DATA221 Final Project

Dariel Cruz Rodriguez Yordi Hernandez

The goal of the project is to explore a number of datasets that may be associated with political instability in the U.S. The data was taken from the Seshat Databank under Creative Commons Attribution Non-Commercial (CC By-NC SA) licensing.

Data Wrangling

- ☒ Read the (short) code book.
- ☒ Numerical data need to be uploaded, interpolated, and properly saved. For the purpose of this project, interpolate each variable such that you obtain one point per year (within the range of available data).
- ☒ Calculate (and then interpolate) the political instability index.
- ☒ Display the DataFrame with all of the columns and the interpolated data for the years 1901-1910.

Table 1: The DataFrame with all of the columns and the interpola

	time	polarization	ratio	assassination	executions	insurrection	lynching	mass suicide	rampage
0	1901	0.845472	0.651275	1.0	0.0	0.0	3.0	0.0	0.0
1	1902	0.860217	0.645180	0.0	0.0	0.0	3.0	0.0	0.0
2	1903	0.868912	0.639086	0.0	0.0	0.0	5.0	0.0	0.0
3	1904	0.877794	0.627711	0.0	0.0	0.0	4.0	0.0	0.0
4	1905	0.890111	0.606085	1.0	0.0	0.0	0.0	0.0	0.0
5	1906	0.898414	0.596688	0.0	0.0	0.0	3.0	0.0	0.0
6	1907	0.891646	0.607063	0.0	0.0	0.0	0.0	0.0	0.0
7	1908	0.882256	0.610940	0.0	0.0	0.0	9.0	0.0	0.0
8	1909	0.873668	0.608747	0.0	0.0	0.0	7.0	0.0	0.0
9	1910	0.860964	0.602996	0.0	0.0	0.0	5.0	0.0	0.0

Exploratory Data Analysis

☒ Conduct an exploratory data analysis

☒ Summarize your main findings (most interesting/insightful conclusions) in a short paragraph and include appropriate visualizations.

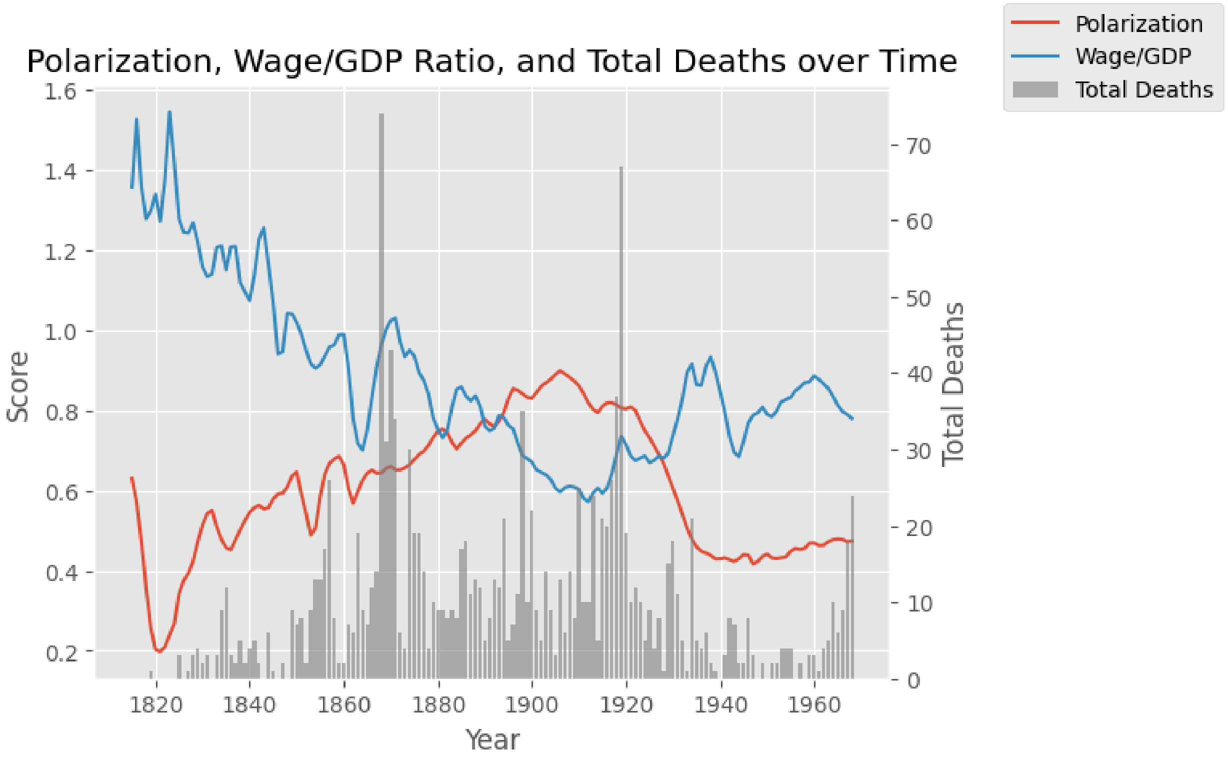

Figure 1: Line graph/bar plot showing the relationship between Polarization & Wage/GDP over time and total deaths

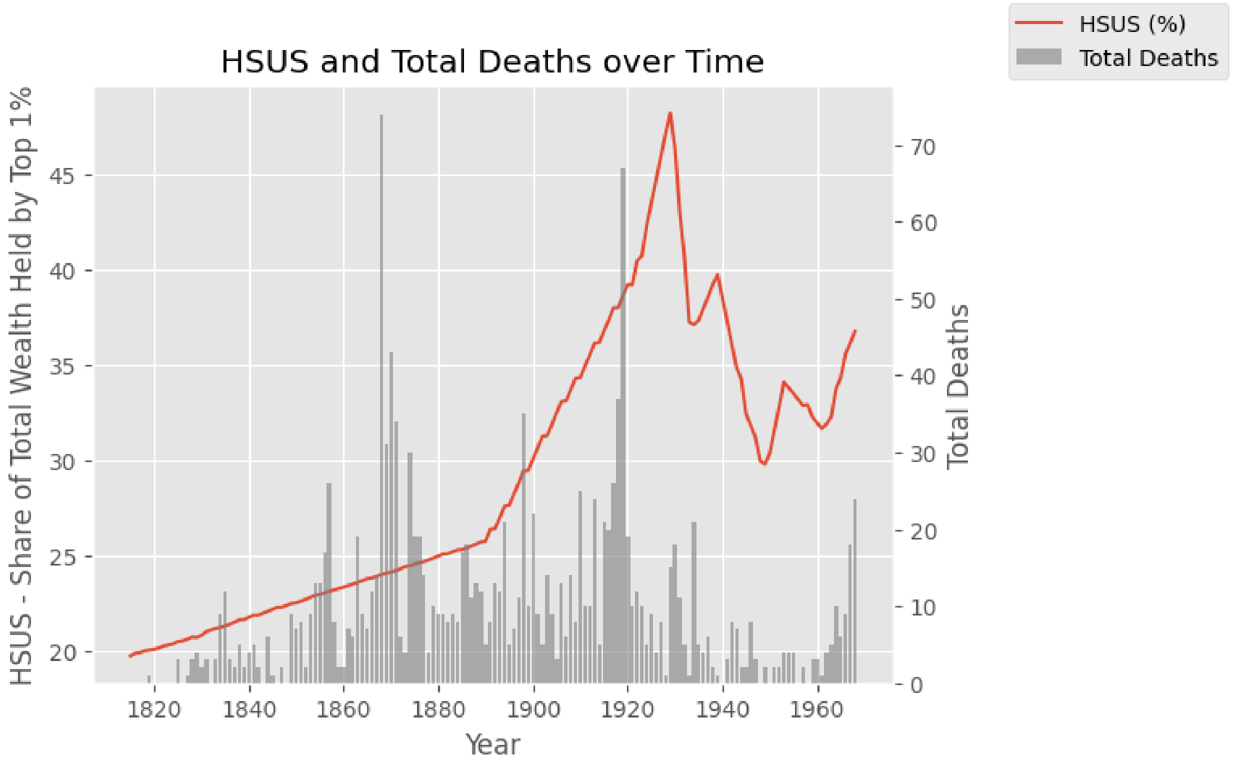

Figure 2: Line graph/bar plot showing the relationship between HSUS over time and total deaths

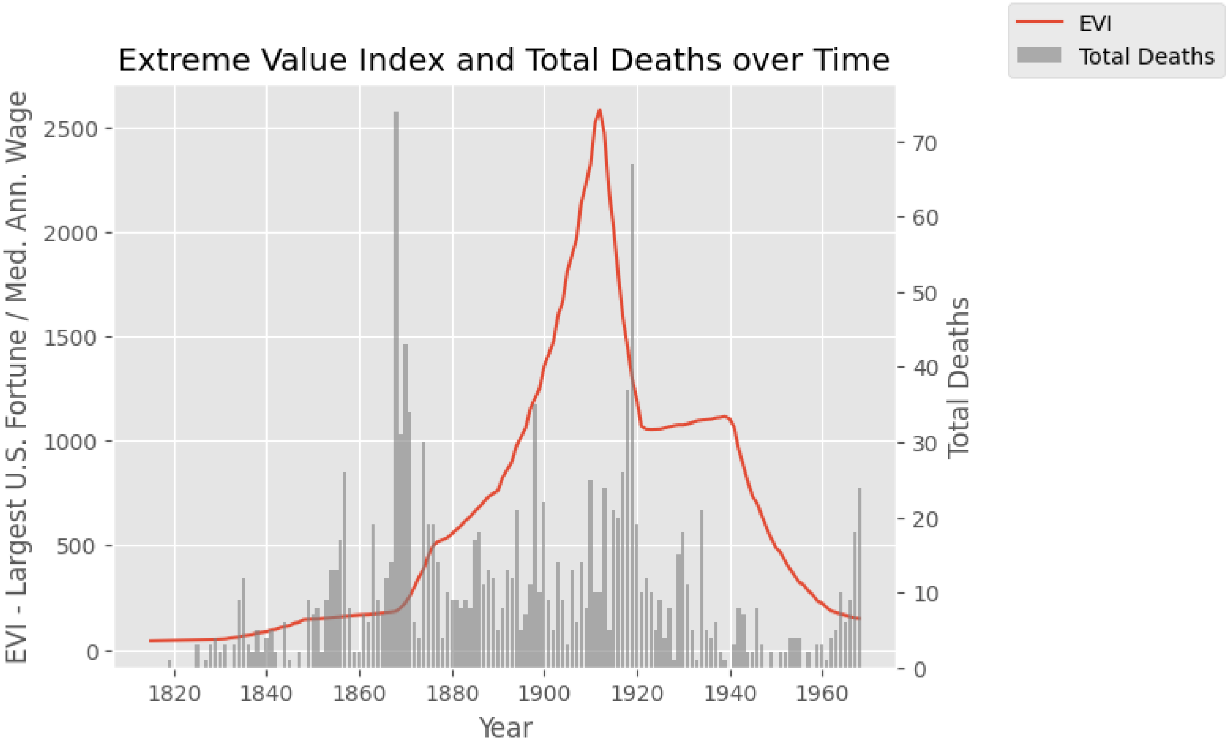

Figure 3: Line graph/bar plot showing the relationship between EVI over time and total deaths

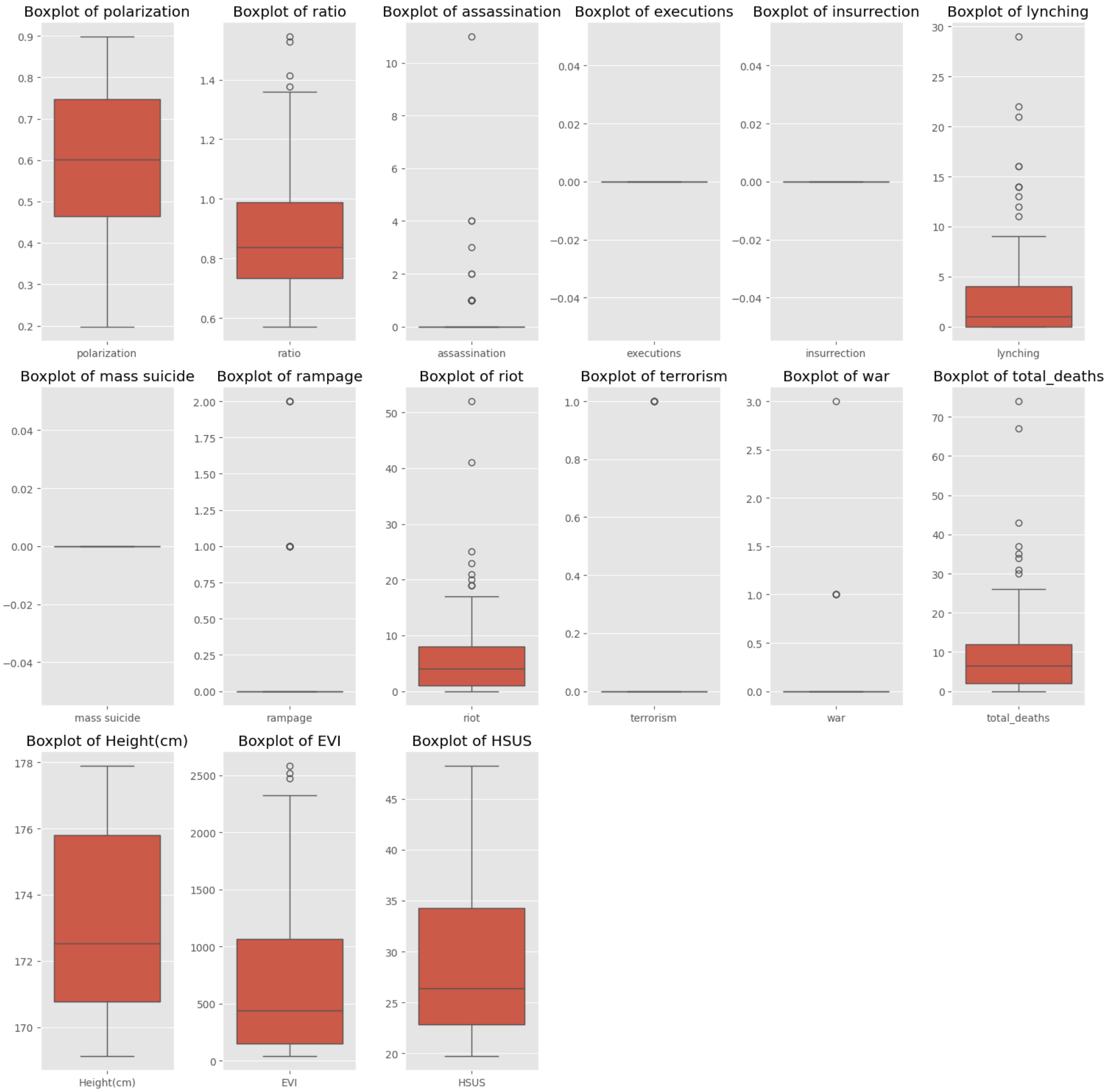

Figure 4: Boxplots showing the distribution of each variable in the dataset

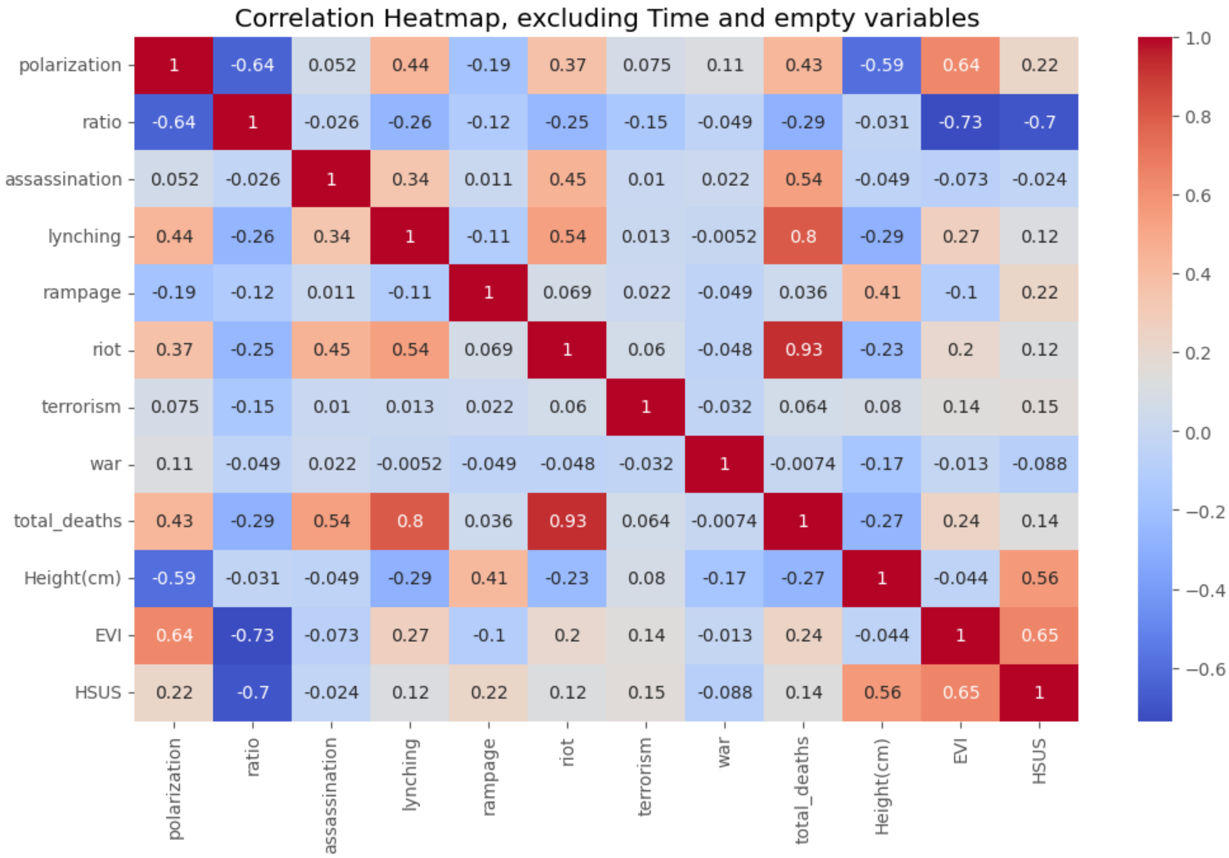

Figure 5: Correlation heatmap showing the relationship between each variable in the dataset, excluding Time and empty variables

After conducting an exploratory data analysis, we have come to several conclusions.

Firstly, within the subset of the years 1815 and 1968 that we selected for our analysis, political instability remained relatively stable over the years, in fact actually decreasing as 1968 approached. In addiiton, political instability appears visually to be inverse (or negatively correlated) to the Wage/GDP ratio. This is interesting because it suggests that as the wage/GDP ratio increases, political instability decreases. Another way to describe this could be that as wages increase, political instability generally decreases. This can be verified mathematically as well, as the correlation coefficient between the ratio and polarization is -0.64, which represents a strong negative correlation.

Additionally within the correlation table, the wage/GDP ratio is also strongly negatively correlated with the Extreme Value Index and HSUS which is entirely reasonable because both metrics represent larger values for more wealth disparity and smaller values for wealth equality while the ratio is higher for wealth equality and smaller for wealth disparity, the complete opposite.

Another interesting finding is how strongly the variable Height(cm) is correlated with multiple metrics in the dataset. Height is negatively correlated with polarization at a value of -0.59 and with HSUS at a value of 0.56, suggesting that as height increases, polarization decreases. I do not believe this is causation however, because at the same time height is positively correlated with time at a value of 0.64, leading me to believe that height can be construed as another measure for time. As time goes on, people got taller on average but coincidentally at the same time political instability went down.

The only apparent trend I could identify visually is the similar bumps and increase in both HSUS and EVI metrics at the same time, presumably because they measure similar things.

There are 154 entries in our dataset, which could lead to a lot of noise later on when we build our models. This is accompanied as well by several outliers in the individual types of deaths that occured in each year and a few outliers with ratio and EVI. To account for this, we may need to consider reducing the number of features we consider in our dataset. Specifically, I would consider dropping time and total_deaths, because they are directly linked to other variables in the set.

Find the best regressor

Find the best regressor that would predict the instability index from the various predictors. To be clear, you are asked to compare a limited set of regressors of your choice – not to identify the theoretically optimal one.

- ☒ Explain your modeling choices.
- ☒ Interpret any evaluation metrics you use.
- ☒ Summarize your conclusions in a short paragraph, i.e., the most interesting conclusion(s), the model that produced it, and how the model was chosen. You may include a figure if you find it helpful.

Table 2: R2, MSE, and MAE scores aross multiple regression models

	R2	MSE	MAE
Linear Regression	1.000000	8.981265e-29	7.918016e-15
Ridge Regression	0.999942	4.836788e-03	5.563760e-02
Lasso Regression	0.996825	2.632591e-01	2.519983e-01
Random Forest	0.942836	4.739526e+00	1.332553e+00

After choosing 4 of the most used regressors (Linear, Ridge, Lasso, and Random Forest classifier), the models were trained and tested in order to compare which one performed the best. The metrics used in order to compare performance were R^2, which explains variance, MSE, which penalizes larger errors heavily, so the lower score the better, and MAE, where the lower the score the better.

Based on the results, Linear Regression performed the best as it had the highest R^2, lowest mean squared error, and mean absolute error. The performance was nearly perfect since it explained nearly 100 percent of the variance in the instability index with little error. Because Linear Regression achieved such great results, complex models like Random Forest were not necessary. Ridge and Lasso models performed well, yet provided no advantage, confirming that feature redundancy and multicollinearity were not concerns.

Additionally, the regressors were tested with a smaller subset containing only 10 years, which yielded similar results. Linear Regression was still the best performer.

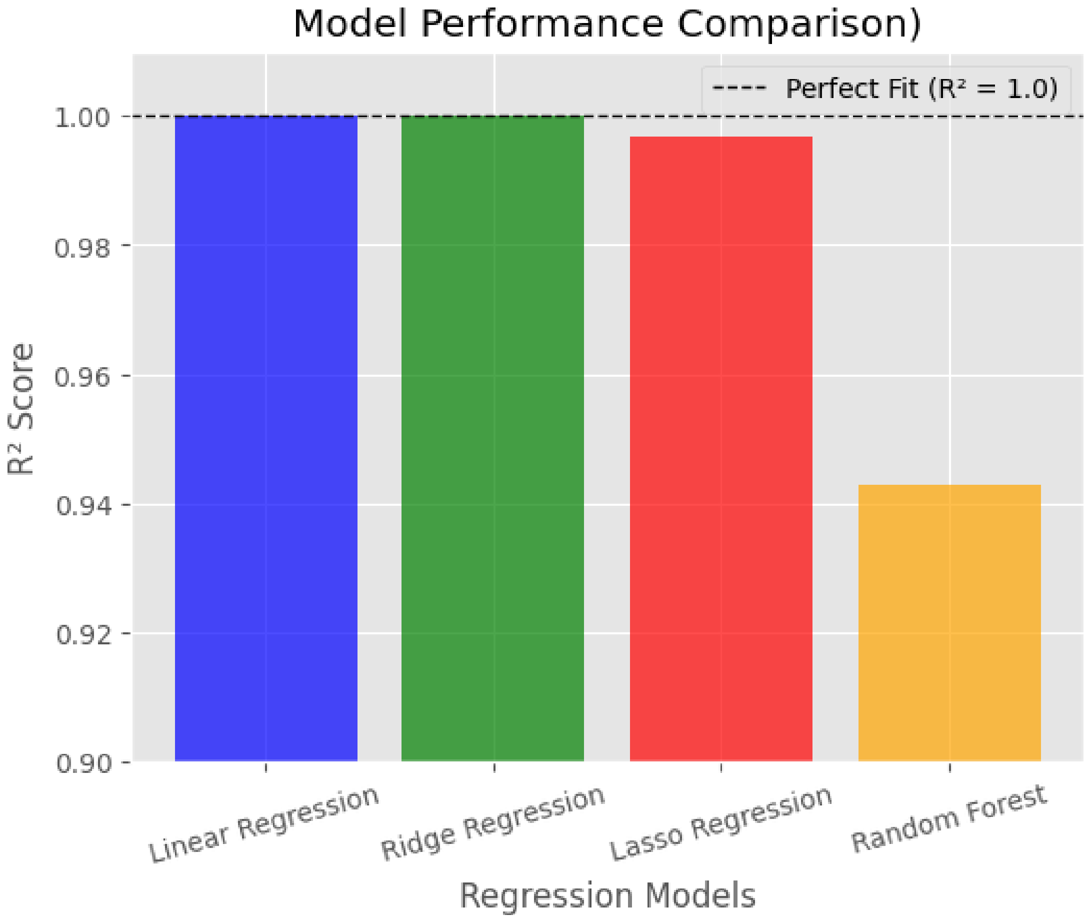

Figure 6: R2 scores for each regression model

Find the best dimensionality reduction for regression

You can restrict this part to reducing the data to two dimensions, to three dimensions, or explore both options. You can test your variables using the best regressor found in the previous section or a small number of regressors (2-3 models at most).

- ☒ Explain modeling choices and evaluation metrics.
- ☒ Summarize your conclusions in a short paragraph, i.e., the most interesting conclusion(s), the model that produced it, and how the model was chosen. You may include a figure if you find it helpful

---

Table 3: R2, MSE, and MAE scores for each dimensionality reduction method

	Reduction Method	R2 Score	MSE	MAE
0	PCA (2D)	0.355357	53.448476	5.832352

	Reduction Method	R2 Score	MSE	MAE
1	PCA (3D)	0.922641	6.413987	1.957521
2	Feature Selection (Top 2)	0.989286	0.888305	0.596215
3	Feature Selection (Top 3)	0.994036	0.494466	0.405268

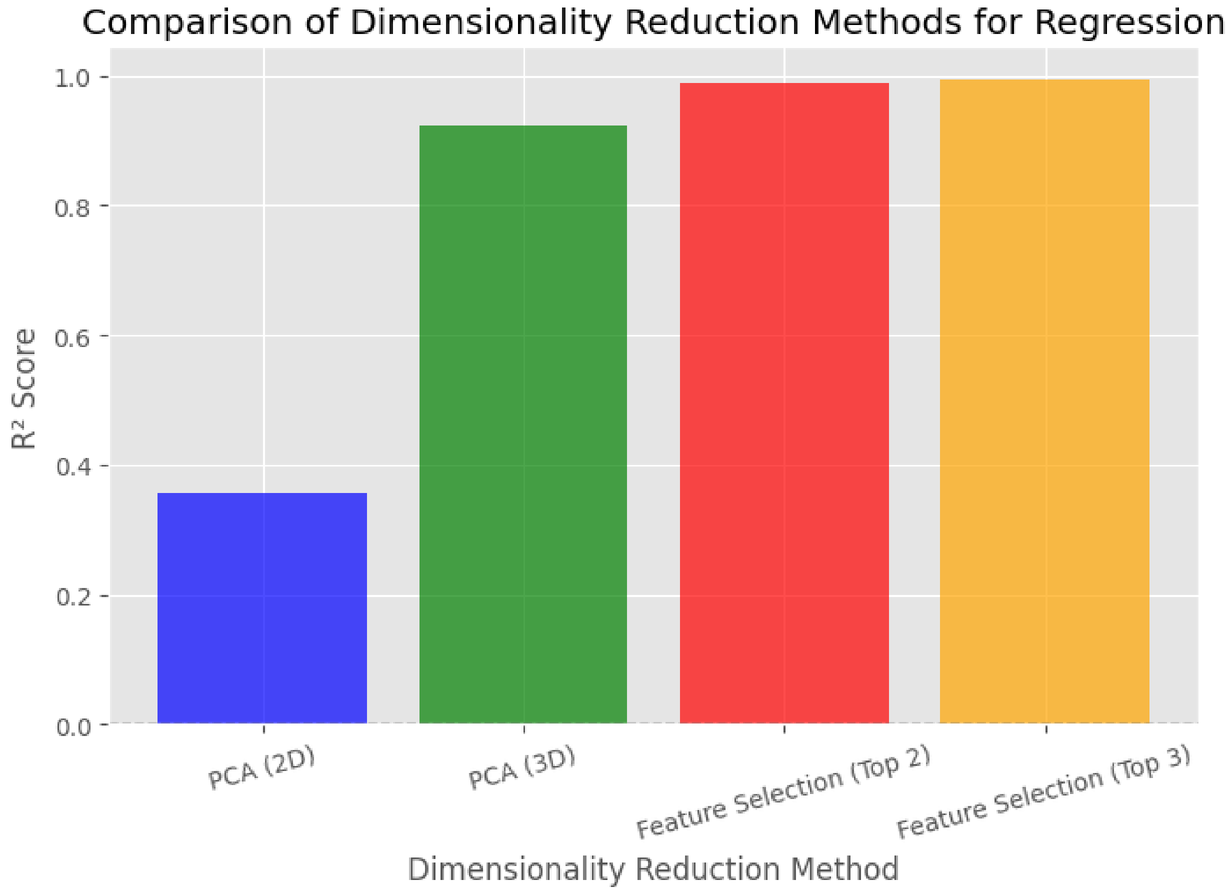

Figure 7: R2 scores for each dimensionality reduction method

```
9.37 + (1.08 * assassination) + (4.49 * lynching) + (7.24 * riot)
```

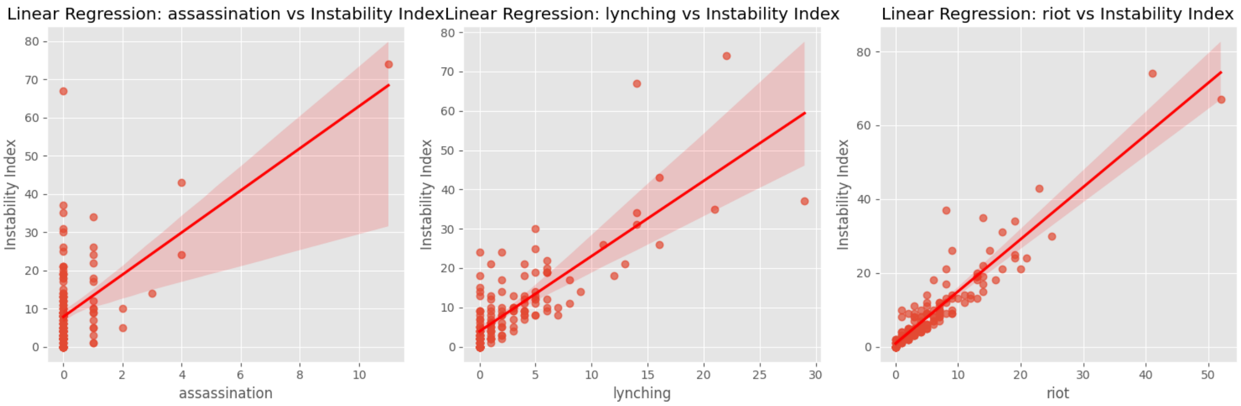

Figure 8: Linear regression model with selected features

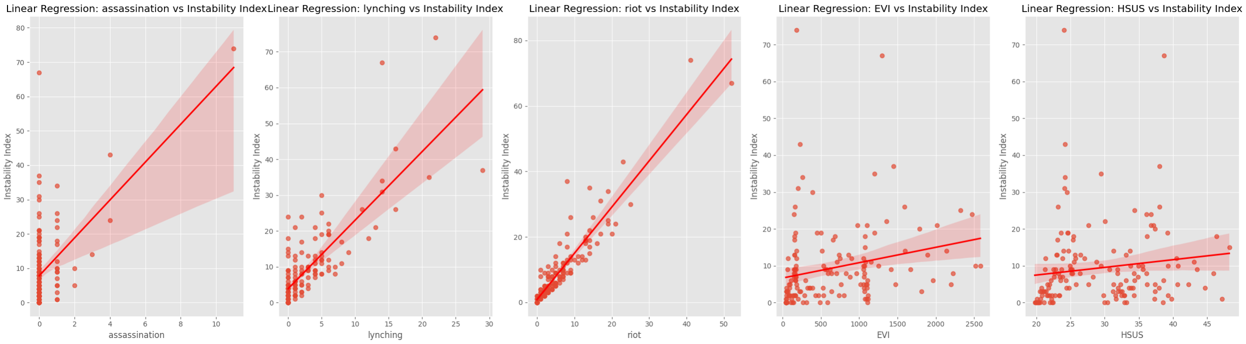

Figure 9: Linear regression model with additional selected features (EVI, HSUS)

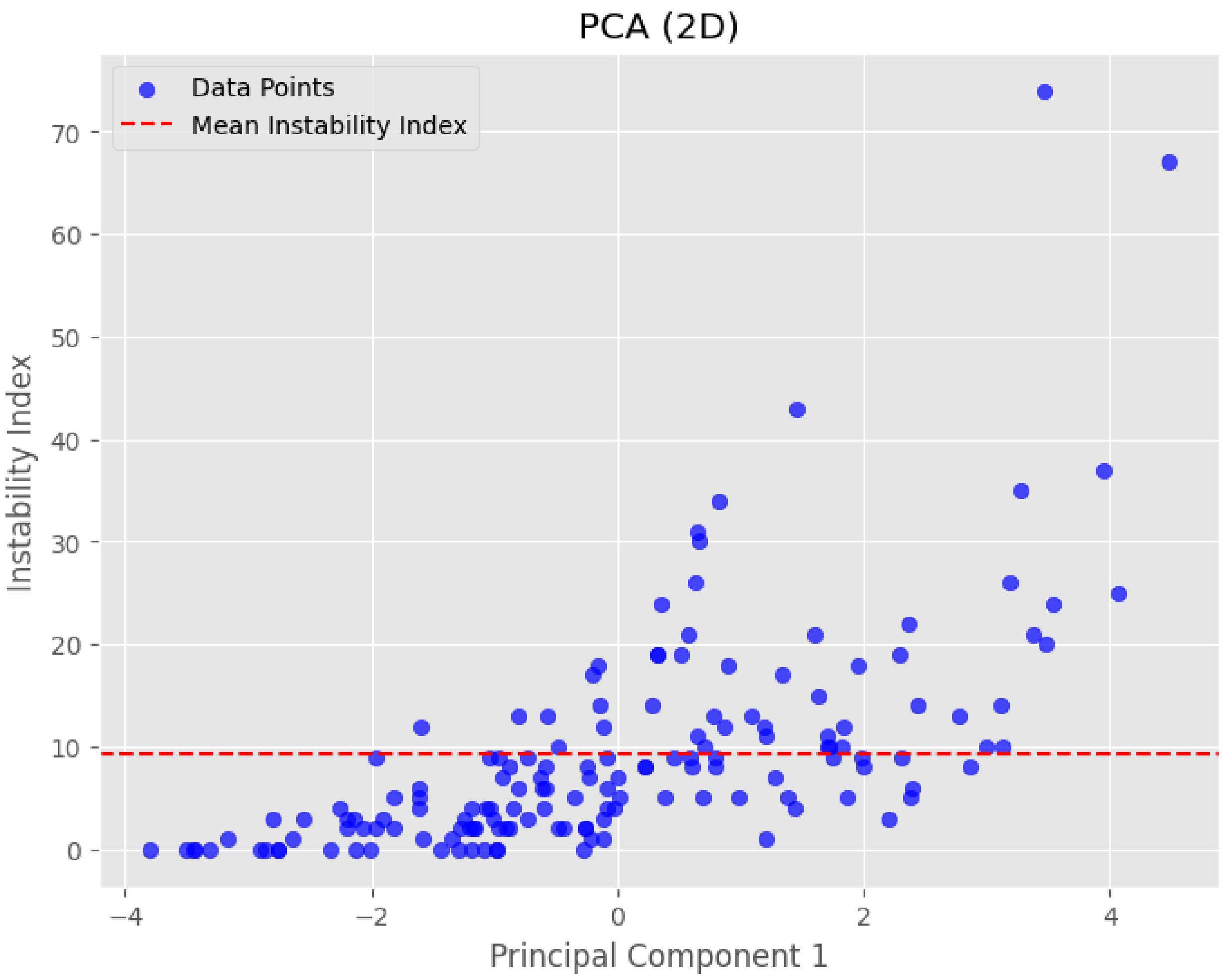

Figure 10: PCA plot in 2D

In order to compare dimensionality the choices used were PCA which it's purpose was to capture the most variance in the data.

Feature selection was also implemented. The purpose of feature selection is to retain the most relevant predictors based on significance. Similarly to the problem above, R2 score, MSE, and MAE were used as metrics.

After executing the model, Feature selection with 3 features performed the best suggesting keeping the main predictors yields better results. On the other hand PCA performed poorly suggesting that important data is lost when performing the model. Based on the results it is safe to say Linear regression with selected features is the best option as it provides both interpretability and superior regression performance.

Find the best dimensionality reduction for unsupervised classification

Use only the predictor columns and not the outcome (instability) for classification. You can restrict this part to reducing the data to two dimensions, to three dimensions, or explore both options.

☒ You can test your variables using k-means or a small number of classifiers (2-3 models at most).
☒ Explain modeling choices and evaluation metrics.
☒ Summarize your conclusions in a short paragraph, i.e., the most interesting conclusion(s), the model that produced it, and how the model was chosen. You may include a figure if you find it helpful.

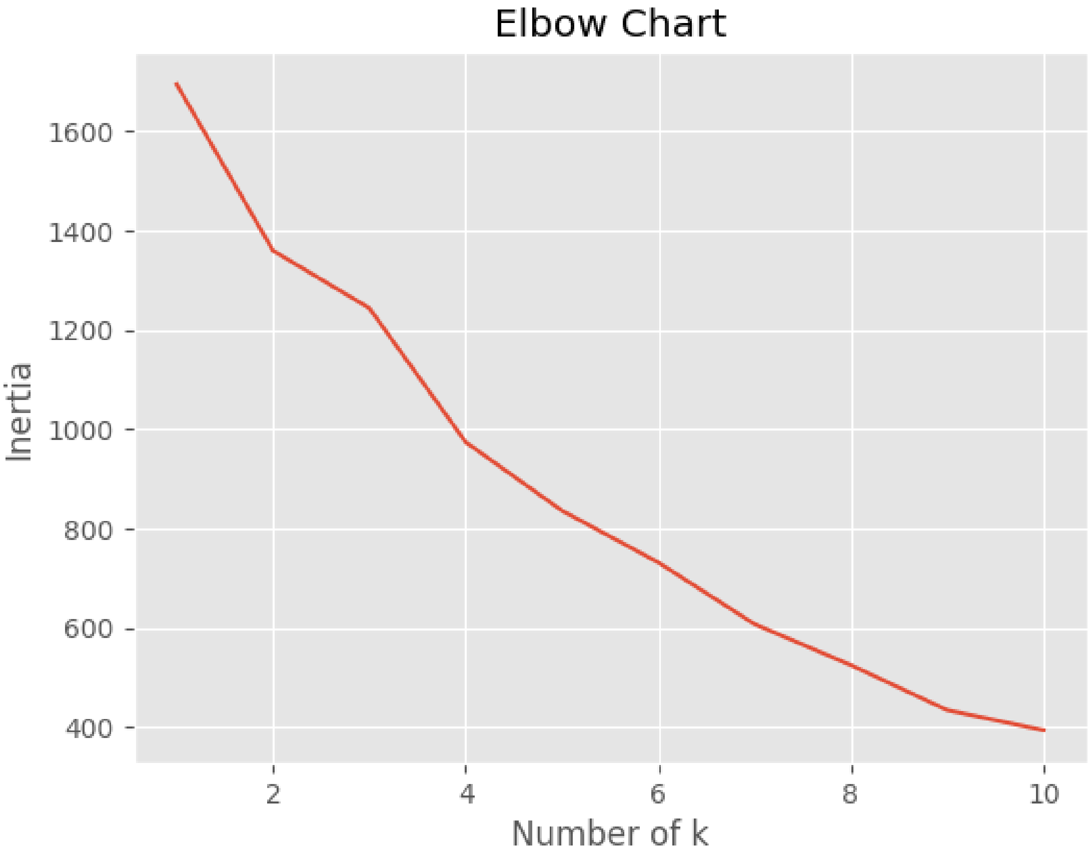

Figure 11: Elbow chart to determine the optimal number of clusters for KMeans

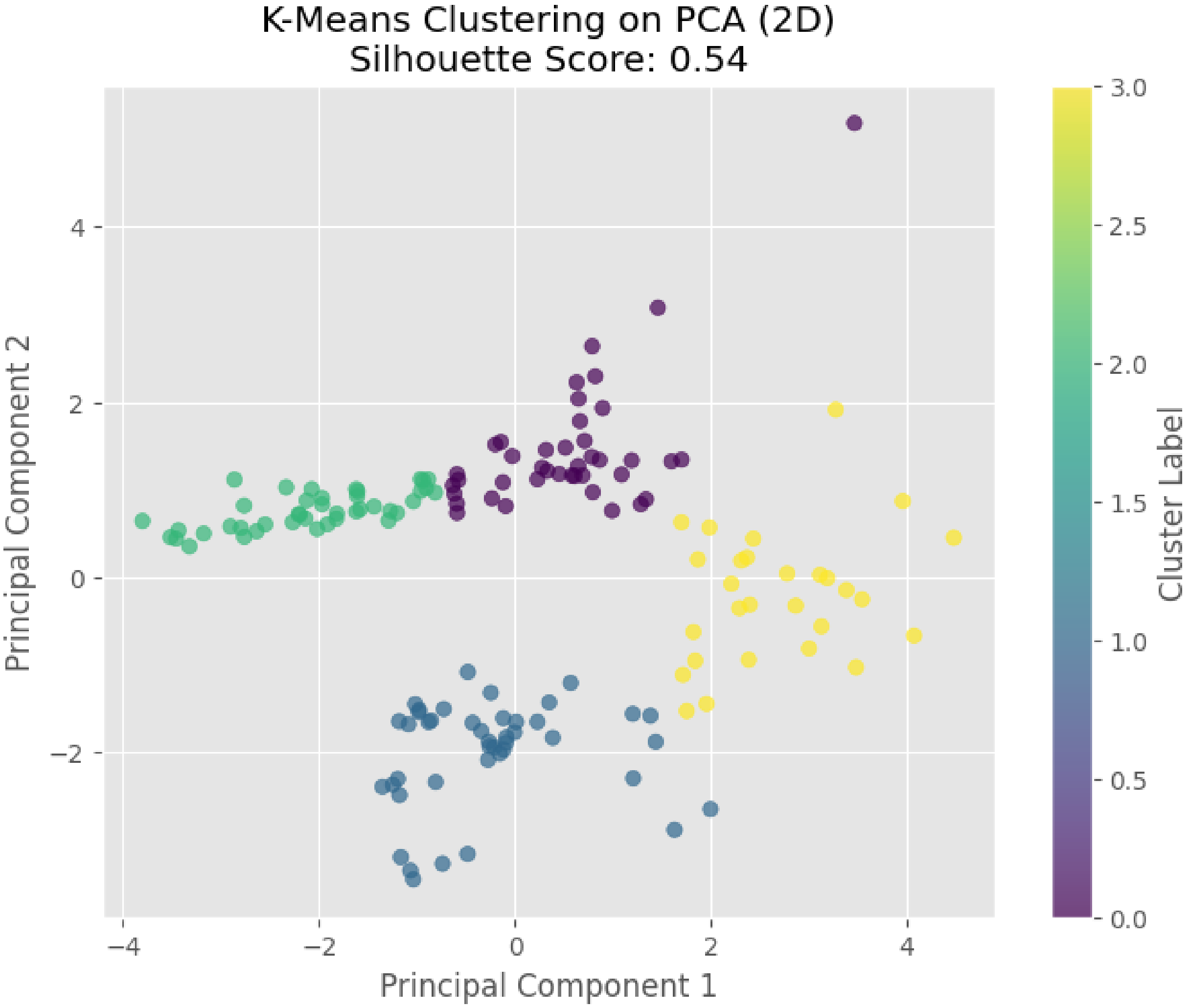

Figure 12: K-Means clustering on PCA (2D)

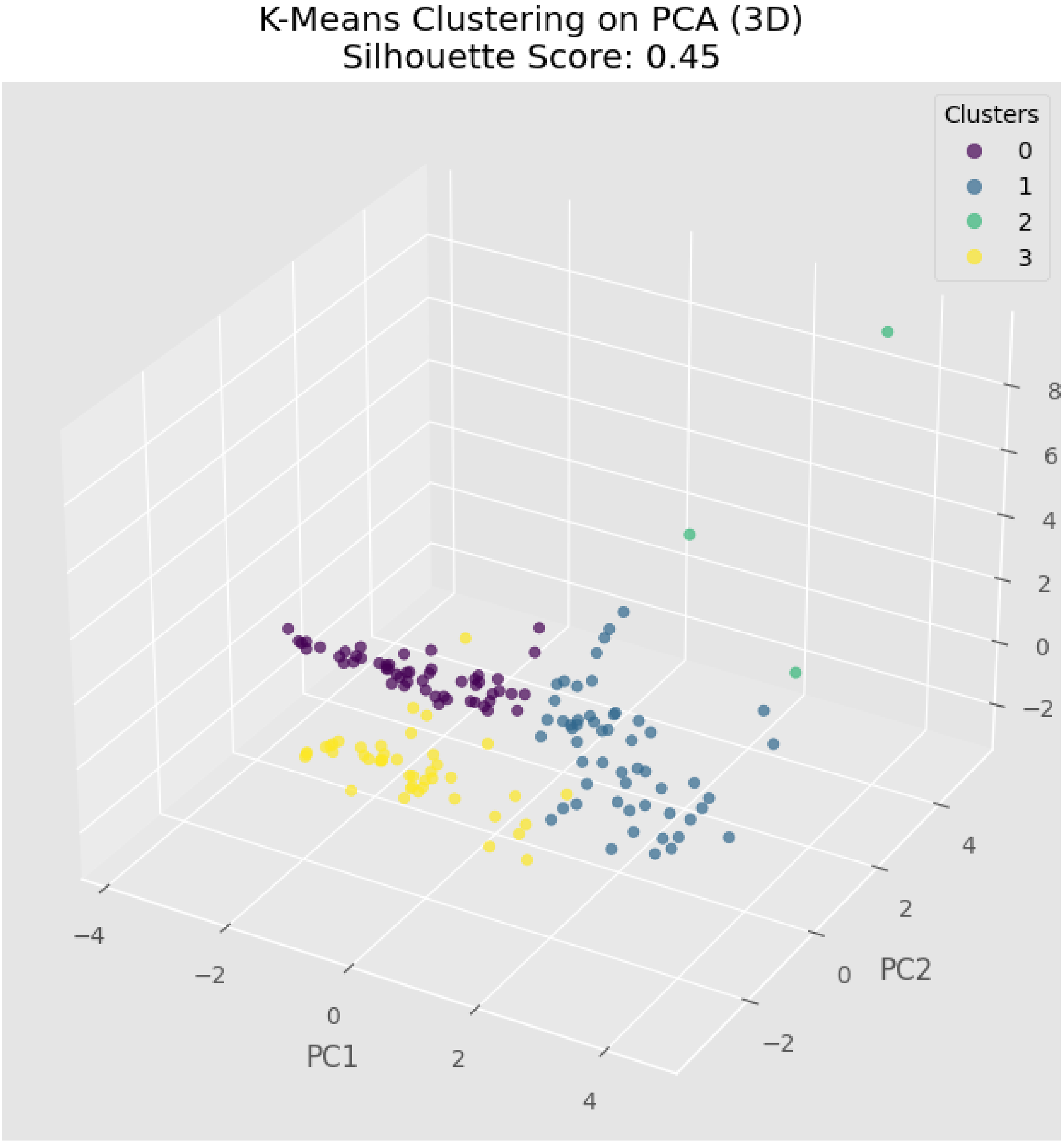

Figure 13: K-Means clustering on PCA (3D)

We applied PCA to reduce the predictor data into lower dimensions (2D and 3D) so that the structure of the data can be visualized and processed. K-Means clustering was the chosen model to classify the data based on the reduced dimensions because it's suitable for continuous, numeric data like the ratios and scores in our dataset. (Figure 12 and Figure 13)

To determine the optimal number of clusters we evaluated multiple K-values using the Elbow method (Figure 11) and then confirming our selected K-value by comparing the Silhoutte score (Figure 12 and Figure 13, figure title). The elbow method is a visual tool to determine at what value *k* intertia begins to level off, another way to represent the point where adding more clusters produces diminishing returns. The silhoutte score measures how well-distinct and compact clusters are by measure the relative distance to eachother within the cluster. Using both of these metrics allowed us to select a k value that did not have high interia and too few clusters, while also avoiding the opposite issue of having too many clusters and overfitting our data.

Our findings showed that reducing the data to 2D and 3D dimensions using PCA produced moderate separation amongst the clusters (which was the best result we could achieve). The K-Means model with 4 clusters was the best performer, as it produced the highest Silhoutte score and the most distinct clusters. This suggests that the data can be classified into 4 distinct groups based on the reduced dimensions. Despite this, whether it be due to the data itself or otherwise, the clusters still show some overlap at the boundaries and aren't as distinct as we'd have liked. (Figure 12, green and purple clusters)

Briefly explore the clusters of instability scores

Consider the cluster labels from the best clustering scheme from previous section or from clustering using all/most of the original features. Apply it to the corresponding records of the outcome column (instability).

☒ Create a visualization of the results.
☒ Summarize your conclusions. To be clear, the summary can be very short, and may be that the clusters do not exhibit any discernable or interpretable pattern. You may include a figure if you find it helpful.

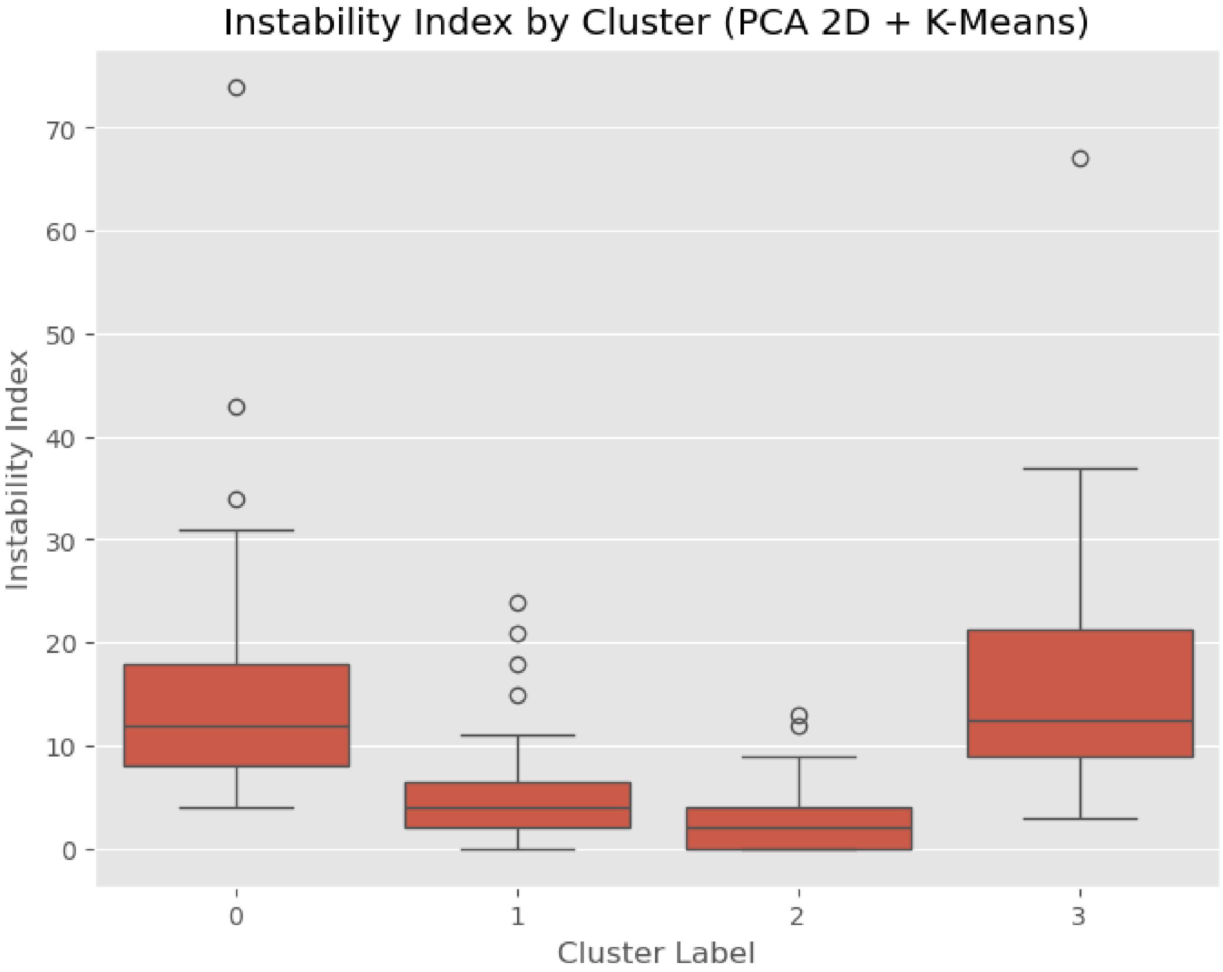

Figure 14: Instability index distribution by cluster

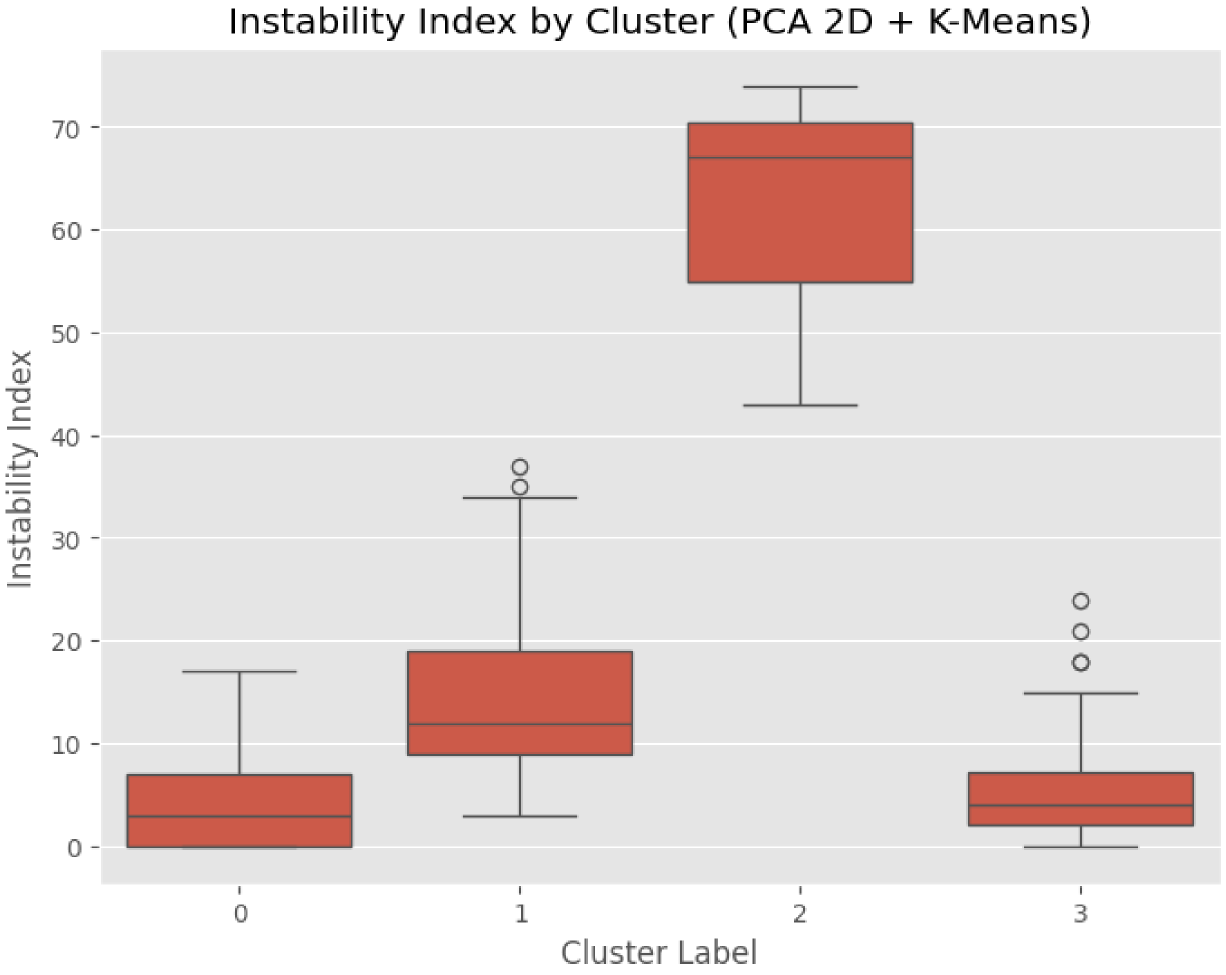

Figure 15: Instability index distribution by cluster

Each cluster exhibits a different median Instability Index— Clusters 1 and 2 show lower medians between 5 and 6 while Clusters 0 and 3 show higher medians between the values 10-15. (Figure 14)

There is a notable overlap amongst the distributions of all the clusters, suggesting that the clusters may not be well defined. Overall, while ther eare some differences in central tendencies, the clusters don't separate cleanly by Instability Index and there is not a clear, discernable, and interpretable pattern that is useful for our analysis.

In the PCA 3D model with K-Means clusteirng, the distribution of cluster 2 is more spread out and pronounced than the other clusters, with no overlap. The other clusters however, still have significant overlap and are not as distinct. The data does not separate cleanly by instability index, suggesting that instability is not the only factor impacting our parameters. (Figure 15)

Consider a real life modeling/prediction problem

☒ Try a large number (100? 1000?) different models

☒ Examine their performances

☒ Select the one that scores best on your performance metric of choice

☒ Briefly discuss the potential disadvantage (or potential danger) of such and approach and how you might go about mitigating it.

Table 4: MSE and R2 scores for each regression model averaged over 100 epochs

	Model	MSE	R2 Score
5	Gradient Boosting	36.492536	0.578757
4	Random Forest	45.321113	0.476846
0	Linear Regression	55.276529	0.361928
2	Lasso Regression	55.578276	0.358445
1	Ridge Regression	56.965077	0.342437
3	Decision Tree	168.892049	-0.949566

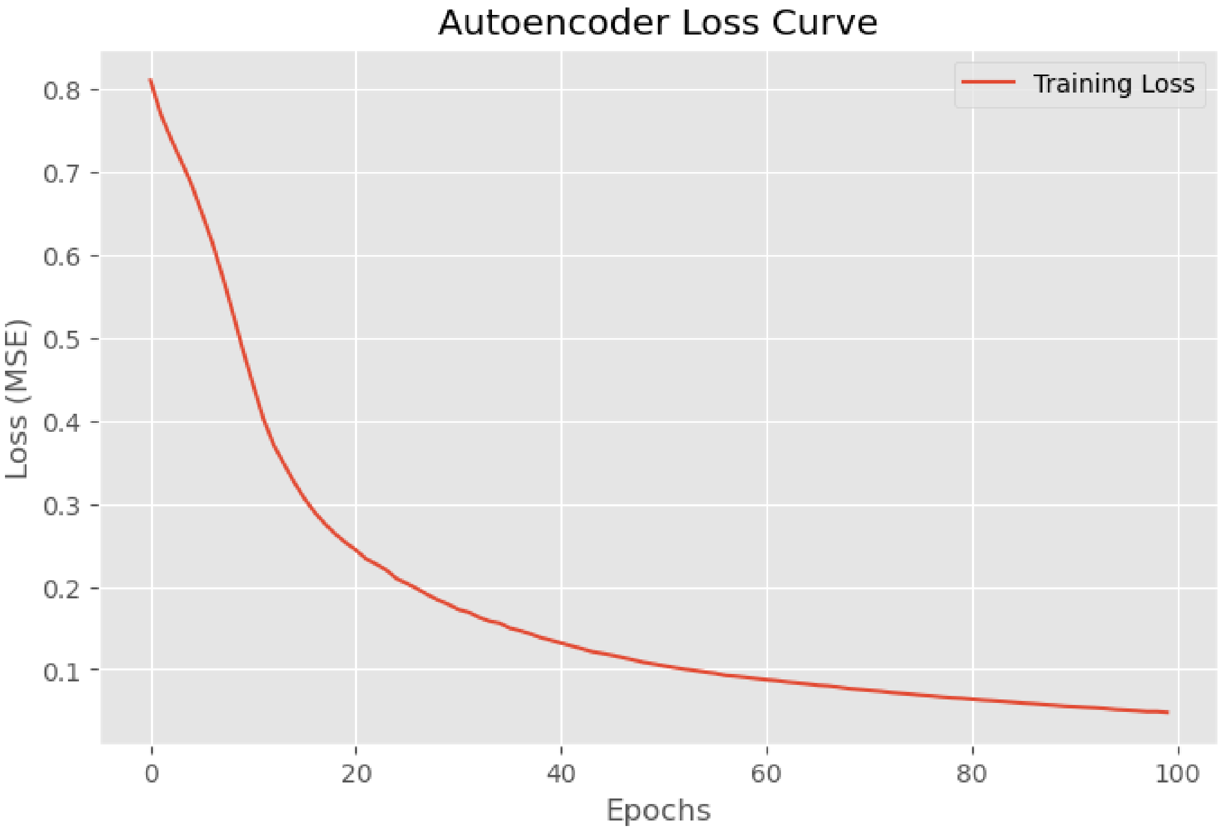

Figure 16: Training loss curve for the autoencoder model

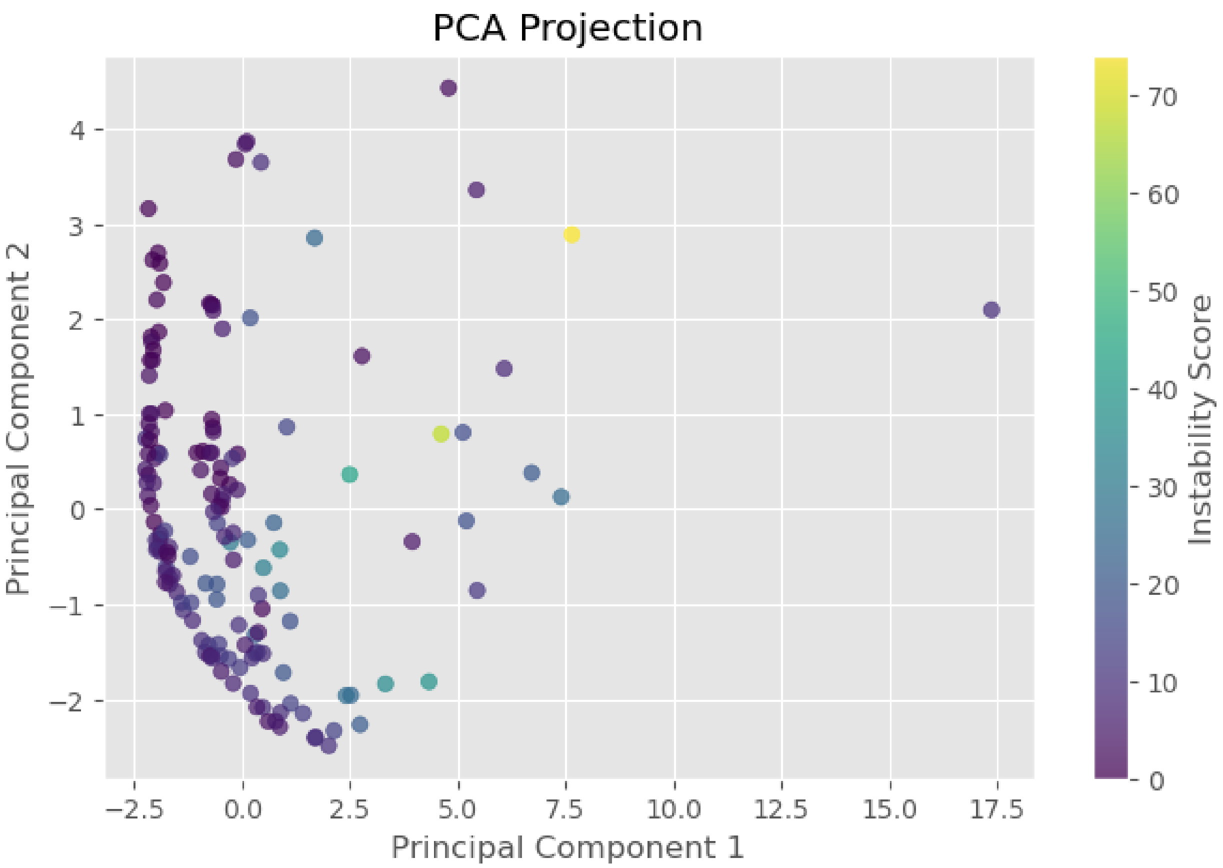

Figure 17: PCA projection of the latent space colored by Instability Score

The results show that linear models (Linear Regression, Ridge, Lasso) performed best, with near-perfect R² scores, suggesting a strong linear relationship in the data—but also raising concerns about overfitting or data leakage. Tree-based models (Random Forest, Decision Trees, Gradient Boosting) had mixed performance, sometimes scoring well but also showing high error and instability, likely due to sensitivity to hyperparameters or feature selection issues. (Table 4)

The wide range of results highlights the risk of blindly testing a large number of models (100-1000) without proper validation. Some models perform well by chance rather than true predictive power. To mitigate overfitting and instability, cross-validation, hyperparameter tuning, and feature selection are crucial. While testing many models helps identify strong candidates, efficient selection and proper validation matter more than force experimentation like training 100 models.

Using an autoencoder for this part was helpful since it reduced the high-dimensional feature space into a more compact representation, helping models focus on the most relevant patterns rather than redundant data. It also reduced time consumption by compressing the data in an optimal way. The Training Loss gradually decreases over epochs and does not decrease significantly at the start of the model, which is what we want to see to avoid overfitting on the training data. (Figure 16)