q_\theta(y_{test}|...)$
Training loss to be optimized across millions of datasets
TabPFN can now be applied to arbitrary unseen real-world datasets
X_{train}
y_{train}
X_{test}
?
TabPFN
Prediction
An arbitrary real-world dataset
b
Input dataset
x_1
x_2
y
Training rows
Test
1.2
6.1
3.0
8.9
9.1
3.1
1.0
2.9
6.7
33.3
2.2
?
We predict this entry
2D TabPFN layer (12×)
1D feature attention
1D sample attention
MLP
Each node represents one entry in the table
Predictions: $\hat{y}_{test}$
The vector is transformed to a piece-wise constant (Riemann) distribution with an MLP
Density
0
5
10
Predicted y distribution

Figure 5.2: **a**, The high-level overview of TabPFN pre-training and usage. **b**, The TabPFN architecture.

The tabpfn model was trained on over 100 million synthetic tasks. The architecture employed is a modified version of the standard transformer encoder, tailored to handle the two-dimensional data structures characteristic of tables (Hollmann, 2025).

5.3 Example Case: Prediction vs. Actual for Selected Blend Ratios

Case 1: A Well-Predicted Blend

Table 1 Well Predicted Blend

roperty	Actual	Predicted	MAPE
BlendProperty1	0.6335	0.6322	0.0021
BlendProperty2	1.8824	1.9149	0.0173
BlendProperty3	-1.1350	-0.9780	0.1380
BlendProperty9	1.8411	1.7819	0.0322

Takeaway: Absolute errors are small, but for low-magnitude targets (e.g., BlendProperty9), percentage errors inflate, exposing MAPE's sensitivity.

Case 2: A Challenging Blend with High MAPE

Table 2 Challenging Blend with High MAPE

Property	Actual	Predicted	MAPE
BlendProperty3	1.4782	1.6219	0.0972
BlendProperty5	3.2372	3.1290	0.0334
BlendProperty7	1.4408	1.7108	0.1870
BlendProperty9	0.0665	0.1836	1.7600

Takeaway: Absolute errors are small, but for low-magnitude targets (e.g., BlendProperty9), percentage errors inflate, exposing MAPE's sensitivity.

Case 3: Large Errors on Certain Properties

Table 3 Large Errors on Certain Properties

Property	**Actual**	**Predicted**	**MAPE**
BlendProperty3	0.814	0.849	0.0430
BlendProperty5	2.694	2.630	0.0240
BlendProperty7	0.788	0.823	0.0430
BlendProperty9	-0.329	-0.401	0.2190

Highlights: Most predictions are close, but some properties diverge sharply.**Takeaway:** Even when predictions are close in absolute terms, percentage errors (especially on small negative/positive values) can look large.

5.4 Discussion: Strengths and Weaknesses

Strengths

- **Strong Generalization:** The AutoTabPFNRegressor achieves an average cross-validation MSE of 0.0070 with low variance, suggesting robust performance without overfitting.
- **Effective Feature Engineering:** Features such as weighted properties, variance, and range enrich the inputs, enabling the model to capture both linear and interactive effects of blending.

- **Sophisticated Ensembling:** Final predictions use an inverse-MAPE weighted ensemble across folds, giving more influence to better-performing models and improving stability over simple averaging.
- **Hybrid Modeling:** Some targets show extreme MAPEs. Replacing predictions for certain blend properties with alternative models (e.g., Linear Regression) could lower MAPE.

Weaknesses

- **High and Volatile MAPE:** The average MAPE is 0.415, varying widely across folds (0.313–0.613%). Errors are small in absolute terms but large as percentages when actual values are near zero.

6. Conclusion and Future Work

This project demonstrates how machine learning and deep learning can be applied to overcome the complexity of predicting and optimizing sustainable fuel blends. By combining tailored feature engineering with hybrid models, both linear and nonlinear property interactions were effectively captured. The approach delivered high accuracy, with some blend properties achieving errors as low as **0.0005 MAPE**, highlighting the reliability of the predictions. In addition, the optimization framework provided a practical way to design blends around multiple objectives, balancing performance, sustainability, and cost. Altogether, the results show a clear pathway toward faster and more intelligent SAF development.

References

Boehm, R.C., Yang, Z., Bell, D.C., Faulhaber, C., Mayhew, E., Bauder, U., Eckel, G. and Heyne, J.S. (2024). Perspective on Fuel Property Blending Rules for Design and Qualification of Aviation Fuels: A Review. *Energy & Fuels*, 38(18), pp.17128–17145. doi:https://doi.org/10.1021/acs.energyfuels.4c02457.

Chakraborty, R. and Mukhopadhyay, P. (2020). Green Fuel Blending: A Pollution Reduction Approach. *Encyclopedia of Renewable and Sustainable Materials*, pp.487–500. doi:https://doi.org/10.1016/b978-0-12-803581-8.11019-7.

de Myttenaere, A., Golden, B., Le Grand, B. and Rossi, F. (2016). Mean Absolute Percentage Error for regression models. *Neurocomputing*, [online] 192, pp.38–48. doi:https://doi.org/10.1016/j.neucom.2015.12.114.

Gonzalez, S.C., Kroyan, Y., Teemu Sarjovaara, Ulla Kiiski, Karvo, A., Toldy, A.I., Martti Larmi and Annukka Santasalo-Aarnio (2021). Prediction of Gasoline Blend Ignition Characteristics Using Machine Learning Models. *Energy & Fuels*, 35(11), pp.9332–9340. doi:https://doi.org/10.1021/acs.energyfuels.1c00749.

Hodson, T., Over, T. and Foks, S. (2021). Mean Squared Error, Deconstructed. *Journal of Advances in Modeling Earth Systems*, 13. doi:https://doi.org/10.1029/2021MS002681.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S.B., Schirrmeister, R.T. and Hutter, F. (2025). Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045), pp.319–326. doi:https://doi.org/10.1038/s41586-024-08328-6.

König, A., Marquardt, W., Mitsos, A., Viell, J. and Dahmen, M. (2020). Integrated design of renewable fuels and their production processes: recent advances and challenges. *Current Opinion in Chemical Engineering*, 27, pp.45–50. doi:https://doi.org/10.1016/j.coche.2019.11.001.

Pahl, R.H. and McNally, M.J. (1990). Fuel Blending and Analysis for the Auto/Oil Air Quality Improvement Research Program. *SAE technical papers on CD-ROM/SAE technical paper series*. doi:https://doi.org/10.4271/902098.