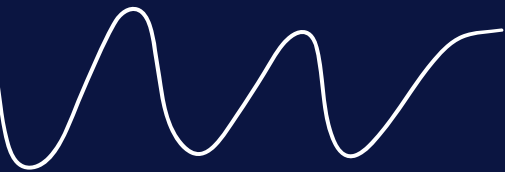


PROJECT REPORT



GARMENT WORKER PRODUCTIVITY




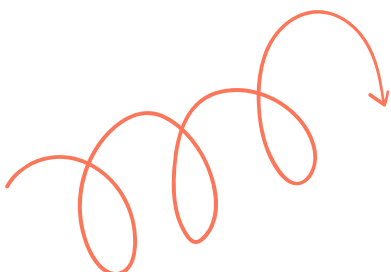
Evangeline Suciadi

<https://github.com/Eva918/Worker-Productivity-Prediction>

TABLE OF CONTENTS



DATA PREPARATION	1
EXPLORATORY DATA ANALYSIS	2
DATA PREPROCESSING	7
MACHINE LEARNING MODEL	9
 EVALUATION	10



DATA PREPARATION

EXPLANATION

The data preparation process begins with loading the dataset into a Pandas DataFrame using `pd.read_csv()`. This dataset contains information about productivity in a garment factory, including variables like date, department, targeted productivity, and actual productivity. Initial inspections of the dataset using `df.dtypes` and `df.info()` reveal that the dataset has 15 columns with different data types, such as categorical (object) for fields like date and department, and numerical (float64 and int64) for fields like targeted_productivity, over_time, and wip. The dataset comprises 1197 rows, and some columns, such as wip (Work in Progress), contain missing values.

To address missing values, the `isnull().sum()` function is used to identify the extent of the problem. The analysis shows that only the wip column has missing values, with 506 entries missing, while all other columns are complete. Missing data is handled by replacing the missing values in the wip column with the median of the column. The median is chosen because it is less affected by outliers compared to the mean, ensuring a more robust and reliable imputation. After filling in the missing values, the dataset is rechecked using `isnull().sum()` to confirm that there are no longer any missing values.

By completing this process, the dataset is now fully cleaned and prepared for analysis or modeling. The use of median imputation ensures that the statistical properties of the wip column are preserved while maintaining data integrity. This step eliminates potential errors caused by missing values and ensures that the data is ready for machine learning algorithms or further statistical exploration. The cleaned dataset now provides a strong foundation for deriving meaningful insights and building accurate models.

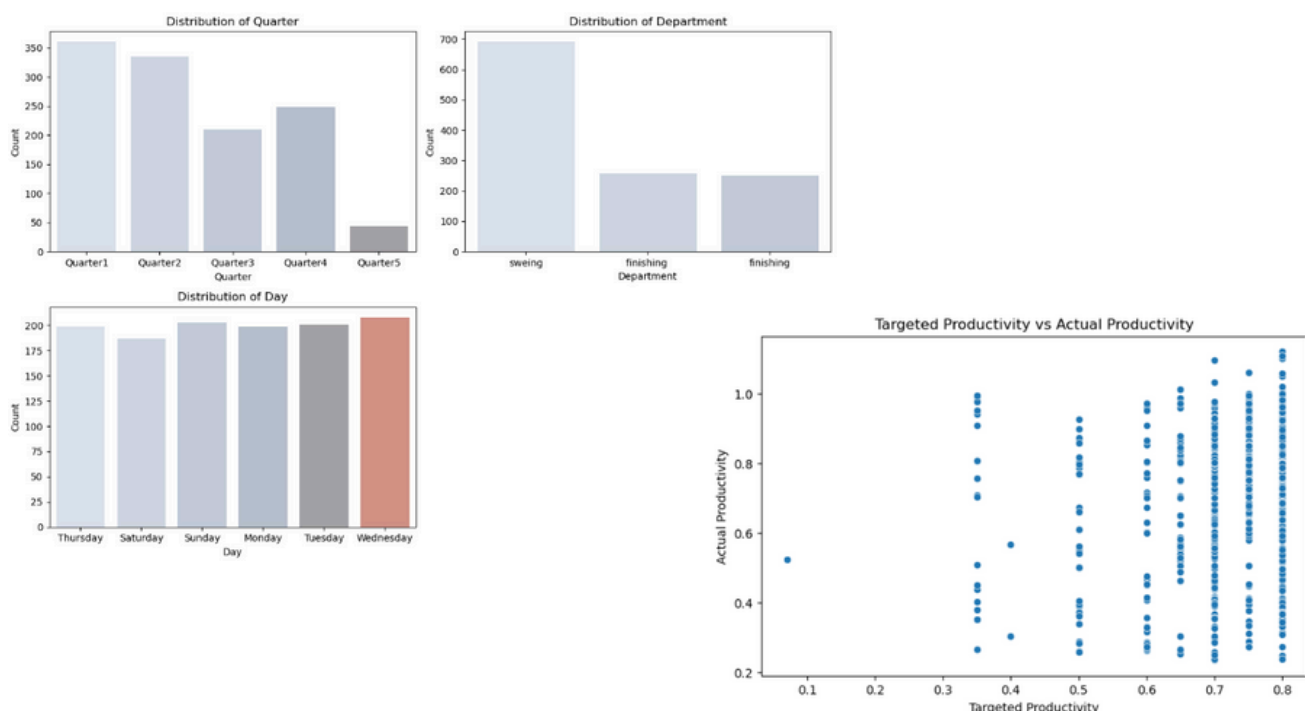
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   date                                  1197 non-null   object
1   quarter                              1197 non-null   object
2   department                           1197 non-null   object
3   day                                   1197 non-null   object
4   team                                  1197 non-null   int64
5   targeted_productivity                 1197 non-null   float64
6   smv                                   1197 non-null   float64
7   wip                                   691 non-null    float64
8   over_time                            1197 non-null   int64
9   incentive                            1197 non-null   int64
10  idle_time                             1197 non-null   float64
11  idle_men                             1197 non-null   int64
12  no_of_style_change                   1197 non-null   int64
13  no_of_workers                        1197 non-null   float64
14  actual_productivity                  1197 non-null   float64
dtypes: float64(6), int64(5), object(4)
memory usage: 140.4+ KB
```

EXPLORATORY DATA ANALYSIS

EXPLANATION

The distribution of quarter (top-left chart) indicates that Quarter 1 and Quarter 2 have the highest counts, suggesting these periods are more prominent in the dataset. Quarter 3 and Quarter 4 show moderate frequencies, while Quarter 5 has a significantly lower count. This pattern may reflect seasonal trends, uneven data collection, or variations in activity levels across quarters. The distribution of department (top-right chart) highlights a clear dominance of the sewing department, which accounts for the majority of the data. In contrast, the finishing department is represented less frequently, appearing twice in the plot. The duplicate labels for finishing may indicate inconsistencies or errors in the dataset that require cleaning before further analysis. The distribution of day (bottom-left chart) shows a nearly even count across all days of the week, with a slight emphasis on Wednesday. This even distribution suggests that the dataset covers activities or events that are consistently distributed throughout the week, without significant weekday or weekend bias.

Meanwhile, This scatter plot highlights the general alignment between targeted and actual productivity while also revealing inconsistencies. The positive correlation indicates that higher targets typically result in better actual performance, but deviations from this pattern point to potential inefficiencies or challenges. Further investigation into the factors affecting these deviations could provide actionable insights to improve overall productivity.



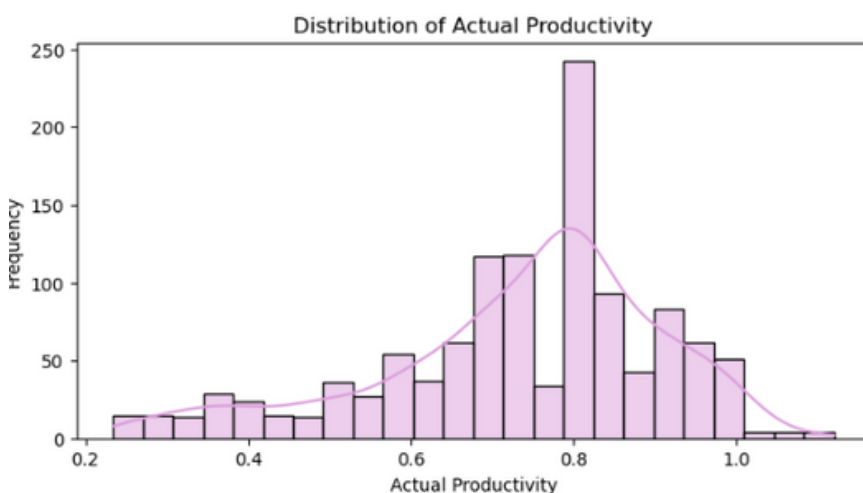
EXPLORATORY DATA ANALYSIS

DISTRIBUTION

The histogram provides a detailed visualization of the distribution of actual productivity in the dataset, accompanied by a kernel density estimate (KDE) curve for a smoother representation of the distribution's shape. The x-axis represents actual productivity values ranging from approximately 0.2 to 1.0, while the y-axis shows the frequency of occurrences within each range. The plot is styled with a plum-colored histogram and a pink KDE curve to enhance clarity and interpretability.

The distribution of actual productivity appears to be slightly left-skewed (negatively skewed) but is close to symmetric. Most of the data is concentrated around productivity values of 0.7 to 0.8, with a clear peak near 0.8. The frequency decreases gradually as we move toward lower productivity values (0.2 to 0.6), indicating that instances of low productivity are relatively uncommon. The KDE curve further emphasizes the slight skew, with the distribution's tail extending more toward the lower productivity range.

The histogram suggests that actual productivity is often close to its maximum potential, as the majority of the values are clustered near the higher end (0.7 to 0.8). This indicates that most operations in the dataset are achieving high productivity levels. However, the presence of a few lower productivity instances, while less frequent, may require further analysis to identify potential inefficiencies or challenges affecting these cases.



IS IT SKEWED?

Overall, the slight left skew in the data highlights that while the dataset exhibits a trend of high performance, a small proportion of lower productivity cases exists. Investigating these outliers can provide valuable insights into areas for improvement and help enhance overall productivity outcomes.

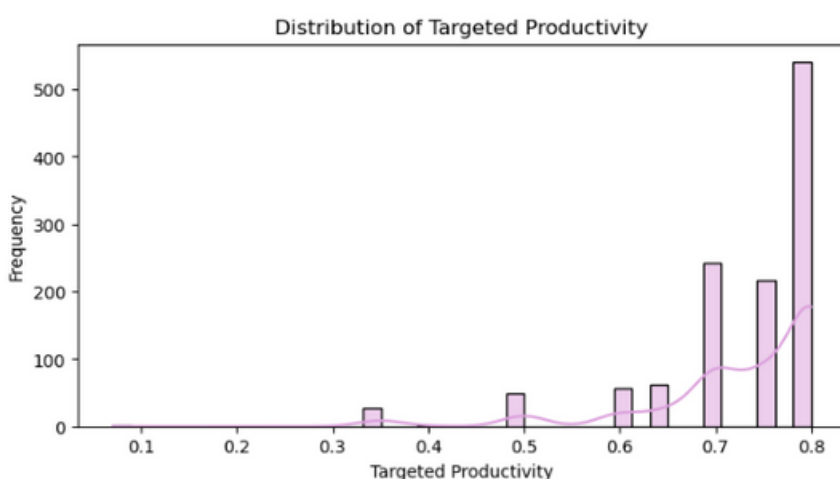
EXPLORATORY DATA ANALYSIS

DISTRIBUTION

The histogram illustrates the frequency of values in the `targeted_productivity` dataset. Most observations are concentrated near the 0.8 mark, indicating a clustering of high productivity values. The distribution tapers off toward smaller values, suggesting fewer instances of low productivity. The KDE overlay further emphasizes this trend by smoothing the data, revealing a peak near 0.8 and a gradual decline toward lower values.

From visual inspection, the distribution appears to be left-skewed (negatively skewed). The tail extends toward smaller values on the left, implying that while most data points represent high productivity, a minority of significantly lower productivity values pulls the mean toward the lower end. This skewness suggests the presence of outliers or unique cases within the dataset.

To confirm this observation, it would be helpful to calculate the skewness value using a method like `scipy.stats.skew`. Understanding the reasons for low productivity instances could provide valuable insights, and further statistical analysis could help determine whether skewness impacts modeling assumptions. If necessary, data transformations (such as logarithmic or square transformations) might be applied to address the skewness.



IS IT SKEWED?

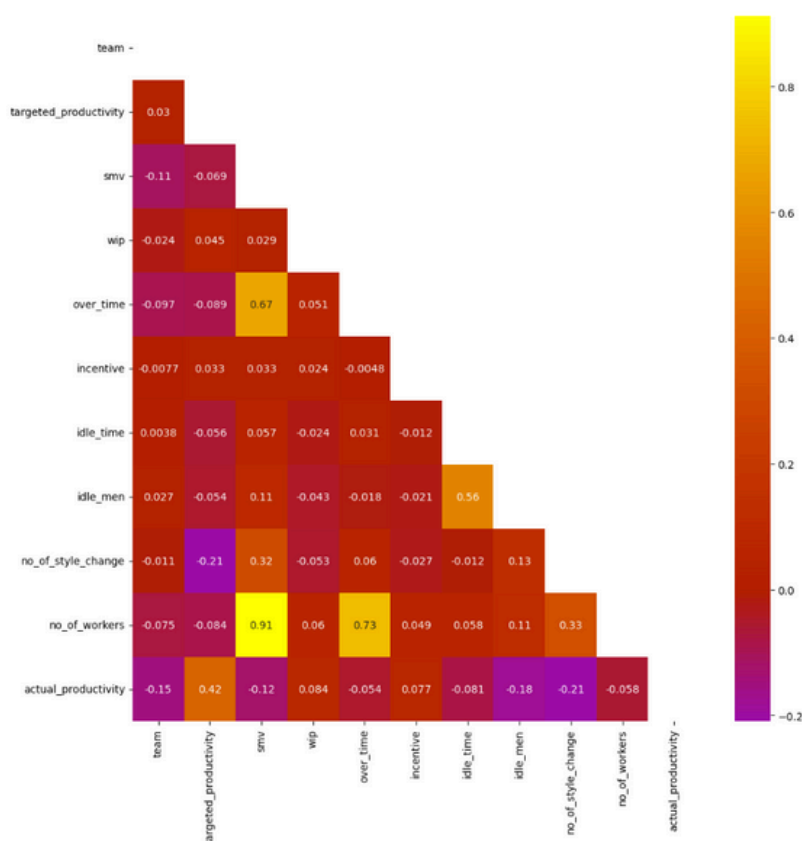
The left skewness implies that the data contains a cluster of high productivity values and a minority of significantly lower values. This might reflect systematic factors or special cases affecting the data.

EXPLORATORY DATA ANALYSIS

HEATMAP

The image contains a heatmap that visualizes the correlation matrix of various variables. It provides insights into the relationships between key factors affecting productivity. Notably, there is a strong positive correlation between `no_of_workers` and `wip` (work-in-progress) with a value of 0.91, indicating that an increase in the number of workers is directly associated with a rise in work-in-progress levels. Similarly, `overtime` and `no_of_workers` exhibit a positive correlation (0.73), suggesting that overtime tends to increase when more workers are involved. Moderate positive correlations are also observed, such as between `actual_productivity` and `smv` (0.42), implying that higher Standard Minute Value (SMV) tasks are linked to slightly higher productivity. On the other hand, variables like `idle_men` and `idle_time` are moderately correlated (0.56), which is expected since idle workers typically contribute to idle time.

The heatmap also highlights some negative correlations. For example, `actual_productivity` shows weak negative correlations with both `idle_time` (-0.12) and `idle_men` (-0.18), suggesting that increased idle time and idle workers slightly reduce productivity. Another interesting observation is the mild negative correlation between `overtime` and `actual_productivity` (-0.097), which could indicate that extended working hours might lead to diminishing productivity, possibly due to worker fatigue or inefficiencies.



EXPLORATORY DATA ANALYSIS

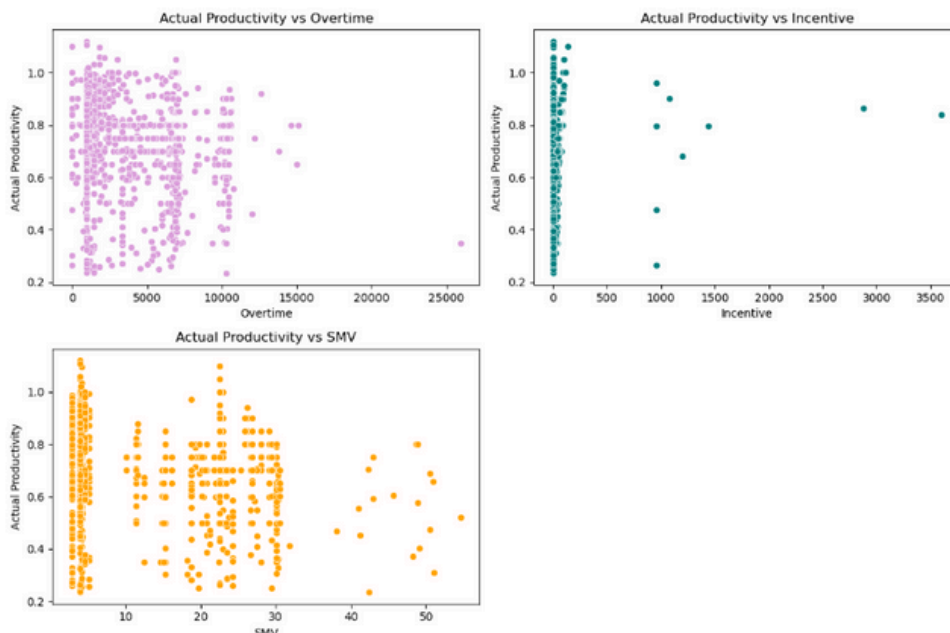
SCATTER PLOT

The image includes three scatter plots that examine the relationship between actual_productivity and other variables: overtime, incentive, and SMV. The first scatter plot, Actual Productivity vs. Overtime, shows a wide distribution with no strong trend, though there is a slight negative relationship where higher overtime appears to correlate with reduced productivity. This finding aligns with the negative correlation observed in the heatmap and may indicate the drawbacks of excessive overtime on worker performance.

The second plot, Actual Productivity vs. Incentive, reveals that most data points are clustered at lower incentive levels, with no clear relationship between incentives and productivity within this range. Interestingly, outliers at higher incentive levels do not consistently result in higher productivity, suggesting that increasing financial incentives may not always yield proportional improvements in productivity.

The third scatter plot, Actual Productivity vs. SMV, shows a more structured pattern, with distinct clusters of data points at specific SMV values. There is a positive trend indicating that tasks with higher SMV values tend to have higher productivity. This could suggest that tasks requiring more time are managed more efficiently, possibly due to better resource allocation or planning.

In conclusion, the heatmap and scatter plots provide valuable insights into the factors influencing productivity. While SMV positively impacts productivity, the relationship between overtime and incentives appears more complex, with diminishing returns observed in some cases. Further investigation into the optimal levels of overtime and effective incentive strategies is recommended. Additionally, exploring task-level differences within SMV clusters may provide actionable insights for improving overall productivity.



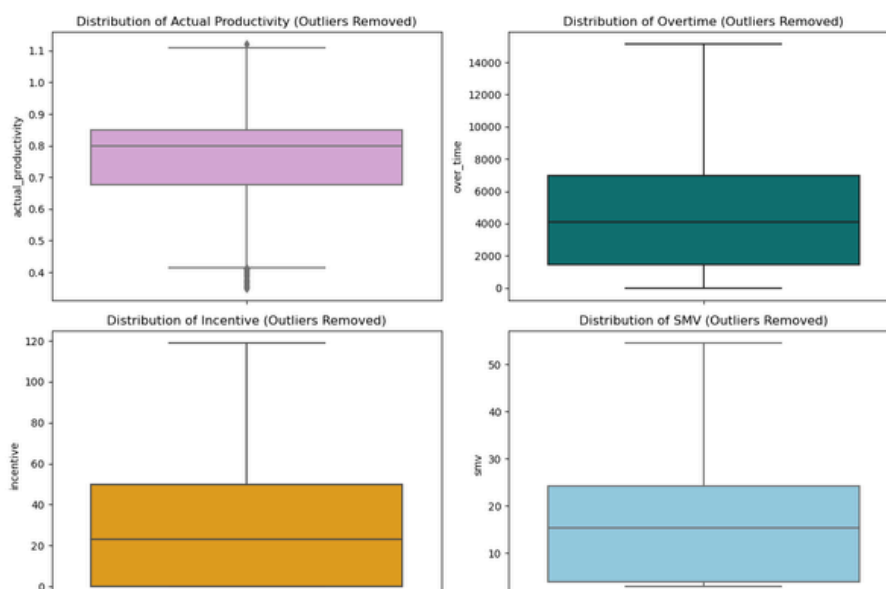
DATA PREPROCESSING

BOXPLOT, TREATING OUTLIERS

The initial boxplot for `actual_productivity` reveals that most values lie within a narrow range, with a few outliers at both the lower and upper ends. This indicates that while productivity is generally stable, there are occasional extreme cases of under- or over-performance. For `over_time`, the boxplot shows a wide range of values with several high outliers, suggesting that while most workers perform overtime within a typical range, there are exceptional cases of significantly extended overtime.

The incentive variable shows a heavily skewed distribution, with the majority of values concentrated at lower levels and a few high outliers. This suggests that higher incentives are uncommon, indicating that most workers receive standard or minimal additional compensation. The boxplot for `smv` shows a relatively uniform distribution, but with a few high outliers. These could represent unusually complex tasks requiring significantly more time than average. To address the influence of outliers, the Interquartile Range (IQR) method was applied. The first quartile (Q1) and third quartile (Q3) were calculated for each variable, and the IQR ($Q3 - Q1$) defined the range of typical values. Outliers were identified as values falling below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$. These values were removed from the dataset, reducing its original shape from (1197, 15) to (1133, 15). This indicates that 64 rows (5.4%) contained outliers.

The boxplots after outlier removal showed notable improvements. For `actual_productivity`, the cleaned data had a tighter distribution with no extreme values, confirming that the removed outliers were isolated cases. The `over_time` distribution became narrower, reflecting more representative overtime values. Similarly, the cleaned distribution of incentive focused on lower values, reducing skewness. For `smv`, the cleaned data eliminated extreme task durations, showcasing a more compact range of standard task times.



DATA PREPROCESSING

EXPLANATION

- Unique Values in Categorical Variables:

The quarter and department columns were inspected for unique values. It was observed that the department column had inconsistencies (e.g., "finishing " had a trailing space). To address this, `.str.strip()` was applied to ensure consistent formatting.

- Encoding Categorical Variables:

Several categorical variables were transformed into numerical representations. The quarter column was mapped to integers (e.g., "Quarter1" → 1). Similarly, the department column was mapped to integers (e.g., "sweing" → 1, "finishing" → 2). Additionally, the day column was mapped to integers representing days of the week (e.g., "Monday" → 1, "Tuesday" → 2).

- Feature Scaling:

Continuous numerical columns were standardized using `StandardScaler` to normalize the range of values. This step was crucial to ensure that features like `over_time` and `smv`, which have different numerical scales, do not disproportionately influence the model. The scaler was fit to the training data and then applied to the test data.

MACHINE LEARNING MODEL

EXPLANATION

Data Splitting

The dataset was split into training (80%) and testing (20%) subsets. The feature set (X) excluded the target variable `actual_productivity` and `date`, while the target variable (y) contained the `actual_productivity` values. This division ensured that the model was trained and tested on separate data to evaluate its generalization capabilities.

Hyperparameter Optimization

A parameter grid was defined for tuning the XGBoost Regressor. Key hyperparameters included:

- `n_estimators`: Number of trees in the model (100, 200, 300).
- `learning_rate`: Step size for weight updates (0.01, 0.05, 0.1).
- `max_depth`: Maximum depth of a tree (3, 6, 9).
- `subsample`: Fraction of samples used for training each tree (0.5 to 0.9).
- `colsample_bytree`: Fraction of features used for training each tree (0.5 to 0.9).

`RandomizedSearchCV` was used for hyperparameter tuning. It searched through 50 random parameter configurations using 5-fold cross-validation. Negative Mean Absolute Error (MAE) was used as the scoring metric to evaluate prediction accuracy.

Best Model and Parameters

After fitting 250 models, the best-performing hyperparameters were identified:

- `subsample`: 0.9 (90% of samples used per tree).
- `n_estimators`: 100 (uses 100 trees).
- `max_depth`: 6 (balances model complexity and risk of overfitting).
- `learning_rate`: 0.1 (moderate step size for weight updates).
- `colsample_bytree`: 0.5 (50% of features used per tree).

The best model, an optimized XGBoost Regressor, was configured with these parameters. This model was then used to make predictions on the test set (`X_test`).

EVALUATION

EVALUATION METRICS

1. Mean Absolute Error (MAE)

Value: 0.0708

MAE measures the average magnitude of errors between predicted and actual values. A MAE of 0.0708 indicates that, on average, the model's predictions are off by about 7% when predicting `actual_productivity`. Lower MAE values suggest better model performance.

2. Mean Squared Error (MSE)

Value: 0.0127

MSE calculates the average of squared differences between predicted and actual values. The squaring penalizes larger errors more heavily than smaller ones. A value of 0.0127 indicates that the model's squared errors are relatively low, suggesting reasonable predictive accuracy.

3. Root Mean Squared Error (RMSE)

Value: 0.1129

RMSE is the square root of MSE and expresses errors in the same units as the target variable (`actual_productivity`). An RMSE of 0.1129 implies that, on average, the model's predictions deviate from the actual values by approximately 11.3%. RMSE provides a more intuitive interpretation compared to MSE because it is in the same scale as the target variable.

4. R-squared (R^2)

Value: 0.5199

R^2 measures the proportion of variance in the target variable that is explained by the model. An R^2 of 0.5199 (approximately 52%) suggests that the model explains just over half of the variance in `actual_productivity`. While this indicates moderate predictive power, there is still room for improvement, as nearly 48% of the variance remains unexplained.

EVALUATION

FEATURE IMPORTANCE

This plot illustrates the feature importance of a model (likely XGBoost, based on the code). Feature importance here is determined by the weight, which represents the number of times a feature is used in the model's decision trees.

Key Observations:

1. Dominant Features:

- day (1101.0): This is the most important feature, indicating that it is frequently used by the model for making predictions.
- team (1081.0): Nearly as important as day, showing it plays a significant role in the model's decisions.
- quarter (953.0): Also contributes significantly.

2. Moderately Important Features:

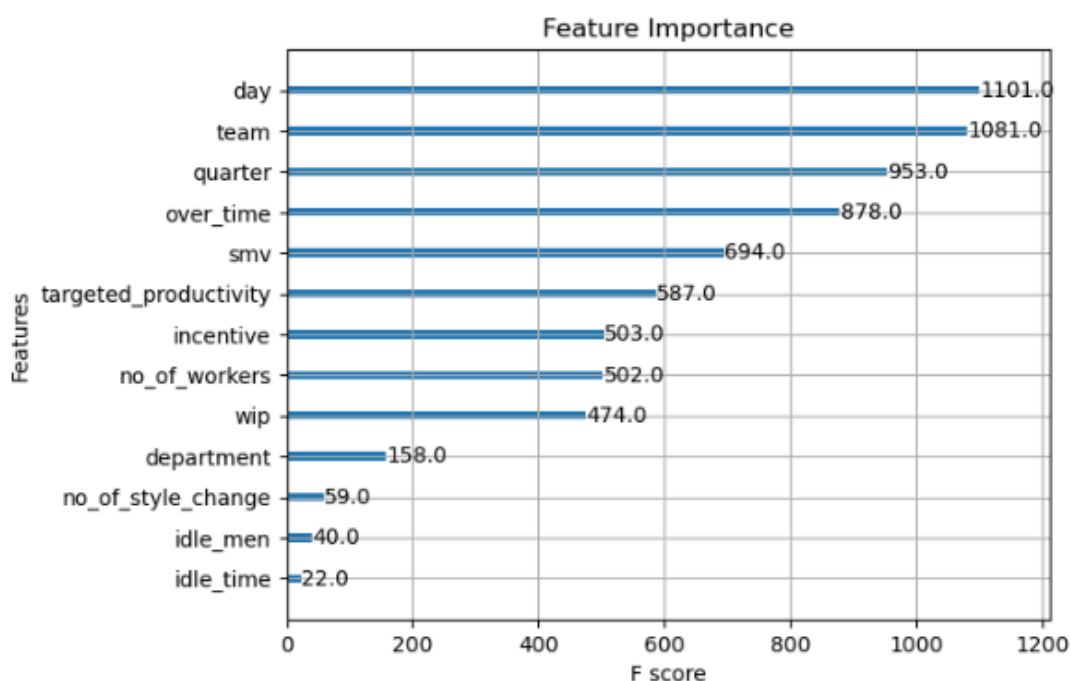
- over_time (878.0), smv (694.0), targeted_productivity (587.0): These features are also frequently used and thus contribute meaningfully to the predictions.

3. Less Important Features:

- incentive, no_of_workers, wip (approximately 500): These have moderate influence.
- department (158.0): Much less used compared to others.

4. Least Important Features:

- no_of_style_change (59.0), idle_men (40.0), idle_time (22.0): These features are rarely used and thus have minimal impact on predictions.



EVALUATION

LEARNING CURVE

This plot represents a learning curve for an XGBoost model, showing the relationship between the training set size and the model's error (Mean Squared Error or MSE) on both the training and validation datasets.

Key Observations:

1. Training Error (Blue Line):

- The training error is very low and increases slightly as the training set size grows.
- This is expected because with smaller datasets, the model overfits to the limited data, leading to very low errors. As the dataset grows, the model generalizes better, causing a slight increase in training error.

2. Validation Error (Orange Line):

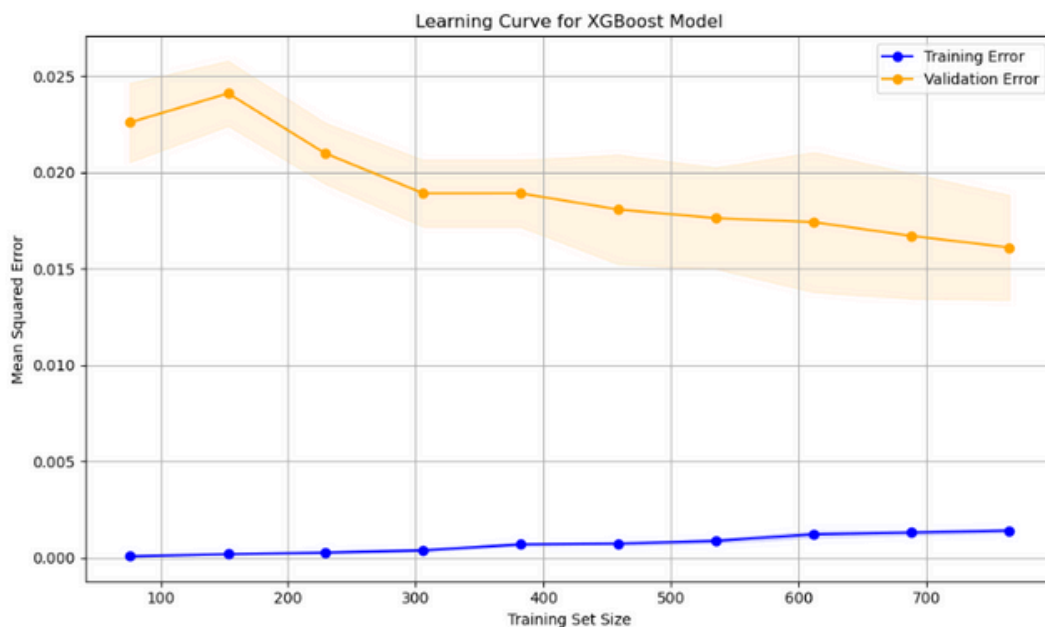
- The validation error starts relatively high with small training sets and gradually decreases as the training set size increases.
- This indicates that as the model sees more data, it learns better and generalizes more effectively, reducing validation error.

3. Convergence Trend:

- The gap between the training and validation errors decreases as the training set size grows, indicating that the model's performance on unseen data improves with more training data.
- However, the validation error still remains higher than the training error, which is common and indicates there is room for improvement.

4. Error Stabilization:

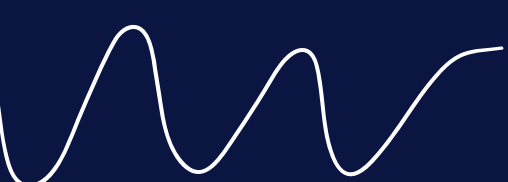
- Towards the right (larger training sizes), the validation error curve begins to flatten, suggesting the model is approaching its optimal performance with the given data and features.



PROJECT REPORT



THANK YOU



Evangeline Suciadi