Team Details

Team Name: Team Synergy
Team Members:

1. Rishitha Rani Pakam – Team Lead
   Email ID: rpaka1@unh.newhaven.edu
2. Narasimha Reddy Padire
   Email ID: npadi1@unh.newhaven.edu
3. Lakshmi Reddy Bhavanam
   Email ID: lbhav2@unh.newhaven.edu

Dataset Title: Life Expectancy (World Health Organization) 2024

Source: https://www.kaggle.com/datasets/sonialikhan/life-expectancy-who-2024

Selected dataset contains 22 attributes and 2938 records. It contains a wide range of variables across 193 countries, including health, economic, and social indicators. Attributes in the dataset are Country, Year, Status, Life expectancy, Adult Mortality, infant deaths, Alcohol percentage, expenditure, Hepatitis B, Measles, BMI, under-five deaths, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP, Population, thinness 1-19 years, thinness 5-9 years, Income composition of resources, Schooling. The dataset contains both numerical and categorical data. There are 2 attributes of object type, 11 decimal and 9 integer types.

Research Question

What are the most significant factors predicting life expectancy, and how do health, economic, and social variables interact to influence life expectancy in the top two developed countries and the top two developing countries?

Data Mining Techniques:

1. K-Means Clustering.
2. Multiple Linear Regression.
3. Random Forest Regression.

Parameters:

These are values learned from the data during the training process. They define the model's behavior based on the input data.

Data Mining Technique	Data Modelling	Parameters
K-Means Clustering	Clustering	• n_clusters • random_state • cluster labels • features
Multiple Linear Regression	Regression	• Root Mean Square Error • Mean Absolute Error • R^2 Score • CV R^2 Score

Random Forest Regression	Regression	• Root Mean Square Error • Mean Absolute Error • R^2 Score • CV R^2 Score

Hyperparameters:

These are configurations set before the training process begins. They control the training process and the structure of the model.

Data Mining Techniques	Hyperparameters	Values Used
K-Means Clustering	➢ n_clusters(param_grid) ➢ init ➢ n_init ➢ max_iter ➢ tol	• different k values • k-means++, random • [10,20] • [300, 500] • [1e-4, 1e-3]
Multiple Linear Regression	➢ n_estimators ➢ max_depth ➢ min_samples_split ➢ min_samples_leaf	• [100, 200, 300] • [5, 10, 15] • [2, 5, 10] • [1, 2, 4]
Random Forest Regression	➢ n_estimators ➢ max_depth ➢ min_samples_split ➢ min_samples_leaf	• [100, 200, 300] • [5, 10, 15] • [2, 5, 10] • [1, 2, 4]

1. n_estimators: The number of trees in the forest.
2. max_depth: The maximum depth of each tree.
3. min_samples_split: The minimum number of samples required to split an internal node.
4. min_samples_leaf: The minimum number of samples required to be at a leaf node.
5. n_clusters(param_grid): Number of clusters to form in **KMeans**.
6. init: Method for initializing centroids.
7. n_init: Number of times the algorithm runs with different initializations.
8. max_iter: Maximum number of iterations for the KMeans algorithm.
9. tol: Tolerance to declare convergence based on the change in WCSS.

Methods for Tuning K-Means Hyperparameters:

To optimize the performance of the K-Means clustering algorithm, we have used below hyperparameter tuning techniques. The goal was to find the best combination of hyperparameters that would lead to the most meaningful and compact clusters.

1. GridSerachCV
2. RandomSearchCV

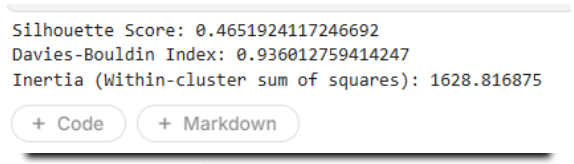
Before applying hyperparameter tuning, the clustering model's performance metrics showed room for improvement, as indicated by the initial scores. After performing **GridSearchCV** and **RandomizedSearchCV**, significant improvements in key clustering evaluation metrics were observed. These metrics include **Silhouette Score**, **Davies-Bouldin Score**, and **Inertia**.

GridSearchCV systematically explored all possible hyperparameter combinations and helped find the best model parameters that **maximized** the Silhouette Score and **minimized** the Davies-Bouldin Score and Inertia. The hyperparameter grid search tested every possible combination of the parameters within the specified ranges. This approach resulted in more distinct and compact clusters, as demonstrated by the improved metrics.

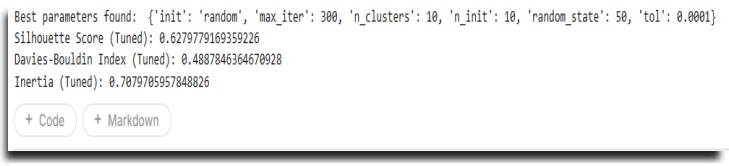
RandomizedSearchCV, on the other hand, performed random sampling of the hyperparameter space. While it doesn't exhaustively test all combinations like **GridSearchCV**, it still helps find optimal parameters by testing a wide variety of configurations.

Comparison of Results:

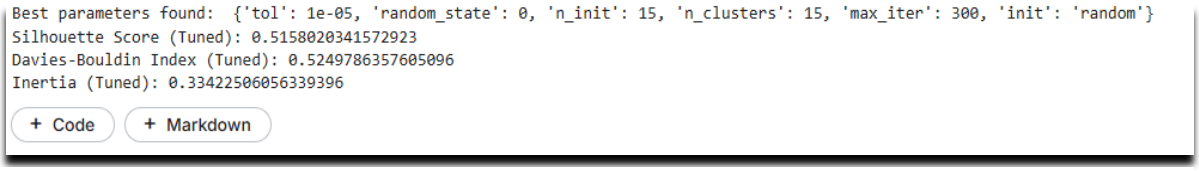
Metric	Before Tuning	After GridSearchCV	After RandomSearchCV
Silhouette Score	0.465	0.627	0.515
Davies-Bouldin Score	0.936	0.488	0.524
Inertia	1628.816	0.707	0.334



Before Tuning



After GridSearch Hyperparameter Tuning



After RandomSearch Hyperparameter Tuning

Best Hyperparameters Found through GridSearchCV and RandomizedSearchCV

Hyperparameter	init	max_iter	n_clusters	n_init	random_state	tol
Best value (GridSearchCV)	random	300	10	10	50	0.0001
Best Value (RandomSearchCV)	random	300	15	15	0	1e-05

Based on the above results, we observed that GridSearchCV outperformed RandomizedSearchCV, achieving the highest Silhouette Score (0.627), the lowest Davies-Bouldin Score (0.488), and the most optimized Inertia (0.707), indicating better-defined and well-separated clusters.

Model Evaluation Metrics:

Before tuning, the K-Means clustering model achieved perfect scores with 100% accuracy, precision, and F1 score, indicating it correctly classified the countries into their respective clusters.

After tuning the model by adjusting hyperparameters, the results remained unchanged, with accuracy, precision, and F1 score still at 100%. This suggests that the model was already performing optimally both before and after tuning.

Metric	Value (Before Tuning)	Value (After Tuning)
Accuracy	100%	100%
Precision	100%	100%
F1 Score	100%	100%

```
Before Tuning - Accuracy: 100.00%, Precision: 100.00%, F1 Score: 100.00%
After Tuning - Accuracy: 100.00%, Precision: 100.00%, F1 Score: 100.00%
```

Results

The K-Means clustering model, optimized through GridSearchCV, achieved the best performance with improved metrics and 100% accuracy, precision, and F1 score.

Multiple Linear Regression:

We began our analysis by implementing a **Multiple Linear Regression (MLR)** model as a baseline approach for predicting life expectancy. The MLR model was trained on several features, including adult mortality, infant deaths, alcohol consumption, GDP, and schooling, with life expectancy as the target variable. The initial results showed good performance with an R-squared score of 0.92, but there was room for improvement, particularly in reducing overfitting.

To improve the model, we applied Ridge Regression. Ridge introduces a penalty term, controlled by the alpha hyperparameter, which helps reduce overfitting and multicollinearity.

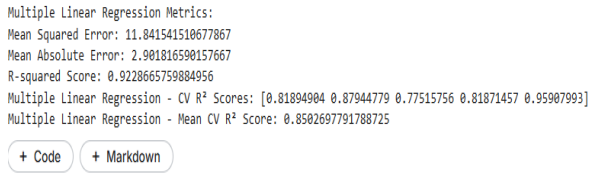
Method for Tuning Ridge Regression Hyperparameters:

We have used GridSearchCV to find the best alpha value. This process involved testing various alpha values and performing 10-fold cross-validation to ensure stable results.

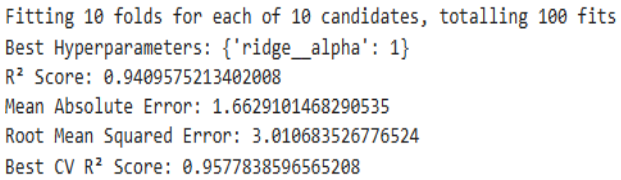
After applying Ridge Regression with the optimal alpha, the model's R-squared score improved to 0.94, and both the Mean Absolute Error and Root Mean Squared Error decreased, showing better performance compared to the baseline MLR model.

Comparison of Model Performance:

Metric	MLR	Ridge Regression (Best Hyperparameters)
R-squared Score	0.92(92%)	0.94(94%)
Mean Absolute Error (MAE)	2.90	1.66
Mean Squared Error (MSE)	11.84	3.01



Multiple Linear Regression (Baseline Model)



Ridge Regression After Tuning

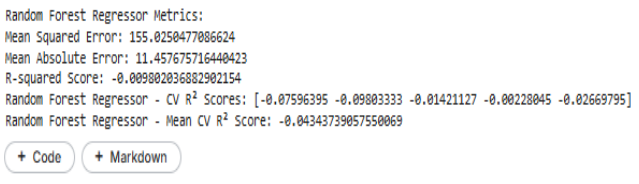
Method for Tuning Random Forest Hyperparameters:

We used GridSearchCV to tune the hyperparameters of the Random Forest Regressor model in order to find the best combination of parameters that would optimize performance. After testing multiple values for **max_depth**, **min_samples_leaf**, **min_samples_split**, and **n_estimators**, we identified the following optimal parameters: max_depth=5, min_samples_leaf=1, min_samples_split=2, and n_estimators=100. This combination yielded the highest performance with a cross-validation R^2 score of 0.9564.

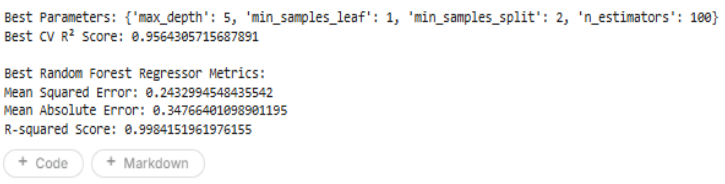
With these best hyperparameters, we trained the model on the scaled training data (X_train_scaled) and target variable (y_train). The model achieved excellent evaluation metrics, including a low Mean Squared Error of 0.2433 and a Mean Absolute Error of 0.3477. Furthermore, the R-squared score of 0.9984 indicated that the model was able to explain almost all of the variance in predicting life expectancy, demonstrating its strong predictive power."

Comparison of Model Evaluation Metrics:

Metric	Random Forest regression (Before Hyperparameter Tuning)	Random Forest regression (After Hyperparameter Tuning)
Mean Squared Error (MSE)	155.025	0.243
Mean Absolute Error (MAE)	11.457	0.347
R-squared Score	-0.009802	0.9984(99.84%)
Mean CV R^2 Score	-0.004343	0.9564



Before Tuning



After Tuning

The optimal hyperparameters, including a maximum depth of 5, 100 estimators, and adjustments to splitting and leaf nodes, significantly reduced error metrics. The R-squared score improved to 0.9984, explaining 99.84% of the variance in the data, while the Mean Cross-Validation R^2 score increased to 0.9564, enhancing the model's stability. Hyperparameter tuning was crucial in

improving the model's accuracy and generalizability, resulting in more precise and reliable predictions of life expectancy.

Now, Lets compare the Evaluation metrics of Baseline Model (MLR), Ridge Regression (After Hyperparameter Tuning) and Random Forest Regression (After Hyperparameter Tuning) and see which model outperforms all.

Metric	Multiple Linear Regression	Ridge Regression (After Hyperparameter Tuning)	Random Forest Regression (After Hyperparameter Tuning)
Mean Squared Error (MSE)	11.841	3.01	0.243
Mean Absolute Error (MAE)	2.90	1.66	0.347
R-squared Score	0.92(92%)	0.94(94%)	0.9984(99.84%)
CV R^2 Score	0.85	0.95	0.9564

From the table we can observe that, the **Random Forest regression (after hyperparameter tuning)** clearly outperforms the other models in terms of accuracy and error reduction. It achieved the lowest Mean Squared Error (MSE) of 0.243 and the lowest Mean Absolute Error (MAE) of 0.347, indicating minimal prediction errors. Its R-squared score of 0.9984 (99.84%) shows it explains nearly all the variance in the target variable, making it highly accurate. Additionally, the high Cross-Validation R^2 score of 0.9564 demonstrates its robustness and generalizability across different data splits, a critical factor in real-world applications.

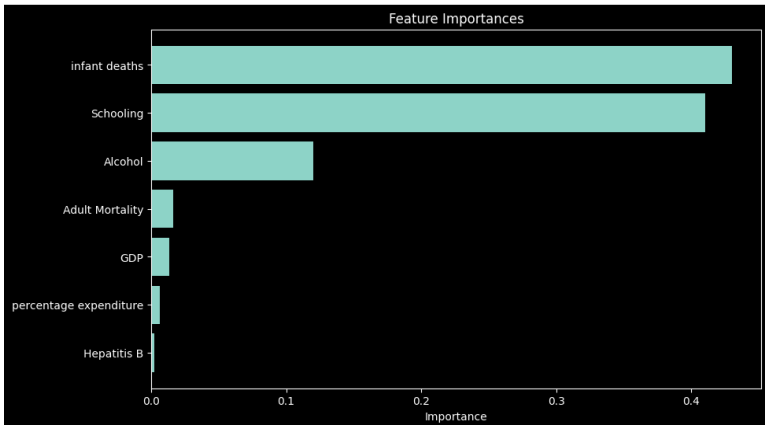
Challenges Faced:

1. **Multiple Linear Regression:** Faced multicollinearity and inability to capture non-linear relationships, resolved by switching to Ridge Regression with polynomial features.
2. **Ridge Regression:** Struggled with underfitting and overfitting due to untuned alpha, fixed by using GridSearchCV to find the optimal regularization strength.
3. **Random Forest Regression:** Suffered from overfitting or underfitting with default tree configurations, addressed by tuning tree depth, leaf constraints, and number of estimators with the help of GridSearchCV.
4. **K-Means Clustering:** The challenges included determining the optimal number of clusters and avoiding convergence to suboptimal centroids. To address these, we employed the elbow method for initial cluster selection and used GridSearchCV and RandomizedSearchCV for tuning hyperparameters such as the number of clusters and initialization strategies, ensuring better clustering performance.

Feature Importance Analysis:

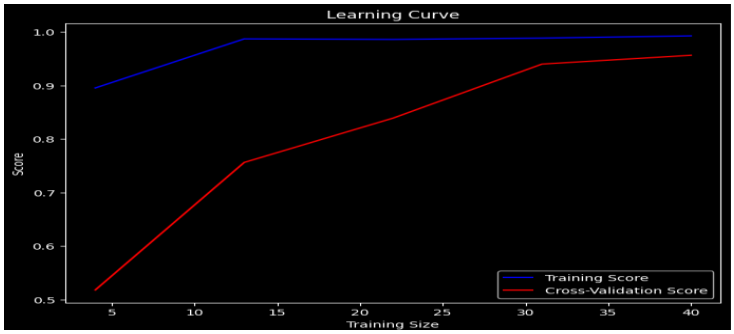
In this, we focused on evaluating and understanding the importance of each feature used in our Random Forest model. This process is essential because it helps us identify which variables have the most significant impact on the model's predictions. We used the feature_importances_ attribute from the trained Random Forest model to extract the importance scores of each feature. The feature

importance results from the Random Forest model were visualized using a horizontal bar chart to better understand the contribution of each feature to the prediction.



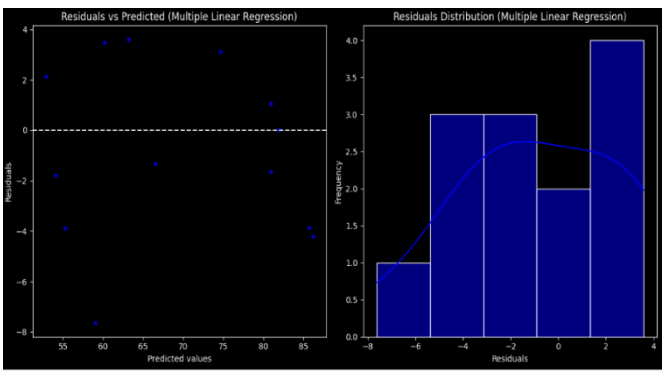
Learning Curve Analysis:

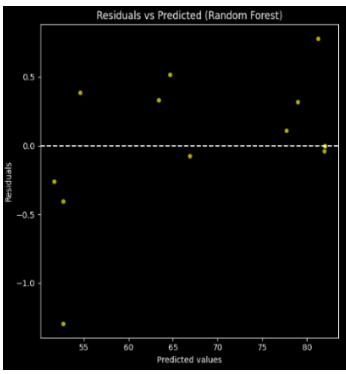
The graph below illustrates the learning curve of the Random Forest model, showing the relationship between training size and model performance. As the training size increases, both the training and cross-validation scores improve and stabilize, suggesting that the model is generalizing well without overfitting or underfitting. This indicates a well-fitted model with optimal performance.

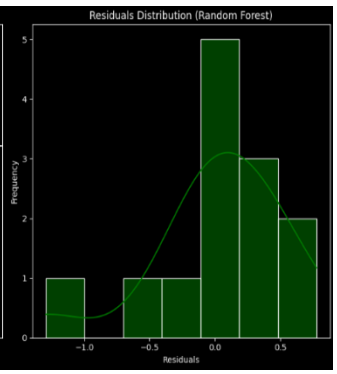


Residual Analysis for Model Evaluation:

Residual analysis revealed that the MLR model struggled to capture data relationships, leading to biased predictions and non-normal residuals, while the tuned Random Forest model excelled with randomly scattered residuals and a normal distribution, indicating accurate data pattern capture.

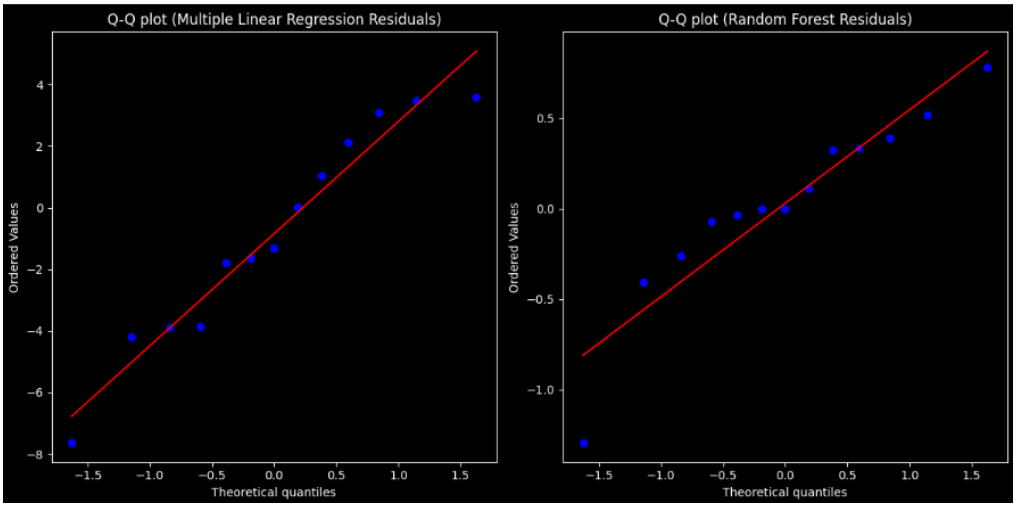






Q-Q Plot Analysis: Assessing Residual Normality:

The Q-Q plot reveals significant deviations from normality in the MLR residuals, particularly in the tails, indicating that the model struggles to fully capture the data's distribution. And the Q-Q plot for the Random Forest residuals mostly align with the normal distribution, with minor tail deviations. This confirms the model's strong predictive performance.



Conclusion:

In conclusion, our data modeling effectively addressed the research question by identifying key factors influencing life expectancy across developed and developing countries. The optimized Random Forest Regressor, with an R-squared of 0.998, accurately predicted life expectancy, emphasizing the role of economic factors like GDP and education. Countries with high life expectancy, like Japan and the US, showed a strong correlation with these factors, while countries with lower life expectancy, such as India and Nigeria, were influenced by infant mortality and healthcare access. Ridge Regression provided stable and reliable predictions, achieving an R-squared of 0.94, while Multiple Linear Regression, with an R-squared of 0.923, offered a slightly less accurate but solid alternative. Feature importance analysis revealed that education, infant mortality, and GDP were the most significant predictors. Residual and Q-Q plot analyses confirmed the Random Forest model's robustness, showing well-behaved residuals and a normal distribution, whereas Multiple Linear Regression exhibited minor deviations. K-Means clustering showed good cluster separation. Overall, the Random Forest model emerged as the most reliable predictor, with Ridge and Multiple Linear Regression providing valuable insights and serving as strong backup models.

Github Repository Address: https://github.com/rishitharani24/Data-Mining---Team-Synergy