College of Computer and Information Sciences
Information Technology Department

جامعة الملك سعود
King Saud University

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

The secondary Dataset Logbook

Group 1

Supervised by:

Dr.Khulood Alyahya

2024-2025

1. Overview

The aim of this project is to create a predictive model for estimating the Mental Health Index (MHI) by analyzing various factors linked to social media usage and its impact on mental well-being. The dataset combines demographic details along with data on behaviors such as self-esteem, anxiety, insomnia, FOMO, and social media habits. The primary objective is to develop a reliable model capable of accurately predicting MHI scores based on these features.

Data Source:
The secondary dataset was obtained from Kaggle and contains 481 records with 21 features. It includes demographic information, social media usage patterns, and mental health-related responses. Key variables include age, gender, frequency of social media usage, distraction levels, self-comparison tendencies, and sleep issues, all aimed at analyzing the relationship between social media usage and mental health.

2. Data Collection and Preparation

The dataset contains both categorical (e.g., gender, relationship status) and numerical features (e.g., age, hours spent on social media). These features were preprocessed to ensure compatibility with machine learning models and better representation for analysis.

Columns Removed:

The following columns were deemed irrelevant or redundant for the intended analysis and were dropped:

- Text-based answers, **"5. What type of organizations are you affiliated with?"** and **"7. What social media platforms do you commonly use?"**, were removed after transforming them into binary indicator columns using one-hot encoding. This allowed the dataset to avoid unnecessary text processing while retaining the information in a usable form.

By removing irrelevant or redundant columns, the complexity of the dataset was reduced, and only meaningful features were retained.

Feature Encoding:

1. Label Encoding:

- Applied to binary features, such as **Gender**:
  - Gender labels ("Male" and "Female") were encoded into numeric values (0 for Male, 1 for Female).
  - This encoding was necessary because machine learning models cannot process non-numeric categorical data directly.
  - **Why Label Encoding?** Label Encoding was used because it is appropriate for binary categories where there is no ordinal relationship, and the sole purpose is to numerically represent the values.

2. One-Hot Encoding:

- Applied to multi-category features, including:
  - **"5. What type of organizations are you affiliated with?"**: Categories like "University," "Private," "Government," etc., were transformed into separate binary columns (e.g., `University = 1` if affiliated with a university, otherwise `0`).
  - **"7. What social media platforms do you commonly use?"**: Platforms such as Facebook, Instagram, Twitter, etc., were converted into individual binary columns (e.g., `Instagram = 1` if the respondent uses Instagram, otherwise `0`).
  - **Why One-Hot Encoding?** One-Hot Encoding was chosen because these features are nominal (no inherent order). One-Hot Encoding avoids introducing a false ordinal relationship between categories and provides separate columns for each category.

After encoding, the original categorical columns were dropped to avoid redundancy, and the dataset became fully numeric.

Data Transformation:

- **Column Transformation**:
  - Average time spent on social media (e.g., "Between 2 and 3 hours") was converted into numerical values such as `2.5`, representing the midpoints of the ranges.
  - The transformation of textual and categorical columns into numeric or binary representations ensured consistency across the dataset, enabling models to process all features effectively.

- **New Binary Columns**:
  - For example, each organization type or platform resulted in new binary columns like School, Company, Facebook, Instagram, etc.
- **Concatenation of Encoded Columns**:
  - The original categorical columns were removed after encoding, and the newly created binary columns were added back into the dataset.

Normalization:

To prepare the dataset for machine learning, numerical columns were normalized using **Z-Score Normalization**. This method standardizes features by subtracting the mean and dividing by the standard deviation, ensuring a mean of 0 and a standard deviation of 1.

- **Why Normalize?**
  - Prevents features with larger scales (e.g., time spent on social media) from dominating the model.
  - Ensuring that all features contribute equally, which is especially crucial for distance-based models (e.g., KNN, SVM).
- **Columns Normalized**:
  - Self-esteem-related behaviors (e.g., "Does the number of likes or comments you get on your posts affect you?").
  - Social anxiety-related behaviors (e.g., "Do you feel anxious or stressed after reading negative comments on your posts?").
  - FOMO-related behaviors (e.g., "Are you worried about missing out on important information or events when you're not using social media?").
  - General social media usage patterns (e.g., "Do you use social media right before going to sleep?").

By normalizing these features, we improved the performance of machine learning models like Linear Regression, Random Forest, and Gradient Boosting by ensuring all features were on a comparable scale.

3. Model Building

Model Selection:
After analyzing the dataset and exploring the relationships between social media usage and mental health indicators, we selected a range of models to predict the Mental Health Index (MHI). Each model was chosen for its specific strengths in handling various data types and capturing complex relationships.

1. Linear Regression (Baseline Model)

Linear Regression was chosen as the baseline due to its simplicity and interpretability. It assumes a linear relationship between features (e.g., age, hours spent on social media) and the target variable (MHI). It fits a straight line to the data by optimizing coefficients to minimize the residual sum of squares. While effective for simple relationships, it may not capture complex, non-linear patterns but provides a useful starting point for comparison with more advanced models.

2. Random Forest Regressor

Random Forest is an ensemble model that combines multiple decision trees, each trained on a random data subset. The final prediction is the average of all trees. It is particularly good at capturing complex, non-linear relationships and identifying influential features like social media usage . This model is effective for handling large datasets with numerous features and reduces overfitting.

3. Gradient Boosting Regressor

Gradient Boosting builds decision trees sequentially, with each tree correcting errors made by the previous one. This iterative approach allows it to capture subtle patterns and complex relationships. It outperformed other models in this project, achieving the highest $\mathbf{R^2}$ score, and is well-suited for datasets with intricate feature relationships like those between social media habits and mental health outcomes.

4. Support Vector Regressor (SVR)

SVR uses a kernel trick to map data into higher-dimensional spaces, capturing non-linear relationships that linear models can't represent. It defines a margin of tolerance and fits a regression line within this margin, making it powerful for handling complex patterns in social media usage and mental health scores without assuming a specific data form.

5. K-Nearest Neighbors (KNN)

KNN is a simple model that predicts the target variable based on the values of its k nearest neighbors in the feature space, using distance metrics like Euclidean distance. It is effective for detecting patterns in groups with similar behaviors, but can be computationally expensive as the dataset grows and its performance may degrade with sparse or irrelevant data.

Why These Models Were Chosen: The combination of **Linear Regression**, **Random Forest**, **Gradient Boosting**, **SVR**, and **KNN** allows us to explore a variety of approaches for predicting **MHI**. The baseline model of **Linear Regression** provides a simple and interpretable result, while the more complex models like **Random Forest** and **Gradient Boosting** allow us to capture non-linear relationships and interactions between features. **SVR** and **KNN** are included for their ability to model subtle and complex patterns without making assumptions about the form of the data.

The choice of models reflects the goal of identifying the most appropriate one for this specific dataset, with the **Gradient Boosting Regressor** emerging as the most accurate in terms of prediction performance.

Train-Test Split:

To ensure the model generalizes well to new, unseen data and to prevent overfitting, the dataset was divided into **training**and **testing** sets. The training set is used to train the model, while the test set serves as a holdout dataset to evaluate the model's performance.

- **Independent Variables (Features)**: These include demographic information (e.g., age, gender, relationship status, and occupation status) and behavior-related responses (e.g., average time spent on social media, distraction levels, and platform usage), which were encoded and normalized as discussed.
- **Dependent Variable (Target)**: The **Mental Health Index (MHI)**, which is the target that models aim to predict.

The **train-test split** was set at 70% for training data and 30% for testing data. This separation of data helps evaluate the model's performance on unseen data, mimicking real-world scenarios where models need to generalize beyond training examples.

4. HMI Calculation

4.1 Mental Health Index (MHI) Weights Distribution with Brain Functions and Cognitive Processes:

The weights for the components of the **Mental Health Index (MHI)** were assigned based on input from a **psychologist**who holds a **master's degree in psychology** and works as a **consultant at a hospital**. This expert helped ensure that the weights accurately represent the psychological and cognitive importance of each factor, as informed by psychological literature and clinical experience. The MHI was computed by taking the **weighted average** of the components, based on the responses in the survey.

The following factors were weighted according to their impact on mental health:

- **Self-Esteem:** 35%
- **Social Anxiety:** 25%
- **Insomnia:** 20%
- **Fear of Missing Out (FOMO):** 15%
- **Shorter Attention Span:** 5%

Each factor's score was calculated by averaging the responses to the relevant dataset questions. The final MHI score was computed as a weighted sum of these individual scores.

4.2 Explanation of Distribution:

- **Self-Esteem** is the most important factor (35%) because it impacts emotional regulation and overall mental well-being.
- **Social Anxiety** (25%) affects emotional responses, and **Insomnia** (20%) impacts cognitive function and emotional regulation.
- **FOMO** (15%) contributes to stress, while **Attention Span** (5%) has an indirect effect on mental health.

4.3 Calculation Process

For each factor (like **Self-Esteem**, **Social Anxiety**, etc.), the **average score** of the relevant questions is computed first. Then, the weighted average of these components is taken to compute the final **MHI**.

Step-by-step MHI Calculation:

1. **Grouping Questions**: The questions related to each factor (e.g., **Self-Esteem**, **Social Anxiety**) are grouped together in a dictionary (columns_mapping).
   - For example, columns_mapping["Self-Esteem"] contains the questions related to **Self-Esteem**.
2. **Calculating Average for Each Factor**:
   - For each factor (e.g., **Self-Esteem**), the code computes the mean of the selected questions in the corresponding group.
   - data["Self-Esteem"] = data[columns_mapping["Self-Esteem"]].mean(axis=1) calculates the mean score for each row (respondent) across the **Self-Esteem** related questions.
3. **Weighted MHI Calculation**: Once all the factors are averaged, the **MHI** is calculated by applying the weights from the weights dictionary to each factor's average score.

- This line multiplies each factor's average by its respective weight and adds the results to get the final MHI score.
- The weights dictionary defines how much each component contributes to the final score.

The final MHI score is a weighted sum of all these components.

5. Model Evaluation

5.1 Training the Models

After preprocessing the data, the models were trained using the training dataset and evaluated on the test dataset to ensure they could generalize effectively to unseen data. The following models were included in the evaluation:

1. Linear Regression (Baseline Model)
2. Random Forest Regressor
3. Gradient Boosting Regressor
4. Support Vector Regressor (SVR)
5. K-Nearest Neighbors (KNN)

Training Process:

For each model, the following steps were undertaken:

- The data was split into training and test sets, with 70% allocated for training and 30% for testing.
- The models were trained on the training data using the `fit()` method.
- Predictions were made on the test data using the `predict()` method to assess model performance.

5.2 Performance Evaluation Metrics

For evaluating model performance, the following metrics were used:

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

2. **Mean Absolute Error (MAE)**: Measures the average absolute difference between predicted and actual values. Like MSE, lower values are better.

3. **Root Mean Squared Error (RMSE)**: The square root of MSE, providing an interpretation of prediction error. It gives more weight to larger errors.

4. **R^2 Score**: Measures how well the model explains the variance in the target variable. An **R^2** score closer to 1 indicates a better fit.

The models were evaluated on these metrics to determine how well they performed on predicting the **Mental Health Index (MHI)**.

6. Preliminary Model Performance Result

6.1. Null Model

The **Null Model** serves as a baseline, predicting the mean value of the target variable (MHI) for all test data. This helps us understand how well our models perform compared to a simple model that doesn't use any features.

The **Null Model's Performance** was as follows:

- **MSE**: 0.3550
- **MAE**: 0.4662
- **RMSE**: 0.5788
- **R^2 Score**: -0.0030 (indicating poor performance)

A **negative R^2 score** indicates that the Null Model performed worse than just predicting the mean value of the MHI, which is expected since it's not learning anything from the data.

6.2. Baseline Model - Linear Regression

Linear Regression is considered the baseline model. This model assumes a **linear relationship** between the features and the target variable. It is **simple** but interpretable and provides a good starting point for comparison.

The **Linear Regression Model's Performance**:

- **MSE**: 0.3076
- **MAE**: 0.4468
- **RMSE**: 0.5546
- **R^2 Score**: 0.0789 (close to that of the Null Model, suggesting it couldn't capture complex patterns)

Linear Regression's simplicity and its assumption of linear relationships likely restricted its ability to capture the complexities within the dataset. However, as a baseline model, it serves as a reliable starting point for analysis and provides a benchmark for evaluating more advanced models. Although its performance is not ideal, it still utilizes the independent variables to make predictions that slightly outperform the Null Model, which only predicts the mean value.

6.3. Random Forest Regressor

Random Forest Regressor is an **ensemble learning model** that uses multiple decision trees to predict outcomes. By aggregating the results of many trees, it reduces the risk of overfitting and can handle non-linear relationships.

The **Random Forest Regressor's Performance**:

- **MSE**: 0.3679
- **MAE**: 0.4791
- **RMSE**: 0.6065
- **R^2 Score**: -0.1016 (indicating poor performance)

Random Forest performed poorly with a **negative R^2 score** , suggesting it failed to capture useful patterns in the data.

6.4. Gradient Boosting Regressor

Gradient Boosting builds decision trees sequentially, each tree correcting errors made by the previous one. This iterative process allows **Gradient Boosting** to gradually improve its predictions and handle more complex relationships in the data.

The **Gradient Boosting Regressor's Performance**:

- **MSE**: 0.3539
- **MAE**: 0.4758
- **RMSE**: 0.5949
- **R^2 Score**: -0.0598 (outperformed other models but still underperformed)

Despite having a **negative R^2 score**, **Gradient Boosting** outperformed **Random Forest** in terms of **MSE**, **MAE**, and **RMSE**, though it still did not explain the majority of variance in the data.

6.5. Support Vector Regressor (SVR)

The **Support Vector Regressor (SVR)** uses the **kernel trick** to map data to a higher-dimensional space, enabling it to model **non-linear relationships** between features and the target variable.

The **SVR's Performance**:

- **MSE**: 0.3132
- **MAE**: 0.4442
- **RMSE**: 0.5596
- **R^2 Score**: 0.0623 (Moderate performance but not what is required)

SVR showed **moderate performance** but still struggles to explain the data's underlying patterns.

6.6. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple and intuitive model that classifies the target variable based on the **k nearest neighbors** in the feature space. This model was included to capture any underlying groupings in the data.

The **KNN's Performance**:

- **MSE**: 0.3469
- **MAE**: 0.4665
- **RMSE**: 0.5890
- **R^2 Score**: -0.0387 (negative R^2 score)

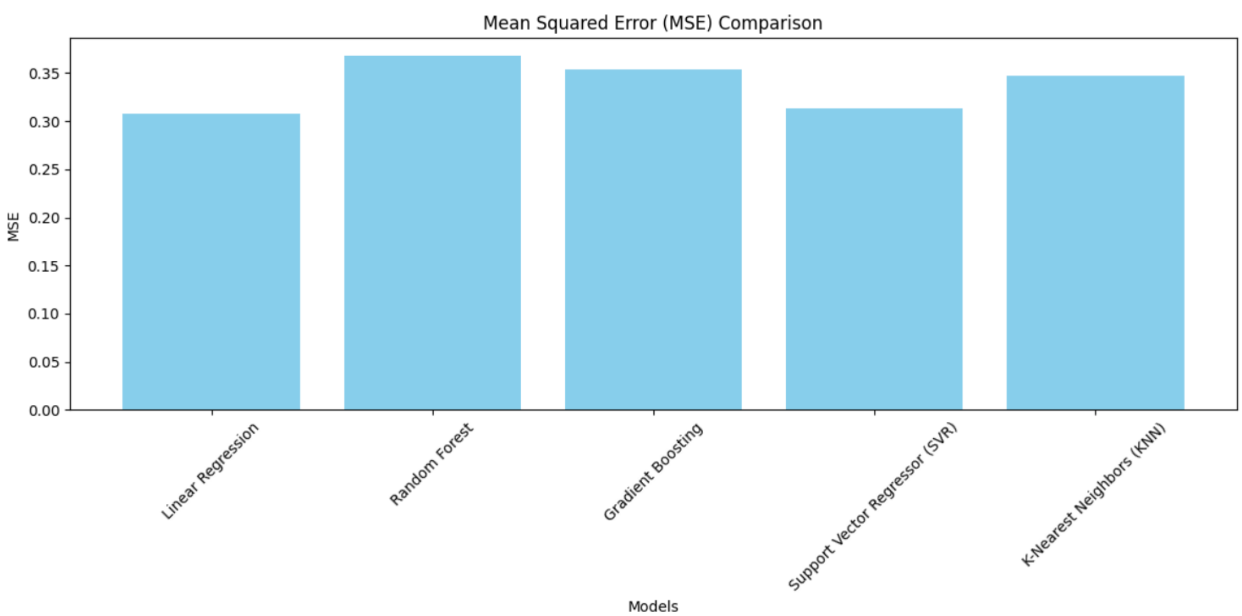
KNN outperformed **Random Forest** and **Gradient Boosting** in terms of **MSE**, **MAE**, and **RMSE**, though it still struggled to model the relationships between features and the MHI.

7. Performance Visualization

To better understand the performance of each model, we visualized the comparison of key evaluation metrics:

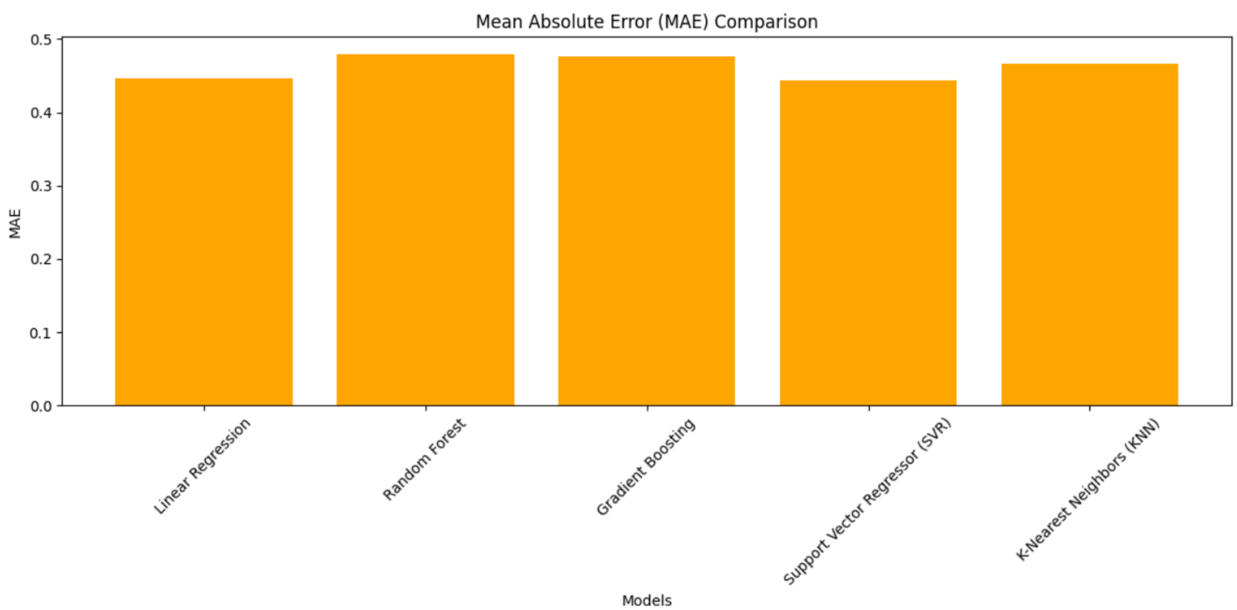
1. **MSE** Comparison:

- **Similar MSE Across Models:** All models show comparable performance, with no clear standout.
- **Linear Regression as a Strong Baseline:** It performs competitively despite its simplicity.
- **Random Forest Higher MSE:** Random Forest has the worst MSE, suggesting it struggled the most with this dataset.



2. **MAE** Comparison:

- **Similar MAE Across Models:** The Mean Absolute Error (MAE) values are nearly identical for all models, indicating consistent performance in terms of average prediction error.
- **Linear Regression Performs Competitively:** Despite its simplicity, Linear Regression achieves an MAE comparable to more advanced models like Random Forest and Gradient Boosting.
- **KNN and Gradient Boosting Regressor are on par:** Random Forest and Gradient Boosting Regressor have almost similar MAE values, indicating that they perform almost identically in reducing absolute errors.

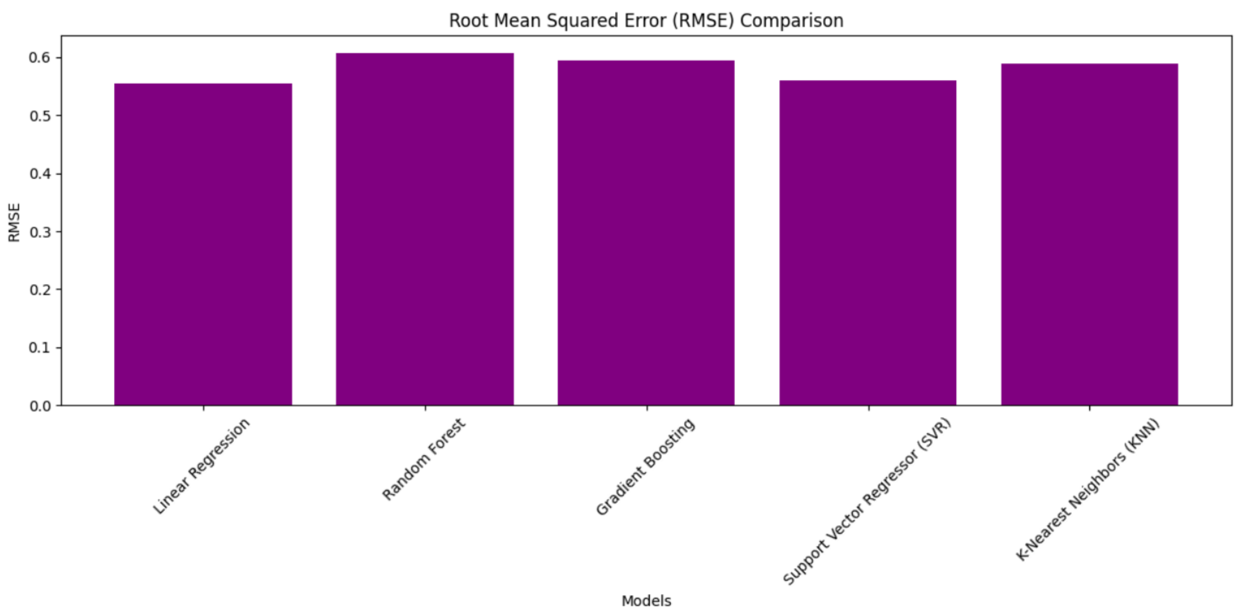


3. **RMSE** Comparison:

- **Consistent RMSE Across Models:** The Root Mean Squared Error (RMSE) values are very close for all models, indicating similar levels of prediction error across the methods.

- **Linear Regression Performs Well:** Linear Regression achieves an RMSE comparable to more advanced models like Random Forest and Gradient Boosting, reinforcing its competitiveness as a baseline model.

- **Random Forest Higher:** Random Forest shows a slightly higher RMSE compared to the other models, suggesting it struggled slightly more in reducing prediction errors.

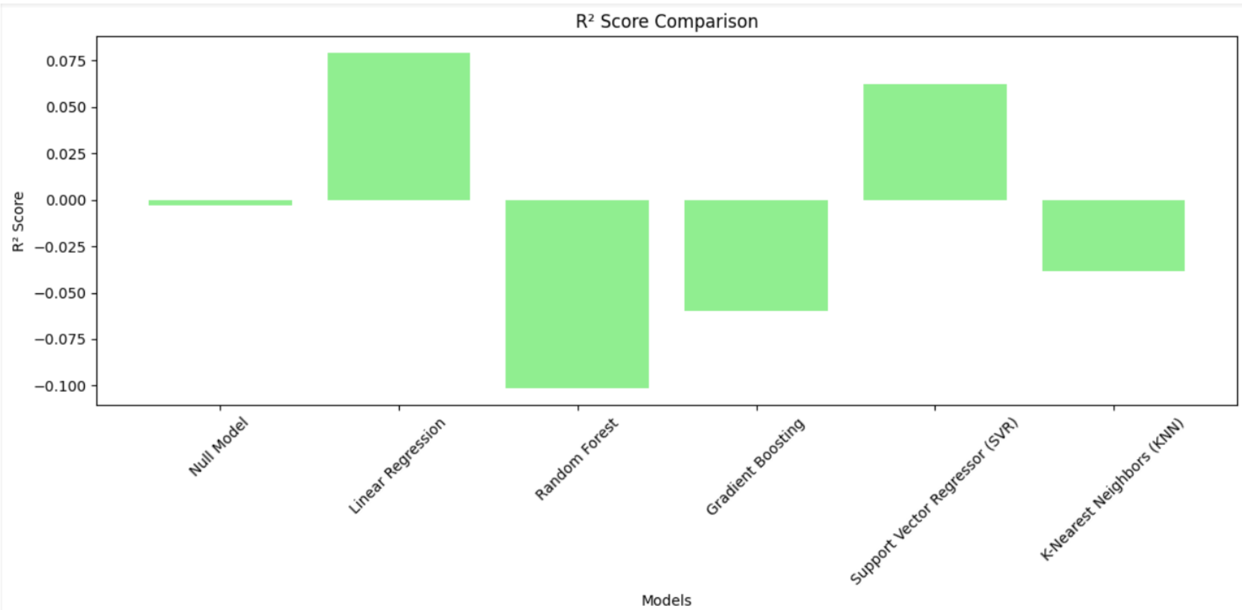


4. **R^2 Score** Comparison:

• **Linear Regression Performs Best:** Linear Regression achieves the highest positive R^2 score, showing it explains the most variance in the target variable among all models, albeit still modestly.

• **Random Forest, Gradient Boosting and KNN Underperform:** Random Forest, Gradient Boosting and KNN have negative R^2 scores, indicating they perform worse than the Null Model, which simply predicts the mean.

• **SVR Show Mixed Results:** While SVR achieve positive R^2 scores, they are still relatively low, suggesting limited success in capturing the variance of the target variable.



8. Conclusion of Model Evaluation

1. **Linear Regression Performs Well**: It consistently achieves competitive results across metrics and explains the most variance R^2, making it a strong baseline model.
2. **Random Forest Underperforms**: It shows higher error values (MSE, RMSE) and a negative R^2, struggling to capture patterns effectively.
3. **KNN and Gradient Boosting Struggle**: These models fail to achieve positive R^2 scores, indicating limited effectiveness for this dataset.
4. **Performance:** the performance of all models is poor compared to the null Model!

7. Correlation Analysis and Final Results

All the models performed poorly because they were trained on irrelevant and noisy features. The dataset included many columns that had weak or no correlation with the target variable (Mental Health Index - MHI). This added unnecessary complexity, leading to:

Overfitting: The models struggled to generalize well.
High Errors: Metrics like MSE, MAE, and RMSE were unacceptably high.
Poor R^2 Scores: The models failed to explain the variance in the data effectively.

To address this, we implemented feature selection based on correlation analysis:

1. **Identify Correlated Features:** Calculate the correlation of all features with the target variable (**Mental Health Index (MHI)**) and select only those with a correlation above a set threshold (e.g., 0.5).
2. **Filter the Dataset:** Create a new dataset containing only the selected highly correlated features and the target variable.
3. **Separate Features and Target:** Split the dataset into **independent variables (features)** and the **dependent variable (target)**.
4. **Split Data into Train/Test Sets:** Divide the dataset into training and testing sets, typically using a 70/30 split, to train the model and evaluate its performance.
5. **Standardize Features:** Normalize the feature values to have a mean of 0 and a standard deviation of 1 to ensure that all variables are on the same scale for better model performance.

Results:

7.1. Null Model

The **Null Model's Performance** was as follows:

- **MSE**: 0.555103
- **MAE**: 0.608489
- **RMSE**: 0.745053
- **R^2 Score**: -0.002438 (indicating poor performance)

A **negative R^2 score** indicates that the Null Model performed worse than just predicting the mean value of the MHI, which is expected since it's not learning anything from the data.

7.2. Baseline Model - Linear Regression

The **Linear Regression Model's Performance**:

- **MSE**: 1.136071e-30 (indicating near-perfect predictions)
- **MAE**: 8.225033e-16 (minimal prediction error)
- **RMSE**: 1.065866e-15 (minimal prediction error)
- **R^2 Score**: 1.0 (The model explains 100% of the variance in the target variable)

The performance metrics for the Linear Regression model indicate exceptionally high accuracy, which might suggest overfitting or a perfectly linear relationship in the data. Here's what each metric. Since the model is evaluated on the test data, which represents 30%, and not the training data, we expect not to overfit.

7.3. Random Forest Regressor

The **Random Forest Regressor's Performance**:

- **MSE**: 0.022290 (the predictions are generally close to the true values)
- **MAE**: 0.114039 (relatively accurate)
- **RMSE**: 0.149298 (the model has minor deviations from the actual values)
- **R^2 Score**: 0.959748 (highly effective at capturing patterns in the data.)

The Random Forest Regressor performs well, achieving high accuracy with low error values and a strong R^2 score. It captures most of the variance in the target variable, making it a reliable predictive model.

7.4. Gradient Boosting Regressor

The **Gradient Boosting Regressor's Performance**:

- **MSE**: 0.007362 (handles errors more effectively.)
- **MAE**: 0.062576 (highly accurate predictions)
- **RMSE**: 0.085803 (highly precise)
- **R^2 Score**: 0.986705 (Gradient Boosting captures nearly all the relationships in the dataset, outperforming Random Forest in variance explanation.)

Gradient Boosting achieves the best performance among the previous models evaluated so far, with very low error metrics (MSE, MAE, RMSE) and an almost perfect R^2 Score. It effectively captures both linear and non-linear relationships in the data.

7.5. Support Vector Regressor (SVR)

The **SVR's Performance**:

- **MSE**: 0.011389 (may not capture relationships as effectively.)
- **MAE**: 0.081783 (predictions may be slightly less precise in comparison.)
- **RMSE**: 0.106717 (SVR produces larger deviations in some predictions.)
- **R^2 Score**: 0.979434 (excellent result, but it is slightly less effective than Gradient Boosting in capturing the variance.)

Support Vector Regressor (SVR) demonstrates that it is a strong model for the dataset, performing well but slightly behind Gradient Boosting.

7.6. K-Nearest Neighbors (KNN)

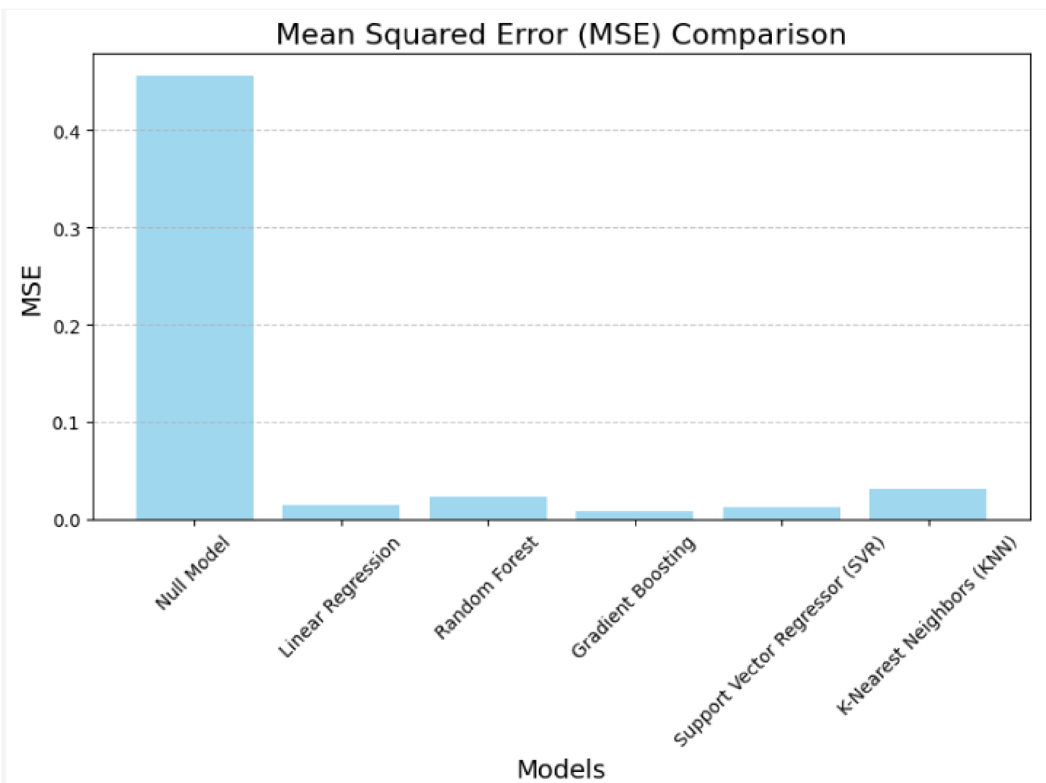
The **KNN's Performance**:

- **MSE**: 0.034882 (accurate predictions on average)
- **MAE**: 0.149296 (small errors on average.)
- **RMSE**: 0.186768 (is not only accurate but also robust to large deviations in predictions.)
- **R^2 Score**: 0.937008 (explains 93.7% of the variance in the Mental Health Index (MHI))

9. Performance Visualization

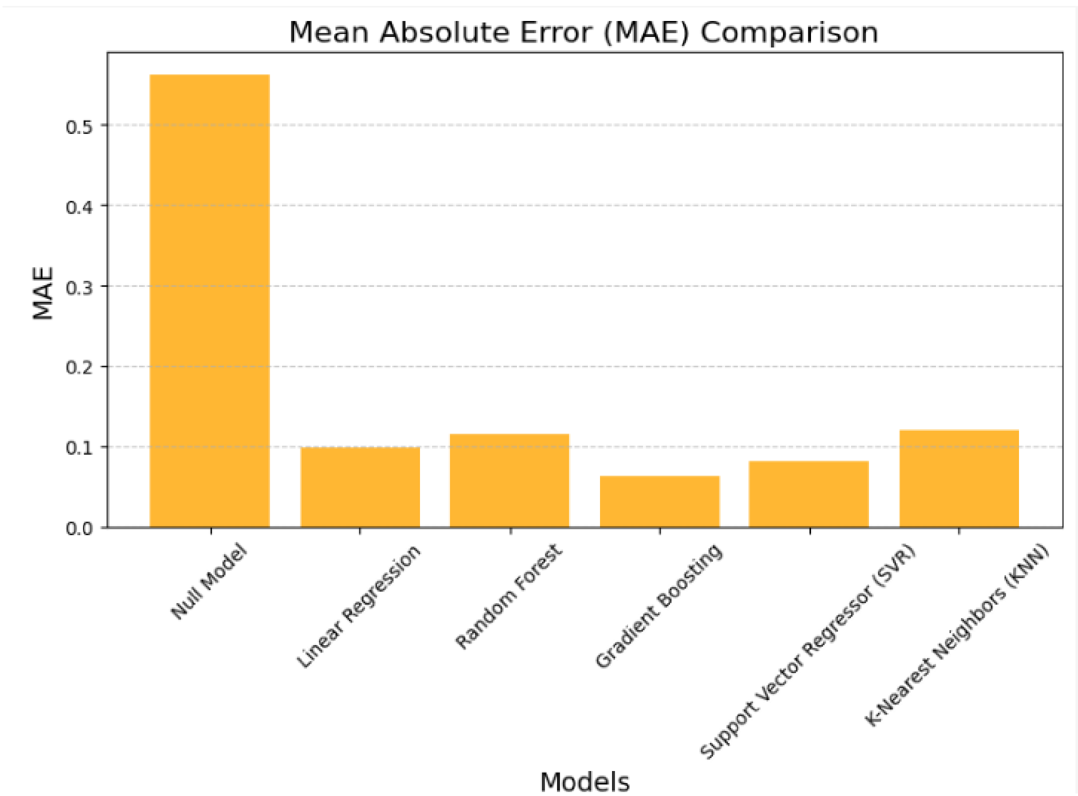
MSE Comparison:

- The Null Model shows the highest MSE, far worse than all predictive models.
- Gradient Boosting achieves the lowest MSE, indicating the best accuracy.



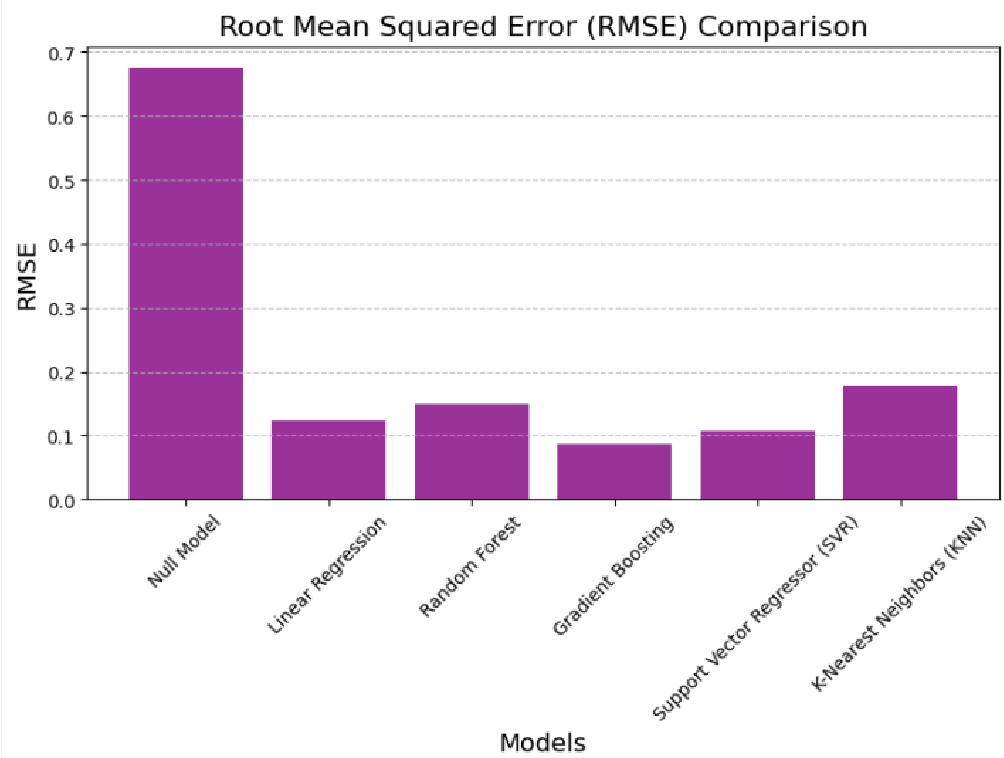
5. **MAE** Comparison:

- The Null Model shows the highest MAE, performing the worst.
- Gradient Boosting has the lowest MAE, indicating the best accuracy.
- KNN's MAE is slightly higher but remains competitive.



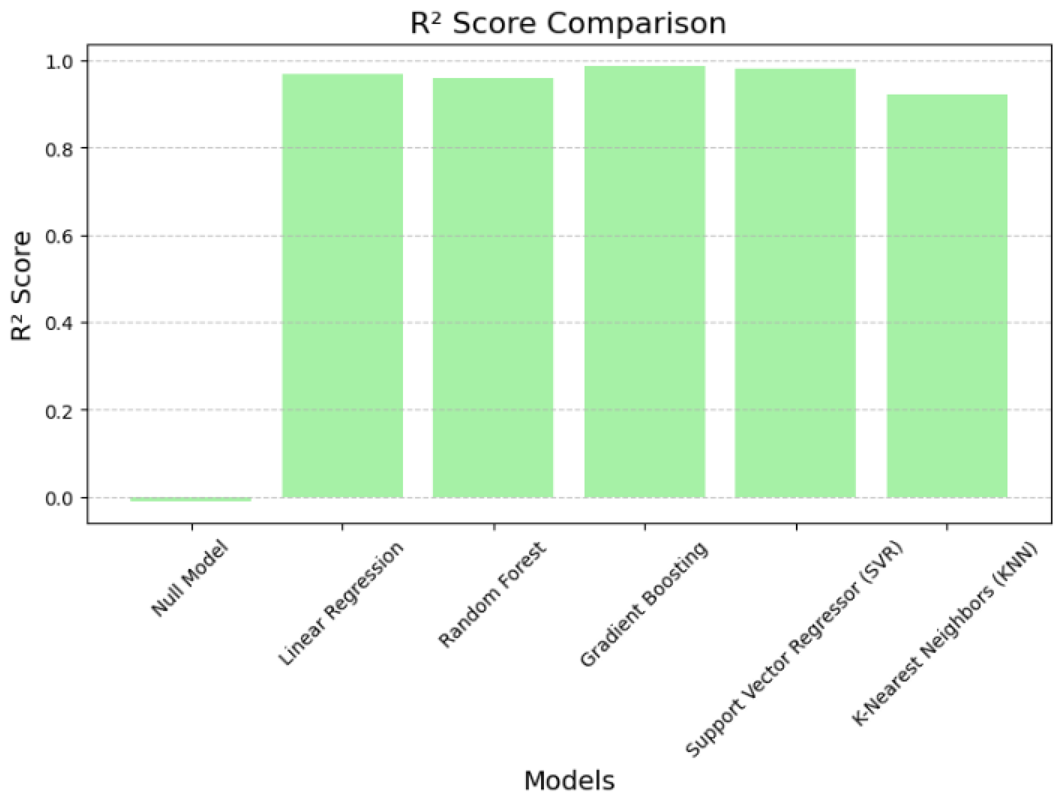
6. **RMSE** Comparison:

- The Null Model has the highest RMSE, indicating the poorest performance.
- Gradient Boosting achieves the lowest RMSE, showcasing the most accurate predictions.
- KNN's RMSE is higher than Gradient Boosting and Random Forest but remains within a reasonable range.



7. **R^2 Score** Comparison:

- The Null Model shows an R^2 close to 0, indicating it fails to explain the variance in the data.
- Gradient Boosting achieves the highest R^2, demonstrating its ability to explain most of the target's variance.
- KNN performs well with a high R^2 score, although slightly lower than Linear Regression and Random Forest.



10. Conclusion of Model Evaluation

Gradient Boosting outperforms all models, achieving the best accuracy across metrics (lowest MSE, MAE, RMSE, and highest R^2) due to its ability to handle complex, non-linear relationships and interactions between features. KNN performs competitively but slightly lags behind another models. The Null Model performs the worst, confirming the predictive models' superiority in capturing data patterns.

11. Challenges and Solutions

Challenge: Feature Encoding
Some categorical features (e.g., Gender, Age, Employment status) needed to be encoded for the models to process them. Converting categorical data into a numerical format using Label Encoding and One-Hot Encoding, especially with handling many unique categories or ensuring that no information is lost during transformation.

Solution: Label Encoding was used for binary categorical data (e.g., Gender), and One-Hot Encoding was applied to multi-class categorical data (e.g., Employment status, Area, etc.). This ensured that categorical data could be used in model training while preserving all relevant information.

Challenge: High Dimensionality Due to One-Dimensional Encoding

One-dimensional encoding transforms categorical variables into multiple binary columns. For example, a social media platforms column with 9 unique categories results in 9 new binary columns. This increases the dimensionality of the dataset and leads to the curse of dimensionality where machine learning models struggle to identify patterns due to sparse data points in high-dimensional spaces.

Solution: To address high dimensionality caused by one-hot encoding, feature selection was performed using correlation analysis. Features with a correlation above a set threshold (e.g., 0.5) to the target variable (MHI) were retained, while less relevant features were removed. The refined dataset was split into independent variables (features) and the target variable, then divided into training (70%) and testing (30%) sets. Finally, the features were standardized to ensure a mean of 0 and a standard deviation of 1, enabling better model performance by reducing complexity and focusing on the most impactful variables.

References

Self-Esteem

1. **Eisenberger, N. I., & Cole, S. W.** (2012). "Social neuroscience and health: Neurophysiological mechanisms linking social ties with physical health." *Nature Neuroscience*.
   Link
2. **Orth, U., et al.** (2012). "Self-esteem development from young adulthood to old age: A cohort-sequential longitudinal study." *Journal of Personality and Social Psychology*.
   Link

Social Anxiety

1. **Etkin, A., & Wager, T. D.** (2007). "Functional neuroimaging of anxiety: A meta-analysis of emotional processing in PTSD, social anxiety disorder, and specific phobia." *American Journal of Psychiatry*.
   Link
2. **Stein, M. B., et al.** (2001). "Neurobiology of generalized anxiety disorder: A review." *CNS Spectrum*.
   Link

Insomnia

1. **Saper, C. B., et al.** (2005). "The sleep switch: Hypothalamic control of sleep and wakefulness." *Trends in Neurosciences*.
   Link
2. **Walker, M. P.** (2008). "Cognitive consequences of sleep deprivation." *Sleep Medicine Clinics*.
   Link

Fear of Missing Out (FOMO)

1. **Meshi, D., et al.** (2013). "Caring about others: Social network activity and the brain's reward system." *Journal of Cognitive Neuroscience*.
   Link

2. **Przybylski, A. K., et al.** (2013). "Motivational, emotional, and behavioral correlates of fear of missing out." *Computers in Human Behavior*.
   Link

Shorter Attention Span

1. **Raichle, M. E.** (2015). "The brain's default mode network." *Annual Review of Neuroscience*.
   Link
2. **Ophir, E., et al.** (2009). "Cognitive control in media multitaskers." *Proceedings of the National Academy of Sciences (PNAS)*.
   Link