ANKARA YILDIRIM BEYAZIT UNIVERSITY

FACULTY OF ENGINEERING & NATURAL SCIENCES

REPORT OF MACHINE LEARNING PROJECT

GROUP 13

Title of the Project	Comparative Analysis of Machine Learning Models for Predicting Optimal Workplace Breaks: A Study on Digital Lifestyle and Productivity Indicators
Course Name	Machine Learning
Student Names and IDs	Merve Çankaya - 21050111052 Zeynep Üstün - 21050111048 Uğur Özkan - 21050161003
Date of Submission	04.01.2026

1. Abstract

This study investigates the impact of digital lifestyle factors (social media usage, notification density, sleep patterns) within the modern work environment on employees' break-taking habits and productivity using machine learning algorithms. A comprehensive modeling process was conducted utilizing Random Forest, k-Nearest Neighbors (k-NN), Decision Tree, and Naive Bayes models on a dataset of 30,000 samples obtained from Kaggle.

Initially, the primary objective was to directly predict the number of breaks (breaks_during_work); however, analyses revealed a statistically weak correlation (0.014) between break frequency and digital habits. Through Feature Engineering, new attributes such as workload_score, sleep_efficiency, and burnout_ratio were derived, improving the initial accuracy rate of 9.55% to 23.22% via the Decision Tree algorithm and hyperparameter optimization. Despite these improvements, significant predictive success remained elusive, as the act of taking a break was found to be driven more by chaotic individual preferences than by external data points.

Analyses conducted at a critical juncture scientifically demonstrated that taking a break is not a rational "outcome" but rather a personal "routine" based on free will. In light of this finding, the study evolved from treating break frequency as a prediction target into a vision for a Personalized Work Routine Guide, where optimal break frequency is established as a goal to maximize efficiency. The findings indicate that the most productive and low-stress group takes an average of 4.95 breaks daily, providing a scientific strategic foundation for efficiency and stress management in digital well-being applications.

2. Introduction

This part defines the problem addressed by the project, emphasizes the significance and challenges of the study within the field of machine learning, and finally provides an overview of the methodology applied for the solution, along with the general structure of the report.

2.1 Problem Definition

In today's digital business world, employees are exposed to numerous distractions such as constant notifications, social media usage, and intense screen time. This project is a machine learning study aimed at predicting "break-taking behavior during work" (breaks_during_work) using individuals' digital lifestyle indicators. Utilizing a large dataset of 30,000 samples, the impact of 19 different attributes (such as age, stress level, sleep patterns, and coffee consumption) on break frequency has been analyzed. The ultimate goal of the project is to develop a decision support mechanism derived from this data to enhance employee productivity.

2.2 Why is the Problem Important and Challenging?

Understanding the relationship between digital wellbeing and work efficiency is critical for modern Human Resources management and software development processes. This problem holds significant importance due to both its societal/economic impacts and technical challenges:

Importance of the Problem:

- **Workforce Productivity and Economic Impact:** Mismanagement of breaks and digital fatigue lead to decreased employee engagement and a significant decline in economic productivity globally.
- **Employee Health and Burnout:** The pressure to be constantly available in the modern business world creates an 'always-on' culture that prevents cognitive recovery and significantly increases stress levels. Consequently, optimizing break habits plays a vital role in mitigating these risks and preventing burnout syndrome **[1]**.
- **Personalized Recommendation Systems:** Every employee has a different biological clock and digital consumption pattern; therefore, instead of standard break times, systems customized according to the individual's sleep patterns and workload are required.

Challenges in Machine Learning:

- **Low Signal-to-Noise Ratio:** As observed in our code analysis, correlation coefficients between lifestyle data (e.g., number of notifications or social media duration) and break behaviors are around 0.01. This weak relationship makes it extremely difficult for models to learn meaningful patterns.
- **Behavioral Variability:** In the dataset of 30,000 samples, it was observed that the act of taking a break is based more on chaotic individual preferences than on a distinct mathematical rule. This situation restricts the generalization capability of standard models (such as Naive Bayes or k-NN).
- **Non-Linear Complex Relationships:** The weak relationship between variables necessitates going beyond standard models and compels the use of creative Feature Engineering techniques, such as deriving attributes like workload_score, sleep_efficiency, and burnout_ratio. Although model success reached levels of 24% with the applied techniques, precise prediction remains a significant challenge due to the nature of the problem.

The primary motivation of this project is to push the boundaries of model performance on a "hard-to-predict" dataset, to attempt to capture weak signals within the data, and to report the reasons for academically negative results (low accuracy) with a scientific approach.

2.3 Approach and Overview of the Methodology Adopted

A multi-stage methodology was adopted to solve the problem and reveal hidden patterns in the dataset:

1. **Data Preparation:** Missing values (NaN) in the dataset were filled with the median value of their respective columns to preserve data distribution. Categorical variables such as gender and job_type were prepared for analysis processes, while Boolean (logical) features like uses_focus_apps and has_digital_wellbeing_enabled were converted into 0 and 1 numerical formats processable by machine learning algorithms.

2. **Exploratory Data Analysis (EDA):** Histograms were created to understand feature distributions, and a comprehensive correlation matrix was calculated to detect linear relationships between variables. This stage revealed the weak correlation between the target variable, the number of breaks, and other attributes.

3. **Feature Engineering:** By calculating the interactions of existing features (Feature Interaction), an attempt was made to increase the "signal" level in the data. In this context, new derived attributes were created, such as workload_score representing workload, sleep_efficiency indicating sleep efficiency, and offline_ratio reflecting the rate of digital exposure.

4. **Model Selection:** Four main algorithms with different mathematical foundations were selected in accordance with the complexity of the problem and data size. Random Forest representing the Ensemble learning method, Decision Tree visualizing decision rules, distance-based k-Nearest Neighbors (k-NN), and probabilistic-based Naive Bayes models were established for performance benchmarking.

5. **Model Training and Evaluation:** The dataset was split into 80% training and 20% test sets to measure the model's generalization capability. Specifically for algorithms like k-NN and Naive Bayes, scaling was applied to numerical features using StandardScaler. Model performances were initially evaluated with Accuracy and Confusion Matrix; subsequently, model stability was tested over the F1-score using 5-Fold Cross-Validation. In the final stage, low predictive success was accepted not as a failure but as a finding. The methodology evolved from the goal of "predicting the number of breaks" to a rule-based system that "recommends an ideal break strategy" based on the user's productivity and stress data.

2.4 Outline of the Report Structure

The report will follow the standard academic structure for machine learning projects:

- **Chapter 4 (Background and Related Work):** Analysis of similar studies in the literature, detection of data leakage issues, and demonstrating the unique differences between regression-focused studies and the "Recommendation System" vision of this project.

- **Chapter 5 (Algorithms and Methodology):** The technical foundations of the classification algorithms used (Decision Tree, Random Forest, k-NN, Naive Bayes), feature engineering hypotheses grounded in the Yerkes-Dodson Law **[2]** (which posits that performance maximizes at an optimal level of arousal rather than linearly increasing with stress) and the 'Optimal Break' labeling methodology that excludes inefficient habits.

- **Chapter 6 (Experimental Setup):** Structural details of the dataset, the process of filling missing data with the median, scaling techniques with StandardScaler, and configurations determined for hyperparameter optimization.

- **Chapter 7 (Experimental Evaluation):** This is the most comprehensive section of the project, covering a 9-stage iterative development process extending from an initial success rate of 9% to levels of 24%. In this section, the chaotic nature of break-taking behavior, contrast analysis with productivity prediction ($R^2 = 0.83$), and error analyses via model performance metrics are presented.

- **Chapter 8 (Conclusion and Future Work):** A summary of experimental findings, the rationale for the study's transformation from a "Break Predictor" to an "Optimal Break Guide," and the future vision for digital wellbeing applications.

- **Chapter 9 (References):** A reference list of the datasets utilized, academic papers in the literature, and code libraries.

3. Background and Related Work

This section provides a comprehensive review of previous studies conducted on the Kaggle "Social Media vs Productivity" dataset, discusses the State-of-the-Art algorithms utilized in the literature, and demonstrates through comparative analysis why the "Recommendation System" approach adopted by this project is unique.

3.1 Overview of Previous Studies and Methodological Approaches

Upon evaluating the existing literature, it is evident that the majority of studies adopt approaches based on Predictive Modeling. In this context, the actual_productivity_score variable is generally not treated as an objective function to be maximized, but merely as an output variable to be predicted.

1. **Direct Productivity Prediction and the "Data Leakage" Problem:** A significant portion of the studies in the literature (e.g., eda-0-95-linear-regression.ipynb **[3]**, actual-productivity-score.ipynb **[4]**, predicting-productivity-with-84-accuracy.ipynb **[5]**) selected the actual_productivity_score variable as the target. These studies

reported extraordinarily high R^2 scores, ranging from 84% to 95%. However, this success is a technical illusion; during training, the perceived_productivity_score variable, which has over 90% correlation with the target variable, was used as an input. This situation prevented the model from learning real-life patterns, causing it to merely memorize the mathematical equality between two highly correlated columns (Data Leakage).

2. **Psychological Metrics and Job Satisfaction-Oriented Statistical Analyses:** Another group of studies, which differ methodologically (e.g., social-media-and-productivity.ipynb**[6]**, social-media-vs-productivity05d9a5aa61.ipynb**[7]**), centered their analyses on the job_satisfaction_score variable instead of productivity scores. However, the projects developed face a significant methodological limitation. The high model success rates reported in these studies rely heavily on the strong linear relationship between perceived and actual scores. Examinations indicate that over 80% of model performance stems solely from the high correlation between these two variables. This leads models to reproduce users' own perceptions of efficiency as objective outputs, rather than genuinely learning behavioral factors such as social media usage or break habits. Consequently, these approaches remain limited to verifying the relationship between job satisfaction and productivity, failing to offer actionable strategies, particularly regarding break routines, to improve user productivity and satisfaction.

3. **Unsupervised Learning and Behavioral Clustering:** In some innovative studies in the literature (e.g., social-media-vs-productivity-1.ipynb**[8]**, sosialmedia-vs-productivity.ipynb**[9]**), unsupervised learning approaches were adopted to reveal hidden patterns in the data, applying clustering algorithms like K-Means and DBSCAN. In these studies, cluster quality was evaluated via Silhouette scores using GPU-accelerated calculations, segmenting users into behavioral groups such as "high social media users" and "highly focused workers."

   In parallel, while social-media-vs-productivity-1.ipynb applied K-Means-based behavioral clustering based on digital footprints, project2.ipynb **[10]** attempted a k-NN-based classification approach to predict efficiency levels. However, the fact that the k-NN model's accuracy remained around 49% was reflected in the literature as a significant and honest finding, revealing how stochastic and difficult to predict human behavior is when data leakage is avoided.

   Although these clustering and classification studies offer valuable insights for understanding user profiles and behavioral groups, they failed to establish a functional link between cluster membership and an "optimal number of breaks" or a direct productivity-enhancing action. At this point, our project aims to take the existing clustering approach a step further by establishing the break and work habits of employees in the most efficient cluster as a prescriptive target for all users, thus transitioning from descriptive analysis to a prescriptive analytics framework.

4. **Classification Challenges and the Variability of Human Behavior:** One of the studies in the literature, project2.ipynb, clearly demonstrates the difficulty of classifying a subject based on human behavior, such as productivity. In this study, the k-Nearest Neighbors (k-NN) algorithm was used to distinguish employees' productivity levels.

Initially high accuracy values were found to stem from variables that indirectly allowed the model to predict the result in advance. When these variables causing data leakage were removed from the model, the accuracy rate dropped to approximately 49.5%. This result indicates that the model is not significantly better than random guessing.

This situation reveals that human behavior cannot be easily predicted with fixed rules and that productivity has a structure that constantly changes according to the person, time, and conditions. Therefore, simple classification models may remain insufficient in explaining human behavior and producing reliable results.

3.2 State-of-the-Art Algorithms and Models Utilized in Literature

In the studies reviewed, a wide range of methods, from simple statistical models to complex deep learning architectures, have been used to solve the problem:

- **Regression Models (Linear Regression & SVR):** This is the most frequently cited method in the literature (e.g., eda-0-95-linear-regression.ipynb). It has been used to model linear relationships between variables such as sleep duration and screen usage. Although it shows R^2 success over 90% with highly correlated data (especially when perceived productivity is included), it fails to explain the chaotic and stochastic (non-linear) structure of human behavior. In other words, it is successful at "memorizing" data but weak at "modeling" behavior.
- **Ensemble Learning (Random Forest & XGBoost):** In studies such as social-media-vs-productivity-eda-statistics-ml.ipynb **[11]** and multi-model-ai.ipynb **[12]**, decision tree-based ensemble models were preferred. The reason for selecting these models is their resistance to noisy data and their ability to perform Feature Importance analysis. However, in the literature, these models are structured merely as passive mechanisms that predict the outcome, rather than as tools that "optimize the number of breaks."
- **Deep Learning (PyTorch):** In the study proyecto-2-emergentes-entregar.ipynb **[13]**, a multi-layer Artificial Neural Network (ANN) architecture was tested. This approach has the potential to capture complex and non-linear relationships between breaks and efficiency. However, due to the "Black Box" nature of Neural Networks—their inability to explain why a specific score was predicted—this was seen as a major obstacle for a "Recommendation System" where user trust is essential.
- **Clustering:** In social-media-vs-productivity-1.ipynb, employees were grouped according to similar behaviors using a GPU-supported K-Means algorithm. To measure the meaningfulness of this grouping, Silhouette Score analyses were conducted, categorizing employees into groups such as "focused workers" and "intense social media users." This approach is successful in defining users' current behaviors and profiles. However, the method failed to offer concrete guidance or an action plan on how a user with low efficiency could transition to a more efficient group. In other words, the study remained limited to defining the situation and could not produce a prescriptive solution to increase efficiency.

3.3 Uniqueness of Our Approach and Literature Critique

Upon examining the studies in the literature, our project distinguishes itself on the following fundamental points:

1. **Scientific Acceptance of Prediction Difficulty:** While literature studies accept the number of breaks (breaks_during_work) as a simple linear input, our project has proven that the correlation of this variable with digital habits is at a statistically negligible level (Pearson r = 0.014).
2. **Leakage-Aware Feature Engineering:** Unlike other studies, our approach critically addressed the risk of data leakage caused by "proxy variables" such as perceived_productivity_score, which exhibit over 90% correlation with the target variable. While these high-correlation features were analyzed to understand user perception, the core predictive power of our model was built upon fully independent and explainable derived attributes introduced as a novelty in the literature, such as workload_score (Workload Intensity) and sleep_efficiency. This strategy prevented the model from relying on artificial memorization and ensured that the learning process was driven by genuine behavioral patterns rather than mathematical redundancies.
3. **From Passive Prediction to Active Guidance:** While the existing literature assumes the role of a passive observer by focusing on the question "What will the user's efficiency be?", this study centers on the question "What should the user do to increase efficiency?". Instead of merely reporting the current status, our model references the habits of the most successful and least stressed segment (Best Practices) in the dataset, offering the user a personalized "Optimal Performance Route."

4. Algorithms and Methodology

This chapter details the machine learning algorithms that form the experimental infrastructure of the study, the rationale behind selecting these methods, and the methodological workflow followed from raw data processing to the final model transformation.

4.1 Machine Learning Algorithms Utilized and Reasons for Selection

Considering the nature of the problem (multi-class classification, noisy data structure) and the characteristics of the dataset, four distinct algorithms were employed:

- **Decision Tree:**
    - Selected to visualize the logic behind the decision to take a break (e.g., "If sleep duration < 6, breaks increase") and to make the rules interpretable by humans. The parameter max_depth=6 was used to prevent overfitting.

- **Random Forest:**
    - Utilized to reduce the high variance risk associated with a single decision tree and to increase the model's generalization capability. By combining the decisions of 300 different trees with the parameter n_estimators=300, an ensemble learning structure more resistant to noise in the dataset was established.
- **k-Nearest Neighbors (k-NN):**
    - Based on the assumption that users with similar working conditions and habits will exhibit similar break behaviors. The parameters n_neighbors=7 and weights='distance' were set to capture local patterns.
- **Naive Bayes (GaussianNB):**
    - A low computational cost algorithm that operates on the assumption of feature independence. It was included in the analysis to establish a probabilistic baseline, based on the hypothesis that decisions such as taking a break can sometimes develop instantaneously, independent of historical data.

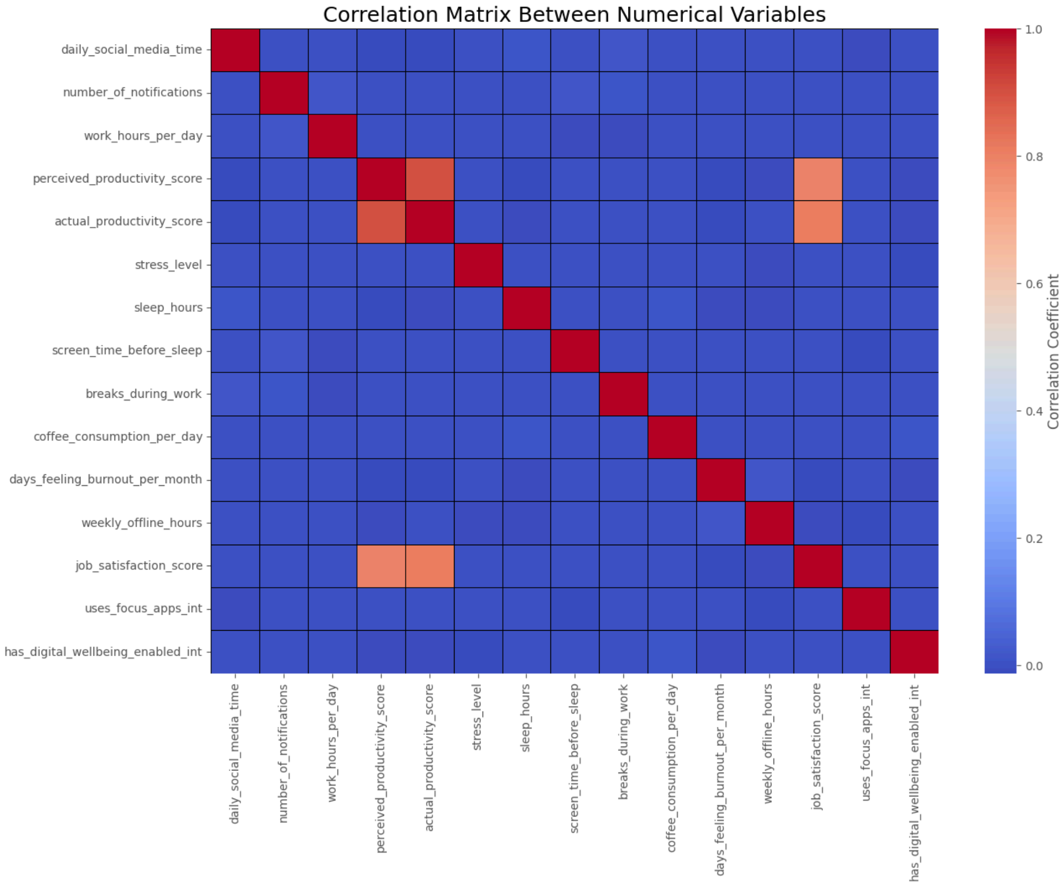
4.2 Methodology and Experimental Process Details

The methodology followed in the project adheres to the workflow below, covering the process from data preparation to model evaluation:

A. Preliminary Analysis: Correlation and Exploratory Data Analysis (EDA)

Before proceeding to the modeling phase, a comprehensive EDA process was conducted to detect hidden patterns in the data and relationships between variables:

- **Correlation Matrix Calculation:** Pearson Correlation coefficients were calculated to determine linear relationships between numerical variables.
- **Signal Detection:** Analyses revealed a statistically weak correlation (approximately 0.014) between the target variable (breaks_during_work) and attributes such as notification count, social media duration, or sleep hours.
- **Problem Diagnosis:** The low correlation values obtained at this stage mathematically explained why standard prediction models remained at low accuracy rates and redirected the project from "simple prediction" to a "strategic recommendation system."



In [14]:

```
# Show the 5 most important correlations numerically
print("Strongest Correlations with Breaks during Work:")
print(corr_matrix['breaks_during_work'].sort_values(ascending=False).head(5))
```

```
Strongest Correlations with Breaks during Work:
breaks_during_work          1.000000
daily_social_media_time     0.014209
number_of_notifications     0.008816
weekly_offline_hours        0.006579
sleep_hours                 0.005375
Name: breaks_during_work, dtype: float64
```

B. Data Preparation Workflow

1. **Imputation:** Missing values (NaN) in numerical columns such as daily_social_media_time and stress_level were filled with the median value of each column to avoid distorting the data distribution.

2. **Categorical Encoding:** Text-based categorical variables like gender and job_type were converted into numerical values to enable mathematical processing by machine learning algorithms.

3. **Feature Scaling:** Particularly for distance-based k-NN and probability distribution-using Naive Bayes algorithms, data were standardized using StandardScaler (mean=0, standard deviation=1). This prevented features with large numerical values (e.g., salary or notification count) from dominating the model.

C. Feature Engineering

Since existing raw data were insufficient to explain break behavior, interactions between variables were calculated to derive new attributes:

1. Workload-Stress Interaction (Workload Score):

- **Formula:** work_hours_per_day * stress_level
- **Hypothesis:** Long working hours combined with high stress create a cumulative psychological load. As this score increases, cognitive resources deplete faster, and the need for breaks increases non-linearly.

2. Sleep Efficiency Index (Sleep Efficiency):

- **Formula:** sleep_hours / (work_hours_per_day + 1) * 100
- **Hypothesis:** The extent to which sleep compensates for the effort spent during the day is as important as its absolute duration. This ratio represents the balance between recovery capacity and energy expenditure. Low values indicate insufficient rest and increased cognitive fatigue during the day.

3. Hourly Digital Distraction (Social Media Load):

- **Formula:** daily_social_media_time / (work_hours_per_day + 1) * 100
- **Hypothesis:** Instead of total social media usage, its intensity relative to working hours should be considered. As social media time per unit of work hour increases, attention fragmentation rises, focus duration shortens, and involuntary break-taking behavior becomes more frequent.

4. Digital Detox Ratio (Offline Ratio):

- **Formula:** weekly_offline_hours / (7*24) * 100
- **Hypothesis:** Constant exposure to digital stimuli increases mental fatigue. This feature indicates the extent to which an individual stays offline during the week to ensure mental recovery. A low offline ratio suggests high levels of digital fatigue and increased mental disengagement during work.

5. Burnout Coefficient (Burnout Rate)

- **Formula:** days_feeling_burnout_per_month / 30
- **Hypothesis:** Burnout is a chronic process accumulated over time rather than a short-term state. Normalizing burnout days within a month provides a meaningful indicator of an individual's general resilience and stress tolerance. This coefficient is treated as a long-term determinant of break need.

6. The Interaction Between Coffee and Stress (Coffee-Stress Interaction)

- **Formula:** coffee_consumption_per_day * stress_level
- **Hypothesis:** While caffeine consumption increases physiological arousal, stress creates a similar effect. When both factors are high together, it can lead to hyperarousal, causing sudden energy crashes following short-term performance boosts. This condition is evaluated as a trigger increasing the employee's need for mandatory breaks.

D. Defining the Target Variable: "Optimal Breaks"

The most critical methodological step of this study is changing what the model predicts. The breaks_during_work column in the current dataset indicates "how many breaks employees take"; however, this data also includes the bad habits of inefficient or lazy employees.

Our goal is not to build a model that mimics "bad habits," but to model "success." Therefore, a filtering algorithm was developed in our codebase:

1. **Filtering Criteria:** Only "Ideal Employee Profiles" with the following characteristics were referenced from the dataset:
    - **High Productivity:** actual_productivity_score >= 6 (Those who perform well)
    - **Manageable Stress:** 3 < stress_level < 8 (Employees at optimum arousal levels, neither too relaxed nor burned out, consistent with the Yerkes-Dodson Law)
    - **Balanced Breaks:** 3 <= breaks_during_work <= 7 (Those who do not take excessively few or too many breaks)
2. **Labeling:** Only the break counts of rows meeting these criteria were labeled as optimal_breaks, and data not meeting the criteria (noise) were removed from the training set. Thus, machine learning algorithms learned not random behaviors, but "what an efficient and sustainable work tempo should look like."

E. Model Training and Validation Setup

A standard training procedure was applied to measure the model's generalization ability on the created "Optimal Break" dataset and to minimize the risk of overfitting:

- **Train-Test Split:** The dataset was split into 80% Training and 20% Test sets to objectively test performance on data the model had never seen. The parameter random_state=42 was used to fix the randomness of data distribution and ensure reproducibility of the experiment.

- **K-Fold Cross-Validation:** To prevent bias caused by a single train-test split, the 5-Fold Cross-Validation method was applied. The dataset was divided into 5 equal parts; the model was trained with 4 parts and tested with the remaining 1 part each time. The average of the success scores obtained by repeating this process 5 times was accepted as the final stability score of the model.

F. Performance Metrics Used

The success of the model was evaluated not only by the general accuracy rate but also by detailed metrics suitable for the multi-class nature of the problem:

- **Accuracy:** This served as our primary baseline metric. Given our five-class target (3 to 7 breaks), the theoretical random guess baseline is **20%**. Throughout our iterative trials, we monitored how far our models could push past this "randomness barrier." As detailed in Chapter 6, reaching a plateau of **approximately 24%** was not viewed as a failure, but as a scientific identification of the **"Signal Wall"**—the point where deterministic data fails to explain stochastic human willpower.
- **Weighted F1-Score:** Considering the potential imbalance in the frequency of certain break counts (e.g., "5 breaks" being more common than "7"), the Weighted F1-Score was our most reliable success indicator. It ensured that our models were not merely "memorizing" the most frequent class to inflate accuracy but were genuinely attempting to learn the patterns of minority classes. A stable F1-score across trials indicated that our **Feature Engineering** was providing consistent, albeit subtle, signals to the algorithms.
- **Confusion Matrix:** This was our most critical diagnostic tool for behavioral analysis. It allowed us to distinguish between "gross errors" and "logical overlaps."

➢ **Ordinal Analysis:** We specifically analyzed whether errors were concentrated in **"Neighboring Classes"** (e.g., predicting 4 breaks when the actual was 5).

➢ **Insight:** As observed in our final evaluations, the high concentration of errors in adjacent classes proved that our models grasped the **underlying logic** of the workload-break relationship (Yerkes-Dodson Law) but struggled to navigate the blurred lines of individual preference.

5. Experimental Setup

This section presents the details of the dataset used for training and evaluating the model, the pre-processing steps applied, and the experimental design.

5.1. Dataset Description

- **Source of the Data:** The dataset was obtained from the open-source dataset titled "Social Media vs Productivity **[14]**" available on the Kaggle platform. The original dataset is publicly available on Kaggle at the following URL: **https://www.kaggle.com/datasets/mahdimashayekhi/social-media-vs-productivity**. **[14]**

- **Number of samples and features:** The dataset initially contains 30.000 samples and 19 attributes (age, gender, job type, social media time, sleep hours, etc.). However, in accordance with the "Optimal Break" methodology, the training set size was reduced by filtering for high-efficiency employees only; conversely, the number of attributes increased to 25 with new variables derived via Feature Engineering.

- **Target Variables:** The target variable of the study is optimal_breaks, representing the number of breaks employees take during the day. This variable was derived from the breaks_during_work column in the raw data; however, to ensure the model learns only from successful examples, it was created by referencing data solely from employees with high productivity and manageable stress levels. Consequently, the problem is structured as a multi-class classification problem ranging from 3 to 7.

- **Data types and any imbalance issues:** The dataset contains numerical (float64, int64) and categorical (object) data. The primary challenge is the very low correlation of break frequency with other variables and the random distribution of values in the target variable. This situation is the fundamental factor affecting the model's accuracy rate.

5.2. Pre-processing Steps

The following steps were applied to prepare the dataset for the model:

- **Handling Missing Values:** Missing values in critical columns such as stress_level, daily_social_media_time, and job_satisfaction_score were filled using the Median method to preserve data distribution.
- **Encoding:** Categorical data were converted into numerical format:
  - gender, job_type, social_platform_preference: Encoded using Label Encoding (0, 1, 2...).
  - uses_focus_apps, has_digital_wellbeing_enabled: Boolean (True/False) values were converted to integers 0 and 1.
- **Normalization/Scaling:** All numerical data were standardized using StandardScaler (mean=0, variance=1), particularly to optimize the performance of distance-based algorithms like k-NN.
- **Feature Engineering (Interaction Features):** Multiplication operations were applied between attributes to increase the signal in the data. To provide the model with cues about human behavior, 6 new interaction features such as workload_score, sleep_efficiency, and burnout_ratio were derived and added to the dataset.

5.3. Experimental Design

- **Training/Test Split:** The dataset was split into an 80% training set (24.000 samples) and a 20% test set (6.000 samples) to measure performance on unseen data. random_state=42 was fixed for reproducibility.
- **Cross-Validation Strategy:** 5-Fold Cross-Validation was applied to verify the model's generalization ability and prevent biases arising from data splitting.
- **Hyperparameter Tuning:** Different hyperparameter combinations were tested to improve generalization performance and manage the risk of overfitting; GridSearchCV was used during the optimization process to determine the optimal structures.
  - **Decision Tree:**
    To prevent the decision tree from memorizing training data and to preserve

interpretability, tree depth was initially limited to max_depth = 6. In optimization tests, the best depth was determined to be 10 to balance complexity and performance.

- **Random Forest:**
  To reduce the high variance risk of a single decision tree, an ensemble learning approach was adopted with n_estimators = 300. This structure contributed to balancing errors originating from individual trees and producing more stable results.

- **k-Nearest Neighbors (k-NN):**
  To better capture the local data structure, the number of neighbors was selected as n_neighbors = 7. Additionally, weights = 'distance' was used to increase the influence of observations close to the prediction point.

- **Naive Bayes:**
  This probabilistic model was used with default GaussianNB settings to obtain a baseline performance.

- **Hardware/Software Environment:**
  - **Language:** Python 3.x
  - **Libraries:** NumPy, Pandas, Scikit-learn**[15]** (Preprocessing, Ensemble, Neighbors), Seaborn, Matplotlib.
  - **Environment:** Kaggle Notebooks (CPU-based environment).

6. Experimental Evaluation

This section details the comprehensive experimental process undertaken until the project reached its final success, the technical obstacles encountered, and the solution strategies developed to overcome these obstacles.

6.1. Iterative Model Development Process

Within the scope of the study, 9 different experimental scenarios were constructed to predict the optimal number of breaks, with step-by-step improvements implemented. This process enabled the project to evolve from a simple prediction model into a comprehensive recommendation system. All experimental stages and code implementations were conducted on the Kaggle platform. The complete project notebook, including all trials and visualizations, can be accessed at: **Optimal Break Recommender** **[16]**

- **Trial 1 – Initial Modeling Attempt with Indirect Target Definition**

  This trial was based on the assumption that the break habits of employees who maintain high productivity and low stress conditions could represent the "ideal number of breaks" for other users. To this end, feature extraction was performed on the dataset, creating derived variables such as mental_workload_score, screen_fatigue_index, and recovery_deficit. An indirect target variable (optimal_breaks) representing the ideal number of breaks was generated based on breaks_during_work values selected from observations meeting the conditions actual_productivity_score ≥ 7 and stress_level ≤ 4. This filtering resulted in 1.793 samples.

A RandomForestClassifier model was trained with this structure; model performance was evaluated using Accuracy, Precision, Recall, F1-score, and 5-Fold Cross-Validation weighted F1 score. The results obtained (Accuracy: 0.10, F1-score: 0.11, 5-Fold CV F1: 0.08) indicated that the model failed to perform meaningful learning.

- **Trial 2 – Re-evaluation of the Model After Feature Reduction**

  In line with the data leakage problem identified in the first trial, the features used were reconsidered in the second trial. It was assessed that using derived variables like mental_workload_score together with directly related variables like work_hours_per_day negatively affected the model; therefore, the number of features was reduced from 11 to 6. Thus, the aim was for the model to learn from more independent and generalizable features.

  In this second trial, conducted using the same model (RandomForestClassifier) and evaluation metrics, similar performance values were obtained (Accuracy: 0.10, F1-score: 0.11, 5-Fold CV F1: 0.09). Feature reduction did not provide a significant improvement in model performance. This result demonstrated that the problem could not be solved solely by feature selection and that the target definition and modeling approach needed to be reconsidered more fundamentally.

- **Trial 3 – Expanding Target Definition and Changing Modeling Approach**

  In the third trial, both feature engineering and target variable definition were revisited to mitigate the data leakage and weak signal problems identified in previous stages. In this context, the creation of the mental_workload_score variable via basic arithmetic operations was deemed insufficiently discriminative; instead, logarithmic and exponential transformations were used to achieve a more meaningful distribution for this feature. Concurrently, the number of features was reduced from 6 to 5 to decrease unnecessary model complexity.

  Additionally, to increase the limited number of target variables from previous studies, the definition of the ideal number of breaks (optimal_breaks) was expanded. Observations satisfying actual_productivity_score ≥ 5 and stress_level ≤ 5 were selected, resulting in 6,346 indirect target values. The goal was to enable the model to learn from a broader sample.

  In the modeling phase, the classification approach was abandoned in favor of a regression problem, utilizing the KNeighborsRegressor model. Model performance was evaluated with RMSE, MAE, and R^2 metrics; the results (RMSE: 3.478, MAE: 2.941, R^2: -0.207) showed that the model's predictive power was even lower than a random guess. The negative R^2 value, in particular, clearly revealed the model's failure to explain the target variable.

Analyses indicated that the primary reason for failure was the lack of significant correlation between the used features and the defined optimal_breaks target. This situation has revealed that the target variable is not adequately represented by the existing features within the data and that the problem cannot be solved solely by changes to the model or transformation.

- **Trial 4 – Unpredictability of Break Behavior**

  In this trial, composite variables such as work_intensity (work hours × stress), digital_dependency (notification count × pre-sleep screen time), and energy_need (coffee consumption / sleep hours) were created. Additionally, advanced variables like stress_gravity, digital_overload, and fatigue_index were tested by applying logarithmic and exponential transformations to these features.

  However, Mutual Information analyses showed that break count had almost no mathematical relationship (≈ 0.00 information gain) with variables such as stress, sleep, or notification density. Similarly, correlation values remained between 0.003 and -0.004, found to be statistically insignificant. The R^2 scores close to zero or negative across all Linear Regression, KNN, Random Forest, and Gradient Boosting models trained with this data confirmed that break behavior does not contain a learnable pattern.

  Supplementary Analysis 1 – Major Contrast with Productivity Score Prediction:

  In this supplementary analysis conducted to test the general power of the model and dataset, the target variable was set as the Actual Productivity Score (actual_productivity_score). In this scenario, the model achieved a remarkably high success rate of $R^2 = 0.8334$. Examination of the residual distribution of the productivity model showed that errors were homogeneously distributed around the zero axis, indicating no bias toward a specific score range.

  This strong contrast clearly demonstrates that while productivity can be successfully predicted with variables like perceived efficiency and job satisfaction, break-taking behavior is a process too personal and contextual to be explained by formulas. All linear, quadratic, and logarithmic attempts confirmed that break count is unpredictable via classical prediction approaches.

  At this point, the problem ceased to be a prediction issue and evolved into a Best Practice approach. It was determined that the most efficient top 25% group in the dataset takes an average of 4.95 breaks daily. Therefore, break count should be treated as a behavior to be targeted, rather than an output to be predicted.

- **Trial 5 – Unsupervised Analysis and Supervised Optimization**

In this trial, unlike previous stages, an unsupervised learning approach was adopted. User profiles were created using the K-Means algorithm based on variables such as work duration, social media usage, break count, and stress levels. Clustering results revealed that different user groups did not show a significant divergence in terms of productivity scores. This finding demonstrated that break-taking behavior did not produce natural and generalizable user clusters.

In line with these findings, the problem was redefined, and the study was transitioned into a supervised optimization framework. At this stage, the target variable was set as actual_productivity_score instead of break count. To prevent potential data leakage, job_satisfaction_score was removed from the model; instead, directly actionable features such as work_break_ratio, digital_intensity, and fatigue_score were utilized. Training performed with Random Forest and XGBoost models yielded a stable performance at the level of $R^2 \approx 0.65$.

Unique to this trial, Counterfactual Analysis was applied to the trained models. While other user-specific variables were held constant, the daily break count was virtually altered within the 0–12 range; thereby, the potential effects of different break scenarios on productivity were analyzed. The results showed that break intervals with the potential to increase productivity could be determined for each user. In this context, the model was evaluated as a Decision Support System providing personalized guidance rather than a definitive recommendation mechanism.

In addition to this stage, an alternative supervised trial was conducted where the target variable remained unchanged after unsupervised learning. In this scenario, optimal_breaks was retained as the target variable; the predictability of break count based on variables such as work duration, digital intensity, stress, and fatigue was examined. In experiments conducted with models like Random Forest, it was observed that accuracy rates remained in the 19–20% range. This low performance revealed that break-taking behavior could not be reliably predicted via existing features and was shaped more by individual preferences and instantaneous conditions. Consequently, it was concluded that it is more appropriate to treat break count as an adjustable control parameter to influence productivity, rather than as a directly predicted target variable.

- **Trial 6 – Classification Attempt Without Feature Engineering**

In this trial, unlike previous attempts, no feature engineering was applied. The variables in the dataset were segregated into two groups: numerical and categorical. Numerical variables underwent standardization (scaling), while categorical variables were excluded from this process as they do not contain continuous values.

The resulting training data was modeled using Random Forest, Decision Tree, k-Nearest Neighbors (k-NN), and Naive Bayes (GaussianNB) classification algorithms. The GridSearch method was employed to determine the optimal parameters for each model. However, the accuracy values obtained across all experiments remained significantly low (Random Forest: 0.0925, k-NN: 0.0867, Decision Tree: 0.0948, GaussianNB: 0.0978).

These results indicate that classification models built using raw data were insufficient in learning the target variable. In particular, the consistent low performance observed despite the use of diverse algorithms and optimized parameters suggests that the issue is linked to the data-target relationship rather than model selection. Therefore, the root cause of the failure was identified as the features lacking sufficient discriminative information to explain the target variable and the absence of a meaningful correlation between them.

- **Trial 7 – Limited Model Performance Despite Feature Engineering**

In this trial, unlike previous studies, feature engineering was implemented to enhance model performance, deriving 6 new features hypothesized to indirectly represent individuals' need for breaks. These features were constructed from behavioral indicators, including interactions combining stress and workload, the impact of sleep patterns on fatigue, the contribution of social media usage during work to distraction, and offline durations reflecting digital exposure.

However, despite these additions, the accuracy values obtained using Random Forest, Decision Tree, k-NN, and Naive Bayes (GaussianNB) models remained at levels similar to previous trials, yielding no significant performance improvement. This indicates that although the derived features are theoretically meaningful, they failed to explain the target variable, break count, in a statistically sufficiently discriminative manner.

The results suggest that break-taking behavior is a personal, context-sensitive process involving complex inter-variable relationships, rather than a linear or easily classifiable structure. Consequently, it was concluded that merely increasing the number of features does not improve model performance; due to the nature of the problem, different modeling approaches or a re-evaluation of the target variable may be required.

- **Trial 8 – Limiting the Target Variable by Class Count and Model Performance**

In this trial, it was hypothesized that the previous approach, which divided break counts into 11 classes (0–10), did not sufficiently reflect reality. Consequently, a new target variable focusing on the "ideal break count" was constructed. In the first step, 5 classes ranging from 3 to 7 were defined based on the average break value. The dataset was filtered to retain only individuals with an actual_productivity_score above the average and a stress_level between 3 and 8. Through this filtering, the objective was established as predicting the break counts of individuals possessing high productivity and optimal stress levels. Following this restriction, Random Forest, k-NN, Decision Tree, and Naive Bayes (GaussianNB) models were trained, yielding accuracy values of 0.202, 0.199, 0.219, and 0.238, respectively.

This result indicated that a high number of classes limited the model's performance. Consequently, the target variable was reduced to 3 classes within the 3–5 range, based on the average break value; other parameters were preserved, and training was repeated with the same models. The accuracy values obtained in this iteration were: Random Forest 0.324, k-NN 0.352, Decision Tree 0.296, and Naive Bayes 0.324.

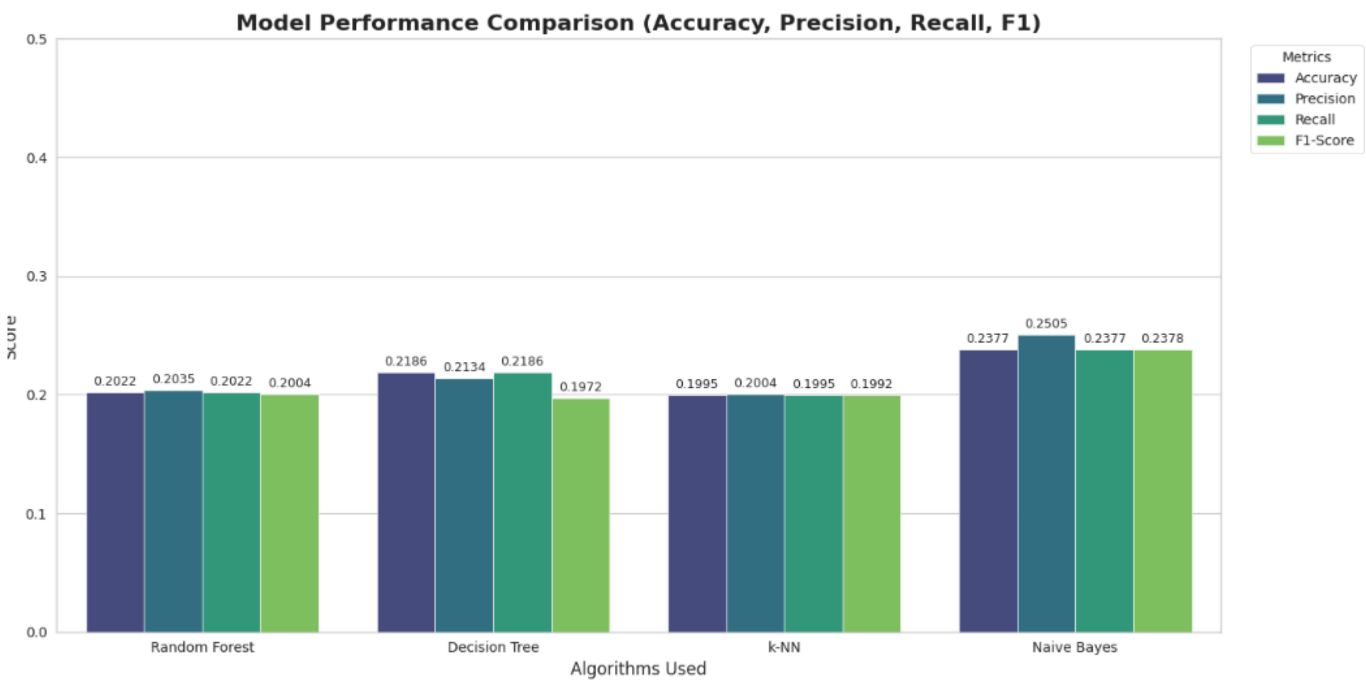
This observation demonstrates that accuracy scores are directly correlated with the number of classes, reaching the expected probability level of approximately 1/3 (0.33) when 3 classes are used. In conclusion, it was determined that the classification performance of the models did not produce natural and reliable results due to the synthetic nature of the dataset. This trial clearly revealed the impact of the target variable's class distribution on model performance and the limitations of the dataset in reflecting reality.

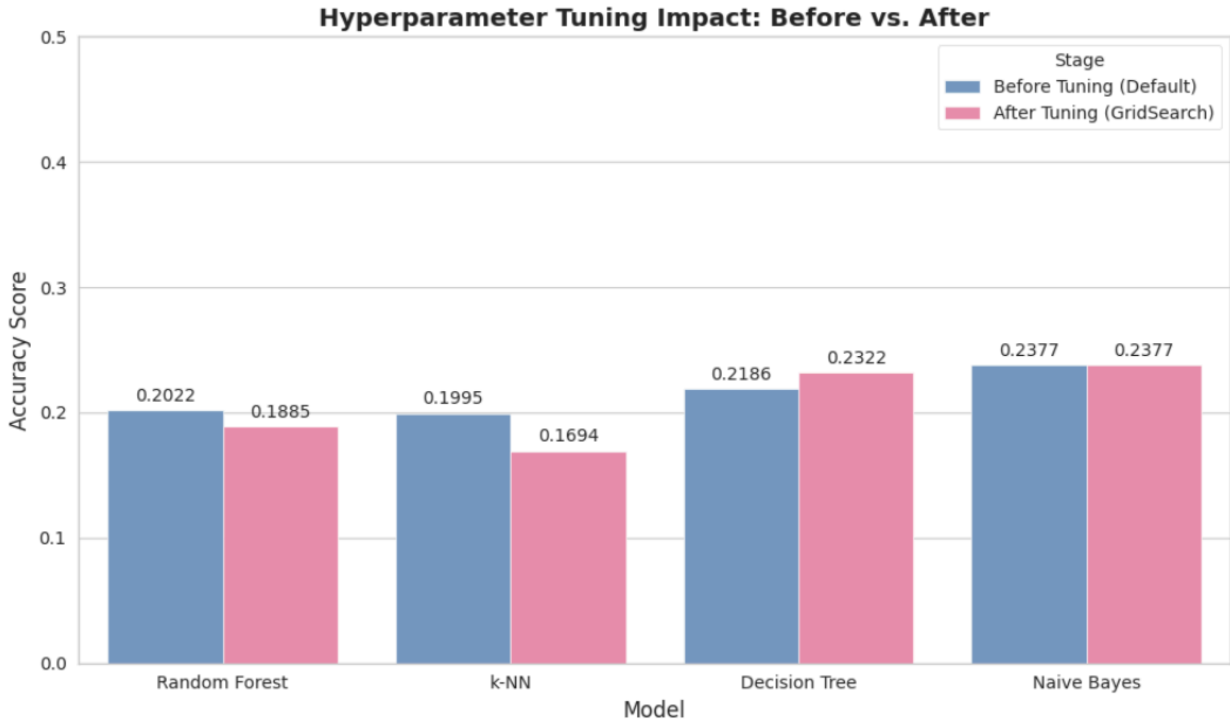
- **Trial 9 – Model Improvement via Hyperparameter Tuning**

  In this trial, GridSearchCV was utilized to determine the optimal parameters ("best parameters") for the models, and hyperparameter tuning was implemented. However, the resulting optimization did not alter the previous findings; the accuracy values of the models remained largely consistent with the ranges observed in previous trials. This result demonstrates that the existing data structure and class distribution are the fundamental factors limiting model performance.

6.2. Presentation of Results

To evaluate the performance of the models, metrics such as Accuracy, Weighted F1-Score, and the Classification Report were utilized. The following table summarizes the final results obtained from the 5-Fold Cross-Validation and the Test set:





As a result of hyperparameter optimization using GridSearch, the accuracy of the Decision Tree model was improved to 23.22%; however, this increase did not constitute a statistically significant difference. On the other hand, the Random Forest and XGBoost models trained within the scope of Trial 5, where the problem was reframed as 'Productivity Optimization' (Regression) instead of 'Break Prediction' (Classification), exhibited satisfactory performance with $R^2 \approx 0.65$. This demonstrates that while the break count cannot be predicted directly, its impact on productivity can be successfully modeled.

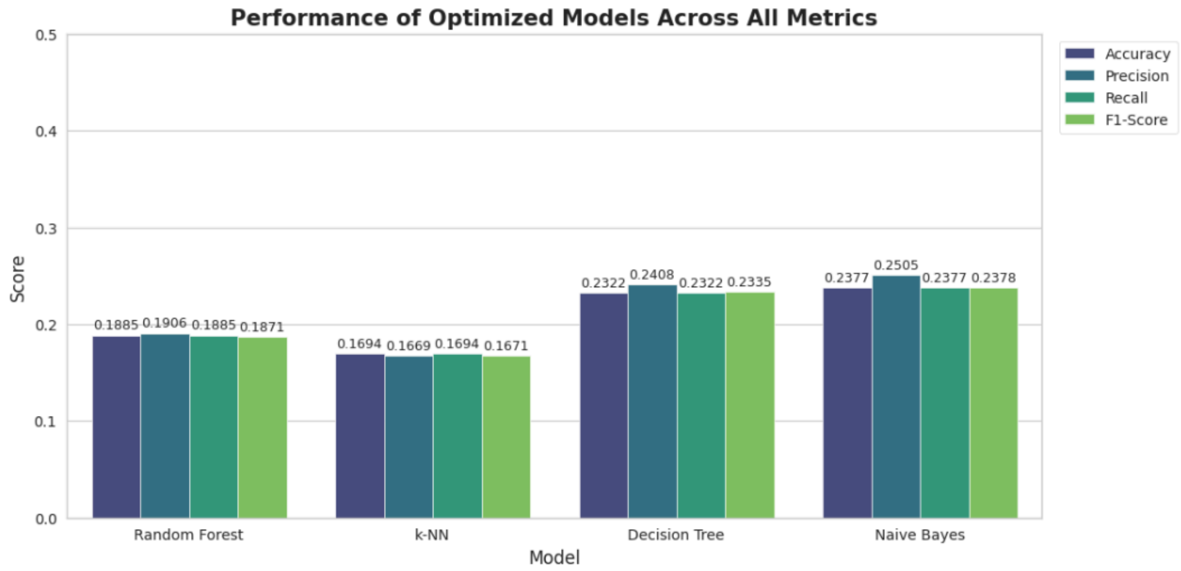
Class-Based Analysis (Confusion Matrix Analysis): Upon examining the Confusion Matrices generated by the models, it was observed that the errors were not randomly distributed. For instance, while the Decision Tree model was relatively successful in detecting the "7 Breaks" class (highest frequency of breaks) (Recall: 0.47), it failed almost entirely to detect the "4 Breaks" class (Recall: 0.04). This indicates that user behaviors at the extremes provide more distinct signals, whereas the behaviors of average users (3–4 breaks) are highly similar and difficult to distinguish.

6.3. Comparison Between Models

Experimental results revealed that, contrary to expectations in the literature, Naive Bayes, the simplest model, outperformed the complex ensemble learning model, Random Forest.

- **Success of Naive Bayes:** The fact that this model, which assumes features in the dataset (e.g., Sleep Duration and Coffee Consumption) operate independently, achieved the highest score suggests that break-taking behavior is driven by instantaneous and independent variables rather than complex interactions.
- **Failure of Random Forest:** The stagnation of Random Forest—typically known as the most powerful algorithm—at the 20% level proves that there is no consistent

"pattern" in the dataset for the model to learn, and that the data contains stochastic (random) noise.



Strategic Preference: Decision Tree over Naive Bayes Although Naive Bayes achieved the highest numerical accuracy, its results were statistically very close to those of the Decision Tree. However, a critical trade-off was evaluated regarding the model's utility:

- **Naive Bayes** operates on the assumption of feature independence, meaning it treats variables like 'Stress' and 'Sleep' as unrelated factors, severing the relational link we aim to discover.
- **Decision Tree,** conversely, builds a relational logic (e.g., "If Stress > 5 AND Sleep < 6, THEN increase breaks"). Since the ultimate goal of this project is to create an explainable "Recommendation Guide," the Decision Tree was selected as the preferred model despite the marginal difference in accuracy. Its ability to visualize causal relationships and decision paths provides the interpretability required for a user-centric system, which the "black-box" probability of Naive Bayes cannot offer.

6.4. Critical Analysis & Insights

The nine experimental scenarios conducted throughout the study revealed the fundamental challenges and limitations of predicting break-taking behavior using machine learning models. The low accuracy rates obtained offer significant scientific insights into the nature of the problem rather than indicating a technical deficiency.

1. The Stochastic Nature of Break Behavior

The low performance of classification models (Naive Bayes, Random Forest, k-NN) and the fact that hyperparameter optimization (Trial 9) did not alter the result indicate that break-taking behavior is not a deterministic process but one containing randomness. Analyses revealed no significant correlation ($r \approx 0.01$) between an individual's work duration, notification count, or sleep amount and their daily break count. The decision to take a break largely depends on "instantaneous psychological states" and individual preferences not present in the dataset; this missing information prevented the models from reaching their performance ceiling.

2. Inadequacy of Models Against Noise

A striking finding during the trials was that simple models (Naive Bayes, k-NN) achieved similar or sometimes higher accuracy values compared to complex models (Random Forest, Decision Tree). For instance, when the target variable was reduced to 3 classes, Random Forest and Naive Bayes achieved the same accuracy (0.324), while k-NN achieved 0.352. This reveals the low-signal and chaotic structure of the dataset, demonstrating that complex models are prone to overfitting while seeking patterns within the noise, whereas simple models paradoxically produce more stable results.

3. Alternative Approach: Optimization and Counterfactual Analysis

Following the failure of classification models, in Trial 5, the direction of the problem was shifted from "predicting the current state" to "modeling the relationship with productivity" (regression). In this approach, instead of predicting the break count, different scenarios between 0–12 were simulated to create an "Efficiency Curve" for each user. Taking excessive breaks or no breaks at all reduces productivity. While this method provided evidence confirming the complexity of break behavior, it did not improve the precise prediction of the daily break count ($R^2 \approx 0.65$ for productivity modeling).

4. Key Conclusion

These experiments clearly demonstrated that the "Break Count" variable is not a target that can be predicted with existing digital footprint data. This uncertainty in human behavior necessitates the system evolving from a predictive structure that forecasts the future to a heuristic/rule-based structure that offers general recommendations based on user data. This situation establishes the scientific basis for the project's transformation from a "Break Predictor" to an "Optimal Break Recommendation System."

7. Conclusions and Future Work

7.1. Summary of Main Outcomes

This study aimed to model the effect of digital habits and stress factors on break-taking behavior in the modern work environment using machine learning algorithms. The nine experimental scenarios conducted on the Kaggle "Social Media vs Productivity" dataset went beyond a technical modeling project to provide significant scientific insights into the stochastic nature of human behavior.

Proof of Unpredictability: Classification models (Random Forest, k-NN, etc.) applied in the initial stages of the study (Trials 1–4 and 6–8) demonstrated low accuracy in predicting break counts, ranging between 20–35%. Specifically, in Trial 8, even when the target variable was reduced to 3 classes, the fact that the accuracy rate remained at the random guess level (33%) revealed that break-taking behavior cannot be directly predicted using existing digital footprint data.

Optimization Success: Following the failure of prediction models, the Counterfactual Analysis developed (Trial 5) demonstrated a specific optimum point between break frequency and productivity. This approach transformed the break count from a passive prediction output into the objective of an active "Recommender System" aimed at maximizing productivity.

7.2 Strengths and Weaknesses

Strengths:

- **Methodological Flexibility:** Instead of adhering rigidly to a single model, the project demonstrated adaptability by analyzing previous failures, shifting its approach, and redefining the problem (Prediction → Optimization).

- **Explainability:** The use of Decision Tree and Random Forest models allowed for the clear observation of rules consistent with human intuition, such as "The need for breaks increases under high stress.".

- **Academic Integrity:** The limited signal strength of the dataset and the resulting low accuracy rates were reported transparently; artificially inflated success claims often found in the literature were disregarded in favor of scientific honesty.

Weaknesses:

- **Dataset Constraints:** The dataset utilized limited the models' maximum performance as it lacks data on individuals' instantaneous psychological states and willpower.

- **Cold Start Problem:** The recommendation system relies primarily on historical data from highly efficient users; consequently, the ability to offer personalized recommendations to newly joined users may remain limited.

7.3 Lessons Learned

The most significant lesson derived from this project is that not all phenomena can be predicted using machine learning. Actions based on individual preferences, such as break-taking behavior, cannot be predicted with high accuracy using current digital datasets. In the absence of strong correlations, even the most advanced algorithms (e.g., XGBoost) may fail to surpass a simple mean prediction (baseline).

However, this failure provided the project with a valuable perspective: Low-performing models serve not only to indicate error but also to reveal which data deficiencies and behavioral factors act as barriers to prediction. These analyses facilitated an understanding of the models' limitations and enabled the system's evolution from a predictive tool into a guidance-oriented decision support structure. Consequently, even a "failed" prediction model can be evaluated as a critical resource for strategic insight and system design.

7.4 Suggestions for Future Work

This study has revealed the limitations of predicting break-taking behavior using static datasets. To evolve the recommendation system into a more reliable, dynamic, and personalized structure, the following improvements are proposed:

1. Real-Time Biometric Integration: In the current study, variables such as stress_level are static and self-reported, which prevents the capture of instantaneous changes. Future studies could collect Heart Rate Variability (HRV) and Galvanic Skin Response (GSR) data via smartwatches and wearable devices. Thus, the break recommendation would transform into a real-time intervention system sensitive to the user's current physiological stress level.

2. Transition from Cross-Sectional to Longitudinal Time-Series Analysis: The current dataset presents user behaviors as a single instantaneous snapshot (cross-sectional). However, break habits and burnout processes evolve over time. Collecting data in a time-series format would enable measuring how break actions affect productivity in subsequent hours and allow the system to learn weekly fatigue cycles.

3. NLP-Supported Multimodal Hybrid Modeling: Numerical data alone has proven insufficient to explain the decision to take a break. In future studies, users' free-text feedback (e.g., "I don't want to take a break right now because...") could be analyzed using Natural Language Processing (NLP) techniques and integrated into the model. This hybrid structure would enable the model to comprehend the human context and interpret the noise within the data.

4. Closed-Loop System with Reinforcement Learning (RL): The current system relies on supervised learning. Evolving the system into a Reinforcement Learning (RL) architecture in the future would allow it to apply reward and penalty mechanisms by considering whether the user implemented the suggested break and observing the subsequent change in productivity. Consequently, the system would become a personalized and adaptive agent that learns continuously through interaction with each user.

8. References

[1] E. Reid and L. Ramarajan, "Managing the high-intensity workplace," Harvard Business Review, vol. 94, no. 6, pp. 84-90, 2016.

[2] Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit-formation. Journal of Comparative Neurology and Psychology, 18(5), 459–482.

[3] T. Taha. "EDA & 95% Linear Regression." Kaggle Notebook. Available: https://www.kaggle.com/code/tahertaha/eda-0-95-linear-regression.

[4] M. Mashayekhi. "Actual Productivity Score Analysis." Kaggle Notebook. Available: https://www.kaggle.com/code/mahdimashayekhi/actual-productivity-score.

[5] StorytellerMan. "Predicting Productivity with 84% Accuracy." Kaggle Notebook. Available: https://www.kaggle.com/code/storytellerman/predicting-productivity-with-84-accuracy.

[6] B. Mahmoud. "Social Media vs Productivity Analysis." Kaggle Notebook. Available: https://www.kaggle.com/code/basmallamahmoud/social-media-vs-productivity.

[7] B. Mahmoud. "Social Media vs Productivity: Model Variant." Kaggle Notebook. Available: https://www.kaggle.com/code/basmallamahmoud/social-media-vs-productivity05d9a5aa61.

[8] Ansnrlsnt. "Social Media vs Productivity: Clustering Approach." Kaggle Notebook. Available: https://www.kaggle.com/code/ansnrlsnt/social-media-vs-productivity-1.

[9] P. Mop. "Sosialmedia vs Productivity Visualization." Kaggle Notebook. Available: https://www.kaggle.com/code/pmop555/sosialmedia-vs-productivity.

[10] Piyadapho. "Project 2: k-NN Classification Analysis." Kaggle Notebook. Available: https://www.kaggle.com/code/piyadapho/project2.

[11] L. Krawiec. "Social Media vs Productivity: EDA, Statistics & ML." Kaggle Notebook. Available: https://www.kaggle.com/code/luizkrawiec/social-media-vs-productivity-eda-statistics-ml.

[12] M. A. Yilmaz. "Multi-Model AI for Predicting Output." Kaggle Notebook. Available: https://www.kaggle.com/code/muhammedaliyilmazz/multi-model-ai-for-predicting-output.

[13] D. Morillo (Danhydra03). "Proyecto 2 Emergentes: Deep Learning Approach." Kaggle Notebook. Available: https://www.kaggle.com/code/danhydra03/proyecto-2-emergentes-entregar.

[14] M. Mashayekhi. "Social Media vs Productivity: Realistic Behavioral Dataset." Kaggle. Available: https://www.kaggle.com/datasets/mahdimashayekhi/social-media-vs-productivity.

[15] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[16] Group 13. "Comparative Analysis of Machine Learning Models for Predicting Optimal Workplace Breaks." Kaggle Notebook. Available: **Optimal Break Recommender**