**Detection of Vehicle System Failures Using Sensor Data**

Guga Gugaratshan, Roghayeh Hazratirad, Shubin Luan

# Introduction

The reliable operation of a vehicle hinges on the seamless performance of its components, necessitating a mechanism for early detection of potential failures to minimize maintenance costs and ensure safety. Traditional vehicle alert systems, which trigger warnings based on predefined thresholds, often fail to identify issues until the system's performance has significantly deteriorated. This reactive approach to maintenance not only increases the risk of component failure but also leads to higher repair costs and downtime.

Recognizing the limitations of conventional methods, this project explores the application of machine learning (ML) techniques to revolutionize the way vehicle health is assessed. By leveraging data from sensors integrated within a vehicle, specifically a diesel truck, this project aims to identify early signs of system failures before they become critical. The project's motivation stems from the significant impact that early detection of such failures can have on reducing maintenance expenses, improving vehicle safety, and extending the lifespan of vehicle components.

Employing a combination of supervised and unsupervised learning methods, the project introduces novel approaches to analyze and interpret sensor data, enabling the prediction of potential system failures with greater accuracy than traditional engineering methods. The methodology encompasses outlier detection among other techniques to flag deviations in component performance, indicative of potential issues.

The main findings from this study reveal that supervised learning methods are adept at modeling vehicle performance and identifying outliers, while the unsupervised learning approach, specifically utilizing the Mahalanobis distance, effectively detects anomalies closely aligned with actual injector failure events.

In the following sections, we will delve deeper into the problem statement, discuss the impact and motivation behind choosing this project, outline the methodology employed, and present a detailed analysis of our main findings, underscoring the efficacy of machine learning techniques in early detection of vehicle component failures.

# Existing Project Comparison

The data to be utilized by the team has not been employed in any public projects, and no publicly available codes are associated with this specific dataset. Nonetheless, similar projects addressing predictive maintenance can be found in the public domain, attesting to the widespread interest in this sought-after subject. The researchers (Kim, et al., 2020), introduce an unsupervised anomaly detection method using sensor streams from a marine engine to identify anomalous system behavior that may indicate potential system failure. Another paper (Hulbert, 2022) focuses on a method utilizing short-term memory (LSTM) to predict faults before they occur using data from the vehicle's Controller Area Network (CAN). The approach includes pre-processing the vehicle data for dimensionality reduction and employing a confusion matrix to measure accuracy. There is a Kaggle project (Meraki) where the notebook applies the Principal Component Analysis (PCA) and Isolation Forest model for data analysis and anomaly detection. It demonstrates how PCA can be used to reduce the dimensionality of the dataset. It then employs the Isolation Forest algorithm to identify outliers in the data.

Our project leverages a dataset acquired from a diesel truck that has not previously been utilized in public projects. Notably, the dataset is extensive, containing approximately 150 features. This distinctiveness sets this project apart from others within the same domain. Regarding the methodology, the team plans to employ a combination of supervised and unsupervised learning methods and conduct a comparative analysis of their results to determine their applicability to predictive maintenance. In contrast, the cited references primarily focus on specific techniques. Furthermore, this dataset's available sensors and system data types significantly differ from those used in existing projects. As a result, this project exhibits a high degree of uniqueness while remaining relevant to trucks closely resembling the Inline 6-cylinder, 4-stroke-cycle configuration.

# Datasets

The data was sourced from a diesel truck and provided by Hottinger Bruel & Kjaer Solutions LLC for the express purposes of academic research and methodological development. A member of our team has received authorization to employ this data in a collaborative effort with the University of Michigan for the Milestone II project. While there is no direct public link to the dataset, it will be made accessible to the project team via GitHub (Gugaratshan, 2024). This comprehensive dataset contains approximately 1.7 million entries across roughly 173 distinct features. The primary goal of this project is to monitor vehicle system performance and predict potential anomalies that may indicate system failures.

It is crucial to emphasize that this project is self-contained; it solely depends on the provided dataset for all research and analysis without recourse to any external data sources.

The sensor measurements utilized in this study conform to two pivotal industry standards, ensuring both compatibility and comprehensive coverage of vehicle diagnostics and communications. The primary standard, the Society of Automotive Engineers (SAE) J1939, serves as a vehicle bus protocol widely used for the exchange of diagnostic information and commands across vehicle components. Originally developed for the automotive and heavy-duty truck industries within the United States, the J1939 standard has since found widespread application on a global scale. Complementing this, the J1587 protocol, primarily employed within heavy-duty vehicles, facilitates the exchange of data between network nodes, driver information systems, and diagnostic tools. The integration of both the SAE J1939 and J1587 standards into this research allows for a nuanced capture of system behavior, and unless specified in the feature, all of them are J1939 standard.

# Data Selection and Feature Engineering

The dataset selected for this investigation encompasses data from the years 2017 and 2018, a period during which the subject vehicle experienced a fuel system failure, subsequently repaired in October 2018 (Refer Figure 1). This temporal frame provides a unique opportunity to identify potential precursors to the failure, thereby offering insights into the conditions leading to such system malfunctions thus providing a chance to identify outlier for this project. For this project, the team selected the 2017 data for training and 2018 data for testing, and anomaly detection effort. This is avoid training the models long before the recorded failure event happens.

A graph of data in blue and green

Description automatically generated

Figure 1. Training and testing data, distribution, fuel system failure event

In the preliminary stage of our analysis, a meticulous data preprocessing routine was employed to refine the dataset for these, with the objective of augmenting data quality and securing robust analytical results. This routine comprised several crucial steps, each tailored to mitigate specific concerns related to data integrity and utility:

1. **Data Selection**: The initial phase, the team focused on the domain knowledge and identified key features that have close association with the engine, fuel, transmission system. From 173 features, many features were irrelevant for these system. After completing the data preprocessing and cleaning steps, we observed the following: the training data for 2017 comprised 182,451 entries across 24 features, whereas the 2018 dataset contained 123,511 entries, also across 24 features. Contrary to the expectation that a larger dataset is essential for effective model training, our findings indicate that a substantial amount of data is not necessary to achieve high-quality model performance. In fact, we determined that approximately 10,000 data points were sufficient for training a robust model.
2. **NaN Value Threshold Filtering**: To uphold data integrity, a filtering strategy was implemented to exclude columns with a significant proportion of missing values (NaNs). Columns with less than 90% non-NaN values were omitted from the dataset. This threshold-based approach guarantees the comprehensiveness of the remaining data, reducing the potential bias introduced by missing information.
3. **Missing Data Analysis**: Following column filtering, an evaluation of the residual data for missing values was conducted, determining the percentage of NaNs within each column. This analysis offered critical insights into the dataset's integrity post-filtering, directing subsequent cleaning measures and shaping our strategy regarding further data imputation or exclusion.
4. **Row Exclusion Based on NaN Presence**: In a final cleaning step, rows containing any NaN values were excluded from the dataset to preserve its overall integrity. This measure ensures the reliability of the dataset, as incomplete data could distort analytical outcomes and contribute to erroneous interpretations. A deliberate decision was made against data imputation to avoid introducing non-representative conditions of vehicle operation into the analysis.

This thorough cleaning process has markedly enhanced the dataset's quality, rendering it more conducive to the ensuing phases of our analysis. Through the systematic resolution of issues pertaining to missing data, we have fortified the dataset's reliability and robustness, laying a solid groundwork for precise and insightful machine learning investigations.

To effectively analyze the performance of a diesel truck, especially during driving conditions, it's crucial to filter out idle or non-driving data from the dataset. This preprocessing step is aimed at isolating instances when the vehicle is actively engaged in driving to ensure that the data reflects true vehicle performance and is relevant for machine learning model training. The criteria for preprocessing are based on specific vehicle parameter thresholds that indicate active driving conditions:

1. **Vehicle Speed (VehSpeedEng):** Data is selected where the vehicle speed is greater than 0 but less than 60 mph. This range ensures that the vehicle is moving but not exceeding typical speed limits for urban or controlled environments. It excludes idle conditions and extremely high-speed scenarios which might not be common during normal operations.
2. **Engine Speed (EngSpeed):** The engine speed must be greater than 700 RPM. This condition filters out data when the engine is running at very low speeds or idling, highlighting periods when the engine is under load and providing meaningful insights into its performance.
3. **Fuel Rate (FuelRate):** A minimum threshold of 0.5 gallons/hour ensures that only data points where the engine consumes fuel at a significant rate are considered. This helps in focusing on moments of active combustion and energy usage, which are critical for capturing performance through the machine learning model
4. **Accelerator Pedal Position (AccelPedalPos):** This condition filters the data to only include instances where the accelerator pedal is engaged (position greater than 0). It signifies active driver input, distinguishing between idle periods and active driving.

By applying these criteria, the dataset is refined to include only those data points that reflect the truck's operational conditions under active use. This preprocessing step is essential for building a machine learning model aimed at detecting abnormal performance, as it ensures the model is trained on relevant, meaningful data that accurately represents the dynamics of diesel truck operations. This approach leverages fundamental engineering principles related to vehicle dynamics, engine operation, and fuel consumption to identify key performance indicators that are most affected by and reflective of the vehicle's condition during active driving scenarios.

Table 1. Features selected for the supervised and unsupervised learning

The Table 1 provides the list of features (channels) that are used in the machine learning project after performing initial exploratory data analysis.

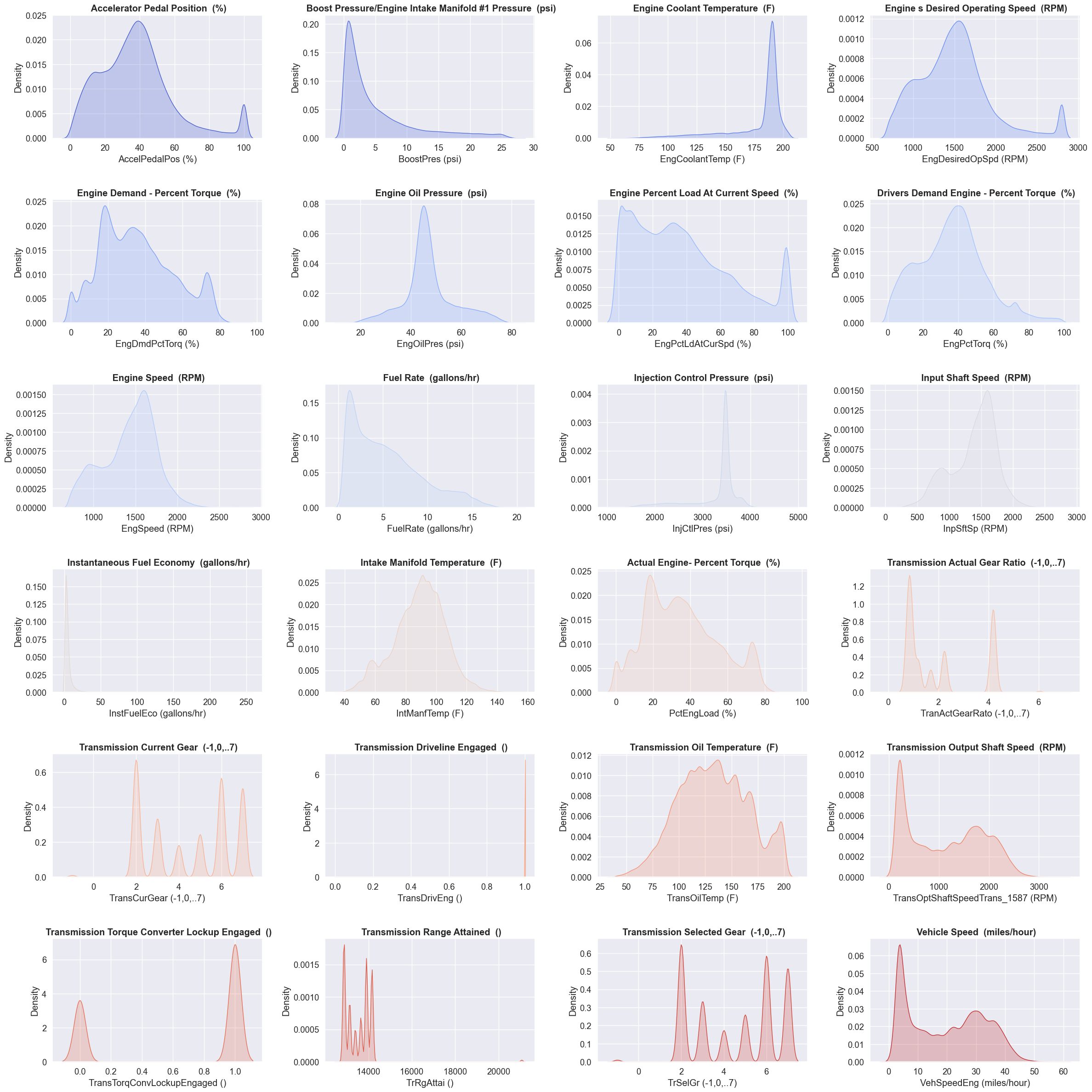


Figure 2. Features Selected for the model training

Figure 2 show the histogram of the features which reveals distinct operational behaviors. The accelerator pedal position tends to have two preferred states at 20% and 75% as indicated by its bimodal distribution. Similarly, the engine torque demand also shows a bimodal pattern with peaks at 20% and 80%. This suggests certain efficiency points in vehicle operation. Boost pressure and fuel rate parameters display skewed distributions, with most occurrences at the lower end of the scale, peaking at around 5 psi and 3 gallons per hour respectively, which implies that lower values are prevalent under normal driving conditions. The engine coolant temperature and the engine's desired operating speed predominantly peak at 190°F and 2100 RPM respectively, showing unimodal distributions which point to a common operating range. Engine oil pressure, intake manifold temperature, and input shaft speed exhibit normal distributions, suggesting stable operation around a central tendency. In contrast, transmission-related parameters such as actual gear ratio, current gear, and selected gear show multimodal distributions, indicating the selective use of certain gears. The driveline engagement and torque converter lockup are either engaged or disengaged with equal frequency, as shown by their bimodal distributions. The vehicle speed histogram is right-skewed, with most measurements clustering at the lower end, indicating a tendency for lower driving speeds. Overall, these observations provide insights into this vehicle performance

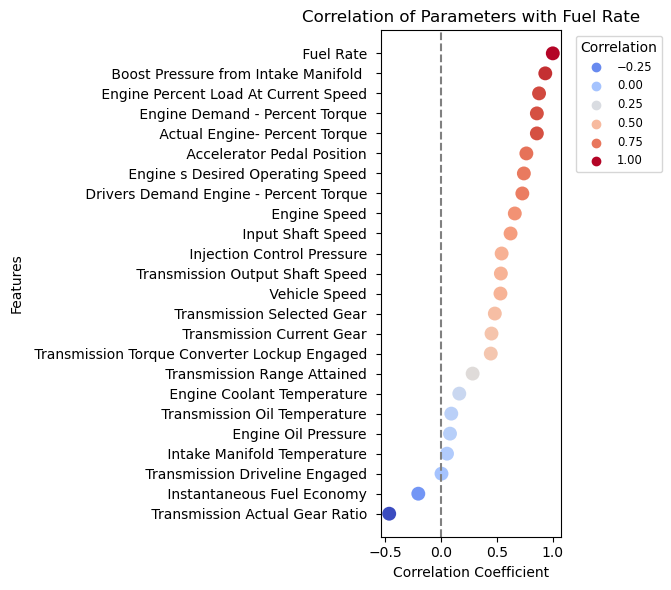
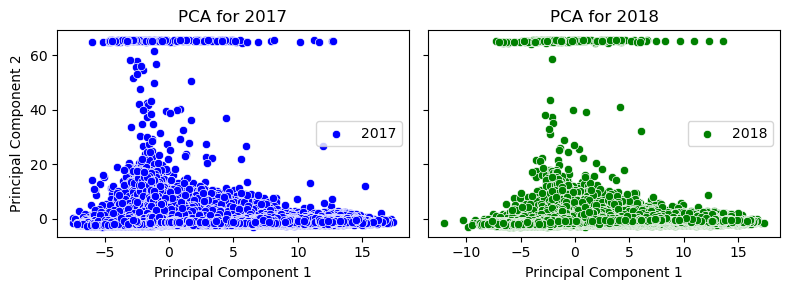


Figure 3. Correlation with the Fuel Rate

The correlation analysis plot shown in Figure 3, the majority of vehicle parameters show a positive correlation with fuel rate, indicating that as these parameters increase, fuel consumption tends to rise correspondingly. Notably, boost pressure, engine load, accelerator pedal position, and engine speed are among the parameters with the strongest positive correlations, suggesting a direct and significant impact on fuel rate. Conversely, the transmission actual gear ratio displays a negative correlation, aligning with the understanding that higher gears can be more fuel-efficient. Parameters with neutral to low positive correlations suggest a more nuanced relationship with fuel rate, potentially affected by various driving conditions and other operational factors. Overall, the data reveals a skew towards positive correlations, emphasizing the influence of most engine and transmission feature parameters on fuel consumption.

* + - * 1. Feature Extraction



A graph with a red line

Description automatically generated

Figure 4. Feature extraction (Principal Component) of 2017 and 2018 Data

Principal Component Analysis (PCA) applied to 2017 and 2018 datasets revealed diverse dispersion patterns across the first two principal components, signaling distinct annual variances. As shown in Figure 4, Visual examination through scatter plots confirmed the disparity in data distribution between the two years. Quantitatively, the first principal component dominated the variance capture, but to exceed 90% variance encapsulation, inclusion of up to five components was necessary—a requirement substantiated by the cumulative explained variance plateauing at the fifth principal component. Although PCA effectively reduced dimensionality, the retention of over 90% of variance necessitated five components, indicating that subtler data nuances essential for an effective machine learning model were not wholly captured. Consequently, the team decided to proceed with the original, untransformed data for subsequent analyses.

# Supervised Learning

In supervised learning, the team developed and evaluated predictive models for estimating fuel rates based on historical data gathered in 2017 and 2018. Year 2017 data was used for training the model. Utilizing Python libraries such as NumPy, pandas, and scikit-learn, the analysis began with the preparation of datasets, focusing on selected features while excluding the 'FuelRate' column as the target variable. A RobustScaler was applied to standardize the feature values, enhancing the models' robustness against outliers.

Three predictive models—Polynomial Regression, Random Forest regressor, and MLP Regressor—were considered for comparison. Additionally, Polynomial Features transformation was employed to explore non-linear relationships, with a custom function facilitating the evaluation of polynomial degrees against the Linear Regression model. Cross-validation scores were computed to assess model performance, employing R-squared as the scoring metric for its interpretability in explaining variance.

* + - * 1. Hyperparameter Tuning

Hyperparameter tuning is a critical step in optimizing machine learning models to enhance their performance. Hyperparameter tuning was applied to all three models using Python's scikit-learn library. The dataset used was from year 2017 data .

For Polynomial Regression, the hyperparameter tuning focused on identifying the optimal polynomial degree. The process involved evaluating the model across degrees 1 through 5 using cross-validation and the R² scoring metric. The results indicated that a first-degree polynomial achieved the highest R² score, suggesting that a linear relationship was most appropriate for the data.

The Random Forest Regressor tuning utilized a GridSearchCV approach, systematically iterating over a predefined grid of hyperparameters, including the number of estimators (trees), maximum depth of the trees, minimum samples required to split a node, and minimum samples required at a leaf node. The optimal parameters resulted in a model with no maximum depth, one sample required at each leaf node, a minimum of two samples to split a node, and 100 estimators, achieving a impressive R² score.

Finally, the Multi-Layer Perceptron (MLP) Regressor tuning also employed GridSearchCV to explore various network architectures, activation functions, solvers, regularization terms (alpha), and learning rate schedules. The best-performing MLP configuration used the 'tanh' activation function, an alpha value of 0.0001, a two-layer architecture with 50 neurons each, a constant learning rate, and the 'adam' solver. This model yielded the highest R² score among the three, indicating exceptional predictive power.

Overall, the hyperparameter tuning techniques applied in this exercise were instrumental in selecting the most suitable parameters for each model, thereby improving their predictive accuracy. The systematic approach and rigorous evaluation provide a robust framework for model optimization in a machine learning workflow.

* + - * 1. Sensitivity Analysis

A close-up of a graph

Description automatically generated

Figure 5. Sensitivity analysis on the Random Forest hyper parameters

A comprehensive sensitivity analysis was conducted on a Random Forest model to evaluate the impact of two key hyperparameters: the number of estimators (n\_estimators) and the maximum depth of the trees (max\_depth). The analysis utilized two performance metrics: the R² score and the model accuracy (1 - MAPE). Surface plots were generated to visualize the dependency of model performance on the hyperparameter values.   The results are presented in two 3D surface plots. The first plot correlates the R² score with varying n\_estimators and max\_depth, revealing a positive relationship between these parameters and the R² score. A higher number of estimators and greater tree depth correspond to improved R² scores, suggesting a better fit of the model to the data.

The second plot examines the model accuracy, defined as 1 minus the Mean Absolute Percentage Error (MAPE). Similar to the R² score, a positive correlation is observed where increases in both n\_estimators and max\_depth are associated with higher accuracy levels.  The sensitivity analysis indicates that for this dataset, a Random Forest model's predictive power improves with an increase in the number of estimators and the depth of the trees. This trend is consistent across both evaluation metrics used. These insights can be leveraged for hyperparameter tuning to enhance model performance.

A graph with a line

Description automatically generated

Figure 6. Sensitivity analysis on the MLP hyper parameters

The sensitivity analysis of the Multilayer Perceptron Regressor indicated that model accuracy, inversely related to the Mean Absolute Percentage Error (MAPE), was largely unaffected by variations in hidden layer sizes within the tested range, suggesting a plateau in performance gains from this parameter. However, the regularization strength (alpha) and initial learning rate demonstrated a more pronounced effect on performance, with optimal values yielding the highest accuracy, indicating that these parameters are crucial for tuning. Future work should focus on a broader and more interactive hyperparameter search, including advanced optimization techniques like Bayesian optimization, to potentially uncover more effective model configurations and improve the predictiveness of the model.

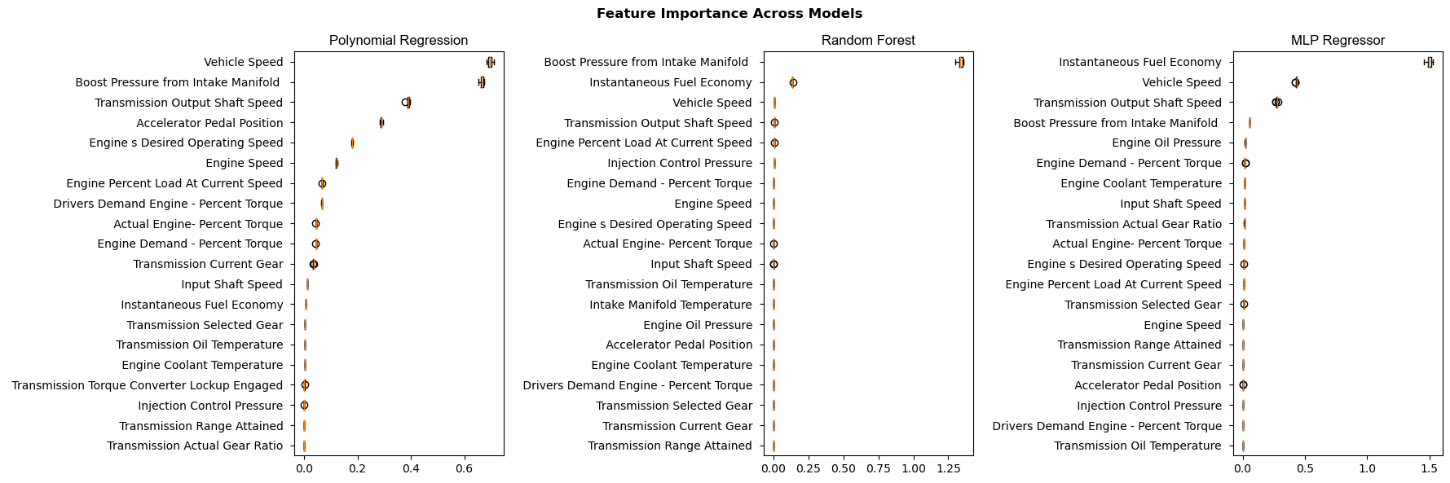
* + - * 1. Feature Importance

Figure 7 illustrates the feature importance across three distinct machine learning models: Polynomial Regression, Random Forest, and MLP (Multi-Layer Perceptron) Regressor. The feature importance scales vary across the models, with Polynomial Regression and Random Forest ranging from 0 to approximately 1.25, and the MLP Regressor extending up to 1.5. Standardized scaling was not employed, as it could obscure the nuanced differences between features' contributions to the models. The Random Forest model demonstrates a relatively even distribution of feature importance, identifying 'Boost Pressure from Intake Manifold' as the most influential feature. Conversely, the Polynomial Regression model exhibits a marked decrease in feature importance beyond the initial features, suggesting a heightened sensitivity to these primary factors.

Figure 7. Feature importance across three machine learning models

Distinctly, the MLP Regressor designates 'Instantaneous Fuel Economy' as its primary feature of importance, which diverges from the findings of the other two models. This variation in the ranking of feature importance across the models implies that each may be detecting different data interactions and correlations.

Notably, 'Engine Percent Load At Current Speed' and 'Transmission Output Shaft Speed' emerge as consistently pivotal features in all three models, underscoring their integral role in the modeling process. Although conducting a robustness assessment through cross-validation could affirm the feature importance's consistency across diverse data subsets, our team has opted to retain all features for subsequent analysis, considering the intricate nature of this engineering challenge.

* + - * 1. Model Evaluation

Table 2. Model evaluation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Training Data | | | | |
| Model | MSE | R2 Score | MAPE | Accuracy |
| Polynomial Regression | 0.67 | 0.96 | 0.18 | 0.82 |
| Random Forest | 0.04 | 0.99 | 0.03 | 0.97 |
| MLP Regressor | 0.1 | 0.99 | 0.05 | 0.95 |
| Testing Data | | | | |
| Model | MSE | R2 Score | MAPE | MAPE |
| Polynomial Regression | 1.33 | 0.92 | 0.22 | 0.78 |
| Random Forest | 0.77 | 0.95 | 0.1 | 0.9 |
| MLP Regressor | 0.27 | 0.98 | 0.07 | 0.93 |

Model evaluation metrics are shown in the Table 2.

**Polynomial Regression:** The model's performance on the training data is quite strong with a high R2 Score of 0.96 and an accuracy of 82%. However, there is a noticeable drop in performance when evaluated on the testing data, with the R2 Score falling to 0.92 and accuracy to 78%. The increase in MSE from 0.67 to 1.33 and in MAPE from 0.18 to 0.22 on the testing data suggests that the model may be overfitting to the training data and not generalizing well to unseen data.

**Random Forest:** Random Forest shows a high level of consistency between training and testing datasets. It maintains the highest R2 Score (.99 on training and 0.95 on testing) and accuracy (97% on training and 90% on testing), along with the lowest MAPE (0.03 on training and 0.10 on testing). The increase in MSE from 0.04 during training to 0.77 during testing indicates some loss in prediction precision, but the model still performs very well on unseen data

**MLP Regressor:** The MLP Regressor model also exhibits high consistency between training and testing. While the R2 Score slightly decreases from 0.99 to 0.98 when moving from training to testing, it still indicates excellent predictive power. The accuracy slightly drops from 95% to 93%, and the MAPE increases from 0.05 to 0.07, which are minor changes considering the complexity of the model. The increase in MSE from 0.10 to 0.27 on testing data is modest and does not significantly detract from the model's strong performance.

The Random Forest and MLP Regressor models show a high degree of robustness and reliability, with only minor performance decreases from training to testing. This suggests that both models have generalized well and are not overfitting the training data. Polynomial Regression, on the other hand, experiences a more significant drop in performance metrics when applied to testing data, indicating potential overfitting issues.

A graph with numbers and symbols

Description automatically generated with medium confidenceA shown in Figure 8, the confusion matrices for the Polynomial Regression, Random Forest, and MLP Regressor models offer insight into their predictive accuracy across different value ranges for the Fuel Rate variable. These matrices serve as both a visual and numerical method to evaluate the performance of the regression models. It is essential to acknowledge, though, that confusion matrices are typically employed in classification problems. In this regression context, we have categorized the continuous output into discrete intervals, using these matrices to approximate and evaluate the models' performance in a more segmented manner.

Figure 8. Confusion Matrix Assessment

In the Polynomial Regression matrix, there's a fair distribution of correct predictions (diagonal elements), although misclassifications are evident, particularly in intermediate value ranges. The Random Forest model's confusion matrix displays a more distinct diagonal, indicating a higher rate of accurate predictions, with some notable confusion at the extreme value ranges. The MLP Regressor's matrix shows an even stronger diagonal, signifying a superior rate of correct predictions, especially for lower values, with relatively fewer inaccuracies compared to the other models. This analysis suggests that the MLP Regressor slightly outperforms the other models in terms of prediction accuracy across different value ranges.

# Team

**Guga Gugaratshan:** Project proposal, data selection, data science methods, evaluation methods

**Roghayeh Hazratirad:** Project planning, project requirements, data preprocessing, visuals, supervised learning Report, Meeting Preparation

**Shubin Luan:** ML pipeline, data engineering, unseupervised learning, model tuning, testing multiple methods, reports.

## References

**Gugaratshan. 2024.** Diesel Truck. [Online] Github, 2024. https://github.com/gugaumich/Diesel\_Truck.git.

**Hulbert, S., Mollan, C., and Pandey, V. 2022.** Fault Diagnosis and Prediction in Automotive Systems with Real-Time Data Using Machine Learning. [Online] SAE Technical Paper 2022-01-0217, 2022. https://doi.org/10.4271/2022-01-0217.

**Kim, D., Lee, S. und Lee, J. An. 2020.** An Ensemble-Based Approach to Anomaly Detection in Marine Engine Sensor Streams for Efficient Condition Monitoring and Analysis. [Online] 2020. https://doi.org/10.3390/s20247285.

**Meraki.** Predictive maintenance for vehicule's engines using advanced anomaly detection. [Online] Kaggle.[Zitat vom: 16. 01 2024.] https://www.kaggle.com/code/orcldsapp0369/notebook2d0a365e05.

Report Requirement

**Introduction**

* **Problem Statement:** Clearly define the problem you're addressing.
* **Impact:** Discuss the significance of solving this problem.
* **Motivation:** Explain why you chose this project.
* **Methodology Overview:** Summarize the methods used in both supervised and unsupervised learning aspects of the project, highlighting any novel approaches.
* **Main Findings:** Present key results from both supervised and unsupervised learning.

**Related Work**

* **Existing Projects/Studies:** Identify at least three related works, providing brief descriptions and how your project differs or improves upon these.

**Data Source and Feature Engineering**

* **Data Description:** Detail the properties of your dataset(s) or API service, including location, formats, important variables, record counts, and time periods covered.
* **Preprocessing:** Describe initial steps taken to clean and prepare the data.
* **Feature Engineering:** Explain the process of transforming raw data into final features used in your models.

**Part A. Supervised Learning**

* **Methods Description:** Outline the supervised learning workflow, methods used, feature representations, and justification for these choices.
* **Supervised Evaluation:** Provide a comprehensive evaluation, including metrics, overall results in a summary table, feature importance analysis, sensitivity analysis, and identification of key trade-offs.

**Part B. Unsupervised Learning**

* **Methods Description:** Describe the unsupervised learning workflow, methods, feature representations, and their justification.
* **Unsupervised Evaluation:** Similar to supervised evaluation, include metrics, summary results, visualizations, and sensitivity analysis.

**Discussion**

* **Learnings and Surprises:** Reflect on insights gained from both parts of the project, challenges encountered, and potential extensions with more resources.
* **Ethical Considerations:** Discuss potential ethical issues related to deploying your machine learning solutions.

**Statement of Work**

* **Team Contributions:** Detail each team member's contributions to ensure clarity and fairness.

**References**

* **Citations:** Include all sources referenced throughout your project to maintain academic integrity.

**Tips for an Effective Report**

1. **Justify Choices:** Beyond describing methods, justify why specific models, features, and evaluation methods were chosen.
2. **Insight into Successes and Failures:** Provide insight into why certain approaches worked or didn't, with a focus on feature analysis and model comparisons.
3. **Highlight Trade-offs:** Discuss potential trade-offs encountered during the project, even if not fully explored.
4. **Ethical Considerations:** Consider the broader implications of deploying your solution, including privacy, fairness, and impact on policy or decision-making.

Following these guidelines will help ensure your report is clear, comprehensive, and reflective of the work put into your project, while also adhering to the course's requirements and academic standards.

|  |  |  |
| --- | --- | --- |
| Section | Requirements | Points |
| Introduction | - Problem being solved - Impact of solving the problem - Motivation for the project - Summary of supervised and unsupervised methods - Main findings for supervised and unsupervised learning | 5 |
| Related Work | - At least three relevant projects or studies - Brief descriptions and comparisons | 5 |
| Data Source | - Location of data/API - Data formats - Important variables - Number of records or API calls - Time periods covered - Initial preprocessing for noisy/missing data | 5 |
| Feature Engineering | - Steps from raw data to final features - Complete list of final features (appendix if needed) | 10 |
| Part A. Supervised Learning | Methods Description: - Supervised learning workflow - Learning methods and feature representations - Justification for method choices - Minimum of three diverse model families - Hyperparameter tuning/exploration description Supervised Evaluation: - Evaluation metrics justification - Summary table comparing models - Feature importance/ablation analysis - Sensitivity analysis - Identification of important tradeoffs Failure Analysis: - Analysis of specific prediction failures - Future improvement suggestions | Methods: 10 Evaluation: 22 Failure: 6 |
| Part B. Unsupervised Learning | Methods Description: - Unsupervised learning workflow - Learning methods and feature representations - Justification for method choices - Minimum of two unsupervised methods - Hyperparameter tuning/exploration description Unsupervised Evaluation: - Evaluation metrics justification - Summary table comparing models - Visualizations summarizing analysis - Sensitivity analysis | Methods: 10 Evaluation: 15 |
| Discussion | - Insights and surprises from Part A and B - Challenges encountered and responses - Extensions with more time/resources | 8 |
| Ethical Considerations | - Potential ethical issues in Part A and B - Solutions to address these issues | 4 |
| Statement of Work | - Description of each team member's contributions | 0 (required) |
| References | - Citations to all resources used | - |
| Report Formatting & General | - Use of Times New Roman or Arial font, 10-12 point size - Black text on white background - Single-spaced, single column - Title page with team members' names - Main body: 10-15 pages (excluding references, including tables and figures) - Appendix for extra material (not counted in page limit) - Self-contained PDF file - Number all tables and figures - Jupyter notebook PDF exports must adhere to formatting rules - SlideDoc format not accepted | - |
| Bonus Points | - Up to 10 bonus points for high-quality, creative, or insightful projects | Up to 10 |