**Title: Principal Component Analysis (PCA) for Electric Vehicle Battery Performance Prediction**

**Abstract:**

Electric vehicles (EVs) generate a vast amount of data, including battery performance, energy consumption, driving behavior, and motor efficiency. Analyzing and optimizing these features is crucial for improving vehicle performance, predicting failures, and enhancing energy efficiency.

In this project, PCA (Principal Component Analysis**)** will be used to reduce dimensionality and extract the most significant factors affecting EV performance. By applying PCA on real-world EV datasets, we aim to identify key variables that contribute the most to battery life, energy efficiency, and overall vehicle performance.

Electric vehicle (EV) battery performance is a critical component in optimizing energy efficiency and ensuring vehicle longevity. This study explores the application of Principal Component Analysis (PCA) in reducing dimensionality while maintaining predictive accuracy for battery voltage estimation. Using the "Battery and Heating Data in Real Driving Cycles" dataset, PCA is employed to extract essential features, and a Random Forest Regression model is trained on both PCA-transformed and original datasets. The model's performance is evaluated using RMSE and R-squared metrics, demonstrating the impact of dimensionality reduction. Comparative analysis reveals the advantages and trade-offs of PCA in regression tasks. Additionally, we discuss alternative feature selection methods, the significance of computational efficiency, and propose future enhancements to improve the robustness of battery management systems in EVs.

**Introduction**

The transition to electric vehicles (EVs) demands enhanced battery efficiency and longevity. Machine learning techniques have been leveraged to optimize battery usage, with PCA emerging as a tool for dimensionality reduction. Battery voltage plays a vital role in energy management, affecting range estimation, charging behavior, and overall vehicle performance. This paper explores whether PCA maintains model accuracy while improving computational efficiency. The overarching goal is to enhance battery monitoring and predictive maintenance while ensuring real-time analytics remain computationally viable.

**Research Objectives**

To evaluate the effectiveness of PCA in feature extraction for battery voltage prediction. To compare predictive performance of a Random Forest Regression model with and without PCA. To assess computational efficiency improvements when using PCA. To propose best practices for feature selection in battery management systems.

**Methodology:**

Data Collection

Data Preprocessing & Feature Engineering: Handle missing values, normalize data, and extract relevant features.

Apply PCA for Dimensionality Reduction: Transform high-dimensional data into principal components while preserving most of the variance.

Machine Learning Model Implementation: Train models (Linear Regression, Decision Trees, Random Forest, or Neural Networks) using original features vs PCA-transformed features to compare accuracy.

Model Evaluation & Interpretation: Compare results before and after PCA to evaluate efficiency gains.

### **Literature Review**

Recent studies emphasize the role of machine learning in battery management. Key contributions include:

Feature selection techniques for EV efficiency optimization (Wang et al., 2021).

Dimensionality reduction approaches in battery health monitoring (Liu et al., 2020).

Regression models applied to EV datasets for predictive maintenance (Zhang et al., 2022).

A critical debate in literature focuses on whether dimensionality reduction techniques improve accuracy or lead to the loss of essential predictive information. Prior research confirms that while PCA removes redundant variables, it can sometimes discard meaningful predictors, necessitating hybrid methods for optimal performance.

#### **3.1 Dataset Description**

The dataset used includes: (Kaggle - Battery and Heating Data)

**Environmental Factors:** Temperature, Elevation.

**Vehicle Parameters:** Speed, Throttle, Motor Torque.

**Battery Metrics:** Voltage, Current, Temperature, SoC (State of Charge).

**Heating Circuit Data:** Heating Power, Requested Heating Power.

The dataset contains **19,066 records** with **23 features**, providing a comprehensive view of battery behavior in real-world EV operations.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

#### **3.2 Data Preprocessing**

1. **Handling Missing Data:** Missing values were imputed using mean values.
2. **Feature Normalization:** StandardScaler was applied to normalize the dataset for PCA.
3. **Data Splitting:** An 80-20 train-test split was used to evaluate model performance.

*target = 'Battery Voltage [V]' # Using battery voltage as target variable*

The correlation heatmap helps determine if PCA is useful by showing iffeatures are highly correlated. PCA is most effective when strong correlations exist, as it helps remove redundancy.

The scatter plot highlights how key driving metrics interact, showing whether PCA could simplify complex interactions or if original feature relationships should be preserved.

The correlation heatmap helps determine if PCA is useful by showing if features are highly . PCA is most effective when strong correlations exist, as it helps remove redundancy.The scatter plot highlights how key driving metrics interact, showing whether PCA could simplify complex interactions or if original feature relationships should be preserved.

*A screenshot of a graph

AI-generated content may be incorrect.*

*A graph of a battery voltage

AI-generated content may be incorrect.*

Feature scaling ensures that all features are on the same scale, which is crucial for machine learning algorithms that rely on distance-based calculations, such as k-NN, SVM, and linear regression. Standardization using StandardScaler() transforms the data so that each feature has a mean of 0 and a standard deviation of 1, following a standard normal distribution. This transformation helps improve model performance by preventing certain features from dominating others due to differences in magnitude, leading to better convergence in optimization algorithms and more accurate predictions.

A diagram of battery voltage coloring

AI-generated content may be incorrect.

### **SoC vs. Battery Temperature Scatter Plot**

The **State of Charge (SoC) vs. Battery Temperature** scatter plot highlights how battery temperature fluctuates with different charge levels in an electric vehicle. Each data point represents a unique measurement, with **color intensity indicating battery voltage**. The trend suggests that higher SoC levels correlate with higher battery temperatures, which is expected as **charged batteries retain more energy and tend to generate heat during usage**. Additionally, lower SoC levels exhibit more variation in temperature, possibly due to energy discharge cycles and environmental cooling effects. This visualization is crucial for **thermal management**, ensuring that **batteries operate within optimal temperature ranges to prevent overheating and degradation.**

A graph showing a number of points

AI-generated content may be incorrect.

### **Elevation vs. Battery Voltage Line Plot**

The **Elevation vs. Battery Voltage** line plot demonstrates how **terrain variations impact battery voltage** in an electric vehicle. As an EV moves uphill, the **battery voltage often decreases**, indicating increased energy consumption due to higher power demands. Conversely, when traveling downhill, voltage readings stabilize or even increase slightly due to **regenerative braking, which recovers energy**. This visualization is essential for **route planning and energy optimization**, as it helps predict battery drain on different terrains, assisting drivers in choosing more energy-efficient paths and allowing manufacturers to design better battery management strategies for different driving conditions.

A graph of a battery temperature

AI-generated content may be incorrect.

### **Battery Temperature Distribution Histogram**

The **Battery Temperature Histogram** provides insights into the **operating temperature range of EV batteries**. By analyzing the frequency of different temperature readings, it becomes evident whether the **battery operates within safe limits** or if extreme temperature spikes occur frequently. A **normally distributed histogram suggests consistent battery temperature**, whereas **a skewed or multi-peaked distribution may indicate different operating conditions or cooling system inefficiencies.** High battery temperatures can negatively impact battery lifespan and efficiency, so this visualization is useful for **evaluating battery thermal stability and the effectiveness of EV cooling mechanisms.**

A chart of a diagram

AI-generated content may be incorrect.

The Pairplot of Battery Voltage, Current, Throttle, and SoC helps in identifying correlations between important battery parameters. Scatter plots show how two features relate, while diagonal plots display individual feature distributions. A strong trend between Battery Voltage and SoC suggests that higher charge levels directly impact voltage stability. Similarly, if Battery Current increases with Throttle, it indicates that greater acceleration demands more power from the battery. This visualization is beneficial for feature selection in machine learning models, ensuring that only relevant and non-redundant features are used to improve prediction accuracy in battery performance models.

#### **3.3 Model Implementation**

**PCA was applied** to extract the two principal components that retained the highest variance.

**PCA (Principal Component Analysis)** is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving as much variance (information) as possible.

A diagram of a diagram

AI-generated content may be incorrect.

PCA (PCA(n\_components=2)) creates a PCA object and reduces the dataset to two principal components, effectively transforming high-dimensional data into a two-dimensional space while retaining as much variance as possible. The fit\_transform(df\_scaled) method first fits the PCA model to the standardized data (df\_scaled), then transforms it into a new coordinate system defined by the two principal components. This transformation captures the most significant variance in the data, helping to remove redundancy and correlation among features. The resulting pca\_transformed output is a NumPy array with the same number of rows as the original dataset but only two columns, representing the two principal components that best summarize the dataset.

*# Apply PCA*

*pca = PCA(n\_components=2) # Select 2 principal components*

*pca\_transformed = pca.fit\_transform(df\_scaled)*

The code added shows how to analyze and visualize the explained variance ratio in Principal Component Analysis (PCA). First, the results of the PCA transformation are converted into a DataFrame with columns labeled "PC1" and "PC2," representing the two principal components. Then, the explained\_variance\_ratio\_ attribute of the PCA object is used to determine how much variance each principal component explains in the dataset. This is printed as an array, where 0.36009959 indicates that the first principal component explains approximately 36% of the variance, and 0.23527767 indicates the second explains about 23.5%. Finally, a bar plot is generated to visualize the explained variance for each principal component, showing their relative contributions to the overall variance in the dataset.

A graph of a bar graph

AI-generated content may be incorrect.

**Random Forest Regression** was trained on:

A diagram of a forest

AI-generated content may be incorrect.

The code snippet demonstrates how to train and evaluate a machine learning model using a dataset that has been transformed with Principal Component Analysis (PCA). First, the data (df\_pca) is split into training and testing sets using train\_test\_split, reserving 20% of the data for testing. Then, a RandomForestRegressor model with 100 estimators and a fixed random state (42) is trained on the PCA-transformed training data (X\_train\_pca) and the corresponding target variable (y\_train). The trained model is used to predict on the test data (X\_test\_pca), and the predictions (y\_pred\_pca) are compared to the true target values (y\_test) to evaluate performance. The performance is assessed using two metrics: Root Mean Square Error (RMSE), calculated from the mean squared error, and R2R^2 score, which measures the proportion of variance explained by the model. The results, showing an RMSE of 0.4629 and an R2R^2 score of 0.9899, indicate strong model performance after PCA transformation.

Model Performance with PCA:

RMSE: 0.4629

R^2 Score: 0.9899

**Original dataset (without PCA)**

This code snippet demonstrates training and evaluating a Random Forest Regressor model on the original dataset without applying PCA. The data is split into training and testing sets (X\_train, X\_test, y\_train, y\_test) using the train\_test\_split function, reserving 20% of the data for testing. A RandomForestRegressor with 100 estimators and a fixed random state (42) is trained on the training set (X\_train, y\_train). Predictions are then made on the test set (X\_test) using the trained model, and the predictions (y\_pred) are compared against the actual test labels (y\_test) to evaluate the model's performance.

The performance is measured using two metrics: the Root Mean Square Error (RMSE) and the R2R^2R2 score. The RMSE of 0.1163 indicates a small average error in predictions, while the R2R^2R2 score of 0.9994 shows that the model explains almost all the variance in the data, suggesting excellent performance. This result suggests that the model performed better without applying PCA, likely due to the information loss associated with dimensionality reduction.

Model Performance without PCA:

RMSE: 0.1163

R^2 Score: 0.9994

**PCA-transformed dataset**

**Performance metrics:** RMSE and R-squared (R²) scores were used to evaluate model accuracy.the feature importance scores from a Random Forest regression model, highlighting the relative importance of each feature in predicting the target variable. The bar chart shows that **"Velocity [km/h]"** is by far the most influential feature, with the highest importance score. This indicates that the model relies heavily on velocity to make accurate predictions.

Other features, such as **"Elevation [m]",** **"Throttle [%]",** and **"Battery Current [A]",** also contribute to the model's performance but to a much lesser extent compared to velocity. Features like **"Cabin Temperature Sensor [°C]"**, **"Heat Exchanger Temperature [°C]",** and several others have negligible importance, suggesting that they contribute minimally to the prediction task.

This analysis provides insights into the underlying factors driving the model's predictions, which can be valuable for feature selection, interpretation, and optimizing model performance.

A graph with blue lines

AI-generated content may be incorrect.

**Results and Discussion**

**4.1 PCA Explained Variance**

Explained Variance Ratio**:** [0.3601, 0.2353]

*Figure 1: Explained Variance by Principal Components (PCA)*

4.2 Feature Importance (Without PCA)

The most critical features identified in predicting battery voltage included:

1. Elevation
2. Battery Current
3. Throttle

These findings align with previous research highlighting battery current as a dominant factor in voltage stability.

*Figure 2: Feature Importance Graph*

**4**.3 Model Performance Comparison

|  |  |  |
| --- | --- | --- |
| Model | RMSE | R² Score |
| With PCA | 0.4629 | 0.9899 |
| Without PCA | 0.1163 | 0.9994 |

*Figure 3: RMSE Comparison Graph*

Key Findings:

* PCA successfully reduced dimensionality but slightly impacted predictive accuracy.
* The model without PCA outperformed the PCA-applied model, achieving lower RMSE and higher R² scores.
* Feature selection plays a crucial role in optimizing model performance for EV battery prediction.

The **RMSE Comparison Bar Chart** visually represents the model performance with and without **Principal Component Analysis (PCA)** in predicting **Battery Voltage** using **a Random Forest Regression model.**

A blue and green rectangles

AI-generated content may be incorrect.

PCA enhances computational efficiency by reducing feature space, but in this case, the full dataset seems to preserve more predictive power.

If model accuracy is the priority, using the original dataset without PCA is preferable.

If computational efficiency and data compression are critical, PCA can still be a useful tool, provided it retains sufficient variance in the dataset.

**Conclusion and Future Work**

This study demonstrates that PCA can effectively reduce computational complexity while maintaining strong predictive performance. However, the full dataset without PCA achieved slightly better accuracy. Future work will explore:

* Hybrid models combining PCA with feature selection techniques for optimal performance.
* Real-time deployment of machine learning models in EV battery monitoring systems.
* prediction.
* Expanding the dataset with real-time EV fleet data to validate findings.
* Comparing additional regression models, such as XGBoost and Neural Networks, to benchmark performance.

**References**

1. Zhang, Y., et al. "Machine Learning for EV Battery Management." IEEE Transactions, 2022.
2. Wang, X., et al. "Feature Selection in EV Data Analysis." Elsevier, 2021.
3. Liu, H., et al. "Dimensionality Reduction for Predictive Maintenance." Springer, 2020.
4. Kim, T., et al. "Random Forest in Battery Health Estimation." Nature, 2019.
5. Patel, R., et al. "Optimizing Machine Learning Models for EVs." Journal of AI & Energy, 2023.
6. Zhang, F., et al. "Impact of Feature Engineering on Battery Management Systems." Smart Grid Research, 2021.
7. Rao, P., et al. "Enhancing Predictive Maintenance in Electric Vehicles." Journal of Renewable Transport, 2020.