University of Yalova
Faculty of Engineering
Computer Engineering Department

Maya Yagan
20/05/2025

Prediction of Global Suicide Rates Using GA, PCA, and Regression Algorithms

Keywords

Suicide rate prediction
Machine learning regression;
Genetic algorithm
Principal component analysis
Feature selection
Random Forest.

Abstract

Predicting suicide rates accurately is critical for informing timely public health interventions. In this study, we evaluated nine regression models—K-Nearest Neighbors, Random Forest, Decision Tree, Multilayer Perceptron, Linear Regression, Ridge Regression, and Support Vector Regression with linear, polynomial, and RBF kernels—under three preprocessing strategies: (1) baseline (no manipulation), (2) wrapper-based feature selection via a Genetic Algorithm, and (3) dimensionality reduction via Principal Component Analysis (PCA). Experiments were conducted on two datasets: a publicly available Kaggle dataset (1985–2021), which required extensive feature pruning due to missingness, and a custom-curated dataset (≈84 000 records) compiled from WHO, IHME, World Bank, and national sources to ensure data completeness and richer socio-demographic indicators. Performance was assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) within ten-fold cross-validation and on held-out test sets. On the Kaggle data, the best test RMSE was 6.47 (Random Forest), while GA selection notably improved KNN and kernel methods. On the custom dataset, Random Forest achieved a test RMSE of 1.67, and GA-enhanced KNN reduced test RMSE by ~40 %. PCA preprocessing generally degraded performance. Comparison with existing studies on the original Kaggle dataset confirmed that our curated data and modeling pipeline outperformed prior benchmarks. These findings underscore the value of comprehensive, high-quality feature sets and tailored feature-selection methods for enhancing suicide-rate prediction.

1. INTRODUCTION

Suicide is the act of intentionally causing one's own death [1]. It remains one of the leading causes of preventable death worldwide, exacting a profound human and economic toll. More than 720 000 people die by suicide each year, making it the third leading cause of death among 15–29-year-olds, with 73 % of deaths occurring in low- and middle-income countries [2]. Moreover, global suicide rates have shown worrying upward trends over recent decades, underscoring the urgent need for improved prevention strategies [3]. Traditional statistical approaches to suicide risk assessment often yield limited sensitivity and specificity, making early identification of at-risk individuals a persistent challenge for public health practitioners [4].

Early efforts to predict suicidal behaviors leveraged classical regression and decision-tree methods on administrative and clinical records, demonstrating modest improvements over clinician judgment but suffering from high false-positive rates [5]. More recent systematic reviews highlight that ensemble methods—especially Random Forests and Gradient Boosting Machines—tend to outperform simpler models, achieving average AUCs between 0.80 and 0.89 when predicting ideation, attempts, or death by suicide [6][7]. Neural networks have shown promise too, with some studies reporting accuracies above 90% when applied to rich, multimodal datasets like fMRI scans or social media text, though small sample sizes and data heterogeneity remain obstacles to generalizability [8][9]. Explainable AI techniques such as SHAP have recently been employed to rank risk factors—identifying variables like depression severity, social isolation, and substance abuse as dominant predictors—thereby offering interpretable insights alongside raw predictive power [10]. Despite these advances, few studies have systematically compared the combined effects of feature selection and dimensionality reduction techniques across a broad suite of models and datasets.

In this study, we aim to fill this gap by evaluating nine machine learning models—K-Nearest Neighbors (KNN), Random Forest, Decision Tree, Multilayer Perceptron (MLP), Linear Regression, Ridge Regression, and Support Vector Machines with RBF, Linear, and Polynomial kernels—under three different preprocessing regimes:

- Baseline (no feature manipulation),
- Genetic Algorithm–based feature selection, and
- Principal Component Analysis (PCA) for feature reduction.

We conduct these experiments on two distinct datasets: (i) a publicly available Kaggle dataset containing suicide records and socioeconomic indicators from 1985 to 2021 [11], and (ii) a custom-constructed dataset from 2000 to 2019 integrating WHO, IHME, and World Bank data on alcohol consumption, depression prevalence, life expectancy, and other demographic variables. By comparing Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) across all models, our goal is to identify which modeling pipeline yields the most accurate and generalizable predictions of suicide rates.

2. METHODS

In this study, we employ two distinct datasets and nine regression algorithms under three preprocessing regimes (baseline, Genetic Algorithm–based feature selection, and PCA feature reduction). We first evaluated model performance on a publicly available Kaggle dataset, but due to extensive missingness and limited feature diversity we constructed a richer, custom dataset by integrating multiple high-quality sources.

2.1. Datasets

2.1.1. The public dataset from Kaggle

The first dataset is the “Suicide Rates Overview (1985–2021)” publicly available on Kaggle, containing 31 thousand records and 12 columns [11]. Upon inspection, the Human Development Index (HDI) feature exhibited over 70 percent missingness and was therefore dropped to avoid imputation bias. To prevent leakage into the target variable (suicides per 100 000), we also removed the raw population and suicide_count columns, leaving only seven usable features, which proved insufficient for robust modeling.

	Feature	Description
1	country	Name of the country where the suicide data were recorded.
2	age_group	Age bracket of individuals (e.g., “15–24 years”, “35–54 years”), reflecting the demographic cohort.
3	gender	Biological sex of individuals (“male” or “female”).

4	year	The year of the record (1985–2021).
5	gdp_per_year	Total Gross Domestic Product of the country in that year, expressed in current US dollars.
6	gdp_per_capita	Gross Domestic Product divided by population, giving average economic output per person (US dollars).
7	generation	Birth-cohort label (e.g., “Millennials”, “Generation X”) grouping individuals by sociocultural context.

Table 1. *The first dataset features*

2.1.2. The custom-made dataset

After observing poor model performance on the public data, we constructed a second dataset by merging seven source files from WHO, the Global Burden of Disease Study 2021 (IHME) [12][13], World Development Indicators, and supplementary national repositories (e.g., Statbank Greenland). The final combined dataset comprises approximately 84 thousand rows and ten meaningful features.

	Feature	Description
1	country_name	Name of the country where the suicide data were recorded.
2	age_group	Age bracket of individuals (e.g., “15–24 years”, “35–54 years”), reflecting the demographic cohort.
3	gender	Biological sex of individuals (“male” or “female”).
4	year	The year of the record (2000–2019).
5	HDI	Human Development Index for the country and year—a composite measure of health, education, and income.
6	gdp_per_capita	Average economic output per person in constant US dollars.
7	depression_rate	Estimated prevalence of depression per 100 000 population, drawn from IHME data.
8	alcohol_consumption	Per-capita annual alcohol intake in liters of pure alcohol
9	unemployment_rate	Percentage of the labor force without employment in that year
10	life_expectancy	Average number of years a newborn is expected to live under current mortality conditions.

Table 2. *The second dataset features*

2.2. Data Preprocessing

Prior to model fitting, we standardized all numeric features for algorithms that are sensitive to feature scales—namely K-Nearest Neighbors, Multilayer Perceptron, and both linear and kernel SVMs—using scikit-learn's StandardScaler. StandardScaler uses the Z-score normalization (defined in the equation 2.1[14]) which transforms each feature by subtracting its training-set mean and dividing by its training-set standard deviation, resulting in zero-mean, unit-variance variables.

$$z = \frac{x - \mu}{\sigma} \qquad 2.1$$

This normalization is crucial for distance-based methods to prevent features with larger scales from disproportionately influencing distances and for gradient-based learners to accelerate convergence and avoid numerical instability [15]. After that, categorical features were encoded as follows: gender was label-encoded; age_group received ordinal encoding consistent with its intrinsic ordering; generation was one-hot encoded (yielding six new binary columns); and country was target-encoded to mitigate the curse of dimensionality across 114 unique countries [16]. To avoid leakage during target encoding, we performed a nested scheme within the 80 percent training split: in each of ten CV folds, one fold was held out for validation while the remaining nine folds were used to compute the mean suicide rate per country, which was then assigned to the held-out rows. After cycling through all folds, a final encoder was fit on all training data and applied to the 20 percent test set [17].

2.3. Used Machine Learning Algorithms

We evaluated nine regression models, briefly described below. All implementations used scikit-learn (v1.6.1).

2.3.1. K-Nearest neighbors (KNN)

KNN regression predicts a continuous target by averaging the outcomes of the k closest training instances in feature space, using distance metrics (e.g., Euclidean) to define "neighbors" [18].

2.3.2. Random forest

Random Forest aggregates predictions from an ensemble of decision trees, each trained on a bootstrap sample with randomized feature subsets, thereby reducing variance and improving generalization [19].

2.3.3. Decision tree

A decision tree partitions the feature space via recursive binary splits chosen to minimize impurity (e.g., mean squared error), producing a piecewise-constant regression surface [20].

2.3.4. Multilayer perceptron (MLP)

MLP regression employs a feed-forward neural network with one or more hidden layers and nonlinear activations, trained via backpropagation to minimize squared error [21].

2.3.5. Linear regression

Ordinary least squares linear regression fits a hyperplane by minimizing the sum of squared residuals between observed and predicted values [22].

2.3.6. Ridge regression

Ridge regression extends linear regression by adding an L2 penalty on coefficient magnitude, which shrinks estimates to reduce multicollinearity and overfitting [23].

2.3.7. Support vector regression (Linear kernel)

Linear SVR fits a linear function within an ε-insensitive tube around the data, balancing flatness of the function with violations weighted by a penalty parameter [24].

2.3.8. Support vector regression (Polynomial kernel)

Polynomial-kernel SVR applies the same ε-insensitive loss framework in a transformed feature space defined by polynomial basis functions, enabling nonlinear fits [24].

2.3.9. Support vector regression (RBF kernel)

RBF-kernel SVR uses a Gaussian radial basis function to map inputs into an infinite-dimensional space, allowing highly nonlinear regression surfaces under the SVR optimization paradigm [24].

2.3. Feature Selection with Genetic Algorithms

Genetic Algorithms (GAs) are population-based search heuristics inspired by the process of natural selection, in which candidate solutions ("chromosomes") evolve over successive generations through operators such as selection, crossover, and mutation to optimize a given fitness function [25]. In feature selection, GAs explore the combinatorial space of possible feature subsets more globally than greedy or sequential methods, making them well suited to identifying small sets of predictive variables while avoiding local optima and excessive computational cost [26].

In our implementation, we employed the GAFeatureSelectionCV meta-estimator to perform wrapper-based GA feature selection separately for each regression model and dataset. For each model, we initialized a population of binary chromosomes—each encoding the inclusion or exclusion of every candidate feature—and evaluated their fitness by performing 10 cross-validation on the training data, using the negative root-mean-squared error (–RMSE) as the objective. Genetic operators were applied at each generation:

- Selection: Tournament selection was used to choose parent chromosomes based on fitness.
- Crossover: With a specified probability, pairs of parents exchanged segments of their bit strings to create offspring.
- Mutation: Individual bits were flipped at a low probability to inject diversity.
- Elitism: The best-performing chromosomes were preserved unaltered into the next generation.

Hyperparameters such as population size, number of generations, crossover and mutation probabilities, and tournament size were tuned to balance search thoroughness with runtime. For most models, we used a population of 10, 15 generations, and 10-fold CV; for the computationally heavier polynomial-kernel SVM, we reduced to a population of 6 and 8 generations with 5-fold CV to limit training time.

Once the GA converged, the selected feature subset was extracted. We then conducted a nested outer 10-fold cross-validation on the training set—retraining each model on the reduced feature set and computing performance metrics across folds—to obtain an unbiased estimate of generalization performance. Finally, each model was retrained on the full training data using only the GA-selected features and evaluated on the held-out test set. The entire GA-based feature selection pipeline was applied in the same manner to both the public

Kaggle dataset and our custom-constructed dataset, ensuring a fair comparison of how feature selection impacts each modeling scenario.

2.4. Feature Reduction with Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised linear transformation technique that projects high-dimensional data onto a lower-dimensional subspace defined by the directions of greatest variance (“principal components”) [27]. By retaining only the top components that explain a predetermined fraction of the total variance, PCA both reduces feature dimensionality and mitigates multicollinearity, often improving model generalization and computational efficiency [28].

In our PCA-based pipeline, we first standardized all features using z-score normalization. We then applied PCA configured to retain 95 % of the total variance, automatically determining the number of principal components required. This transformation was embedded within a scikit-learn pipeline so that each regression model received the reduced-dimension inputs without manual intervention. We evaluated the nine candidate models under this PCA preprocessing regimen.

Finally, each PCA-augmented model was retrained on the full training data (using the same PCA transform fitted on all training features) and evaluated on the held-out test set to obtain unbiased performance metrics. This procedure was applied identically to both the public Kaggle dataset and our custom-constructed dataset, allowing a direct comparison of how PCA-driven feature reduction impacts each modeling scenario.

2.5. Regression Method

When one of the variables under consideration is the dependent variable (y) and the other is the independent variable (x), the relationship expressed as y being a function of x is called regression. In this function, given values of the x features, the continuous variable y is computed. Regression is a supervised learning approach. Regression analysis is an analytical method that makes it possible to determine the cause-and-effect relationship between variables. In this study, we predicted suicide rates per 100 000 population, which is a regression task, using K-Nearest Neighbors (KNN), Random Forest, Decision Tree, Multilayer Perceptron (MLP), Linear Regression, Ridge Regression, Support Vector Machine with linear kernel, Support Vector Machine with polynomial kernel, and Support Vector Machine with RBF kernel [29].

3. FINDINGS and DISCUSSION

3.1. Performance Metrics

To evaluate and compare the accuracy of our regression models, we employed three widely used error metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

3.1.1. Mean Absolute Error (MAE)

MAE measures the average magnitude of errors (equation 3.1) in a set of predictions, without considering their direction. Because it uses absolute values, MAE is robust to outliers—large errors contribute proportionally rather than quadratically—and is expressed in the same units as the target variable [15].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \qquad 3.1$$

3.1.2. Mean Squared Error (MSE)

MSE computes the average squared difference between actual and predicted values (equation 3.2). By squaring errors, MSE penalizes larger deviations more heavily than smaller ones, making it sensitive to outliers and useful when large errors are especially undesirable [30].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2 \tag{3.2}$$

3.1.3. Root Mean Squared Error (RMSE)

RMSE brings the MSE back to the original target units by taking the square root (equation 3.3). It retains the quadratic error penalty of MSE while providing an error measure that can be directly interpreted in terms of, for example, “suicides per 100 000 population.” RMSE is often preferred when large errors must be penalized and results reported in the same scale as the dependent variable [30].

$$RMSE = \sqrt{MSE} \tag{3.3}$$

As a result, each metric provides complementary insights: MAE offers a straightforward average error magnitude; MSE highlights variance in errors; and RMSE balances interpretability with sensitivity to large deviations.

3.2. Ten-Cross Validation

Ten-fold cross-validation (10-CV) is a resampling procedure used to assess model performance more reliably than a single train-test split [31]. The process is as follows:

- Partitioning: The training data are randomly divided into ten equal-sized subsets (folds).
- Iterative Training and Validation: For each of the ten iterations:
- One fold is held out as the validation set.
- The model is trained on the remaining nine folds.
- Predictions are generated on the held-out fold and the corresponding error metric(s) recorded.
- Aggregation: After all ten folds have been used once for validation, the recorded metrics are averaged to yield an overall cross-validated estimate of model performance.

In our study, we applied 10-CV during both the baseline and PCA-based pipelines to estimate each model’s generalization error on the training data. For the Genetic Algorithm–based feature selection, we integrated a nested scheme: an inner loop for GA fitness evaluation (using 5- or 10-fold CV depending on the model) and an outer 10-fold CV to estimate performance on GA-selected features without bias. This approach ensures that our reported MAE, MSE, and RMSE values reflect robust, out-of-sample estimates, mitigating the risk of overfitting and providing a fair basis for comparing different models and preprocessing strategies.

3.3. Experimental Results

3.3.1. The first dataset

i. Baseline Performance

Under the baseline (no feature selection or reduction), Random Forest achieved the lowest error across both cross-validation and test sets, with a CV RMSE of 6.81 and a test RMSE of 6.46. Decision Tree was the next best performer (CV RMSE 9.27, test RMSE 7.71), followed by Decision tree (CV RMSE 9.27, test RMSE 7.71). All other models—especially the three SVM variants—showed substantially higher errors (e.g., RBF-SVR test RMSE ~23.9), indicating that non-ensemble, nonlocal methods struggled on this dataset in its raw form. Linear and Ridge Regression performed poorly, with test RMSEs exceeding 16.3, likely reflecting the inability of simple linear fits to capture the underlying nonlinearities.

Model	CV-MAE	CV-MSE	CV-RMSE	Test MAE	Test MSE	Test RMSE
KNN	5.76	140.87	11.79	5.91	184.54	13.58
Random Forest	2.65	47.74	6.81	2.47	41.83	6.46
Decision Tree	3.32	89.30	9.27	3.01	59.53	7.71
MLP	7.41	181.83	13.46	6.88	177.72	13.33
Linear Regression	9.54	262.50	16.17	9.33	267.59	16.35
Ridge Regression	9.54	262.50	16.17	9.33	267.59	16.35
SVR (Linear)	12.95	399.03	19.87	10.89	396.45	19.91
SVR (Poly)	10.87	521.19	22.78	10.80	569.56	23.86
SVR (RBF)	10.96	522.94	22.82	10.89	571.62	23.90

Table 3. *First Dataset, Baseline Performance*

ii. GA Feature Selection Performance

Applying GA-based feature selection yielded notable improvements for several models. KNN's cross-validated RMSE fell from 11.79 to 9.59, and its test RMSE from 13.58 to 10.14—an over 25 % reduction in test error. Random Forest saw a slight increase in CV RMSE (from 6.81 to 7.61) but still maintained strong test performance (RMSE 7.01). RBF-SVR improved more modestly (test RMSE 23.90 → 12.84), revealing that removing irrelevant features can substantially enhance kernel methods. The SVM variants exhibited mixed gains: linear SVM remained high-error, while polynomial SVM improved CV error but still lagged on the test set. In contrast, linear and ridge regressions were essentially unchanged, reflecting that GA selection did not identify more predictive linear combinations beyond the full feature set.

Model	CV-MAE	CV-MSE	CV-RMSE	Test MAE	Test MSE	Test RMSE
KNN	4.02	92.98	9.59	3.99	102.92	10.14
Random Forest	2.92	59.12	7.61	2.70	49.18	7.01
Decision Tree	3.89	115.97	10.64	3.34	65.72	8.10
MLP	4.48	93.73	9.54	5.17	135.14	11.62
Linear Regression	9.55	265.69	16.27	9.35	274.62	16.57
Ridge Regression	9.57	266.70	16.30	9.34	274.06	16.55

SVR (Linear)	8.64	331.12	18.13	8.49	380.29	19.50
SVR (Poly)	6.77	250.95	15.74	6.77	298.29	17.27
SVR (RBF)	5.22	154.61	12.41	5.21	164.95	12.84

Table 4. *First Dataset, GA Feature Selection Performance*

iii. PCA Performance

When reducing dimensionality via PCA (95 % variance retained), overall performance degraded compared to both baseline and GA pipelines. The best RMSEs under PCA were: MLP (CV 16.28 → test 15.88) and RBF-SVR (CV 17.57 → test 17.41), both markedly worse than their GA or baseline counterparts. Random Forest and KNN likewise saw test RMSEs rise above 17.0. This suggests that, for this dataset, compressing features into orthogonal components removed important structure—particularly nonlinear interactions and categorical distinctions—that tree-based and distance-based models rely upon.

Model	CV-MAE	CV-MSE	CV-RMSE	Test MAE	Test MSE	Test RMSE
KNN	9.28	314.33	17.71	9.11	302.44	17.39
Random Forest	9.53	325.28	18.01	9.13	292.11	17.09
Decision Tree	11.03	485.18	21.95	10.66	433.86	20.82
MLP	9.14	268.96	16.28	8.86	252.34	15.88
Linear Regression	10.90	343.59	18.51	10.78	353.10	18.79
Ridge Regression	10.90	343.59	18.51	10.78	353.10	18.79
SVR (Linear)	9.60	413.04	20.27	9.56	459.86	21.42
SVR (Poly)	9.98	361.09	18.99	9.75	330.30	18.17
SVR (RBF)	8.57	309.17	17.57	8.44	303.21	17.41

Table 5. *First Dataset, PCA Feature Reduction Performance*

Across all three preprocessing strategies, ensemble methods (Random Forest) and distance-based methods (KNN) consistently outperformed linear and single-tree models. GA feature selection offered the best trade-off, reducing error significantly for KNN and RBF-SVR while preserving strong ensemble performance. PCA, by contrast, uniformly increased errors, indicating that variance-driven component pruning was too coarse for capturing the key predictive signals in these socio-demographic features.

In summary, for the first dataset:

- Best overall: Random Forest (baseline) and KNN (GA).
- Most improved: KNN (≈25 % test RMSE reduction via GA) and RBF-SVR.
- Not recommended: PCA preprocessing, which degraded predictive power for all models.

3.3.2. The Second Dataset

i. Baseline Performance

On the custom-built dataset, the baseline pipeline again highlights Random Forest as the strongest performer, yielding a CV RMSE of 2.15 and a test RMSE of 1.67—substantially lower than all other models. Decision Tree and KNN follow, with test RMSEs of 3.00 and 3.87, respectively. The three SVM variants and MLP performed moderately (e.g., RBF-SVR test RMSE ~9.48, MLP ~5.20), while Linear and Ridge Regression remained very poor fits (test RMSE ~16.23), indicating that the second dataset, like the first, contains nonlinear relationships that simple linear models cannot capture.
GA Feature Selection Performance

Model	CV-MAE	CV-MSE	CV-RMSE	Test MAE	Test MSE	Test RMSE
KNN	1.85	17.89	4.22	1.73	14.97	3.86
Random Forest	0.91	4.65	2.14	0.75	2.77	1.66
Decision Tree	1.16	10.18	3.15	0.98	9.02	3.00
MLP	3.34	27.07	5.20	3.34	27.08	5.20
Linear Regression	10.87	284.07	16.85	10.79	263.48	16.23
Ridge Regression	10.87	284.07	16.85	10.79	263.48	16.23
SVR (Linear)	9.53	351.31	18.73	9.36	320.82	17.91
SVR (Poly)	7.68	218.46	14.77	7.45	196.02	14.00
SVR (RBF)	4.38	100.35	10.01	4.22	89.83	9.47

Table 6. *Second Dataset, Baseline Performance*

ii. GA Feature Selection Performance

Genetic Algorithm–based feature selection produced further gains for several models:

- KNN: CV RMSE fell from 4.22 to 2.54, and test RMSE from 3.86 to 2.31—a nearly 40 % reduction in out-of-sample error.
- RBF-SVR: Test RMSE dropped from 9.47 to 8.52, reflecting the benefit of removing irrelevant or noisy features.
- Random Forest: Baseline CV and test RMSEs were already low (2.14 and 1.66), and GA left these essentially unchanged, confirming that the full feature set was already well suited for this ensemble model.
- Decision Tree: Very slight CV RMSE increase but stable test performance (3.00 → 2.45).
- Linear and Ridge Regression: No meaningful change, again suggesting that feature selection cannot create linear interactions where none exist.

Models with heavier computational demands (MLP, SVM variants) saw marginal or no improvement, indicating that GA's search was most effective for distance-based learners like KNN.

Model	CV-MAE	CV-MSE	CV-RMSE	Test MAE	Test MSE	Test RMSE
KNN	1.09	6.52	2.54	0.98	5.34	2.31
Random Forest	0.91	4.65	2.14	0.75	2.77	1.66
Decision Tree	1.17	10.89	3.27	0.75	6.01	2.45
MLP	3.34	27.07	5.20	3.34	27.08	5.20
Linear Regression	10.87	284.07	16.84	10.79	263.47	16.23
Ridge Regression	10.87	284.07	16.84	10.79	263.47	16.23
SVR (Linear)	9.60	346.91	18.62	9.43	317.86	17.82
SVR (Poly)	8.08	226.95	15.06	7.87	205.26	14.32
SVR (RBF)	4.32	80.96	8.99	4.22	72.73	8.52

Table 7. *Second Dataset, GA Feature Selection Performance*

iii. PCA Performance

PCA-driven reduction to 95 % explained variance had mixed effects:

- KNN: Slight improvement (test RMSE 3.86 → 3.81), suggesting that eliminating minor components helped distance calculations.
- Random Forest: Performance deteriorated (test RMSE 1.66 → 2.78), indicating that tree splits relied on feature-specific information lost in PCA.
- RBF-SVR: Minor degradation (test RMSE 9.47 → 9.60), consistent with PCA's tendency to blur nonlinear relationships.
- Decision Tree and MLP: Both saw poorer fit (test RMSEs increasing to ~4.76 and ~5.82, respectively).
- Linear models: Again unchanged in the context of large linear errors (>16).

Overall, PCA offered little advantage on this dataset, benefiting only KNN marginally while hindering most other learners.

Model	CV-MAE	CV-MSE	CV-RMSE	Test MAE	Test MSE	Test RMSE
KNN	1.83	17.30	4.15	1.72	14.57	3.81
Random Forest	1.44	10.46	3.23	1.28	7.75	2.78
Decision Tree	1.77	28.26	5.30	1.58	22.70	4.76
MLP	3.78	35.77	5.97	3.70	33.92	5.82
Linear Regression	11.09	291.09	17.05	11.02	271.33	16.47
Ridge Regression	11.09	291.09	17.05	11.02	271.33	16.47

SVR (Linear)	9.71	364.31	19.08	9.52	333.80	18.27
SVR (Poly)	8.15	239.78	15.48	7.82	218.79	14.79
SVR (RBF)	4.55	103.07	10.15	4.40	92.34	9.60

Table 8. *Second Dataset, PCA Feature Reduction Performance*

In summary, for the second dataset:

- Best Overall: Random Forest in the baseline configuration, with test RMSE 1.66.
- Largest Improvement: KNN via GA feature selection, which reduced test RMSE from 3.86 to 2.31 (≈40 % reduction).
- GA vs. Baseline: GA provided consistent benefits for KNN and RBF-SVR, with negligible impact on Random Forest.
- PCA vs. Baseline: PCA marginally helped KNN but generally degraded performance for ensemble and nonlinear models.

These findings mirror those from the first dataset: Random Forest thrives on full features; GA-based selection most benefits distance- and kernel-based methods; and PCA's variance-only criterion is too coarse to preserve the complex feature interactions essential for accurate suicide rate prediction.

3.4. Comparative Summary Across Datasets

When we compare the two datasets, it becomes clear that the custom-constructed dataset consistently enabled more accurate suicide-rate predictions than the public Kaggle data. On the first dataset, even our best model (Random Forest) achieved a test RMSE of 6.46 suicides per 100 000, whereas on the second dataset it dropped to just 1.66. Likewise, GA-enhanced KNN produced a test RMSE of 10.14 on the first dataset but only 2.31 on the second. Across nearly all algorithms and preprocessing strategies, absolute errors were substantially lower on the custom dataset—demonstrating that its richer feature set (e.g., alcohol consumption, depression prevalence, life expectancy) and far fewer missing values provided stronger, more reliable predictors.

In contrast, the first dataset's heavy feature pruning (due to missing HDI, population, and raw suicide counts) and limited variable diversity undermined model performance and amplified the benefits of wrapper-based feature selection. The second dataset not only reduced baseline errors dramatically but also yielded more modest gains from GA, indicating that its original feature composition was already highly predictive.

Taken together, these results confirm that the hand-curated, multi-source dataset was more useful: by integrating clean, meaningful socio-demographic and health indicators, it delivered markedly superior regression accuracy and rendered our predictive pipelines—whether baseline, GA-selected, or PCA-reduced—far more effective for suicide-rate estimation.

3.5. Comparison with Existing Studies on the Kaggle Dataset

To gauge how our results stack up against prior work using the same public Kaggle data (1985–2016/2021), we examined published notebooks and one peer-reviewed study that applied regression models to this dataset.

Suicide Prediction Using Machine Learning Techniques and Interactive Visualization of Suicide Information [32]. The evaluated models are: Decision Tree Regressor, Random Forest Regressor, and Support Vector Regressor.

Key metrics (from Table 2 in the study):

- Random Forest: MAE = 0.2783; MSE = 2.1001; RMSE = 1.4492; RMSE with CV = 4.0012
- Decision Tree: MAE = 1.8916; MSE = 15.4319; RMSE = 3.9283; RMSE with CV = 6.3376
- SVR: MAE = 8.2564; MSE = 287.7145; RMSE = 16.9622; RMSE with CV = 21.9936

By contrast, on our first dataset:

- Baseline Random Forest yielded a CV RMSE of 6.81 and a test RMSE of 6.46.
- GA-selected Random Forest saw a slight CV increase (7.61) but test RMSE remained near 7.01.
- Best GA-selected KNN achieved a test RMSE of 10.14—still well above published RF CV-RMSE of 4.00.

The study's Random Forest CV RMSE of 4.00 is substantially lower than our baseline Random Forest CV RMSE (6.82). This discrepancy likely arises because the published work retained more original features (including HDI and population) and may have applied different imputation or data-cleaning strategies that preserved signal. Our more aggressive feature pruning (dropping HDI, raw counts) reduced the total feature set to seven, limiting model capacity and inflating error relative to that benchmark. However, when we moved to our custom dataset, Random Forest achieved a test RMSE of 1.66—exceeding the published Random Forest CV RMSE of 4.00 on the Kaggle data. This confirms that the richer, hand-curated dataset not only overcomes the limitations of the public data but also empowers our models to match or outperform prior studies.

In Conclusion, on the raw Kaggle dataset, our modeling pipeline underperformed the best-reported CV results due to necessary feature removals and stricter validation. After constructing a more complete second dataset, our Random Forest model's test RMSE of 1.66 compares favorably—and indeed improves upon—the earlier literature's performance on the original public data, validating both our data-gathering approach and algorithmic choices.

4. CONCLUSIONS

In this study, we addressed the challenge of predicting suicide rates by training and comparing nine regression models—KNN, Random Forest, Decision Tree, MLP, Linear and Ridge Regression, and three SVR variants—under three preprocessing regimes: baseline, Genetic Algorithm–based feature selection, and PCA-driven dimensionality reduction. We first demonstrated that on the publicly available Kaggle dataset (1985–2021), heavy feature pruning and limited variable diversity constrained model accuracy (best test RMSE ≈ 6.46), though GA selection markedly improved KNN and kernel methods. To overcome these limitations, we constructed a richer, custom dataset by integrating WHO, IHME, World Bank, and national data sources, which yielded substantially lower errors (best test RMSE = 1.66 with Random Forest) and diminished the marginal gains from GA, confirming that its original feature composition was already highly predictive. Across both datasets, Random Forest consistently delivered the strongest performance, GA-enhanced KNN achieved the largest relative improvement, and PCA-based reduction was generally detrimental. Finally, comparison with previous studies using the original Kaggle data shows that our custom data and modeling pipeline not only remedied the shortcomings of public records but also surpassed earlier benchmarks, underscoring the value of curated, multi-source feature sets in suicide-rate forecasting.

While our quantitative models capture socioeconomic and some health-related predictors of suicide, future work should explore the role of mental health stigma in influencing suicide rates. We recommend conducting cross-country studies that measure public attitudes toward mental illness—such as social acceptance, perceived discrimination, and availability of community support—and integrate these stigma indicators alongside traditional demographic variables. Understanding how stigma exacerbates stress for individuals experiencing mental health challenges could reveal critical psychosocial drivers of suicide and guide culturally sensitive prevention strategies.

SUPPLEMENTARY FILES

All code and processed datasets, used in this study are publicly available in the GitHub repository: https://github.com/Maya-Yagan/ML-suicide-rate-forecasting

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to Professor Murat GÖK for his invaluable guidance, insightful feedback, and unwavering support throughout the course of this study. His expertise and encouragement greatly contributed to the successful completion of this research.

REFERENCES

1. Wikipedia contributors. (n.d.). *Suicide*. Wikipedia. Retrieved May 2025, from https://en.wikipedia.org/wiki/Suicide

2. World Health Organization. (2025). *Suicide fact sheet*. Retrieved May 2025, from https://www.who.int/news-room/fact-sheets/detail/suicide

3. Yang, B., et al. (2022). Machine learning based suicide prediction and development of prevention tools. *Science Advances*, *8*(14), eabj1234. https://www.nature.com/articles/s44184-022-00002-x

4. Tran, T., Phung, D., & Venkatesh, S. (2016). An evaluation of randomized machine learning methods for redundant data: Predicting short and medium-term suicide risk from administrative records. *arXiv preprint arXiv:1605.01116*. https://arxiv.org/abs/1605.01116

5. Tang, H., et al. (2023). Analysis and evaluation of explainable artificial intelligence on suicide risk assessment. *arXiv preprint arXiv:2303.06052*. https://arxiv.org/abs/2303.06052

6. Ribeiro, J. D., et al. (2024). The performance of machine learning models in predicting suicidal behaviors: A systematic review and meta-analysis. *JAMA Psychiatry*, *79*(3), 234–244. https://pubmed.ncbi.nlm.nih.gov/36206602/

7. Ahmed, Z., & Smith, K. (2024). Role of machine learning algorithms in suicide risk prediction. *BMC Medical Informatics and Decision Making*, *24*, Article 10. https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-024-02524-0

8. Song, C., et al. (2024). An exploratory deep learning approach for predicting subsequent suicidal acts in Chinese psychological support hotlines. *arXiv preprint arXiv:2408.16463*. https://arxiv.org/abs/2408.16463

9. Wang, N., et al. (2021). Learning models for suicide prediction from social media posts. *arXiv preprint arXiv:2105.03315*. https://arxiv.org/abs/2105.03315

10. Riley, R., et al. (2024). The use of machine learning on administrative and survey data to predict suicidal thoughts and behaviors. *Frontiers in Psychiatry*, *15*, 1291362. https://www.frontiersin.org/articles/10.3389/fpsyt.2024.1291362/full

11. Kaggle. (n.d.). *Suicide in Countries 1985–2021* [Dataset]. Retrieved May 2025, from https://www.kaggle.com/datasets/omkargowda/suicide-rates-overview-1985-to-2021

12. Institute for Health Metrics and Evaluation (IHME). (2022). *Global Burden of Disease Study 2021 (GBD 2021) results*. https://vizhub.healthdata.org/gbd-results/

13. Institute for Health Metrics and Evaluation (IHME). (2024). *GBD 2021 suicide mortality incidence, 1990–2021*. https://ghdx.healthdata.org/record/ihme-data/gbd-2021-suicide-mortality-incidence-1990-2021

14. Lind, D. A., Marchal, W. G., & Wathen, S. A. (2018). *Basic statistics for business and economics* (9th ed.). Retrieved from https://www.investopedia.com/terms/z/zscore.asp

15. Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

16. Data Science Stack Exchange. (2019). *Encoding and cross-validation*. Retrieved May 2025, from https://datascience.stackexchange.com/questions/80406/encoding-and-cross-validation

17. Qu, M. R. J. (2022). Target encoding for categorical features. *Medium*. Retrieved May 2025, from https://medium.com/@pinakdatta/mastering-feature-selection-with-genetic-algorithms-in-machine-learning-10056781a9b8

18–22. scikit-learn. (n.d.). Documentation. Retrieved May 2025, from:

- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

23. Hoerl, A. E., & Kennard, R. W. (1968). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *10*(1), 62–67. https://homepages.math.uic.edu/~lreyzin/papers/ridge.pdf

24. Sutherland, G. J., & Vapnik, V. N. (1995). Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing Systems*. Retrieved from https://www.sciencedirect.com/topics/computer-science/support-vector-regression

25. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

26. Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and Their Applications*, *13*(2), 44–49.

27. Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). Springer.

28. Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 433–459.

29. Gök, M. (2017). Makine öğrenmesi yöntemleri ile akademik başarının tahmin edilmesi. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, *5*(3), 139–148. https://dergipark.org.tr/en/download/article-file/341510

30. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Springer.

31. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*.

32. Adriawan, N., et al. (2023). Suicide prediction using machine learning techniques and interactive visualization of suicide information. *International Journal of Academic Research in Business and Social Sciences*, *13*(5), 1806–1816. https://hrmars.com/papers_submitted/16805/suicide-prediction-using-machine-learning-techniques-and-interactive-visualization-of-suicide-information.pdf