Exploring Machine Learning Models for Predicting Employee Productivity

Fall 2023 CSC-322: Molly Nelson and Johana Papa

Abstract

In the pursuit of operational efficiency and competitive advantage, organizations prioritize understanding and forecasting factors influencing employee productivity. This study leverages a dataset sourced from Kaggle [1], focusing on employees within a garment manufacturing company. The following paper evaluates four machine learning models: Decision Trees, Neural Networks, Linear Regression, and Random Forest, assessing their efficacy in predicting actual productivity.

The Decision Tree algorithm, chosen for its ability to capture complex relationships, demonstrates favorable performance with a Mean Squared Error (MSE) of 0.0294, and an R-squared value of 0.0552. A Neural Network model is also implemented with the dataset, featuring three layers and using the sigmoid and the linear activation functions. The MSE of this model is 0.0238 and the R-squared is 0.3041 The next model used is a Multivariate Linear Regression that yielded an MSE of 0.0252 and an R-squared value of 0.1866. The final model implemented is Random Forest, a widely used ensemble learning algorithm based on Decision Trees. This model is explored in two versions, the first using the default hyperparameters yielding an MSE of 0.0136, and an R-squared score of 0.5611. The second version is tuned through a grid search and presents an improved MSE of 0.0130. However, it produces a slightly lower R-squared score of 0.5805

The comparative analysis highlights the Random Forest model as the best performer based on the R-squared metric and the MSE value. Drawing from insights from previous research, the study concludes with suggestions for future exploration of the dataset using the XGBoost model. This study has the potential to contribute valuable insights for organizational strategies, paving the way for enhanced employee productivity predictions.

Introduction

As organizations and businesses strive for operational efficiency to gain a competitive advantage, the ability to forecast and understand the drivers of productivity is vital. Multiple factors can impact an employee's productivity, such as monetary incentives, the hours worked, overtime, an interruption in production, the employee's department, and the number of workers on the employee's team. The dataset used in this paper was found on Kaggle [1] and gathered specifically from employees at a garment manufacturing company. The outcomes of this study have the potential to inform future organizational strategies within companies. The primary goal of this project is to develop a machine-learning model that can best predict employee performance scores based on a set of relevant factors. The following sections in this paper will discuss previous research in the area, the machine learning models used on the dataset, and an interpretation of the results gained after running the models.

Previous Work

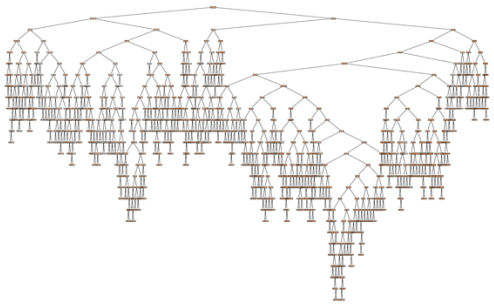
Another research paper by Muttineni Sai Rohith also aims to predict employees' productivity using the same dataset that we did [2]. However, while they also used the Random Forest model and found similar R-squared values, this paper also used another model called XGBoost and found that their model performed the best with an R-squared value of 0.2567 and a mean squared error of 0.01399.
Another paper by Anu Singh Lather also aimed to predict employee performance [3]. However, they used a different dataset with variables such as the level of schooling, socio-economic status, and psychological factors of employees. They used supervised learning techniques, such as

SVM, Random Forest, Naive Bayes, Logistic Regression, and Neural Networks. The SVM model proved to have the best accuracy among the five models.
Yet another paper by Mohamed Awada, explored the same goal [4]. Their methodology was very different from our model and the other models implemented by the researchers of the previous papers. They performed an experimental study with 48 volunteers and measured various physiological factors using sensors on the participants' wrists and chests. They then used their collected data and ran it through various Machine Learning Models. Ultimately, they also found that the XGBoost model had better predictive accuracy.

Decision Trees

We decided to implement a Decision Tree algorithm because of its ability to capture complex relationships between the input features and the target variable [5].

For the Decision Tree algorithm, all the features of the dataset are used. The dataset is preprocessed by combining all the relevant feature columns into the input matrix X and extracting the target variable y which measures actual productivity. A train-test split is performed with 70% of the data used for training and 30% for testing. A decision tree regressor is employed with a specified random state defined for reproducibility purposes. The model is trained on the training set (X_train, y_train). The model is then evaluated using a couple of metrics, including MSE, RMSE, and the R-squared metric on the test set (X_test, y_test). The calculated metrics provide insight into the accuracy of the predictive model. The decision tree structure is visualized using the plot_tree function from the tree module. Additionally, a Graphviz DOT file (tree_structure.dot) is exported for a detailed representation of the tree structure. This visualization makes it easier to interpret the results of the Decision Tree.



The Decision Tree algorithm performs pretty well in all the metrics that it is tested on. The RMSE is 0.1710, the MSE is 0.0294 and the R-squared value is 0.0552.

Neural Networks

Our Neural Network model comprises of three layers. The first is an input layer with 9 nodes to match the number of input variables. The hidden layer has 64 nodes and uses the sigmoid activation function. Finally, the output layer has 32 nodes as well as linear activation [6]. Since the data is already normalized between 0 and 1, the sigmoid activation in the hidden layers helps propagate these normalized values through the network. The sigmoid function, also known as the logistic function, is a mathematical function that has an S-shaped curve and maps any real-valued number to a value between zero and one [7]. The sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

The Linear Activation function represents a straight line, resulting in a linear relationship between the input and output. This means that the output of the function is directly proportional to the input. Unlike the Sigmoid function, the linear activation function does not alter the scale or shape of the input. It simply scales the input by a constant factor of 1. It is defined as:

$$f(x) = x$$

The target variable, the actual_productivity, is a percentage between 0 and 1. Since this is a continuous value, the linear activation function

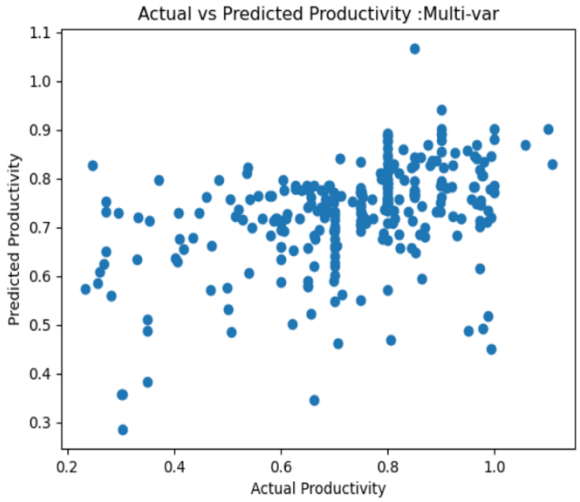
allows the network to output values in the same range as the target variable.

The model is compiled using the Adam optimizer and MSE loss function. The model was tested using three different optimizers: Adam, Stochastic Gradient Descent (SGD), and Adagrad. Adam, an adaptive optimization algorithm, combines the benefits of both momentum and RMSprop. It adapts the learning rates individually for each parameter, providing efficient and robust convergence across a variety of tasks. The SGD optimizer updates model parameters by considering the gradient of the loss concerning each parameter. It is computationally efficient and widely used, especially in large-scale datasets, but may require careful tuning of the learning rate. The final Optimizer Adagrad adapts the learning rate for each parameter based on the historical squared gradients. It performs well in sparse data scenarios by assigning larger learning rates to infrequently occurring features [8].

The Adam optimizer was found to have a significantly lower loss than the other two. The learning rate for the optimizer is 0.02. After testing multiple learning rate sizes (0.1, 0.2, 0.01, 0.02, 0.001, and 0.0001), this learning rate was found to minimize the model loss as well. The model is trained over 30 epochs with a batch size of 32, utilizing a 70-30 train-test split of the dataset. It should be noted that different epoch sizes were also tested to find the optimal number. The evaluation is based on MSE, and the coefficient of determination (R-squared) is computed for a more comprehensive assessment. The MSE of the neural network is 0.0238 and the R-squared value is 0.3041

Linear Regression

A Multivariate Linear Regression model is employed to model the relationship between multiple independent variables and the target variable [9], which is in this case, actual productivity. The same dataset is used and the following features are selected for the independent variables: 'team', 'idle_men', 'incentive', 'over_time', 'no_of_workers', 'no_of_style_change', 'wip', 'smv', and 'targeted_productivity' (X). The target variable is the 'actual_productivity'. The data is then split into 70-30 training and testing sets using the train_test_split function. The model is instantiated using the LinearRegression class from scikit-learn. It is trained on the training data using the fit method. The MSE (MSE) of the model is 0.0252. The Absolute Squared Error is 0.1153. Finally, the R-squared value is 0.1866. The following depicts a scatterplot of the actual productivity (X-axis) versus the predicted productivity (Y-axis).

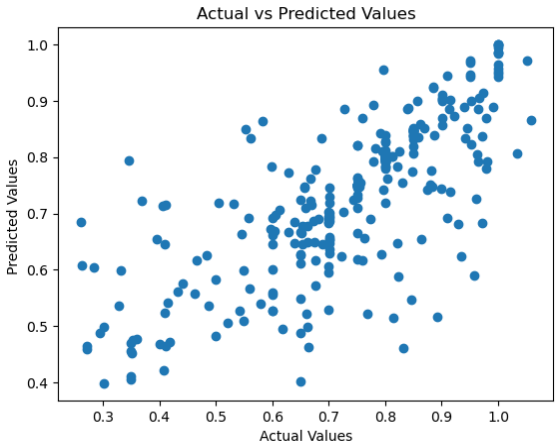


Random Forest

A Random Forest is a commonly used machine-learning algorithm that combines the output of multiple decision trees to reach a single result [10]. For the purpose of our project, two versions of the random forest algorithm are implemented.

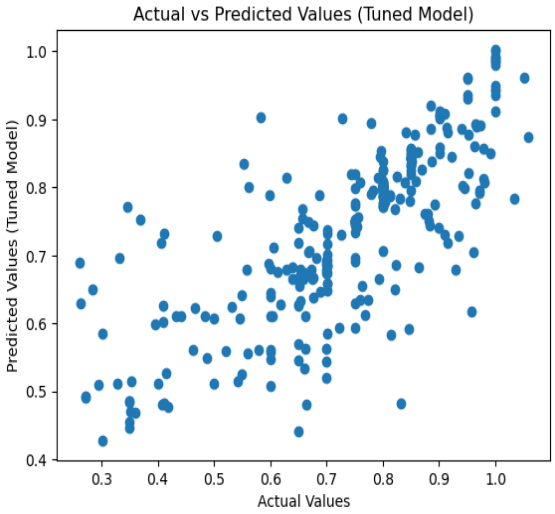
The first version is deployed with a Random Forest Regressor chosen as the base regression model using the default hyperparameters. The model uses 1000 decision trees to get to a singular result and a random seed for reproducibility purposes. The dataset is split into 70% training and 30% testing and metrics such as MSE, RMSE, and the R-squared are implemented to test the accuracy of this model. The model's root mean squared error is 0.117, MSE is 0.0136 and the R-squared score is

0.5611. A scatter plot visualizes the relationship between the actual values and the predicted values. Each point in the plot corresponds to a data point, with the x-coordinate being the actual value and the y-coordinate being the predicted value. The points of this dataset form an almost diagonal line, which indicates a relatively close match between the actual and predicted values.

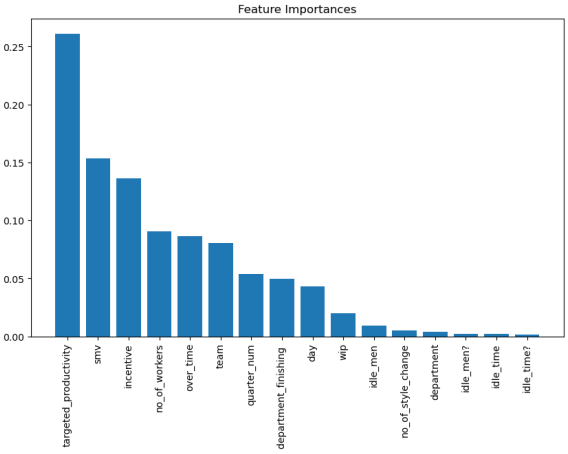


To try and get a more accurate result, a second version of the random forest regressor is implemented. A grid search is conducted to explore various combinations of hyperparameter values, aiming to identify the configuration that maximizes the predictive performance. The hyperparameters under consideration included the number of decision trees in the forest (this time small numbers of decision trees are tested), the maximum depth of each decision tree, the minimum number of samples required to split an internal node, and the minimum number of samples required to be at a leaf node [11]. The results obtained are 4 samples required to be at the leaf node, 100 decision trees used, no maximum depth, and 10 samples to split an internal node. The same metrics are used to test accuracy. The model's root MSE is 0.1142, MSE is 0.0130 and the R-squared score is 0.5805. This model performs better based on the Root MSE, the MSE and R-squared score.

A similar graph is also plotted to show the relationship between the actual values and the new predicted values.



Additionally, feature importances of the tuned version are computed and visualized by a bar plot. This visualization clearly shows that the targeted_productivity feature has the most impact out of all the variables, beating the runner-up (smv) by almost two-fold.



Conclusion

Our study investigated the performance of four machine learning models, Decision Trees, Neural Networks, Linear Regression, and Random Forest. Each model offered unique insights into the dataset, with Decision Trees capturing complex relationships, Neural Networks offering a nuanced approach, Linear Regression providing simplicity, and Random

Forest showcasing the strength of ensemble learning.

We ran the dataset through each model intending to predict the actual employee productivity. We used different metrics to test for the accuracy of the model and eventually decided to compare the models by the R-squared metric, as we discovered it gave the best insights for our particular dataset. The best-performing model based on this metric was the tuned Random Forest model, with an R-squared score of 0.5805. The worst-performing model was the Decision Tree with an R-squared score of 0.0552.

In terms of future work, after reading about previous projects in this area, we would like to explore and implement the XGBoost model to see if it will have a better performance concerning the R-squared metric.

References

[1] S. Siri, "Productivity prediction of garment employees," Kaggle, https://www.kaggle.com/datasets/ishadss/productivity-prediction-of-garment-employees (accessed Dec. 1, 2023).

[2] T. A. Team, "Productivity prediction of employees using machine learning python," Towards AI, https://towardsai.net/p/l/productivity-prediction-of-employees-using-machine-learning-python (accessed Dec. 1, 2023).

[3] Anu Singh Lather Delhi Technological University et al., "Prediction of employee performance using Machine Learning Techniques: Proceedings of the 1st International Conference on Advanced Information Science and System," ACM Other conferences, https://dl.acm.org/doi/abs/10.1145/3373477.3373696 (accessed Dec. 1, 2023).

[4] M. Awada, B. Becerik-Gerber, G. Lucas, and S. C. Roll, "Predicting Office Workers' Productivity: A machine learning approach integrating physiological, behavioral, and psychological indicators," MDPI, https://www.mdpi.com/1424-8220/23/21/8694 (accessed Dec. 1, 2023).

[5] A. Thevapalan and J. Le, "R decision trees tutorial: Examples & code in R for Regression & Classification," DataCamp, https://www.datacamp.com/tutorial/decision-trees-R (accessed Dec. 1, 2023).

[6] "Tensorflow 2 quickstart for beginners : Tensorflow Core," TensorFlow, https://www.tensorflow.org/tutorials/quickstart/beginner (accessed Dec. 1, 2023).

[7] M. Saeed, "A gentle introduction to sigmoid function," MachineLearningMastery.com, https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/ (accessed Dec. 1, 2023).

[8] A. Gupta, "A comprehensive guide on Optimizers in deep learning," Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/ (accessed Dec. 1, 2023).

[9] M. Badole, "Mastering multiple linear regression: A comprehensive guide," Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/05/multiple-linear-regression-using-python-and-scikit-learn/ (accessed Dec. 1, 2023).

[10] "What is Random Forest?," IBM, https://www.ibm.com/topics/random-forest (accessed Dec. 1, 2023).

[11] W. Koehrsen, "Hyperparameter tuning the random forest in python," Medium, https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74 (accessed Dec. 1, 2023).