Evaluating the Determinants of Absenteeism at Workplace Using Machine Learning Algorithms

ISBN 978-1-943295-24-1

J. Pooja Sree
P. Siva Naga Lakshmi
KLEF Deemed to be University
(2301510227@kluniversity.in)
(2301510046@kluniversity.in)

In large organizations, absenteeism challenges productivity, employee morale, and operational efficiency. This study examines absenteeism determinants using machine learning on a sample of 2,000 entries from the Kaggle absenteeism dataset focusing on key factors like age, department, and service length. Among various models, the Bagging Random Forest achieved the highest accuracy with an MAE of 19.14 and R of 0.747. Key predictors identified include age job role and length of service. Findings suggest that wellness programs, flexible work and structured feedback can help manage absenteeism by offering large organizations actionable insights to enhance productivity and employee engagement.

Keywords: Absenteeism, Machine learning, Bagging Random Forest, MAE, Productivity

1. Introduction

In the current landscape of increasingly complex work environments, absenteeism has emerged as a growing concern for businesses worldwide. With rising job demands, employees often face pressures that extend beyond traditional job roles, contributing to heightened stress and the temptation to take time away from work. Factors such as mental health concerns, evolving family dynamics, and the shift toward hybrid and remote work have all contributed to the increase in absenteeism rates across various industries. The pressure to meet organizational goals in fast-paced settings has also led to higher levels of burnout, making absenteeism a growing issue. Organizations now face the direct effects of absenteeism more frequently, leading to productivity declines, difficulties in maintaining morale, and challenges with consistent workload management. Furthermore, the shift toward remote and hybrid work introduces unique difficulties. In these new work settings, measuring engagement and productivity can be challenging, potentially resulting in subtle forms of absenteeism, such as disengagement or frequent breaks. Overall, the rise in absenteeism highlights the need for effective strategies to address its underlying causes to ensure both organizational effectiveness and employee well-being.

Need for the Study

Workplace absenteeism is a multifaceted issue that impacts businesses at both operational and cultural levels. When absenteeism rates are high, organizations bear significant costs financially, through hiring temporary replacements, and operationally, as frequent absences disrupt workflow and create team imbalances. At a cultural level, absenteeism can lead to lower morale and engagement among the remaining workforce, as they may feel burdened with extra responsibilities, leading to a cycle of potential burnout. In addition to these logistical and cultural challenges, absenteeism can contribute to negative organizational culture, diminishing trust between employees and management. When absences become common, team dynamics often shift, leading to potential resentment or communication barriers. Ultimately, understanding absenteeism's root causes is essential for organizations committed to foster supportive, healthy work environments. Various factors contribute to absenteeism, including employee health, satisfaction, workload, and the overall organizational culture. However, traditional analysis methods have limitations in accurately capturing the complexity of these influences, often overlooking intricate relationships and less obvious factors. Therefore, there is a pressing need for more sophisticated analytical tools capable of providing comprehensive insights into absenteeism and its drivers. Considering the current shift towards data-driven decision-making in business, using advanced machine learning (ML) techniques provides a promising way to delve deeper into absenteeism factors. ML algorithms allow organizations to analyse large, complex datasets, uncover patterns, and predict absenteeism with greater accuracy than traditional methods. This study aims to bridge this gap, using ML to uncover absenteeism's determinants and help organizations develop targeted strategies to address it effectively.

2. Review of Literature

Machine learning models have been widely researched for predicting absenteeism across various sectors, with notable contributions in healthcare, education, workplace, and social settings.

In healthcare, Salazar et al. (2020) explored outpatient absenteeism in Brazil, using Logistic Regression, Random Forest, and Decision Trees. Their study found that the Decision Tree model was most effective for predicting patient attendance, though challenges remained in forecasting no-shows. The research emphasized the need for including additional variables to enhance

healthcare efficiency (Luiz Henrique Salazar 1, 2020). Similarly, Dalia Alzu'bi et al. (2024) focused on nurse absenteeism at King Abdullah University Hospital, where an Artificial Neural Network (ANN) achieved 82% accuracy. This work underscores the potential of ANN in improving operational efficiency and predicting absenteeism in healthcare settings (Dalia Alzu'bi1 · Mwaffaq El Heis2 · Anas Ratib Alsoud3 · Mothanna Almahmoud8 · Laith Abualigah3, 2024).

Herrera Montano et al. (2020) conducted a review of 58 studies on machine learning techniques for absenteeism and temporary disability prediction, highlighting the effectiveness of ANNs for predictive accuracy. They observed trends in absenteeism research from Brazil and India and called for further studies with larger datasets to strengthen the findings (Isabel Herrera Montano, 2020). Likewise, Ajmi (2019) studied absenteeism in a Brazilian courier company and found that the Decision Tree model outperformed other algorithms, achieving 83.33% accuracy and an AUC of 0.834. The study suggested that future research should explore external factors and consider industry-specific variables to improve model generalization (Ajmi, 2019).

In education, Fernandes and Chiavegatto Filho (2021) applied machine learning to predict absenteeism among teachers in São Paulo public schools. Their study found that the best-performing model, ANN, achieved an AUC of 0.79, suggesting the potential for machine learning to improve teacher management and reduce absenteeism (Fernando Timoteo FernandesI, 2021). Similarly, Mukli and Rista (2022) examined student absenteeism at the University Aleksandër Moisiu in Albania, identifying factors such as family income and parental education. However, their study's focus on a single institution limits the generalizability of the findings (L. Mukli, 2022).

Bowen et al. (2022) investigated student absenteeism and achieved 90.2% classification accuracy, emphasizing the role of socio-emotional learning (SEL) in reducing absenteeism. Their research advocates for integrating SEL into curricula to address absenteeism effectively (Francis Bowen, 2022). Similarly, Park (2024) studied absenteeism among platform workers in South Korea and found higher absenteeism rates among replacement drivers, underlining the need for targeted interventions in non-traditional work settings (Heejoo Park, 2024).

In workplace settings, Araujo et al. (2019) proposed a hybrid model combining fuzzy logic and neural networks for predicting absenteeism, providing HR managers with valuable tools to reduce financial losses, though they acknowledged limitations related to dataset size and model complexity (Vanessa S. Araujo, 2019). Lawrance et al. (2021) introduced a decision support system in Belgium, using cost-sensitive learning to predict absenteeism. The study showed that predictive analytics could reduce absenteeism and allocate resources effectively, although the findings are limited to the Belgian context (Natalie Lawrance, 2021).

Vakili et al. (2024) examined how knowledge management infrastructures, such as motivational salary systems and knowledge-sharing cultures, can help reduce mental absenteeism and improve employee engagement. Their study suggests that these systems play a crucial role in absenteeism reduction (Ali Akbar Vakili, 2024). Trivedi (2018) enhanced a neural network model for absenteeism prediction by incorporating pruning techniques, which significantly improved the model's accuracy from 34% to 58%. However, the study noted that further exploration of dataset biases and limitations was needed (Trivedi, 2018).

Nath et al. (2022) developed a hybrid model integrating machine learning to predict absenteeism, coupled with a web-based tool to assist managers in decision-making. The study made a significant contribution, though concerns about the model's real-world application and generalizability remain (Gopal Nath, 2022).

Jayme (2013) compared the predictive accuracy of various machine learning methods, including Random Forest, Support Vector Machines (SVM), and ANN, with ANN outperforming others, achieving 77% accuracy. The study highlighted the importance of feature importance analysis in predicting absenteeism, though its real-world application was not deeply discussed (Alejandra Jayme, 2013). Ali Shah (2020) introduced a deep neural network (DNN) model for workplace absenteeism, achieving 90.6% accuracy, but noted the limitations due to the small dataset size (740 instances) used for training (Syed Atif Ali Shah, 2020).

Kamepalli et al. (2020) analyzed absenteeism's impact on workplace discipline using machine learning classifiers like SMO and Multi-layer Perceptron, which achieved 100% accuracy in predicting disciplinary failures. However, the study's scope was limited by the small number of classifiers and the lack of consideration for external factors (Dr. Sujatha Kamepalli, 2020).

Naganaidu (2022) explored the use of multinomial logistic regression alongside Decision Tree and K-Nearest Neighbor models to predict absenteeism, revealing challenges in predicting specific absenteeism classes. The study also emphasized the importance of metrics like precision and recall evaluating model performance and suggested future research could benefit from comparisons with similar studies (Demudu Naganaidu, 2022).

3. Objectives of the Research

The core objective of this research is to identify and analyse the main factors influencing absenteeism within workplace settings and to create predictive models that offer actionable insights for management. By understanding absenteeism drivers in depth, organizations can develop interventions tailored to enhance employee well-being, job satisfaction, and overall engagement.
The specific aims of this study include:

1. Evaluating Machine Learning Algorithms' Performance: Testing various machine learning algorithms to assess their accuracy in predicting absenteeism based on selected predictors.
2. Analysing Relationships Between Predictors and Absenteeism: Investigating how factors like employee age, job title, length of service, and department affect absenteeism rates.
3. Providing Targeted Recommendations: Offering data-backed recommendations to guide organizations in managing and reducing absenteeism effectively.

4. Methodology

Research Design

The study employs a predictive and analytical research design to examine factors influencing workplace absenteeism. The primary goal is to develop predictive models that identify significant absenteeism determinants by analyzing demographic and job-related features in the dataset.

Sample Design

The original dataset named 'Absenteeism' from Kaggle consisted of 8,336 records across 13 variables. To improve computational efficiency and relevance, a refined sample of 2,000 records with 10 selected variables was created. Irrelevant personal identifiers (e.g., "*Employee number", "name", "surname*") were excluded to focus on variables most pertinent to absenteeism, such as age, length of service, job title, department, and absent hours.

Data Collection

The Kaggle 'Absenteeism' dataset includes comprehensive details about employees, such as age, length service, job title, department, and absence hours, providing insights into workforce demographics and job-specific patterns. These factors were chosen for their potential relevance in predicting absenteeism trends and understanding organizational dynamics.

Data Analysis

In this study, all data analysis was performed using Python, a versatile programming language commonly used in data science and machine learning. The dataset was processed and cleaned using Python libraries such as Pandas and NumPy. For machine learning model implementation, Scikit-learn and other specialized libraries like XG Boost and TensorFlow were utilized. The entire process, including data cleaning, feature encoding, model training, and evaluation, was conducted within the Python environment, ensuring reproducibility and efficiency, Data analysis was performed using the following procedures.

1. Data Preprocessing

- **Handling Missing Values**

Missing values in the dataset were imputed to ensure data integrity and improve model robustness.

- **Categorical Encoding**

Label encoding was applied to categorical fields like job titles and departments, allowing for compatibility with machine learning models.

- **Normalization**

  Standardization was applied to numerical features, helping models train effectively and ensuring consistent performance.

2. Model Selection and Evaluation

To fulfill Objective [1] (Evaluating Machine Learning Algorithms Performance):
Several machine learning models were implemented to predict absenteeism, including traditional regression, bagging, and boosting models.

- **Bagging Models:** Bagging Regressor, Bagged KNN Regressor, Bagging Decision Tree, Bagging Random Forest, Bagging Extra Tree Regressor.
- **Boosting Models:** Gradient Boosting, AdaBoost.
- **Traditional Models:** Linear Regression, Support Vector Regressor, K-Nearest Neighbors.
- **Performance Metrics:**

Models were evaluated based on Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 scores. These metrics enabled a comparison of each model's accuracy and generalization ability.

3. Feature Importance

To fulfill Objective [2] (Analysing Relationships Between Predictors and Absenteeism):
The analysis of feature importance was conducted using the XG Boost model. This analysis identified key predictors that most significantly impacted absenteeism. The F-score of each feature indicated its contribution to the model's predictions, with *age* and *length of service*, Job *Title* and follows emerging as the most influential factors.

4. Descriptive Analysis

Summary statistics, including the mean and standard deviation of variables such as *age* and *tenure*, were calculated to assess central tendencies and data dispersion. Additionally, department-based absence patterns were analysed to provide a deeper understanding of how specific job characteristics correlate with absenteeism.

5. Results and Discussion

The findings provide a comprehensive evaluation of machine learning models for predicting absenteeism, highlighting their strengths and limitations. Key performance metrics such as MAE, RMSE, and R^2 were used to compare traditional, bagging, and boosting methods. Feature importance analysis further identified critical predictors, offering actionable insights for

managing absenteeism effectively. The discussion begins with traditional models, progressing to ensemble techniques, to outline their practical implications and predictive capabilities.

Traditional Models

Linear Regression

Linear Regression is a foundational statistical technique used to establish and quantify the relationship between a dependent variable and one or more independent variables, making it a widely used method in predictive modeling

.

```
In [39]: # Build and fit the linear regression model
         model = LinearRegression()
         model.fit(X_train, y_train)

Out[39]: ▾ LinearRegression
         LinearRegression()
```

Code Snippet 1 *Linear Regression Model*

Table 1*Performance Metrics for Linear Regression Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	21.4517	768.3934	27.7199	1.688125	0.67748	Optimal
Testing	21.41302	743.04819	27.2589103	2.23342	0.70284	Optimal

Support Vector Regressor

SVR is a machine learning model based on Support Vector Machines, primarily used for regression tasks. It attempts to fit the best line or hyperplane within a margin of tolerance for a given dataset, making it robust for outlier handling and complex data patterns.

```
In [45]: # SV regressor
         from sklearn.svm import SVR
         model= SVR()
         model.fit(X_train,y_train)

Out[45]: ▾ SVR
         SVR()
```

Code Snippet 2 *Support Vector Regressor Model*

Table 2 *Performance Metrics Support Vector Regressor Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	22.427	862.839	29.374	1.425611	0.63784	Slight Overfitting
Testing	22.306	843.45	29.0422	1.759079	0.66268	Slight Overfitting

KNN Regressor

KNN Regressor is a non-parametric model that makes predictions based on the average output of the 'k' nearest data points in the feature space. It is simple yet effective for capturing non-linear relationships in data.

```
In [51]: # KNN
         from sklearn.neighbors import KNeighborsRegressor
         model= KNeighborsRegressor()
         model.fit(X_train,y_train)

Out[51]: ▾ KNeighborsRegressor
         KNeighborsRegressor()
```

Code Snippet 3 *KNN Regressor Model*

Table 3*Performance Metrics for KNN Regressor Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	17.477	531.68	23.0583	1.202015	0.77683	Optimal
Testing	20.206	706.075	26.572	1.202015	0.7176	Optimal

Decision Tree Regressor

Decision Tree Regressor uses a tree-like structure to make decisions by splitting the data into subsets based on feature values. It is interpretable and capable of capturing complex non-linear patterns but is prone to overfitting.

```
In [56]: #decision tree regressor # variance reduction method is used in backend
         from sklearn.tree import DecisionTreeRegressor
         model= DecisionTreeRegressor()
         model.fit(X_train,y_train)

Out[56]: DecisionTreeRegressor
         DecisionTreeRegressor()
```

Code Snippet 4 *Decision Tree Regressor Model*

Table 4 *Performance Metrics for Linear Regression Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	0.3266	0.85337	0.9237	1.54145	0.99964	High Overfitting
Testing	28.12	1369.51	37.007	1.54145	0.4523	High Overfitting

Random Forest Regressor

Random Forest Regressor is an ensemble model that constructs multiple decision trees during training and averages their outputs. It enhances accuracy and reduces overfitting by leveraging the power of bagging and random feature selection.

```
In [62]: # Random forest regressor
         from sklearn.ensemble import RandomForestRegressor
         model= RandomForestRegressor()
         model.fit(X_train,y_train)

Out[62]: RandomForestRegressor
         RandomForestRegressor()
```

Code Snippet 5 *Random Forest Regressor Model*

Table 5 *Performance Metrics for Random Forest Regressor Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	7.85	112.02664	10.584263	1.385211	0.952	Moderate Overfitting
Testing	21.316	755.847	27.49268	1.385211	0.697721	Moderate Overfitting

Bayesian Regression

Bayesian Regression incorporates prior knowledge and uncertainty in the model by applying Bayesian principles. It provides a probabilistic interpretation, making it particularly useful when dealing with small datasets or prior domain knowledge.

```
In [69]: # Bayesian Regression Model
         from sklearn.linear_model import BayesianRidge
         model= BayesianRidge()
         model.fit(X_train,y_train)

Out[69]: BayesianRidge
         BayesianRidge()
```

Code Snippet 6 *Bayesian Regression Model*

Table 6*Performance Metrics for Bayesian Regression Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	21.45	768.417	27.72	1.6864	0.6774	Optimal
Testing	21.4099	742.829	27.25488	2.2311	0.70292	Optimal

Boosting Models

XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is an advanced ensemble learning algorithm designed to enhance predictive accuracy by leveraging the principles of gradient boosting. It is known for its speed, efficiency, and ability to handle large datasets effectively.

```
In [74]: import xgboost as xgb
         from xgboost import XGBRegressor
         model = XGBRegressor()
         model.fit(X_train,y_train)

Out[74]: XGBRegressor
         XGBRegressor(base_score=None, booster=None, callbacks=None,
                      colsample_bylevel=None, colsample_bynode=None,
                      colsample_bytree=None, device=None, early_stopping_rounds=None,
                      enable_categorical=False, eval_metric=None, feature_types=None,
                      gamma=None, grow_policy=None, importance_type=None,
                      interaction_constraints=None, learning_rate=None, max_bin=None,
                      max_cat_threshold=None, max_cat_to_onehot=None,
                      max_delta_step=None, max_depth=None, max_leaves=None,
                      min_child_weight=None, missing=nan, monotone_constraints=None,
                      multi_strategy=None, n_estimators=None, n_jobs=None,
```

Code Snippet 7 *XGBoost Regressor Model*

Table 7*Performance Metrics for XGBoost Regressor Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	7.45741	102.3731	10.1179	1.5751821	0.9570311	Optimal
Testing	21.840449	842.34195	29.023127	1.5751821	0.6631311	Optimal

AdaBoost Regressor

AdaBoost Regressor is a boosting algorithm that improves model accuracy by focusing on data points that were previously mis predicted. It iteratively adjusts the weights of weak learners to enhance predictive performance.

```
In [94]: # AdaBoostRegressor
         from sklearn.ensemble import AdaBoostRegressor
         # Initialize the base estimator
         base_estimator = DecisionTreeRegressor(max_depth=3)
         # Initialize the AdaBoost Regressor with Decision Tree as base estimator
         ada_regressor = AdaBoostRegressor(base_estimator=base_estimator, n_estimators=50, learning_rate=0.1, ran
         # Fit the model
         ada_regressor.fit(X_train, y_train)

         C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:166: FutureWarning: `base_estimator
         ` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
           warnings.warn(

Out[94]: AdaBoostRegressor
         base_estimator: DecisionTreeRegressor
           DecisionTreeRegressor
```

Code Snippet 8 *AdaBoost Regressor Model*

Table 8*Performance Metrics for AdaBoost Regressor Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	20.26134	680.1569	24.11334	1.7192	0.755947	Slight Overfitting
Testing	19.1427	631.0012	26.4891	1.7192	0.719386	Slight Overfitting

Bagging Models

Bagging Regressor

Bagging Regressor, or Bootstrap Aggregating, is an ensemble technique that improves the stability and accuracy of machine learning algorithms by combining predictions from multiple models trained on different subsets of the data.

```
In [83]: # Fit the model
         bagged_svm.fit(X_train, y_train)

         C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:166: FutureWarning: `base_estimator
         ` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
           warnings.warn(

Out[83]: BaggingRegressor
         base_estimator: SVR
           SVR
```

Code Snippet 9 *Bagging Regressor Model*

Table 9*Performance Metrics for Bagging Regressor Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	22.3007	857.1112	29.2885	1.4112	0.6402	Slight Overfitting
Testing	22.2249	826.8671	28.7527	1.499	0.6635	Slight Overfitting

Bagging KNN Regressor

Bagging KNN Regressor applies the bagging approach to the KNN model, enhancing its performance by aggregating predictions from multiple KNN models trained on different data subsets. This helps in reducing variance and improving robustness.

```
In [105]: # Bagged KNN for Regression
          from sklearn.ensemble import BaggingRegressor
          from sklearn.neighbors import KNeighborsRegressor
          # Initialize the base estimator (KNN for regression)
          knn = KNeighborsRegressor(n_neighbors=5)
          # Initialize the Bagging Regressor with KNN as base estimator
          bagged_knn_regressor = BaggingRegressor(base_estimator=knn, n_estimators=10, random_state=42)
          # Fit the model
          bagged_knn_regressor.fit(X_train, y_train)

          C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:166: FutureWarning: `base_estimator
          ` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
            warnings.warn(

Out[105]: BaggingRegressor
          base_estimator: KNeighborsRegressor
            KNeighborsRegressor
```

Code Snippet 10 *Bagging KNN Regressor Model*

Table 10 *Performance Metrics for KNN Regressor Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	18.2074	585.8521	24.1898	1.4557	0.7541	Moderate Overfitting
Testing	21.4434	721.1879	26.8543	1.4557	0.6851	Moderate Overfitting

Bagging Decision Tree

Bagging Decision Tree Regressor leverages ensemble learning by training multiple decision trees on various data subsets. It mitigates the overfitting issues comm on with individual decision trees and improves overall predictive performance.

```
In [89]: # Initialize the base estimator (Decision Tree for regression)
         tree = DecisionTreeRegressor()
         # Initialize the Bagging Regressor with Decision Tree as base estimator
         bagged_tree = BaggingRegressor(base_estimator=tree, n_estimators=10, random_state=42)
         # Fit the model
         bagged_tree.fit(X_train, y_train)

         C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:166: FutureWarning: `base_estimator
         ` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
           warnings.warn(

Out[89]: BaggingRegressor
         base_estimator: DecisionTreeRegressor
           DecisionTreeRegressor
```

Code Snippet 11 *Bagging Decision Tree Model*

Table 11*Performance Metrics for Bagging Decision Tree Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	9.4651	145.6448	12.0684	1.4071	0.9388	High Overfitting
Testing	22.0113	804.936	28.3639	1.4071	0.678	High Overfitting

Bagging Random Forest

Bagging Random Forest Regressor combines the advantages of random forests and bagging. It uses multiple random forest models trained on different subsets to enhance predictive accuracy and reduce overfitting further.

```
In [99]: # Bagging Random Forest Regressor
         from sklearn.ensemble import BaggingRegressor, RandomForestRegressor
         # Initialize the base estimator (Random Forest)
         base_estimator = RandomForestRegressor(n_estimators=100, max_depth=5, random_state=42)
         # Initialize the Bagging Regressor with Random Forest as base estimator
         bagging_rf_regressor = BaggingRegressor(base_estimator=base_estimator, n_estimators=10, random_state=42)
         # Fit the model
         bagging_rf_regressor.fit(X_train, y_train)

         C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:166: FutureWarning: `base_estimator
         ` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
           warnings.warn(

Out[99]: BaggingRegressor
         base_estimator: RandomForestRegressor
           RandomForestRegressor
```

Code Snippet 12 *Bagging Random Forest Model*

Table 12*Performance Metrics for Bagging Random Forest Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	18.4587	581.4355	24.1133	1.3766	0.9559	Low Error
Testing	19.1427	631.0012	25.1189	1.3766	0.7476	Low Error

Bagging Extra Trees

Bagging Extra Trees Regressor utilizes extremely randomized trees in a bagging framework. This technique introduces additional randomness in tree splits, improving robustness and predictive accuracy.

```
In [126]: #ExtraTreeRegressor
          from sklearn.ensemble import ExtraTreesRegressor
          # Initialize the Extra Trees Regressor
          extra_trees_regressor = ExtraTreesRegressor(n_estimators=100, random_state=42)
          # Fit the model
          extra_trees_regressor.fit(X_train, y_train)

Out[126]: ExtraTreesRegressor
          ExtraTreesRegressor(random_state=42)
```

Code Snippet 13 *Bagging Extra Trees Model*

Table 13*Performance Metrics for Bagging Extra Tree Model on Training and Testing Sets*

Data Split	MAE	MSE	RMSE	MAPE	R^2	Fitting Status
Training	0.0327	0.0853	0.9238	1.4115	0.9996	Extremely Low Error
Testing	21.9234	820.4318	28.6416	1.4115	0.6719	Extremely Low Error

Interpretation

The performance metrics for various models (Tables 1–13) reveal distinctive capabilities in predicting absenteeism. Linear Regression and Bayesian Regression (Tables 1 and 6) exhibit balanced performance across training and testing, indicating effective capture of linear patterns without substantial overfitting. Support Vector Regressor (SVR) (Table 2) displays slight overfitting, suggesting robustness in handling non-linear patterns, though tuning could enhance generalization. K-Nearest Neighbors (KNN) (Table 3) achieves low errors on both training and testing sets, demonstrating an optimal fit that effectively captures non-linear trends in absenteeism data. However, Decision Tree Regressor (Table 4) suffers from high overfitting, with near-perfect training scores but significantly reduced testing accuracy, highlighting a lack of generalizability.

Ensemble methods improve stability and performance, as seen with Random Forest Regressor (Table 5), which shows moderate overfitting but achieves high training accuracy and reasonable testing performance, benefitting from reduced variance. XGBoost Regressor (Table 7) stands out with optimal fitting on training data and robust testing performance, effectively capturing complex data patterns with minimal overfitting, making it one of the strongest models for this dataset. Similarly, AdaBoost Regressor (Table 8) balances performance with slight overfitting and may further improve with tuning. Among the Bagging Models (Tables 9–13), Bagging Random Forest (Table 12) yields the best combination of accuracy and stability, showcasing lower error rates and strong predictive capability by leveraging the ensemble effect. Overall, ensemble models like Random Forest, XGBoost, and Bagging Random Forest emerge as superior for this absenteeism dataset, effectively balancing bias and variance to enhance predictive power.

Feature Importance Plot

The XGBoost model was selected based on its lower Akaike Information Criterion (AIC) value, which indicates superior model performance and suitability for the dataset. Subsequently, feature importance analysis was conducted using the XGBoost model to identify the key predictors influencing absenteeism.

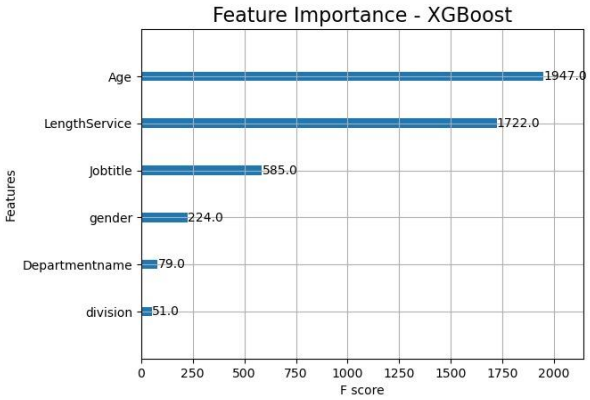

Figure 1 *Feature Importance Plot for XG Boost*

Interpretation

From the Figure (1) XGBoost model's feature importance plot reveals that age and length of service are the most significant predictors of absenteeism, with F-scores of 1947 and 1722, respectively. This suggests that age and tenure strongly influence absenteeism rates, potentially due to factors like age-related health or employee burnout. Job title (F-score: 585) also plays a role, indicating that absenteeism varies across different job responsibilities. Gender (F-score: 224) has a moderate impact, reflecting potential work-life balance or health-related factors. Department name (F-score: 79) and division (F-score: 51) are less influential, implying that departmental or organizational structure has minimal effect on absenteeism. These findings suggest management should focus on age and tenure when developing absenteeism reduction strategies.

Summary

- **Model Performance**: The Bagging Random Forest model delivered the most accurate predictions, achieving an MAE of 19.14 and an R^2 score of 0.747. This model effectively balanced accuracy and generalization, while other models, such as Gradient Boosting, showed overfitting on training data, and simpler models, like Linear Regression, exhibited underfitting on complex patterns in absenteeism behaviour.
- **Significant Predictors**: *Age*, *Length service*, *Job title* and department emerged as strong predictors of absenteeism, highlighting that experience and job characteristics are influential in determining absence patterns.

Recommendations

To fulfill Objective [3] (Providing Targeted Recommendations):

The analysis of absenteeism determinants provides actionable insights that can aid organizations in effectively managing and reducing absenteeism. By leveraging the results from data-backed models, organizations can

1. Identify key drivers of absenteeism within their workforce.
2. Develop targeted intervention strategies, such as enhancing employee well-being programs or adjusting workloads to improve job satisfaction.
3. Prioritize resources toward areas with the highest impact on attendance, thereby optimizing productivity and reducing costs.

These targeted recommendations provide a practical framework for decision-makers to proactively address absenteeism, aligning organizational strategies with the data-driven insights obtained from machine learning analysis.

6. Conclusion

This current study highlights the importance of leveraging machine learning algorithms to address workplace absenteeism, a pervasive challenge impacting organizational productivity and employee well-being. By analyzing key predictors such as age, job title, length of service, and department, the research underscores the multifaceted nature of absenteeism and the need for tailored interventions.

Among the models evaluated, the Bagging Random Forest Regressor emerged as the most effective, demonstrating a strong balance between accuracy and generalization. Other models, such as Gradient Boosting and AdaBoost, offered valuable insights but showed tendencies to overfit or underperform in specific contexts. The results validate the potential of advanced predictive techniques, offering actionable recommendations for organizations to proactively manage absenteeism.

However, the study also acknowledges limitations, including potential biases in data and varying model performance across different scenarios. Future research could explore integrating additional predictors or deploying hybrid models to enhance predictive capabilities. Overall, the findings provide a robust foundation for organizations to implement data-driven strategies, fostering healthier work environments and mitigating the adverse effects of absenteeism.

7. References

1. Ajmi, S. Q. (2019). Predicting Absenteeism at Work Using Machine Learning Algorithms. MJPS, 9.
2. Alejandra Jayme, P. D. (2013). Comparison of Machine Learning Methods for Predicting Employee Absences. Preprint Series of the Engineering Mathematics and Computing Lab (EMCL), 25.
3. Ali Akbar Vakili, M. B. (2024). Identifying Knowledge Management Infrastructures to Reduce Employee Mental Absenteeism through Data Mining Techniques. International Journal of Knowledge Processing Studies (KPS), 12.
4. Dalia Alzu'bi1 · Mwaffaq El Heis2 · Anas Ratib Alsoud3 · Mothanna Almahmoud8 · Laith Abualigah3, 4. (2024). Classification Model for Reducing Absenteeism of Nurses at Hospitals Using Machine Learning and Artificial Neural Network Techniques. Springer, 13.
5. Demudu Naganaidu, Z. M. (2022). Prediction of Absenteeism at Work with Multinomial Logistic Regression Model. Advances and Applications in Mathematical Sciences , 11.
6. Dr. Sujatha Kamepalli, D. B. (2020). MACHINE LEARNING CLASSIFIER MODELS IN ANALYZING DISCIPLINE OF INDIVIDUALS BASED ON VARIOUS REASONS FOR ABSENTEEISM FROM WORK. PAIDEUMA JOURNAL, 7.
7. Fernando Timoteo FernandesI, I. D. (2021). Prediction of Absenteeism in Public School Teachers Using Machine Learning. Revista Saude Publications, 11.
8. Francis Bowen, C. G.-G. (2022). Revealing Underlying Factors of Absenteeism: A Machine Learning Approach. Frontier in Psychology, 15.
9. Gopal Nath, Y. W. (2022). Incorporating a Machine Learning Model into a Web-Based Administrative Decision Support Tool for Predicting Workplace Absenteeism. MDPI (information), 14.
10. Heejoo Park, J. S.-h. (2024). Comparison of the Association Between Presenteeism and Absenteeism among Replacement Workers and Paid Workers: Cross-sectional Studies and Machine Learning Techniques. OSHRI, 7.
11. Isabel Herrera Montano, M. L.-C. (2020). Predicting Absenteeism and Temporary Disability Using Machine Learning: A Systematic Review and Analysis. Journal of Medical Systems , 11.
12. L. Mukli, A. R. (2022). Predicting and Analyzing Student Absenteeism Using Machine Learning Algorithms. ИНТЕГРАЦИЯ ОБРАЗОВАНИЯ. Т. 26, № 2. 2022, 13.
13. Luiz Henrique Salazar 1, A. F. (2020). Using Different Models of Machine Learning to Predict Attendance at Medical Appointments. Journal of Information Systems Engineering and Management , 11.
14. Natalie Lawrance, G. P.-A. (2021). Predicting Employee Absenteeism for Cost-Effective Interventions. ELSEVIER, 10.
15. Syed Atif Ali Shah, I. U.-K. (2020). An Enhanced Deep Neural Network for Predicting Workplace Absenteeism. WILEY (Hindawi), 12.
16. Trivedi, H. (2018). Explaining Absenteeism at Workplace Predicted by a Neural Network. International Conference on Artificial Intelligence and Business Computing (ABCs), 5.
17. Vanessa S. Araujo, T. S. (2019). A Hybrid Approach of Intelligent Systems to Help Predict Absenteeism at Work in Companies. Springer Nature Switzerland, 13.