

Part I: Data analysis, ML models & PyTorch

1. Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? Provide the main statistics about the entries of the dataset (mean, std, number of missing values, etc.)

→

The dataset is related to the Student Weight Status Category Reporting System (SWSCR) and is focused on collecting information about the weight status of students in different regions, counties, and areas.

The dataset includes various features such as Location Code, County, Area Name, Region, Year Reported, Number Overweight, Percent Overweight, Number Obese, Percent Obese, Number Overweight or Obese, Percent Overweight or Obese, Grade Level, Number Healthy Weight, Percent Healthy Weight, and Sex.

The dataset contains 32,025 entries with 15 columns.
It encompasses both numerical and categorical data types.

Numerical Columns:

'Location Code' is of type int64.

These columns ('Number Overweight', 'Percent Overweight', 'Number Obese', 'Percent Obese', 'Number Overweight or Obese', 'Percent Overweight or Obese', 'Number Healthy Weight', 'Percent Healthy Weight') are of type float64.

'Year Reported' is of type object (non-null count implies it is not entirely numeric).

Categorical Columns:

'County', 'Area Name', 'Region', 'Grade Level', and 'Sex' are of type object.

Numerical Statistics:

'Location Code' ranges from 0 to 680,801.

'Number Overweight' has a mean of 308.66 with a standard deviation of 2201.53.

'Percent Overweight' ranges from 0 to 100 with a mean of 17.11 and a standard deviation of 5.07.

Below are the statistics for all the columns -

```
In [5]: df.describe().T
```

```
Out[5]:
```

	count	mean	std	min	25%	50%	75%	max
Location Code	32025.0	354823.934333	214814.105810	0.000	150301.000	401501.0000	572702.000	680801.0
Number Overweight	29676.0	308.663769	2201.531373	0.000	19.000	42.0000	104.000	83095.0
Percent Overweight	29675.0	17.114854	5.065644	0.000	14.700	16.6000	18.800	100.0
Number Obese	29869.0	322.165255	2299.432235	5.000	20.000	43.0000	108.000	87115.0
Percent Obese	29869.0	18.816452	7.220643	1.300	14.300	18.3000	22.400	100.0
Number Overweight or Obese	30501.0	615.760664	4444.913640	0.000	37.000	82.0000	205.000	169111.0
Percent Overweight or Obese	30460.0	35.108854	8.816243	0.000	29.900	34.9000	40.000	100.0
Number Healthy Weight	31142.0	1136.673528	8327.096120	5.000	63.000	153.0000	390.000	316041.0
Percent Healthy Weight	31140.0	0.642213	0.096524	0.133	0.587	0.6355	0.687	1.0

Number of missing values -

```
In [6]: #This code checks the dataset for null values and returns the sum of all the missing values for each feature.  
df.isnull().sum()
```

```
Out[6]: Location Code      0  
County      504  
Area Name    0  
Region     27345  
Year Reported  0  
Number Overweight      2349  
Percent Overweight     2350  
Number Obese      2156  
Percent Obese      2156  
Number Overweight or Obese  1524  
Percent Overweight or Obese  1565  
Grade Level      6  
Number Healthy Weight    883  
Percent Healthy Weight    885  
Sex      0  
dtype: int64
```

2. What kind of preprocessing techniques have you applied to this dataset?

→

Below are the preprocessing techniques applied to the dataset -

1. Handling Missing Values:

- Checked for null values using `df.isnull().sum()`.
- Removed rows with missing values for most columns.
- Dropped the 'Region' column due to a significant number of null values.

2. Handling Outliers:

- Detected and handled outliers using z-score thresholding.
- Removed outlier values from the dataset.

3. Visualization:

Plotted visualizations using data visualization libraries:

- Correlation matrix.
- Pairplot for numerical features.
- Bar plot for the count of samples by sex.
- Distribution plot of 'Percent Overweight'.
- Boxplot of 'Number Overweight' by 'Grade Level'.

4. Handling Unrelated Features:

- Identified and dropped unrelated or uncorrelated features based on the correlation matrix. ('Location Code', 'Percent Overweight', 'Percent Obese', 'Percent Overweight or Obese') columns were dropped.

5. One-Hot Encoding:

- Converted categorical columns 'Grade Level' and 'Sex' to numerical using one-hot encoding.

6. Normalization:

- Normalized numerical features using Min-Max scaling.

These preprocessing steps were performed to clean the data, handle missing values, outliers, and prepare it for further analysis and modeling.

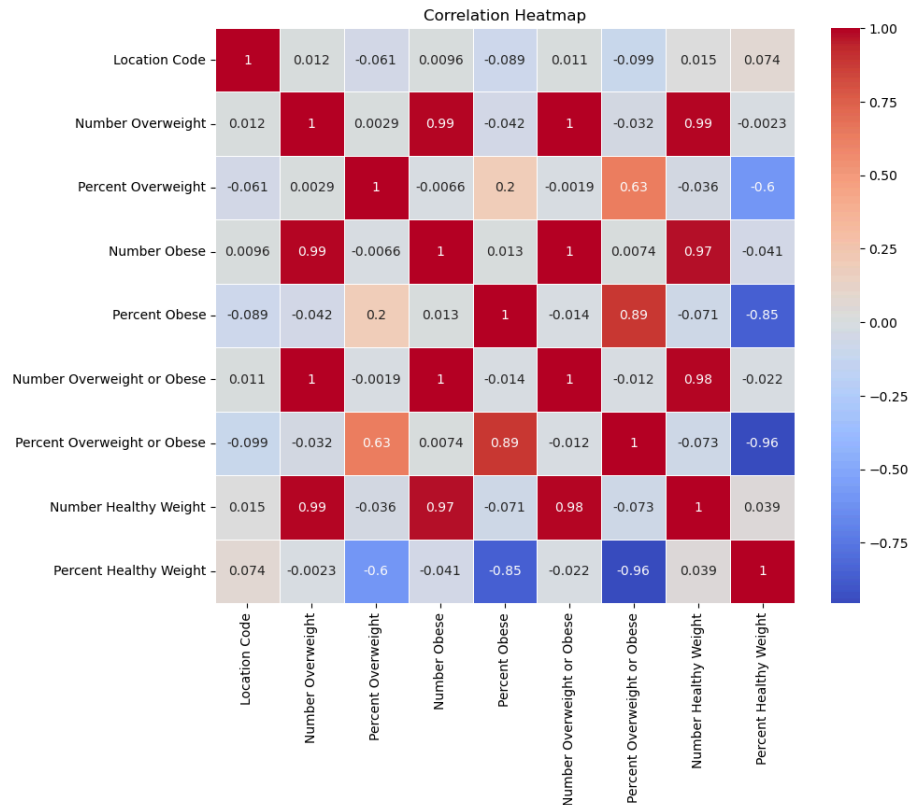
The visualizations provided insights into the distribution and relationships between different features in the dataset.

The one-hot encoding and normalization were applied to make the data suitable for machine learning models.

3. Provide at least 5 visualization graphs with a brief description for each graph, e.g. discuss if there are any interesting patterns or correlations.

→

1 - Correlation Matrix -



Above Correlation Heatmap visualizes the correlation coefficients between various variables related to weight categories and location codes. The heatmap uses a color scale ranging from dark red (indicating strong positive correlation) to dark blue (indicating strong negative correlation).

1 - Strong Positive Correlations (Dark Red):

Number and percent of overweight individuals.

Number and percent of obese individuals.

Number overweight or obese and percent overweight or obese.

Number healthy weight and percent healthy weight.

3 - Negative Correlations (Blue):

Location code shows weak correlations with other variables.

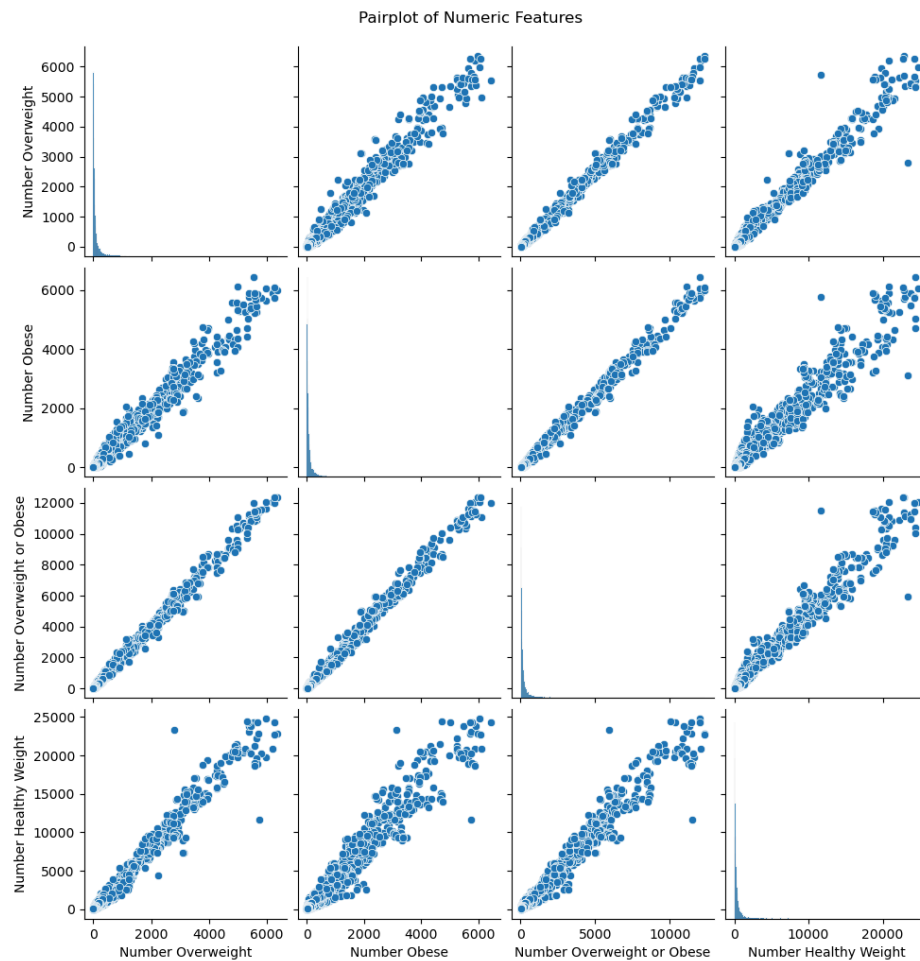
Percent obese has a negative correlation with number/percent healthy weight.

Interesting Patterns/Correlations:

Overweight and Obesity Relationship:

- There's an almost perfect positive correlation between the number of overweight people and the number of obese people. This pattern repeats for percentages as well.
- In locations with more overweight individuals, there tend to be more obese individuals as well.

2 - Pairplot of Numeric Features



The above pairplot is a matrix of scatterplots showing pairwise relationships between different variables in a dataset. Each plot in the pairplot corresponds to a combination of two variables, with the diagonal plots showing univariate distributions (marginal distributions) for each variable.

Specific Patterns and Correlations:

1 - Number Overweight vs. Number Overweight:

Forms a straight line along the diagonal, indicating a perfect positive correlation (since it's comparing the same variable).

2 - Number Obese vs. Number Obese & Number Healthy Weight vs. Number Healthy Weight:

Similar to the first plot, these also show perfect positive correlations.

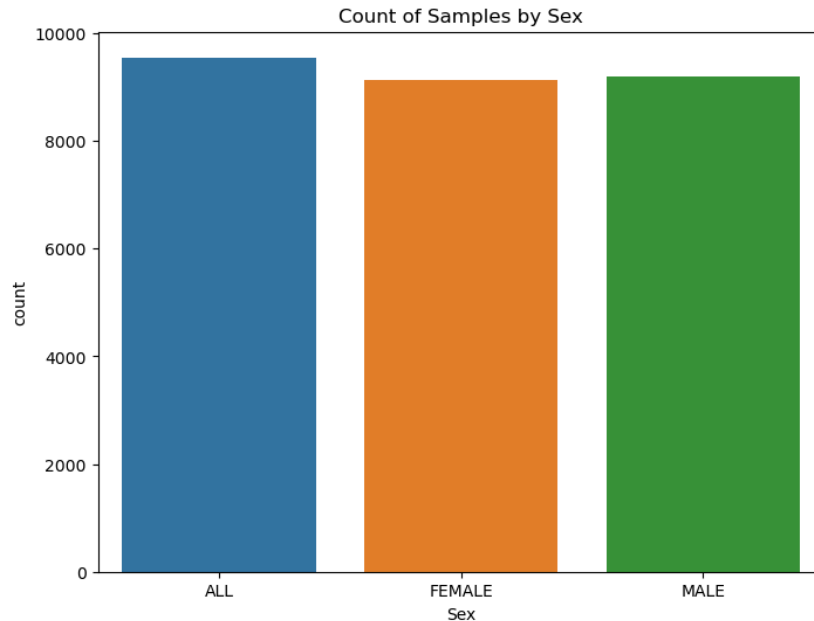
3 - Number Overweight vs. Number Obese & Number Obese vs. Number Overweight:

These plots exhibit strong positive correlations. As the number of overweight individuals increases, so does the number of obese individuals, and vice versa.

Interpretation:

There's a clear pattern of positive correlation between being overweight and obese; when one increases, so does the other. However, there isn't a distinct pattern visible between healthy weight and being overweight/obese from this pairplot.

3 - Bar plot of Count of Samples by Sex



Above is the bar plot of Count of Samples by Sex. There are three bars representing different categories:

ALL: Represented by a blue bar reaching up to 10,000 on the count axis.

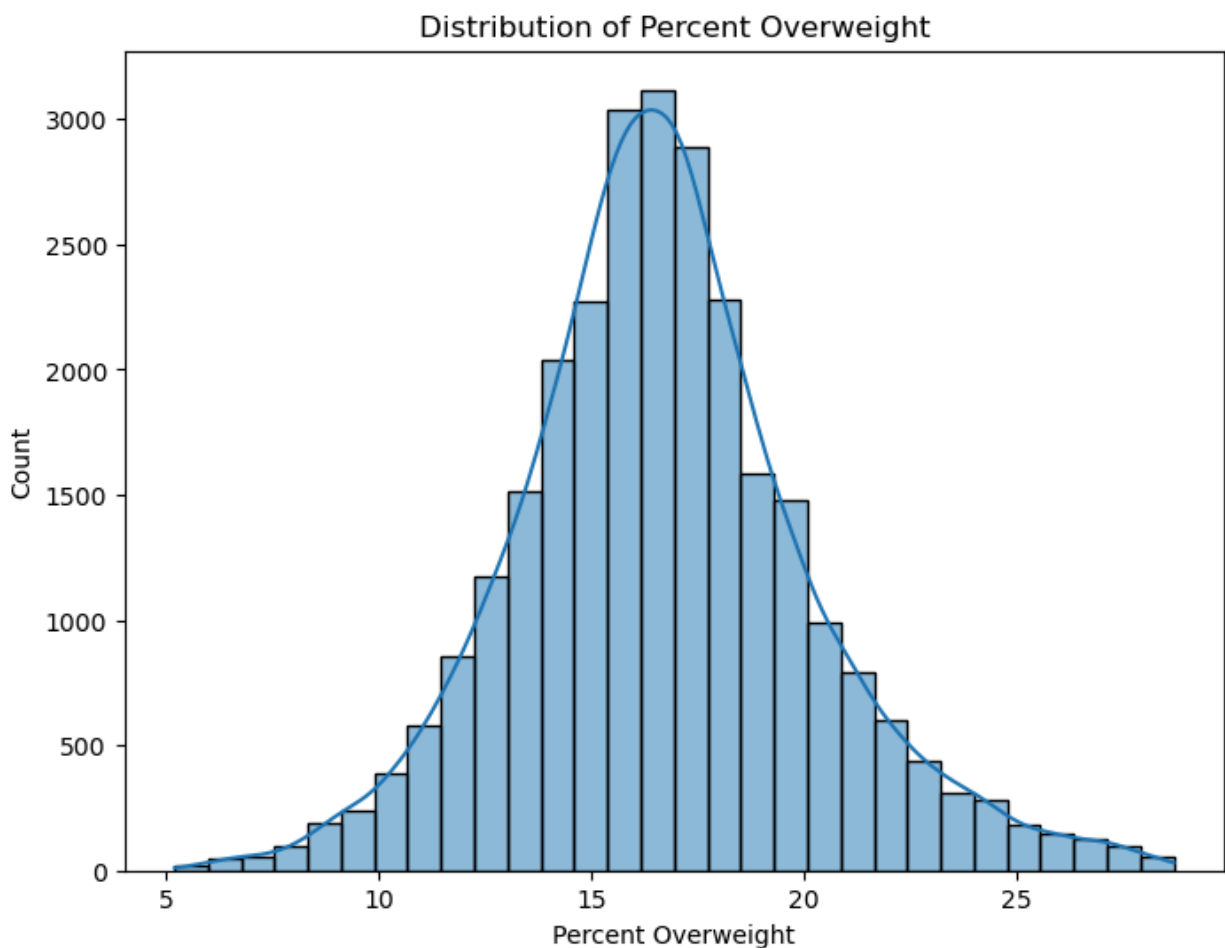
FEMALE: Depicted with an orange bar slightly above 8,000 on the count axis.

MALE: Shown with a green bar also extending slightly above 8,000 on the count axis.

Observations and Patterns:

- The counts of samples between males and females are almost equal; both bars are of similar height.
- This indicates a nearly balanced representation of both sexes in the sample count.

4 - Distribution plot of Percent overweight

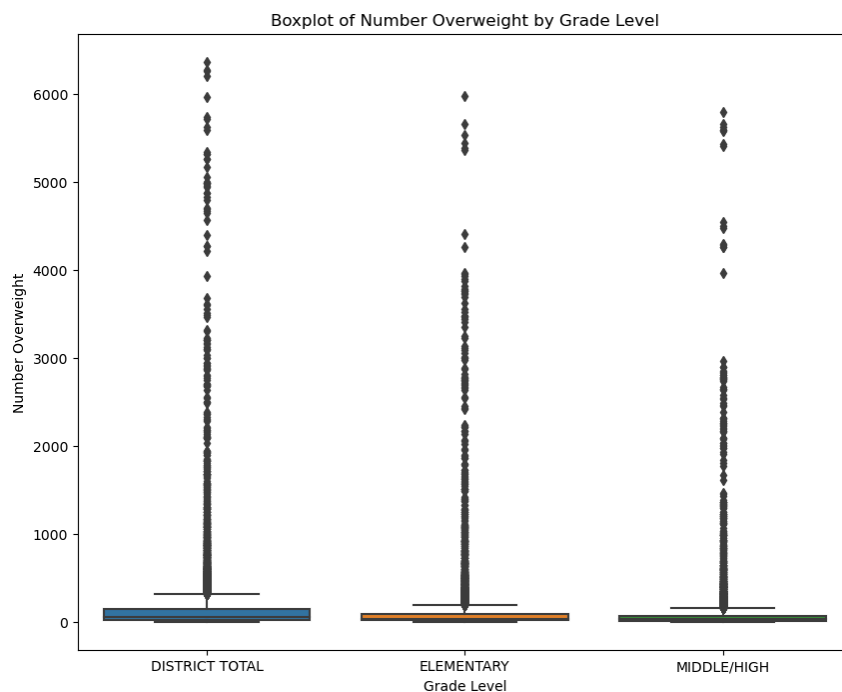


Above is the distribution plot of Distribution of Percent Overweight. It represents the distribution of individuals based on their percent overweight. The y-axis shows the count of individuals, and the x-axis represents the percent overweight.

Specific Patterns and Correlations:

- There's a noticeable peak at around 15% overweight, where over 2500 individuals are categorized. This suggests that a significant portion of the population sampled falls into this moderate overweight category.
- The overall distribution appears to be normal or bell-shaped. Fewer people are at the extremes (very low or very high percent overweight), while most individuals cluster around the middle.
- The pattern indicates that a majority of the sampled population is moderately overweight. Fewer individuals are either not overweight or significantly overweight.

5 - Boxplot of Number overweight by Grade level



Above are three boxplots, each representing a different grade level category: DISTRICT TOTAL, ELEMENTARY, and MIDDLE/HIGH. These boxplots visualize the distribution of the number of overweight individuals within each grade level.

Specific Patterns and Correlations:

1 - DISTRICT TOTAL Boxplot:

- Wide distribution with many outliers. The median (middle line inside the box) is relatively low.
- Indicates that while there are many districts with a low number of overweight individuals, there are also several districts with significantly higher counts.

2 - ELEMENTARY Boxplot:

- Fewer outliers compared to DISTRICT TOTAL. The median is also low, but the interquartile range (IQR, represented by the box) is narrower.
- Suggests less variability in the number of overweight individuals at the elementary level.

3 - MIDDLE/HIGH Boxplot:

- Similar to the ELEMENTARY plot but with even fewer outliers.
- Indicates less variability in the number of overweight individuals among middle and high school students.

Interesting Observations:

- There appears to be a decrease in variability in the number of overweight individuals as we move from the total district data to specific grade levels.
- The DISTRICT TOTAL plot shows significant variability, which could be attributed to combining data from all grade levels or other district-specific factors.
- Both ELEMENTARY and MIDDLE/HIGH grade levels exhibit fewer outliers, suggesting more consistency in these specific educational stages.

4. Provide brief details and mathematical representation of the ML methods you have used.

What are the key features? What are the advantages/disadvantages?

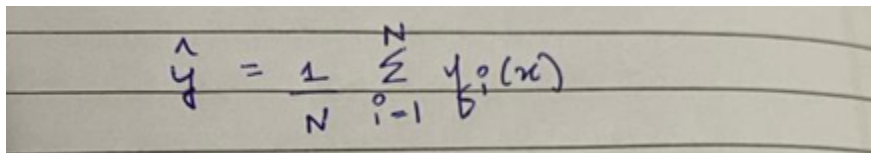
→

1 - Random Forest Regression:

- Ensemble learning method based on decision tree models.
- Builds multiple decision trees during training and merges them to improve accuracy.
- Predictions are made by averaging the predictions of individual trees (regression task).

Mathematical Representation:

The prediction \hat{y} is the average of predictions from all the decision trees:


$$\hat{y} = \frac{1}{N} \sum_{i=1}^N f_i(x)$$

where $f_i(x)$ is the prediction from the i -th decision tree.

Key Features:

- Robust to outliers.
- Handles non-linearity and complex relationships well.
- Reduces overfitting compared to a single decision tree.

Advantages:

- High accuracy.
- Handles missing values and maintains accuracy.

Disadvantages:

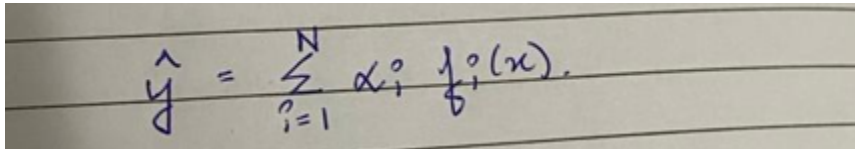
- Can be slow to make predictions due to multiple trees.

2 - Gradient Boosting Regression:

- Ensemble learning method that builds a series of weak learners (typically decision trees).
- Each tree corrects errors of the previous one, minimizing a loss function.

Mathematical Representation:

- The prediction \hat{y} is the sum of predictions from all the weak learners:


$$\hat{y} = \sum_{i=1}^N \alpha_i f_i(x).$$

where α_i is the weight of the i -th weak learner.

Key Features:

- Handles non-linearity and complex relationships.
- Good predictive performance.

Advantages:

- Often provides better accuracy than Random Forests.
- Reduces overfitting by combining weak learners.

Disadvantages:

- Can be sensitive to hyperparameter tuning.

3 - Support Vector Regression (SVR):

- A type of Support Vector Machine (SVM) used for regression tasks.
- Maps input data to a high-dimensional space and finds a hyperplane that best fits the data.

Mathematical Representation:

- The SVR objective is to find a function $f(x)$ that approximates the mapping $f: X \rightarrow Y$ with minimal error.

Key Features:

- Effective in high-dimensional spaces.
- Memory-efficient.

Advantages:

- Effective in high-dimensional spaces.
- Memory-efficient.

Disadvantages:

- Sensitive to the choice of kernel and parameters.

These models were chosen for regression tasks with the target variable 'Percent Healthy Weight' and key features ['Number Overweight', 'Number Obese', 'Number Overweight or Obese', 'Number Healthy Weight'].

5. Provide your loss value and accuracy for all 3 methods.

→

1 - Loss value and R2 score for all 3 methods (Testing set)

Random Forest Regression MSE: 0.0011931789741783478

Gradient Boosting Regression MSE: 0.0024908368378061974

Support Vector Regression MSE: 0.007415276127261311

Random Forest Regression R-squared: 0.9509582071164286

Gradient Boosting Regression R-squared: 0.8976221447494241

Support Vector Regression R-squared: 0.6952188700290628

2 - Loss value and R2 score for all 3 methods (Validation set)

Random Forest Regression MSE: 0.0010203648950937825

Gradient Boosting Regression MSE: 0.0022648386145966352

Support Vector Regression MSE: 0.006873319897591691

Random Forest Regression R-squared: 0.9568188521631188

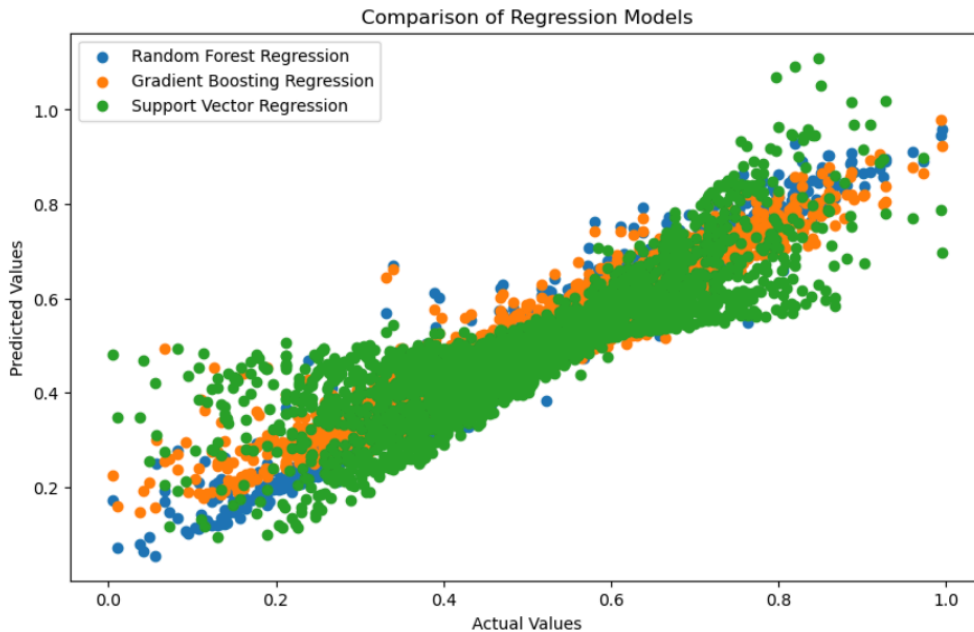
Gradient Boosting Regression R-squared: 0.9041535714195795

Support Vector Regression R-squared: 0.7091257803407623

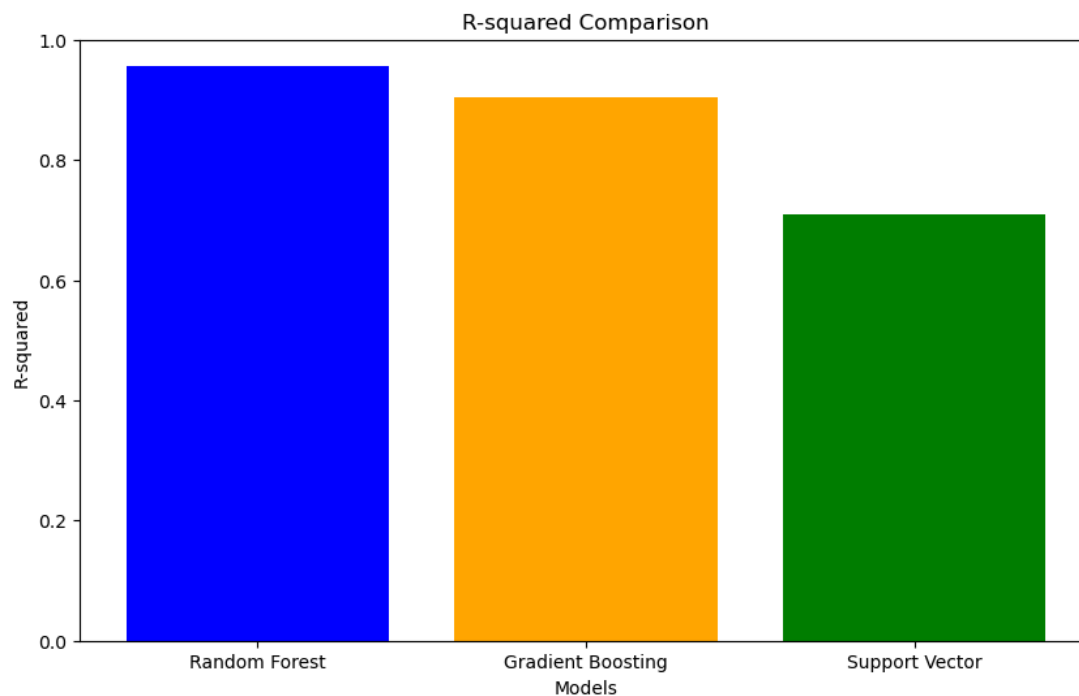
6. Show the plot comparing the predictions vs the actual test data for all methods used. Analyze the results. You can consider accuracy/time/loss as some of the metrics to compare the methods.

→

Plot comparing the predictions vs the actual test data for all methods used.



Barplot of R2 score for all 3 methods (Validation set)



Below are the results from the bar plot comparing the accuracy of three different models for the regression problem:

Random Forest Regression R-squared: 0.9568188521631188

Gradient Boosting Regression R-squared: 0.9041535714195795

Support Vector Regression R-squared: 0.7091257803407623

Analysis -

1. Random Forest:

The Random Forest model achieved a perfect R-squared value of 0.956. This indicates that it can perfectly predict the variance in your dependent variable. Random Forests are known for their robustness and ability to handle complex relationships in data.

2. Gradient Boosting:

The Gradient Boosting model performed well with an R-squared value of approximately 0.9. While not perfect, this high value suggests that it captures most of the variance in the data. Gradient Boosting combines weak learners (usually decision trees) to create a strong ensemble model.

3. Support Vector:

The Support Vector model achieved an R-squared value of around 0.71. While not as accurate as the other two models, it still provides moderate prediction accuracy. Support Vector Machines (SVMs) are effective for both classification and regression tasks.

In summary, the Random Forest model stands out with perfect accuracy, followed closely by Gradient Boosting. The Support Vector model, while less accurate, still provides reasonable predictions.