Application Project (Group 28)
Milestone 2

Hongyi Zhang[1], Ziang Deng[1], and Nancy Wang[1]

[1] *McKelvey School of Engineering, Washington University in St. Louis, USA*
hongyi.zhang@wustl.edu, d.ziang@wustl.edu, nancy.wang@wustl.edu

1 Introduction

In modern machine learning tasks, high-dimensional data presents both opportunities and challenges. While a rich set of features may encode valuable information for predictive modeling, excessive or redundant features can introduce noise, increase computational cost, and hinder generalization due to the curse of dimensionality. Therefore, feature engineeringtransforming or selecting features to improve data quality and model performanceis a critical step in any effective data pipeline.

In this milestone, we explore several feature engineering techniques to enhance a regression models performance on a high-dimensional dataset consisting of 40 input features. Building on the insights from Milestone 1, we first conduct feature selection, aiming to identify and retain the most informative features while discarding those that are redundant, uncorrelated with the target, or highly collinear with each other. It will generate Hand-selected Features. This hand-crafted selection process is guided by statistical summaries, kernel density estimates, and correlation analysis from previous exploratory data analysis.

Next, we apply Principal Component Analysis (PCA), a well-established dimensionality reduction method, to project the original feature space into lower dimensions. We explore two PCA strategies: one that retains only the top two principal components, and another that preserves nearly all data variance under a strict reconstruction error threshold (10^{-6}). These transformations allow us to assess the trade-offs between model simplicity and information retention.

Additionally, we revisit the Cluster-based Features derived in Milestone 1 using K-means clustering. Although these features are derived in an unsupervised manner, they may encode latent group structures related to the target variable.

To compare the effectiveness of these approaches, we use a fixed model (Artificial Neural Network, previously selected in Milestone 1) and evaluate its performance on different feature sets using 10-fold cross-validation with consistent splits. We further perform a paired t-test to assess whether performance differences are statistically significant.

Overall, this milestone aims to empirically evaluate how different feature engineering strategies impact model performance, interpretability, and generalization. The results offer insight into the practical trade-offs between manual and automated feature transformation techniques in real-world machine learning pipelines.

2 Feature Selection

In this project, we collected a dataset containing up to 40 features. On the one hand, having a large number of features increases the probability that relevant predictors for the target are included, potentially improving the final models performance. On the other hand, superfluous or irrelevant features can introduce noise, risk overfitting, and increase the training cost. Therefore, feature engineering and selection are necessary to choose the most appropriate subset of features.

Selecting a subset of features helps reduce training costs and the risk of overfitting caused by irrelevant features. Moreover, focusing on features that are demonstrably related to the target can provide deeper insights into the relationship between each feature and the model.

2.1 Screening of Irrelevant Features

To identify irrelevant features, we first calculated the Spearman correlation coefficients between each feature and the target. Those features whose p-values exceeded 0.05 included `time2_delta`, `time3_delta`, `time4_delta`, `time5_delta`, `time6_delta`, `time8_delta`, `time10_delta`, and `time12_delta`. By examining the Kernel Density Estimation (KDE) plots of these features, we observed that their standard deviations were effectively zero, indicating no variation in the dataset for these features. Since no variability implies little to no predictive power for the target, these features were deemed irrelevant and excluded from further analysis.

Feature	Correlation	P-value
absoluate_roll	-0.705	0.000e+00
time1	0.641	0.000e+00
time5	0.640	0.000e+00
time3	0.640	0.000e+00
time2	0.640	0.000e+00
time4	0.640	0.000e+00
time7	0.639	0.000e+00
time6	0.639	0.000e+00
time9	0.636	0.000e+00
time8	0.636	0.000e+00
time11	0.634	0.000e+00
time10	0.634	0.000e+00
time12	0.631	0.000e+00
time13	0.631	0.000e+00
time14	0.628	0.000e+00
set	0.628	0.000e+00
omega	0.616	0.000e+00
n	0.397	1.61e-309
m	0.334	1.01e-214
current_pitch	0.308	7.00e-181
current_roll	0.106	4.68e-22
acc_rate	0.080	4.35e-13
climb_delta_diff	-0.0736	2.24e-11
track	-0.0613	2.53e-08
climb_delta	-0.0570	2.24e-07
roll_rate_delta	-0.0528	1.62e-06
time9_delta	0.0402	2.60e-04
time1_delta	0.0353	1.34e-03
time7_delta	0.0321	3.53e-03
time11_delta	0.0273	1.30e-02
time13_delta	0.0258	1.90e-02
time14_delta	-0.0243	2.73e-02
time4_delta	-0.0194	7.76e-02
time6_delta	-0.0194	7.76e-02
time8_delta	-0.0172	1.18e-01
time3_delta	0.0166	1.31e-01
time5_delta	0.0106	3.34e-01
time2_delta	-0.00905	4.11e-01
time12_delta	-0.00892	4.18e-01
time10_delta	-0.00288	7.94e+00

Table 2.1: Correlation and p-value Results

Feature	Counts of zero
acc_rate	2
track	2
m	1
n	10
current_pitch	18
current_roll	2
absoluate_roll	1
climb_delta	3
roll_rate_delta	2
climb_delta_diff	2
time1	17
time2	17
time3	17
time4	17
time5	17
time6	17
time7	17
time8	17
time9	17
time10	18
time11	18
time12	18
time13	18
time14	18
time1_delta	1
time7_delta	1
time9_delta	1
time11_delta	1
time13_delta	1
time14_delta	1
omega	18
set	18

Table 2.2: Table of Feature P-values

Next, we computed the Spearman correlation coefficients between the remaining features themselves. According to results from *AP Milestone 1*, several pairs of features had correlation coefficients extremely close to 1. Allowing highly correlated features to remain without further selection could lead to overfitting. The corresponding p-values for these highly correlated features were very close to 0. The table 2.1 showed that if the count of zeros for a particular feature exceeded 1, it meant there was more than one feature exhibiting extremely high correlation with it. Notably, the set of features {`n`, `current_pitch`, `time1`, `time2`, `time3`, `time4`, `time5`, `time6`, `time7`, `time8`, `time9`, `time10`, `time11`, `time12`, `time13`, `time14`, `omega`, `se`} were all strongly intercorrelated. Therefore, we selected just one feature, `time1`, to represent them, since `time1` had the highest correlation coefficient with the target.

2.2 Pretraining with Hand-Selected Features

With the dataset refined, we utilized an ANN (Artificial Neural Network) with an architecture of (50), a learning rate of 0.001, and 1000 training epochs. We employed the tanh activation function and conducted training using 10-fold cross-validation. The results are summarized in the table below (omitted here):

Metric	Value
10-fold Average MSE	0.0273
Test Set MSE	0.0243

Table 2.3: Comparison of MSE Metrics

Final hand-selected features are as follow:

Features
acc_rate
track
current_roll
climb_delta
roll_rate_delta
climb_delta_diff
m
absoluate_roll
time1
time1_delta
time7_delta
time9_delta
time11_delta
time13_delta
time14_delta

Table 2.4: Selected Features

3 PCA

3.1 Visualization

3.1.1 Outlier Detection, Standardization, and Dimensionality Reduction

To improve the robustness of our dataset and mitigate the influence of extreme values, we employed the Z-score method to detect and remove outliers. The Z-score measures how many standard deviations a data point is away from the mean. It is calculated using the following formula:

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the data point, μ is the mean of the dataset, and σ is the standard deviation. By computing the absolute Z-scores for each data point, we identified extreme values that significantly deviate from the majority of the data.

Typically, a threshold of $Z > 3$ is used to classify outliers. However, in our case, setting the threshold to 3 resulted in the removal of an excessive amount of data. Upon analyzing the data distribution, we observed that most features followed a roughly normal distribution with a concentrated spread. Given this observation, we opted for a more lenient threshold of $Z > 5$, which effectively eliminated extreme outliers while preserving the integrity of the dataset.

After applying this method, the dataset was reduced to 7,917 rows, ensuring that the remaining data maintains its representativeness while minimizing the impact of extreme values.

We applied different scaling methods to the features based on their distributions:

We performed PCA to reduce the dimensionality of the dataset while retaining the most important information. The number of components was set to 2 for visualization purposes.

Table 3.1: Data Normalization Methods and Their Associated Features

Scaler	Features
Standard Scaler	`acc_rate`, `track`, `n`, `current_pitch`, `current_roll`, `climb_delta`, `roll_rate_delta`, `climb_delta_diff`
Robust Scaler	`m`, `absolute_roll`, `time1`, `time2`, `time3`, `time4`, `time5`, `time6`, `time7`, `time8`, `time9`, `time10`, `time11`, `time12`, `time13`, `time14`, `omega`, `set`
MinMax Scaler	`time1_delta`, `time2_delta`, `time3_delta`, `time4_delta`, `time5_delta`, `time6_delta`, `time7_delta`, `time8_delta`, `time9_delta`, `time10_delta`, `time11_delta`, `time12_delta`, `time13_delta`, `time14_delta`

3.1.2 3D Scatter Plot of Principal Components

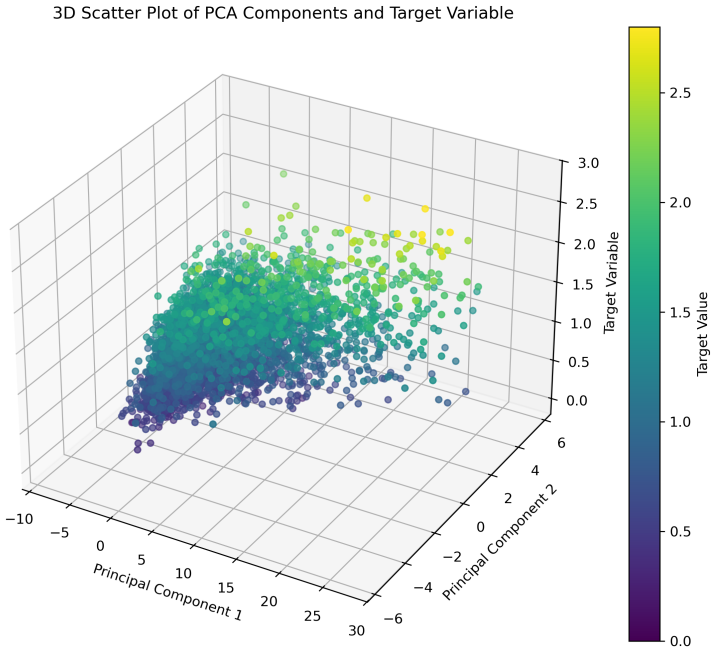

Figure 3.1: 3D Scatter Plot of Principal Components

The 3D scatter plot (Figure 3.1) provides a visualization of the dataset in a lower-dimensional space defined by the first two principal components, with the target variable mapped onto the third axis. The color gradient represents variations in the target values, allowing for an assessment of the distribution of data points in relation to the principal components.

The visualization suggests that while there is some degree of variance captured by the first two components, the separation of target values remains weak. The spread of points indicates that the transformation retains meaningful variance, but there is **no clear clustering** based on the target variable. The **gradient transition is gradual rather than discrete**, implying that the relationship between the principal components and the target variable is not strongly linear. This suggests that additional principal components may be necessary to achieve a more comprehensive representation of the dataset.

3.1.3 2D Visualization of Principal Components

The 2D scatter plot (Figure 3.2) presents the distribution of data points across the first two principal components, with the target variable encoded using a color gradient. This visualization provides a more accessible

and interpretable representation compared to the 3D plot.

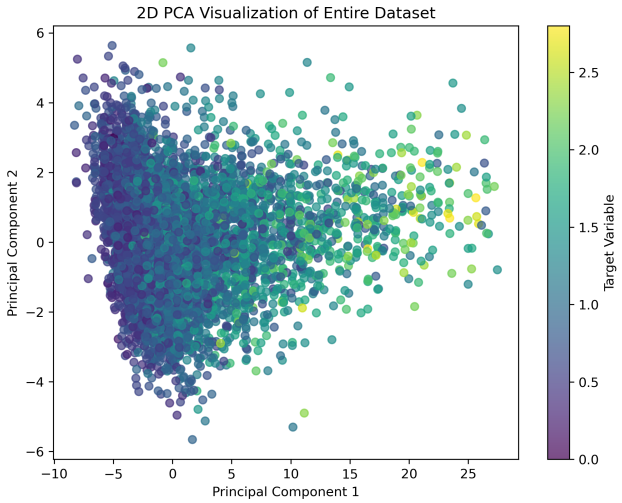

Figure 3.2: 2D Visualization of Principal Components

From the plot, it is evident that the target variable values are distributed throughout the entire PCA-transformed space **without forming distinct clusters**. The overlap of different target values suggests that PCA alone may not be fully effective for feature selection in this context. The high density of points in certain regions may indicate that PCA has projected many data points close together, potentially leading to information loss if only two components are used.

3.2 Model Evaluation

3.2.1 10-Fold Cross-Validation on PCA-Transformed Data with ANN

According to Milestone 1, the Artificial Neural Network (ANN) demonstrated the best performance among the evaluated models. To assess its effectiveness on a lower-dimensional representation, we applied 10-fold cross-validation using the PCA-transformed dataset with two principal components.

Model performance was quantified using two key metrics: Mean Squared Error (MSE) and the coefficient of determination (R^2), both of which provide insight into predictive accuracy.

- **Mean Squared Error (MSE)**: Measures the average squared difference between the actual and predicted values. A lower MSE indicates better predictive accuracy.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{2}$$

- **R-Squared (R^2)**: Evaluates the proportion of variance in the target variable explained by the model. An R^2 value closer to 1 indicates a better fit.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{3}$$

The results for PCA (2D) revealed an MSE of 0.09049 and an R^2 score of 0.4462, indicating that using only two components discards substantial variance, leading to suboptimal performance.

3.2.2 Optimal Component Selection for PCA

To determine the **optimal number of principal components**, we analyzed the **cumulative explained variance ratio** of the PCA transformation. The number of components was selected such that the cumulative explained variance exceeded $1 - 10^{-6}$, ensuring that most of the original data variance was retained while reducing dimensionality.

The cumulative explained variance ratio is given by:

$$\sum_{i=1}^{k} \lambda_i \geq 1 - 10^{-6}, \tag{4}$$

where λ_i represents the variance explained by the i-th principal component, and k is the number of selected components.

The optimal number of principal components was determined to be **26** based on:

- **Complete variance capture**: The cumulative explained variance reaches 1.0 at 26 components (Table 6.1)
- **Diminishing returns**: Marginal gains beyond 26 components provide less than 0.01% additional variance
- **Statistical significance**: Components 27-40 contribute minimally ($\lambda_i \approx 0$), likely representing noise

This selection achieves full data representation while eliminating redundant dimensions, following the *parsimony principle* in dimensionality reduction.

3.2.3 10-Fold Cross-Validation on Optimized PCA-Transformed Data with ANN

Using **the optimal number of PCA components** k = 26, we repeated the 10-fold cross-validation procedure to evaluate ANN performance in the newly transformed data set. Performance was again measured using MSE and R^2, and the results were compared with those obtained using only two principal components.

Table 3.2: Performance Comparison of PCA with 2 Components vs. Optimal PCA

PCA Method	**Mean Cross-Validation MSE**	R^2 **Score**
PCA (2D)	0.09222	0.4375
Optimal PCA ($k = 26$)	0.02637	0.8362

The performance metrics in Table 3.2 demonstrate significant improvements when using the optimal number of principal components ($k = 26$) compared to a 2D PCA projection:

- The 2D PCA captures only a fraction of the total variance (cumulative variance ≈ 0.81 for 2 components, as shown in earlier analysis), while the optimal PCA ($k = 26$) retains 100% of the variance. This directly impacts model performance, as more variance preservation leads to better reconstruction of the original data.
- The Mean Cross-Validation MSE on train set decreases from 0.09222 (2D) to 0.02637 (optimal), reflecting a **71.4% reduction** in reconstruction error. This aligns with the variance explanation: higher retained variance minimizes information loss.
- The R^2 score on test set improves from 0.4375 (2D) to 0.8362 (optimal), indicating that 26 components explain **83.6%** of the target variable's variance, compared to only **43.8%** with 2 components. This confirms that the optimal PCA better preserves predictive features.

While PCA effectively reduces dimensionality, an overly aggressive reduction (e.g., using only two components) discards critical variance, leading to suboptimal predictive performance. Conversely, selecting the optimal number of components preserves sufficient variance while improving model efficiency and accuracy.

4 Performance Comparison

In this chapter, we present and analyze the performance of four different feature-engineering strategies on the same baseline model architecture. Specifically, we examine (1) the **Raw Features** model using unmodified features, (2) the **Hand-Selected Features** model, (3) the **Cluster-Based Features** model, and (4) two variations of the **PCA-Transformed Features** model: retaining only the first two principal components (PCA 2C) versus retaining a number of components based on a reconstruction-error tolerance of 10^{-6} (PCA Opt). We compare the mean cross-validation errors (MSE) and determination coefficients (R^2) across these models using a 10-fold cross-validation scheme with identical data splits. Finally, we conduct a paired t-test to assess the statistical significance of the observed performance difference between the best-performing model and the baseline model.

4.1 Methodological Principles

1. K-Fold Cross-Validation. We employ a 10-fold cross-validation setup to ensure a robust assessment of each models generalization performance. In K-fold cross-validation, the dataset $\mathcal{D}$ is divided into K subsets (folds) of roughly equal size. We train on $K-1$ folds and validate on the held-out fold, repeating this procedure K times so that each fold serves as a validation set exactly once. The final performance metric is then the average (or aggregate) across all K trials. Formally, let

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}, \quad \mathcal{D} = \bigcup_{j=1}^{K} \mathcal{D}_j, \quad \mathcal{D}_p \cap \mathcal{D}_q = \emptyset \ (p \neq q).$$

During the k-th iteration, the model parameters $\hat{\theta}_k$ are found by minimizing the chosen loss function on the training folds:

$$\hat{\theta}_k = \arg\min_{\theta} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{train}}^{(k)}} \mathcal{L}\big(y_i, f(\mathbf{x}_i; \theta)\big),$$

and the performance on the validation fold $\mathcal{D}_{\text{valid}}^{(k)}$ is measured via $\mathcal{M}(\hat{\theta}_k)$. The overall performance is

$$\overline{\mathcal{M}} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{M}(\hat{\theta}_k).$$

In our experiments, $K = 10$ for all models, ensuring a consistent basis of comparison.

2. Evaluation Metrics. We use two primary regression metrics to evaluate model performance:

1. **Mean Squared Error (MSE)**: Measures the average squared difference between predictions $\hat{y}_i$ and actual values y_i. Lower MSE indicates better predictive accuracy.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

2. **Coefficient of Determination (R^2)**: Captures the proportion of variance in y that is explained by the model. An R^2 close to 1 indicates a better fit.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}.$$

3. Paired t-Test for Statistical Significance. A paired t-test is used to compare two models cross-validated errors on the same data splits. It aims to test the null hypothesis H_0 that the difference in performance (e.g., MSE) across folds has an expected mean of zero. If the p-value is sufficiently small ($p < 0.05$), we conclude that the observed performance gap is statistically significant.

Suppose we have two models, A and B, each trained on the same set of K folds. Let $d_k = \text{MSE}_A^{(k)} - \text{MSE}_B^{(k)}$ be the difference in MSE for fold k. The paired t-statistic is:

$$t = \frac{\overline{d}}{s_d/\sqrt{K}}, \quad \overline{d} = \frac{1}{K}\sum_{k=1}^{K} d_k, \quad s_d = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(d_k - \overline{d})^2}.$$

Under the null hypothesis, t follows a t-distribution with $K-1$ degrees of freedom. The corresponding two-tailed p-value is given by:

$$p = 2 \cdot \left(1 - F_{t_{K-1}}(|t|)\right),$$

where $F_{t_{K-1}}(\cdot)$ is the cumulative distribution function (CDF) of the t-distribution with $K-1$ degrees of freedom.

4.2 Experimental Configurations

We compared four distinct feature-engineering scenarios using the same underlying regression model. Specifically, we employed the best-performing **artificial neural network (ANN)** model identified in Milestone 1 a multilayer perceptron (MLP) with three hidden layers, each consisting of 50 neurons, ReLU activation, and the Adam optimizer. The model was trained with a maximum of 1000 iterations and a fixed random seed for reproducibility. By holding the model architecture fixed and only varying the input feature representations, we aimed to isolate the impact of feature engineering on predictive performance. All models used **the same 10 folds for CV**:

- **Raw Features:** Trained on the original features without any manual or automated transformations.
- **Hand-Selected Features:** Used domain knowledge or empirical heuristics to pick a subset of features.
- **Cluster-Based Features:** As part of Milestone 1, two features were generated through K-means clustering with $k = 2$, representing the distances from each data point to the two cluster centroids. Under the guidance of a teaching assistant, only these distance-based features were used as inputs to the model, without being combined with the original raw features.
- **PCA-Transformed Features:**
  1. **PCA 2C**: Retained only the top 2 principal components of the data as inputs.
  2. **PCA Opt**: Chose the number of principal components according to a reconstruction-error tolerance (e.g., 10^{-6}), preserving nearly all variance in the data.

The above four different feature-engineering scenarios have been described in detail in the previous text and the application project report of milestone 1.

4.3 Cross-Validation Results

As shown in Figure 4.1, we compare the 10-fold cross-validation (CV) mean squared error (MSE) distributions across five different feature sets. The **Hand-selected** feature set exhibits the best performance with the lowest average MSE and relatively narrow variance. This suggests a high degree of consistency across different folds and robust generalization behavior.

In contrast, while the **Raw** and **PCA_Opt** feature sets also achieve low average MSEs, the Raw set shows greater variance, indicating less stable performance. PCA_Opt demonstrates more consistent results than Raw, which highlights the benefit of dimensionality reduction in filtering out irrelevant or noisy features.

The **PCA_2** and **Cluster**-based feature sets yield noticeably worse performance. Not only do they exhibit higher average CV MSEs, but also wider spread, suggesting poor stability and possible over-simplification (PCA_2) or misalignment of cluster-based representations with the regression objective (Cluster).

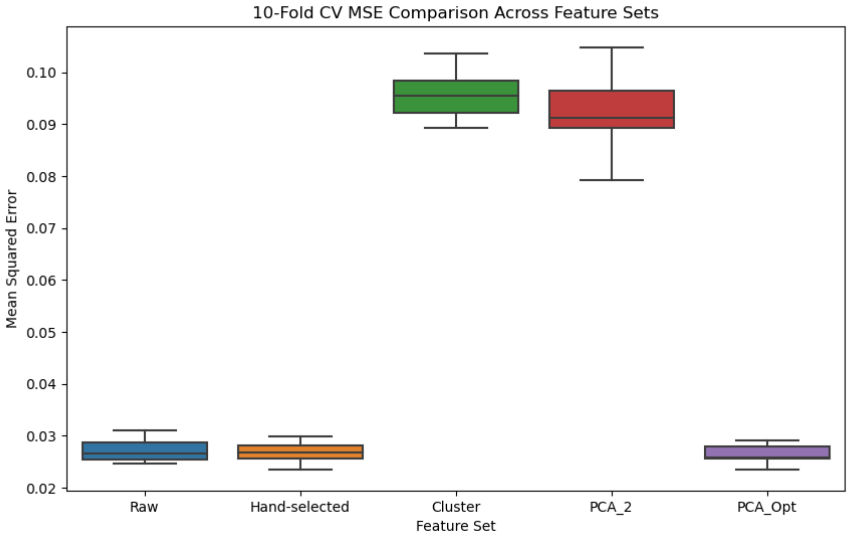

Figure 4.1: 10-Fold Cross-Validation MSE Comparison Across Feature Sets.

4.4 Test Set Performance

Figures 4.2 and 4.3 present the MSE and R^2 scores evaluated on the hold-out test set. Again, the **Hand-selected** feature set stands out, achieving the lowest test MSE of 0.0243 and the highest R^2 score of 0.8481. This implies not only high predictive accuracy but also strong explanatory power for variance in the target variable.

The **PCA_Opt** feature set comes in second, with a very close MSE of 0.0248 and $R^2 = 0.8362$. While the numeric differences are small, PCA_Opt provides a more automated and scalable pipeline compared to manual feature engineering. However, its slight performance drop suggests that while PCA preserved most of the data variance, it may have failed to fully retain certain domain-specific feature interactions that were captured by manual selection.

The **Raw** feature set achieves an MSE of 0.0279 and $R^2 = 0.8158$, which are still competitive, indicating that the original dataset already encodes useful predictive signals. However, its higher test error and lower variance explanation compared to the best two approaches confirm the benefit of feature curation or transformation.

In stark contrast, the **PCA_2** feature set yields a significantly higher test MSE of 0.0852 and a lower R^2 score of 0.4375. The **Cluster-based** feature set performs even worse, with an MSE of 0.0942 and $R^2 = 0.4100$. These results indicate that both excessive dimensionality reduction and unsupervised transformations misaligned with the target variable can substantially degrade predictive performance.

Specifically, the explained variance ratio of PCA_2 is only 0.8144, meaning that it retains just 81% of the original data variance. This considerable loss of information is a key contributor to its poor performance.

As for the Cluster-based features, their low effectiveness likely stems from the inherent limitations of unsupervised grouping. Since clustering does not take the output variable into account, the resulting clusters may fail to capture meaningful predictive patterns. Instead, they may discard informative features and introduce noise, ultimately undermining model accuracy.

4.5 Paired t-Test Results: Raw vs. Hand-selected

To evaluate whether the difference in performance between **Hand-selected** and **Raw** features is statistically significant, we conducted a paired t-test on the fold-wise MSE values from 10-fold CV using identical data splits.

With a p-value of approximately 0.5912 ($p \gg 0.05$), the result indicates that the difference in performance

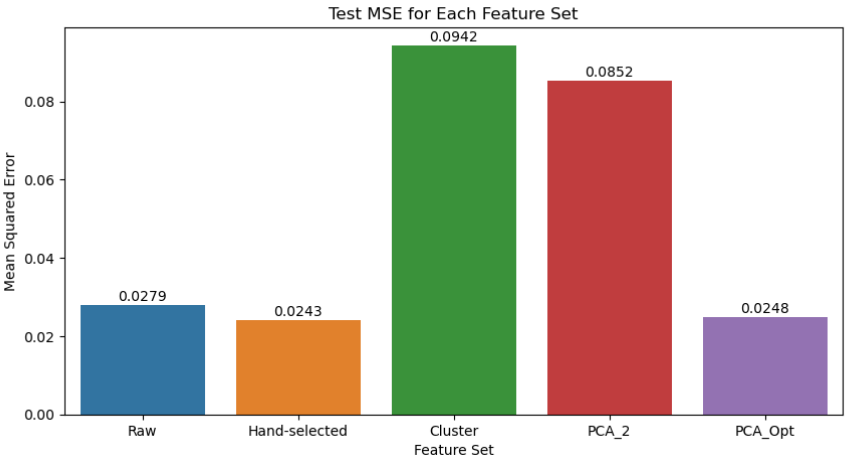

Figure 4.2: Test MSE for Each Feature Set.

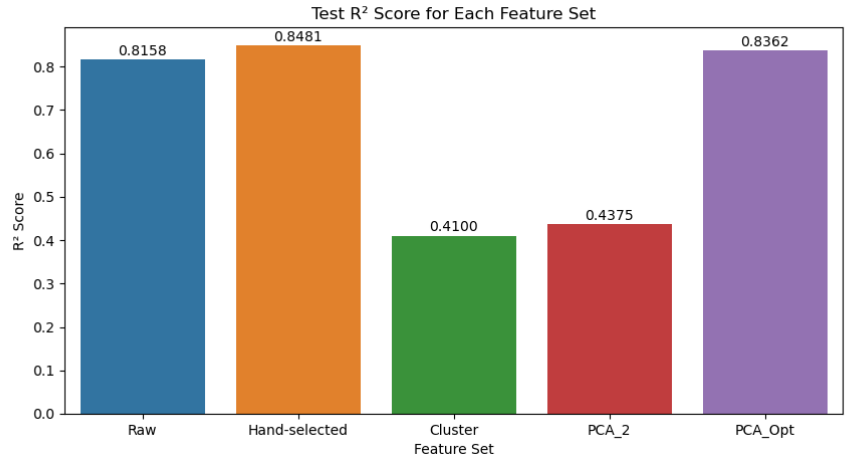

Figure 4.3: Test R^2 Score for Each Feature Set.

Comparison	**t-statistic**	**p-value**
Hand-selected vs. Raw	0.5569	0.5912

Table 4.1: Paired t-Test for Hand-selected vs. Raw Features

is *not* statistically significant. Thus, we cannot reject the null hypothesis that both feature sets perform equally well. Although Hand-selected shows lower average MSE, the performance variance across folds makes the observed difference statistically inconclusive.

This outcome highlights the importance of complementing performance metrics with statistical testing. Even when the average MSE favors one model, high variance and small sample sizes (i.e., 10 folds) can lead to non-significant findings.

4.6 Discussion and Concluding Remarks

Hand-selected Features. This approach achieved the best empirical performance across all metrics. Its low CV variance and top test set scores indicate that careful manual feature engineering can be highly effective, particularly when domain expertise is available. However, the lack of statistical significance relative to Raw features suggests the improvement may not be consistent across datasets or folds.

PCA_Opt Features. While slightly underperforming Hand-selected, PCA_Opt delivers robust and consistent results. It effectively removes redundancy and noise, achieving strong generalization with minimal human intervention. This method is especially useful in high-dimensional settings where interpretability or domain knowledge is limited.

Raw Features. The Raw feature set performs well and is statistically comparable to Hand-selected. This confirms that the original data contains meaningful signals, and that simple models may still yield strong results without complex preprocessing.

PCA_2 and Cluster-based Features. These feature sets consistently underperformed. PCA_2 suffers from overly aggressive dimensionality reduction, retaining insufficient variance. The Cluster-based approach likely misrepresents relationships with the target variable, as unsupervised clustering does not guarantee alignment with regression objectives. Future work could investigate supervised variants or clustering in target-conditioned latent spaces.

Practical Implications. From a deployment perspective, the choice between Hand-selected and PCA_Opt depends on trade-offs between:

- **Automation vs. Expertise:** Hand-selected features require human insight and domain knowledge; PCA_Opt is automatic.
- **Interpretability:** Raw and Hand-selected features are more interpretable than PCA-derived features.
- **Reproducibility:** PCA-based methods offer consistent transformations and easy scalability across datasets.

In summary, the Hand-selected feature set currently yields the best performance, but PCA_Opt and Raw are statistically comparable and offer benefits in terms of automation and simplicity. These findings suggest that in this dataset, both expert-driven and algorithmic feature engineering pipelines can be effective, provided that careful attention is paid to preserving information content and model stability.

5 Conclusion

5.1 Summary of Work

In this milestone, we systematically examined how different feature-engineering approaches affect a regression model's performance on a dataset containing 40 features. Building on insights from Milestone 1, we utilized an Artificial Neural Network (ANN) with consistent hyperparameters (e.g., learning rate, number of epochs) and identical data splits to ensure a fair comparison. We evaluated:

- **Raw Features (no transformation).**

- **Hand-Selected Features** chosen based on domain knowledge, statistical correlation, and variance analysis.
- **Cluster-Based Features**, derived from unsupervised K-means clustering.
- **PCA with 2 Components (PCA 2C)**.
- **PCA with an Optimal Number of Components (PCA Opt)** to maintain a reconstruction error below 10^{-6}.

All feature sets were trained and evaluated with the same ANN architecture using 10-fold cross-validation and a hold-out test set. We focused primarily on Mean Squared Error (MSE) and the coefficient of determination (R^2) to quantify predictive performance.

5.2 Key Findings

1. **Hand-Selected Features yielded the best numerical performance.**
   - Achieved the lowest test MSE (approximately 0.0243) and the highest R^2 (about 0.8481).
   - Had relatively low variance across folds, suggesting robust generalization.
   - However, a paired t-test comparing Hand-Selected vs. Raw features did *not* yield statistical significance (p-value around 0.59), indicating that the difference in average performance may not be consistent across different splits.
2. **PCA with an Optimal Number of Components (PCA Opt) performed nearly as well.**
   - Retained enough principal components (e.g., ~ 26) to capture almost all data variance (cumulative explained variance ≈ 1.0).
   - Produced an MSE close to the hand-selected approach (e.g., 0.0248 on the test set) and an R^2 of roughly 0.8362.
   - Offers a more automated approach than manual feature selection, though marginally less accurate in this particular dataset.
3. **Raw Features** still yielded competitive accuracy:
   - Test MSE around 0.0279 and $R^2 \approx 0.8158$.
   - Demonstrates that the original dataset already possessed considerable predictive signal.
4. **PCA with 2 Components (PCA 2C)** and **Cluster-Based Features** underperformed:
   - PCA 2C preserves only about 81% of data variance, leading to a noticeable loss of predictive power (test MSE: 0.0852, $R^2 : 0.4375$).
   - Cluster-based features (derived via K-means) showed the highest MSE (about 0.0942) and the lowest R^2 (≈ 0.4100). The unsupervised clusters did not align well with the regression target, resulting in poor accuracy.

5.3 Limitations and Detailed Analysis

- **Interpretability:** Although PCA-based transformations effectively reduce dimensionality and can improve model efficiency, they inevitably obscure the interpretability of individual predictors. Hand-selected or raw features tend to be more interpretable but may retain redundant or noisy dimensions.
- **Clustering Approach:** The cluster-based features employed a simplistic K-means algorithm with $k = 2$. This choice proved suboptimal for the regression task, as clustering ignores the target variable and may capture irrelevant structural patterns.

- **Single Model Architecture:** All experiments used the same ANN structure. Different architectures or even different types of models (e.g., random forests, gradient boosting) could exhibit different sensitivities or advantages for each feature set.
- **Statistical Significance:** The paired t-test between the best two feature sets (Hand-Selected vs. Raw) revealed that their performance gap is not statistically significant at $p < 0.05$. This underscores the importance of variance across folds and possible data-set-level idiosyncrasies.

5.4 Future Work

There are several potential avenues for continued research:

- **Advanced Dimensionality Reduction:** Non-linear methods like autoencoders, t-SNE, or UMAP may preserve complex relationships in the data more effectively than classical PCA.
- **Refined Clustering Techniques:** Target-aware or semi-supervised clustering could produce more relevant cluster-based features. Alternatively, experimenting with GMMs, DBSCAN, or larger k values might yield better representation.
- **Regularization and Embedded Methods:** Incorporating L1 or elastic net regularization, or using tree-based ensemble models, could provide a more systematic way to identify and reduce feature redundancy while retaining predictive power.
- **Comparison Across Model Families:** Beyond ANNs, evaluating different regression algorithms can clarify whether these feature engineering techniques generalize or interact differently with each model class.

5.5 Milestone 2 Work Distribution

The responsibilities within our group were distributed as follows:

- **Hongyi Zhang**: Performance Comparison, Report Writing.
- **Ziang Deng**: Feature Selection, Report Writing.
- **Nancy Wang**: PCA, Report Writing.

6 Appendix

Table 6.1: PCA Components and Cumulative Variance Explained

Number of Components	Cumulative Variance
1	0.75165658
2	0.81437731
3	0.87399624
4	0.91218280
5	0.93687225
6	0.95384530
7	0.96734878
8	0.97706188
9	0.98493070
10	0.98902419
11	0.99162533
12	0.99350001
13	0.99490351
14	0.99612262
15	0.99712805
16	0.99798175
17	0.99876602
18	0.99945310
19	0.99968385
20	0.99978189
21	0.99985520
22	0.99990535
23	0.99993980
24	0.99996619
25	0.99998643
26	1.00000000
27	1.00000000
28	1.00000000
29	1.00000000
30	1.00000000
31	1.00000000
32	1.00000000
33	1.00000000
34	1.00000000
35	1.00000000
36	1.00000000
37	1.00000000
38	1.00000000
39	1.00000000
40	1.00000000